

Human skin segmentation using correlation rules on dynamic color clustering

Rodrigo Augusto Dias Faria

THESIS SUBMITTED
TO THE
INSTITUTE OF MATHEMATICS AND STATISTICS
OF THE
UNIVERSITY OF SÃO PAULO
TO
OBTAIN THE TITLE
OF
MASTER IN SCIENCE

Program: Computer Science

Advisor: Prof. Dr. Roberto Hirata Jr

São Paulo, August 2018

Human skin segmentation using correlation rules on dynamic color clustering

This version of the thesis contains the corrections and changes suggested by the Examining Committee during the defense of the original work, performed on August 31st, 2018. A copy of the original version is available at the Institute of Mathematics and Statistics of the University of São Paulo.

Examining Committee:

- Prof. Dr. Roberto Hirata Jr (advisor) - IME-USP
- Prof. Dr. Manuel Menezes de Oliveira Neto - INF-UFRGS
- Prof. Dr. João Eduardo Kogler Junior - EP-USP

To my beloved wife and daughter.

Acknowledgements

I would like to express my sincere gratitude to my advisor, Prof. Dr. Roberto Hirata Jr, who has believed in my potential since the beginning of my research. His motivation, patience, and enthusiasm in teaching me were essential during the accomplishment of all this work. More than that, this work would not have been done without his flexibility in admitting me as a part-time student to continue to work in the industry. We all know how difficult it is to reconcile these activities, but his comprehension is something that will be marked forever in my life. Thank you so much!

No matter how tired, discouraged, hopeless she was, but my wife Tamires gave me unconditional support. She accompanied me at absolutely every moment. She is the most incredible woman I have ever met. And, to my happiness, she gave me the greatest gift of my life: my daughter Júlia. I love you unconditionally.

I would also like to thank all my family, for the example they have always given me, for their care and dedication. In particular, I thank my mother for showing me the importance of education in my entire life. I have no doubt that your choices have always been thought the best for me and my siblings.

I can not forget two great friends I made during these years, among others who were part of this journey, Ana Lucia and Luiz Medina, for having spent so much time together and for providing discussions of the highest level, of the most varied subjects. They were also able to teach me the beauty of the proof by induction. I will always remember those days!

I should also remember the Professors who were members of my Examining Committee, Prof. Dr. Roberto Marcondes Cesar Junior and Prof. Dr. Peter Sussner, for kindly giving up their limited time to grant me valuable suggestions and corrections that have brought me to this result. In particular, to Prof. Dr. Roberto Marcondes for having indicated to me the reading of the paper that served as the basis for all this project.

I also thank the University of São Paulo for giving me the opportunity and for providing structure and excellence in teaching for the entire community. I may say many thanks also to everyone in the university who support us including, but not limited to, Professors, secretaries, co-workers, and colleagues.

And, last but not least, I would like to be grateful to Inatel and Qualcomm, the companies I have worked for during the years of the realization of this work. They were extremely supportive of me in all the difficult moments I faced during the course of this project.

You were all key players in this achievement!

Resumo

FARIA, R. A. D. **Segmentação de pele humana usando regras de correlação baseadas em agrupamento dinâmico de cores.** Dissertação (Mestrado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2018.

A pele humana é constituída de uma série de camadas distintas, cada uma das quais reflete uma porção de luz incidente, depois de absorver uma certa quantidade dela pelos pigmentos que se encontram na camada. Os principais pigmentos responsáveis pela origem da cor da pele são a melanina e a hemoglobina. A segmentação de pele desempenha um papel importante em uma ampla gama de aplicações em processamento de imagens e visão computacional. Em suma, existem três abordagens principais para segmentação de pele: baseadas em regras, aprendizado de máquina e híbridos. Elas diferem em termos de precisão e eficiência computacional. Geralmente, as abordagens com aprendizado de máquina e as híbridas superam os métodos baseados em regras, mas exigem um conjunto de dados de treinamento grande e representativo e, por vezes, também um tempo de classificação custoso, que pode ser um fator decisivo para aplicações em tempo real. Neste trabalho, propomos uma melhoria, em três versões distintas, de um novo método de segmentação de pele baseado em regras que funciona no espaço de cores YCbCr. Nossa motivação baseia-se nas hipóteses de que: (1) a regra original pode ser complementada e, (2) pixels de pele humana não aparecem isolados, ou seja, as operações de vizinhança são levadas em consideração. O método é uma combinação de algumas regras de correlação baseadas nessas hipóteses. Essas regras avaliam as combinações de valores de crominância Cb, Cr para identificar os pixels de pele, dependendo da forma e tamanho dos agrupamentos de cores de pele gerados dinamicamente. O método é muito eficiente em termos de esforço computacional, bem como robusto em imagens muito complexas.

Palavras-chave: detecção de pele, segmentação de pele humana, modelo de cores YCbCr, regras de correlação, agrupamento dinâmico de cores.

Abstract

FARIA, R. A. D. **Human skin segmentation using correlation rules on dynamic color clustering.** Thesis (Master's Degree) - Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2018.

Human skin is made of a stack of different layers, each of which reflects a portion of impinging light, after absorbing a certain amount of it by the pigments which lie in the layer. The main pigments responsible for skin color origins are melanin and hemoglobin. Skin segmentation plays an important role in a wide range of image processing and computer vision applications. In short, there are three major approaches for skin segmentation: rule-based, machine learning and hybrid. They differ in terms of accuracy and computational efficiency. Generally, machine learning and hybrid approaches outperform the rule-based methods but require a large and representative training dataset and, sometimes, costly classification time as well, which can be a deal breaker for real-time applications. In this work, we propose an improvement, in three distinct versions, of a novel method for rule-based skin segmentation that works in the YCbCr color space. Our motivation is based on the hypotheses that: (1) the original rule can be complemented and, (2) human skin pixels do not appear isolated, i.e. neighborhood operations are taken into consideration. The method is a combination of some correlation rules based on these hypotheses. Such rules evaluate the combinations of chrominance Cb, Cr values to identify the skin pixels depending on the shape and size of dynamically generated skin color clusters. The method is very efficient in terms of computational effort as well as robust in very complex images.

Keywords: skin detection, human skin segmentation, YCbCr color model, correlation rules, dynamic color clustering.

Contents

| | |
|---|-------------|
| List of Acronyms | xi |
| List of Symbols | xiii |
| List of Figures | xvii |
| List of Tables | xix |
| 1 Introduction | 1 |
| 1.1 Motivation | 4 |
| 1.2 Objectives | 4 |
| 1.3 Contributions | 5 |
| 1.4 Organization | 5 |
| 2 Related Work | 7 |
| 3 Theoretical Background | 13 |
| 3.1 Digital image | 13 |
| 3.2 Basic relationship between pixels | 14 |
| 3.2.1 Neighborhood | 14 |
| 3.2.2 Connectivity | 15 |
| 3.2.3 Image boundaries | 15 |
| 3.3 Image histogram | 16 |
| 3.4 Image segmentation | 17 |
| 3.4.1 Thresholding | 18 |
| 3.5 Color models | 19 |
| 3.5.1 Munsell color model | 19 |
| 3.5.2 CIE color model | 19 |
| 3.5.3 RGB color model | 22 |
| 3.5.4 CMY color model | 23 |
| 3.5.5 Color models of the YUV family | 24 |
| 3.5.6 Color models of the HSI family | 24 |

| | |
|---|-----------|
| 4 Correlation rules and proposed extension | 29 |
| 4.1 Correlation rules on YCrYCb colormap | 29 |
| 4.2 Original method | 30 |
| 4.3 Complementary method | 34 |
| 4.4 Neighborhood extended method | 35 |
| 4.5 Heuristics to fix neighborhood extended method | 36 |
| 5 Experimental Evaluation | 39 |
| 5.1 Datasets | 39 |
| 5.1.1 SFA | 39 |
| 5.1.2 Pratheepan | 40 |
| 5.1.3 HGR | 40 |
| 5.1.4 Compaq | 40 |
| 5.2 Evaluation measures | 41 |
| 5.3 Rule-based experiments | 43 |
| 5.4 Experiments with heuristics to fix extended neighborhood method | 49 |
| 5.5 Parameters tuning via a grid search strategy | 50 |
| 6 Conclusions | 63 |
| 6.1 Final considerations | 63 |
| 6.2 Future work | 64 |
| A Trapezoids Parameters Tuning Results | 65 |
| Bibliography | 81 |

List of Acronyms

| | |
|--------|--|
| AR | Aleix and Robert Face Database |
| CIE | Commission Internationale de l'Eclairage |
| CMY | Cyan, Magenta and Yellow |
| FERET | Face Recognition Technology database |
| GIS | Geographic Information System |
| HGR | Hand Gesture Recognition database |
| HSI | Hue, Saturation, Intensity |
| HSL | Hue, Saturation, Lightness |
| HSV | Hue, Saturation, Value |
| IHLS | Improved, Hue, Luminance and Saturation |
| LUT | Look-UP Table |
| NTSC | National Television System Committee |
| PAL | Phase Alternating Line |
| RGB | Red, Green and Blue |
| SCT | Scitex Continuous Tone Format |
| SECAM | Sequential Color with Memory |
| SFA | Skin of FERET and AR Database |
| UCS | Uniform Chromaticity Scale |
| UV | Ultra Violet |
| VISAPP | International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications |
| YIQ | Luma, Hue and Saturation |
| YUV | Luma and Chrominance |

List of Symbols

| | |
|-------------|--|
| $f(x, y)$ | Intensity function of an image |
| H | Number of rows of an image |
| W | Number of columns of an image |
| L | Number of gray levels |
| L_{min} | The minimum gray level of a range |
| L_{max} | The maximum gray level of a range |
| L_n | The n -th channel of a pixel |
| L_i | The i -th channel of a pixel |
| L_{min}^i | The minimum gray level of the i -th channel of a pixel |
| L_{max}^i | The maximum gray level of the i -th channel of a pixel |
| $N_4(p)$ | Four neighbors of a pixel p |
| $N_D(p)$ | Diagonal neighbors of a pixel p |
| $N_8(p)$ | Eight neighbors of a pixel p |
| AND | AND logic operator |
| OR | OR logic operator |
| XOR | XOR logic operator |
| NOT | NOT logic operator |
| T | Threshold value in image histogram split |
| T_n | Multi-threshold values in image histogram split |
| L^* | Luminance |
| a^* | Green/red axis on $L^*a^*b^*$ color model |
| b^* | Blue/yellow axis on $L^*a^*b^*$ color model |
| u^* | Green/red axis on $L^*u^*v^*$ color model |
| v^* | Blue/yellow axis on $L^*u^*v^*$ color model |
| θ | Hue angle on HSI color model |
| max | Max operator |
| min | Min operator |
| $argmax$ | Arguments of maxima operator |
| P | A point (pixel) in an image |
| P_Y | Y component of a pixel P in $YCbCr$ model |

| | |
|---------------------|--|
| Y_{Cb} | Subspace of Cb points in function of Y |
| Y_{Cr} | Subspace of Cr points in function of Y |
| T_{YC_b} | Trapezoid of the Y_{Cb} subspace |
| T_{YC_r} | Trapezoid of the Y_{Cr} subspace |
| $A_{T_{YC_b}}$ | Area of Y_{Cb} trapezoid |
| $A_{T_{YC_r}}$ | Area of Y_{Cr} trapezoid |
| Y_{min} | Minimum value of Y luminance distribution |
| Y_{max} | Maximum value of Y luminance distribution |
| Y_0 | Top-left coordinate of T_{YC_r} trapezoid |
| Y_1 | Top-right coordinate of T_{YC_r} trapezoid |
| Y_2 | Top-left coordinate of T_{YC_b} trapezoid |
| Y_3 | Top-right coordinate of T_{YC_b} trapezoid |
| Y_{min} | Minimum value of Y component |
| Y_{max} | Maximum value of Y component |
| Cr_{min} | Minimum Cr value used to compute T_{YC_r} trapezoid |
| Cr_{max} | Maximum Cr value used to compute T_{YC_r} trapezoid |
| Cb_{min} | Minimum Cb value used to compute T_{YC_b} trapezoid |
| Cb_{max} | Maximum Cb value used to compute T_{YC_b} trapezoid |
| h_{Cr} | Height of T_{YC_r} trapezoid |
| h_{Cb} | Height of T_{YC_b} trapezoid |
| $H_{Cr}(P_Y)$ | Height of other T_{YC_r} coordinates |
| $H_{Cb}(P_Y)$ | Height of other T_{YC_b} coordinates |
| $\Delta_{Cr}(P_Y)$ | Distance between $(P_Y, H_{Cr}(P_Y))$ coordinate and the base of T_{YC_r} |
| $\Delta_{Cb}(P_Y)$ | Distance between $(P_Y, H_{Cb}(P_Y))$ coordinate and the base of T_{YC_b} |
| $\Delta'_{Cr}(P_Y)$ | Normalized distance with respect to the difference in size of the T_{YC_r} trapezoid |
| $\Delta'_{Cb}(P_Y)$ | Normalized distance with respect to the difference in size of the T_{YC_b} trapezoid |
| α | Rate between $\Delta'_{Cr}(P_Y)$ and $\Delta'_{Cb}(P_Y)$ normalized distances |
| sf | Rate between the longer and shorter upper side of T_{YC_r} and T_{YC_b} |
| P_{Cr} | Cr component of a pixel P |
| P_{Cb} | Cb component of a pixel P |
| P_{Cr_s} | Estimated value of P_{Cr} point |
| P_{Cb_s} | Estimated value of P_{Cb} point |
| dP_{Cr_s} | Estimated distance value of P_{Cr} point |
| dP_{Cb_s} | Estimated distance value of P_{Cb} point |
| dP_{Cr} | Distance between P_{Cr} point and Cr_{min} |
| dP_{Cb} | Distance between P_{Cb} point and Cb_{max} |
| I_P | Minimum difference between P_{Cr} and P_{Cb} with respect to P_{Cb_s} |

| | |
|-----------|--|
| J_P | Maximum distance between the points (P_Y, P_{Cb}) and (P_Y, P_{Cb_s}) |
| I'_P | Minimum difference between P_{Cr} and P_{Cb} with respect to P_{Cr_s} |
| J'_P | Maximum distance between the points (P_Y, P_{Cr}) and (P_Y, P_{Cr_s}) |
| P_{min} | Minimum percentile of the histogram of Y luminance component used to compute trapezoid coordinates |
| P_{max} | Maximum percentile of the histogram of Y luminance component used to compute trapezoid coordinates |

List of Figures

| | | |
|------|---|----|
| 1.1 | The layers of human skin | 1 |
| 1.2 | Map of skin color reflectance | 2 |
| 3.1 | Representation of the raster order of an image | 14 |
| 3.2 | The <i>4-neighbors</i> representation of a pixel p | 15 |
| 3.3 | The <i>8-neighbors</i> representation of a pixel p | 15 |
| 3.4 | Representation of a <i>4-neighbors</i> window mask going beyond image borders . | 16 |
| 3.5 | Grayscale image with its respective histogram | 17 |
| 3.6 | Gray level histograms that can be partitioned by (a) a single threshold, and (b) multiple thresholds | 18 |
| 3.7 | Munsell color model. | 20 |
| 3.8 | CIE 1931 chromaticity diagram | 21 |
| 3.9 | Unit cube representing the colors of the RGB model | 22 |
| 3.10 | CMY subtractive color model | 23 |
| 3.11 | CbCr plane of the YCbCr color model | 25 |
| 3.12 | Graphical representation of the HSI model | 26 |
| 4.1 | 3-dimensional view of the YCbCr channels of some image patches of the SFA dataset | 30 |
| 4.2 | Histogram of Cb and Cr channels of some image patches of the SFA dataset | 30 |
| 4.3 | Skin pixels distribution in the YCr and YCb subspaces of a sample image . | 31 |
| 4.4 | Graphical representation of the trapezoids as well as their parameters . . . | 32 |
| 4.5 | Computation of Cr_{max} based on Cr values histogram of a 724 x 526 image . | 33 |
| 4.6 | Neighbors evaluation with respect to a pixel P | 35 |
| 4.7 | Flowchart of our proposed neighbors method | 36 |
| 4.8 | Image samples with the diagonal effect after the neighbors method segmentation | 38 |
| 5.1 | Structure of the windows that form the SFA patch samples | 40 |
| 5.2 | Examples of SFA face image database | 41 |
| 5.3 | Examples of Pratheepan skin dataset | 42 |
| 5.4 | Examples of HGR skin dataset | 42 |
| 5.5 | Examples of Compaq skin/non-skin dataset | 43 |

| | | |
|------|--|----|
| 5.6 | Image samples with the results of each method in SFA dataset | 46 |
| 5.7 | Image samples with the results of each method in Pratheepan dataset | 47 |
| 5.8 | Image samples with the results of each method in HGR dataset | 47 |
| 5.9 | Image samples with the results of each method in Compaq dataset | 48 |
| 5.10 | The output template table with result measures of the grid search algorithm | 52 |
| 5.11 | Scatter plot with the quality measures for grid search parameters in Compaq dataset. | 55 |
| 5.11 | Scatter plot with the quality measures for grid search parameters in Compaq dataset (cont.). | 56 |
| 5.12 | Scatter plot with the quality measures for grid search parameters in Pratheepan dataset. | 57 |
| 5.12 | Scatter plot with the quality measures for grid search parameters in Pratheepan dataset (cont.). | 58 |
| 5.13 | Scatter plot with the quality measures for grid search parameters in HGR dataset. | 59 |
| 5.13 | Scatter plot with the quality measures for grid search parameters in HGR dataset (cont.). | 60 |
| 5.14 | Scatter plot with the quality measures for grid search parameters in SFA dataset. | 61 |
| 5.14 | Scatter plot with the quality measures for grid search parameters in SFA dataset (cont.). | 62 |

List of Tables

| | | |
|-----|--|----|
| 5.1 | Confusion matrix table used during experiments | 43 |
| 5.2 | Quantitative result measures of the proposed enhancements and original method | 44 |
| 5.3 | Specificity of the proposed enhancements and original method for non-skin images of Compaq dataset | 45 |
| 5.4 | Quantitative result measures of the adaptation to fix neighborhood extended method | 49 |
| 5.5 | Quantitative result measures of the proposed grid search parameters tuning . | 53 |
| A.1 | Trapezoids parameters tuning results for Compaq dataset using combined rules | 65 |
| A.2 | Trapezoids parameters tuning results for Pratheepan dataset using combined rules | 69 |
| A.3 | Trapezoids parameters tuning results for HGR dataset using combined rules | 72 |
| A.4 | Trapezoids parameters tuning results for SFA dataset using combined rules . | 76 |

Chapter 1

Introduction

The study and understanding of human skin color date from many years ago. Being some of the contributors of this field, [Edwards and Duntley \(1939\)](#) were one of those who tried to precisely analyze the color formation of this particular material. According to them ([Edwards and Duntley, 1939](#)), skin is made of a stack of different layers (see Fig. 1.1), each of which reflects a portion of impinging light, after absorbing a certain amount of it by the pigments which lie in the layer. The light which is neither reflected nor absorbed, however, is transmitted through each successive layer to the underlying one, where absorption, reflection, and transmission again take place. The absorption bands of the pigment in each layer are thus "imprinted" on both the reflected and transmitted light from each layer. The total reflected light, consequently, has the absorption characteristics of the pigments in all the layers. This phenomenon gives rise to what we call skin color.

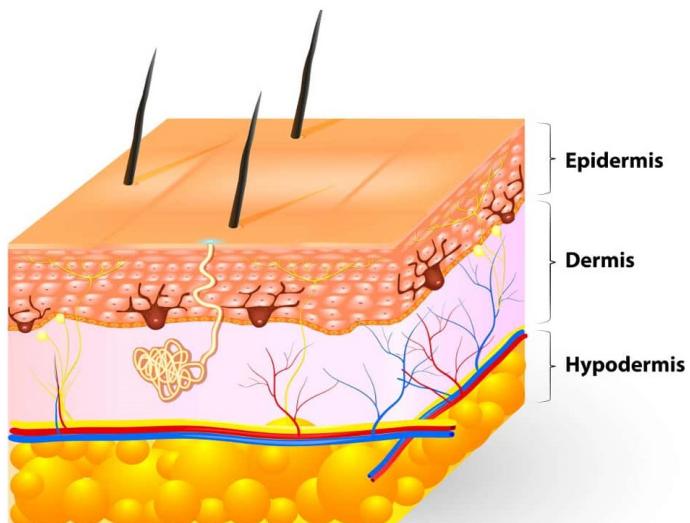


Figure 1.1: The layers of human skin. Source: [Wajnberg \(2018\)](#).

[Edwards and Duntley \(1939\)](#) observed yet that other pigments, out of melanin and hemoglobin, also play a role in the origin of skin color, along with an additional optical effect, designated as scattering. The pigments are melanoid – derivative of melanin –, oxyhemoglobin, and carotene. Furthermore, they stated that variations in the amount of melanin in the epidermis are responsible for the difference in human skin coloration.

Later, [Anderson and Parrish \(1981\)](#) provided an integrated review of the transfer of optical radiation into human skin. They ([Anderson and Parrish, 1981](#)) noticed that the absorption of Ultra Violet (UV) radiation as well as optical scattering, to a lesser degree, build an optical barrier in the epidermis.

In fact, there is evidence that human skin pigmentation is an adaption for the regulation of penetration of UV radiation into the epidermis. Skin pigmentation of populations has been changed genetically as they moved to parts of the world with different UV radiation levels, a necessary fine-tuning that made it possible for them to tan easily (Jablonski and Chaplin, 2000). Tanning is the ability to develop temporary melanin pigmentation in the skin in response to UV radiation and has evolved numerous times in people living under highly seasonal patterns of sunshine (Jablonski and Chaplin, 2010).

Therefore, a natural evolution of the skin coloration has been occurred to accommodate the physiological needs of humans as they have dispersed to regions of widely varying annual average UV radiation (Jablonski and Chaplin, 2010). On the basis of this observation, Chaplin (2004) built a model for the correlation between the skin reflectance and the seasonal UV radiation levels along with other environmental variables. Based on this model, they (Chaplin, 2004) could be able to determine the contribution of each variable to skin reflectance. Chaplin (2004) combined the data of environmental variables with the UV radiation data recorded by satellite and data on human skin reflectance in a Geographic Information System (GIS). A predicted map of skin color reflectance was produced as a result of the visual and statistical analysis of this system (see Fig. 1.2).

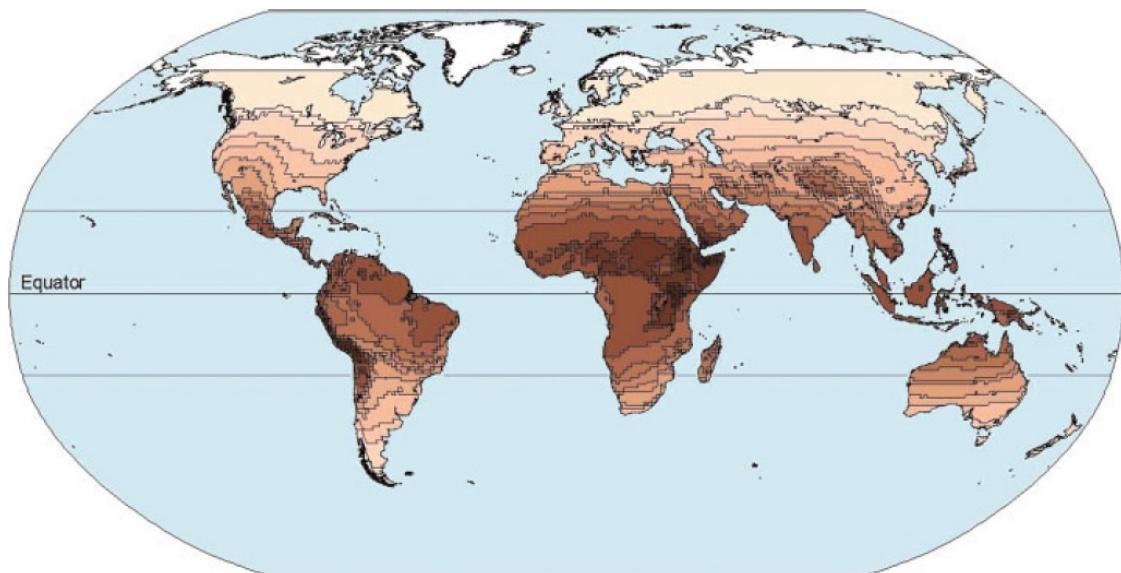


Figure 1.2: Map of skin color reflectance. The map is generalized to reduce the number of polygons due the variety of the human skin colors. Source: Chaplin (2004).

We can clearly see darker shades of skin near the Equator and tropics due to exposition to high UV radiation. This is a very singular feature of human skin, once it acts as a sun shield to protect the body from solar UV radiation (Jablonski, 2004). The great number of shades of skin shown on the map gives us an idea of how complex can be a system to automatically detect human skin in images, based on color only.

Skin detection can be defined as the process of identifying skin-colored pixels in an image. It plays an important role in a wide range of image processing and computer vision applications such as face detection, pornographic image filtering, gesture analysis, face tracking, video surveillance systems, medical image analysis, and other human-related image processing applications (Kakumanu *et al.*, 2007).

The problem is complex because of the numerous similar materials with human skin tone and texture, and also because of illumination conditions, ethnicity, the large number of shades, sensor capturing singularities, geometric variations, etc. Because it is a primary

task in image processing, additional requirements such as real-time processing, robustness, and accuracy are also desirable.

It is worth mentioning that image processing is one of the most important tasks in a computer vision system. Its goal is to create a suitable description – typically based on shapes, textures, gray levels or color – with enough information to differentiate the objects in the scene. With this description, useful interpretation can be extracted from the image by means of an automatic computer system that facilitates human perception (Gonzalez and Woods, 2002).

There is no general agreement among authors regarding where image processing stops and computer vision starts. The first, as the title says, processes the image by applying some transformations on it which will produce a more enhanced and readable image. In addition, the input and output of the process are always images. On the other hand, computer vision requires the control of acquisition and decisions about the visual input process and, requires a visual agent which performs this control (Gonzalez and Woods, 2002). In general, computer vision systems benefit from image processing techniques as pre-processing steps to build better applications. Thus, we can see that they definitely are not different fields, but there is an overlapping between them.

This work is intended to explore new methods on human skin detection. We will use techniques from both image processing and computer vision fields. Color space transformation from image processing, for example, as well as human skin segmentation and understanding as part of computer vision. This is a tentative to imitate the human visual system and its capability to recognize others from the same species – of course, humans use other characteristics to identify other humans like shape, height, gender, and others, but the skin is also part of this recognition system.

One of the powerful features used in this task is definitely skin color, which is a strong attribute and it is used in most algorithms for skin detection. It is normally used along with other features such as shape, texture, and geometry, or even as a preliminary step to classify regions of interest in an image.

The human skin color pixels have a restricted range of hues and are not deeply saturated since the appearance of skin is formed by a combination of hemoglobin (red) and melanin (brown, yellow), which leads the human skin color to be clustered within a small area in the color space (Fleck *et al.*, 1996).

Color has the ability of functioning as a descriptor that often simplifies the identification and extraction of an object in a scene. There are thousands of tonalities and intensities compared to only a few dozen levels of gray, which puts the color as a strong candidate feature in computer vision and image processing applications (Gonzalez and Woods, 2002).

In general, the colors are represented by their brightness, hue, and saturation, which are usually the features used to distinguish one color from another. The brightness gives the notion of chromatic intensity. Hue represents the dominant color perceived by an observer. Saturation refers to the relative purity or amount of white light applied to the hue. Combined, hue and saturation are known as chromaticity and, therefore, a color must be characterized by its brightness and chromaticity (Gonzalez and Woods, 2002).

Colors can be specified by mathematical models in tuples of numbers in a coordinate system and a subspace within that system where each color is represented by a single point. Such models are known as the color models (Gonzalez and Woods, 2002).

The choice of a color space is also a key point of a feature-based method when using skin color as a detection cue. Due to sensitivity to illumination in the scene, the input image is, in general, first transformed into a color space whose luminance and chrominance components can be separated to mitigate the problem (Vezhnevets *et al.*, 2003).

For the case of skin detection methods, there are, basically, three approaches: rule-based, machine learning based, and hybrid. They differ in terms of classification accuracy and computational efficiency. Machine learning and hybrid methods require a training set, from which the partition configuration is learned. In general, such approaches outperform the rule-based methods but require a large and representative training dataset as well as it takes a long training time and, regularly, a costly classification time as well, which can be a deal breaker for real-time applications (Brancati *et al.*, 2017; Kakumanu *et al.*, 2007).

In this work, we propose an improvement of a novel method for rule-based skin detection that works in the YCbCr color space (Brancati *et al.*, 2017). Our motivation is based on the hypothesis that the original rule can be complemented by another rule that is a reversal interpretation of the one originally proposed. Besides that, we also take into consideration that a skin pixel does not appear isolated, so we propose another variation based on neighborhood operations. The set of rules evaluate the combinations of chrominance Cb, Cr values to identify the skin pixels depending on the shape and size of dynamically generated skin color clusters (Brancati *et al.*, 2017). The method is very efficient in terms of computational effort as well as robust in very complex images (i.e. images with varied backgrounds, numerous colors, textures, and shapes).

1.1 Motivation

The subject of the research has been something that was attractive for us from the very beginning of the program. First, with a project on race classification in partnership with the industry. Then, with the intensification in the search for related works, with the problem of skin detection.

The latter led us to the brilliant work of Brancati *et al.* (2017): a new rule-based skin detection method that works in the YCbCr color space. Here, more specifically, our motivation was to propose improvements based on the hypotheses that: (1) the original rule can be complemented and, (2) human skin pixels do not appear isolated, i.e. neighborhood operations are taken in consideration.

1.2 Objectives

Skin detection is a very complex problem due to the numerous similar materials with human skin tone and texture, and also because of illumination conditions, ethnicity, sensor capturing singularities, geometric variations, etc. Because it is a primary task in image processing, additional requirements such as real-time processing, robustness, and accuracy are also desirable.

Although many advances have been observed in the literature, from what we have seen until the development of this research, it is possible to say that it is not yet a completely solved problem. Basically, there are three approaches for skin detection: rule-based, machine learning based, and hybrid. They differ in terms of classification accuracy and computational efficiency. In general, rule-based methods do not require a training step and they can be very competitive in terms of computational cost.

Therefore, the main objective of this research is to create or improve new methods for skin detection, according to the rule-based approach. More specifically, our main objective is to achieve improvements in the method proposed by Brancati *et al.* (2017) in order to reduce the false positive rate.

1.3 Contributions

In this work, we have done a comprehensive and detailed study of various methods of skin detection within those based on rules. On the basis of Brancati *et al.* (2017), seen by us as the state of the art in this field, we created variations that brought significant improvements. In addition, we analyzed the methods in turn in order to provide the researchers, practitioners, enthusiasts, and other readers with a detailed understanding of the nuances involved in those methods, such as parameter selection and optimization, through a series of quantitative experiments, as well as qualitative analysis based on our observations.

Thus, we can enumerate some of the contributions that came as a result of this research project:

1. Three variations of the human skin segmentation method proposed by Brancati *et al.* (2017) on the basis of the inversely proportional behavior of the chrominance components (Cb and Cr) of the YCbCr color model. Furthermore, extensive quantitative and qualitative experiments performed in a wide range of image datasets well-known in this field;
2. Adaptation by means of heuristics to fix the neighborhood method (8-neighbors window) presented in Section 4.4;
3. A grid search implementation to try different combinations of trapezoids parameters in order to optimize them (if possible) and understand those who have been used so far.

Part of our contributions was published earlier in 2018 in the *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - VISAPP* (Faria and Hirata Jr., 2018).

1.4 Organization

In this chapter, we have presented the background of this work as well as the motivation, main contributions, and objectives behind it. Chapter 2 presents other relevant research works that also addresses the problem of skin detection in several distinct approaches. In Chapter 3, we provide an overview of the theoretical concepts that apply to this research. Next, in Chapter 4, we present a state of the art skin detection method recently developed by Brancati *et al.* (2017). We review the method and extend it adding more rules to enforce the constraints and seeking for a better accuracy in terms of false positive rate without hurting the performance of the original method. Then, in Chapter 5, we present the evaluation of the proposed extensions along with the original method in four widely known datasets: SFA, Pratheepan, HGR, and Compaq. In addition, a brief definition of the evaluation metrics used is shown for the sake of clarity. Finally, Chapter 6 winds up this thesis by discussing our observations along the research, focused on the experimental results, and directs the readers towards future works.

Chapter 2

Related Work

There is a large number of works of skin detection based on color information and there are a couple of them comparing different techniques and classifiers, mainly from the point of view of performance, color models, skin color modeling and different datasets (Kakumanu *et al.*, 2007; Mahmoodi and Sayedi, 2016; Vezhnevets *et al.*, 2003).

On different techniques, statistical models are those which estimate the probability that an observed pixel is associated with skin, based on a estimation dataset. One approach is the single histogram based Look-UP Table (LUT) that is capable to obtain the distribution of skin pixels in a particular color space, by using a set of training pixels (Mahmoodi and Sayedi, 2016). Considering the Red, Green, Blue (RGB) color model for instance, a histogram with 256 bins per channel - 256³ in total - can be constructed for further counting the probability (see Eq. 2) of each possible RGB value (Jones and Rehg, 2002).

$$P(rgb) = \frac{\#counts\ of\ rgb}{total\ counts}$$

In Jones and Rehg (2002), the authors applied this technique to figure out the decision boundary of skin pixels distribution by using a 3-dimensional histogram model constructed from approximately 2 billion pixels. Those pixels were collected from 18,696 images over the Internet to perform skin detection. First, visualization techniques were used to examine the shape of these distributions. Then, by examining the 3D histogram from several angles, Jones and Rehg (2002) realized that its overall shape could be inferred.

So, two different histograms for skin and non-skin in the RGB color model were calculated. Using those histograms along with training data, a surprisingly accurate pixel-wise classifier was derived. The best performance at a success rate of 88% was reached for histograms of size 32 (Jones and Rehg, 2002).

On the basis of the output of the skin detector, Jones and Rehg (2002) also trained a classifier to determine whether a naked person is present or not in the scene. The features used from the output to create the feature vector were (Jones and Rehg, 2002):

- Percentage of pixels detected as skin
- Average probability of the skin pixels
- Size in pixels of the largest connected component of skin
- Number of connected components of skin
- Percent of colors with no entries in the skin and nonskin histograms

- Height of the image
- Width of the image

In the last step, 10,679 images were manually classified in 5,453 naked and 5,226 non-naked image sets to train a neural network classifier. The neural network outputs a number between 0 and 1, with 1 indicating a naked person in the image. The detection rate achieved was of 88%, with a false alarm rate of 11.3% ([Jones and Rehg, 2002](#)).

Skin detection is sometimes used as a primary task for other applications. [Hsu et al. \(2002\)](#) have embedded skin segmentation with the purpose to build a face detector application. The algorithm first performs a lighting compensation on the R, G, and B components of the pixels of the image. Then, these color corrected components are transformed into YCbCr space. [Hsu et al. \(2002\)](#) observed that a skin tone cluster is formed in the YCb and YCr subspaces. Based on the premise that skin tone color is independent of the luminance component, they ([Hsu et al., 2002](#)) non-linearly transformed the YCbCr. Then, in this transformed YCbCr space, the skin pixels are detected using an elliptical model.

The experiments have shown good skin detection results, around 96% and 80% detection rate respectively on HH1 MPEG7 video and on Champion datasets, which contain images with frontal, near-frontal, half-profile and profile faces under varying illumination conditions and backgrounds. On the other hand, the results also showed a high false positive rate, that is reduced further with facial feature detection procedure based on the spatial arrangement of the detected skin patches ([Hsu et al., 2002](#)).

Another typical method is to define explicitly, through a number of rules, the boundaries that delimit the grouping of skin pixels in some color space ([Vezhnevets et al., 2003](#)). This approach has been adopted by [Kovac et al. \(2003\)](#) in the RGB color model, resulting in a true positive rate of 90.66%. Basically, to find out the skin cluster in the RGB color model, the skin color is determined with the following rules ([Kovac et al., 2003](#)):

Skin color at uniform daylight illumination

$$R > 95, G > 40, B > 20$$

$$\max(R, G, B) - \min(R, G, B) > 15$$

$$|R - G| > 15$$

$$R > G$$

$$R > B$$

Skin color at light daylight illumination

$$R > 220, G > 210, B > 170$$

$$|R - G| \leq 15$$

$$R > B$$

$$G > B$$

They ([Kovac et al., 2003](#)) also performed experiments in comparison with [Hsu et al. \(2002\)](#), where only the chromaticity channels Cb and Cr from the YCbCr color space are used. The results showed that the performance of the classifier is inferior in relation to the approach using all the three channels. [Hsu et al. \(2002\)](#) method indeed diminished the influence of noise in dark images, but in images that are captured under standard daylight

illumination, they label too many pixels as skin, decreasing the performance of the face detector by increasing the number of false positive pixels (Kovac *et al.*, 2003).

Chai and Ngan (1999) proposed a similar method in YCbCr color space, where a skin color map was designed using a histogram approach based on a given set of training images. Chai and Ngan (1999) observed that the Cb and Cr distributions of skin color fall in the ranges [77, 127] and [133, 173], respectively, regardless the skin color variation in different races.

However, after an exhaustive image histogram analysis, Basilio *et al.* (2011) found that the thresholds given by Chai and Ngan (1999) was robust only for images with Caucasian people. Once their (Basilio *et al.*, 2011) purpose was to find human skin from different people races, from any place of the world, a new threshold for each chromaticity (Cb, Cr) channel has been set up, regardless of skin color:

$$\begin{aligned} 80 \leq Cb \leq 120 \\ 133 \leq Cr \leq 173 \end{aligned}$$

The key advantage of this method is the simplicity of skin detection rules that lead to the construction of a very fast classifier. On the other hand, achieving high recognition rates with this method is difficult because it is necessary to find a good color space and empirically appropriate decision rules (Vezhnevets *et al.*, 2003).

Differently from Kovac *et al.* (2003), the authors of Yogarajah *et al.* (2011) developed a technique where the thresholds defined in the rules are dynamically adapted. The method consists of detecting the region of the eye and extracting an elliptical region to delimit the corresponding face. A Sobel filter is applied to detect the edges of the resulting region which is subjected to a dilation in order to get non-smooth regions, (i.e. eyes and mouth). The resulting image is subtracted from the elliptical region. As a result, there is a more uniform skin region where the thresholds are calculated.

Every single pixel in the colored image is classified as skin or non-skin, based on the calculated dynamic threshold values. When the algorithm detects multiple possible face regions in the image, a dynamic threshold is constructed for each of them and, subsequently, submitted to perform skin segmentation on the whole image. Finally, a logical OR operation is applied in all of the segmented regions obtained as of each dynamic threshold (Yogarajah *et al.*, 2011).

The technique was used as part of a preprocessing step in Tan *et al.* (2012) in a strategy combining a 2-dimensional density histogram and a Gaussian model for skin color detection. First, human eyes are located and, then, an elliptical mask model is used to generate the elliptical face region in the image. Due to computational simplicity, a Sobel filter is employed to detect non-smooth (i.e., eyes, eyebrows, mouth, etc.) regions. Then, the detected edge pixels are further submitted to a dilation operation to get the optimal non-smooth regions. Finally, a new image, that only consists of face regions, is obtained.

It is worth mentioning that the dynamic threshold with smoothed 2-D histogram is based on the assumption that the face and body of a person always share the same colors (Tan *et al.*, 2012).

Thereafter, a 2-D histogram with smoothed densities and a Gaussian model are used as features to represent the skin and non-skin distributions, respectively. They (Tan *et al.*, 2012) also applied a fusion technique that uses the product rule on the two features to obtain better skin detection results.

Experiments were carried out using three public databases: Pratheepan, ETHZ PASCAL,

and Stottinger. A comparison between different color spaces as well as proposed fusion and non-fusion approaches were also established. Quantitative results are only available for Stottinger dataset with $F - measure = 0.6490$, $Precision = 0.6403$ and $Recall = 0.6580$ measures (Tan *et al.*, 2012).

Naji *et al.* (2012) built a rule-based classifier in the HSV color space for 4 different skin ethnic groups in parallel (standard skin, shadow/blackish skin, light skin and red concentrated skin and lips). Basically, the classifier applies a rule-based region growth algorithm, after primitive segmentation, in which the output of the first layer is used as a seed, and then, the final mask in subsequent layers is constructed iteratively by neighboring skin pixels. The problem of shadow regions has been also addressed by lightening them with a skin color correction approach. A number of 125 images were collected from two different datasets for experiments. The rate of true positive pixels reported was of 96.5% with a very low false positive rate of 0.76%.

The model proposed by Kawulok *et al.* (2013) is based on global and local image information from where a skin probability map is built and, next, used to generate the initial seed for spatial analysis of skin pixels. The spatial analysis result is improved once these seeds, extracted from a local model, are highly adapted to the image.

In their (Kawulok *et al.*, 2013) proposal, first, human faces are captured in the input image through a multi-level ellipse detector, from where a local skin color model is learned. The local model is used to perform skin detection in the input image. Next, the local skin probability map is used to apply a high-probability threshold and obtain the seeds. Finally, based on the seeds gathered on the previous step, the spatial analysis is carried out in the global probability map, created on the basis of the global skin color model.

In general, color can be considered one of the most decisive properties that affect the performance of skin detection algorithms, even though some of them do not use the color as a feature directly (Mahmoodi and Sayedi, 2016). Therefore, the choice of a color space is directly related to the outcome produced by a particular approach (Mahmoodi and Sayedi, 2016). However, Albiol *et al.* (2001) proved that the optimum performance of the skin classifiers is independent of the color space chosen.

RGB is the most commonly used color space for storing and representing digital images since the cameras are enabled to provide the images in such model as well as displays to show them. However, RGB suffers a high influence of the environment illumination. In order to reduce this influence, the RGB channels can be normalized (Kakumanu *et al.*, 2007).

This particular characteristic led Bergasa *et al.* (2000) to build an adaptive and unsupervised Gaussian model to segment skin in the normalized RGB color space, using only the r and g channels (the normalized R and G channels), after the evaluation of other color spaces for the same application.

In Jayaram *et al.* (2004), a comparative study using a Gaussian and a histogram approaches in a dataset of 805 color images in 9 different color spaces has been performed. The results revealed that the absence of the luminance component, that means, using only two channels of the color space, significantly impacts the performance. The best results were obtained in the SCT, HSI, and CIELab color spaces with histogram approach.

In Chaves-González *et al.* (2010), the authors compared, under the same circumstances, the performance of 10 color spaces based on the k -means clustering algorithm on 15 images of the Aleix and Robert (AR) face image database (Martínez and Benavente, 1998). The ground truth images have been carefully generated by the authors. This means that parts of the photo which do not include skin color, such as hair, beard, lips, eyes, and background, were completely removed.

Experiments have been executed over the test set images provided in AR database. For

each color space, Chaves-González *et al.* (2010) have done several tests with each channel lonely, using two channels – with all possible combinations –, and with the three channels.

The tests have been carried out comparing, in a detailed quantitative manner, each color space with an accurate and objective criterion. In other words, the output image produced by the k -means was compared, in a pixel-wise fashion, to the ground truth images to get the quantitative measures. According to the results obtained, using two channels combined did not produce a good outcome. On the other hand, the results with each channel isolated were surprisingly good in comparison to the three channels fused. Lastly, among all the color spaces used for the skin detection experiments, the most appropriate were, in this order, HSV, YCgCr, and YDbDr (Chaves-González *et al.*, 2010).

A similar study on color spaces and different skin color modeling have been provided in Khan *et al.* (2012). A total of six color spaces (IHLS, HSI, RGB, normalized RGB, YCbCr and CIELab) and nine algorithms (AdaBoost, Bayesian network, J48, Multilayer Perceptron, Naive Bayesian, Random Forest, RBF network, SVM and Histogram based) were evaluated on 8991 manually pixel-wise annotated images by means of F-measure. Khan *et al.* (2012) concluded that color space transformation, as well as the removal of the luminance channel, degrades the performance. In addition, classification results can be improved with the usage of lighting correction algorithms.

In Kaur and Kranthi (2012), an algorithm similar to that proposed by Kovac *et al.* (2003) have been implemented, where the boundaries that delimit the grouping of skin pixels are defined by explicit rules. After segmenting the image with the explicit rules, the algorithm also performs morphological and filtering operations to improve the accuracy of the method. The authors applied the algorithm in the YCbCr and CIELab color spaces, ignoring the Y and L luminance components, respectively. The results were more satisfactory when the algorithm was applied on CIELab. A similar technique was implemented in Shaik *et al.* (2015) and Kumar and Malhotra (2015) in the HSV and YCbCr color spaces, the latter providing the best results in both.

Finally, Brancati *et al.* (2017) proposed a novel rule-based skin detection method that works in the YCbCr color space. Basically, to classify skin pixels in the image, the algorithm evaluates the combinations of chrominance (Cb, Cr) components by way of correlation rules. These correlation rules are calculated relying upon the shape and size of dynamically generated skin color clusters. Geometrically, the clusters create trapezoidal polygons in the YCb and YCr subspaces whose apparent arrangement reflects an inversely proportional behavior observed in relation to chrominance components. They (Brancati *et al.*, 2017) compared the results, in terms of quantitative performance metrics, with six other well known rule-based methods in the literature, outperforming them in almost all of the metrics.

Chapter 3

Theoretical Background

In this chapter, the theoretical concepts that apply to this research are stated. First, we define and explain the concept of a digital image in Section 3.1 as well as basic relationship between pixels in Section 3.2. In Section 3.3 we will see how image histograms are fundamental for the methods described in Chapter 4 of the proposed solution. Also, the background of image segmentation, mainly for the thresholding technique is shown in Section 3.4. Thereafter, a brief introduction to color models is provided in Section 3.5 in order to give an overview of the main characteristics of some of the most used in the computer vision and image processing area, on which this research is based.

3.1 Digital image

By definition, an image is a two-dimensional function where its amplitude, at any pair of coordinates, is called the *intensity* or *gray level* of the image at that point. The image is said digital when this function is converted to a discrete form, $f(x, y)$ let's say. This is made by a process called *digitalization*, which consists of two steps: *sampling* and *quantization* (Gonzalez and Woods, 2002).

Each element of the discrete function $f(x, y)$ is called *pixel* (picture element), where $0 \leq x \leq H - 1$ and $0 \leq y \leq W - 1$. This means that the image can be represented in a matrix form (see Eq. 3.1), where H is the number of rows and W the number of columns of the matrix. Therefore, H and W define the size or resolution of the image (Pedrini and Schwartz, 2008).

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \cdots & f(0, W - 1) \\ f(1, 0) & f(1, 1) & \cdots & f(1, W - 1) \\ \vdots & \vdots & \ddots & \vdots \\ f(H - 1, 0) & f(H - 1, 1) & \cdots & f(H - 1, W - 1) \end{bmatrix} \quad (3.1)$$

Usually, pixels are stored, and therefore read, in this matrix in an order known as *raster* (see Fig. 3.1). This information is important so that capture and display devices may be able to establish a common interface, and make necessary transformations in the coordinates, when needed.

In a monochromatic digital image, the value of a pixel is a scalar in the range $[L_{min}, L_{max}]$, where L_{min} and L_{max} are, in general, positive integers¹, respectively, the minimum and maximum gray level values (Pedrini and Schwartz, 2008).

¹In some rare occasions, L_{min} can be a negative integer.

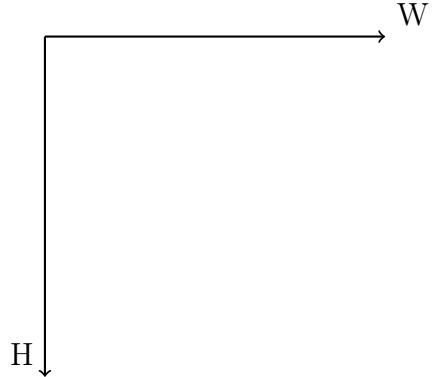


Figure 3.1: Representation of the raster order of an image. The origin coordinates $(0, 0)$ starts in the top left corner, where both axes rise. Source: proposed by the author.

In a multispectral image, each pixel has a vector value such that $f(x, y) = (L_1, L_2, \dots, L_n)$ where $L_{min}^i \leq L_i \leq L_{max}^i$ and $i = 1, 2, 3, \dots, n$. In general, L_i can represent different measures for each of (x, y) coordinate as well as different intervals (Pedrini and Schwartz, 2008).

A colored image is a multispectral image, where the color in each (x, y) point is given by three variables: brightness, hue and saturation (Pedrini and Schwartz, 2008). The brightness gives the notion of chromatic intensity. Hue represents the dominant color perceived by an observer. Saturation refers to the relative purity or amount of white light applied to the hue. Combined, hue and saturation are known as chromaticity and, therefore, a color must be characterized by its brightness and chromaticity (Gonzalez and Woods, 2002).

3.2 Basic relationship between pixels

There is a number of applications in image processing and computer vision that uses information of relationship among pixels to create knowledge. Some of these important relationships will be described in the following sections once we will apply it further in Chapter 4 further. It is worth mentioning that we defined an image as a function $f(x, y)$. In this section, when referring to a particular pixel, we will denote it in lowercase letters such as p .

3.2.1 Neighborhood

A pixel p with coordinates (x, y) can have up to four² horizontal and vertical neighbors whose coordinates are given by:

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)$$

This set of pixels, called the *4-neighbors* of p , is denoted by $N_4(p)$ (Gonzalez and Woods, 2002). See Figure 3.2 for a reference on how this neighborhood looks like.

Some neighbors of p might lie outside of the image boundaries if (x, y) is on the border of the image (Gonzalez and Woods, 2002). Those who will use neighborhood operations in the image might take this in consideration to avoid index out of range in those regions.

The four coordinates of the diagonals of p , if they exist in the domain of the image, are given by:

$$(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1)$$

²Respecting the image domain limits.

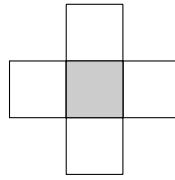


Figure 3.2: The 4-neighbors representation of a pixel p . The pixel p is centered on the grid with gray background. Source: adapted from [Pedrini and Schwartz \(2008\)](#).

This set of pixels are denoted by $N_D(p)$. When combined, the 4-neighbors and $N_D(p)$ will generate the 8-neighbors of p , known as $N_8(p)$ (Gonzalez and Woods, 2002). Formally, we have:

$$N_8(p) = N_4(p) \cup N_D(p)$$

See Figure 3.3 for a reference on how the $N_8(p)$ neighborhood looks like.

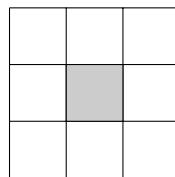


Figure 3.3: The 8-neighbors representation of a pixel p . The pixel p is centered on the grid with gray background. Source: adapted from [Pedrini and Schwartz \(2008\)](#).

Despite these are the most common neighbors used in applications, other different distances from p as well as connectivity can be applied. The idea of neighborhood can also be extended to 3-dimensional images where, instead of pixels, the voxels are the coordinates considered (Pedrini and Schwartz, 2008).

3.2.2 Connectivity

The connectivity between pixels is a very important concept used to establish the boundaries of objects and regions in an image. To figure out if two pixels are connected, it must be determined if they are neighbors and if their gray levels satisfy some similarity criteria, such as gray levels, color or texture equality. For instance, in a binary image, where the pixels values vary in the range $[0, 1]$, two pixels may be 4-neighbors, but they are connected if and only if they have the same value (Gonzalez and Woods, 2002).

3.2.3 Image boundaries

When working with neighborhood operations, the mask being used, independently of its size, can fall beyond the image boundaries. In other words, this means that, for an image f , with size $W \times H$, some parts of the mask operator will be outside of the domain of the image (i.e. out of the range given by $W \times H$ matrix) (Pedrini and Schwartz, 2008). Figure 3.4 shows an example of this phenomenon in a 10×8 image being used as input for an 4-neighbors operator.

There are several ways to work around this problem. One simple mechanism is to simply ignore the pixels on the border where the mask go beyond – this is the approach we used in this work. Despite this will avoid index out of bound errors, border pixels of the image will not be looked at. Another approach is to copy the corresponding pixels from the input image. Once again, the resulting image will have some not processed pixels in the border. Another

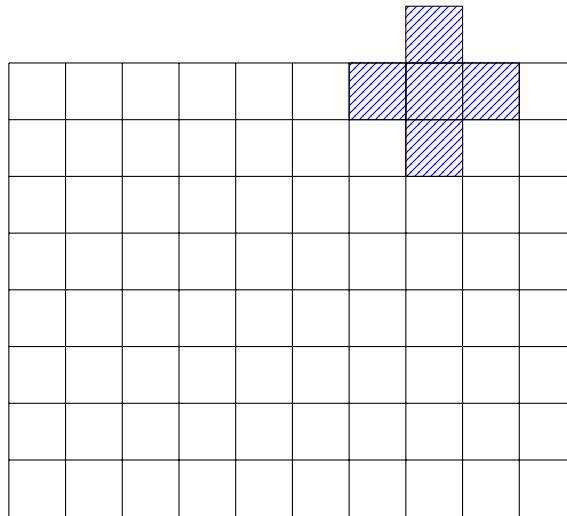


Figure 3.4: Representation of a 4-neighbors window mask going beyond image borders. Source: adapted from [Pedrini and Schwartz \(2008\)](#).

strategy is to apply a different mask for the borders, which may consider the difference in the corners as well, to perform the operation, which can turn the operation more complex and computational costly ([Pedrini and Schwartz, 2008](#)).

3.3 Image histogram

In particular for image processing, when we are referring to an image histogram, we are talking about to count the frequency of the pixels for each intensity level in separate bins, in a multi-valued (intensity levels) sequence ([Fisher et al., 2003](#)). The most common visual representation is a graph or plot, which gives an overall idea about the intensity distribution of an image. The foundation of a histogram can be seen as a set of bins, where each bin is representing a certain intensity value from a given range. A simple algorithm to compute a histogram of an image may examines all pixels in the image and assigns each to a bin depending on the pixel intensity. In the end, each bin will have the number of pixels of its own intensity value ([Gonzalez and Woods, 2002](#)).

For instance, in an 8-bit grayscale image there are 256 different possible intensities, and so, the histogram will graphically display 256 integer numbers showing the counting of occurrences of those grayscale values. It is also possible to compute histograms of color images. In this particular case, the channels of the color space in turn are split individually from where a separate histogram is calculated ([Fisher et al., 2003](#)). We are using color histograms of the YCbCr channels image in this work.

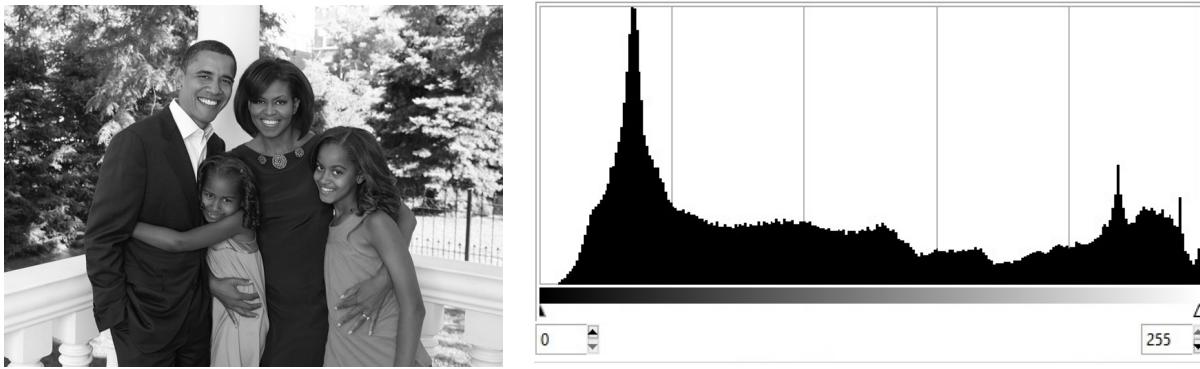


Figure 3.5: Grayscale image with its respective histogram. Left image is a sample from Pratheepan dataset, transformed to grayscale. On the right we have its histogram of pixel intensity values in the range [0, 255]. Source: Tan et al. (2012) and the author.

Several statistical measures can also be obtained from a histogram such that minimum and maximum values, mean, median, standard deviation, and percentiles (Pedrini and Schwartz, 2008). Those measures, specifically percentiles, as well as the histogram itself are fundamental for the methods described in Chapter 4 of the proposed solution.

3.4 Image segmentation

Image segmentation is the task of partitioning the image domain so each part is semantically different from each of its neighbors. These regions which, in general, are the representations of specific objects, are the fundamental parts of an image. This partitioning process is also used by humans, through their visual perception, into surrounding environment to help themselves to recognize them, guide their movements, and for almost every other day by day tasks in their lives (Plataniotis and Venetsanopoulos, 2000).

Segmentation of complex images is one of the most difficult tasks in the field of image processing, where the accuracy determines whether a successful operation or not. It is a heavy process that consists of many steps such as the analysis of color, shape, motion, geometry, and texture of objects in images (Gonzalez and Woods, 2002).

Usually, the first task of any image analysis process is image segmentation. All subsequent tasks, such as feature extraction and object recognition rely heavily on the quality of the segmentation (Plataniotis and Venetsanopoulos, 2000). The algorithms created for image segmentation generally are based on one of two basic properties of intensity values: *discontinuity*, where abrupt changes in intensity, such as points, lines, and edges are detected, and *similarity*, whose approaches are based on regions partitioning, according to a set of predefined criteria (Gonzalez and Woods, 2002).

One of the *similarity* methods it is the thresholding: a popular and simple method used to segment regions of an image, especially in applications where speed is an important factor – that is the case for skin detection, once it is used, in general, for face detection, gesture analysis, face tracking, video surveillance systems, medical image analysis, and other human-related image processing applications. Due to the simplicity of implementation, several authors apply this technique for skin detection task (Basilio et al., 2011; Chai and Ngan, 1999; Kaur and Kranthi, 2012; Kovac et al., 2003; Kumar and Malhotra, 2015; Shaik et al., 2015). Others extend this application by using adaptive thresholding (Tan et al., 2012; Yogarajah et al., 2011).

3.4.1 Thresholding

Histogram thresholding is one of the simplest pixel-based techniques for image segmentation due to its intuitive properties and easy implementation (Gonzalez and Woods, 2002). Roughly speaking, if an image is composed of distinct regions – that means, each region in the histogram is represented by a peak –, adjacent regions could be split into separated groups, once adjacent peaks are likely separated by a valley (Plataniotis and Venetsanopoulos, 2000). Those valleys can be determined by means of one or more threshold values. However, to find out the threshold value that is the lower bound of a valley is definitely not a trivial task (Plataniotis and Venetsanopoulos, 2000).

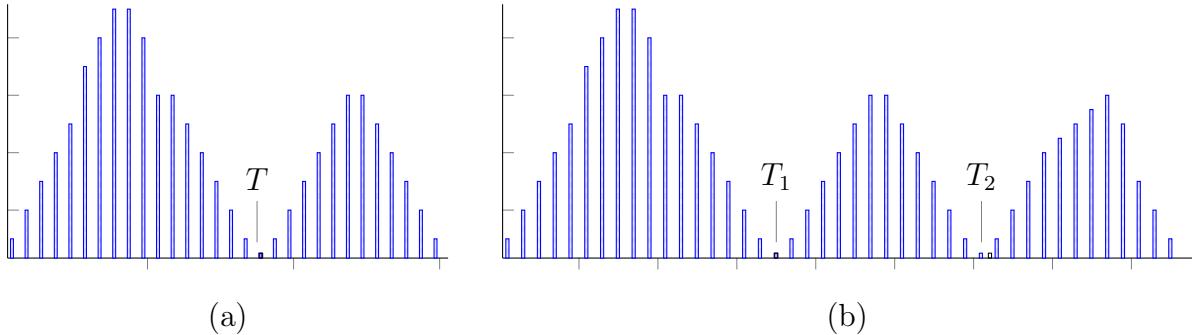


Figure 3.6: Gray level histograms that can be partitioned by (a) a single threshold, and (b) multiple thresholds. Source: adapted from Gonzalez and Woods (2002).

Let Figure 3.6(a) be the gray levels histogram of an image $f(x, y)$. We can clearly see two dominant groups of pixels. One obvious way to separate the objects in this image is to select a threshold T that splits these groups. In other words, for every single point (x, y) , if $f(x, y) \leq T$, then (x, y) belongs to the first object (or group), otherwise it belongs to the second object. In general, one of these objects is seen as the background of the image and the other, the portion of interest in the image (Gonzalez and Woods, 2002).

The resulting image $g(x, y)$ after the thresholding is given by (Gonzalez and Woods, 2002):

$$g(x, y) = \begin{cases} 0, & \text{if } f(x, y) \leq T \\ 1, & \text{otherwise} \end{cases} \quad (3.2)$$

Therefore, pixels labeled 0 corresponds to the object left of T in the histogram, and pixels labeled 1 corresponds to the object right of T in histogram – or background as commonly used in literature. The *single-level thresholding* procedure that produces $g(x, y)$ image is known as *binarization* (Gonzalez and Woods, 2002).

Figure 3.6(b) shows a slightly more general case of thresholding, where three distinct groups of the histogram are split by T_1 and T_2 threshold values. Here, *multi-level thresholding* classifies a point (x, y) as belonging to a specific object class as follows (Gonzalez and Woods, 2002):

$$g(x, y) = \begin{cases} l_1, & \text{if } f(x, y) \leq T_1 \\ l_2, & \text{if } T_1 < f(x, y) \leq T_2 \\ l_3, & \text{if } f(x, y) > T_2 \end{cases} \quad (3.3)$$

where l_1, l_2, l_3 are different gray levels used to represent each interval given by the thresh-

olding values.

When T depends only on $f(x, y)$, that is, only on gray level values, the threshold is called *global*. When another property – e.g. the average gray level of a neighborhood centered on (x, y) – is used to define T , the threshold is called *local*. If, in addition, T depends on spatial coordinates x and y , the threshold is called *dynamic* or *adaptive* (Gonzalez and Woods, 2002).

3.5 Color models

In general, the colors are represented by their brightness, hue, and saturation, which are usually the features used to distinguish one color from another. The brightness gives the notion of chromatic intensity. Hue represents the dominant color perceived by an observer. Saturation refers to the relative purity or amount of white light applied to the hue. Combined, hue and saturation are known as chromaticity and, therefore, a color must be characterized by its brightness and chromaticity (Gonzalez and Woods, 2002).

Colors can be specified by mathematical models in tuples of numbers in a coordinate system and a subspace within that system where each color is represented by a single point. Such models are known as the color models (Gonzalez and Woods, 2002).

These models can be classified as of two types: the additive models in which the primary color intensities are added to produce other colors and subtractive, where colors are generated by subtracting the length of the dominant wave from the white light.

The following sections briefly describe some color models, as well as their variants and main areas of application.

3.5.1 Munsell color model

Pioneer in an attempt to organize the perception of the colors in a color space, Albert Henry Munsell was able to combine the art and science of colors in a single theory (Plataniotis and Venetsanopoulos, 2000).

The principle of equality of perceptual separation between the components of the model is the essential idea of the Munsell color model. These components are hue, value, corresponding to luminance, and chroma, corresponding to saturation (Plataniotis and Venetsanopoulos, 2000).

The model is represented by a cylindrical coordinate system and it can be seen in the Figure 3.7. The hue is arranged in the circular axis consisting of five bases as well as five secondary colors, the saturation in the radial axis and the luminance in the vertical axis in a range varying from 0 to 10.

3.5.2 CIE color model

In 1931, the CIE established the first mathematical model of a color numerical specification, whose objective was to analyze the relationship between the physical aspects of colors in the electromagnetic spectrum and their perception by the human visual system to determine how an ordinary person perceives the color. A review of this specification was published in 1964 (Gonzalez and Woods, 2002).

The experiment that originated the standard consisted of detecting the colors perceived by an observer from a mixture of three primary colors X, Y and Z, called tristimulus values. These coordinates gave rise to the CIE XYZ color space which encompasses all the colors

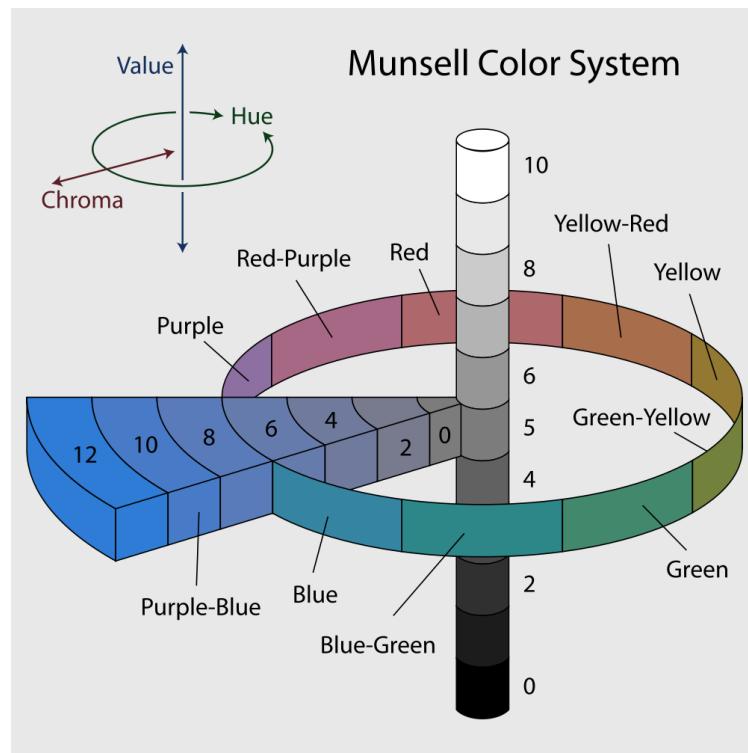


Figure 3.7: Munsell color model represented by a cylindrical coordinate system. The hue is arranged on the circular axis consisting of five bases and five secondary colors, the chroma (saturation) on the radial axis and the luminance (value) on the vertical axis in a range varying from 0 to 10. Source: Rus (2007).

that can be perceived by an ordinary human being. For this reason, it is considered a device independent representation (Plataniotis and Venetsanopoulos, 2000).

The system proposed by the CIE XYZ to describe a color is based on a luminance component Y, and two additional components X and Z, that bring the chromaticity information. This system is formed by imaginary colors that can be expressed as combinations of the normalized measures shown in Equations 3.4, 3.5 and 3.6 (Plataniotis and Venetsanopoulos, 2000).

$$x = \frac{X}{X + Y + Z} \quad (3.4)$$

$$y = \frac{Y}{X + Y + Z} \quad (3.5)$$

$$z = \frac{Z}{X + Y + Z} \quad (3.6)$$

where $x + y + z = 1$.

Combinations of negative values and other problems related to selecting a set of real primaries are eliminated. The chromaticity coordinates x and y allow to represent all colors in a two-dimensional plane, also known as a chromaticity diagram, which can be seen in the Figure 3.8.

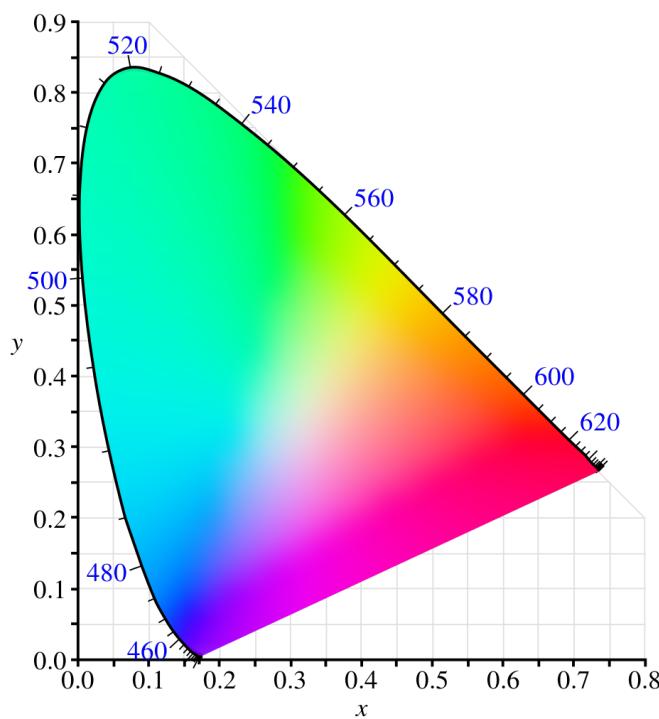


Figure 3.8: CIE 1931 chromaticity diagram. The points representing pure colors in the electromagnetic spectrum are labeled according to their wavelengths and are located along the curve from the right end of the x -axis, corresponding to the red color, to the left end of the same axis, corresponding to the violet color, forming a polygon similar to a horseshoe. The internal points correspond to all possible combinations of visible colors. Source: Ben (2009).

The coordinates $(x = 1/3, y = 1/3)$ correspond to the location of the white light, also known as white point, and serve as a reference in the process of image capture, coding, or reproduction.

CIE also derived and standardized two other color models based on CIE XYZ specification and, likewise, are device independent. Both are perceptually uniform, which means that equal perceptual distances separate all colors in the system (Vezhnevets *et al.*, 2003). As an example, the gray scale of the space should allow for a smooth transition between black and white.

The first one was designed to reduce the problem of perceptual non-uniformity. Some Uniform Chromaticity Scale (UCS) diagrams were proposed based on mathematical equa-

tions to transform the values XYZ or the coordinates x, y into a new set of values (u, v) , which gave rise to the 1960 CIE uv chromaticity diagram (Gevers *et al.*, 2012).

With unsatisfactory results yet, the CIE made a new change by multiplying the v component by a factor of 1.5. In addition, the brightness scale given by the Y component has been replaced by $L^* = [0, 100]$ to better represent the differences in luminosity that are equivalent. This revision originated the CIE 1976 $L^*u^*v^*$ color model, commonly known by the acronym CIELuv (Gevers *et al.*, 2012).

In 1976 the CIE adopted a new color model, based on the L, a, b model, proposed by Richard Hunter in 1948, which best represented the uniform spacing of colors. Named CIE $L^*a^*b^*$ and known by the acronym CIELab, it is a space based on opponent colors³ in which the color stimuli of retina is converted to distinctions between light and dark, red and green, and blue and yellow, represented by the axes L^* , a^* , and b^* , respectively (Gevers *et al.*, 2012).

3.5.3 RGB color model

The RGB model, an acronym for Red, Green, and Blue, is an additive color model in which the three primary colors red, green and blue are added to produce the others (Gonzalez and Woods, 2002).

This system was based on the trichromatic theory of Thomas Young and Hermann Helmholtz in the mid-19th century and can be represented graphically (see Fig. 3.9) through the unit cube defined on the axes R, G, and B (Plataniotis and Venetsanopoulos, 2000).

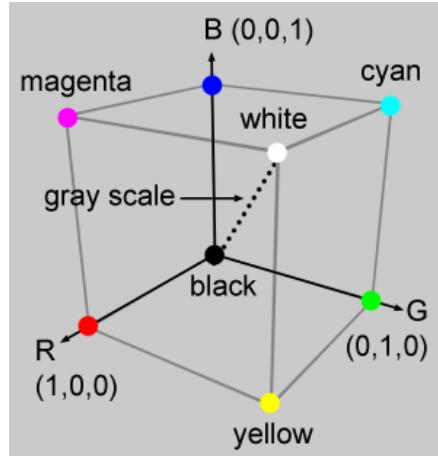


Figure 3.9: Unit cube representing the colors of the RGB model. The origin, given by the vertex $(0, 0, 0)$, represents the black color. The vertex $(1, 1, 1)$, opposite to the origin, represents the white color. The highlighted vertices on the axes represent the primary colors and the others are the complement of each. Each point inside the cube corresponds to a color that can be represented by the triple (r, g, b) , where $r, g, b \in [0, 1]$. The shades of gray are represented along the main diagonal of the cube, with each point along this diagonal being formed by equal contributions of each primary color. Source: adapted from Gonzalez and Woods (2002).

It is noteworthy that there are two ways to represent the RGB space: linear and non-linear. The aforementioned system shows the non-linear model, whose abbreviation is $R'G'B'$

³Theory started around 1500 when Leonardo da Vinci concluded that colors are produced by mixing yellow and blue, green and red, and white and black. In 1950, this theory was confirmed when optically-colored signals were detected at the optical connection between the retina and the brain (Gevers *et al.*, 2012).

and is most used by devices and applications because of their similarity to the human visual system. In the literature, this system is frequently cited with the acronym RGB, which makes the nomenclature dubious, since the linear model is also called RGB and, therefore, the conversion between color spaces must be done with some caution. It is also important to note that linear RGB values are rarely used to represent an image since they are perceptually highly non-uniform (Plataniotis and Venetsanopoulos, 2000).

3.5.4 CMY color model

The CMY model is based on the complementary primary colors of the R, G, and B coordinates, which are given by Cyan, Magenta, and Yellow. Unlike RGB, it is a subtractive color model, and so, non-additive, in which colors are generated by subtracting the length of the dominant wave from the white light and, therefore, the resulting color corresponds to the light that is reflected (Gonzalez and Woods, 2002).

One way to get the CMY coordinates from the RGB system is:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} B \\ R \\ G \end{bmatrix} + \begin{bmatrix} G \\ B \\ R \end{bmatrix} \quad (3.7)$$

or by making a change of coordinates by subtracting the primary colors R, G and B of the white color $W = (1, 1, 1)$ (Gonzalez and Woods, 2002):

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3.8)$$

Similarly to RGB, CMY is device dependent. The model is widely used in equipment that deposits colored pigments on paper, such as color printers or photocopiers. The Figure 3.10 shows how the model components are combined to generate the other colors.

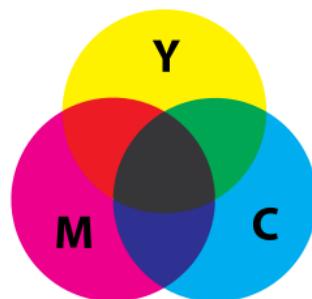


Figure 3.10: CMY subtractive color model. It is noteworthy that the intersection of yellow with magenta generates the red color, magenta with cyan generates the blue color and cyan with yellow generates the green color. Source: Rus (2008).

Overlapping the CMY primary colors in equal amounts to generate the black color typically creates a tint that is close to the brown or dark green. To avoid this undesired effect, the black component is usually added to the system, represented by the letter K. This operation gives rise to a new model known as CMYK (Gonzalez and Woods, 2002).

3.5.5 Color models of the YUV family

Color models of the YUV family are also known as orthogonal color spaces. In addition, one of their key features is to reduce the redundancy present in RGB color channels (Kakumanu *et al.*, 2007).

The acronym YUV stands to a set of color spaces of which the luminance information, represented by the Y component, is coded separately from the chrominance, given by the components U and V. The components U and V are representations of signals of the difference of the blue subtracted from luminance ($B - Y$) and red subtracted from luminance ($R - Y$). It is used to represent colors in analog television transmission systems in the Phase Alternating Line (PAL) and Sequential Color with Memory (SECAM) (Pedrini and Schwartz, 2008).

The transformation of the RGB space to YUV is given by (Pedrini and Schwartz, 2008):

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3.9)$$

where $0 \leq R, G, B \leq 1$.

Analogous to the YUV, the YIQ model was adopted in 1950 by the National Television System Committee (NTSC), an American standard for color television signal transmission. In this model, the Y component corresponds to luminance and the components I and Q encode the chrominance information (Pedrini and Schwartz, 2008).

The transformation of the RGB space to YIQ is given by (Pedrini and Schwartz, 2008):

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & -0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3.10)$$

where $0 \leq R, G, B \leq 1$.

Another color model of the YUV family is the YCbCr, mathematically defined by a coordinate transformation with respect to some RGB space (Pedrini and Schwartz, 2008). The Figure 3.11 shows the CbCr plane of the YCbCr color model for a specific Y value.

The YCbCr model is widely used in digital videos. In this system, the Y component represents luminance, computed as a weighted sum of RGB values. Cb component gives the measurement of the difference between the blue color and a reference value, similar to the Cr component which is the measurement of the difference between the red color and a reference value (Pedrini and Schwartz, 2008).

The transformation of the RGB space to YCbCr is given by (Pedrini and Schwartz, 2008):

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3.11)$$

3.5.6 Color models of the HSI family

Hue, Saturation, and Intensity (HSI) models are best suited for image processing applications from the user point of view, due to the correlation with human perception of the color (Plataniotis and Venetsanopoulos, 2000).

In this model, as well in YIQ color space, the intensity given by I component is decomposed from the chrominance information, represented by the hue (H) and saturation (S).

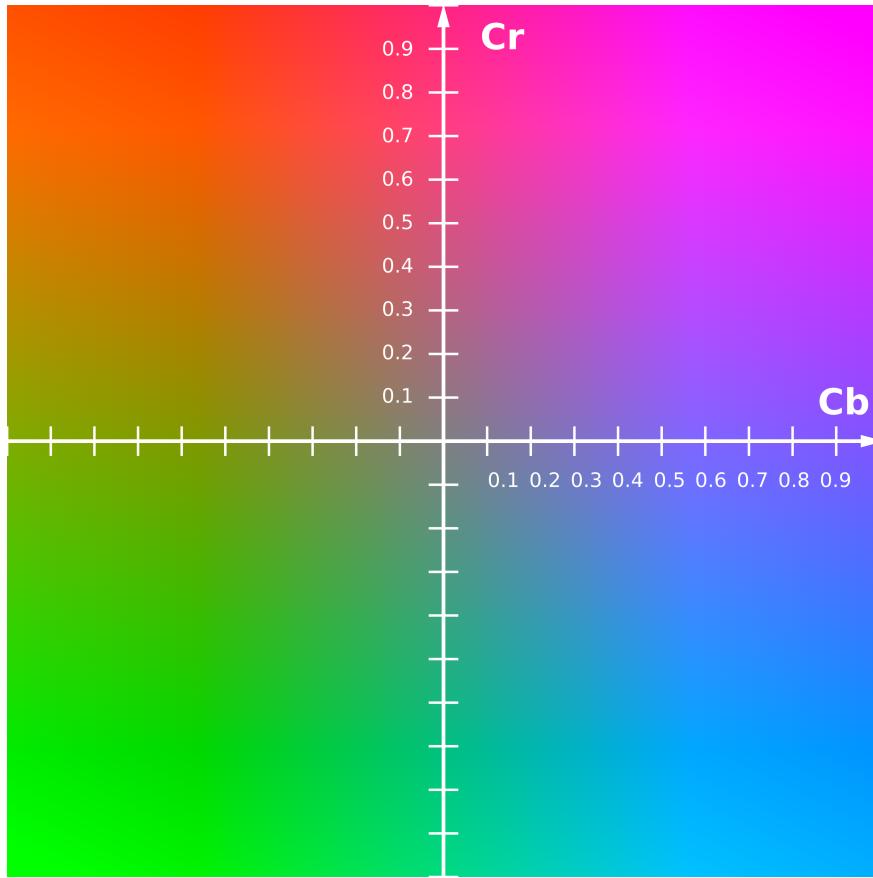


Figure 3.11: *CbCr plane of the YCbCr color model. Y value is set to 0.5, Cb on the horizontal axis in the range [-1, 1], Cr on the vertical axis in the range [-1, 1]. Source: Eugster (2010).*

The combination of these components results in a pyramidal structure which can be seen in Figure 3.12 (Plataniotis and Venetsanopoulos, 2000).

The transformation of the components of the RGB space to HSI is given by the equations (Pedrini and Schwartz, 2008):

$$\theta = \cos^{-1} \left(\frac{(R - G) + (R - B)}{2\sqrt{(R - G)^2 + (R - B)(G - B)}} \right)$$

$$H = \begin{cases} \theta, & \text{if } B \leq G \\ 360 - \theta, & \text{otherwise} \end{cases}$$

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

$$I = \frac{R + G + B}{3}$$
(3.12)

It is important to note that the values R, G, and B must be normalized in the interval between 0 and 1. The intensity I and the saturation S are also normalized between 0 and 1.

Another model of this family is formed by the components Hue, Saturation and Value (HSV) and its three-dimensional graphical representation is a hexagonal pyramid derived from the RGB cube (Pedrini and Schwartz, 2008). Value, in this context, is the luminance component.

The various hue shades are represented at the top of the pyramid, the saturation is

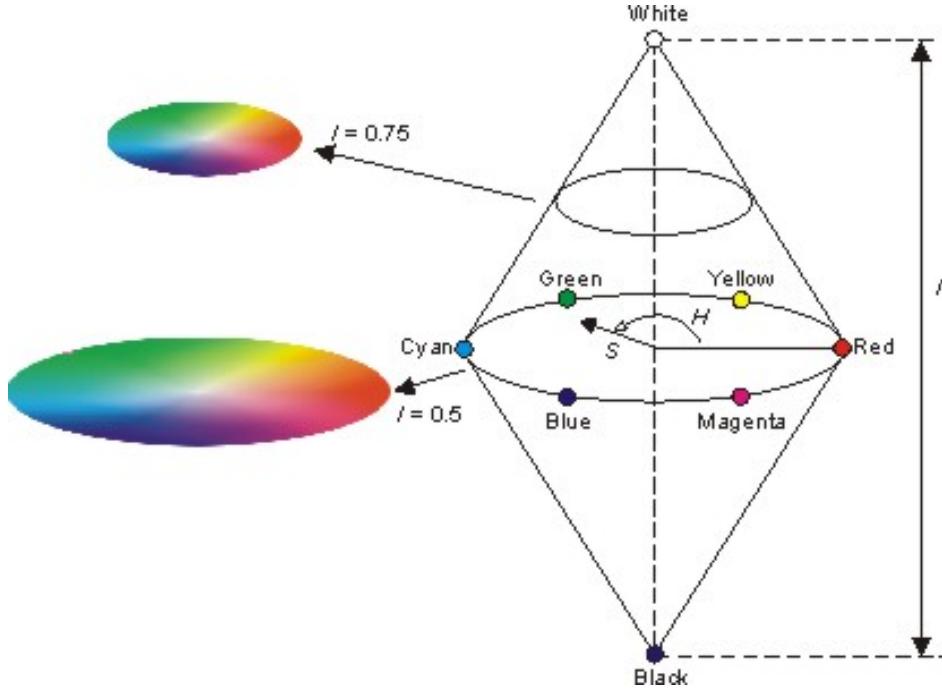


Figure 3.12: Graphical representation of the HSI model. The hue describes the color itself, in the form of an angle θ , where $\theta \in [0, 360]$. Red is at 0 degrees, yellow at 60, green at 120, and so on. The saturation component, which varies between 0 and 1, indicates how much color is polluted with white color. The intensity scale is between [0, 1], where 0 means black and 1, white. Source: [Ice \(2016\)](#).

measured along the horizontal axis and value is measured along the vertical axis, which passes through the center of the pyramid. The hue, which corresponds to the edges around the vertical axis, varies from 0 (red) to 360 degrees and the angle between the vertices is 60 degrees. The saturation varies from 0 to 1 and it is represented as the ratio of the purity of a given hue to its maximum purity, that is when $S = 1$. Value varies from 0, at the peak of the pyramid representing the black color, to 1 at the base, where the intensities of the colors are maximum (Pedrini and Schwartz, 2008).

The transformation of the components of the RGB space to HSV is given by the equations (Pedrini and Schwartz, 2008):

$$H = \begin{cases} 60 \frac{(G - B)}{M - m}, & \text{if } M = R \\ 60 \frac{(B - R)}{M - m} + 120, & \text{if } M = G \\ 60 \frac{(R - G)}{M - m} + 240, & \text{if } M = B \end{cases} \quad (3.13)$$

$$S = \begin{cases} \frac{(M - m)}{M}, & \text{if } M \neq 0 \\ 0, & \text{otherwise} \end{cases}$$

$$V = M$$

where $m = \min(R, G, B)$ and $M = \max(R, G, B)$. The luminance V and saturation S are normalized between 0 and 1. The H hue ranges from 0 to 360 degrees.

Similarly to HSV, the Hue, Saturation, and Lightness (HSL) model is a three-dimensional

representation (cylindrical geometry) and is formed by two cones of height 1, whose bases are coincident (Pedrini and Schwartz, 2008).

The hue is determined by the points in the circle of the common bases to the cones. The saturation varies from 0 to 1, depending on the distance to the axis of the cone. The lightness is along the vertical axis common to the two cones and varies in the range [0, 1], where 0 means black and 1, white (Pedrini and Schwartz, 2008).

The conversion of the RGB space to HSL is given by the equations (Pedrini and Schwartz, 2008):

$$H = \begin{cases} 60 \frac{(G - B)}{M - m}, & \text{if } M = R \\ 60 \frac{(B - R)}{M - m} + 120, & \text{if } M = G \\ 60 \frac{(R - G)}{M - m} + 240, & \text{if } M = B \end{cases} \quad (3.14)$$

$$S = \begin{cases} \frac{(M - m)}{M + m}, & \text{if } 0 < L \leq 0,5 \\ \frac{(M - m)}{2 - (M + m)}, & \text{if } L > 0,5 \\ 0, & \text{if } M = m \end{cases}$$

$$L = \frac{M + m}{2}$$

where $m = \min(R, G, B)$ and $M = \max(R, G, B)$. The lightness L and saturation S are normalized between 0 and 1. Note that the transformation of the H component is the same as that used in the conversion of the RGB to HSV space in Eq. 3.13 and varies between 0 and 360 degrees.

All the color models of this family have the property of thinking of lighter colors, obtained by increasing the brightness or lightness, and darker colors, by the diminution of the same values. The intermediate colors are produced by decreasing the saturation (Pedrini and Schwartz, 2008).

Chapter 4

Correlation rules and proposed extension

A state of the art skin detection method has been recently developed by [Brancati *et al.* \(2017\)](#). In this chapter, we review the method and extend it¹ adding more rules to enforce the constraints and seeking for a better accuracy in terms of false positive rate without hurting the performance of the original method.

4.1 Correlation rules on YCrYCb colormap

The pixels of human skin have a very particular color. They fall into a restricted range of hues and they are not deeply saturated. This phenomenon is due the appearance of skin: formed by a combination of blood (red) and melanin (brown, yellow), which leads the human skin color to be clustered within a small area in the color space ([Fleck *et al.*, 1996](#)).

Although this cluster can be seen in different color spaces, authors often use those where it is possible to split the chrominance from the luminance information. YCbCr is one of these color spaces. [Chai and Ngan \(1999\)](#) firstly observed this cluster within this particular color space (see Fig. 4.1). Based on a given set of training images, they ([Chai and Ngan, 1999](#)) built a skin color map using a histogram approach. In this map, the Cr and Cb distributions of skin color fall into the ranges [133, 173] and [77, 127], respectively, regardless the skin color variation in different races (see Fig. 4.2).

Therefore, a very simple and practical approach to detect human skin pixels would be to create a set of rules, based in those ranges, that identify the presence of chrominance (Cr, Cb) values who fit into the rules. In fact, this was the approach used by [Chai and Ngan \(1999\)](#).

Another important finding regarding the skin color clusters in the YCbCr color space is their behavior when looking for the compositions of YCb and YCr separately. In other words, where rely this distribution into the YCb and YCr subspaces. In Figure 4.3, we can see the distribution (clusters) for an image of the Pratheepan dataset. We can clearly see their shapes as taking a trapezoidal form ([Hsu *et al.*, 2002](#)).

In fact, the skin color pixels distribution in the YCb and YCr subspaces is a pattern. However, this trapezoidal shape and size will change according to many factors. [Brancati *et al.* \(2017\)](#) observed that change and identified that they are caused mainly due illumination conditions (i.e. the lighting of the scene when the image was acquired influences the size, height, and position of these trapezoidal shape). Moreover, they ([Brancati *et al.*, 2017](#)) observed an inversely proportional behavior of the chrominance components (Cr, Cb) that could be fitted into a model for skin pixels detection. We will explain in details how this

¹All the implementations can be found at <https://bitbucket.org/rodrigoafaria/skin-detector/>.

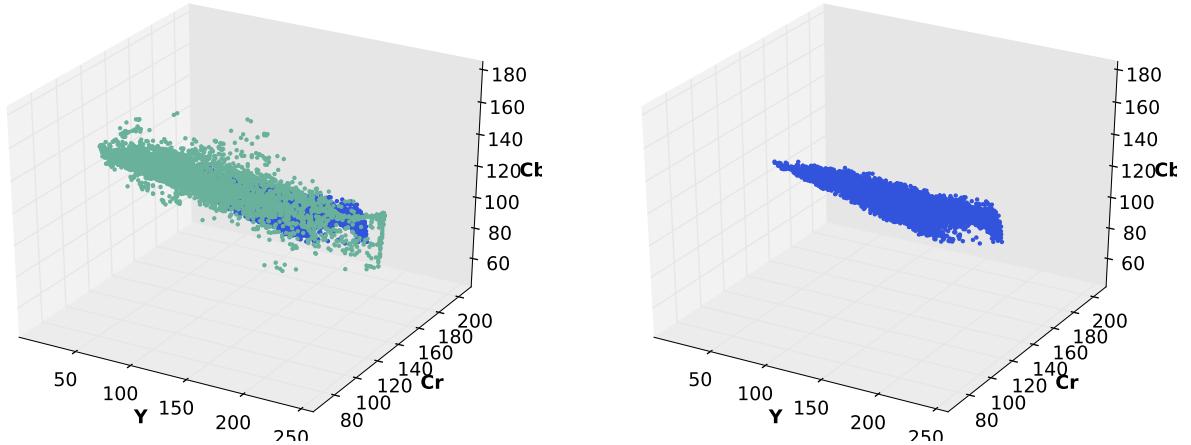


Figure 4.1: 3-dimensional view of the YCbCr channels of some image patches of the SFA dataset. We used the patches with skin samples of size 15×15 . The blue points are skin samples and the green ones are non-skin. On the right (skin samples only), we can clearly see a narrow and thin cluster. Source: adapted from Chai and Ngan (1999).

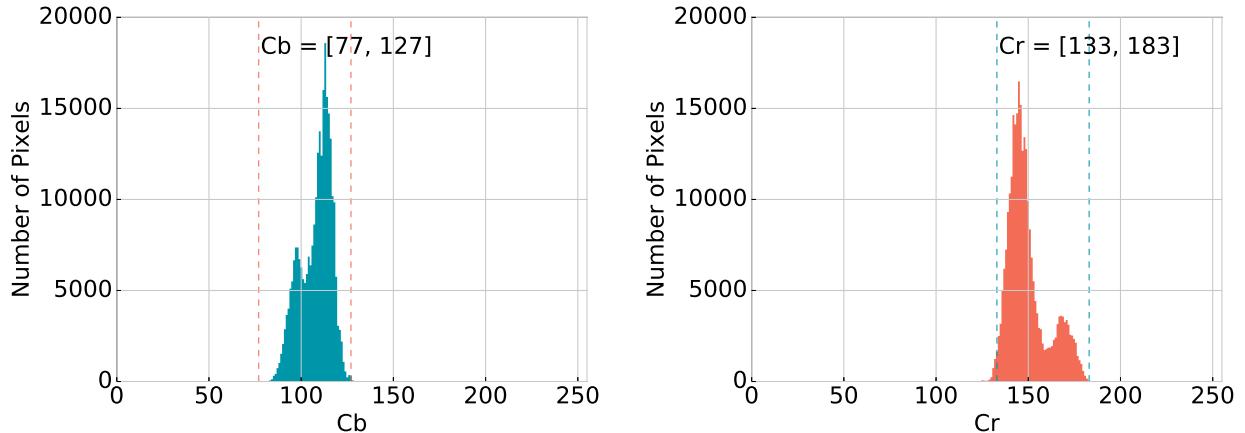


Figure 4.2: Histogram of Cb and Cr channels of some image patches of the SFA dataset. We used the patches skin samples of size 15×15 . Clearly, the samples (pixels) fall into the intervals observed by Chai and Ngan (1999). Source: adapted from Chai and Ngan (1999).

model has been created in Section 4.2.

4.2 Original method

In order to describe the proposed extensions, we will first present the original method that is based on the definition of image-specific trapezoids, named T_{YCb} and T_{YCr} , in the YCb and YCr subspaces, respectively. The trapezoids are essential to verify a relationship between the chrominance components Cb and Cr in these subspaces (Brancati *et al.*, 2017).

To show the correlations, Brancati et. al. present the YCbCr space as a 2D graph where the Y is presented in the abscissa and the Cr and Cb components are in the ordinate (see Fig. 4.4). The base of the trapezoids T_{YCr} and T_{YCb} are given by the coordinates (Y_{min}, Cr_{min}) and (Y_{min}, Cb_{max}) in the YCr and YCb , respectively (Brancati *et al.*, 2017). The values $Cr_{min} = 133$, $Cb_{max} = 128$ were selected according to Chai and Ngan (1999) where a skin color map was designed using a histogram approach based on a given set of training images. Chai and Ngan observed that the Cr and Cb distributions of skin color fall

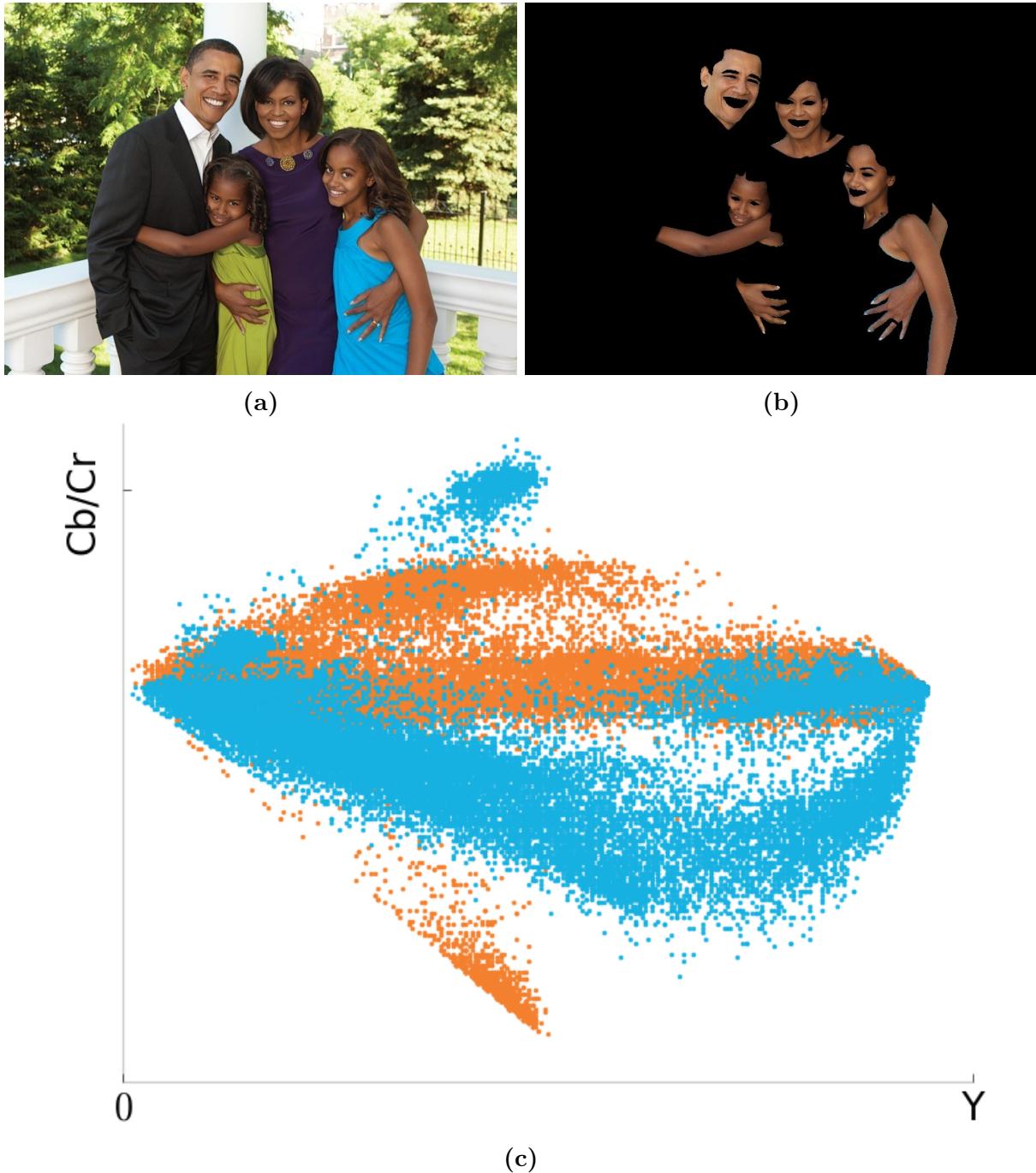


Figure 4.3: Skin pixels distribution in the YCr and YCb subspaces of a sample image. Each image is, respectively, (a) sample image from Pratheepan (b) ground truth (c) skin pixels distribution in YCr (orange) and YCb (blue) subspaces. We can clearly see a trapezoidal shape of the pixels distribution. These trapezoids are inversely positioned reflecting the inversely proportional behavior of the chrominance components (Cr , Cb). Source: adapted from Brancati et al. (2017).

in the ranges [133, 173] and [77, 127], respectively, regardless of the skin color variation in different races (see details in Section 4.1).

The Cr_{max} parameter is calculated dynamically, taking into account the histogram of the pixels with Cr values in the range $[Cr_{min}, 183]$, looking for the maximum value of Cr associated with at least 0.1%² of pixels in the image. The same applies to Cb_{min} , taking

²In Brancati et al. (2017) this rate is reported to be equal to 10%. However, in the distributed source

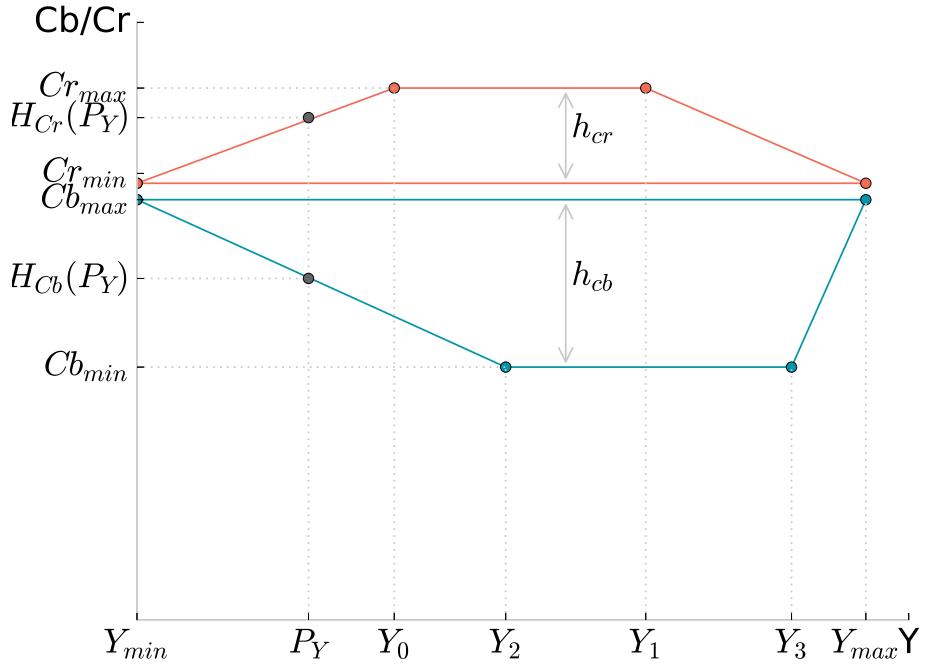


Figure 4.4: Graphical representation of the trapezoids as well as the parameters $Y_{min} = 0$, $Y_{max} = 255$, Y_0 , Y_1 , Y_2 , Y_3 , Cr_{min} , Cr_{max} , Cb_{min} , Cb_{max} , h_{Cr} , h_{Cb} , $H_{Cr}(P_Y)$, $H_{Cb}(P_Y)$. Source: adapted from (Brancati et al., 2017).

the histogram with Cb values in the range $[77, Cb_{max}]$. Y_0 and Y_1 (shorter base of the upper trapezoid) are, respectively, the 5th and 95th percentile of the luminance values associated with the pixels of the image with $Cr = Cr_{max}$ (Brancati et al., 2017). A similar procedure is used to find the values of the shorter base of the other trapezoid, Y_2 and Y_3 (see Fig. 4.5 for an example).

The correlation rules' parameters between the chrominance components P_{Cr} and P_{Cb} of a pixel P are specified as (Brancati et al., 2017):

- the minimum difference between the values P_{Cr} and P_{Cb} , denoted I_P ;
- an estimated value of P_{Cb} , namely P_{Cb_s} ;
- the maximum distance between the points (P_Y, P_{Cb}) and (P_Y, P_{Cb_s}) , denoted J_P .

Therefore, to determine if P is skin, the following correlation rules, expressed in terms of equations, must hold (Brancati et al., 2017):

$$P_{Cr} - P_{Cb} \geq I_P \quad (4.1)$$

$$|P_{Cb} - P_{Cb_s}| \leq J_P \quad (4.2)$$

code we found the value 0.1%, that we are using in the experiments.

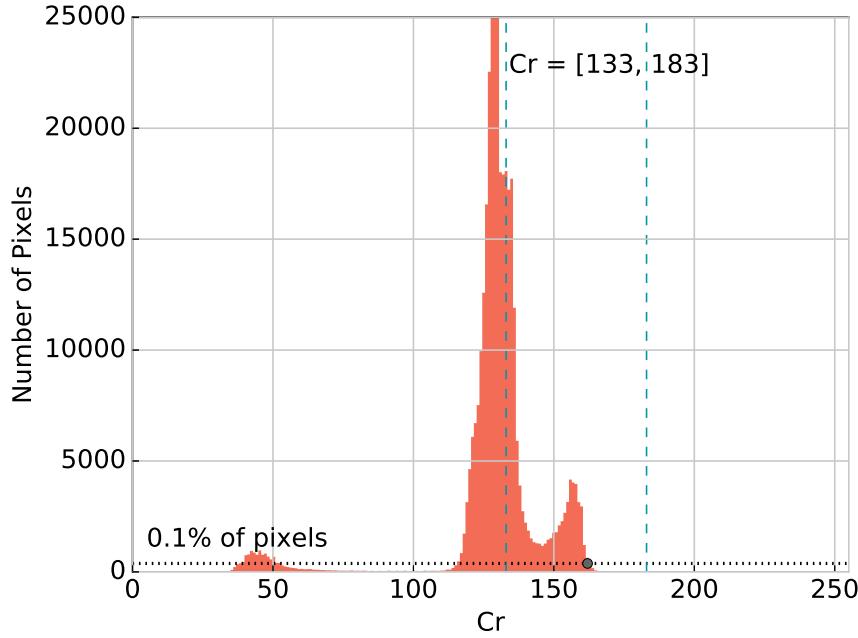


Figure 4.5: Computation of $Cr_{max} = 162$ based on Cr values histogram of a 724×526 image. Source: adapted from (Brancati et al., 2017).

The estimated value P_{Cb_s} is given by ³:

$$P_{Cb_s} = Cb_{max} - dP_{Cb_s} \quad (4.3)$$

where ⁴:

$$dP_{Cb_s} = \alpha \cdot dP_{Cr} \quad (4.4)$$

$$dP_{Cr} = P_{Cr} - Cr_{min} \quad (4.5)$$

The coordinates of the other sides of the trapezoids are given by $[P_Y, H_{Cr}(P_Y)]$ and $[P_Y, H_{Cb}(P_Y)]$, such that (Brancati et al., 2017):

$$H_{Cr}(Y) = \begin{cases} Cr_{min} + h_{Cr} \left(\frac{Y - Y_{min}}{Y_0 - Y_{min}} \right) & Y \in [Y_{min}, Y_0] \\ Cr_{max} & Y \in [Y_0, Y_1] \\ Cr_{min} + h_{Cr} \left(\frac{Y - Y_{max}}{Y_1 - Y_{max}} \right) & Y \in [Y_1, Y_{max}] \end{cases} \quad (4.6)$$

$$H_{Cb}(Y) = \begin{cases} Cb_{min} + h_{Cb} \left(\frac{Y - Y_2}{Y_{min} - Y_2} \right) & Y \in [Y_{min}, Y_2] \\ Cb_{min} & Y \in [Y_2, Y_3] \\ Cb_{min} + h_{Cb} \left(\frac{Y - Y_3}{Y_{max} - Y_3} \right) & Y \in [Y_3, Y_{max}] \end{cases} \quad (4.7)$$

where $h_{Cr} = Cr_{max} - Cr_{min}$ and $h_{Cb} = Cb_{max} - Cb_{min}$, which are the heights of T_{YCr} and T_{YCb} , respectively.

The computation of those points are useful for the calculation of α . We first compute the distances $\Delta_{Cr}(P_Y)$ and $\Delta_{Cb}(P_Y)$ between the points $(P_Y, H_{Cr}(P_Y))$, $(P_Y, H_{Cb}(P_Y))$ and

³ dP_{Cb_s} is the distance between the points (P_Y, P_{Cb_s}) and (P_Y, Cb_{max}) in the YCb subspace, calculated on the basis of dP_{Cr} , observing the inversely proportional behavior of the components. α is the rate between the normalized heights of the trapezoids in relation to the P_Y value (Brancati et al., 2017).

⁴ dP_{Cr} is the distance between (P_Y, P_{Cr}) and (P_Y, Cr_{min}) points in the YCr subspace (Brancati et al., 2017).

the base of the trapezoids (Brancati *et al.*, 2017):

$$\Delta_{Cr}(P_Y) = H_{Cr}(P_Y) - Cr_{min} \quad (4.8)$$

$$\Delta_{Cb}(P_Y) = Cb_{max} - H_{Cb}(P_Y) \quad (4.9)$$

Next, the distances are normalized with respect to the difference in size of the trapezoids (Brancati *et al.*, 2017):

$$\Delta'_{Cr}(P_Y) = \begin{cases} \Delta_{Cr}(P_Y) \cdot \frac{A_{T_{YCr}}}{A_{T_{YCr}}} & \text{if } A_{T_{YCr}} \geq A_{T_{YCb}} \\ \Delta_{Cr}(P_Y) & \text{otherwise} \end{cases} \quad (4.10)$$

$$\Delta'_{Cb}(P_Y) = \begin{cases} \Delta_{Cb}(P_Y) & \text{if } A_{T_{YCr}} \geq A_{T_{YCb}} \\ \Delta_{Cb}(P_Y) \cdot \frac{A_{T_{YCr}}}{A_{T_{YCb}}} & \text{otherwise} \end{cases} \quad (4.11)$$

where $A_{T_{YCr}}$ and $A_{T_{YCb}}$ are the areas of trapezoid T_{YCr} and T_{YCb} , respectively.

Then, the value of α is given by (Brancati *et al.*, 2017):

$$\alpha = \frac{\Delta'_{Cb}(P_Y)}{\Delta'_{Cr}(P_Y)} \quad (4.12)$$

Finally, I_P ⁵ and J_P are given by (Brancati *et al.*, 2017):

$$I_P = sf \cdot |(\Delta'_{Cr}(P_Y) - dP_{Cr}) + (\Delta'_{Cb}(P_Y) - dP_{Cb_s})| \quad (4.13)$$

$$J_P = dP_{Cb_s} \cdot \frac{dP_{Cb_s} + dP_{Cr}}{\Delta'_{Cb}(P_Y) + \Delta'_{Cr}(P_Y)} \quad (4.14)$$

where:

$$sf = \frac{\min((Y_1 - Y_0), (Y_3 - Y_2))}{\max((Y_1 - Y_0), (Y_3 - Y_2))} \quad (4.15)$$

4.3 Complementary method

The hypothesis assumed in the original method is based on rules that an estimated value of the point P_{Cb} , namely P_{Cb_s} , must hold in order for the correlation to be valid. On the basis of the inversely proportional behavior of the chrominance components, we will rewrite the correlation rules with respect to the P_{Cr} point.

Thus, we have to refactor the correlation rules' parameters to put them in terms of the estimated value of P_{Cr} , that we denote as P_{Cr_s} ⁶:

$$P_{Cr_s} = dP_{Cr_s} + Cr_{min} \quad (4.16)$$

where⁷:

$$dP_{Cr_s} = \alpha \cdot dP_{Cb} \quad (4.17)$$

⁵There is a difference between the source code and the equation that defines I_P in Brancati *et al.* (2017). Basically, part of the equation must be taken its absolute value, which we have fixed here.

⁶ dP_{Cr_s} is the distance between the points (P_Y, P_{Cr_s}) and (P_Y, Cr_{min}) in the YCr subspace, calculated on the basis of dP_{Cb} , observing the inversely proportional behavior of the components. α is the rate between the normalized heights of the trapezoids in relation to the P_Y value.

⁷ dP_{Cb} is the distance between (P_Y, P_{Cb}) and (P_Y, Cb_{max}) points in the YCb subspace.

$$dP_{Cb} = Cb_{max} - P_{Cb} \quad (4.18)$$

Next, the constraints given by I_P and J_P in the Eq. 4.13 and 4.14 respectively, can be redefined as:

$$I'_P = sf \cdot |(\Delta'_{Cr}(P_Y) - dP_{Cr_s}) + (\Delta'_{Cb}(P_Y) - dP_{Cb})| \quad (4.19)$$

$$J'_P = dP_{Cr_s} \cdot \frac{dP_{Cb} + dP_{Cr_s}}{\Delta'_{Cb}(P_Y) + \Delta'_{Cr}(P_Y)} \quad (4.20)$$

Therefore, to determine if the pixel P is skin, we have to modify the correlations rules given by Eq. 4.1 and 4.2:

$$P_{Cr} - P_{Cb} \geq I'_P \quad (4.21)$$

$$|P_{Cr} - P_{Cr_s}| \leq J'_P \quad (4.22)$$

Doing this simple extension, we need now to apply the method to the same sets of images to evaluate, in fact, the inversely proportional behavior of the chrominance components. More than that, we can combine all these constraints, given by the pair equations 4.1 and 4.2, 4.21 and 4.22, to reinforce the firstly defined hypothesis.

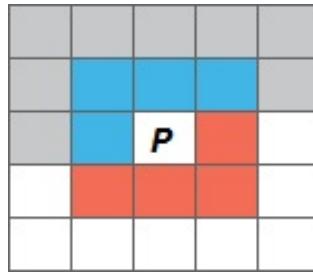


Figure 4.6: Neighbors evaluation with respect to P . If the image is scanned in raster order, $N_8^-(P)$ is the set of points that can be reached before P in an 8-neighbors window. In other words, $N_8^-(P)$ are the blue points which we already have evaluated. Source: proposed by the author.

4.4 Neighborhood extended method

Both methods presented in Sections 4.2 and 4.3 can be applied to detect skin pixels, either separated or combined (i.e. the four equations of the correlation rules of each method – original and complementary – must hold). However, skin pixels do not usually appear isolated and we could improve the method using some of the already processed neighbors of a pixel P , in order to decide if P represents human skin, or not.

To do that, let $N_8^-(P)$ be the 8-neighbors of P that can be reached before P when scanning the image in raster order (Rosenfeld and Pfaltz, 1966). We can see this idea graphically represented by the blue points in Figure 4.6.

Thus, we classify P as skin in the following manner: if the constraints given by the pair of equations 4.1 and 4.2, as well as 4.21 and 4.22 hold, then P is classified as skin. When only one of the conditions is satisfied, then we check the decision in $N_8^-(P)$. If three or more pixels are skin, then P will also be classified as a skin pixel. Figure 4.7 shows a flowchart of the aforementioned procedure described.

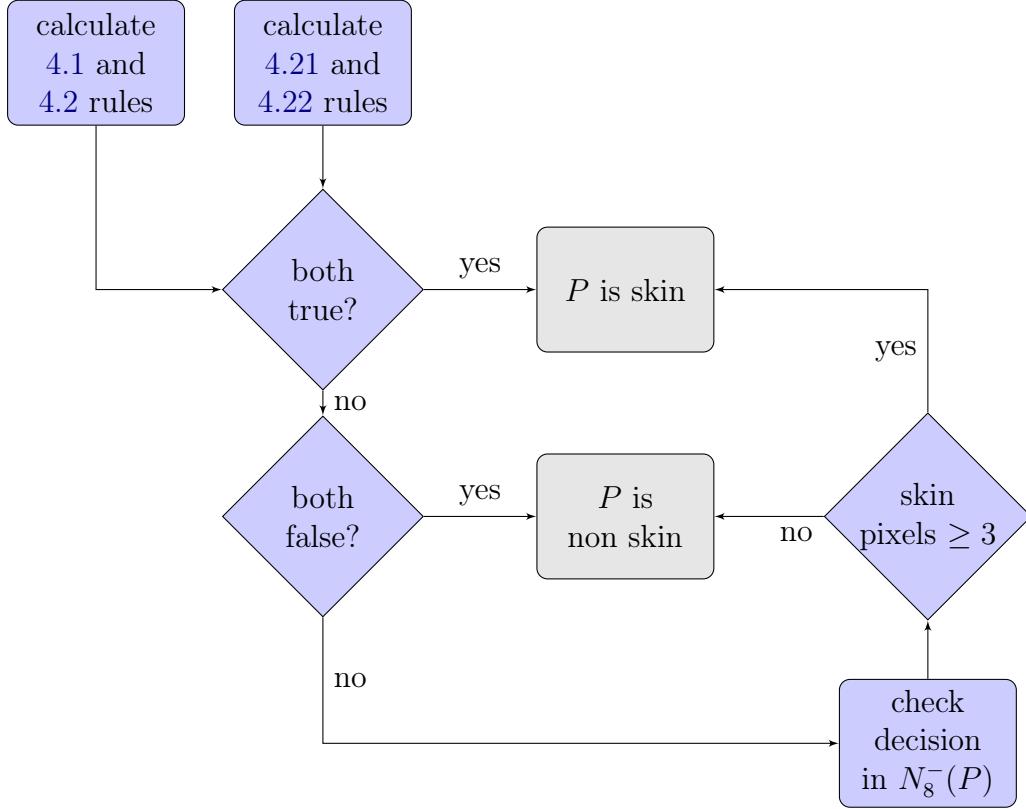


Figure 4.7: Flowchart of our proposed neighbors method. In **both false** decision, the **no** path means that one of the rules is true and we are in doubt if P is skin or not – here is where the neighbors are used to find out the label of P . Source: proposed by the author.

4.5 Heuristics to fix neighborhood extended method

The neighborhood extended method presented in Section 4.4 will end up with an undesired behavior on the output images that we called *diagonal effect* (see Fig. 4.8). In addition, besides being visually undesirable, the *diagonal effect* phenomenon causes us to have an increase in the false positive rate. This is caused due to the shape of the window being used. Once we look only for the four already visited pixels of the 8-neighbors window, the operation is so based in a non-symmetrical mask. Ideally, we could use another neighborhood strategy and look for all the eight neighbors of the pixel P being evaluated. However, this particular implementation can add extra computational time and affect the performance of the method.

Therefore, we created an adaptation of the neighborhood method shown in Section 4.4. In this version, we scan the image, with a size of $W \times H$, in the raster order, and apply the original and the extended complementary correlation rules for every single pixel. We keep both results in a matrix of the same size ($W \times H$) of the input image. For each coordinate of this output matrix, we will have a two-position vector with the result of the original and complementary rules answer for this pixel. Next, we read each position of this output matrix and we apply an 8-neighbors operations in four different implementations, looking for the majority (five at least) neighbors:

- (1) we look in the correlation rules answer performing an AND. In other words, if both original and complementary correlation rules are saying this pixel is skin, then we classify it as skin;

- (2) we look in the correlation rules answer performing an OR. In other words, if one of the correlation rules (original or complementary) is saying this pixel is skin, then we classify it as skin;
- (3) we look in the neighbors only querying the original (P_{Cb_s}) correlation rules;
- (4) we look in the neighbors only querying the complementary (P_{Cr_s}) correlation rules.

Of course, this variation will add some additional computational cost once we will scan the image one more time. This implementation can be enhanced, but the idea here is to only explore better the connectivity of the 8-neighbors window and check, on the basis of a symmetric mask window, if the *diagonal effect* is gone as well as the measures are improved. Some experiments can be seen further in Section 5.4.

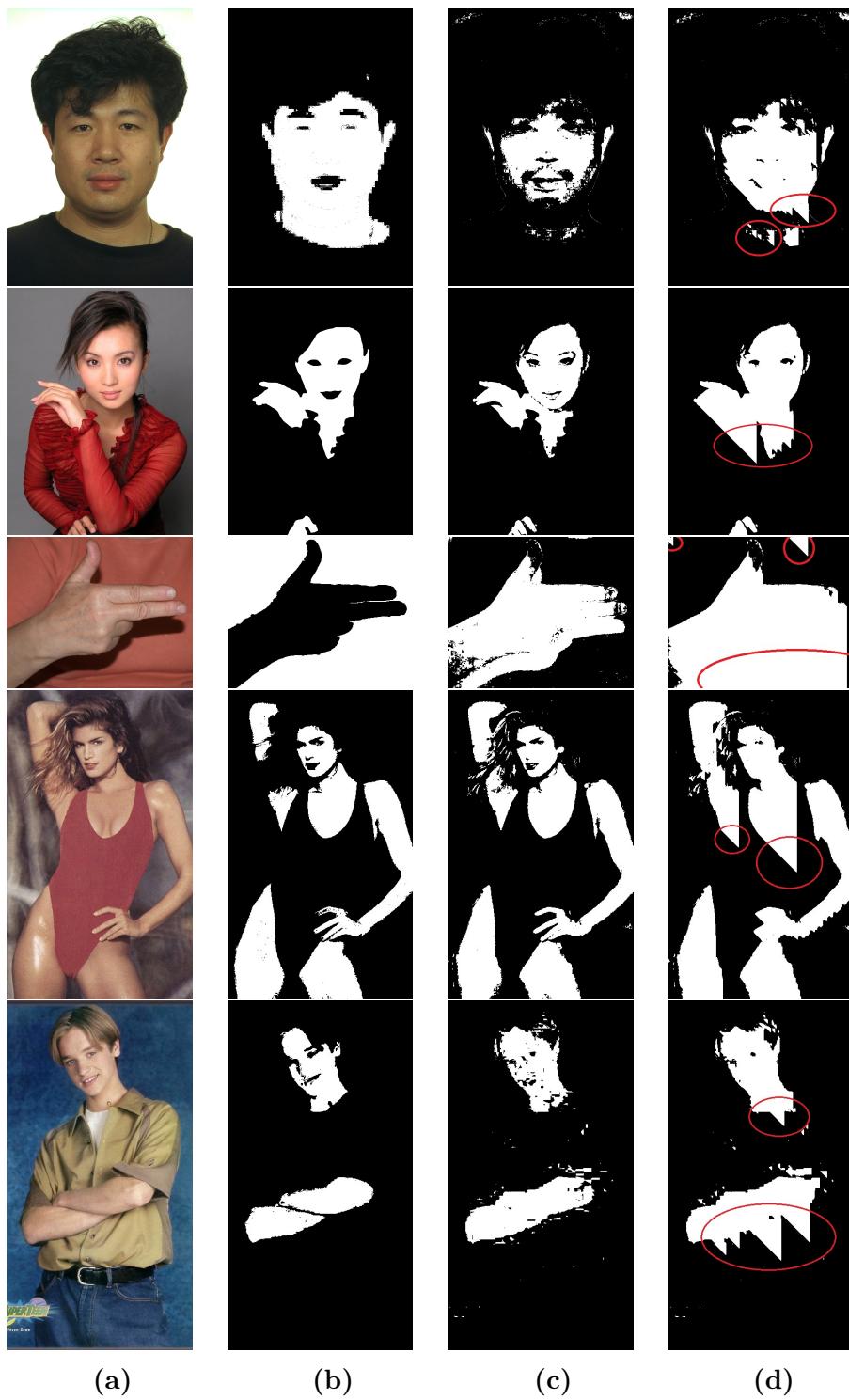


Figure 4.8: Image samples with the diagonal effect after the neighbors method segmentation. Each image is from (top-down) SFA, Pratheepan, HGR, and Compaq (latest two) datasets, respectively, where: (a) original image (b) ground truth (c) combined method (f) neighbors method. Independently of the classification accuracy, we can clearly see the diagonal effect present in the output of the neighbors method segmentation in comparison with combined. Besides being a visually undesirable effect, this phenomenon causes us to have an increase in the false positive rate.

Chapter 5

Experimental Evaluation

In this chapter, we present some experimental evaluations of the proposed extensions along with the original method in four widely known datasets: SFA, Pratheepan, HGR, and Compaq. In addition, a brief definition of the evaluation measures used is shown for the sake of clarity. We also provide some results of the heuristics to fix the neighborhood method that we built to remove the *diagonal effect* as well as to explore different neighbors techniques. Finally, we show the results of the grid search parameters experiments aiming the trapezoids parameters' tuning. All the experiments are analyzed and discussed in each section separately.

5.1 Datasets

Datasets are very important in the field of computer vision for training, testing and benchmarking. In the particular case of computer vision, datasets consist, primarily, of images or videos for tasks such as object detection, motion tracking, segmentation, and classification. In general, they are constructed with tens, hundreds, or even thousands of images in different environments, distinct illumination conditions, various quality and resolution, and many other aspects.

We gathered four widely known datasets, SFA, Pratheepan, HGR, and Compaq, which will be briefly introduced in next sections, to be used in our experiments. Together, they sum up 7,423 images of different size and resolution, and more than 1.5 billion pixels tested in all the experiments, counting only the original images.

5.1.1 SFA

SFA is an acronym for Skin of FERET and AR Database, a dataset proposed by Casati *et al.* (2013). This is a dataset of frontal faces' skins obtained from two color image databases: FERET, created by Phillips *et al.* (1996), and AR proposed by Martínez and Benavente (1998). The datasets are composed of 876 and 242 images each, respectively and AR's images have a white background and small variations of skin color, i.e., a more controlled dataset than FERET's (Casati *et al.*, 2013). Figure 5.2 shows some of the 1,118 samples available.

Casati *et al.* (2013) also extracted different window patches from each skin and non-skin samples to facilitate future research. The samples were randomly generated considering the ground truth mask ¹ of each image, being three samples of skin and five of non-skin. Each

¹Ground truth is the term used to denote an image whose point of interest is properly segmented and highlighted, discarding the remaining pixels giving them uniform colors.

sample is a image patch of size $n \times n$, where n is odd (to contain a central pixel), from which other sample sizes have been created, ranging from 1×1 to 35×35 , as can be seen in Figure 5.1.

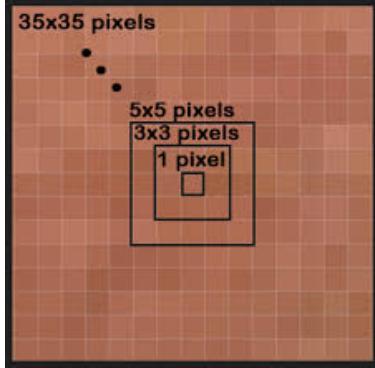


Figure 5.1: Structure of the windows that form the SFA patch samples. In total, there are 3,354 skin samples and 5,590 non-skin samples for each window size. Source: Casati *et al.* (2013).

It is worth mentioning that we do not use these patches in our experiments, since the methods tested in this work only depends on the input image itself. Therefore, we can simply ignore these sampling patches during experiments. However, one could use them to evaluate the ability of the methods in terms of false detection rate, once the patches are made available for skin and non-skin labeled images separately.

5.1.2 Pratheepan

The images in the Pratheepan dataset were downloaded randomly from Google Images for human skin detection research. There are 78 images of family and face captured with a range of distinct cameras using different color enhancement and under different illumination conditions Tan *et al.* (2012). Figure 5.3 shows some of the 78 samples available.

5.1.3 HGR

The database for Hand Gesture Recognition (HGR) contains the gestures from Polish and American Sign Language. There are 1,558 images acquired in different conditions of background, dimensions, and lightning. In addition to original and ground truth binary skin mask images, it includes hand feature points locations in separate files. Figure 5.4 shows some of the 1,558 samples available (Grzejszczak *et al.*, 2016; Kawulok *et al.*, 2014; Nalepa and Kawulok, 2014).

The images within it were acquired in three different series. A set of 899 was captured with uncontrolled background and lighting. A small set of 85 was obtained in gray (44) and uncontrolled (41) background; the lighting was uniform. The third group contains 574 images in controlled background (green tone), using uniform lighting conditions (Grzejszczak *et al.*, 2016; Kawulok *et al.*, 2014; Nalepa and Kawulok, 2014).

5.1.4 Compaq

Compaq can be considered the first large skin dataset and, probably is the most used for skin detection classifiers. It consists of 13,635 images crawled from the Internet, which 4,670 contain skin regions and another subset of 8,965 images not containing any skin.

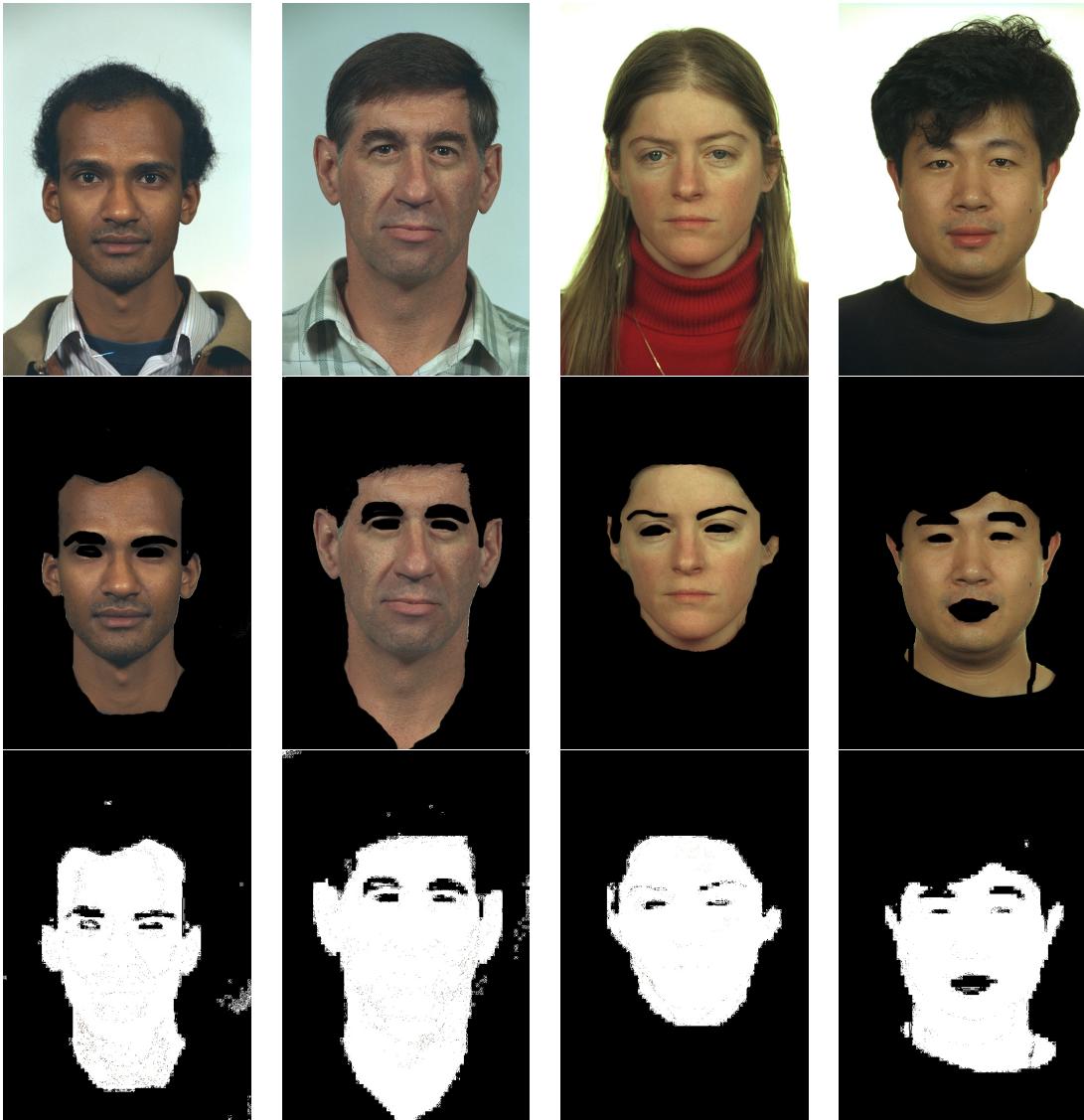


Figure 5.2: Examples of SFA face image database. Original images (first row) and colored ground truth (second row) with the skin color pixels annotated manually. The black color $RGB = (0, 0, 0)$ was assigned to all pixels in the background. Finally, binary ground truth images, that we generated, in the third row. One can see some noise in the results, but the samples were enough for further experiments. In addition, the original images were not perfectly annotated. Therefore, some salt noise can be seen in non-skin regions. Source: Casati et al. (2013).

The ground truth images are poorly annotated on the basis of an automatic software tool (Mahmoodi and Sayedi, 2016)².

We had to fix some few images due to lack of ground truth or files corrupted. The final amount of images with skin used in the experiments is 4,669. Figure 5.5 shows some of the 4,669 images with skin samples available used in the experiments (Jones and Rehg, 2002).

5.2 Evaluation measures

Precision, Recall, Specificity and *F-measure* have been used as evaluation measures. They are the same used in Brancati et al. (2017) to compare the performance with state-of-the-art

²We acknowledge the authors for sending us their database with the ground truth images.



Figure 5.3: Examples of Pratheepan skin dataset. Original images (top row), ground truth with the skin color pixels annotated (bottom row). Source: Tan et al. (2012).

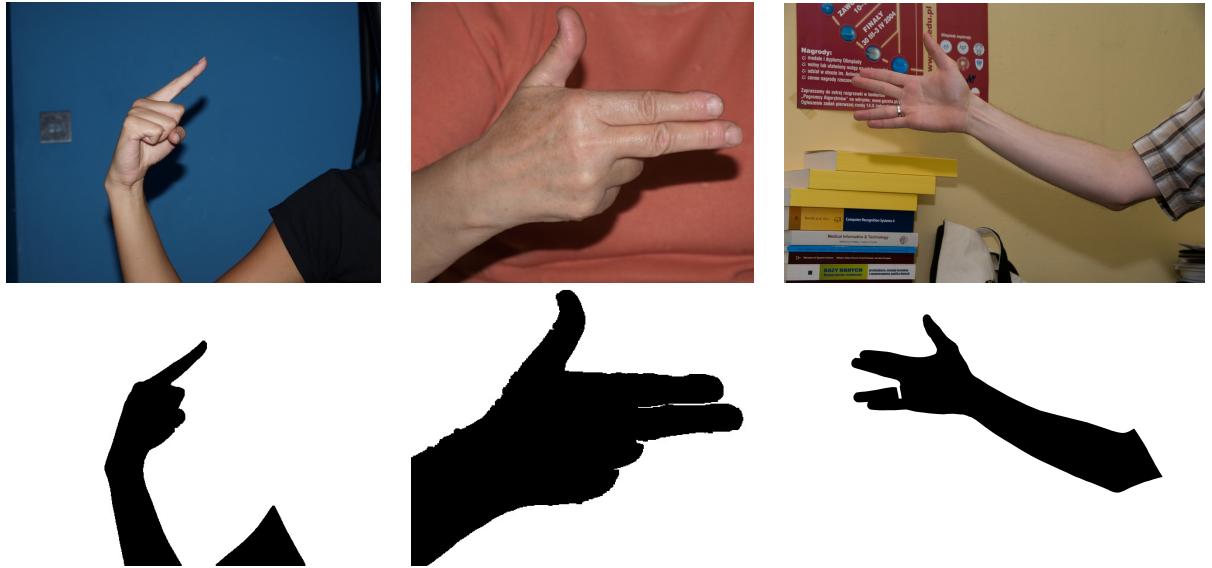


Figure 5.4: Examples of HGR skin dataset. Original images (top row) and the respective ground truth with the skin color pixels annotated (bottom row). Different from Pratheepan and SFA, the images of the ground truth are binary images with black color ($RGB = (0, 0, 0)$) assigned to all skin patches' pixels. Source: Grzejszczak et al. (2016); Kawulok et al. (2014); Nalepa and Kawulok (2014).

methods. They are also widely used by the scientific community. These measures are given by the following formulas:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$F\text{-measure} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

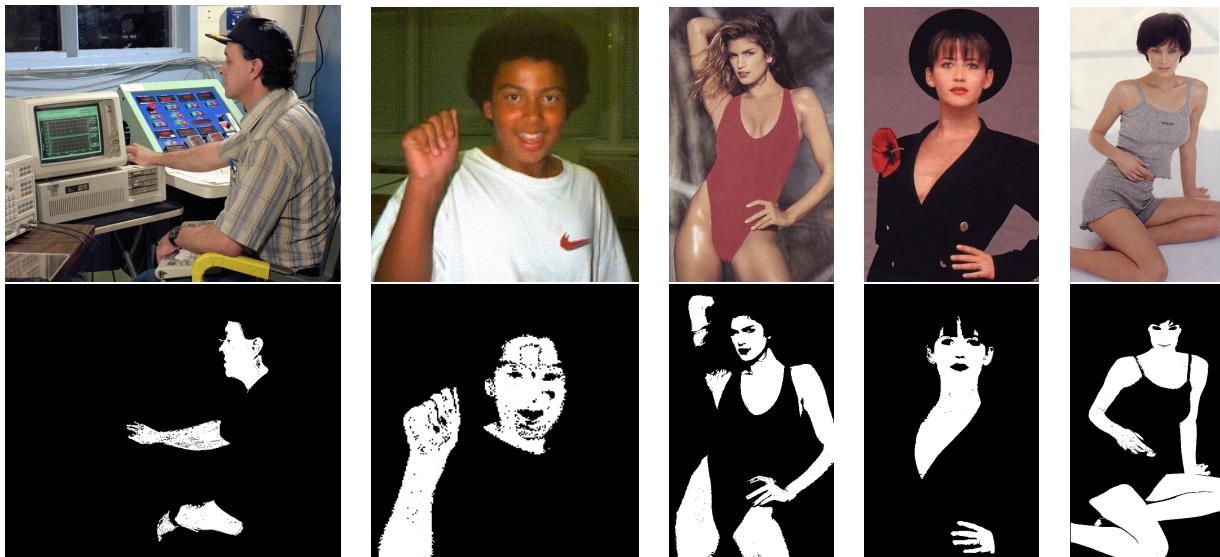


Figure 5.5: Examples of Compaq skin/non-skin dataset. Original images (top row) and the ground truth images with skin color pixels annotated (bottom row). Source: Jones and Rehg (2002).

where TP, TN, FP, FN are, respectively, the number of true positive, true negative, false positive, and false negative pixels counted in the image, which are obtained from the confusion matrix (see Table 5.1).

| | | prediction outcome | |
|--------------|----------|--------------------|----------------|
| | | skin | non-skin |
| ground truth | skin | True Positive | False Negative |
| | non-skin | False Positive | True Negative |

Table 5.1: Confusion matrix table used to count the number of true positive, true negative, false positive, and false negative pixels in the image during experiments. These numbers are fundamental input for evaluation measures.

5.3 Rule-based experiments

In this section, we present some experimental evaluations of the proposed extensions described in Sections 4.3 and 4.4, as well as the original method in four widely known datasets: SFA, Pratheepan, HGR, and Compaq. The latest three of them have also been used in Brancati *et al.* (2017).

Table 5.2 shows quantitative result measures of the experiments. Column 1 refers to the dataset used. Column 2 refers to the method being experimented: Original for the original hypothesis; Complementary refers to the complementary hypothesis with respect to P_{Cr_s} parameter; Combined refers to the combination of both of the former methods (see Sec.

4.3); Neighbors refers to the extension of the method using the neighborhood approach (see Sec. 4.4).

The original method was compared with six well known rule-based methods in the literature using four different datasets, three of them, HGR, Pratheepan and Compaq, we have also been used here. We applied the methods against a fourth dataset (SFA) to increase and strengthen the different number of samples tested.

Because the original method had the best *F-measure* in the HGR and Pratheepan datasets in comparison with the other six methods and, in addition, because it performed the top first *Precision* in HGR and second in Pratheepan, we decided to compare the proposed extensions only to the original method.

As one can see in Table 5.2, the complementary hypothesis performed better than the original method and achieved the best *Recall* in HGR and SFA. It also achieved the best *F-measure* in SFA with a 0.8125 rate, which gave almost 0.25 in gain compared to the original.

| Dataset | Hypothesis | Precision | Recall | Specificity | F-measure |
|------------|---------------|---------------|---------------|---------------|---------------|
| Compaq | Original | 0.4354 | 0.8046 | 0.8046 | 0.5650 |
| | Complementary | 0.3971 | 0.7232 | 0.7921 | 0.5127 |
| | Combined | 0.4906 | 0.6251 | 0.8856 | 0.5498 |
| | Neighbors | 0.4708 | 0.7421 | 0.8463 | 0.5761 |
| Pratheepan | Original | 0.5513 | 0.8199 | 0.8230 | 0.6592 |
| | Complementary | 0.5249 | 0.7326 | 0.8188 | 0.6116 |
| | Combined | 0.6681 | 0.6683 | 0.9164 | 0.6682 |
| | Neighbors | 0.6280 | 0.7515 | 0.8871 | 0.6843 |
| HGR | Original | 0.8938 | 0.7664 | 0.9274 | 0.8252 |
| | Complementary | 0.7929 | 0.8429 | 0.8337 | 0.8171 |
| | Combined | 0.8994 | 0.6952 | 0.9390 | 0.7843 |
| | Neighbors | 0.8818 | 0.7935 | 0.9211 | 0.8353 |
| SFA | Original | 0.8636 | 0.4214 | 0.9692 | 0.5664 |
| | Complementary | 0.8563 | 0.7730 | 0.9381 | 0.8125 |
| | Combined | 0.9288 | 0.3958 | 0.9894 | 0.5551 |
| | Neighbors | 0.9176 | 0.5111 | 0.9826 | 0.6565 |

Table 5.2: Quantitative result measures of the proposed enhancements and Brancati et al. (2017). For each dataset, we have four different applications: the original hypothesis with respect to P_{Cb_s} , the complementary hypothesis with respect to P_{Cr_s} , the one which combines both, and the extension using the neighborhood approach. The highlighted cells (bold) are those with the best result for the respective method and measure in each dataset.

It is important to note that, the significant improvement of the *Recall* in HGR and SFA datasets is, probably, due to the nature of the images that form each of them. HGR and SFA contain images where we have a great concentration of skin pixels – hands and arms in the case of HGR, and faces in the case of SFA occupying a considerable region of the image. Furthermore, a large part of them was collected in controlled environments in terms of background and lighting conditions. Thus, this behavior must be taken into consideration when selecting the method to apply within an application, since this can be decisive for a potentially performance improvement.

In general, the complementary method increased the *Recall* but did not perform well in *Precision* and *Specificity* measures. When we combined both methods, the best *Precision* and

Specificity were achieved for all datasets but it loses some performance in *Recall*. However, it still has very good *F-measure* rates.

The neighborhood approach achieved the best *F-measure* results in Compaq, HGR, and Pratheepan. Moreover, the other measures still are at a very high rate for all datasets, being in the top second in almost all cases. If we compare the neighbors approach with the original, except for the *Recall* – where we only got the best result in SFA –, we can see that our implementation achieved the best measures practically for all other measures, in the four datasets.

Therefore, the combined and neighbors approaches are very competitive compared to the original method. Furthermore, all the variations of the original method are still computed in linear time in the number of pixels, maintaining the desired computational efficiency that is useful in different application domains, mainly near real-time systems (processing time of about 10ms for a typical image of dimensions 320x400).

We have also tested the subset of non-skin images from Compaq dataset in terms of *Specificity*. Basically, this measure tells us the ability of the method to correctly identify the non-skin pixels in the image, in other words, the true negative rate. It is important to remember that the original (Brancati *et al.*, 2017) method was compared with six well known rule-based methods in the literature, and it outperformed all of them in this experiment. We can see in Table 5.3 that both combined and neighbors are in a better position, with a gain of about 5.7% and 2%, respectively, which shows us, once again, how robust are the results produced by these proposed enhancements.

| Hypothesis | Specificity |
|--|---------------|
| Original (Brancati <i>et al.</i> , 2017) | 0.8681 |
| Complementary | 0.7876 |
| Combined | 0.9177 |
| Neighbors | 0.8866 |

Table 5.3: Specificity of the proposed enhancements and original method for non-skin images of Compaq dataset. The combined method obtained the best result and neighbors is the second ranked method.

Figures 5.6, 5.7, 5.8 and 5.9, present some qualitative results for each method tested. Column (a) are image samples, column (b) presents the respective ground truth for each image in column (a), column (c) presents the original method Brancati *et al.* (2017) results, column (d) presents the respective complementary method results, column (e), the combined method results and column (f) the extended neighborhood method.

Qualitatively, we can see that, in general, the complementary method removes a large part of the reddish pixels from the background of the image (e.g. see the third and sixth images in the Figure 5.6, first and fourth images in the Figure 5.7, second image in Figure 5.8, and third image in Figure 5.9), which the original method was not able to. On the other hand, the complementary method has not been successful in taking off yellowish-toned background regions (e.g. see the third image in Figure 5.8, and fifth image in Figure 5.9).

However, when we blend the original method along with the complementary in the method that we call combined, both background regions of reddish and yellowish pixels are almost completely removed. In addition, salt-like noise is also dropped, since the correlation rules together are more rigid (e.g. see column (e) combined method for the third image in Figure 5.8, and fifth image in Figure 5.9). Obviously, this causes an undesired effect of removing some of the skin pixels of which we are interested. This helps us understand why the recall decreases with the application of the combined method.



Figure 5.6: Image samples with the results of each method in SFA dataset: (a) original image (b) ground truth (c) original Brancati et al. (2017) (d) complementary (e) combined (f) neighbors.

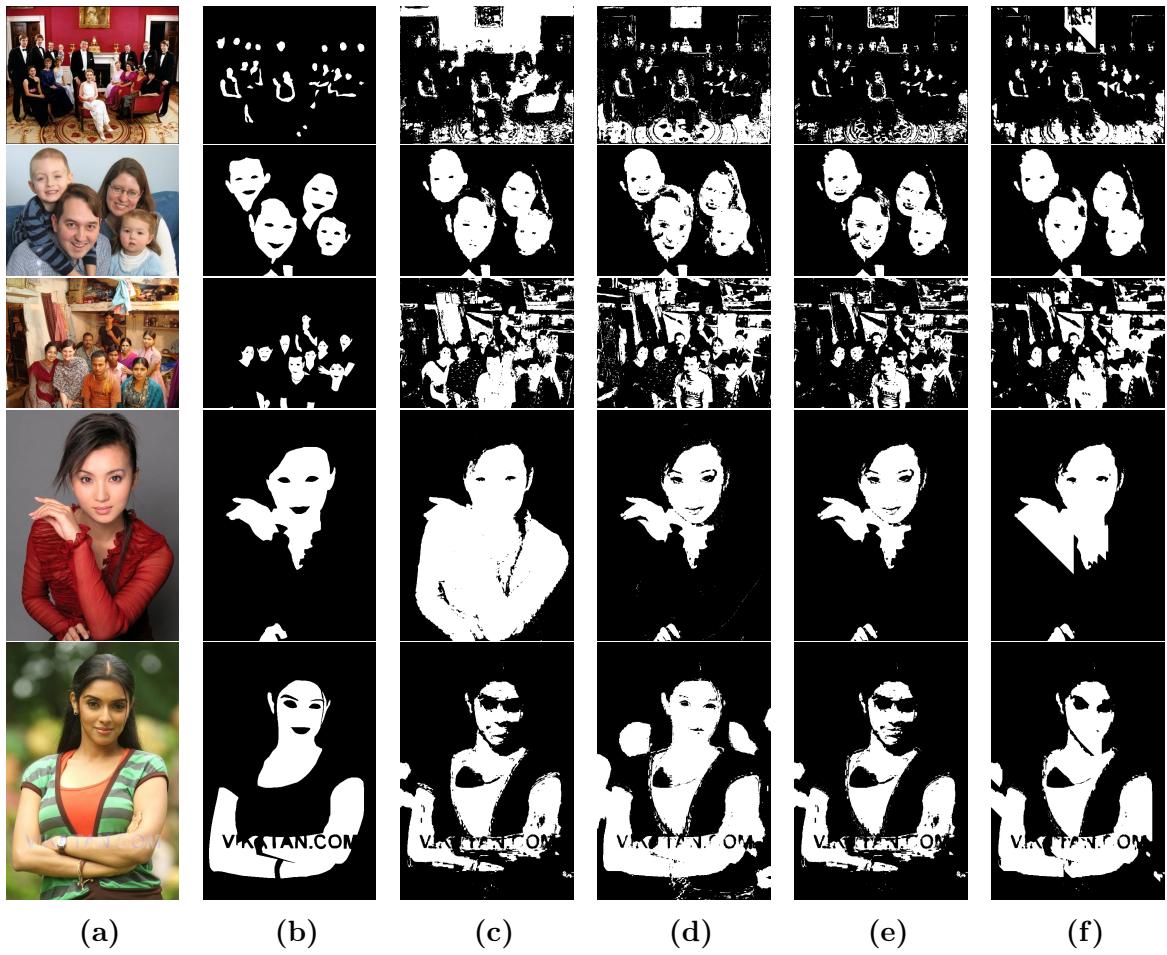


Figure 5.7: Image samples with the results of each method in Pratheepan dataset: (a) original image (b) ground truth (c) original Brancati et al. (2017) (d) complementary (e) combined (f) neighbors.

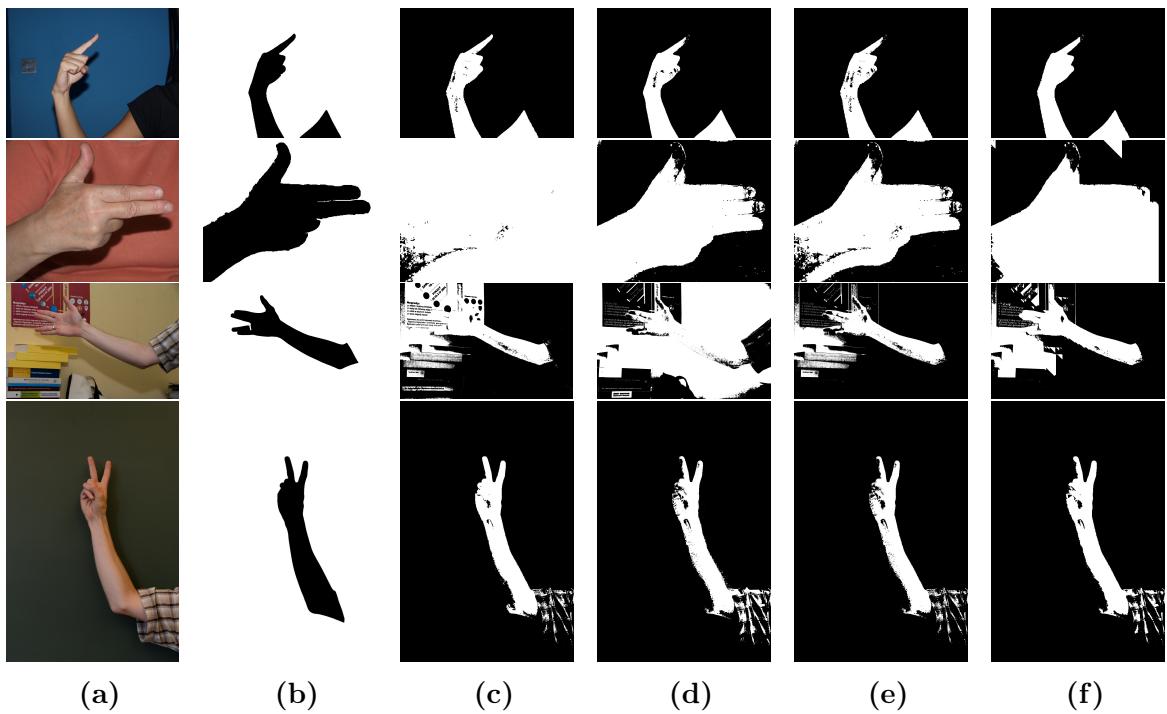


Figure 5.8: Image samples with the results of each method in HGR dataset: (a) original image (b) ground truth (c) original Brancati et al. (2017) (d) complementary (e) combined (f) neighbors.

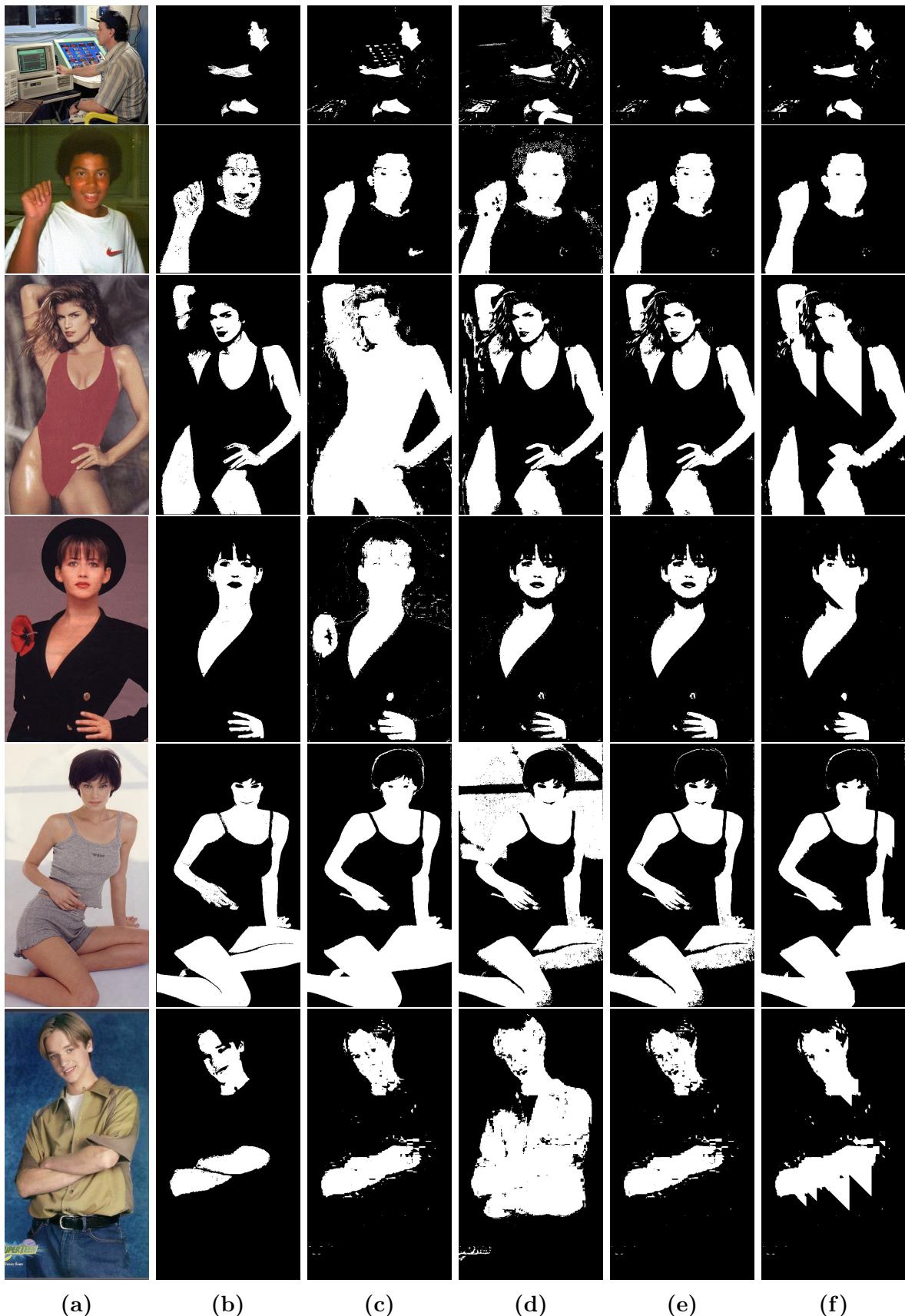


Figure 5.9: Image samples with the results of each method in Compaq dataset: (a) original image (b) ground truth (c) original Brancati et al. (2017) (d) complementary (e) combined (f) neighbors.

5.4 Experiments with heuristics to fix extended neighborhood method

In this section, we will show some experimental results of the neighborhood adaptation described in Section 4.5. In short, we scan the image, with a size of $W \times H$, in raster order, and apply the original and complementary rules for every single pixel. We keep the result in a matrix of the same size ($W \times H$) of the input image. For each coordinate of this output matrix, we will have a two-position vector with the result of the original and complementary rules answer for this pixel. Finally, we count those answers into four different strategies. The results can be seen in Table 5.4.

| Dataset | Hypothesis (Neighbors) | Precision | Recall | Specificity | F-measure |
|------------|------------------------|---------------|---------------|---------------|---------------|
| Compaq | Proposed 4.4 | 0.4708 | 0.7421 | 0.8463 | 0.5761 |
| | AND | 0.5121 | 0.6252 | 0.8941 | 0.5630 |
| | OR | 0.3786 | 0.9037 | 0.7203 | 0.5336 |
| | P_{Cr_s} only | 0.4132 | 0.7254 | 0.8032 | 0.5265 |
| | P_{Cb_s} only | 0.4478 | 0.8053 | 0.8120 | 0.5755 |
| Pratheepon | Proposed 4.4 | 0.6280 | 0.7515 | 0.8871 | 0.6843 |
| | AND | 0.6731 | 0.6789 | 0.9127 | 0.6760 |
| | OR | 0.4624 | 0.8837 | 0.7321 | 0.6072 |
| | P_{Cr_s} only | 0.5285 | 0.7414 | 0.8163 | 0.6171 |
| | P_{Cb_s} only | 0.5630 | 0.8218 | 0.8292 | 0.6682 |
| HGR | Proposed 4.4 | 0.8818 | 0.7935 | 0.9211 | 0.8353 |
| | AND | 0.9007 | 0.7203 | 0.9378 | 0.8005 |
| | OR | 0.7978 | 0.9084 | 0.8238 | 0.8495 |
| | P_{Cr_s} only | 0.7937 | 0.8600 | 0.8331 | 0.8256 |
| | P_{Cb_s} only | 0.8941 | 0.7715 | 0.9288 | 0.8283 |
| SFA | Proposed 4.4 | 0.9176 | 0.5111 | 0.9826 | 0.6565 |
| | AND | 0.9345 | 0.3947 | 0.9899 | 0.5549 |
| | OR | 0.8345 | 0.8181 | 0.9176 | 0.8262 |
| | P_{Cr_s} only | 0.8612 | 0.7922 | 0.9375 | 0.8252 |
| | P_{Cb_s} only | 0.8953 | 0.7690 | 0.9286 | 0.8273 |

Table 5.4: Quantitative result measures of the adaptation to fix neighborhood extended method. For each dataset, we have five different applications of the neighbors operations. The first line is the result of the proposed method detailed in Section 4.4 for comparison. The following four lines are the adaptation to fix neighborhood extended method shown in Section 4.5 which are considering, respectively, an AND between the original and complementary rules, an OR between the original and complementary rules, the P_{Cr_s} (complementary) only, and the P_{Cb_s} (original) only. The highlighted cells (bold) are those with the best result for the respective hypothesis and measure in each dataset.

It is worth mentioning that the goal here is to better explore the connectivity of the 8-neighbors window and check on the basis of a symmetric mask window, if the *diagonal effect* is gone, as well as if the measures were improved. For these experiments we will pay the cost of scanning the image one more time. The implementation can be improved, for instance, by keeping the latest three lines of the image scanned in memory to be verified with the 8-neighbors window backward to have a final decision when evaluating a pixel.

From the Table 5.4 we can see that the application of the neighborhood approach using

AND gives us the best *Precision* and *Specificity* in all datasets. On the other hand, we lost performance in the *Recall* in relation to the other approaches. Concerning *Recall*, the loss is in the order of 20% up to 30% in the Compaq, Pratheepan and HGR datasets, and approximately 50% in the SFA.

Clearly, there is a trade-off between increasing *Precision* and decreasing *Recall* and vice-versa, just as in the experiments done in Section 5.3 with all methods. When we use a more relaxed rule, as in the case of the OR (or the isolated rules), we get a better *Recall*, but there is also an abrupt drop in *Precision* and *Specificity*. Another side effect is to increase in the false positive rate.

The AND conjunction approach is most similar to that implemented in the neighborhood extended method, as we can see in the results of the first row (named Proposed 4.4) of each dataset. The results obtained by that approach (Proposed 4.4) show low variability in relation to the adaptations shown in the experiments in this section. In fact, we can see the best *F-measure* obtained with this method both in Compaq and Pratheepan, and the second in HGR. In addition, it performed the second best *Precision* and *Specificity* in Compaq, Pratheepan, and SFA while keeping solid results with respect to *Recall*.

Therefore, the approach given by the neighborhood implementation (Section 4.4) may be a good starting point. Nevertheless, we can apply some other complementary technique, such as weighting the distance among the pixels, to find those more or less relevant to the decision making in the neighborhood. This can potentially diminish the effect of this trade-off and obtain more harmonic measures. For the *diagonal effect*, we can use a more symmetric window, similar to the one used in the adaptations experimented here, to produce better results and increase this undesired anomaly.

5.5 Parameters tuning via a grid search strategy

As we could see in Chapter 4, this model is based on the definition of trapezoids to fit the skin color pixels distribution from a given image. According to Brancati *et al.* (2017), the coordinates of each trapezoid within the YCr and YCb sub-spaces are calculated based on the Y luminance component value. Just to remember what was seen before, Y_0 and Y_1 are those values used to define the shorter side of the upper trapezoid, and Y_2 and Y_3 are the points used to define the shorter side of the lower trapezoid.

Y_0 and Y_1 are, respectively, the 5th and 95th percentile of the luminance values associated with the pixels of the image with $Cr = Cr_{max}$ (see Fig. 4.5 for an example). Similarly, Y_2 and Y_3 are, respectively, the 5th and 95th percentile of the luminance values associated with the pixels of the image with $Cb = Cb_{min}$.

To the best of our knowledge, there is no justification to choose the 5th and 95th percentile as the right parameters to define the trapezoids coordinates. For this reason, we decided to apply different combinations of these parameters to figure out which pair better works for the model fitting. In addition, we would like to answer the question: 'why 5th and 95th percentile have been used'.

Thus, we used a well-known technique called grid search to find the best range combination from a hyper-parameter space. By constructing the model in this manner, we can leverage the classification results by finding the optimized parameters' combination, if other different from the ones defined earlier by Brancati *et al.* (2017). Despite we do not exhaustively consider all parameter combinations, we used an efficient search strategy by sampling a given number of candidates. For each chosen parameters candidate, we dynamically used

them in the combined method³ to test every single image of each dataset described in Section 5.1. Lastly, we sort the results table using respectively, *F-measure*, *Precision*, and *Recall* measures, as detailed in Section 5.2, and established a comparison to get optimized parameters. The algorithm TRAPEZOIDS-PARAMETERS-GRID-SEARCH(*dataset*) shows how the grid search have been performed.

```

TRAPEZOIDS-PARAMETERS-GRID-SEARCH(dataset)
1    $P_{min} = 5$ 
2    $Results = []$ 
3
4   while  $P_{min} \leq 95$ 
5        $P_{max} = P_{min}$ 
6       while  $P_{max} \leq 95$ 
7           SET-COMBINED-METHOD-Y-PARAMETERS( $P_{min}, P_{max}$ )
8           SEGMENT-DATASET-IMAGES(dataset)
9           Precision, Recall, Fmeasure = COMPUTE-MEASURES(dataset)
10          PUSH(Results,  $P_{min}, P_{max}, Precision, Recall, Fmeasure$ )
11
12           $P_{max} = P_{max} + 5$ 
13           $P_{min} = P_{min} + 5$ 
14
15      SORT-BY(Results, Fmeasure, Precision, Recall)
16  return Results
```

In short, TRAPEZOIDS-PARAMETERS-GRID-SEARCH(*dataset*) will apply combinations of P_{min} and P_{max} ⁴ percentiles in the range [5, 95] with a step of 5. In line 7, the parameters are changed in the combined method and applied further in 8 to classify each image within the dataset. Next, the measures are computed on every single image in line 9 by comparing the output with the ground truth. The resulting measures are pushed into *Results* matrix in line 10. We keep an average of overall images classified in each dataset. Finally, we sort the results matrix according to the criteria aforementioned at the end of the procedure.

It is worth mentioning that the grid search algorithm starts with P_{min} percentile index in 5 and P_{max} percentile lower bound is set to P_{min} in the inner loop. Thus, we will no longer have a combination of parameters such as [15, 10], that means, we always have combinations such that $P_{max} \geq P_{min}$. This will produce a table with results according to the template given in Figure 5.10.

We performed this experiment with all four datasets: Compaq, Pratheepan, HGR, and SFA. For each dataset, respectively, we produced a scatter plot of *F-measure*, *Precision*, and *Recall*, with their respective variance, which can be seen in Figures 5.11, 5.12, 5.13, and 5.14. Each chart shows the resulting measures, in *y* (vertical) axis⁵, by using a fixed P_{min} and varying P_{max} along *x* (horizontal) axis. The shaded path of the measures are the variance and tell us how far we are from the mean when looking for each image within the dataset individually. The results with these measures, for each dataset, can also be seen in Appendix A.

³In fact, we could apply this approach in any of the described methods, once trapezoids definitions do not change among them. So, we think that using combined method is sufficient for the parameters optimization.

⁴ P_{min} is the minimum percentile index of the *Y* luminance values. P_{max} is the maximum percentile index of the *Y* luminance values.

⁵We set the range in the charts to start at 0.25 for better visualization of the series.

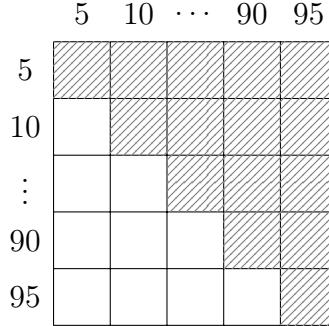


Figure 5.10: The output template table with result measures of the grid search algorithm. P_{min} percentile index is in vertical axis and P_{max} percentile index is in horizontal axis. The highlighted cells – from the main diagonal and above – are the ones which will have values of the measures after running the method for the given combination of parameters. Source: proposed by the author.

We can see predominantly that larger intervals – e.g. [5, 85], [5, 90], [5, 95] – resulted in the best *Precision* in all datasets. On the other hand, smaller intervals, with both P_{min} and P_{max} near the upper bound (95) – e.g. [80, 90], [85, 95], [90, 95] –, performed better in *Recall*. Thus, if we set up a specific pair of parameters for the percentile to compute the trapezoid coordinates, it can produce a good *Precision* but it may affect the *Recall* and vice-versa.

Therefore, evaluate *F-measure* in this situation seems to be the correct path, since it gives us a harmonic measure between *Precision* and *Recall*. That is why we sorted the results (see Tables A.1, A.2, A.3, and A.4 for details) using the order *F-measure*, *Precision*, and *Recall* as a criteria. Another important observation is the behavior of variance in each result, that means, even good measures with huge variance could be a problem for real world applications.

In Compaq, we can see that top *F-measure* results were obtained with pair of parameters such as [5, 25], ..., [5, 95], while keeping good measures in both *Precision* and *Recall*. A similar behavior can be seen in Pratheepan, but among the highest *F-measure* scores are those where $P_{min} = 10$. Next are those starting with $P_{min} = 5$. So, we could state that the values [5, 95] chosen by Brancati *et al.* (2017) are fair enough since the results are close to the highest ranked ones (see Table 5.5).

However, in HGR as well as in SFA top *F-measure* results were obtained with pair of parameters such that $P_{min} = P_{max}$. For instance, we can see in HGR results that *Recall* = 0.8641 and *F-measure* = 0.8667 when $P_{min} = 5$ and $P_{max} = 5$, while [5, 95] interval resulted in *Recall* = 0.6952 and *F-measure* = 0.7843 – a gain in the order of 24% and 10% for each measure, respectively. A similar behavior can be seen in the results of SFA, where the gain in *Recall* and *F-measure* are in the order of 67% and 39% for each measure, respectively, which is a very huge difference. It is worth mentioning that, despite *Precision* and *Specificity* have a slightly decreasing, they are still around the Brancati *et al.* (2017) results.

Based on this scenario, we observed that when we have $P_{min} = P_{max}$, the resulting measure is absolutely the same for any combination of equal percentile parameters. For instance, in Compaq dataset, when we tried $P_{min} = 5$ and $P_{max} = 5$, we got *Precision* = 0.4172, *Recall* = 0.7333, *Specificity* = 0.8096, and *F-measure* = 0.5318. The result is the same for $P_{min} = 10$ and $P_{max} = 10$ onward up to $P_{min} = 95$ and $P_{max} = 95$. This behavior will repeat for every single dataset tested.

Based on this result, we started to investigate the source code given in the original method by Brancati *et al.* (2017). We found out that, when the percentiles are the same, the computation of the parameters who define the trapezoids is impacted. In other words, $Y_0, Y_1, Y_2, Y_3, Y_{min}, Y_{max}, Cr_{min}, Cb_{max}$ are always the same, i.e. assume default values. Even

| Dataset | P_{min} | P_{max} | Precision | Recall | Specificity | F-measure |
|----------------|-----------|-----------|------------------|---------------|--------------------|------------------|
| Compaq | 5 | 95 | 0.4906 | 0.6251 | 0.8856 | 0.5498 |
| | 5 | 25 | 0.4819 | 0.6459 | 0.8762 | 0.5520 |
| | 5 | 35 | 0.4866 | 0.6374 | 0.8808 | 0.5519 |
| | 5 | 40 | 0.4877 | 0.6356 | 0.8818 | 0.5519 |
| | 5 | 30 | 0.4844 | 0.6408 | 0.8789 | 0.5517 |
| | 5 | 45 | 0.4884 | 0.6335 | 0.8827 | 0.5516 |
| Pratheepan | 5 | 95 | 0.6681 | 0.6683 | 0.9164 | 0.6682 |
| | 10 | 55 | 0.6560 | 0.6955 | 0.9055 | 0.6752 |
| | 10 | 80 | 0.6558 | 0.6950 | 0.9056 | 0.6748 |
| | 10 | 85 | 0.6557 | 0.6949 | 0.9056 | 0.6748 |
| | 10 | 95 | 0.6557 | 0.6950 | 0.9057 | 0.6747 |
| | 10 | 90 | 0.6556 | 0.6950 | 0.9056 | 0.6747 |
| HGR | 5 | 95 | 0.8994 | 0.6952 | 0.9390 | 0.7843 |
| | 5 | 5 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| | 10 | 10 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| | 15 | 15 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| | 20 | 20 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| | 25 | 25 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| SFA | 5 | 95 | 0.9288 | 0.3958 | 0.9894 | 0.5551 |
| | 5 | 5 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| | 10 | 10 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| | 15 | 15 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| | 20 | 20 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| | 25 | 25 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |

Table 5.5: Quantitative result measures of the proposed grid search parameters tuning. For each dataset, we have different applications of the combined method with different pairs of P_{min} and P_{max} percentiles. The first line is the default one [5, 95], as reported in the experiments given in Table 5.2. Next five lines are the top five results ordered by F-measure, Precision, and Recall, respectively.

Cb_{min} , Cr_{max} , that are still computed dynamically by the internal method functions, do not change when P_{min} , P_{max} percentiles are changed since they are dependent on the previous parameters and the histogram of the image itself that never change. The same behavior will be observed when $P_{min} > P_{max}$.

It is important to note that, in the case of Compaq and Pratheepan, where $P_{min} = P_{max}$, the results worsen significantly in terms of *Precision*. On the other hand, they have a significant gain in the HGR and SFA datasets.

This particular phenomenon occurs due to the nature of the images that form each of these images databases. Both Compaq and Pratheepan contain images with more complex scenes, uncontrolled environment, distinct lighting conditions and, in general, images randomly collected from the Internet. On the other hand, HGR and SFA contain images where we have a great concentration of skin pixels – hands and arms in the case of HGR, and faces in the case of SFA occupying a considerable region of the image. Furthermore, a large part of them was collected in controlled environments, on the background and lighting conditions point of view.

Therefore, this behavior must be taken into consideration when selecting a pair of P_{min} and P_{max} to apply in the method in turn, since this can be decisive for a potential performance improvement.

With respect to the variance for all the measures, we can observe that, regardless of the combination of P_{min} and P_{max} parameters, there is no great deviation in the results in relation to the mean. For all datasets, the combinations where P_{min} and P_{max} are closer to the upper bound (95) – e.g. [80, 90], [85, 90], [85, 95], [90, 95] – are the ones with the highest variance. On the other hand, the variance had little fluctuation where $P_{min} = P_{max}$.

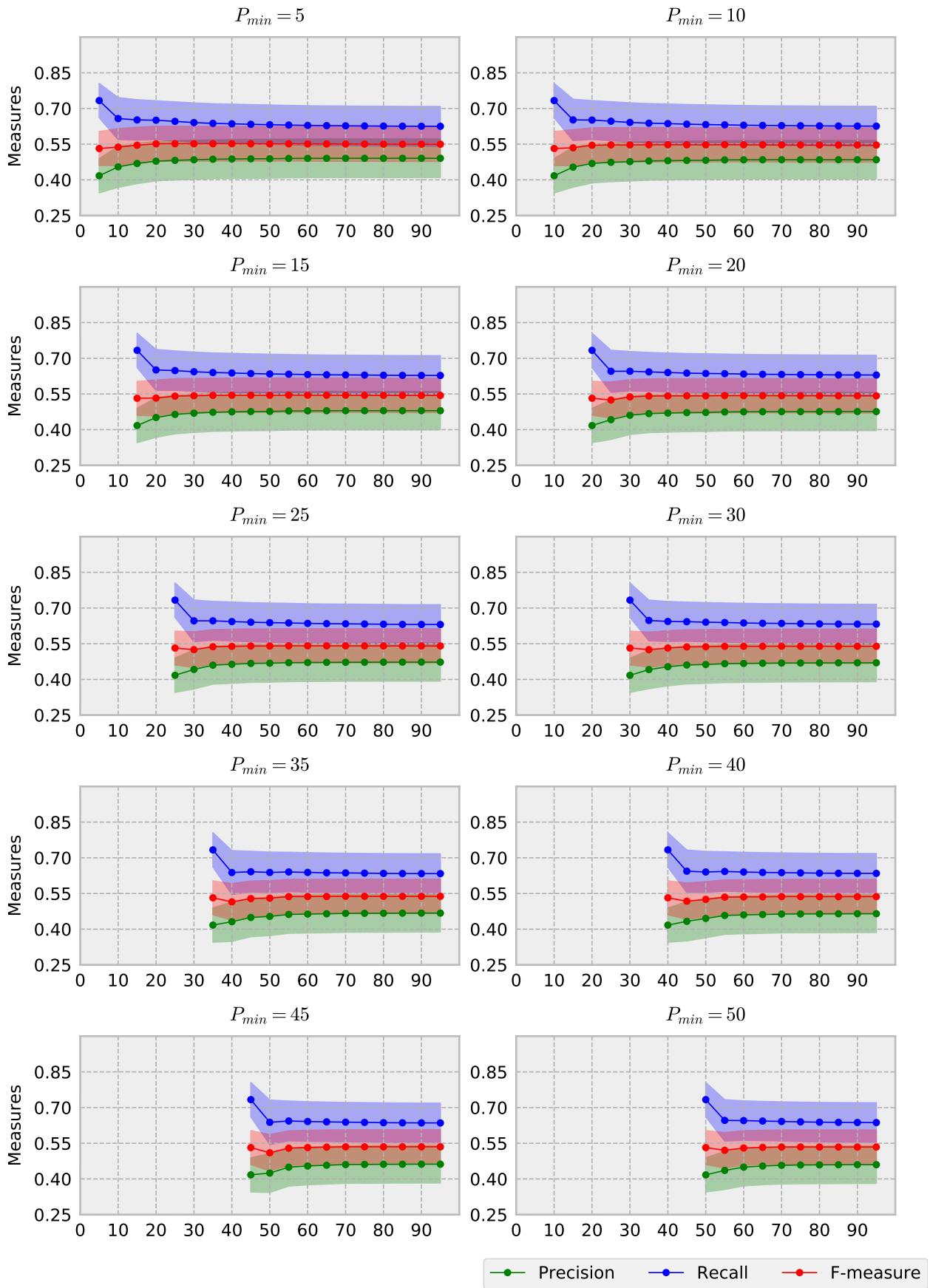


Figure 5.11: Scatter plot with the quality measures for grid search parameters in Compaq dataset.

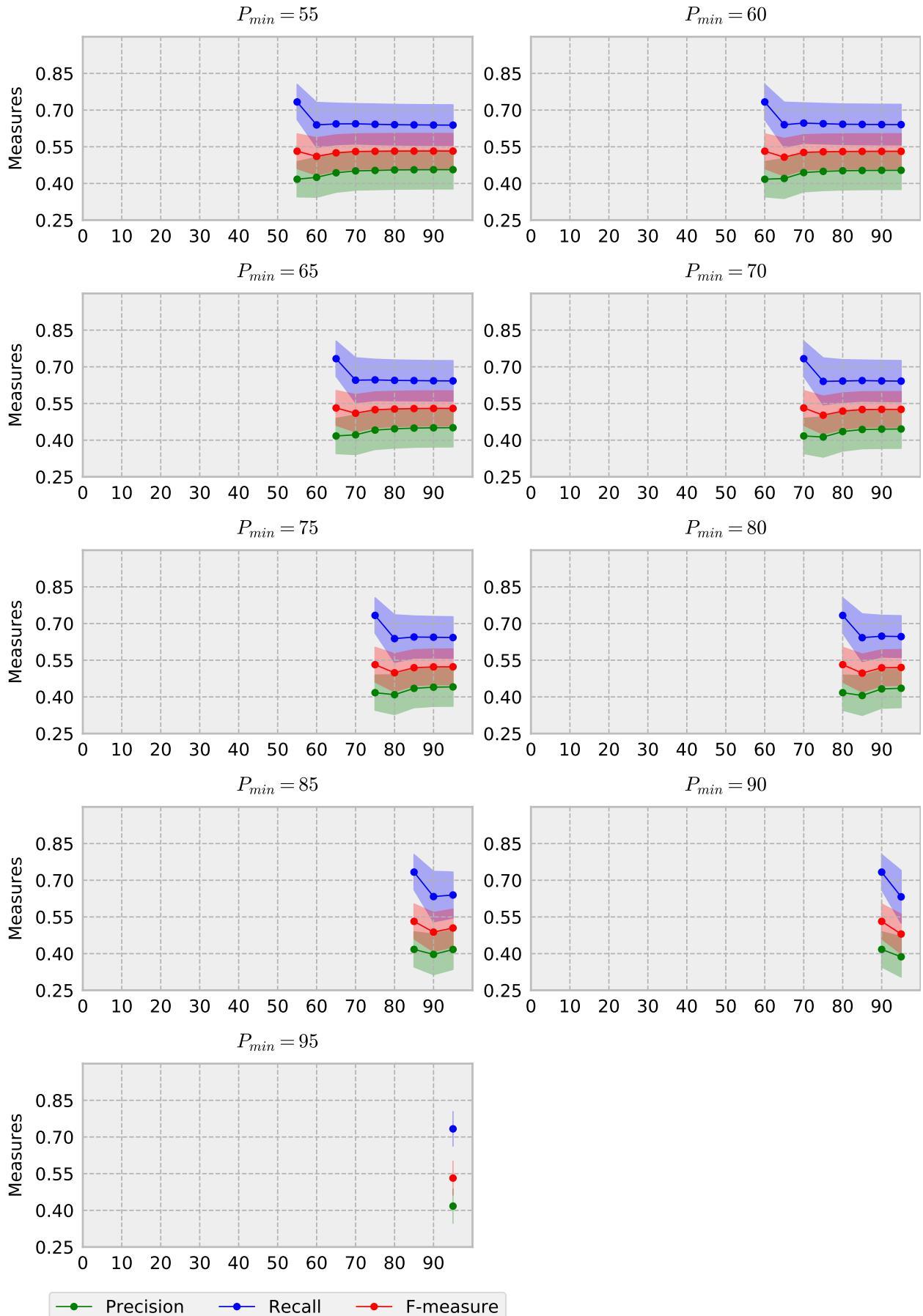


Figure 5.11: Scatter plot with the quality measures for grid search parameters in Compaq dataset (cont.).

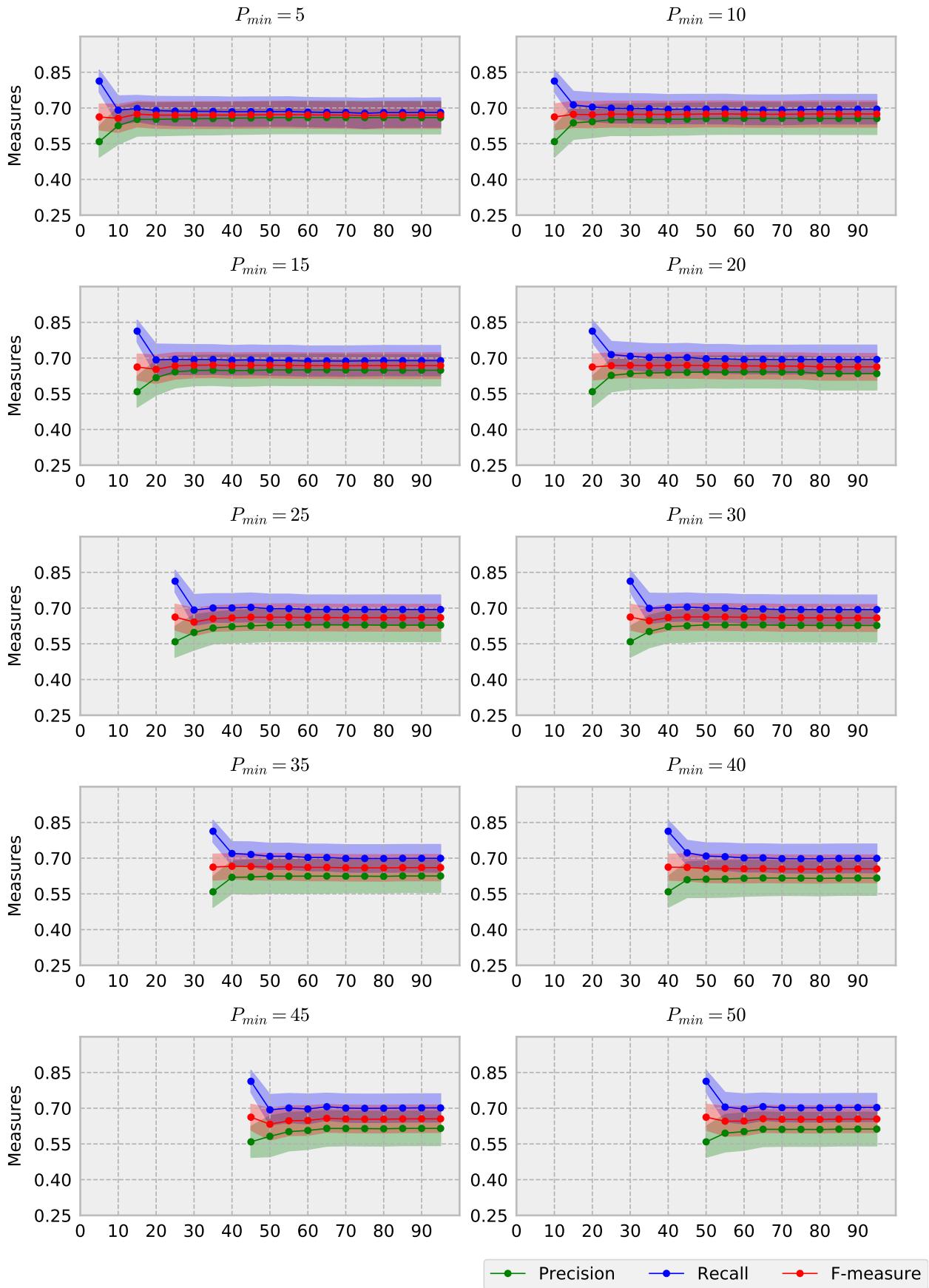


Figure 5.12: Scatter plot with the quality measures for grid search parameters in Pratheepan dataset.

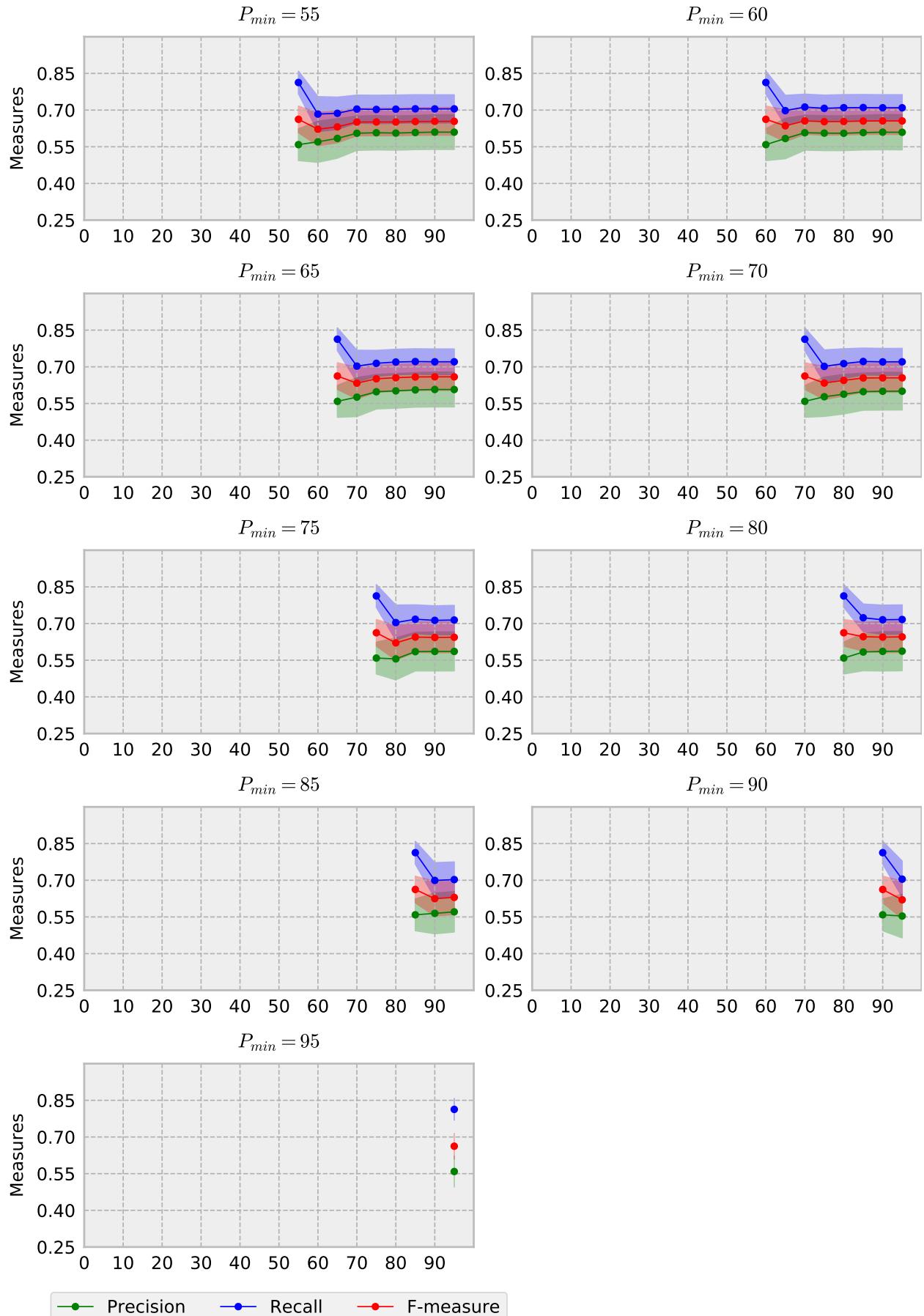


Figure 5.12: Scatter plot with the quality measures for grid search parameters in Pratheepan dataset (cont.).

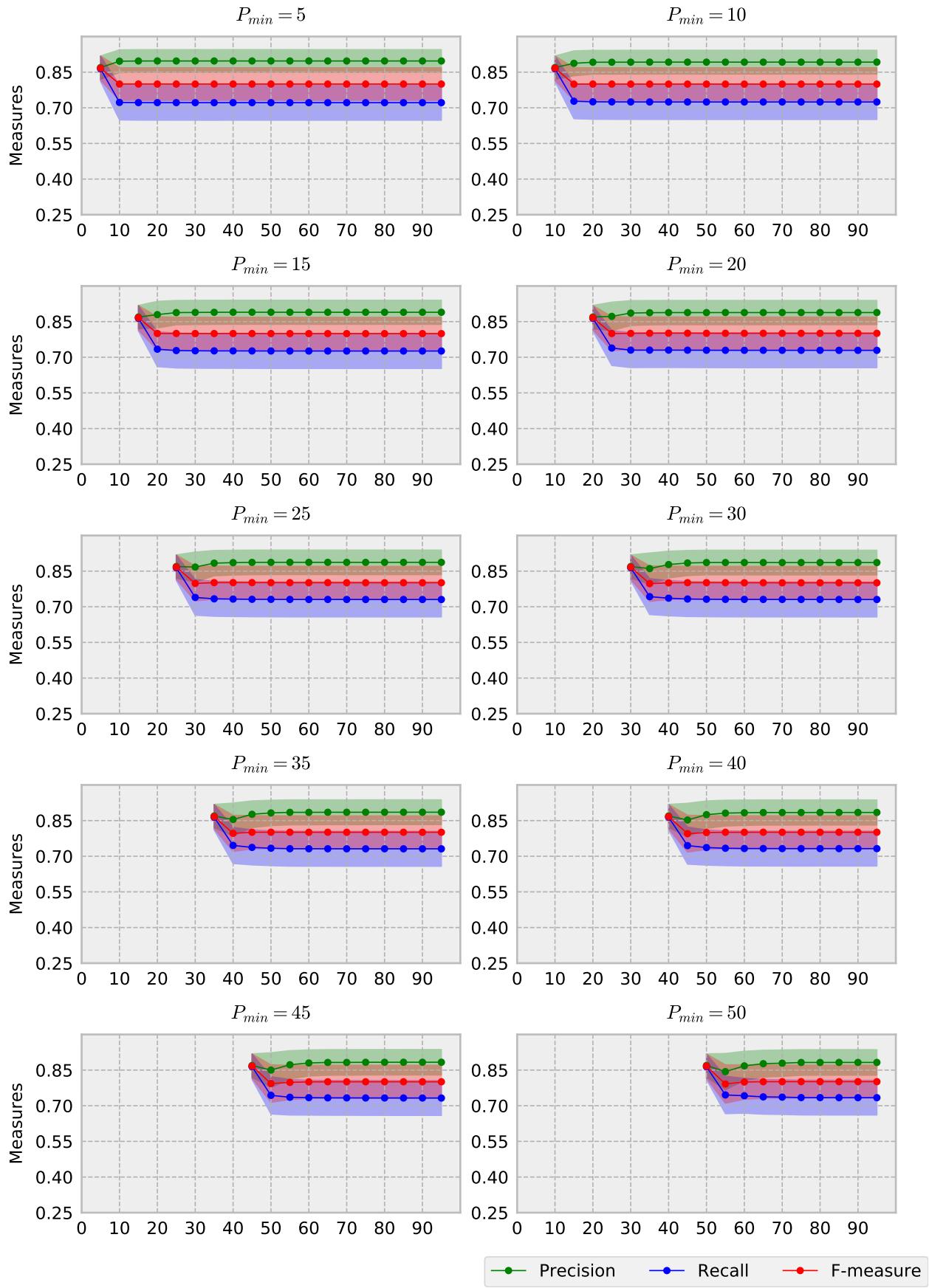


Figure 5.13: Scatter plot with the quality measures for grid search parameters in HGR dataset.

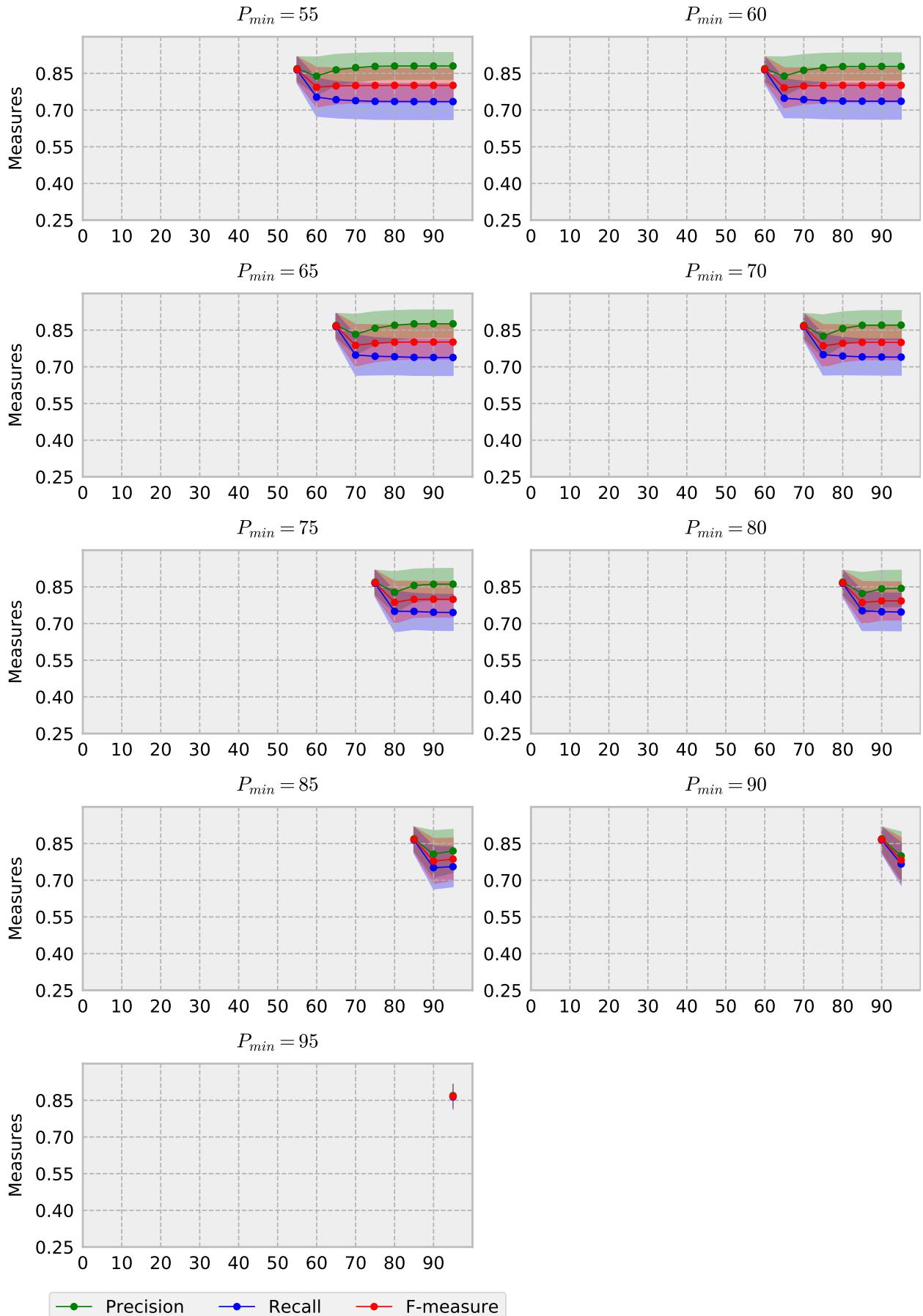


Figure 5.13: Scatter plot with the quality measures for grid search parameters in HGR dataset (cont.).

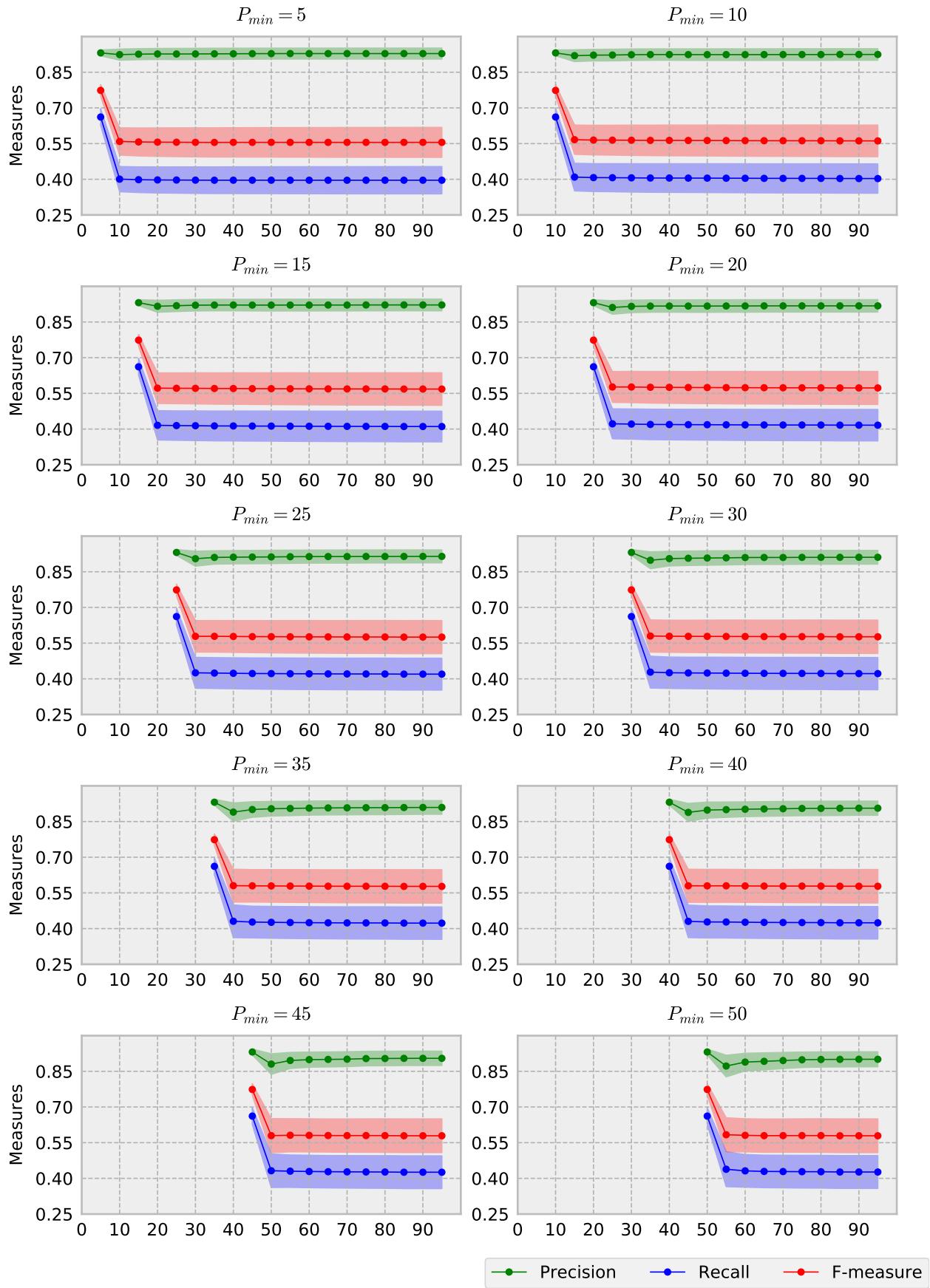


Figure 5.14: Scatter plot with the quality measures for grid search parameters in SFA dataset.

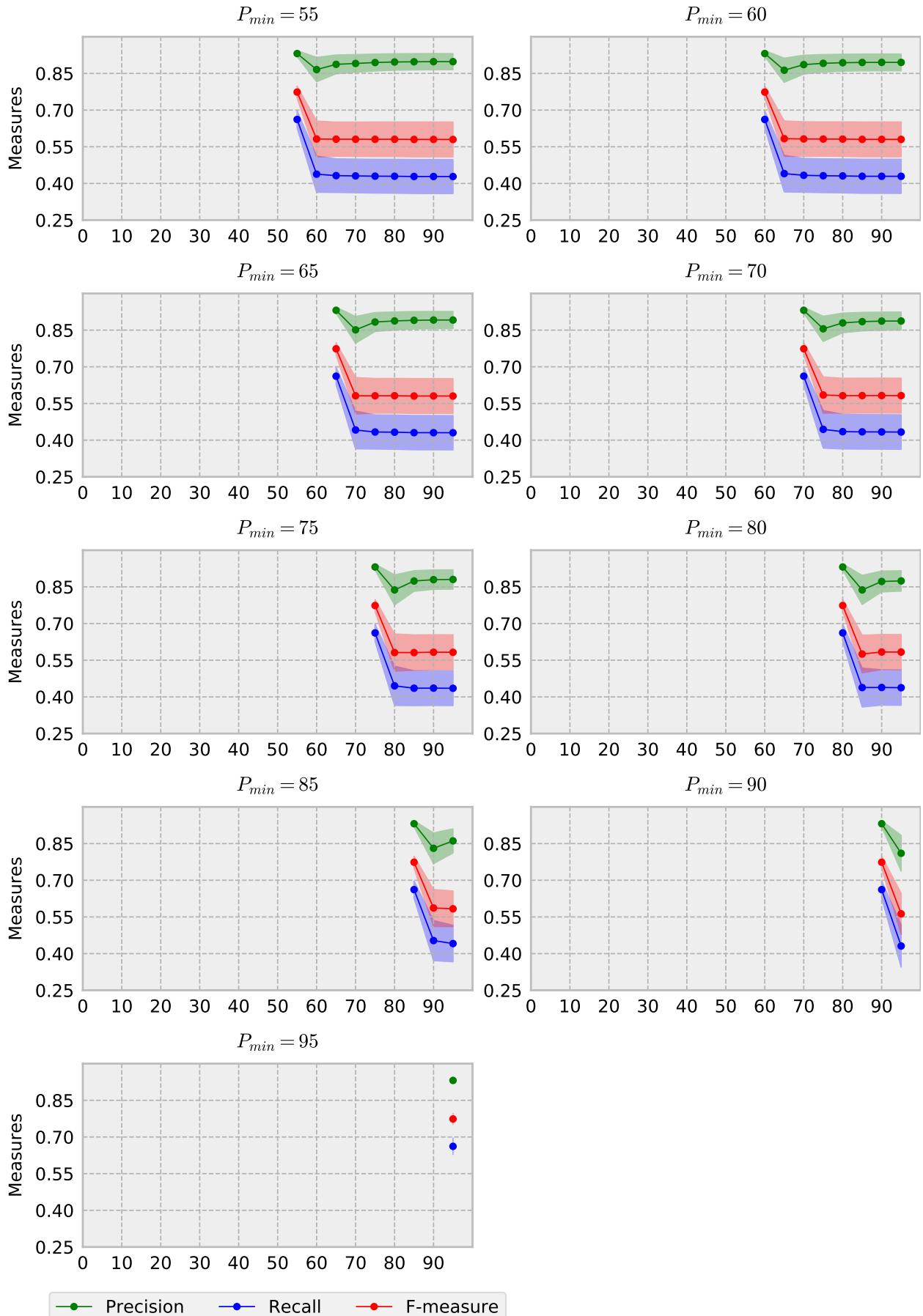


Figure 5.14: Scatter plot with the quality measures for grid search parameters in SFA dataset (cont.).

Chapter 6

Conclusions

We started this thesis by describing the problem of skin detection, explaining our motivation and determining our main objectives. Further, we reviewed the literature with a large number of works of skin detection, based on color information, comparing their differences in terms of techniques and classifiers, mainly from the point of view of performance, color models, skin color modeling, and datasets. Next, we stated the theoretical concepts that apply to this research where we briefly showed some techniques from image processing and computer vision fields used during the project, such as color space transformation, human skin segmentation, and understanding, etc. Then, we reviewed the method proposed by [Brancati *et al.* \(2017\)](#) and we added new variations of it.

Also, we showed an implementation of a grid search strategy to figure out the best combination of parameters to the method. Finally, we presented some experimental evaluations of the proposed extensions along with the original method in four widely known datasets: SFA, Pratheepan, HGR, and Compaq.

This chapter address some conclusions regarding our research, objectives, and contributions. In Section 6.1 we sum up the final considerations. Finally, in Section 6.2, we present some possible future work.

6.1 Final considerations

The method proposed by [Brancati *et al.* \(2017\)](#) is a novel rule-based skin detection method that works in the YCbCr color space based on correlation rules that evaluate the combinations of chrominance Cb, Cr values to identify human skin pixels depending on the shape and size of dynamically generated skin color clusters (trapezoids). The method is surprisingly simple and clever and it established a new tier.

Because the authors left the code available, we reproduced the original experiments and also checked if the same color patterns were presented in RGB, HSV, and Lab color spaces, or other applications as finding tree leaves, but the results were not similar as the original approach for human skin using YCbCr space.

In this research project, we introduced two extensions based on a hypothesis that the original rule could be complemented and also taking into consideration that a human skin pixel does not appear isolated. We also made a third extension that combines the original rule with the complementary one in a more strengthen method in terms of precision. All these extensions are simple and do not affect the efficiency of the original method.

We tested the extensions in four standard public datasets and the experiments showed that our methods improve the accuracy of skin detection, even when there exists a huge variation in ethnicity and illumination. Moreover, our approach proved to be very competitive,

outperforming alternative state-of-the-art work. In addition, we implemented a grid search for the parameters tuning and the supplementary neighborhood operations as well as new tools to process the datasets, binarize images, and a visualization web application for the problem of skin detection¹.

Our results confirm that skin color is an extremely powerful cue for detecting human skin in unconstrained imagery. Other local properties can be experimented to be used in a future work, along with the methods presented here, such as texture, shape, geometry, and other neighborhood operations.

Finally, part of the work has been published in an international conference (VISAPP) in earlier 2018.

6.2 Future work

In the future, extended works could explore further the connectivity of the skin pixels and, because there is so far no explanation why the original method works so well, it would be valuable to statistically analyze the shape of the trapezoids in the YCbCr space and try to correlate with the classification accuracy.

Our intuition, based on the experimental results, says that trapezoids features such as size, area, symmetry, and others, could be used to establish a relationship with the classification accuracy. Moreover, the shape of the trapezoids could be previously processed, for instance by filtering image illumination, to obtain better classification results.

The neighborhood operations add some additional computational cost but the implementation can be enhanced as well as other techniques applied, such as applying weights when counting neighboring pixels.

We have also implemented a grid search algorithm to understand the parameters selection in Section 5.5. This approach can be improved and we could try to learn the best parameters of the trapezoids by means of machine learning techniques.

With respect to machine learning techniques, a valuable extension of this work would provide a comparison of the rule-based methods presented here along with machine-learning-based ones, either to review the accuracy as well as the performance in terms of classification time.

A recent work comparing different color spaces suggests that YCgCr is also a good candidate for this problem ([Chaves-González et al., 2010](#)). In fact, YCbCr and YCgCr color spaces are very similar. Therefore, the same color relations can be presented in this space.

Another important investigation that could be addressed is related to the ranges of the chromaticity (Cr, Cb) channels used in the definition of the correlation rules. In [Brancati et al. \(2017\)](#), as well as in our proposed methods, the thresholds used were given by [Chai and Ngan \(1999\)](#). However, [Basilio et al. \(2011\)](#) found out that the thresholds were robust only against images with Caucasian people. In the end, [Basilio et al. \(2011\)](#) proposed different values that seems to be more robust to find human skin from different people races. This simple change may also result in improved human skin segmentation quality.

¹Available at <https://bitbucket.org/rodrigoadfaria/skin-detector-ws>.

Appendix A

Trapezoids Parameters Tuning Results

In this appendix we present the results of the grid search algorithm applied on each of the four datasets described in section 5.1. We only applied the combined rules method, once the trapezoids parameters definition do not change among the different methods, as described in chapter 4. Every table is sorted by *F-measure*, *Precision* and *Recall*, respectively. The results are presented in Tables A.1, A.2 A.3, and A.4 for Compaq, Pratheepan, HGR, and SFA, respectively.

Table A.1: Trapezoids parameters tuning results for Compaq dataset using combined rules.

| Pmin | Pmax | Precision | Recall | Specificity | F-measure |
|------|------|-----------|--------|-------------|-----------|
| 5 | 25 | 0.4819 | 0.6459 | 0.8762 | 0.5520 |
| 5 | 35 | 0.4866 | 0.6374 | 0.8808 | 0.5519 |
| 5 | 40 | 0.4877 | 0.6356 | 0.8818 | 0.5519 |
| 5 | 30 | 0.4844 | 0.6408 | 0.8789 | 0.5517 |
| 5 | 45 | 0.4884 | 0.6335 | 0.8827 | 0.5516 |
| 5 | 20 | 0.4784 | 0.6508 | 0.8729 | 0.5514 |
| 5 | 55 | 0.4901 | 0.6303 | 0.8843 | 0.5514 |
| 5 | 50 | 0.4887 | 0.6318 | 0.8832 | 0.5511 |
| 5 | 60 | 0.4904 | 0.6288 | 0.8848 | 0.5511 |
| 5 | 65 | 0.4906 | 0.6279 | 0.8852 | 0.5509 |
| 5 | 70 | 0.4908 | 0.6272 | 0.8854 | 0.5507 |
| 5 | 75 | 0.4907 | 0.6266 | 0.8855 | 0.5504 |
| 5 | 80 | 0.4907 | 0.6261 | 0.8856 | 0.5502 |
| 5 | 85 | 0.4907 | 0.6256 | 0.8857 | 0.5500 |
| 5 | 90 | 0.4908 | 0.6253 | 0.8858 | 0.5500 |
| 5 | 95 | 0.4908 | 0.6251 | 0.8858 | 0.5499 |
| 10 | 55 | 0.4838 | 0.6313 | 0.8808 | 0.5478 |
| 10 | 40 | 0.4806 | 0.6364 | 0.8777 | 0.5477 |
| 10 | 35 | 0.4794 | 0.6382 | 0.8768 | 0.5475 |
| 10 | 45 | 0.4817 | 0.6341 | 0.8788 | 0.5475 |
| 10 | 60 | 0.4842 | 0.6297 | 0.8813 | 0.5474 |
| 10 | 65 | 0.4844 | 0.6287 | 0.8816 | 0.5472 |
| 10 | 50 | 0.4822 | 0.6324 | 0.8794 | 0.5472 |
| 10 | 70 | 0.4846 | 0.6281 | 0.8818 | 0.5471 |
| 10 | 30 | 0.4767 | 0.6414 | 0.8746 | 0.5469 |
| 10 | 25 | 0.4738 | 0.6465 | 0.8715 | 0.5468 |
| 10 | 75 | 0.4845 | 0.6274 | 0.8819 | 0.5468 |
| 10 | 80 | 0.4845 | 0.6269 | 0.8820 | 0.5466 |
| 10 | 85 | 0.4845 | 0.6263 | 0.8821 | 0.5464 |

Continued on next page

Table A.1 – continued from previous page

| Pmin | Pmax | Precision | Recall | Specificity | F-measure |
|-------------|-------------|------------------|---------------|--------------------|------------------|
| 10 | 90 | 0.4846 | 0.6260 | 0.8822 | 0.5463 |
| 10 | 95 | 0.4846 | 0.6258 | 0.8823 | 0.5462 |
| 5 | 15 | 0.4688 | 0.6520 | 0.8683 | 0.5455 |
| 10 | 20 | 0.4689 | 0.6515 | 0.8675 | 0.5453 |
| 15 | 55 | 0.4784 | 0.6331 | 0.8773 | 0.5450 |
| 15 | 60 | 0.4789 | 0.6315 | 0.8779 | 0.5447 |
| 15 | 65 | 0.4793 | 0.6305 | 0.8784 | 0.5446 |
| 15 | 70 | 0.4795 | 0.6300 | 0.8786 | 0.5446 |
| 15 | 45 | 0.4761 | 0.6358 | 0.8752 | 0.5445 |
| 15 | 40 | 0.4747 | 0.6382 | 0.8740 | 0.5444 |
| 15 | 75 | 0.4794 | 0.6293 | 0.8787 | 0.5442 |
| 15 | 50 | 0.4766 | 0.6339 | 0.8759 | 0.5441 |
| 15 | 80 | 0.4796 | 0.6287 | 0.8789 | 0.5441 |
| 15 | 35 | 0.4730 | 0.6401 | 0.8728 | 0.5440 |
| 15 | 85 | 0.4795 | 0.6281 | 0.8790 | 0.5439 |
| 15 | 90 | 0.4797 | 0.6279 | 0.8792 | 0.5438 |
| 15 | 95 | 0.4796 | 0.6277 | 0.8792 | 0.5437 |
| 20 | 55 | 0.4742 | 0.6352 | 0.8747 | 0.5430 |
| 20 | 60 | 0.4748 | 0.6336 | 0.8755 | 0.5429 |
| 20 | 65 | 0.4753 | 0.6326 | 0.8759 | 0.5428 |
| 15 | 30 | 0.4692 | 0.6434 | 0.8701 | 0.5427 |
| 20 | 70 | 0.4755 | 0.6320 | 0.8761 | 0.5427 |
| 20 | 75 | 0.4755 | 0.6312 | 0.8764 | 0.5424 |
| 20 | 80 | 0.4756 | 0.6306 | 0.8766 | 0.5423 |
| 20 | 45 | 0.4714 | 0.6375 | 0.8725 | 0.5420 |
| 20 | 90 | 0.4758 | 0.6297 | 0.8769 | 0.5420 |
| 20 | 85 | 0.4757 | 0.6299 | 0.8768 | 0.5420 |
| 20 | 50 | 0.4722 | 0.6360 | 0.8732 | 0.5420 |
| 20 | 95 | 0.4757 | 0.6296 | 0.8769 | 0.5419 |
| 20 | 40 | 0.4696 | 0.6401 | 0.8710 | 0.5418 |
| 20 | 35 | 0.4674 | 0.6428 | 0.8690 | 0.5412 |
| 25 | 55 | 0.4702 | 0.6374 | 0.8719 | 0.5412 |
| 25 | 70 | 0.4720 | 0.6337 | 0.8736 | 0.5410 |
| 25 | 65 | 0.4716 | 0.6343 | 0.8734 | 0.5410 |
| 25 | 60 | 0.4709 | 0.6355 | 0.8728 | 0.5410 |
| 25 | 75 | 0.4722 | 0.6328 | 0.8740 | 0.5408 |
| 15 | 25 | 0.4639 | 0.6482 | 0.8647 | 0.5408 |
| 25 | 80 | 0.4724 | 0.6319 | 0.8742 | 0.5407 |
| 25 | 85 | 0.4726 | 0.6312 | 0.8744 | 0.5405 |
| 25 | 90 | 0.4727 | 0.6310 | 0.8746 | 0.5405 |
| 25 | 95 | 0.4726 | 0.6309 | 0.8746 | 0.5404 |
| 25 | 45 | 0.4672 | 0.6403 | 0.8695 | 0.5402 |
| 25 | 50 | 0.4681 | 0.6385 | 0.8703 | 0.5402 |
| 30 | 70 | 0.4687 | 0.6353 | 0.8714 | 0.5395 |
| 30 | 75 | 0.4690 | 0.6343 | 0.8719 | 0.5393 |
| 30 | 80 | 0.4694 | 0.6333 | 0.8722 | 0.5391 |
| 30 | 90 | 0.4698 | 0.6325 | 0.8726 | 0.5391 |
| 30 | 65 | 0.4679 | 0.6359 | 0.8711 | 0.5391 |
| 30 | 95 | 0.4698 | 0.6323 | 0.8726 | 0.5391 |
| 30 | 85 | 0.4695 | 0.6327 | 0.8724 | 0.5390 |
| 30 | 60 | 0.4670 | 0.6373 | 0.8704 | 0.5390 |
| 30 | 55 | 0.4659 | 0.6390 | 0.8693 | 0.5389 |
| 25 | 40 | 0.4635 | 0.6431 | 0.8662 | 0.5387 |
| 35 | 70 | 0.4663 | 0.6365 | 0.8697 | 0.5383 |
| 35 | 75 | 0.4667 | 0.6355 | 0.8702 | 0.5382 |

Continued on next page

Table A.1 – continued from previous page

| Pmin | Pmax | Precision | Recall | Specificity | F-measure |
|------|------|-----------|--------|-------------|-----------|
| 35 | 90 | 0.4675 | 0.6337 | 0.8709 | 0.5380 |
| 35 | 95 | 0.4675 | 0.6336 | 0.8709 | 0.5380 |
| 35 | 85 | 0.4672 | 0.6339 | 0.8707 | 0.5379 |
| 35 | 80 | 0.4669 | 0.6344 | 0.8705 | 0.5379 |
| 5 | 10 | 0.4547 | 0.6577 | 0.8579 | 0.5377 |
| 20 | 30 | 0.4605 | 0.6455 | 0.8644 | 0.5375 |
| 25 | 35 | 0.4598 | 0.6463 | 0.8630 | 0.5373 |
| 35 | 65 | 0.4646 | 0.6370 | 0.8689 | 0.5373 |
| 30 | 50 | 0.4627 | 0.6404 | 0.8668 | 0.5372 |
| 35 | 60 | 0.4635 | 0.6385 | 0.8680 | 0.5371 |
| 40 | 70 | 0.4637 | 0.6376 | 0.8681 | 0.5369 |
| 40 | 75 | 0.4641 | 0.6366 | 0.8686 | 0.5368 |
| 35 | 55 | 0.4620 | 0.6405 | 0.8665 | 0.5368 |
| 40 | 90 | 0.4650 | 0.6347 | 0.8695 | 0.5367 |
| 40 | 95 | 0.4650 | 0.6346 | 0.8695 | 0.5367 |
| 40 | 85 | 0.4647 | 0.6349 | 0.8693 | 0.5367 |
| 40 | 80 | 0.4644 | 0.6355 | 0.8691 | 0.5366 |
| 30 | 45 | 0.4605 | 0.6426 | 0.8653 | 0.5365 |
| 40 | 65 | 0.4616 | 0.6385 | 0.8671 | 0.5358 |
| 40 | 60 | 0.4600 | 0.6402 | 0.8659 | 0.5354 |
| 45 | 75 | 0.4609 | 0.6378 | 0.8663 | 0.5351 |
| 45 | 90 | 0.4620 | 0.6356 | 0.8675 | 0.5351 |
| 45 | 80 | 0.4615 | 0.6365 | 0.8670 | 0.5350 |
| 45 | 95 | 0.4620 | 0.6354 | 0.8676 | 0.5350 |
| 45 | 85 | 0.4617 | 0.6359 | 0.8673 | 0.5350 |
| 45 | 70 | 0.4602 | 0.6388 | 0.8656 | 0.5350 |
| 10 | 15 | 0.4534 | 0.6521 | 0.8575 | 0.5349 |
| 40 | 55 | 0.4576 | 0.6426 | 0.8639 | 0.5346 |
| 50 | 75 | 0.4585 | 0.6400 | 0.8646 | 0.5343 |
| 50 | 85 | 0.4596 | 0.6379 | 0.8657 | 0.5342 |
| 50 | 80 | 0.4592 | 0.6386 | 0.8653 | 0.5342 |
| 50 | 90 | 0.4598 | 0.6373 | 0.8660 | 0.5342 |
| 50 | 95 | 0.4599 | 0.6371 | 0.8661 | 0.5342 |
| 50 | 70 | 0.4574 | 0.6417 | 0.8634 | 0.5341 |
| 45 | 65 | 0.4576 | 0.6401 | 0.8641 | 0.5337 |
| 15 | 20 | 0.4509 | 0.6509 | 0.8541 | 0.5327 |
| 50 | 65 | 0.4543 | 0.6434 | 0.8614 | 0.5326 |
| 55 | 90 | 0.4562 | 0.6390 | 0.8635 | 0.5324 |
| 55 | 85 | 0.4559 | 0.6396 | 0.8631 | 0.5323 |
| 55 | 95 | 0.4563 | 0.6387 | 0.8636 | 0.5323 |
| 55 | 80 | 0.4552 | 0.6403 | 0.8627 | 0.5321 |
| 45 | 60 | 0.4544 | 0.6414 | 0.8622 | 0.5320 |
| 30 | 40 | 0.4533 | 0.6436 | 0.8604 | 0.5319 |
| 5 | 5 | 0.4172 | 0.7333 | 0.8096 | 0.5318 |
| 10 | 10 | 0.4172 | 0.7333 | 0.8096 | 0.5318 |
| 15 | 15 | 0.4172 | 0.7333 | 0.8096 | 0.5318 |
| 20 | 20 | 0.4172 | 0.7333 | 0.8096 | 0.5318 |
| 25 | 25 | 0.4172 | 0.7333 | 0.8096 | 0.5318 |
| 30 | 30 | 0.4172 | 0.7333 | 0.8096 | 0.5318 |
| 35 | 35 | 0.4172 | 0.7333 | 0.8096 | 0.5318 |
| 40 | 40 | 0.4172 | 0.7333 | 0.8096 | 0.5318 |
| 45 | 45 | 0.4172 | 0.7333 | 0.8096 | 0.5318 |
| 50 | 50 | 0.4172 | 0.7333 | 0.8096 | 0.5318 |
| 55 | 55 | 0.4172 | 0.7333 | 0.8096 | 0.5318 |
| 60 | 60 | 0.4172 | 0.7333 | 0.8096 | 0.5318 |

Continued on next page

Table A.1 – continued from previous page

| Pmin | Pmax | Precision | Recall | Specificity | F-measure |
|-------------|-------------|------------------|---------------|--------------------|------------------|
| 65 | 65 | 0.4172 | 0.7333 | 0.8096 | 0.5318 |
| 70 | 70 | 0.4172 | 0.7333 | 0.8096 | 0.5318 |
| 75 | 75 | 0.4172 | 0.7333 | 0.8096 | 0.5318 |
| 80 | 80 | 0.4172 | 0.7333 | 0.8096 | 0.5318 |
| 85 | 85 | 0.4172 | 0.7333 | 0.8096 | 0.5318 |
| 90 | 90 | 0.4172 | 0.7333 | 0.8096 | 0.5318 |
| 95 | 95 | 0.4172 | 0.7333 | 0.8096 | 0.5318 |
| 55 | 75 | 0.4532 | 0.6420 | 0.8608 | 0.5313 |
| 60 | 90 | 0.4536 | 0.6409 | 0.8611 | 0.5312 |
| 60 | 95 | 0.4537 | 0.6406 | 0.8612 | 0.5312 |
| 60 | 85 | 0.4530 | 0.6415 | 0.8606 | 0.5310 |
| 55 | 70 | 0.4513 | 0.6440 | 0.8592 | 0.5307 |
| 60 | 80 | 0.4519 | 0.6422 | 0.8597 | 0.5305 |
| 35 | 50 | 0.4536 | 0.6385 | 0.8616 | 0.5304 |
| 50 | 60 | 0.4496 | 0.6453 | 0.8582 | 0.5299 |
| 65 | 95 | 0.4509 | 0.6424 | 0.8593 | 0.5299 |
| 65 | 90 | 0.4506 | 0.6428 | 0.8590 | 0.5298 |
| 60 | 75 | 0.4494 | 0.6445 | 0.8572 | 0.5295 |
| 65 | 85 | 0.4494 | 0.6437 | 0.8582 | 0.5293 |
| 45 | 55 | 0.4494 | 0.6436 | 0.8582 | 0.5293 |
| 35 | 45 | 0.4492 | 0.6414 | 0.8575 | 0.5284 |
| 65 | 80 | 0.4463 | 0.6446 | 0.8550 | 0.5274 |
| 60 | 70 | 0.4444 | 0.6469 | 0.8528 | 0.5269 |
| 70 | 95 | 0.4457 | 0.6415 | 0.8567 | 0.5260 |
| 70 | 90 | 0.4451 | 0.6423 | 0.8560 | 0.5258 |
| 70 | 85 | 0.4437 | 0.6437 | 0.8546 | 0.5253 |
| 55 | 65 | 0.4436 | 0.6435 | 0.8533 | 0.5252 |
| 40 | 50 | 0.4450 | 0.6403 | 0.8554 | 0.5251 |
| 30 | 35 | 0.4413 | 0.6477 | 0.8494 | 0.5249 |
| 25 | 30 | 0.4419 | 0.6462 | 0.8490 | 0.5248 |
| 20 | 25 | 0.4418 | 0.6453 | 0.8515 | 0.5245 |
| 65 | 75 | 0.4412 | 0.6465 | 0.8499 | 0.5245 |
| 75 | 95 | 0.4409 | 0.6427 | 0.8528 | 0.5230 |
| 75 | 90 | 0.4398 | 0.6440 | 0.8519 | 0.5226 |
| 50 | 55 | 0.4357 | 0.6461 | 0.8459 | 0.5205 |
| 80 | 95 | 0.4352 | 0.6467 | 0.8451 | 0.5203 |
| 75 | 85 | 0.4348 | 0.6449 | 0.8486 | 0.5194 |
| 80 | 90 | 0.4327 | 0.6483 | 0.8433 | 0.5190 |
| 70 | 80 | 0.4352 | 0.6418 | 0.8486 | 0.5187 |
| 40 | 45 | 0.4329 | 0.6437 | 0.8428 | 0.5176 |
| 35 | 40 | 0.4313 | 0.6381 | 0.8476 | 0.5147 |
| 55 | 60 | 0.4250 | 0.6395 | 0.8417 | 0.5107 |
| 65 | 70 | 0.4219 | 0.6453 | 0.8335 | 0.5102 |
| 45 | 50 | 0.4247 | 0.6381 | 0.8413 | 0.5100 |
| 60 | 65 | 0.4203 | 0.6402 | 0.8354 | 0.5075 |
| 85 | 95 | 0.4168 | 0.6397 | 0.8335 | 0.5048 |
| 70 | 75 | 0.4129 | 0.6405 | 0.8269 | 0.5021 |
| 75 | 80 | 0.4091 | 0.6383 | 0.8289 | 0.4986 |
| 80 | 85 | 0.4056 | 0.6423 | 0.8215 | 0.4972 |
| 85 | 90 | 0.3967 | 0.6332 | 0.8199 | 0.4878 |
| 90 | 95 | 0.3869 | 0.6323 | 0.8080 | 0.4801 |

Table A.2: Trapezoids parameters tuning results for Pratheepan dataset using combined rules.

| Pmin | Pmax | Precision | Recall | Specificity | F-measure |
|------|------|-----------|--------|-------------|-----------|
| 10 | 55 | 0.6560 | 0.6955 | 0.9055 | 0.6752 |
| 10 | 80 | 0.6558 | 0.6950 | 0.9056 | 0.6748 |
| 10 | 85 | 0.6557 | 0.6949 | 0.9056 | 0.6748 |
| 10 | 95 | 0.6557 | 0.6950 | 0.9057 | 0.6747 |
| 10 | 90 | 0.6556 | 0.6950 | 0.9056 | 0.6747 |
| 10 | 50 | 0.6551 | 0.6953 | 0.9052 | 0.6746 |
| 10 | 75 | 0.6561 | 0.6935 | 0.9057 | 0.6743 |
| 5 | 15 | 0.6521 | 0.6978 | 0.9045 | 0.6741 |
| 10 | 25 | 0.6509 | 0.6991 | 0.9057 | 0.6741 |
| 10 | 60 | 0.6563 | 0.6928 | 0.9057 | 0.6740 |
| 10 | 30 | 0.6508 | 0.6983 | 0.9045 | 0.6737 |
| 10 | 65 | 0.6563 | 0.6920 | 0.9057 | 0.6737 |
| 10 | 70 | 0.6562 | 0.6920 | 0.9057 | 0.6736 |
| 10 | 45 | 0.6530 | 0.6955 | 0.9048 | 0.6736 |
| 10 | 35 | 0.6510 | 0.6972 | 0.9044 | 0.6733 |
| 10 | 15 | 0.6376 | 0.7133 | 0.8916 | 0.6733 |
| 10 | 40 | 0.6525 | 0.6945 | 0.9048 | 0.6728 |
| 5 | 55 | 0.6595 | 0.6852 | 0.9085 | 0.6721 |
| 10 | 20 | 0.6425 | 0.7039 | 0.8985 | 0.6718 |
| 5 | 50 | 0.6591 | 0.6850 | 0.9084 | 0.6718 |
| 5 | 60 | 0.6599 | 0.6827 | 0.9087 | 0.6711 |
| 5 | 45 | 0.6577 | 0.6848 | 0.9083 | 0.6710 |
| 15 | 35 | 0.6487 | 0.6939 | 0.9040 | 0.6706 |
| 5 | 65 | 0.6600 | 0.6813 | 0.9087 | 0.6705 |
| 5 | 40 | 0.6570 | 0.6842 | 0.9084 | 0.6704 |
| 5 | 70 | 0.6599 | 0.6811 | 0.9087 | 0.6703 |
| 5 | 30 | 0.6550 | 0.6861 | 0.9080 | 0.6702 |
| 5 | 35 | 0.6553 | 0.6858 | 0.9080 | 0.6702 |
| 5 | 20 | 0.6522 | 0.6891 | 0.9068 | 0.6701 |
| 5 | 95 | 0.6596 | 0.6809 | 0.9088 | 0.6701 |
| 5 | 90 | 0.6595 | 0.6810 | 0.9088 | 0.6701 |
| 5 | 80 | 0.6595 | 0.6806 | 0.9088 | 0.6699 |
| 5 | 25 | 0.6538 | 0.6868 | 0.9076 | 0.6699 |
| 5 | 85 | 0.6595 | 0.6805 | 0.9088 | 0.6698 |
| 15 | 50 | 0.6503 | 0.6904 | 0.9028 | 0.6698 |
| 15 | 30 | 0.6475 | 0.6935 | 0.9040 | 0.6697 |
| 20 | 45 | 0.6397 | 0.7024 | 0.8915 | 0.6696 |
| 15 | 45 | 0.6478 | 0.6924 | 0.9021 | 0.6694 |
| 20 | 30 | 0.6347 | 0.7077 | 0.8934 | 0.6692 |
| 15 | 55 | 0.6494 | 0.6900 | 0.9027 | 0.6691 |
| 15 | 65 | 0.6503 | 0.6879 | 0.9030 | 0.6686 |
| 20 | 40 | 0.6389 | 0.7012 | 0.8913 | 0.6686 |
| 15 | 80 | 0.6489 | 0.6893 | 0.9028 | 0.6685 |
| 5 | 75 | 0.6595 | 0.6776 | 0.9088 | 0.6685 |
| 15 | 40 | 0.6474 | 0.6909 | 0.9021 | 0.6684 |
| 15 | 85 | 0.6488 | 0.6891 | 0.9028 | 0.6683 |
| 20 | 50 | 0.6418 | 0.6971 | 0.8925 | 0.6683 |
| 15 | 75 | 0.6493 | 0.6884 | 0.9029 | 0.6683 |
| 15 | 95 | 0.6487 | 0.6891 | 0.9028 | 0.6683 |
| 15 | 90 | 0.6487 | 0.6891 | 0.9028 | 0.6683 |
| 15 | 60 | 0.6503 | 0.6873 | 0.9030 | 0.6683 |
| 20 | 35 | 0.6373 | 0.7022 | 0.8938 | 0.6682 |
| 20 | 25 | 0.6271 | 0.7144 | 0.8855 | 0.6679 |

Continued on next page

Table A.2 – continued from previous page

| Pmin | Pmax | Precision | Recall | Specificity | F-measure |
|-------------|-------------|------------------|---------------|--------------------|------------------|
| 15 | 70 | 0.6494 | 0.6872 | 0.9029 | 0.6678 |
| 20 | 55 | 0.6409 | 0.6967 | 0.8923 | 0.6676 |
| 20 | 65 | 0.6420 | 0.6944 | 0.8927 | 0.6672 |
| 15 | 25 | 0.6420 | 0.6943 | 0.8992 | 0.6672 |
| 20 | 60 | 0.6419 | 0.6937 | 0.8926 | 0.6668 |
| 20 | 75 | 0.6411 | 0.6938 | 0.8927 | 0.6664 |
| 20 | 70 | 0.6412 | 0.6933 | 0.8927 | 0.6662 |
| 35 | 40 | 0.6196 | 0.7195 | 0.8793 | 0.6658 |
| 35 | 45 | 0.6208 | 0.7155 | 0.8746 | 0.6648 |
| 35 | 50 | 0.6246 | 0.7077 | 0.8776 | 0.6635 |
| 35 | 55 | 0.6242 | 0.7078 | 0.8774 | 0.6634 |
| 20 | 80 | 0.6349 | 0.6941 | 0.8874 | 0.6632 |
| 30 | 50 | 0.6295 | 0.7004 | 0.8832 | 0.6631 |
| 20 | 90 | 0.6348 | 0.6939 | 0.8875 | 0.6630 |
| 20 | 95 | 0.6348 | 0.6938 | 0.8875 | 0.6630 |
| 20 | 85 | 0.6348 | 0.6938 | 0.8875 | 0.6630 |
| 30 | 55 | 0.6293 | 0.7003 | 0.8828 | 0.6629 |
| 30 | 45 | 0.6256 | 0.7045 | 0.8815 | 0.6627 |
| 5 | 5 | 0.5587 | 0.8129 | 0.8327 | 0.6623 |
| 10 | 10 | 0.5587 | 0.8129 | 0.8327 | 0.6623 |
| 15 | 15 | 0.5587 | 0.8129 | 0.8327 | 0.6623 |
| 20 | 20 | 0.5587 | 0.8129 | 0.8327 | 0.6623 |
| 25 | 25 | 0.5587 | 0.8129 | 0.8327 | 0.6623 |
| 30 | 30 | 0.5587 | 0.8129 | 0.8327 | 0.6623 |
| 35 | 35 | 0.5587 | 0.8129 | 0.8327 | 0.6623 |
| 40 | 40 | 0.5587 | 0.8129 | 0.8327 | 0.6623 |
| 45 | 45 | 0.5587 | 0.8129 | 0.8327 | 0.6623 |
| 50 | 50 | 0.5587 | 0.8129 | 0.8327 | 0.6623 |
| 55 | 55 | 0.5587 | 0.8129 | 0.8327 | 0.6623 |
| 60 | 60 | 0.5587 | 0.8129 | 0.8327 | 0.6623 |
| 65 | 65 | 0.5587 | 0.8129 | 0.8327 | 0.6623 |
| 70 | 70 | 0.5587 | 0.8129 | 0.8327 | 0.6623 |
| 75 | 75 | 0.5587 | 0.8129 | 0.8327 | 0.6623 |
| 80 | 80 | 0.5587 | 0.8129 | 0.8327 | 0.6623 |
| 85 | 85 | 0.5587 | 0.8129 | 0.8327 | 0.6623 |
| 90 | 90 | 0.5587 | 0.8129 | 0.8327 | 0.6623 |
| 95 | 95 | 0.5587 | 0.8129 | 0.8327 | 0.6623 |
| 25 | 45 | 0.6254 | 0.7029 | 0.8815 | 0.6619 |
| 25 | 55 | 0.6290 | 0.6979 | 0.8833 | 0.6617 |
| 25 | 50 | 0.6286 | 0.6977 | 0.8831 | 0.6614 |
| 35 | 65 | 0.6247 | 0.7026 | 0.8780 | 0.6614 |
| 30 | 65 | 0.6295 | 0.6965 | 0.8832 | 0.6613 |
| 30 | 60 | 0.6295 | 0.6966 | 0.8836 | 0.6613 |
| 35 | 60 | 0.6244 | 0.7029 | 0.8780 | 0.6613 |
| 40 | 45 | 0.6094 | 0.7224 | 0.8662 | 0.6611 |
| 25 | 65 | 0.6298 | 0.6947 | 0.8840 | 0.6607 |
| 25 | 60 | 0.6296 | 0.6944 | 0.8840 | 0.6604 |
| 35 | 85 | 0.6252 | 0.6994 | 0.8808 | 0.6602 |
| 35 | 95 | 0.6251 | 0.6995 | 0.8807 | 0.6602 |
| 35 | 90 | 0.6251 | 0.6995 | 0.8808 | 0.6602 |
| 25 | 75 | 0.6291 | 0.6940 | 0.8842 | 0.6600 |
| 25 | 70 | 0.6291 | 0.6938 | 0.8841 | 0.6599 |
| 30 | 40 | 0.6219 | 0.7027 | 0.8804 | 0.6598 |
| 25 | 80 | 0.6284 | 0.6942 | 0.8837 | 0.6597 |
| 25 | 90 | 0.6284 | 0.6942 | 0.8836 | 0.6597 |

Continued on next page

Table A.2 – continued from previous page

| Pmin | Pmax | Precision | Recall | Specificity | F-measure |
|------|------|-----------|--------|-------------|-----------|
| 25 | 95 | 0.6284 | 0.6942 | 0.8836 | 0.6597 |
| 35 | 70 | 0.6244 | 0.6991 | 0.8802 | 0.6596 |
| 25 | 85 | 0.6284 | 0.6941 | 0.8837 | 0.6596 |
| 35 | 75 | 0.6243 | 0.6987 | 0.8802 | 0.6594 |
| 30 | 70 | 0.6277 | 0.6939 | 0.8832 | 0.6592 |
| 30 | 75 | 0.6277 | 0.6939 | 0.8832 | 0.6592 |
| 25 | 40 | 0.6222 | 0.7006 | 0.8803 | 0.6591 |
| 65 | 90 | 0.6073 | 0.7206 | 0.8680 | 0.6591 |
| 65 | 95 | 0.6071 | 0.7206 | 0.8680 | 0.6590 |
| 35 | 80 | 0.6235 | 0.6987 | 0.8802 | 0.6590 |
| 30 | 80 | 0.6272 | 0.6938 | 0.8832 | 0.6589 |
| 30 | 95 | 0.6271 | 0.6937 | 0.8831 | 0.6587 |
| 30 | 85 | 0.6272 | 0.6936 | 0.8832 | 0.6587 |
| 30 | 90 | 0.6271 | 0.6937 | 0.8832 | 0.6587 |
| 65 | 85 | 0.6055 | 0.7215 | 0.8676 | 0.6584 |
| 45 | 65 | 0.6151 | 0.7061 | 0.8718 | 0.6575 |
| 5 | 10 | 0.6257 | 0.6907 | 0.8921 | 0.6566 |
| 40 | 50 | 0.6117 | 0.7087 | 0.8709 | 0.6566 |
| 40 | 65 | 0.6167 | 0.7011 | 0.8739 | 0.6562 |
| 40 | 55 | 0.6126 | 0.7063 | 0.8714 | 0.6561 |
| 60 | 90 | 0.6097 | 0.7095 | 0.8699 | 0.6558 |
| 60 | 70 | 0.6074 | 0.7125 | 0.8707 | 0.6558 |
| 25 | 35 | 0.6163 | 0.7004 | 0.8792 | 0.6557 |
| 50 | 65 | 0.6117 | 0.7064 | 0.8712 | 0.6556 |
| 60 | 95 | 0.6093 | 0.7094 | 0.8697 | 0.6556 |
| 45 | 95 | 0.6153 | 0.7012 | 0.8750 | 0.6554 |
| 65 | 80 | 0.6017 | 0.7197 | 0.8661 | 0.6554 |
| 45 | 90 | 0.6153 | 0.7012 | 0.8750 | 0.6554 |
| 40 | 60 | 0.6154 | 0.7009 | 0.8742 | 0.6554 |
| 60 | 85 | 0.6079 | 0.7104 | 0.8697 | 0.6552 |
| 45 | 85 | 0.6151 | 0.7007 | 0.8750 | 0.6551 |
| 40 | 95 | 0.6163 | 0.6991 | 0.8762 | 0.6551 |
| 40 | 90 | 0.6163 | 0.6991 | 0.8762 | 0.6551 |
| 40 | 85 | 0.6163 | 0.6989 | 0.8763 | 0.6550 |
| 70 | 95 | 0.6004 | 0.7204 | 0.8650 | 0.6549 |
| 45 | 70 | 0.6149 | 0.7005 | 0.8749 | 0.6549 |
| 70 | 90 | 0.6001 | 0.7202 | 0.8647 | 0.6547 |
| 40 | 70 | 0.6163 | 0.6981 | 0.8762 | 0.6546 |
| 50 | 95 | 0.6124 | 0.7031 | 0.8740 | 0.6546 |
| 50 | 90 | 0.6124 | 0.7030 | 0.8740 | 0.6546 |
| 40 | 75 | 0.6161 | 0.6979 | 0.8760 | 0.6545 |
| 70 | 85 | 0.5984 | 0.7218 | 0.8644 | 0.6544 |
| 45 | 75 | 0.6145 | 0.6996 | 0.8748 | 0.6543 |
| 50 | 85 | 0.6121 | 0.7025 | 0.8740 | 0.6542 |
| 45 | 80 | 0.6138 | 0.6998 | 0.8745 | 0.6540 |
| 55 | 90 | 0.6100 | 0.7049 | 0.8722 | 0.6540 |
| 55 | 95 | 0.6098 | 0.7051 | 0.8719 | 0.6540 |
| 40 | 80 | 0.6148 | 0.6978 | 0.8758 | 0.6537 |
| 60 | 80 | 0.6053 | 0.7099 | 0.8688 | 0.6534 |
| 50 | 70 | 0.6111 | 0.7020 | 0.8736 | 0.6534 |
| 50 | 75 | 0.6113 | 0.7015 | 0.8738 | 0.6533 |
| 55 | 85 | 0.6083 | 0.7053 | 0.8721 | 0.6532 |
| 50 | 80 | 0.6108 | 0.7016 | 0.8735 | 0.6530 |
| 15 | 20 | 0.6174 | 0.6924 | 0.8838 | 0.6528 |
| 60 | 75 | 0.6059 | 0.7072 | 0.8686 | 0.6526 |

Continued on next page

Table A.2 – continued from previous page

| Pmin | Pmax | Precision | Recall | Specificity | F-measure |
|-------------|-------------|------------------|---------------|--------------------|------------------|
| 55 | 75 | 0.6073 | 0.7027 | 0.8723 | 0.6515 |
| 55 | 80 | 0.6060 | 0.7038 | 0.8716 | 0.6512 |
| 55 | 70 | 0.6055 | 0.7041 | 0.8713 | 0.6511 |
| 65 | 75 | 0.5980 | 0.7143 | 0.8654 | 0.6510 |
| 45 | 60 | 0.6065 | 0.6967 | 0.8723 | 0.6485 |
| 45 | 55 | 0.6016 | 0.7009 | 0.8689 | 0.6475 |
| 30 | 35 | 0.6015 | 0.6991 | 0.8741 | 0.6466 |
| 80 | 85 | 0.5835 | 0.7235 | 0.8555 | 0.6460 |
| 50 | 60 | 0.6020 | 0.6970 | 0.8703 | 0.6460 |
| 50 | 55 | 0.5952 | 0.7048 | 0.8663 | 0.6454 |
| 80 | 95 | 0.5871 | 0.7165 | 0.8638 | 0.6454 |
| 75 | 85 | 0.5851 | 0.7176 | 0.8640 | 0.6446 |
| 70 | 80 | 0.5880 | 0.7131 | 0.8648 | 0.6445 |
| 80 | 90 | 0.5862 | 0.7154 | 0.8632 | 0.6444 |
| 75 | 95 | 0.5859 | 0.7144 | 0.8638 | 0.6438 |
| 75 | 90 | 0.5859 | 0.7130 | 0.8635 | 0.6432 |
| 25 | 30 | 0.5978 | 0.6921 | 0.8787 | 0.6415 |
| 60 | 65 | 0.5838 | 0.6978 | 0.8689 | 0.6357 |
| 70 | 75 | 0.5776 | 0.7018 | 0.8708 | 0.6337 |
| 65 | 70 | 0.5761 | 0.7027 | 0.8670 | 0.6332 |
| 45 | 50 | 0.5821 | 0.6933 | 0.8669 | 0.6328 |
| 55 | 65 | 0.5842 | 0.6869 | 0.8692 | 0.6314 |
| 85 | 95 | 0.5708 | 0.7028 | 0.8732 | 0.6299 |
| 85 | 90 | 0.5645 | 0.6994 | 0.8702 | 0.6248 |
| 55 | 60 | 0.5703 | 0.6837 | 0.8660 | 0.6219 |
| 75 | 80 | 0.5554 | 0.7039 | 0.8527 | 0.6209 |
| 90 | 95 | 0.5536 | 0.7045 | 0.8603 | 0.6200 |

Table A.3: Trapezoids parameters tuning results for HGR dataset using combined rules.

| Pmin | Pmax | Precision | Recall | Specificity | F-measure |
|-------------|-------------|------------------|---------------|--------------------|------------------|
| 5 | 5 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| 10 | 10 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| 15 | 15 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| 20 | 20 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| 25 | 25 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| 30 | 30 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| 35 | 35 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| 40 | 40 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| 45 | 45 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| 50 | 50 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| 55 | 55 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| 60 | 60 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| 65 | 65 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| 70 | 70 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| 75 | 75 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| 80 | 80 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| 85 | 85 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| 90 | 90 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| 95 | 95 | 0.8693 | 0.8641 | 0.9064 | 0.8667 |
| 25 | 40 | 0.8858 | 0.7323 | 0.9277 | 0.8018 |
| 25 | 45 | 0.8867 | 0.7314 | 0.9288 | 0.8016 |

Continued on next page

Table A.3 – continued from previous page

| Pmin | Pmax | Precision | Recall | Specificity | F-measure |
|------|------|-----------|--------|-------------|-----------|
| 25 | 35 | 0.8836 | 0.7334 | 0.9253 | 0.8015 |
| 25 | 50 | 0.8869 | 0.7311 | 0.9291 | 0.8015 |
| 30 | 50 | 0.8862 | 0.7316 | 0.9284 | 0.8015 |
| 20 | 40 | 0.8881 | 0.7301 | 0.9298 | 0.8014 |
| 50 | 75 | 0.8821 | 0.7342 | 0.9246 | 0.8014 |
| 60 | 80 | 0.8782 | 0.7369 | 0.9211 | 0.8014 |
| 65 | 85 | 0.8757 | 0.7387 | 0.9185 | 0.8014 |
| 50 | 80 | 0.8821 | 0.7342 | 0.9246 | 0.8014 |
| 25 | 55 | 0.8870 | 0.7308 | 0.9294 | 0.8014 |
| 25 | 65 | 0.8870 | 0.7308 | 0.9293 | 0.8013 |
| 50 | 90 | 0.8823 | 0.7339 | 0.9248 | 0.8013 |
| 30 | 45 | 0.8842 | 0.7326 | 0.9263 | 0.8013 |
| 50 | 85 | 0.8823 | 0.7339 | 0.9248 | 0.8013 |
| 50 | 70 | 0.8791 | 0.7361 | 0.9208 | 0.8013 |
| 25 | 60 | 0.8870 | 0.7307 | 0.9293 | 0.8013 |
| 60 | 85 | 0.8785 | 0.7365 | 0.9213 | 0.8013 |
| 20 | 45 | 0.8881 | 0.7299 | 0.9298 | 0.8013 |
| 65 | 90 | 0.8759 | 0.7384 | 0.9186 | 0.8013 |
| 60 | 90 | 0.8786 | 0.7364 | 0.9214 | 0.8013 |
| 50 | 95 | 0.8823 | 0.7339 | 0.9248 | 0.8012 |
| 55 | 80 | 0.8802 | 0.7353 | 0.9230 | 0.8012 |
| 35 | 50 | 0.8824 | 0.7337 | 0.9243 | 0.8012 |
| 25 | 70 | 0.8868 | 0.7307 | 0.9292 | 0.8012 |
| 30 | 55 | 0.8862 | 0.7311 | 0.9287 | 0.8012 |
| 65 | 95 | 0.8759 | 0.7382 | 0.9187 | 0.8012 |
| 20 | 35 | 0.8877 | 0.7300 | 0.9296 | 0.8012 |
| 25 | 85 | 0.8869 | 0.7306 | 0.9293 | 0.8012 |
| 25 | 80 | 0.8869 | 0.7306 | 0.9293 | 0.8012 |
| 25 | 95 | 0.8869 | 0.7306 | 0.9293 | 0.8012 |
| 25 | 90 | 0.8869 | 0.7305 | 0.9293 | 0.8012 |
| 25 | 75 | 0.8868 | 0.7305 | 0.9292 | 0.8011 |
| 60 | 95 | 0.8786 | 0.7362 | 0.9215 | 0.8011 |
| 55 | 85 | 0.8804 | 0.7350 | 0.9232 | 0.8011 |
| 40 | 65 | 0.8840 | 0.7324 | 0.9269 | 0.8011 |
| 55 | 90 | 0.8804 | 0.7349 | 0.9232 | 0.8011 |
| 30 | 60 | 0.8861 | 0.7309 | 0.9286 | 0.8011 |
| 40 | 80 | 0.8842 | 0.7322 | 0.9271 | 0.8011 |
| 20 | 50 | 0.8881 | 0.7296 | 0.9298 | 0.8011 |
| 50 | 65 | 0.8770 | 0.7372 | 0.9185 | 0.8011 |
| 30 | 65 | 0.8861 | 0.7309 | 0.9285 | 0.8010 |
| 40 | 95 | 0.8843 | 0.7321 | 0.9271 | 0.8010 |
| 35 | 65 | 0.8851 | 0.7315 | 0.9279 | 0.8010 |
| 40 | 90 | 0.8843 | 0.7321 | 0.9271 | 0.8010 |
| 35 | 55 | 0.8848 | 0.7317 | 0.9276 | 0.8010 |
| 40 | 85 | 0.8842 | 0.7321 | 0.9271 | 0.8010 |
| 35 | 80 | 0.8853 | 0.7314 | 0.9281 | 0.8010 |
| 55 | 95 | 0.8803 | 0.7348 | 0.9232 | 0.8010 |
| 20 | 55 | 0.8882 | 0.7293 | 0.9303 | 0.8010 |
| 20 | 65 | 0.8882 | 0.7294 | 0.9302 | 0.8010 |
| 40 | 70 | 0.8840 | 0.7322 | 0.9267 | 0.8010 |
| 20 | 70 | 0.8882 | 0.7293 | 0.9302 | 0.8009 |
| 40 | 75 | 0.8841 | 0.7321 | 0.9270 | 0.8009 |
| 35 | 90 | 0.8853 | 0.7312 | 0.9282 | 0.8009 |
| 35 | 95 | 0.8853 | 0.7312 | 0.9282 | 0.8009 |
| 35 | 85 | 0.8853 | 0.7312 | 0.9281 | 0.8009 |

Continued on next page

Table A.3 – continued from previous page

| Pmin | Pmax | Precision | Recall | Specificity | F-measure |
|-------------|-------------|------------------|---------------|--------------------|------------------|
| 20 | 60 | 0.8882 | 0.7293 | 0.9301 | 0.8009 |
| 35 | 75 | 0.8852 | 0.7313 | 0.9280 | 0.8009 |
| 35 | 70 | 0.8851 | 0.7314 | 0.9278 | 0.8009 |
| 55 | 75 | 0.8788 | 0.7357 | 0.9212 | 0.8009 |
| 20 | 80 | 0.8882 | 0.7292 | 0.9302 | 0.8009 |
| 35 | 45 | 0.8767 | 0.7372 | 0.9176 | 0.8009 |
| 35 | 60 | 0.8848 | 0.7315 | 0.9276 | 0.8009 |
| 20 | 75 | 0.8882 | 0.7292 | 0.9302 | 0.8009 |
| 30 | 80 | 0.8860 | 0.7307 | 0.9285 | 0.8009 |
| 30 | 70 | 0.8859 | 0.7308 | 0.9284 | 0.8009 |
| 20 | 95 | 0.8882 | 0.7292 | 0.9303 | 0.8009 |
| 20 | 85 | 0.8882 | 0.7292 | 0.9303 | 0.8009 |
| 20 | 90 | 0.8882 | 0.7292 | 0.9303 | 0.8009 |
| 30 | 75 | 0.8860 | 0.7307 | 0.9284 | 0.8009 |
| 40 | 55 | 0.8815 | 0.7337 | 0.9241 | 0.8008 |
| 30 | 85 | 0.8860 | 0.7306 | 0.9285 | 0.8008 |
| 30 | 90 | 0.8860 | 0.7306 | 0.9285 | 0.8008 |
| 45 | 65 | 0.8822 | 0.7332 | 0.9250 | 0.8008 |
| 30 | 95 | 0.8860 | 0.7306 | 0.9285 | 0.8008 |
| 45 | 80 | 0.8829 | 0.7327 | 0.9257 | 0.8008 |
| 40 | 60 | 0.8831 | 0.7325 | 0.9260 | 0.8008 |
| 45 | 75 | 0.8829 | 0.7327 | 0.9257 | 0.8008 |
| 45 | 85 | 0.8830 | 0.7325 | 0.9259 | 0.8008 |
| 45 | 70 | 0.8823 | 0.7331 | 0.9249 | 0.8008 |
| 45 | 90 | 0.8831 | 0.7325 | 0.9259 | 0.8007 |
| 30 | 40 | 0.8776 | 0.7362 | 0.9184 | 0.8007 |
| 45 | 95 | 0.8830 | 0.7324 | 0.9258 | 0.8007 |
| 65 | 80 | 0.8701 | 0.7415 | 0.9117 | 0.8007 |
| 60 | 75 | 0.8737 | 0.7388 | 0.9161 | 0.8006 |
| 20 | 30 | 0.8861 | 0.7301 | 0.9284 | 0.8005 |
| 55 | 70 | 0.8737 | 0.7385 | 0.9150 | 0.8005 |
| 45 | 60 | 0.8800 | 0.7339 | 0.9227 | 0.8003 |
| 15 | 25 | 0.8880 | 0.7282 | 0.9297 | 0.8002 |
| 15 | 20 | 0.8795 | 0.7340 | 0.9192 | 0.8002 |
| 70 | 85 | 0.8700 | 0.7406 | 0.9122 | 0.8001 |
| 15 | 40 | 0.8895 | 0.7269 | 0.9314 | 0.8000 |
| 5 | 10 | 0.8961 | 0.7225 | 0.9360 | 0.8000 |
| 20 | 25 | 0.8722 | 0.7389 | 0.9105 | 0.8000 |
| 70 | 90 | 0.8703 | 0.7401 | 0.9125 | 0.7999 |
| 15 | 30 | 0.8887 | 0.7273 | 0.9304 | 0.7999 |
| 15 | 35 | 0.8893 | 0.7269 | 0.9312 | 0.7999 |
| 5 | 15 | 0.8970 | 0.7218 | 0.9366 | 0.7999 |
| 5 | 20 | 0.8971 | 0.7217 | 0.9367 | 0.7999 |
| 10 | 20 | 0.8917 | 0.7252 | 0.9332 | 0.7999 |
| 5 | 40 | 0.8971 | 0.7217 | 0.9366 | 0.7999 |
| 50 | 60 | 0.8677 | 0.7419 | 0.9070 | 0.7999 |
| 15 | 45 | 0.8896 | 0.7266 | 0.9315 | 0.7999 |
| 5 | 25 | 0.8970 | 0.7217 | 0.9366 | 0.7999 |
| 5 | 45 | 0.8971 | 0.7216 | 0.9366 | 0.7999 |
| 5 | 55 | 0.8971 | 0.7216 | 0.9366 | 0.7998 |
| 5 | 50 | 0.8971 | 0.7216 | 0.9366 | 0.7998 |
| 5 | 60 | 0.8971 | 0.7216 | 0.9366 | 0.7998 |
| 5 | 30 | 0.8970 | 0.7216 | 0.9366 | 0.7998 |
| 40 | 50 | 0.8750 | 0.7366 | 0.9170 | 0.7998 |
| 5 | 90 | 0.8970 | 0.7216 | 0.9367 | 0.7998 |

Continued on next page

Table A.3 – continued from previous page

| Pmin | Pmax | Precision | Recall | Specificity | F-measure |
|------|------|-----------|--------|-------------|-----------|
| 5 | 95 | 0.8970 | 0.7216 | 0.9367 | 0.7998 |
| 5 | 80 | 0.8970 | 0.7216 | 0.9366 | 0.7998 |
| 5 | 85 | 0.8970 | 0.7216 | 0.9367 | 0.7998 |
| 5 | 75 | 0.8970 | 0.7216 | 0.9366 | 0.7998 |
| 5 | 70 | 0.8970 | 0.7216 | 0.9366 | 0.7998 |
| 70 | 95 | 0.8706 | 0.7397 | 0.9126 | 0.7998 |
| 5 | 65 | 0.8970 | 0.7216 | 0.9366 | 0.7998 |
| 5 | 35 | 0.8970 | 0.7215 | 0.9366 | 0.7998 |
| 15 | 50 | 0.8895 | 0.7265 | 0.9315 | 0.7998 |
| 10 | 30 | 0.8921 | 0.7247 | 0.9334 | 0.7997 |
| 10 | 25 | 0.8918 | 0.7249 | 0.9333 | 0.7997 |
| 15 | 55 | 0.8894 | 0.7264 | 0.9315 | 0.7997 |
| 10 | 45 | 0.8922 | 0.7245 | 0.9334 | 0.7996 |
| 10 | 40 | 0.8922 | 0.7245 | 0.9334 | 0.7996 |
| 10 | 35 | 0.8922 | 0.7245 | 0.9335 | 0.7996 |
| 10 | 95 | 0.8922 | 0.7245 | 0.9335 | 0.7996 |
| 10 | 90 | 0.8922 | 0.7245 | 0.9335 | 0.7996 |
| 10 | 80 | 0.8922 | 0.7245 | 0.9335 | 0.7996 |
| 10 | 85 | 0.8922 | 0.7245 | 0.9335 | 0.7996 |
| 15 | 90 | 0.8894 | 0.7263 | 0.9315 | 0.7996 |
| 10 | 75 | 0.8922 | 0.7245 | 0.9335 | 0.7996 |
| 10 | 70 | 0.8922 | 0.7245 | 0.9335 | 0.7996 |
| 15 | 95 | 0.8894 | 0.7263 | 0.9315 | 0.7996 |
| 15 | 85 | 0.8894 | 0.7263 | 0.9315 | 0.7996 |
| 15 | 65 | 0.8894 | 0.7263 | 0.9314 | 0.7996 |
| 15 | 60 | 0.8894 | 0.7263 | 0.9314 | 0.7996 |
| 15 | 80 | 0.8894 | 0.7263 | 0.9315 | 0.7996 |
| 15 | 70 | 0.8894 | 0.7263 | 0.9315 | 0.7996 |
| 10 | 65 | 0.8921 | 0.7245 | 0.9334 | 0.7996 |
| 15 | 75 | 0.8894 | 0.7263 | 0.9315 | 0.7996 |
| 10 | 60 | 0.8921 | 0.7245 | 0.9334 | 0.7996 |
| 10 | 55 | 0.8921 | 0.7245 | 0.9334 | 0.7996 |
| 10 | 50 | 0.8922 | 0.7244 | 0.9334 | 0.7996 |
| 10 | 15 | 0.8872 | 0.7276 | 0.9275 | 0.7995 |
| 75 | 90 | 0.8609 | 0.7460 | 0.9016 | 0.7993 |
| 75 | 85 | 0.8557 | 0.7496 | 0.8946 | 0.7992 |
| 75 | 95 | 0.8610 | 0.7452 | 0.9017 | 0.7989 |
| 55 | 65 | 0.8644 | 0.7426 | 0.9039 | 0.7989 |
| 60 | 70 | 0.8627 | 0.7429 | 0.9039 | 0.7983 |
| 45 | 55 | 0.8725 | 0.7356 | 0.9156 | 0.7982 |
| 25 | 30 | 0.8671 | 0.7393 | 0.9055 | 0.7981 |
| 30 | 35 | 0.8607 | 0.7426 | 0.8998 | 0.7973 |
| 65 | 75 | 0.8579 | 0.7432 | 0.9004 | 0.7965 |
| 35 | 40 | 0.8547 | 0.7456 | 0.8918 | 0.7964 |
| 70 | 80 | 0.8566 | 0.7441 | 0.8980 | 0.7964 |
| 40 | 45 | 0.8526 | 0.7448 | 0.8922 | 0.7950 |
| 55 | 60 | 0.8392 | 0.7530 | 0.8756 | 0.7938 |
| 45 | 50 | 0.8501 | 0.7437 | 0.8892 | 0.7934 |
| 80 | 90 | 0.8423 | 0.7483 | 0.8849 | 0.7925 |
| 80 | 95 | 0.8437 | 0.7471 | 0.8864 | 0.7925 |
| 50 | 55 | 0.8435 | 0.7454 | 0.8838 | 0.7914 |
| 60 | 65 | 0.8393 | 0.7484 | 0.8793 | 0.7913 |
| 65 | 70 | 0.8328 | 0.7482 | 0.8739 | 0.7882 |
| 75 | 80 | 0.8278 | 0.7501 | 0.8688 | 0.7870 |
| 85 | 95 | 0.8198 | 0.7556 | 0.8570 | 0.7864 |

Continued on next page

Table A.3 – continued from previous page

| Pmin | Pmax | Precision | Recall | Specificity | F-measure |
|-------------|-------------|------------------|---------------|--------------------|------------------|
| 80 | 85 | 0.8228 | 0.7524 | 0.8639 | 0.7861 |
| 70 | 75 | 0.8259 | 0.7495 | 0.8660 | 0.7859 |
| 90 | 95 | 0.7998 | 0.7664 | 0.8337 | 0.7827 |
| 85 | 90 | 0.8071 | 0.7515 | 0.8493 | 0.7783 |

Table A.4: Trapezoids parameters tuning results for SFA dataset using combined rules.

| Pmin | Pmax | Precision | Recall | Specificity | F-measure |
|-------------|-------------|------------------|---------------|--------------------|------------------|
| 5 | 5 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| 10 | 10 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| 15 | 15 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| 20 | 20 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| 25 | 25 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| 30 | 30 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| 35 | 35 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| 40 | 40 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| 45 | 45 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| 50 | 50 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| 55 | 55 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| 60 | 60 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| 65 | 65 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| 70 | 70 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| 75 | 75 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| 80 | 80 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| 85 | 85 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| 90 | 90 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| 95 | 95 | 0.9313 | 0.6617 | 0.9785 | 0.7737 |
| 85 | 90 | 0.8307 | 0.4534 | 0.9519 | 0.5866 |
| 70 | 75 | 0.8554 | 0.4442 | 0.9651 | 0.5848 |
| 80 | 90 | 0.8721 | 0.4385 | 0.9733 | 0.5835 |
| 50 | 55 | 0.8721 | 0.4383 | 0.9718 | 0.5834 |
| 85 | 95 | 0.8610 | 0.4411 | 0.9685 | 0.5833 |
| 80 | 95 | 0.8743 | 0.4376 | 0.9742 | 0.5833 |
| 60 | 65 | 0.8631 | 0.4403 | 0.9682 | 0.5831 |
| 75 | 90 | 0.8795 | 0.4358 | 0.9779 | 0.5828 |
| 70 | 90 | 0.8870 | 0.4337 | 0.9801 | 0.5825 |
| 75 | 95 | 0.8803 | 0.4352 | 0.9780 | 0.5825 |
| 70 | 85 | 0.8849 | 0.4340 | 0.9797 | 0.5823 |
| 70 | 95 | 0.8874 | 0.4332 | 0.9801 | 0.5822 |
| 70 | 80 | 0.8796 | 0.4349 | 0.9784 | 0.5820 |
| 60 | 70 | 0.8860 | 0.4333 | 0.9809 | 0.5819 |
| 65 | 80 | 0.8877 | 0.4326 | 0.9805 | 0.5817 |
| 55 | 60 | 0.8657 | 0.4380 | 0.9725 | 0.5816 |
| 65 | 70 | 0.8512 | 0.4416 | 0.9664 | 0.5815 |
| 60 | 75 | 0.8914 | 0.4315 | 0.9823 | 0.5815 |
| 65 | 75 | 0.8832 | 0.4334 | 0.9796 | 0.5814 |
| 60 | 80 | 0.8938 | 0.4308 | 0.9826 | 0.5814 |
| 75 | 80 | 0.8375 | 0.4452 | 0.9599 | 0.5814 |
| 75 | 85 | 0.8739 | 0.4355 | 0.9768 | 0.5813 |
| 45 | 55 | 0.8956 | 0.4302 | 0.9824 | 0.5812 |
| 55 | 65 | 0.8872 | 0.4320 | 0.9811 | 0.5810 |
| 50 | 60 | 0.8888 | 0.4315 | 0.9806 | 0.5810 |

Continued on next page

Table A.4 – continued from previous page

| Pmin | Pmax | Precision | Recall | Specificity | F-measure |
|------|------|-----------|--------|-------------|-----------|
| 55 | 70 | 0.8907 | 0.4308 | 0.9817 | 0.5808 |
| 45 | 60 | 0.8991 | 0.4289 | 0.9834 | 0.5807 |
| 55 | 80 | 0.8968 | 0.4294 | 0.9830 | 0.5807 |
| 65 | 90 | 0.8913 | 0.4306 | 0.9817 | 0.5807 |
| 55 | 75 | 0.8945 | 0.4298 | 0.9828 | 0.5806 |
| 65 | 85 | 0.8902 | 0.4307 | 0.9816 | 0.5806 |
| 65 | 95 | 0.8913 | 0.4304 | 0.9818 | 0.5805 |
| 35 | 40 | 0.8897 | 0.4308 | 0.9797 | 0.5805 |
| 60 | 90 | 0.8952 | 0.4293 | 0.9828 | 0.5803 |
| 60 | 85 | 0.8949 | 0.4294 | 0.9827 | 0.5803 |
| 60 | 95 | 0.8951 | 0.4292 | 0.9828 | 0.5802 |
| 55 | 90 | 0.8982 | 0.4284 | 0.9833 | 0.5801 |
| 45 | 65 | 0.9000 | 0.4279 | 0.9836 | 0.5800 |
| 55 | 85 | 0.8977 | 0.4284 | 0.9832 | 0.5800 |
| 50 | 80 | 0.8997 | 0.4279 | 0.9837 | 0.5799 |
| 40 | 45 | 0.8885 | 0.4304 | 0.9802 | 0.5799 |
| 55 | 95 | 0.8982 | 0.4282 | 0.9833 | 0.5799 |
| 30 | 35 | 0.8984 | 0.4281 | 0.9821 | 0.5799 |
| 45 | 75 | 0.9035 | 0.4270 | 0.9844 | 0.5799 |
| 50 | 75 | 0.8985 | 0.4280 | 0.9836 | 0.5799 |
| 40 | 55 | 0.9000 | 0.4277 | 0.9837 | 0.5798 |
| 45 | 70 | 0.9011 | 0.4274 | 0.9838 | 0.5798 |
| 45 | 80 | 0.9039 | 0.4268 | 0.9845 | 0.5798 |
| 35 | 45 | 0.9005 | 0.4275 | 0.9836 | 0.5798 |
| 50 | 70 | 0.8953 | 0.4286 | 0.9827 | 0.5797 |
| 45 | 50 | 0.8804 | 0.4321 | 0.9784 | 0.5797 |
| 40 | 50 | 0.8982 | 0.4279 | 0.9835 | 0.5796 |
| 50 | 65 | 0.8919 | 0.4291 | 0.9822 | 0.5795 |
| 35 | 50 | 0.9039 | 0.4263 | 0.9844 | 0.5794 |
| 40 | 60 | 0.9017 | 0.4267 | 0.9841 | 0.5793 |
| 45 | 90 | 0.9046 | 0.4259 | 0.9846 | 0.5792 |
| 50 | 90 | 0.9006 | 0.4268 | 0.9839 | 0.5791 |
| 45 | 85 | 0.9045 | 0.4259 | 0.9846 | 0.5791 |
| 50 | 85 | 0.9002 | 0.4268 | 0.9838 | 0.5791 |
| 45 | 95 | 0.9046 | 0.4258 | 0.9846 | 0.5790 |
| 50 | 95 | 0.9006 | 0.4267 | 0.9839 | 0.5790 |
| 30 | 40 | 0.9054 | 0.4255 | 0.9842 | 0.5790 |
| 40 | 65 | 0.9029 | 0.4260 | 0.9843 | 0.5789 |
| 35 | 55 | 0.9050 | 0.4255 | 0.9846 | 0.5789 |
| 40 | 70 | 0.9038 | 0.4256 | 0.9845 | 0.5787 |
| 40 | 75 | 0.9051 | 0.4254 | 0.9848 | 0.5787 |
| 25 | 35 | 0.9103 | 0.4242 | 0.9855 | 0.5787 |
| 25 | 30 | 0.9049 | 0.4253 | 0.9844 | 0.5786 |
| 40 | 80 | 0.9053 | 0.4252 | 0.9849 | 0.5786 |
| 35 | 60 | 0.9065 | 0.4249 | 0.9849 | 0.5786 |
| 30 | 45 | 0.9069 | 0.4247 | 0.9848 | 0.5785 |
| 30 | 50 | 0.9078 | 0.4243 | 0.9852 | 0.5783 |
| 35 | 65 | 0.9072 | 0.4243 | 0.9850 | 0.5781 |
| 40 | 90 | 0.9061 | 0.4244 | 0.9851 | 0.5781 |
| 25 | 40 | 0.9113 | 0.4233 | 0.9857 | 0.5781 |
| 40 | 85 | 0.9057 | 0.4244 | 0.9850 | 0.5780 |
| 30 | 55 | 0.9086 | 0.4238 | 0.9855 | 0.5780 |
| 35 | 75 | 0.9083 | 0.4238 | 0.9853 | 0.5780 |
| 35 | 70 | 0.9077 | 0.4239 | 0.9851 | 0.5779 |
| 40 | 95 | 0.9061 | 0.4243 | 0.9851 | 0.5779 |

Continued on next page

Table A.4 – continued from previous page

| Pmin | Pmax | Precision | Recall | Specificity | F-measure |
|-------------|-------------|------------------|---------------|--------------------|------------------|
| 35 | 80 | 0.9084 | 0.4237 | 0.9853 | 0.5779 |
| 30 | 60 | 0.9098 | 0.4234 | 0.9857 | 0.5778 |
| 30 | 65 | 0.9101 | 0.4230 | 0.9858 | 0.5776 |
| 35 | 90 | 0.9093 | 0.4231 | 0.9855 | 0.5775 |
| 30 | 70 | 0.9101 | 0.4229 | 0.9859 | 0.5774 |
| 30 | 75 | 0.9106 | 0.4227 | 0.9860 | 0.5774 |
| 35 | 85 | 0.9089 | 0.4231 | 0.9854 | 0.5774 |
| 35 | 95 | 0.9093 | 0.4229 | 0.9855 | 0.5773 |
| 25 | 45 | 0.9119 | 0.4223 | 0.9860 | 0.5773 |
| 30 | 80 | 0.9106 | 0.4226 | 0.9860 | 0.5773 |
| 25 | 50 | 0.9124 | 0.4220 | 0.9861 | 0.5771 |
| 20 | 25 | 0.9111 | 0.4220 | 0.9854 | 0.5768 |
| 30 | 90 | 0.9109 | 0.4219 | 0.9861 | 0.5767 |
| 20 | 30 | 0.9156 | 0.4209 | 0.9865 | 0.5767 |
| 25 | 55 | 0.9132 | 0.4214 | 0.9865 | 0.5767 |
| 30 | 85 | 0.9107 | 0.4219 | 0.9860 | 0.5767 |
| 30 | 95 | 0.9109 | 0.4218 | 0.9861 | 0.5766 |
| 25 | 60 | 0.9138 | 0.4211 | 0.9866 | 0.5765 |
| 25 | 65 | 0.9140 | 0.4207 | 0.9866 | 0.5762 |
| 25 | 70 | 0.9139 | 0.4205 | 0.9867 | 0.5760 |
| 25 | 75 | 0.9144 | 0.4204 | 0.9867 | 0.5760 |
| 25 | 80 | 0.9144 | 0.4203 | 0.9867 | 0.5759 |
| 20 | 35 | 0.9169 | 0.4194 | 0.9869 | 0.5755 |
| 25 | 90 | 0.9147 | 0.4198 | 0.9868 | 0.5754 |
| 25 | 95 | 0.9147 | 0.4197 | 0.9868 | 0.5754 |
| 80 | 85 | 0.8375 | 0.4382 | 0.9608 | 0.5753 |
| 25 | 85 | 0.9144 | 0.4197 | 0.9868 | 0.5753 |
| 20 | 40 | 0.9172 | 0.4190 | 0.9870 | 0.5752 |
| 20 | 45 | 0.9170 | 0.4184 | 0.9870 | 0.5746 |
| 20 | 50 | 0.9170 | 0.4182 | 0.9871 | 0.5745 |
| 20 | 55 | 0.9173 | 0.4178 | 0.9872 | 0.5741 |
| 20 | 60 | 0.9175 | 0.4175 | 0.9872 | 0.5739 |
| 20 | 65 | 0.9178 | 0.4172 | 0.9873 | 0.5737 |
| 20 | 70 | 0.9178 | 0.4170 | 0.9874 | 0.5735 |
| 20 | 75 | 0.9179 | 0.4169 | 0.9874 | 0.5734 |
| 20 | 80 | 0.9179 | 0.4168 | 0.9874 | 0.5733 |
| 20 | 90 | 0.9179 | 0.4164 | 0.9874 | 0.5729 |
| 20 | 85 | 0.9179 | 0.4163 | 0.9874 | 0.5728 |
| 20 | 95 | 0.9179 | 0.4163 | 0.9874 | 0.5728 |
| 15 | 20 | 0.9162 | 0.4157 | 0.9859 | 0.5719 |
| 15 | 25 | 0.9186 | 0.4146 | 0.9867 | 0.5713 |
| 15 | 30 | 0.9206 | 0.4140 | 0.9872 | 0.5711 |
| 15 | 35 | 0.9213 | 0.4131 | 0.9876 | 0.5704 |
| 15 | 40 | 0.9213 | 0.4130 | 0.9876 | 0.5703 |
| 15 | 45 | 0.9211 | 0.4125 | 0.9876 | 0.5698 |
| 15 | 50 | 0.9211 | 0.4124 | 0.9877 | 0.5697 |
| 15 | 55 | 0.9212 | 0.4120 | 0.9878 | 0.5693 |
| 15 | 60 | 0.9211 | 0.4117 | 0.9878 | 0.5690 |
| 15 | 65 | 0.9214 | 0.4114 | 0.9879 | 0.5688 |
| 15 | 75 | 0.9217 | 0.4111 | 0.9880 | 0.5686 |
| 15 | 70 | 0.9213 | 0.4112 | 0.9879 | 0.5686 |
| 15 | 80 | 0.9217 | 0.4111 | 0.9880 | 0.5686 |
| 15 | 90 | 0.9217 | 0.4107 | 0.9880 | 0.5682 |
| 15 | 85 | 0.9218 | 0.4106 | 0.9880 | 0.5682 |
| 15 | 95 | 0.9217 | 0.4106 | 0.9880 | 0.5681 |

Continued on next page

Table A.4 – continued from previous page

| Pmin | Pmax | Precision | Recall | Specificity | F-measure |
|------|------|-----------|--------|-------------|-----------|
| 10 | 15 | 0.9197 | 0.4092 | 0.9870 | 0.5664 |
| 10 | 20 | 0.9218 | 0.4071 | 0.9873 | 0.5648 |
| 10 | 25 | 0.9223 | 0.4066 | 0.9875 | 0.5644 |
| 10 | 30 | 0.9236 | 0.4060 | 0.9879 | 0.5641 |
| 10 | 35 | 0.9242 | 0.4052 | 0.9882 | 0.5634 |
| 10 | 40 | 0.9243 | 0.4052 | 0.9883 | 0.5634 |
| 10 | 45 | 0.9241 | 0.4049 | 0.9883 | 0.5631 |
| 10 | 50 | 0.9240 | 0.4048 | 0.9883 | 0.5630 |
| 90 | 95 | 0.8103 | 0.4311 | 0.9542 | 0.5628 |
| 10 | 55 | 0.9240 | 0.4045 | 0.9884 | 0.5627 |
| 10 | 60 | 0.9239 | 0.4042 | 0.9884 | 0.5624 |
| 10 | 65 | 0.9242 | 0.4039 | 0.9885 | 0.5621 |
| 10 | 70 | 0.9243 | 0.4038 | 0.9885 | 0.5621 |
| 10 | 75 | 0.9243 | 0.4037 | 0.9885 | 0.5620 |
| 10 | 80 | 0.9243 | 0.4037 | 0.9886 | 0.5619 |
| 10 | 85 | 0.9248 | 0.4032 | 0.9886 | 0.5615 |
| 10 | 90 | 0.9247 | 0.4031 | 0.9886 | 0.5615 |
| 10 | 95 | 0.9246 | 0.4030 | 0.9886 | 0.5614 |
| 5 | 10 | 0.9241 | 0.4006 | 0.9873 | 0.5589 |
| 5 | 15 | 0.9260 | 0.3982 | 0.9878 | 0.5569 |
| 5 | 20 | 0.9269 | 0.3972 | 0.9881 | 0.5561 |
| 5 | 25 | 0.9267 | 0.3968 | 0.9882 | 0.5557 |
| 5 | 50 | 0.9282 | 0.3961 | 0.9889 | 0.5553 |
| 5 | 30 | 0.9269 | 0.3963 | 0.9884 | 0.5553 |
| 5 | 60 | 0.9284 | 0.3960 | 0.9890 | 0.5552 |
| 5 | 90 | 0.9284 | 0.3960 | 0.9890 | 0.5552 |
| 5 | 55 | 0.9284 | 0.3960 | 0.9890 | 0.5552 |
| 5 | 40 | 0.9275 | 0.3962 | 0.9887 | 0.5552 |
| 5 | 85 | 0.9284 | 0.3960 | 0.9890 | 0.5552 |
| 5 | 45 | 0.9280 | 0.3961 | 0.9889 | 0.5552 |
| 5 | 95 | 0.9284 | 0.3960 | 0.9890 | 0.5552 |
| 5 | 65 | 0.9283 | 0.3960 | 0.9890 | 0.5551 |
| 5 | 70 | 0.9284 | 0.3959 | 0.9890 | 0.5551 |
| 5 | 80 | 0.9283 | 0.3959 | 0.9890 | 0.5551 |
| 5 | 75 | 0.9284 | 0.3959 | 0.9890 | 0.5551 |
| 5 | 35 | 0.9274 | 0.3960 | 0.9887 | 0.5550 |

Bibliography

- Albiol *et al.*(2001)** Alberto Albiol, Luis Torres and Edward J. Delp. Optimum color spaces for skin detection. In *International Conference on Image Processing*, pages 122–124. 10
- Anderson and Parrish(1981)** R. Rox Anderson and John A. Parrish. The optics of human skin. *Journal of investigative dermatology*, 77(1):13–19. 1
- Basilio *et al.*(2011)** Jorge Alberto Marcial Basilio, Gualberto Aguilar Torres, Gabriel Sánchez Pérez, L. Karina Toscano Medina and Hector M. Perez Meana. Explicit image detection using YCbCr space color model as skin detection. *Applications of Mathematics and Computer Engineering*, pages 123–128. 9, 17, 64
- Ben(2009)** R. G. Ben. CIE 1931 xy color space diagram. https://en.wikipedia.org/wiki/File:CIE1931xy_blank.svg, 2009. Last access 12/10/2016. 21
- Bergasa *et al.*(2000)** Luis Miguel Bergasa, Manuel Mazo, Alfredo Gardel, Miguel A. Sotelo and Luciano Boquete. Unsupervised and adaptive gaussian skin-color model. *Image and Vision Computing*, 18(12):987–1003. 10
- Brancati *et al.*(2017)** Nadia Brancati, Giuseppe De Pietro, Maria Frucci and Luigi Gallo. Human skin detection through correlation rules between the YCb and YCr subspaces based on dynamic color clustering. *Computer Vision and Image Understanding*, 155: 33–42. 4, 5, 11, 29, 30, 31, 32, 33, 34, 41, 43, 44, 45, 46, 47, 48, 50, 52, 63, 64
- Casati *et al.*(2013)** João Paulo Brognoni Casati, Diego Rafael Moraes and Evandro Luis Linhari Rodrigues. SFA: A human skin image database based on FERET and AR facial images. In *IX workshop de Visão Computacional*. 39, 40, 41
- Chai and Ngan(1999)** Douglas Chai and King N. Ngan. Face segmentation using skin-color map in videophone applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(4):551–564. 9, 17, 29, 30, 64
- Chaplin(2004)** George Chaplin. Geographic distribution of environmental factors influencing human skin coloration. *American Journal of Physical Anthropology*, 125(3):292–302. 2
- Chaves-González *et al.*(2010)** Jose M. Chaves-González, Miguel A. Vega-Rodríguez, Juan A. Gómez-Pulido and Juan M. Sánchez-Pérez. Detecting skin in face recognition systems: A colour spaces study. *Digital Signal Processing*, 20(3):806–823. 10, 11, 64
- Edwards and Duntley(1939)** Edward A. Edwards and S. Quimby Duntley. The pigments and color of living human skin. *Developmental Dynamics*, 65(1):1–33. 1
- Eugster(2010)** Simon A. Eugster. The YCbCr color model. https://en.wikipedia.org/wiki/File:YCbCr-CbCr_Scaled_Y50.png, 2010. Last access 26/09/2018. 25

- Faria and Hirata Jr.(2018)** Rodrigo Augusto Dias Faria and Roberto Hirata Jr. Combined correlation rules to detect skin based on dynamic color clustering. In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - VISAPP*, volume 5, pages 309–316. INSTICC, SciTePress. ISBN 978-989-758-290-5. doi: 10.5220/0006618003090316. [5](#)
- Fisher et al.(2003)** Robert Fisher, Simon Perkins, Ashley Walker and Erik Wolfart. Intensity histogram. <http://homepages.inf.ed.ac.uk/rbf/HIPR2/histogram.htm>, 2003. Last access 13/03/2018. [16](#)
- Fleck et al.(1996)** Margaret M. Fleck, David A. Forsyth and Chris Bregler. Finding naked people. In *European Conference on Computer Vision*, pages 593–602. Springer. [3](#), [29](#)
- Gevers et al.(2012)** Theo Gevers, Arjan Gijsenij, Joost van de Weijer and Jan-Mark Geusebroek. *Color in Computer Vision: Fundamentals and Applications*. Wiley. [22](#)
- Gonzalez and Woods(2002)** Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice Hall, 2nd edition. [3](#), [13](#), [14](#), [15](#), [16](#), [17](#), [18](#), [19](#), [22](#), [23](#)
- Grzejszczak et al.(2016)** Tomasz Grzejszczak, Michal Kawulok and Adam Galuszka. Hand landmarks detection and localization in color images. *Multimedia Tools and Applications*, 75(23):16363–16387. ISSN 1573-7721. doi: 10.1007/s11042-015-2934-5. [40](#), [42](#)
- Hsu et al.(2002)** Rein-Lien Hsu, M. Abdel-Mottaleb and A. K. Jain. Face detection in color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):696–706. ISSN 0162-8828. doi: 10.1109/34.1000242. [8](#), [29](#)
- Ice(2016)** Black Ice. The HSI color space. <http://www.blackice.com/images/HSIColorModel.jpg>, 2016. Last access 15/10/2016. [26](#)
- Jablonski(2004)** Nina G. Jablonski. The evolution of human skin and skin color. *Annual Review of Anthropology*, 33:585–623. [2](#)
- Jablonski and Chaplin(2000)** Nina G. Jablonski and George Chaplin. The evolution of human skin coloration. *Journal of Human Evolution*, 39(1):57–106. [2](#)
- Jablonski and Chaplin(2010)** Nina G. Jablonski and George Chaplin. Human skin pigmentation as an adaptation to UV radiation. *Proceedings of the National Academy of Sciences*, 107(Supplement 2):8962–8968. [2](#)
- Jayaram et al.(2004)** Sriram Jayaram, Stephen Schmugge, Min C. Shin and Leonid V. Tsap. Effect of colorspace transformation, the illuminance component, and color modeling on skin detection. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 813–818. IEEE. [10](#)
- Jones and Rehg(2002)** Michael J. Jones and James M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96. [7](#), [8](#), [41](#), [43](#)
- Kakumanu et al.(2007)** Praveen Kakumanu, Sokratis Makrogiannis and Nikolaos Bourbakis. A survey of skin-color modeling and detection methods. *Pattern Recognition*, 40(3):1106–1122. [2](#), [4](#), [7](#), [10](#), [24](#)

- Kaur and Kranthi(2012)** Amanpreet Kaur and B. V. Kranthi. Comparison between YCbCr color space and CIELab color space for skin color segmentation. *International Journal of Applied Information Systems*, 3(4):30–33. [11](#), [17](#)
- Kawulok et al.(2013)** Michal Kawulok, Jolanta Kawulok, Jakub Nalepa and Maciej Papiez. Skin detection using spatial analysis with adaptive seed. In *2013 IEEE International Conference on Image Processing*, pages 3720–3724. IEEE. [10](#)
- Kawulok et al.(2014)** Michal Kawulok, Jolanta Kawulok, Jakub Nalepa and Bogdan Smolka. Self-adaptive algorithm for segmenting skin regions. *EURASIP Journal on Advances in Signal Processing*, 2014(170):1–22. ISSN 1687-6180. doi: 10.1186/1687-6180-2014-170. [40](#), [42](#)
- Khan et al.(2012)** Rehanullah Khan, Allan Hanbury, Julian Stöttinger and Abdul Bais. Color based skin classification. *Pattern Recognition Letters*, 33(2):157–163. [11](#)
- Kovac et al.(2003)** Jure Kovac, Peter Peer and Franc Solina. *Human skin color clustering for face detection*, volume 2. IEEE. [8](#), [9](#), [11](#), [17](#)
- Kumar and Malhotra(2015)** Amit Kumar and Shivani Malhotra. Performance analysis of color space for optimum skin color detection. In *2015 Fifth International Conference on Communication Systems and Network Technologies*, pages 554–558. IEEE. [11](#), [17](#)
- Mahmoodi and Sayedi(2016)** Mohammad Reza Mahmoodi and Sayed Masoud Sayedi. A comprehensive survey on human skin detection. *International Journal of Image, Graphics and Signal Processing*, 8(5):1–35. [7](#), [10](#), [41](#)
- Martínez and Benavente(1998)** Aleix Martínez and Robert Benavente. The AR face database. Technical report, Purdue University. Last access 22/03/2017. [10](#), [39](#)
- Naji et al.(2012)** Sinan A. Naji, Roziati Zainuddin and Hamid A. Jalab. Skin segmentation based on multi pixel color clustering models. *Digital Signal Processing*, 22(6):933–940. [10](#)
- Nalepa and Kawulok(2014)** Jakub Nalepa and Michal Kawulok. Fast and accurate hand shape classification. In Stanislaw Kozielski, Dariusz Mrozek, Paweł Kasprowski, Bozena Malysiak-Mrozek and Daniel Kostrzewa, editors, *Beyond Databases, Architectures, and Structures*, volume 424 of *Communications in Computer and Information Science*, pages 364–373. Springer. ISBN 978-3-319-06931-9. doi: 10.1007/978-3-319-06932-6_35. URL http://dx.doi.org/10.1007/978-3-319-06932-6_35. Last access 18/04/2017. [40](#), [42](#)
- Pedrini and Schwartz(2008)** Hélio Pedrini and William Robson Schwartz. *Análise de imagens digitais: princípios, algoritmos e aplicações*. Thomson Learning, São Paulo. [13](#), [14](#), [15](#), [16](#), [17](#), [24](#), [25](#), [26](#), [27](#)
- Phillips et al.(1996)** P. Jonathon Phillips, Harry Wechsler, Jeffrey Huang and Patrick J. Rauss. The facial recognition technology (FERET) database. <https://www.nist.gov/programs-projects/face-recognition-technology-feret>, 1996. Last access 23/06/2016. [39](#)
- Plataniotis and Venetsanopoulos(2000)** Konstantinos N. Plataniotis and Anastasios N. Venetsanopoulos. *Color Image Processing and Applications*. Springer, 1st edition. [17](#), [18](#), [19](#), [20](#), [22](#), [23](#), [24](#), [25](#)
- Rosenfeld and Pfaltz(1966)** Azriel Rosenfeld and John L. Pfaltz. Sequential operations in digital picture processing. *Journal of the ACM (JACM)*, 13(4):471–494. [35](#)

- Rus(2007)** Jacob Rus. The Munsell color system. <https://commons.wikimedia.org/wiki/File:Munsell-system.svg>, 2007. Last access 12/10/2016. 20
- Rus(2008)** Jacob Rus. The "primary" and "secondary" colors in a four-color print process. <https://en.wikipedia.org/wiki/File:SubtractiveColor.svg>, 2008. Last access 12/10/2016. 23
- Shaik *et al.*(2015)** Khamar Basha Shaik, Ganesan P., V. Kalist, B. S. Sathish and J. Merlin Mary Jenitha. Comparative study of skin color detection and segmentation in HSV and YCbCr color space. *Procedia Computer Science*, 57:41–48. 11, 17
- Tan *et al.*(2012)** Wei Ren Tan, Chee Seng Chan, Pratheepan Yogarajah and Joan Condell. A fusion approach for efficient human skin detection. *IEEE Transactions on Industrial Informatics*, 8(1):138–147. 9, 10, 17, 40, 42
- Vezhnevets *et al.*(2003)** Vladimir Vezhnevets, Vassili Sazonov and Alla Andreeva. A survey on pixel-based skin color detection techniques. In *Proceedings of GRAPHICON-2003*, pages 85–92. 3, 7, 8, 9, 21
- Wajnberg(2018)** Nanette Wajnberg. Skin remedies for psoriasis, eczema, dry scalp and more. <http://www.skin-remedies.com/>, 2018. Last access 18/04/2018. 1
- Yogarajah *et al.*(2011)** Pratheepan Yogarajah, Joan Condell, Kevin Curran, Paul McKevitt and Abbas Cheddad. A dynamic threshold approach for skin tone detection in colour images. *International Journal of Biometrics*, 4(1):38–55. 9, 17