Combined correlation rules to detect skin on colored images based 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, February 2018

Combined correlation rules to detect skin on colored images based on dynamic color clustering

This is the original version of the thesis prepared by the candidate Rodrigo Augusto Dias Faria, such as submitted to the Examining Committee.

Acknowledgements

Texto texto. Texto opcional.

Resumo

FARIA, R. A. D. **Regras de correlação combinadas para detectar pele em imagens coloridas baseadas em agrupamento dinâmico de cores**. Dissertação (Mestrado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2018.

A detecção 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 detecção 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, bem como um tempo de classificação custoso, que pode ser um fator decisivo para aplicações em tempo real. Neste trabalho, propomos uma melhoria de um novo método de detecção de pele baseado em regras que funciona no espaço de cores YCbCr. Nossa motivação baseia-se na hipótese de que: (1) a regra original pode ser revertida 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 cenas de 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. **Combined correlation rules to detect skin on colored images based on dynamic color clustering**. Thesis (Masters Degree) - Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2018.

Skin detection plays an important role in a wide range of image processing and computer vision applications. In short, there are three major approaches for skin detection: 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 as well as costly classification time, which can be a deal breaker for real time applications. In this work, we propose an improvement of a novel method on rule-based skin detection that works in the YCbCr color space. Our motivation is based on the hypothesis that: (1) the original rule can be reversed and, (2) human skin pixels do not appear isolated, i.e. neighborhood operations are taken in consideration. The method is a combination of some correlation rules based on these hypothesis. 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 image scenes.

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

Contents

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
HGR	Hand Gesture Recognition database
HSI	Hue, Saturation, Intensity
HSL	Hue, Saturation, Lightness
HSV	Hue, Saturation, Value
ID3	Iterative Dichotomiser 3
k-NN	k-Nearest Neighbors
LUT	Look-UP Table
NTSC	National Television System Committee
PAL	Phase Alternating Line
RBF	Radial Basis Function
RGB	Red, Green and Blue
SECAM	Sequential Color with Memory
SFA	Skin of FERET and AR Database
SVM	Support Vector Machines
UCS	Uniform Chromaticity Scale
UCI	University of California in Irvine skin/non skin dataset
YIQ	Luma, Hue and Saturation
YUV	Luma and Chrominance
WM	Wang-Mendel method

List of Symbols

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
$\mu_A(x)$	Membership function
$\mathbb{R}$	Set of real numbers
σ	Standard deviation of the Gaussian membership function
a_r	r-th attribute of a feature vector
D	Dataset
D'	Dataset in feature space
N	Dataset cardinality
K	Numbers of dataset partitions in cross-validation
α	Lagrange multiplier
ξ	Variável de relaxamento
C	Regularization parameter
Φ	Nonlinear feature expansion function
γ	Polynomial/RBF kernel parameter of a nonlinear SVM
$d(x_i, x_j)$	Distance function between x_i and x_j vectors
argmax	Arguments of maxima operator
$\delta(y, f(x_i))$	Binary function between y and $f(x_i)$ of a training sample x_i in k-NN
$H(D)$	Entropy of dataset D
$y_\oplus$	Positive samples rate
$y_\ominus$	Negative samples rate
$IG(D, a_r)$	Information gain of D partitioning by a_r attribute
$V(a_r)$	Set of all possible values of a_r
D_v	Subset of D where a_r is equal to v

List of Figures

List of Tables

Chapter 1

Introduction

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.

The problem is complex because of 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 as real time processing, robustness and accuracy are also desirable.

Skin color is a strong characteristic 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 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).

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 its sensitivity to illumination, the input image is, in general, first transformed into a color space whose luminance and chrominance components can be separate to mitigate the problem (Vezhnevets *et al.*, 2003).

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

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. Machine learning and hybrid methods require a training set, from which the decision rules are learned. Such approaches outperform the rule-based methods but require a large and representative training dataset as well as it takes a long classification time, which can be a deal breaker for real time applications (Kakumanu *et al.*, 2007).

In this work we propose an improvement of a novel method on 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 with another rule that is a reversal interpretation of the one proposed originally. Besides that, we also take in 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 image scenes.

1.1 Motivação

Há diversos objetos ou observações na natureza que não podem ser classificados com conjuntos clássicos pelo fato de que a relação de pertinência não é bem definida (Pedrycz e Gomide, 1998). Especificamente no campo da visão computacional, vários problemas reais de classificação só podem ser resolvidos quando da análise do contexto onde o problema está inserido.

Sendo assim, a motivação deste trabalho está em analisar problemas reais de visão computacional cujos conjuntos de dados possuem incerteza e imprecisão, modelando-os por meio de conjuntos *fuzzy*, explorando a capacidade desses conjuntos de expressarem transições graduais de pertinência e não pertinência.

1.2 Contributions

Enter with the contributions here.

1.3 Objetivos

O principal objetivo deste projeto de mestrado está na avaliação de conjuntos *fuzzy* na aplicação em problemas de classificação em visão computacional, estabelecendo mecanismos comparativos, por meio de indicadores quantitativos e qualitativos, para mensurar seu desempenho em relação aos conjuntos clássicos.

Além disso, a escolha do espaço de cores para a modelagem dos dados como conjuntos *fuzzy* também será analisada com o intuito de compreender sua influência nos resultados obtidos pelo classificador.

1.4 Organização do texto

Quanto à organização deste trabalho, no capítulo 3 são explicitados conceitos fundamentais sobre diferentes modelos de cores, teoria *fuzzy* e classificadores utilizados no desenvolvimento deste estudo. No capítulo 5, são expostos os resultados de experimentos preliminares com conjuntos de dados de pele de seres humanos, bem como a avaliação da influência do modelo de cores escolhido para tratar este tipo de problema de classificação. Por fim, o capítulo 6 apresenta o plano de trabalho com as atividades a serem realizadas no âmbito desta pesquisa.

Chapter 2

Related Work

There are 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 e 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 huge training 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 e Sayedi, 2016). Considering the RGB color model for instance, a histogram with 256 bins per channel - 256^3 in total - can be constructed for further counting the probability (see Eq. 2) of each possible RGB value (Jones e Rehg, 2002).

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

In Jones e 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 e Rehg (2002) realized that its overall shape could be inferred.

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

On the basis of the output of the skin detector, Jones e 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:

- 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 into 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%.

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 non-linearly transformed into YCbCr space. The skin pixels are detected using an elliptical skin model with Mahalanobis distance, assuming a Gaussian distribution of skin tone color in the color space.

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

Another typical method explicitly defines, through a number of rules, the boundaries that delimit the grouping of skin pixels in some color space (Vezhnevets *et al.*, 2003). This was the approach adopted by Kovac *et al.* (2003) in the RGB color space, obtaining a true positive rate of 90.66%. Basically, to find out the skin cluster in the RGB color space, the skin color is determined with the following rules (Kovac *et al.*, 2003):

Skin colour 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 colour at flashlight (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).

The key advantage of this method is the simplicity of skin detection rules that leads 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 the optimal non-smooth regions, (i.e. eyes and mouth). The resulting image

is subtracted from the elliptical image. As a result, there is a more uniform skin region where the thresholds are calculated.

Every single pixel in the color image is classified as skin and non-skin, based on the calculated dynamic threshold values. When the algorithm detect 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 a preprocessing step for Tan *et al.* (2012) in a strategy combining a 2-dimensional density histogram and a Gaussian model for skin color detection. The results showed an accuracy of 90.39%.

Naji *et al.* (2012) constructed an explicit classifier in the HSV color space for 4 different skin ethnic groups in parallel. After primitive segmentation, a rule-based region growth algorithm is applied, in which the output of the first layer is used as a seed, and then the final mask in other layers is constructed iteratively by neighboring skin pixels. The number of true positive pixels reported was of 96.5%.

Kawulok *et al.* (2013) combined global and local image information to construct a probability map that is used to generate the initial seed for spatial analysis of skin pixels. Seeds extracted using a local model are highly adapted to the image, which greatly improves the spatial analysis result.

Although color is not used directly in some skin detection approaches, it is one of the most decisive tools that affect the performance of algorithms (Mahmoodi e Sayedi, 2016). Despite the performance of most skin detectors is directly related to the choice of color space, Albiol *et al.* (2001) proved that the optimum performance of the skin classifiers is independent of the color space.

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. To reduce the influence of illumination, the RGB channels can be normalized and the third component can be removed, since it does not provide significant information (Kakumanu *et al.*, 2007). This characteristic led Bergasa *et al.* (2000) to construct an adaptive and unsupervised Gaussian model to segment skin into the normalized RGB color space, using only the channels r and g.

In Jayaram *et al.* (2004), a comparative study using a Gaussian approach and a histogram 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, which means using only two channels of the color space, significantly impacts the performance as well as the selection of the color space. 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 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 e Benavente, 1998). According to the results obtained, the most appropriate color spaces for skin color detection are YCgCr, YDbDr and HSV.

In Kaur e 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 e Malhotra (2015) in the HSV and YCbCr color spaces, the latter providing the best results in both.

Finally, in Brancati *et al.* (2017), 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 the skin pixels depending on the shape and size of dynamically generated skin color clusters was proposed. Geometrically, the clusters create trapezoids in the YCb and YCr subspaces that reflect in the inversely proportional behavior of the chrominance components. The method was compared with six well known rule-based methods in literature outperforming them in terms of quantitative performance evaluation parameters. Moreover, the qualitative analysis shows

that the method is very robust in critical scenarios.

Chapter 3

Theoretical Background

In this chapter, the theoretical concepts that apply to this research are stated. First, a short introduction to color models is provided 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. Thereafter, some machine learning methods are defined and explained, once they were used in the preliminary experiments of this work.

3.1 Color models

The use of color images in computer vision or image processing can be motivated by two main factors. The first refers to the powerful characteristic of color to function as a descriptor that often simplifies the identification and extraction of an object in a scene. The second is related to the ability of humans to discern thousands of tonalities and intensities compared to only a few dozen levels of gray (Gonzalez e Woods, 2002).

The visual perception of color by the human eye should not vary according to the spectral distribution of the natural light incident upon an object. In other words, the color appearance of objects remains stable under different lighting conditions. This phenomenon is known as color constancy (Gevers *et al.*, 2012).

As an example, the grass of a soccer stadium remains green throughout the day, even at dusk when, from a physical point of view, sunlight has a more reddish appearance.

The human perception of colors occurs by the activation of nerve cells that send signals to the brain about brightness, hue and saturation, which are usually the features used to distinguish one color from another (Gonzalez e Woods, 2002).

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 e 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 e 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 of the major color models, as well as their variants and main areas of application.

3.1.1 Munsell color model

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

2000).

The principle of equality of visual spacing 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 e Venetsanopoulos, 2000).

The model is represented by a cylindrical shape and it can be seen in the figure 3.1. The hue is arranged in the circular axis consisting of five base 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.

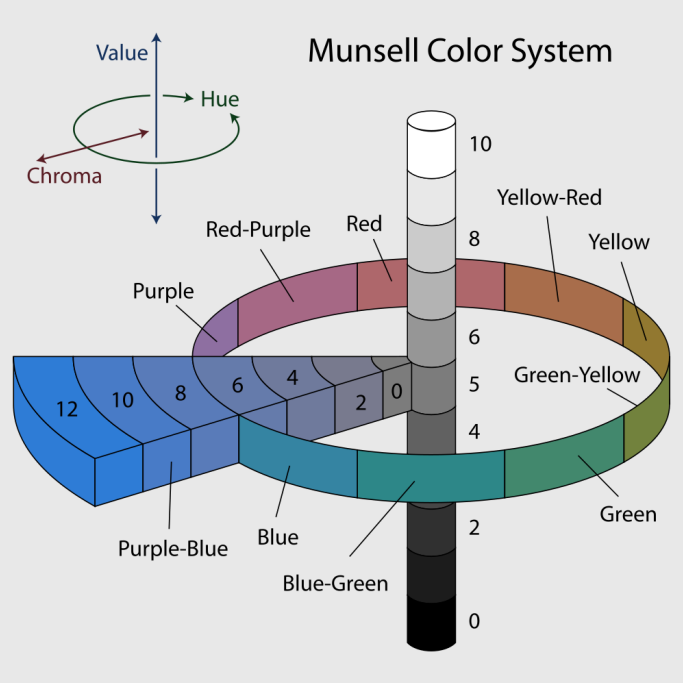


Figure 3.1: *Munsell color model represented by a cylindrical shape. The hue is arranged on the circular axis consisting of five base and five secondary colors, the saturation on the radial axis and the luminance on the vertical axis in a range varying from 0 to 10. Source: Rus (2007).*

3.1.2 CIE color model

In 1931, the CIE established the first mathematical model of 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 e Woods, 2002).

The experiment that originated the standard consisted in 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 that can be perceived by an ordinary human being. For this reason, it is considered an device independent representation (Plataniotis e 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 the equations 3.1, 3.2 and 3.3.

$$x = \frac{X}{X + Y + Z} \tag{3.1}$$

$$y = \frac{Y}{X + Y + Z} \tag{3.2}$$

$$z = \frac{Z}{X + Y + Z} \tag{3.3}$$

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 3.2.

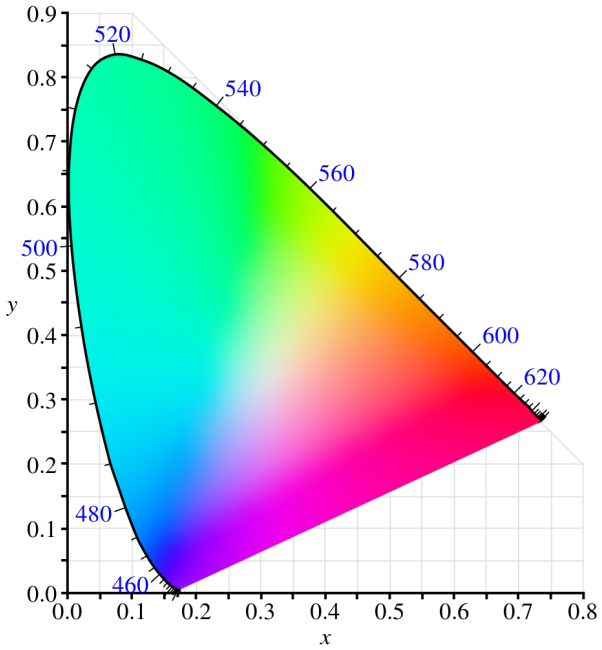


Figure 3.2: *CIE 1931 chromaticity diagram. The points representing pure colors in the electromagnetic spectrum are labeled according to their wavelengths and are located along the curve from the right end of the x-axis, corresponding to the red color, to the left end of the same axis, corresponding 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 white light, also known as white point, and serve as 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 equations 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).

Still with unsatisfactory results, 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 colors [1] in which the color stimuli

[1] Theory 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).

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).

3.1.3 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 e 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 through the unit cube defined on the axes R, G and B, as illustrated in the figure 3.3 (Plataniotis e Venetsanopoulos, 2000).

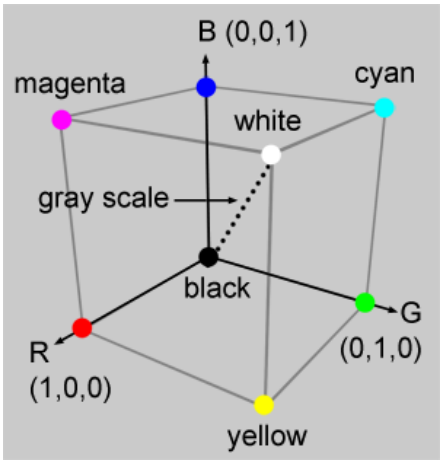


Figure 3.3: *Unit cube representing the colors of the RGB model. The origin, given by the vertex* $(0, 0, 0)$*, represents the black color. The vertex* $(1, 1, 1)$*, opposite the origin, represents the white color. The highlighted vertices on the axes represent the primary colors and the others are the 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 e Woods (2002).*

It is noteworthy that there are two ways of representing the RGB space: linear and non-linear. The above-mentioned system shows the non-linear model, whose abbreviation is $R'G'B'$, 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 e Venetsanopoulos, 2000).

3.1.4 CMY color model

The CMY model is based on the complementary primary colors Cyan, Magenta, and Yellow and, unlike RGB, is a subtractive color model 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 e Woods, 2002).

One way to get the CMY system is:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} B \\ R \\ R \end{bmatrix} + \begin{bmatrix} G \\ B \\ G \end{bmatrix} \tag{3.4}$$

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 e Woods, 2002):

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.5}$$

Likely 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 3.4 shows how the model components are combined to generate the other colors.

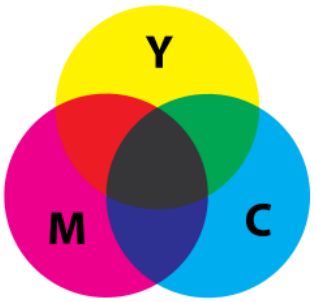


Figure 3.4: *CMY subtractive color model. It is interesting to note that the intersection of yellow with magenta generates the red color, magenta with cyan generates the blue color and cyan with yellow generates the green color. Source: Rus (2008).*

Overlapping the CMY primary colors in equal amounts to generate the black color typically creates a tint that is close to 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 e Woods, 2002).

3.1.5 Color models of the YUV family

Color models of this family is also known as orthogonal color spaces. They are able to reduce the redundancy present in RGB color channels and represent the color with statistically independent components - as independent as possible (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 analogue television transmission systems in the Phase Alternating Line (PAL) and Sequential Color with Memory (SECAM) (Pedrini e Schwartz, 2008).

The transformation of the RGB space to YUV is given by:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.6}$$

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 (hue) and Q (saturation) encode the chrominance information (Pedrini e Schwartz, 2008).

The transformation of the RGB space to YIQ is given by:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & -0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.7}$$

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 e Schwartz, 2008).

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 e Schwartz, 2008).

The transformation of the RGB space to YCbCr is given by:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.8}$$

3.1.6 Color models of the HSI family

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

In this model, as in YIQ, the intensity given by I component is decomposed from the chrominance information, represented by the hue (H) and saturation (S) (Plataniotis e Venetsanopoulos, 2000). The combination of these components results in a pyramidal structure which can be seen in figure 3.5.

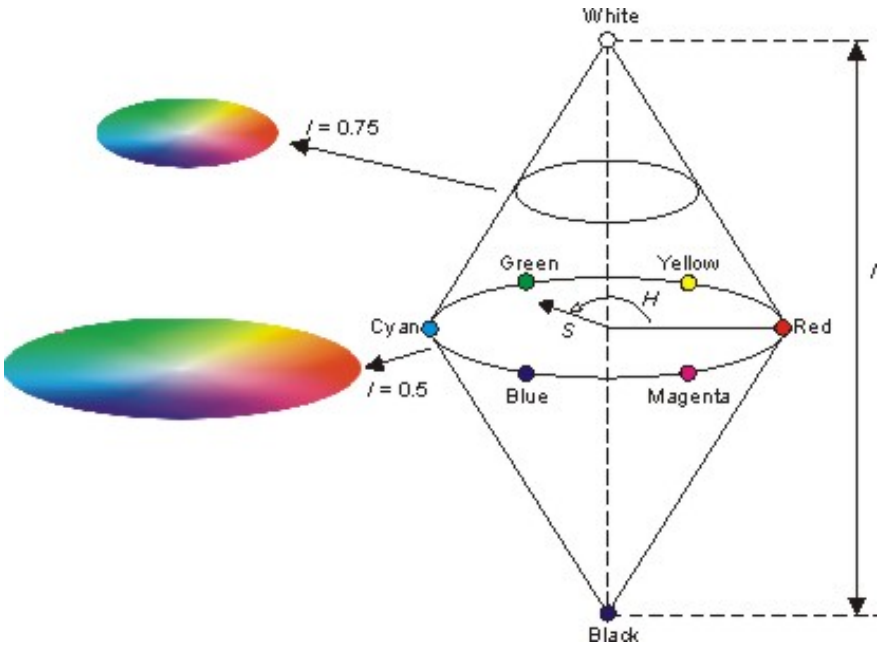


Figure 3.5: *Graphical representation of the HSI model. The hue describes the color itself, in the form of an angle θ, where $\theta \in [0, 360]$. Red is at 0 degree, yellow at 60, green at 120, and so on. The saturation component, which varies between 0 and 1, indicates how much color is polluted with white color. The intensity scale is between $[0, 1]$, where 0 means black and 1, white. Source: Ice (2016).*

The transformation of the components of the RGB space to HSI is given by the equations:

$$\begin{aligned}
\theta &= cos^{-1}\left(\frac{(R-G)+(R-B)}{2\sqrt{(R-G)^2+(R-B)(G-B)}}\right) \\
H &= \begin{cases} \theta, & \text{if } B \leq G \\ 360-\theta, & \text{otherwise} \end{cases} \\
S &= 1-\frac{3min(R,G,B)}{R+G+B} \\
I &= \frac{R+G+B}{3}
\end{aligned} \tag{3.9}$$

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 e 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 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 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 e Schwartz, 2008).

The transformation of the components of the RGB space to HSV is given by the equations:

$$\begin{aligned}
H &= \begin{cases} 60\frac{(G-B)}{M-m}, & \text{if } M = R \\ 60\frac{(B-R)}{M-m}+120, & \text{if } M = G \\ 60\frac{(R-G)}{M-m}+240, & \text{if } M = B \end{cases} \\
S &= \begin{cases} \frac{(M-m)}{M}, & \text{if } M \neq 0 \\ 0, & \text{otherwise} \end{cases} \\
V &= M
\end{aligned} \tag{3.10}$$

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 and is formed by two cones of height 1, whose bases are coincident (Pedrini e 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 scale $[0, 1]$, where 0 means black and 1, white (Pedrini e Schwartz, 2008).

The conversion of the RGB space to HSL is given by the equations:

$$
\begin{aligned}
H &= \begin{cases} 60\dfrac{(G-B)}{M-m}, & \text{if } M = R \\ 60\dfrac{(B-R)}{M-m} + 120, & \text{if } M = G \\ 60\dfrac{(R-G)}{M-m} + 240, & \text{if } M = B \end{cases} \\
S &= \begin{cases} \dfrac{(M-m)}{M+m}, & \text{if } 0 < L \leq 0,5 \\ \dfrac{(M-m)}{2-(M+m)}, & \text{if } L > 0,5 \\ 0, & \text{if } M = m \end{cases} \\
L &= \frac{M+m}{2}
\end{aligned}
\tag{3.11}
$$

where $m = min(R, G, B)$ and $M = max(R, G, B)$. The lightness V 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 3.10 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 e Schwartz, 2008).

3.2 Machine learning

Machine learning is an area that is concerned with the development of computer programs capable of improving automatically with the experience (Mitchell, 1997). This definition is closely linked to the way humans learn. Research efforts in this area have been carried out with the purpose of bringing this relationship closer.

As an example, in showing the image of a tree to a three-year-old child, she will most likely know how to recognize it, as she must have been exposed to situations where she has seen similar images, and was trained to do so answer. Then, she learned what a tree is by looking at them, not necessarily by precise mathematical definitions. In other words, the learning was done on the basis of data that are often used to obtain empirical solutions to certain problems where there is no possibility of creating an analytical solution (Abu-Mostafa *et al.*, 2012).

Similarly, an algorithm can be trained to differentiate a tree from other objects based on a set of data that has descriptions about the trees, such as height, colors, thickness, length, and so on. These descriptions are also called attributes, properties or features, and are submitted to a classifier that evaluates the presented evidences and makes a decision about the object being analyzed (Duda *et al.*, 2012). This task begins with the definition of a feature vector in a d-dimensional space, of the form (Duda *et al.*, 2012):

$$
x = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_d \end{bmatrix}
\tag{3.12}
$$

where $x \in X$ and X is the input space, that is, all of the x possible vectors.

The dataset is formed by N of these vectors and, therefore, the problem now is to partition the feature space in such a way that a decision boundary is formed (Duda *et al.*, 2012). We can then assign a specific class or label y for a given vector, where $y \in Y$ and Y is a finite set of classes which, in the binary case, is of the form $Y = \{+1, -1\}$. Figure 3.6 shows partitioning examples of

a 2-dimensional space of skin and non-skin samples from the SFA dataset [2].

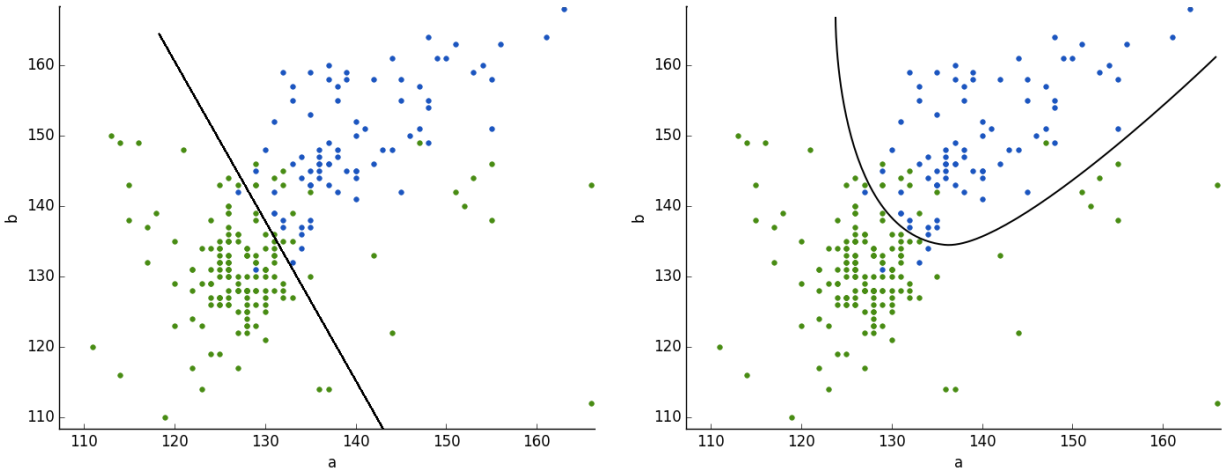


Figure 3.6: *Decision boundary in the 2-dimensional space of skin and non-skin samples of the SFA dataset. The axes represent the channels a and b of the Lab color model. The figures on the left and right show the separation of the feature space by linear and non-linear functions, respectively. It is noteworthy that, computational complexity increases significantly in cases where there are many features. Source: proposed by the author.*

Note that the decision boundary is a hypothetical function that the learning algorithm must produce based on training data. It should be as close as possible to a target function or objective function $f : X \mapsto Y$ that is unknown (Abu-Mostafa *et al.*, 2012).

This type of approach is characterized as supervised learning, that is, the training data contain explicit samples of how the correct output should be on an input vector (Abu-Mostafa *et al.*, 2012).

In unsupervised learning, also known as clustering, it is unknown what classes the training data belong to. The mission of the classifier, in this case, is to agglomerate input data into natural clusters (Duda *et al.*, 2012). Therefore, unsupervised learning can be seen as a task of finding patterns or structures spontaneously from the input data (Abu-Mostafa *et al.*, 2012).

Another approach is reinforcement learning which, just as in unsupervised learning, does not use labeled input data. The output proposed by the training is used along with a measure of its quality to improve the results of the classifier (Abu-Mostafa *et al.*, 2012).

Still on the results of the classifier, it is important to point out that learning the parameters of a target function and testing it on the same data is a fundamental error because the model would repeat the labels of the samples as soon as they were trained, implying a perfect fit of the data. However, it would not be enough to predict anything useful about new input data. This phenomenon is called over-fitting and to avoid it, the dataset is then partitioned so as to store some of the data for a subsequent test step, in order to find the best-performing estimator (Abu-Mostafa *et al.*, 2012).

Splitting the data into disjoint training and testing subsets is an effective approach when a large amount of data is available. However, when data is limited, retaining part of them for the test set further reduces the number of samples available for training (Mitchell, 1997). Therefore, another approach, known as cross-validation, can be applied so that the entire dataset is used in training and testing.

The cross-validation consists of dividing the dataset D into K disjoint subsets $D_1, D_2, \ldots, D_K$, where each subset has an approximate size of N/K. The model is then trained on each of these subsets, except one that is maintained as a validation set, in which the error measure is calculated. This process is repeated K times, in such a way that each of the subsets has the opportunity to act as the test set. By this reason, this approach is also known as *K-fold*. At the end of all K iterations, the mean error obtained by each estimator is used as the performance measure of the classifier

[2] This dataset will be detailed in the section 5.1.

(Abu-Mostafa *et al.*, 2012).

Some of the classifiers that were used in the preliminary experiments of the chapter 5 will be briefly discussed in the sections 3.2.1, 3.2.2 and 3.2.3.

3.2.1 Support vector machines

Support Vector Machines (SVM) are a machine learning technique based on the Statistical Learning Theory (Vapnik, 2013), whose objective was to solve problems of classification of patterns. In practice, an SVM has the ability to generate a hyperplane or set of hyperplanes in a space of high or infinite dimensionality, which can be used for classification, regression, or other tasks (Duda *et al.*, 2012).

Derived from the technique name itself, the support vectors are the training set samples that define the optimal hyperplane and are the most difficult patterns to classify (Duda *et al.*, 2012). Intuitively, they are the most relevant patterns for the classification task, since a change in these vectors directly implies in the result of the optimal hyperplane [3].

Let the training dataset with N samples of the form:

$$D = (x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N) \tag{3.13}$$

where each x_i is a d-dimensional vector of the form given by 3.12, $i = 1, 2, \ldots, N$, $y_i \in Y$ and $Y = \{+1, -1\}$. Therefore, the training dataset contain N observations with its respective labels.

Supposing that D is linearly separable, one can separate the data by means of a hyperplane using a classifier, also linear, defined by the equation (Lorena e Carvalho, 2003):

$$w \cdot x + b = 0 \tag{3.14}$$

where $w \cdot x$ is a dot product, w is the normal vector to the hyperplane and b is a bias term. The parameter $\frac{b}{||w||}$ determines the offset of the hyperplane in relation to the origin.

From this definition, two other parallel hyperplanes to the optimal hyperplane can be obtained, according to the equations in 3.15, so that a delimited region, known as margin, rises between them.

$$\begin{cases} w \cdot x + b = +1 \\ w \cdot x + b = -1 \end{cases} \tag{3.15}$$

In addition, some constraints are defined to avoid that there are no points between $w \cdot x + b = 0$ and $w \cdot x + b = \pm 1$. Formally, we have (Lorena e Carvalho, 2003):

$$\begin{cases} w \cdot x_i + b \geq +1, & \text{if } y_i = +1 \\ w \cdot x_i + b \leq -1, & \text{if } y_i = -1 \end{cases} \tag{3.16}$$

or, similarly:

$$y_i(w \cdot x_i + b) \geq 1 \tag{3.17}$$

According to Campbell (2000), in the system given by equation 3.16, it is assumed that the margin is always greater than the distance between $w \cdot x + b = 0$ and $|w \cdot x + b = 1|$ and, for this reason, SVMs of this nature are usually referred as hard-margin SVMs. Figure 3.7 shows the graphical representation of these concepts.

Geometrically, the distance between the two parallel hyperplanes to the optimal hyperplane is $\frac{2}{||w||}$. Consequently, one can deduce that the distance between the optimal hyperplane and $w \cdot x + b = \pm 1$ is $\frac{1}{||w||}$. Then, the minimization of $||w||$ maximizes the margin and, thus, we have an optimization problem in which we want to minimize $||w||^2$ subject to the constraints in 3.17

[3] The optimal separation hyperplane of an SVM is that of a class of hyperplanes with the highest separation margin between the two training sets (Cortes e Vapnik, 1995).

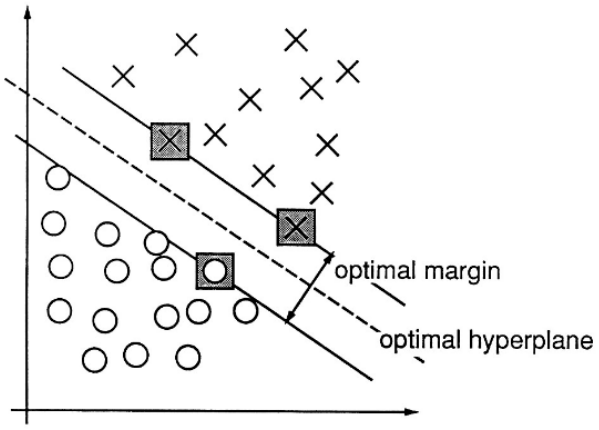


Figure 3.7: *Example of a problem of separable binary classes in a 2-dimensional space. The support vectors, marked with gray squares, define the margin with highest separation between the two classes. Source: Cortes e Vapnik (1995).*

(Lorena e Carvalho, 2003). Therefore, optimal w and b that solve this problem define the classifier and can be obtained by Lagrange multipliers (Campbell, 2000):

$$\text{Maximize: } \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \tag{3.18}$$

$$\text{Subject to: } \begin{cases} \alpha_i \geq 0 \\ \sum_{i=1}^{N} \alpha_i y_i = 0 \end{cases} \tag{3.19}$$

where α are the Lagrange multipliers.

It is important to note that this type of SVM succeeds in linearly separable training datasets. In cases where the data are non-linearly separable, some classification errors are allowed for the training set, by the inclusion of relaxation variables (Lorena e Carvalho, 2003). These changes were produced by Cortes e Vapnik (1995) and are part of a technique called margin smoothing, and therefore, SVMs of this family are also called soft-margin SVMs. So, the optimization problem becomes (Lorena e Carvalho, 2003):

$$\text{Minimize: } ||w||^2 + C \sum_{i=1}^{N} \xi_i \tag{3.20}$$

$$\text{Subject to: } \begin{cases} \xi_i \geq 0 \\ y_i(w \cdot x_i + b) \geq 1 - \xi_i \end{cases} \tag{3.21}$$

where C is an regularization constant, empirically determined, that imposes a different weight for training in relation to generalization. Thus, the problem can be solved in the same way with the equation in 3.18, but with some changes in the constraints (Lorena e Carvalho, 2003):

$$\text{Subject to: } \begin{cases} 0 \leq \alpha_i \leq C \\ \sum_{i=1}^{N} \alpha_i y_i = 0 \end{cases} \tag{3.22}$$

In general, most classification problems prevent linear classifiers from being used because it is not possible to obtain satisfactory results in the partitioning of training data by a hyperplane. However, linear SVMs can be extended to deal with this situation by mapping the input space

into a higher-dimensional feature space, typically much larger than the original space (Duda *et al.*, 2012), of the form:

$$D' = (\Phi(x_1), y_1), (\Phi(x_2), y_2), \ldots, (\Phi(x_N), y_N) \tag{3.23}$$

The proper choice of a function Φ makes the training set linearly separable in the transformed space (Lorena e Carvalho, 2003). The shape of the optimal hyperplane is now defined by:

$$w \cdot \Phi(x) + b = 0 \tag{3.24}$$

Therefore, the concepts of support vectors, margin, parallel hyperplanes, and hence, optimization problem to find the solution for the w and b parameters apply here in a similar manner. Formally, the optimization problem can be solved as (Lorena e Carvalho, 2003):

$$\text{Maximize: } \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \Phi(x_i) \cdot \Phi(x_j) \tag{3.25}$$

subject to the same constrains given by 3.22.

Now, there is a need to define as the inner product $\Phi(x_i) \cdot \Phi(x_j)$ between two vectors $x_i, x_j \in D$ is realized, whose answer lies in the introduction of the concept of *kernels* (Lorena e Carvalho, 2003).

Kernels are functions that have the purpose of projecting the input vectors in a features space with exponential or infinite number of dimensions (Shawe-Taylor e Cristianini, 2004), of the form:

$$k(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \tag{3.26}$$

Thus, it is possible to apply a *kernel* as described in the equation 3.26 in the SVM optimization step, efficiently calculating the internal product from the input data, without even explicitly computing the mapping of the Φ function (Shawe-Taylor e Cristianini, 2004).

Some of the most commonly used *kernels* are (Lorena e Carvalho, 2003):

(i) Linear *kernel*

$$k(x_i, x_j) = (x_i \cdot x_j) \tag{3.27}$$

The linear *kernel* is the simplest of the *kernel* functions and it is given by the internal product of two vectors x_i and x_j.

(ii) Polynomial *kernel*

$$k(x_i, x_j) = (\gamma x_i \cdot x_j + r)^g \tag{3.28}$$

Where g is the degree of the polynomial and r is a constant term. For this *kernel*, the Φ mappings are also polynomial functions, so the complexity is proportional to the choice of g (Lorena e Carvalho, 2003).

(iii) Gaussian *kernel*

$$k(x_i, x_j) = \exp\big(-\gamma ||x_i - x_j||^2\big) \tag{3.29}$$

Also known as Radial Basis Function (RBF), the Gaussian *kernel* corresponds to a features space of infinite dimension (Lorena e Carvalho, 2003).

For the RBF and polynomial *kernels*, the γ parameter can be seen as the inverse of the radius of influence of samples selected by the model as support vectors (Pedregosa *et al.*, 2011).

3.2.2 *k*-Nearest Neighbors

The k-Nearest Neighbors (k-NN) is an algorithm based on instances, which means that the target function is not learned based on training samples; they are simply stored, so that the relation

between a new sample and the stored training samples is evaluated, and a target function is then assigned to it (Mitchell, 1997).

As the name suggests, k-NN classifies a new point x by assigning it the highest frequency class of the k closest samples, that is, the decision of the class of x is made by majority of the votes of the k nearest neighbors, and for this reason, it is interesting that the choice of k be an odd number to avoid draws (Duda *et al.*, 2012).

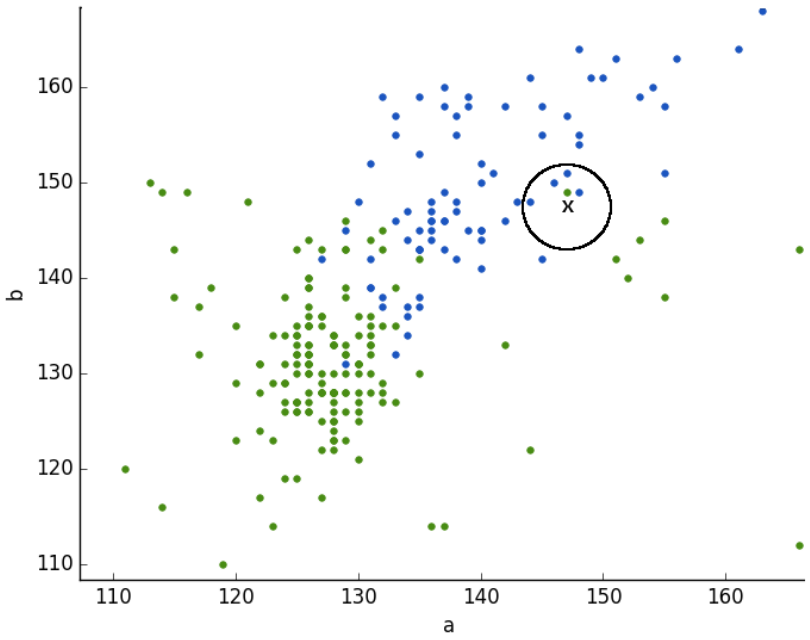


Figure 3.8: *Example of a k-NN application in the classification task in a 2-dimensional space. The algorithm evaluates the k samples near x, creating a spherical region, and labels x with the highest frequency class. In this case, $k = 5$ and the class assigned to x must be the same as the blue points. Source: proposed by the author.*

Let D be the training dataset with N samples as given in 3.13. Therefore, the distance between two samples x_i and x_j can be obtained in terms of the Euclidean distance, defined as:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{d} (a_r(x_i) - a_r(x_j))^2} \tag{3.30}$$

where a_r refers to the rth attribute of the input vector x and $i, j = 1, 2, \ldots, N$. Other distance metrics can be attributed to $d(x_i, x_j)$, such as, Manhattan, Chebyshev, and Minkowski (Duda *et al.*, 2012).

In order to classify a new sample x_q, we take $x_1, \ldots, x_k$ instances of the training set, whose distances given by 3.30, are of the k points nearest to x_q. Thus, the function that estimates the class of x_q is given by (Mitchell, 1997):

$$g(x_q) = \underset{y \in Y}{\operatorname{argmax}} \sum_{i=1}^{k} \delta(y, f(x_i)) \tag{3.31}$$

where

$$\delta(y, f(x_i)) = \begin{cases} 1, & \text{if } y = f(x_i) \\ 0, & \text{otherwise} \end{cases} \tag{3.32}$$

It is important to note that $f(x_i)$ is known, that is, it is the class of the sample x_i of the training set.

An obvious variation of the function given in equation 3.31 is the assignment of weights of each neighbor k, according to its distance, to a point x_q being classified (Mitchell, 1997). This variation

implies that points closer to x_q have more influence on its labeling. Formally, we have (Mitchell, 1997):

$$g(x_q) = \underset{y \in Y}{\operatorname{argmax}} \sum_{i=1}^{k} w_i \delta(y, f(x_i)) \tag{3.33}$$

where

$$w_i = \frac{1}{d(x_q, x_i)^2} \tag{3.34}$$

In the case where $d(x_q, x_i) = 0$, that is, x_q and x_i are exactly the same coordinates, $g(x_q)$ can assume the same value of $f(x_i)$. If there are other training samples x_i with the same characteristic, then x_q can assume the class of the one who is the most of them (Mitchell, 1997).

3.2.3 Decision tree

Decision tree is a method for approximation of discrete target functions, in which the learned function is represented by a decision tree or by a set of rules of type *If-Then* that are easy to interpret. It is one of the most popular learning techniques of inductive inference [4] (Mitchell, 1997).

The samples of a dataset are classified by a decision tree by an iterative process where an attribute (node) is chosen as root to some leaf node, where the class is assigned to the sample. Each branch starting from a node represents one of the possible values of a given attribute (Mitchell, 1997).

There is a family of decision tree algorithms. One of them is the Iterative Dichotomiser 3 (ID3) proposed by Quinlan (1986). The ID3 evaluates each attribute through a statistical test to determine how well it classifies the training samples. The best attribute is selected as the root node of the tree. A descending branch of the root node is created for each possible value of this attribute, and the training samples are classified into the appropriate descendent node. This process is then repeated recursively using the samples associated with each descendent node. This is a greedy search algorithm, since it does not regress to reconsider previous choices (Mitchell, 1997). The algorithm stops when the subset of samples associated with a node is of the same class or when there is no statistical relevance for a new partition (Quinlan, 1986).

To measure the impurity of a partition, ID3 uses the concept of entropy, formally (Quinlan, 1986):

$$H(D) = -y_{\oplus} log_2 y_{\oplus} - y_{\ominus} log_2 y_{\ominus} \tag{3.35}$$

where D is the training dataset with N samples of the form presented in 3.13, $y_{\oplus}$ and $y_{\ominus}$ is the proportion of positive and negative samples of D, respectively. It is important to note that the entropy is 0 when all samples of D belong to the same class, 1 when D contains an equal number of positive and negative samples, and a value between 0 and 1 in all other cases (Mitchell, 1997).

Given the entropy as a measure of impurity of a set of training samples, one can define the statistical test, known as information gain, that measures the effectiveness of an attribute in the classification of training data, formally (Quinlan, 1986):

$$IG(D, a_r) = H(D) - \sum_{v \in V(a_r)} \frac{|D_v|}{|D|} H(D_v) \tag{3.36}$$

where $V(a_r)$ is the set of all possible values of the attribute a_r, D_v is the subset of D in which the a_r attribute is equal to v.

It is worth emphasizing that the first term of the equation given in 3.36 is the entropy of the original set D and the second term is the expected value of the entropy after D is partitioned using

[4] The task of induction is to develop a classification rule that can determine the class of any object from the values of its attributes (Quinlan, 1986).

the attribute a_r (Mitchell, 1997). Another important aspect is that the algorithm starts with the original D set, which is replaced by D_v as the recursion deepens.

Named as C4.5, Quinlan (1993) extended ID3 to enable the use of continuous attributes, attributes with missing data, and improved computational efficiency. In addition, this version takes care of issues such as how deep the tree should be to prevent training data from being perfectly classified, that is, when the training set is partitioned until each subset contains only samples of a single class - the over-fitting phenomenon. The strategy adopted by Quinlan (1993) was to prune the tree after the generation of the adjusted tree. This process, although slower than the previous ID3 proposal, made the algorithm more reliable and with greater generalization capacity.

Chapter 4

Proposed Solution

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

4.1 Original Method

In order to describe the proposed extensions, we will first transcribe 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 relation between the chrominance components Cb and Cr in these subspaces (Brancati *et al.*, 2017).

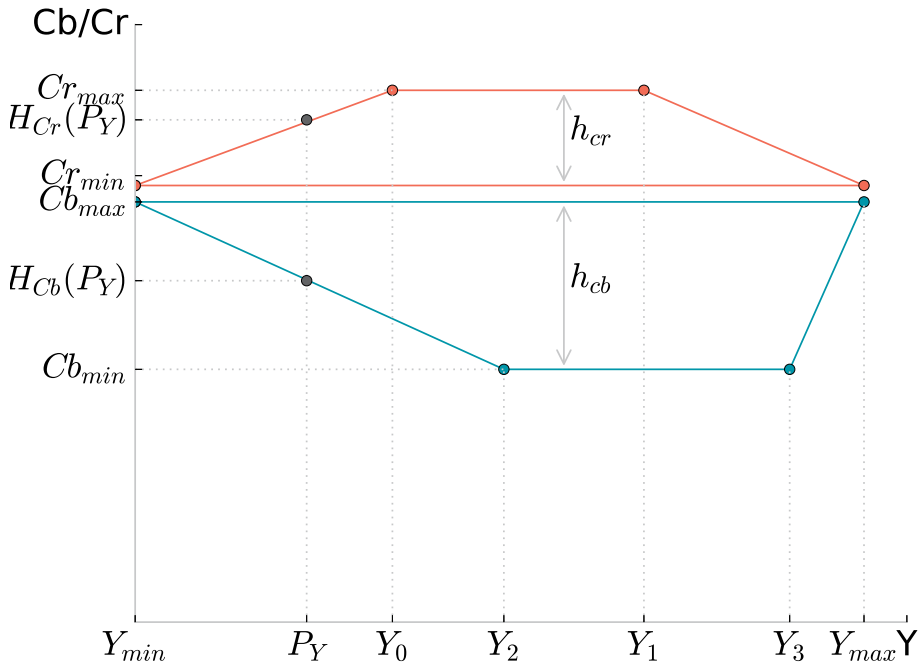


Figure 4.1: *Graphical representation of the trapezoids as well as the parameters* $Y_{min} = 0$, $Y_{max} = 255$, Y_0, Y_1, Y_2, Y_3, Cr_{min}, Cr_{max}, Cb_{min}, Cb_{max}, h_{Cr}, h_{Cb}, $H_{Cr}(P_Y)$, $H_{Cb}(P_Y)$. *Source: adapted from (Brancati et al., 2017).*

The base of the trapezoids T_{YCr} and T_{YCb} (Fig. 4.1) are given by (Y_{min}, Cr_{min}) and (Y_{min}, Cb_{max}) in the YCr and YCb subspaces, respectively. The values $Cr_{min} = 133$, $Cb_{max} = 128$ were selected

according to Chai e 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 falls in the ranges [133, 173] and [77, 127], respectively, regardless the skin color variation in different races.

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%[1] of pixels in the image. The same applies to Cb_{min}, taking 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 $Cr = Cr_{max}$. A similar procedure is used to find the values of the shorter base of the other trapezoid, Y_2 and Y_3 (see Fig. 4.2 for an example).

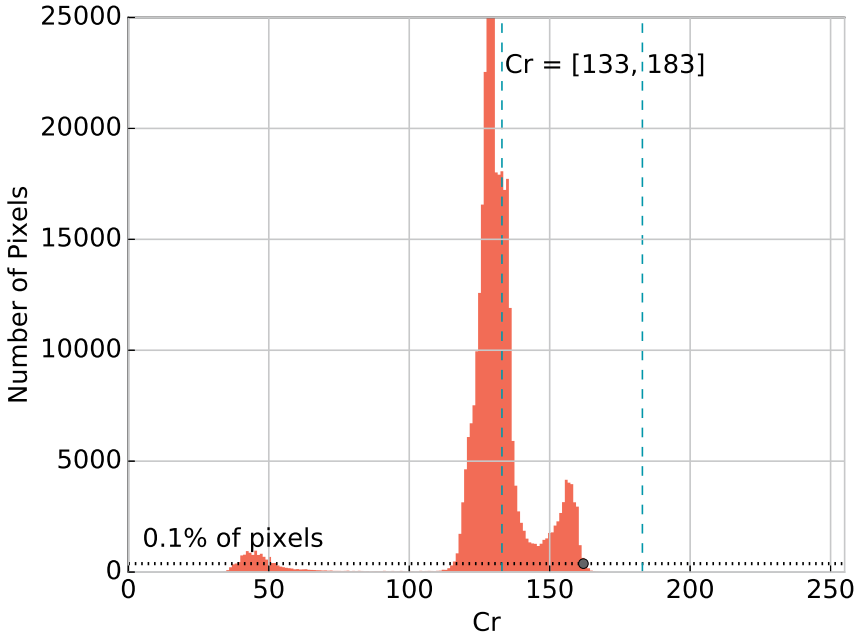


Figure 4.2: *Computation of $Cr_{max} = 162$ based on Cr values histogram of a 724 x 526 image.*

The correlation rules between the chrominance components P_{Cr} and P_{Cb} of a pixel P are defined as:

- 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 equations must hold:

$$P_{Cr} - P_{Cb} \geq I_P \tag{4.1}$$

$$|P_{Cb} - P_{Cb_s}| \leq J_P \tag{4.2}$$

The estimated value P_{Cb_s} is given by [2]:

$$P_{Cb_s} = Cb_{max} - dP_{Cb_s} \tag{4.3}$$

[1] In Brancati *et al.* (2017) this rate is reported to be equal to 10%. However, in the distributed source code we found the value 0.1%, that we are using in the experiments.

[2] 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.

where [3]:

$$dP_{Cb_s} = \alpha \cdot dP_{Cr} \tag{4.4}$$
$$dP_{Cr} = P_{Cr} - Cr_{min} \tag{4.5}$$

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:

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

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

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 the base of the trapezoids:

$$\Delta_{Cr}(P_Y) = H_{Cr}(P_Y) - Cr_{min} \tag{4.8}$$
$$\Delta_{Cb}(P_Y) = Cb_{max} - H_{Cb}(P_Y) \tag{4.9}$$

Next, the distances are normalized with respect to the difference in size of the trapezoids:

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

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

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:

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

Finally, I_P and J_P are given by:

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

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

where:

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

[3] dP_{Cr} is the distance between (P_Y, P_{Cr}) and (P_Y, Cr_{min}) points in the YCr subspace.

4.2 Extended Method

The hypothesis defined 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 to put them in terms of the estimated value of P_{Cr}, that we denote as P_{Cr_s} [4]:

$$P_{Cr_s} = dP_{Cr_s} + Cr_{min} \tag{4.16}$$

where [5]:

$$dP_{Cr_s} = \alpha \cdot dP_{Cb} \tag{4.17}$$

$$dP_{Cb} = Cb_{max} - P_{Cb} \tag{4.18}$$

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

$$I_P^{'} = sf \cdot [(\Delta_{Cr}^{'}(P_Y) - dP_{Cr_s}) + (\Delta_{Cb}^{'}(P_Y) - dP_{Cb})] \tag{4.19}$$

$$J_P^{'} = dP_{Cr_s} \cdot \frac{dP_{Cb} + dP_{Cr_s}}{\Delta_{Cb}^{'}(P_Y) + \Delta_{Cr}^{'}(P_Y)} \tag{4.20}$$

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

$$P_{Cr} - P_{Cb} \geq I_P^{'} \tag{4.21}$$

$$|P_{Cr} - P_{Cr_s}| \leq J_P^{'} \tag{4.22}$$

Doing this simple extension, we are now able 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.

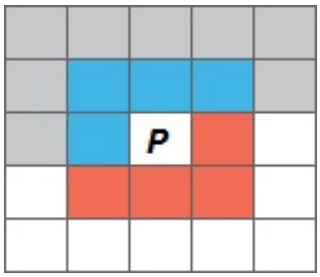


Figure 4.3: *Neighbors evaluation with respect to P. If the image is scanned in raster order, $N_8^-(P)$ is the set of points that can be reached before P in a 8-connected neighborhood.*

4.3 Neighborhood Extended Method

Both methods presented in Sec. 4.1 and 4.2 can be applied to detect skin pixels, either separated or in a conjunction rule. However, skin pixels do not usually appear isolated and we can improve the method using neighbor pixels information, when evaluating a pixel P, in order to decide if P

[4] 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.

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

represents human skin, or not. Let $N_8^-(P)$ the 8-connected neighbors of P that can be reached before P when scanning the image in raster order (blue points in Fig. 4.3).

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 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.

Chapter 5

Evaluation

In this section we present some experimental evaluations of the proposed extensions along with the original method in three widely known datasets: SFA, Pratheepan and HGR. In addition, a brief definition of the evaluation metrics used is shown for the sake of clarity.

5.1 Datasets

5.1.1 UCI

Named UCI in this work, this dataset was proposed by Bhatt e Dhall (2012) and obtained from the machine learning repository of the University of California in Irvine (Lichman, 2013). The dataset consists of images of various skin and non-skin textures obtained from thousands of arbitrary faces images of different ages, sex and races (Minear e Park, 2004; Phillips *et al.*, 1996).

The UCI contains 245,057 samples, composed of 3 attributes that constitute the input vector $x = [x_1, x_2, x_3]$, $x \in \mathbb{R}^d$, where d is the space dimension which represents, respectively, blue (B), green (G) and red (R) channels of the RGB color model. In addition, a fourth column determines the class to which the sample x belongs, denoted by y, where $y \in Y$ and $Y = \{+1, -1\}$. In other words, each sample is an RGB pixel with a given label.

The table 5.1 exemplifies a short excerpt from the UCI database. It is worth mentioning that of the 245,057 samples, 194,198 are non-skin pixels and 50,859 pixels with different skin tones. In addition, the images that were used to extract the dataset were not made available by the authors.

B	G	R	Label
74	85	123	1
207	215	255	1
74	82	122	1
202	211	255	1
54	72	125	1
...	...	...	...
166	164	116	-1
148	150	91	-1
29	26	5	-1
167	166	115	-1
180	177	133	-1

Table 5.1: *Excerpt with samples from the UCI dataset. Each of the first three columns represents a pixel channel of the RGB color space ranging from 0 to 255. The fourth column is the label assigned to the sample, which can assume +1 if it is skin and -1, otherwise. Originally, the class representing a non-skin pixel had value 2, replaced by -1 for compliance with the experiments.*

Since the data are points in the RGB space, it is possible to plot it for a better interpretation of them, as shown in the figure 5.1.

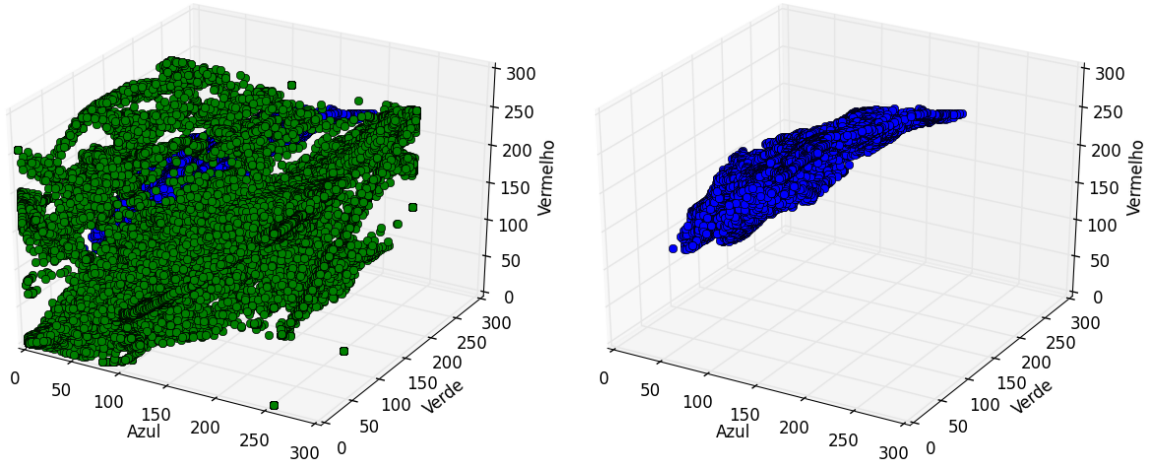


Figure 5.1: *3-dimensional view of the RGB channels of the UCI dataset. The blue points are skin samples and the green ones are non-skin. On the left are all samples of the dataset; to the right only the skin samples. Source: proposed by the author.*

5.1.2 SFA

SFA, the name of the dataset proposed by Casati *et al.* (2013), stands for Skin of FERET and AR Database. The SFA is a set of images of frontal faces obtained from two other color image databases: the FERET, created by Phillips *et al.* (1996), and the AR proposed by Martínez e Benavente (1998), which provided 876 and 242 images each, respectively. It is important to notice that AR images have a white background and small variations of skin color. In other words, the environment is more controlled than the images in FERET Casati *et al.* (2013). Figure 5.3 shows some of the 1,118 samples available.

Casati *et al.* (2013) also extracted different window patches of 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 sample is a window of size $n \times n$, where n is odd, with a central pixel, from which other sample sizes have been created, ranging from 1×1 to 35×35, as can be seen in figure 5.2.

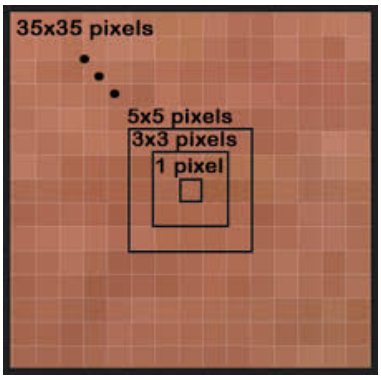


Figure 5.2: *Structure of the windows that form the SFA samples. In total, there are 3,354 skin samples and 5,590 non-skin samples for each window size. Source: Casati et al. (2013).*

[1] Ground 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.

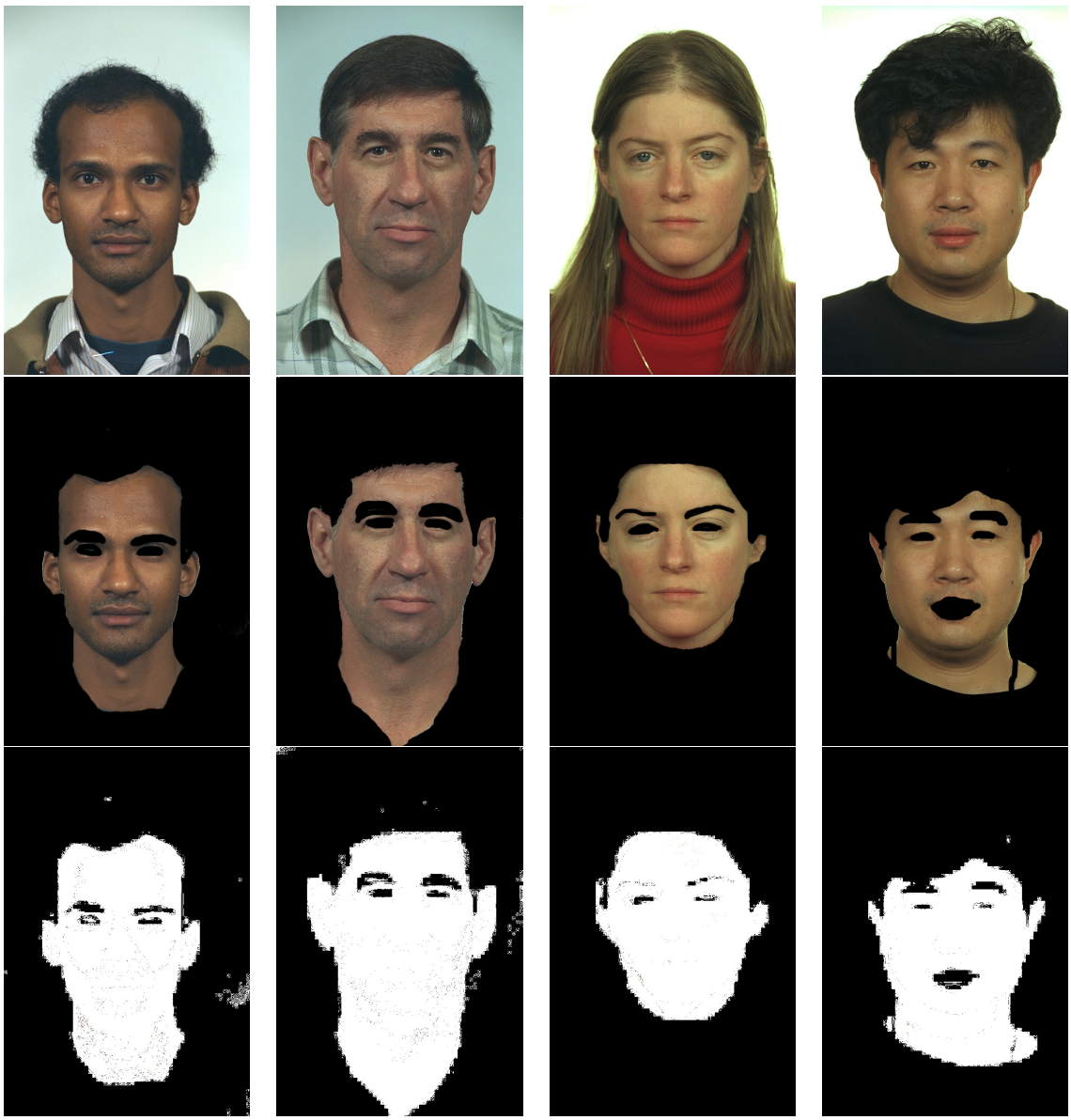


Figure 5.3: *Examples of SFA face image database. First row are the original images and the second contain the colored ground truth with the skin color pixels annotated manually. The black color RGB = (0, 0, 0) was assigned to all pixels in the background. In the third row we have the binary ground truth images. We generated these samples based on the colored ground truth, by creating a mask, assigning (255, 255, 255) for the pixels which were not background (0, 0, 0). 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).*

5.1.3 Pratheepan

The images in the Pratheepan dataset were downloaded randomly from Google for human skin detection search. 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 5.4 shows some of the 78 samples available.

Figure 5.4: *Examples of Pratheepan skin dataset. At the top is the original image and at the bottom the ground truth with the skin color pixels annotated. Here, the ground truth are binary images, where the black color RGB = (0, 0, 0) was assigned to all pixels in the background. Source: Tan et al. (2012).*

5.1.4 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 lightening. In addition to original and ground truth binary skin mask images, it includes hand feature points location in separate files. Figure 5.5 shows some of the 1,558 samples available (Grzejszczak *et al.*, 2016; Kawulok *et al.*, 2014; Nalepa e Kawulok, 2014).

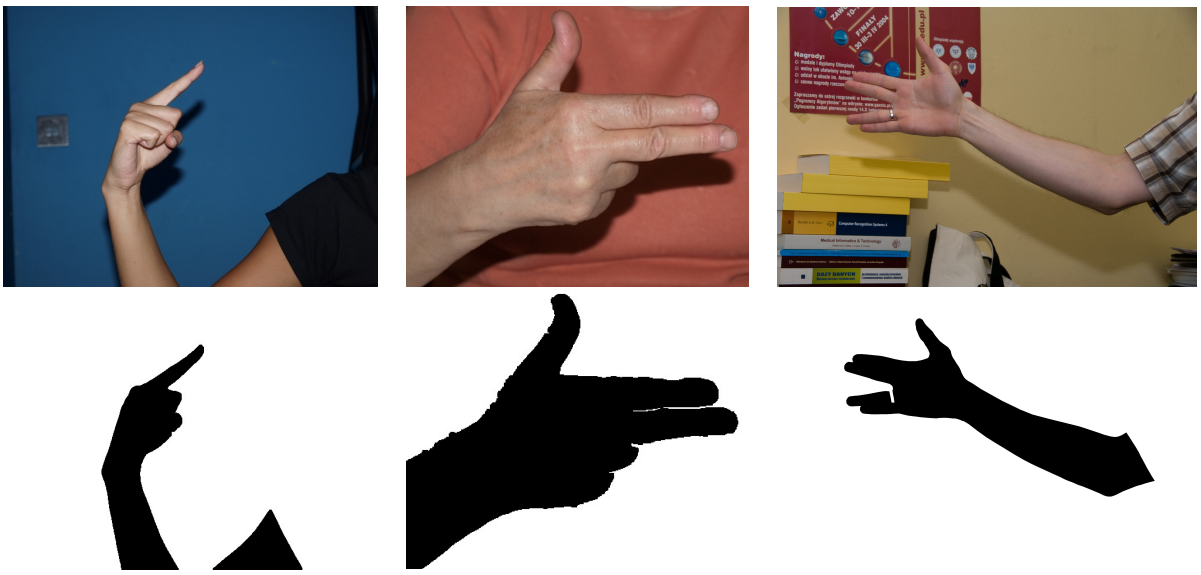


Figure 5.5: *Examples of HGR skin dataset. At the top is the original image and at the bottom the ground truth with the skin color pixels annotated. Differently from Pratheepan and SFA, the ground truth is also binary images, but the black color RGB = (0, 0, 0) was assigned to all pixels when they represent skin patches. Source: Grzejszczak et al. (2016); Kawulok et al. (2014); Nalepa e Kawulok (2014).*

The images within were acquired in three different series. A set of 899 was captured in uncontrolled background and lighting. A small set of 85 was obtained in gray (44) and (41) uncontrolled background; the lighting was uniform. The third group contains 574 images in controlled back-

ground (green tone), using uniform lighting conditions (Grzejszczak *et al.*, 2016; Kawulok *et al.*, 2014; Nalepa e Kawulok, 2014).

5.1.5 Compaq

Compaq can be considered as 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. The ground truth images are poorly annotated on the basis of an automatic software tool (Mahmoodi e Sayedi, 2016).

It is worth mentioning that this database is no longer available and we had obtained a copy of it by contacting the authors. We also had to fix some few images due lack of ground truth or files corrupted. The final amount of images with skin used in the experiments is 4,669. Figure 5.6 shows some of the 4,669 samples available used in the experiments (Jones e Rehg, 2002).

Figure 5.6: *Examples of Compaq skin/non-skin dataset. At the top is the original image and at the bottom the ground truth with the skin color pixels annotated. Here, the ground truth are binary images, where the black color RGB = (0, 0, 0) was assigned to all pixels in the background. Source: Jones e Rehg (2002).*

5.2 Evaluation measures

Precision, Recall, Specificity and *F-measure* have been used as evaluation metrics. They are the same used in Brancati *et al.* (2017) to compare the performance with state-of-the-art methods. They are also widely used by the scientific community. These metrics are given by the following formulas:

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

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

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

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

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 5.2).

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

Table 5.2: *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 Machine learning experiments

Preliminary experiments were carried out using machine learning techniques in order to establish a bottom line for comparison with the proposed method, mainly in the execution time point of view.

The first experiment was carried out with k-NN and SVM, available in the scikit-learn package (Pedregosa *et al.*, 2011). The color space originally used was RGB, as cited in the description of datasets in 5.1. In both cases, the chosen cross-validation strategy was 10-fold, which is a common choice of this approach in practice (Abu-Mostafa *et al.*, 2012). In addition, the technique of grid search of scikit-learn was also used with the objective of finding the most suitable parameters for each classifier.

The grid search is used in scikit-learn to find the optimal parameters of a classifier when they can not be learned by the estimator, such as the *kernel* and γ in the SVM or number of neighbors of k-NN (Pedregosa *et al.*, 2011). The parameter tables used in the training of SVM and k-NN can be seen in 5.3 and 5.4. The parameters of each line are combined in an attempt to find the optimal estimator. For example, a choice in SVM training would be *kernel* = rbf, C = 100, γ = 1e-4. All possible combinations are exploited, with the best of them being returned (Pedregosa *et al.*, 2011).

kernel	***C***			***gamma***			***degree***	
rbf	1	10	100	1e-3	1e-4	1e-5		
poly	1	10	100		1e-4	1e-5	3	4
linear	1	10	100					

Table 5.3: *Grid search table of the parameters of the optimal estimator in the SVM. The kernel column refers to the kernels used in training that are Gaussian, polynomial and linear, respectively. C is a regularization parameter that tells SVM the amount of error allowed during the training. Gamma is a parameter used only in Gaussian and polynomial kernels, as cited in 3.2.1. Degree is the degree of the polynomial; used only in the polynomial kernel (Pedregosa et al., 2011).*

The results of this experiment can be seen in the table 5.5. The dataset used was UCI. It is noteworthy that the training was performed splitting the data with 30% randomly separated for test in both classifiers.

As can be seen in the table 5.5, both classifiers had very high quality measures in the UCI dataset, close to 100%, which is probably an over-fitting situation. One possible root cause is the splitting of training and test data subsets. We observed that there are many replicates among the

samples, so it is possible that samples already seen by the classifier during the training are used in the test step.

In fact, once we used a seed to generate the subsets, we applied the same strategy to split the data and count how many samples of the test subset were seen in the training subset as shown in figure 5.7. Based on the distribution of the splitting and the results of the measures given in table 5.5, we can say that UCI dataset is not suitable for this application.

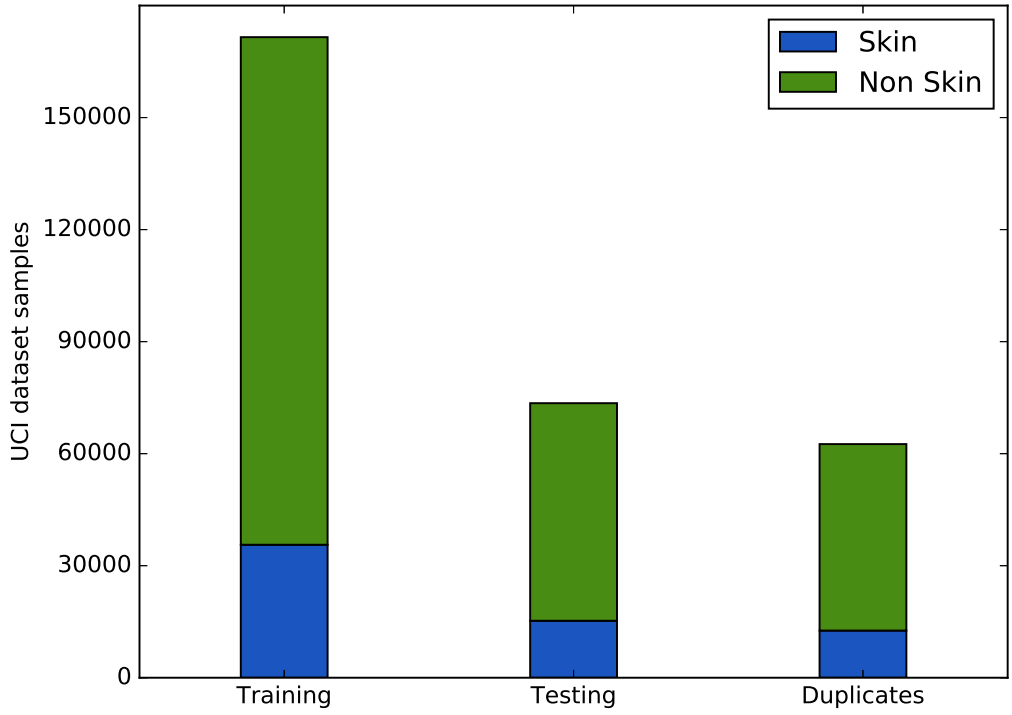


Figure 5.7: *UCI samples distribution after splitting the dataset in training and testing subsets. We used the scikit-learn package function for splitting, where 30% is the size of test set. The right-most bar, entitled as duplicates, shows how many samples of the test set was seen in training set. In the order of 83% for skin samples and 86% for non skin samples. Source: proposed by the author.*

For the benefit of the doubt, we submitted the learned function given by those classifiers in real images of Pratheepan and SFA datasets. This experiment can help us to understand the ability of

n_ neighbors							***weights***		***algorithm***
3	5	9	15	25	31	35	distance	uniform	auto

Table 5.4: *Grid search table of the parameters of the optimal estimator in k-NN. The n_ neighbors column refers to the number of neighbors considered in the training. Weights is the weight function used in the prediction, where uniform indicates that the points have equal weights and distance indicates that the inverse of the distance is applied in the classification, as cited in 3.34. The third column indicates which algorithm should be used; auto means that the algorithm will be decided based on the data (Pedregosa et al., 2011).*

Classifier	**Color model**	***Precision***	***Recall***	***F-measure***
k-NN	RGB	0,9995	0,9995	0,9995
SVM	RGB	0,9995	0,9995	0,9995

Table 5.5: *Results of the experiments with k-NN and SVM in the UCI dataset. The optimal k-NN parameters found during the training in UCI were* $n\_neighbors = 3$, $weights = uniform$. *In the case of SVM, the optimal parameters found in UCI training were* $kernel = rbf$, $C = 100$ *and* $gamma = 1e-3$.

the learned model to generalize for upcoming non seen samples.

Dataset	**Classifier**	**Precision**	***Recall***	***Specificity***	***F-measure***
SFA	k-NN	0.9322	0.5895	0.9835	0.7223
	SVM	0.9079	0.4839	0.9838	0.6314
Pratheepan	k-NN	0.5930	0.7719	0.8568	0.6707
	SVM	0.6179	0.7259	0.8841	0.6675

Table 5.6: *Results of the experiments with k-NN and SVM in the SFA and Pratheepan images datasets. Despite some of the measures are quite good, we can see that they are far way from the ones given during the training in table 5.5, which says that the learned models do not have a good generalization.*

5.4 Rule-based experiments

In this section we present some experimental evaluations of the proposed extensions described in sections 4.2 and 4.3, 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 5.7 shows quantitative result metrics of the experiments. Column 1 refers to the dataset used. Column 2 refers to the method being experimented: Original for the original hypothesis; Reverse refers to the reverse hypothesis with respect to P_{Cr_s} parameter; Combined refers to the combination of both of the former methods (see Sec. 4.2); Neighbors refers to the extension of the method using the neighborhood approach (see Sec. 4.3).

Dataset	Hypothesis	Precision	Recall	Specificity	F-measure
Compaq	Original	0.4354	**0.8046**	0.8046	0.5650
	Reverse	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
	Reverse	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
	Reverse	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
	Reverse	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 5.7: *Quantitative result metrics of the methods. For each dataset, we have four different applications: the original hypothesis with respect to P_{Cb_s}, the reverse hypothesis with respect to P_{Cr_s}, the one which combines both, and the extension using the neighborhood approach.*

Figures 5.8, 5.9, and 5.10, present some qualitative results with image samples in column (a) along with the results for each method tested. 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 reverse method results, column (e), the combined method results and column (f) the extended neighborhood method.

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

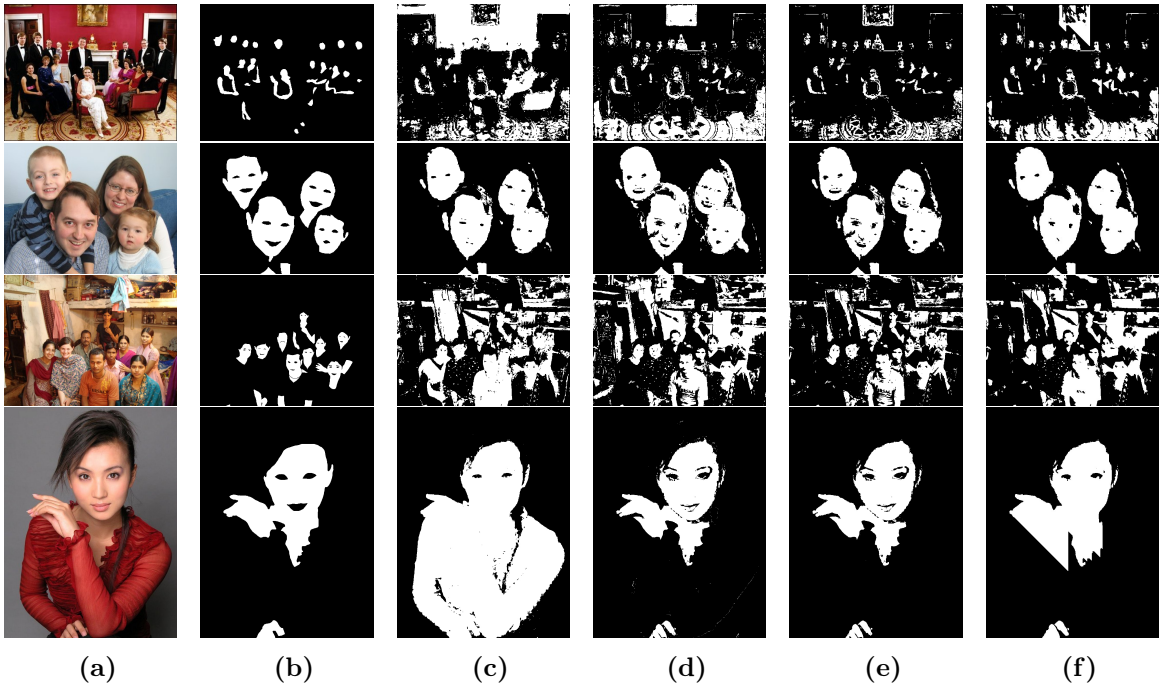
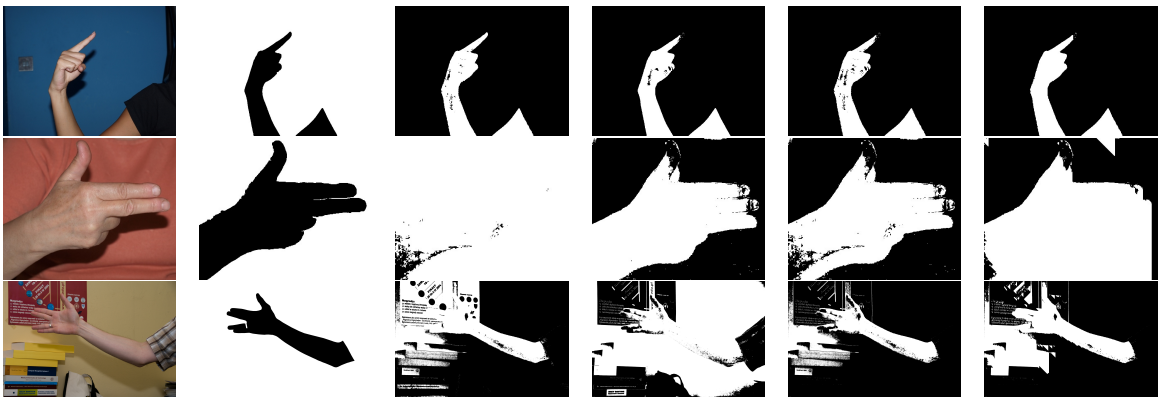


Figure 5.9: *Image samples with the results of each method in Pratheepan dataset: (a) original image (b) ground truth (c) original method Brancati et al. (2017) (d) reverse method (e) combined method (f) neighbors method.*

5.5 Results and Discussion

The original method was compared with six well known rule-based methods in literature using four different datasets, three of them, HGR, Pratheepan and Compaq, have also been used here.



(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Figure 5.10: *Image samples with the results of each method in HGR dataset: (a) original image (b) ground truth (c) original method Brancati et al. (2017) (d) reverse method (e) combined method (f) neighbors method.*

Because the 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.

Table 5.7 shows quantitative result metrics of the experiments. Column 1 refers to the dataset used. Column 2 refers to the method being experimented: Original for the original hypothesis; Reversed refers to the reverse hypothesis with respect to P_{Cr_s} parameter; Combined refers to the combination of both of the former methods (see Sec. 4.2); Neighbors refers to the extension of the method using the neighborhood approach.

As one can see, the reverse 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.

In general, the reverse 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 has very high *F-measure* rates.

The combined method along with the neighborhood approach achieved the best *F-measure* in HGR and Pratheepan. Moreover, the other metrics still are in a very high rate for all datasets, being in the top second in almost cases.

Therefore, the combined and extended approaches are very competitive compared to the original method. Furthermore, all the variations of the original method are still computed in quadratic time, maintaining the desired computational efficiency that are useful in different application domains, mainly near real time systems (processing time of about 10ms for a typical image of dimensions 300x400).

Chapter 6

Plano de Trabalho

Os créditos em disciplinas necessários para o programa de mestrado em Ciência da Computação no IME-USP foram cumpridos de fevereiro de 2015 até junho de 2016, conforme a tabela 6.1. Em meados de 2015, iniciou-se a pesquisa bibliográfica para o desenvolvimento deste projeto. Durante esse período, até o presente momento, foram realizados os experimentos preliminares que estão incorporados neste texto para a etapa de qualificação. Após apresentação, as recomendações da comissão julgadora serão ponderadas e devem ser refletidas nas atividades subsequentes listadas na seção 6.1.

Código	**Disciplina**	**Término**
MAC5832	Aprendizagem Computacional: Modelos, Algoritmos e Aplicações	jun/2015
MAC5744	Introdução à Computação Gráfica	jun/2015
MAC5768	Visão e Processamento de Imagens - Parte I	jun/2015
MAC5711	Análise de Algoritmos	nov/2015
MAC6914	Métodos de Aprendizagem em Visão Computacional	nov/2015
MAC4722	Linguagens, Autômatos e Computabilidade	jun/2016
MAC5714	Programação Orientada a Objetos	jun/2016

Table 6.1: *Disciplinas cursadas no programa de mestrado em Ciência da Computação no IME-USP.*

6.1 Atividades previstas

1. **Revisar leituras adicionais:** outras referências deverão ser adicionadas ao presente trabalho, principalmente, conteúdo relacionado a métodos de *kernel*, que servirão de embasamento para compreender e estudar a técnica de *kernels* em conjuntos *fuzzy* proposta por Guevara *et al.* (2014).

2. **Incorporar novos conjuntos de dados:** essa atividade consiste em analisar e integrar novos conjuntos de dados ao projeto para utilização nos experimentos, tanto de bancos de imagens de cor de pele, como outros conjuntos de imagens coloridas onde a classificação de dados intervalares possa ser aplicada.

3. **Investigar características passíveis de uso em classificadores de dados intervalares:** os classificadores treinados durante a etapa de experimentos preliminares consistiam apenas da informação de cor da imagem, independente do espaço de cores escolhido. Portanto, essa atividade visa compreender o papel de outros atributos, tais como textura e características locais, durante o treinamento e se a inclusão de tais atributos, de fato, pode acarretar em melhoria de performance.

4. **Desenvolver novas ferramentas:** novas ferramentas devem ser desenvolvidas para dar suporte aos experimentos subsequentes, tais como a tabela de busca implementada para otimiza-

ção de parâmetros do *FuzzyDT*. Todo software criado por este projeto de pesquisa será disponibilizado para toda a comunidade científica em formato de código livre.

5. **Elaborar novos experimentos:** com base nas ferramentas desenvolvidas e conjuntos de dados estabelecidos, elaborar e executar novos experimentos.

6. **Analisar os resultados:** os resultados obtidos deverão ser avaliados e reportados no projeto de pesquisa e em publicações a serem realizadas. Eventualmente, correções nas ferramentas desenvolvidas e/ou nos experimentos também serão agregadas nesta etapa.

7. **Publicar resultados:** artigos científicos com os resultados obtidos devem ser publicados em revistas ou conferências sobre processamento de imagens, classificação ou outras áreas relacionadas com o contexto do projeto de pesquisa.

8. **Escrever a dissertação:** a dissertação deverá ser escrita assim que as demais atividades tiverem sido concluídas, principalmente em relação aos experimentos e análise de resultados, o que fornecerá dados consolidados para a conclusão do projeto de pesquisa.

6.2 Cronograma proposto

As atividades listadas na seção 6.1 serão realizadas de acordo com o cronograma disposto na tabela 6.2. As tarefas foram divididas dentro de um período de onze meses de execução e a previsão de defesa da dissertação é outubro de 2017.

Atividade	Meses 2016/2017										
	dez	jan	fev	mar	abr	mai	jun	jul	ago	set	out
1	X	X	X								
2	X	X									
3		X	X								
4			X	X	X	X					
5					X	X	X				
6						X	X	X			
7							X	X			
8								X	X	X	X

Table 6.2: *Cronograma das atividades previstas*

Bibliography

Abu-Mostafa *et al.***(2012)** Yaser S. Abu-Mostafa, Malik Magdon-Ismail e Hsuan-Tien Lin. *Learning from data : a short course*. AMLBook.com. Citado na pág. 14, 15, 16, 34

Albiol *et al.***(2001)** Alberto Albiol, Luis Torres e Edward J. Delp. Optimum color spaces for skin detection. Em *International Conference on Image Processing*, páginas 122–124. Citado na pág. 5

Ben(2009) R. G. Ben. CIE 1931 xy color space diagram. https://en.wikipedia.org/wiki/File:CIE1931xy_blank.svg, 2009. Accessed on 12/10/2016. Citado na pág. 9

Bergasa *et al.***(2000)** Luis Miguel Bergasa, Manuel Mazo, Alfredo Gardel, Miguel A. Sotelo e Luciano Boquete. Unsupervised and adaptive gaussian skin-color model. *Image and Vision Computing*, 18(12):987–1003. Citado na pág. 5

Bhatt e Dhall(2012) Rajen B. Bhatt e Abhinav Dhall. Skin segmentation dataset. https://archive.ics.uci.edu/ml/datasets/Skin+Segmentation, 2012. Accessed on 05/06/2016. Citado na pág. 29

Brancati *et al.***(2017)** Nadia Brancati, Giuseppe De Pietro, Maria Frucci e 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. Citado na pág. 1, 5, 23, 24, 33, 36, 37, 38

Campbell(2000) Colin Campbell. *An Introduction to Kernel Methods*, páginas 155–192. Springer Verlag, Berlin. Citado na pág. 16, 17

Casati *et al.***(2013)** João Paulo Brognoni Casati, Diego Rafael Moraes e Evandro Luis Linhari Rodrigues. SFA: A human skin image database based on FERET and AR facial images. Em *IX workshop de Visão Computacional*. Citado na pág. 30, 31

Chai e Ngan(1999) Douglas Chai e King N. Ngan. Face segmentation using skin-color map in videophone applications. *IEEE Trans. on Circ. and Sys. for Video Tech.*, 9(4):551–564. Citado na pág. 24

Chaves-González *et al.***(2010)** Jose M. Chaves-González, Miguel A. Vega-Rodríguez, Juan A. Gómez-Pulido e Juan M. Sánchez-Pérez. Detecting skin in face recognition systems: A colour spaces study. *Digital Signal Processing*, 20(3):806–823. Citado na pág. 5

Cortes e Vapnik(1995) Corinna Cortes e Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297. Citado na pág. 16, 17

Duda *et al.***(2012)** Richard O. Duda, Peter E. Hart e David G. Stork. *Pattern Classification*. John Wiley & Sons, 2nd edição. Citado na pág. 14, 15, 16, 18, 19

Fleck *et al.***(1996)** Margaret M Fleck, David A Forsyth e Chris Bregler. Finding naked people. Em *European conference on computer vision*, páginas 593–602. Springer. Citado na pág. 1

Gevers *et al.***(2012)** Theo Gevers, Arjan Gijsenij, Joost van de Weijer e Jan-Mark Geusebroek. *Color in Computer Vision: Fundamentals and Applications*. Wiley. Citado na pág. 7, 9, 10

Gonzalez e Woods(2002) Rafael C. Gonzalez e Richard E. Woods. *Digital Image Processing*. Prentice Hall, 2nd edição. Citado na pág. 1, 7, 8, 10, 11

Grzejszczak *et al.***(2016)** Tomasz Grzejszczak, Michal Kawulok e 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. Citado na pág. 32, 33

Guevara *et al.***(2014)** Jorge Guevara, Roberto Hirata Jr e Stephane Canu. Positive definite kernel functions on fuzzy sets. Em *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, páginas 439–446. doi: 10.1109/FUZZ-IEEE.2014.6891628. Accessed on 14/09/2016. Citado na pág. 39

Hsu *et al.***(2002)** Rein-Lien Hsu, M. Abdel-Mottaleb e 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. Citado na pág. 4

Ice(2016) Black Ice. The HSI color space. http://www.blackice.com/images/HSIColorModel.jpg, 2016. Accessed on 15/10/2016. Citado na pág. 12

Jayaram *et al.***(2004)** Sriram Jayaram, Stephen Schmugge, Min C. Shin e Leonid V. Tsap. Effect of colorspace transformation, the illuminance component, and color modeling on skin detection. Em *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, páginas 813–818. IEEE. Citado na pág. 5

Jones e Rehg(2002) Michael J. Jones e James M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96. Citado na pág. 3, 33

Kakumanu *et al.***(2007)** Praveen Kakumanu, Sokratis Makrogiannis e Nikolaos Bourbakis. A survey of skin-color modeling and detection methods. *Pattern recognition*, 40(3):1106–1122. Citado na pág. 1, 3, 5, 11

Kaur e Kranthi(2012) Amanpreet Kaur e 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. Citado na pág. 5

Kawulok *et al.***(2013)** Michal Kawulok, Jolanta Kawulok, Jakub Nalepa e Maciej Papiez. Skin detection using spatial analysis with adaptive seed. Em *2013 IEEE International Conference on Image Processing*, páginas 3720–3724. IEEE. Citado na pág. 5

Kawulok *et al.***(2014)** Michal Kawulok, Jolanta Kawulok, Jakub Nalepa e 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. URL http://asp.eurasipjournals.com/content/2014/1/170. Citado na pág. 32, 33

Kovac *et al.***(2003)** Jure Kovac, Peter Peer e Franc Solina. *Human skin color clustering for face detection*, volume 2. IEEE. Citado na pág. 4, 5

Kumar e Malhotra(2015) Amit Kumar e Shivani Malhotra. Performance analysis of color space for optimum skin color detection. Em *2015 Fifth International Conference on Communication Systems and Network Technologies*, páginas 554–558. IEEE. Citado na pág. 5

Lichman(2013) M. Lichman. UCI machine learning repository. http://archive.ics.uci.edu/ml, 2013. Citado na pág. 29

Lorena e Carvalho(2003) Ana Carolina Lorena e André C. P. L. F. Carvalho. Introdução às Máquinas de Vetores Suporte. Relatório Técnico 192, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Brasil. Citado na pág. 16, 17, 18

Mahmoodi e Sayedi(2016) Mohammad Reza Mahmoodi e Sayed Masoud Sayedi. A comprehensive survey on human skin detection. *International Journal of Image, Graphics and Signal Processing*, 8(5):1–35. Accessed on 23/04/2016. Citado na pág. 3, 5, 33

Martínez e Benavente(1998) Aleix Martínez e Robert Benavente. The AR face database. Relatório técnico, Purdue University. Citado na pág. 5, 30

Minear e Park(2004) Meredith Minear e Denise Park. Productive aging lab face database. https://pal.utdallas.edu/facedb/, 2004. Citado na pág. 29

Mitchell(1997) Tom M. Mitchell. *Machine Learning*. McGraw-Hill Education. Citado na pág. 14, 15, 19, 20, 21

Naji *et al.***(2012)** Sinan A. Naji, Roziati Zainuddin e Hamid A. Jalab. Skin segmentation based on multi pixel color clustering models. *Digital Signal Processing*, 22(6):933–940. Citado na pág. 5

Nalepa e Kawulok(2014) Jakub Nalepa e Michal Kawulok. Fast and accurate hand shape classification. Em Stanislaw Kozielski, Dariusz Mrozek, Pawel Kasprowski, Bozena Malysiak-Mrozek e Daniel Kostrzewa, editors, *Beyond Databases, Architectures, and Structures*, volume 424 of *Communications in Computer and Information Science*, páginas 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. Citado na pág. 32, 33

Pedregosa *et al.***(2011)** F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot e E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830. Citado na pág. 18, 34, 35

Pedrini e Schwartz(2008) Hélio Pedrini e William Robson Schwartz. *Análise de imagens digitais: princípios, algoritmos e aplicações*. Thomson Learning, São Paulo. Citado na pág. 11, 12, 13, 14

Pedrycz e Gomide(1998) Witold Pedrycz e Fernando Gomide. *An Introduction to Fuzzy Sets: Analysis and Design*. A Bradford Book. Citado na pág. 2

Phillips *et al.***(1996)** P. Jonathon Phillips, Harry Wechsler, Jeffrey Huang e Patrick J. Rauss. The facial recognition technology (FERET) database. https://www.nist.gov/programs-projects/face-recognition-technology-feret, 1996. Accessed on 23/06/2016. Citado na pág. 29, 30

Plataniotis e Venetsanopoulos(2000) Konstantinos N. Plataniotis e Anastasios N. Venetsanopoulos. *Color Image Processing and Applications*. Springer, 1st edição. Citado na pág. 7, 8, 10, 12

Quinlan(1986) John Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106. Citado na pág. 20

Quinlan(1993) John Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann. Citado na pág. 21

Rus(2007) Jacob Rus. The Munsell color system. https://commons.wikimedia.org/wiki/File:Munsell-system.svg, 2007. Accessed on 12/10/2016. Citado na pág. 8

Rus(2008) Jacob Rus. The "primary" and "secondary" colors in a four-color print process. https://en.wikipedia.org/wiki/File:SubtractiveColor.svg, 2008. Accessed on 12/10/2016. Citado na pág. 11

Shaik *et al.***(2015)** Khamar Basha Shaik, Ganesan P., V. Kalist, B. S. Sathish e J. Merlin Mary Jenitha. Comparative study of skin color detection and segmentation in HSV and YCbCr color space. *Procedia Computer Science*, 57:41–48. Citado na pág. 5

Shawe-Taylor e Cristianini(2004) John Shawe-Taylor e Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press. Citado na pág. 18

Tan *et al.***(2012)** Wei Ren Tan, Chee Seng Chan, Pratheepan Yogarajah e Joan Condell. A fusion approach for efficient human skin detection. *IEEE Transactions on Industrial Informatics*, 8(1): 138–147. Citado na pág. 5, 31, 32

Vapnik(2013) Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer Science & Business Media. Citado na pág. 16

Vezhnevets *et al.***(2003)** Vladimir Vezhnevets, Vassili Sazonov e Alla Andreeva. A survey on pixel-based skin color detection techniques. Em *IN PROC. GRAPHICON-2003*, páginas 85–92. Citado na pág. 1, 3, 4, 9

Yogarajah *et al.***(2011)** Pratheepan Yogarajah, Joan Condell, Kevin Curran, Paul McKevitt e Abbas Cheddad. A dynamic threshold approach for skin tone detection in colour images. *International Journal of Biometrics*, 4(1):38–55. Citado na pág. 4, 5