

**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

[illegible]

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	ix
List of Symbols	xi
List of Figures	xiii
List of Tables	xv
1 Introduction	1
1.1 Motivação	2
1.2 Trabalhos relacionados	2
1.3 Objetivos	3
1.4 Organização do texto	3
2 Related Work	5
3 Theoretical Background	7
3.1 Color models	7
3.1.1 Munsell color model	7
3.1.2 CIE color model	8
3.1.3 RGB color model	10
3.1.4 CMY color model	10
3.1.5 Color models of the YUV family	11
3.1.6 Color models of the HSI family	12
3.2 Classificadores	14
3.2.1 Máquinas de vetores suporte	16
3.2.2 k -vizinhos mais próximos	19
3.2.3 Árvores de decisão	20
4 Experimentos Preliminares	23
4.1 Conjuntos de dados	23
4.2 Primeiro experimento	24
4.3 Segundo experimento	27
4.4 Terceiro experimento	27

5 Plano de Trabalho	31
5.1 Atividades previstas	31
5.2 Cronograma proposto	32
Bibliography	33

List of Acronyms

AR	Alex and Robert Face Database
CIE	Commission Internationale de l'Eclairage
CMY	Cyan, Magenta and Yellow
FERET	Face Recognition Technology database
HSI	Hue, Saturation, Intensity
HSL	Hue, Saturation, Lightness
HSV	Hue, Saturation, Value
ID3	Iterative Dichotomiser 3
k -NN	k-Nearest Neighbors
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 at 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

3.1	Munsell color model.	8
3.2	CIE 1931 chromaticity diagram	9
3.3	Unit cube representing the colors of the RGB model	10
3.4	CMY subtractive color model	11
3.5	Graphical representation of the HSI model	12
3.6	Fronteira de decisão no espaço de características 2-dimensional	15
3.7	Exemplo de um problema de classes binárias separáveis num espaço 2-dimensional .	17
3.8	Exemplo de aplicação do k -NN na tarefa de classificação num espaço 2-dimensional .	19
4.1	Visão 3-dimensional dos canais RGB do conjunto de dados UCI	24
4.2	Exemplos de imagens do banco de faces SFA	25
4.3	Estrutura das janelas que formam as amostras do SFA	26
4.4	Visão 3-dimensional dos canais RGB do conjunto de dados SFA	26

List of Tables

4.1	Trecho com amostras da base de dados UCI	24
4.2	Tabela de busca dos parâmetros do estimador ótimo na SVM	25
4.3	Tabela de busca dos parâmetros do estimador ótimo no k -NN	26
4.4	Resultados dos experimentos com k -NN e SVM nos conjuntos de dados UCI e SFA .	26
4.5	Resultados dos experimentos com k -NN e SVM no conjunto de dados SFA em espaços de cores distintos	27
4.6	Tabela de busca dos parâmetros do estimador ótimo no <i>FuzzyDT</i>	28
4.7	Resultados dos experimentos com árvore de decisão <i>fuzzy</i> no conjunto de dados SFA em espaços de cores distintos	28
5.1	Disciplinas cursadas no programa de mestrado em Ciência da Computação no IME- USP.	31
5.2	Cronograma das atividades previstas	32

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 Trabalhos relacionados

Check if we will keep fuzzy part + ML approaches to move this part to next chapter.

Em alguns problemas de classificação, o uso de dados intervalares pode ser uma ferramenta poderosa na obtenção de bons resultados. Intervalos representados por conjuntos *fuzzy* também têm grande valia nesta tarefa. Com base nessa característica, Umano *et al.* (1994) propuseram um algoritmo para gerar uma árvore de decisão *fuzzy*, a partir de dados numéricos, usando conjuntos *fuzzy* previamente definidos. Essa implementação é uma adaptação do algoritmo Divisor Iterativo 3 (ID3) clássico proposto por Quinlan (1986), brevemente discutido na seção 3.2.3. A diferença fundamental para o ID3 está no método de seleção do atributo que particiona o conjunto de dados. Assim como no ID3, aqui também é utilizado o ganho de informação como critério, porém, ao invés de aplicar o valor dos atributos no cálculo, o algoritmo computa pelo grau de pertinência dos conjuntos *fuzzy*. Outra propriedade particular da árvore *fuzzy* está no processo de inferência, pois permite que mais de um ramo seja ativado quando um atributo é testado na avaliação do grau de pertinência.

O *fuzzy* ID3 foi utilizado por Bhatt *et al.* (2009) para segmentação de regiões de pele. O conjunto de dados aplicado no experimento contém os três canais de cores do modelo Vermelho, Verde e Azul (RGB) e também foi utilizado nos experimentos deste trabalho, conforme pode ser visto no capítulo 4. Antes da indução da árvore, os dados de treinamento foram aglomerados em cinco *clusters* por meio do algoritmo *fuzzy c-means* (Bezdek *et al.*, 1984). Os *clusters* foram aproximados com função de pertinência gaussiana. A taxa de erro obtida foi de 94,1%.

Outra variante de árvore de decisão para conjuntos *fuzzy* é a proposta de Cintra *et al.* (2013), porém, baseada na versão clássica C4.5, uma versão estendida do ID3 também criada por Quinlan (1993). Chamado de *FuzzyDT*, o algoritmo igualmente usa a entropia e o ganho de informação do C4.5 como medidas para decidir sobre a importância dos atributos. A estratégia de indução da árvore também é a mesma, ou seja, os dados são particionados recursivamente criando ramos até que uma classe é atribuída a cada folha. Porém, antes da indução da árvore, os atributos contínuos são definidos em termos de conjuntos *fuzzy*. Em outras palavras, a árvore é induzida com os valores dos atributos "discretizados", isto é, transformados em intervalos finitos, ou "discretos". O domínio dos conjuntos *fuzzy* devem ser estabelecidos previamente pelo usuário. Experimentos foram realizados com 16 conjuntos de dados distintos obtidos de Lichman (2013), comparando o desempenho do algoritmo com o C4.5 em termos da taxa de erro, número médio de regras e condições geradas por ambas técnicas. Em resumo, o *FuzzyDT* produziu resultados satisfatórios na maioria dos conjuntos de dados avaliados. O algoritmo foi utilizado para experimentos preliminares que podem ser vistos no capítulo 4.

Vale ressaltar que o *FuzzyDT* contém quatro métodos distintos de estimar o número de conjuntos *fuzzy* para cada atributo do conjunto de dados (Cintra *et al.*, 2011). Um deles define um número de conjuntos *fuzzy* uniforme para todos os atributos. Os demais são flexíveis, ou seja, permitem que o número de conjuntos *fuzzy* para cada atributo seja distinto. Um desses métodos é o ganho de informação, calculado e armazenado para cada atributo após o processo de "fuzzificação" dos valores contínuos. O processo é repetido 8 vezes, utilizando de 2 a 9 conjuntos *fuzzy* para cada atributo.

A base de dados final é, então, formada com o número de conjuntos *fuzzy* que produziram o maior ganho de informação para cada atributo. O segundo método é oriundo do algoritmo RELIEF que mede a utilidade dos atributos com base em sua capacidade de distinguir entre amostras semelhantes pertencentes a classes distintas. Ele produz um ranking de mérito médio dos atributos e funciona tanto em dados discretos como contínuos (Kononenko, 1994). O terceiro método é a implementação da técnica proposta por Wang (2003) e conhecida como Wang-Mendel (WM), onde cada atributo é definido usando de 2 a 9 conjuntos *fuzzy* e, em seguida, usado individualmente para gerar um classificador WM. As taxas de erro obtidas do classificador são usadas como critério para selecionar o número de conjuntos *fuzzy*. Os métodos foram avaliados em 11 conjuntos de dados, sendo que os métodos flexíveis mostraram melhores taxas de erro.

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 4, 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 5 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).

In Jones e Rehg (2002), the authors applied a Bayesian decision rule with a 3-dimensional histogram model constructed from approximately 2 billion pixels collected from 18,696 images over the Internet to perform skin detection. They calculated two different histograms for skin and non-skin in the RGB color space. Using those histograms along with training data, a classifier was derived with the standard likelihood ratio approach of a pixel be skin to not be skin. The best performance at an error rate of 88% was reached for histograms of size 32.

Another 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 YCbCr color space, obtaining a true positive rate of 90.66%. They (Kovac *et al.*, 2003) also performed experiments with the chromaticity channels Cb and Cr only. The results showed that the performance of the classifier is inferior in relation to the approach using all Y, Cb and Cr channels. 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. The resulting image is subtracted from the elliptical image. As a result, there is a more uniform skin region where the thresholds are calculated. 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.

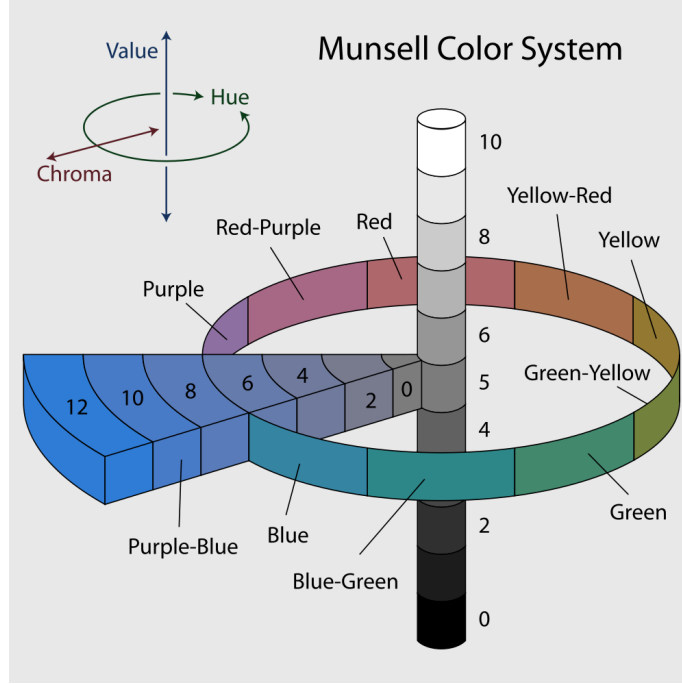


Figure 3.1: *Munsell color model represented by a cylindrical shape. The hue is arranged on the circular axis consisting of five base and five secondary colors, the saturation on the radial axis and the luminance on the vertical axis in a range varying from 0 to 10. Source: Rus (2007).*

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.

$$x = \frac{X}{X + Y + Z} \quad (3.1)$$

$$y = \frac{Y}{X + Y + Z} \quad (3.2)$$

$$z = \frac{Z}{X + Y + Z} \quad (3.3)$$

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.

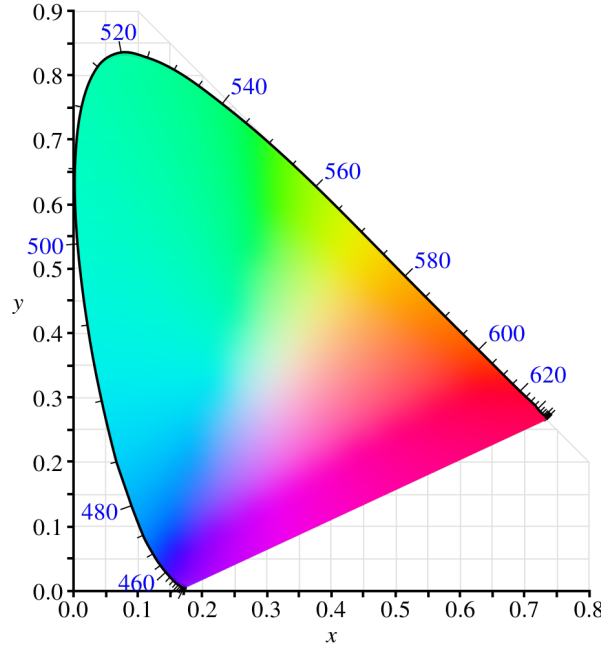


Figure 3.2: CIE 1931 chromaticity diagram. The points representing pure colors in the electromagnetic spectrum are labeled according to their wavelengths and are located along the curve from the right end of the x -axis, corresponding to the red color, to the left end of the same axis, corresponding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

¹Theory 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).

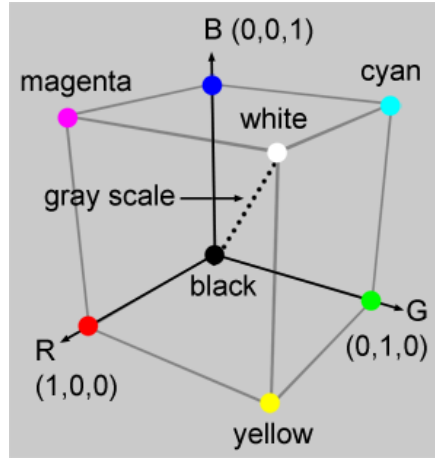


Figure 3.3: Unit cube representing the colors of the RGB model. The origin, given by the vertex $(0,0,0)$, represents the black color. The vertex $(1,1,1)$, opposite the origin, represents the white color. The highlighted vertices on the axes represent the primary colors and the others are the complement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

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

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):

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

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.

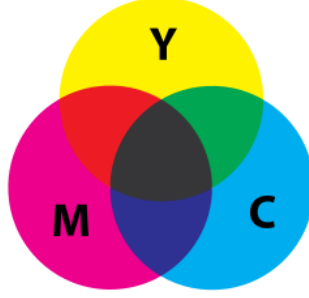


Figure 3.4: CMY subtractive color model. It is interesting to note that the intersection of yellow with magenta generates the red color, magenta with cyan generates the blue color and cyan with yellow generates the green color. Source: Rus (2008).

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

The acronym YUV stands to a family 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:

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

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:

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

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, 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:

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

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 Venetsanopoulos, 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.

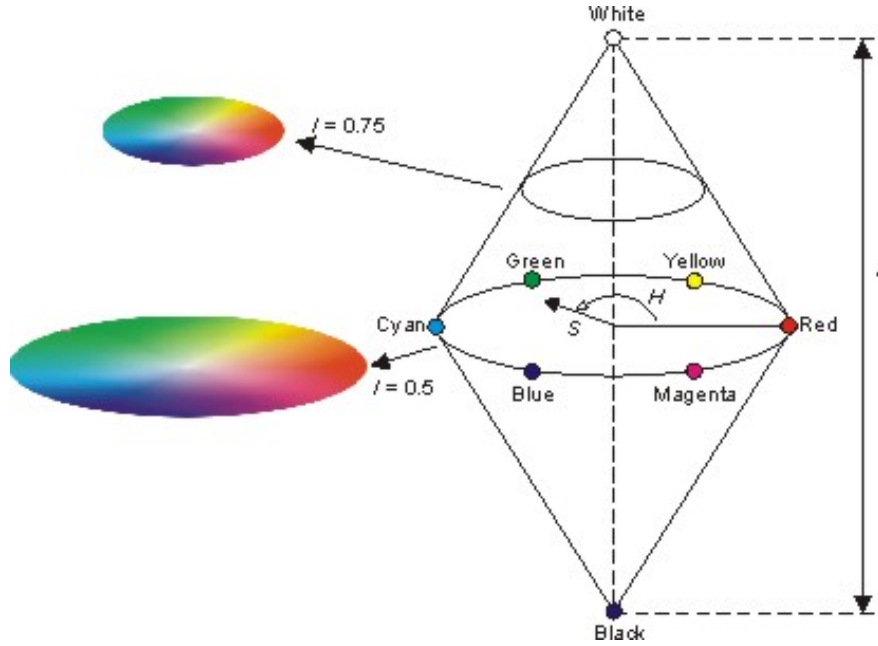


Figure 3.5: Graphical representation of the HSI model. The hue describes the color itself, in the form of an angle θ , where $\theta \in [0, 360]$. Red is at 0 degree, yellow at 60, green at 120, and so on. The saturation component, which varies between 0 and 1, indicates how much color is polluted with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

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

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:

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

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 Classificadores

A aprendizagem de máquina é uma área que está preocupada com o desenvolvimento de programas de computador capazes de melhorar automaticamente com a experiência (Mitchell, 1997). Essa definição está intimamente ligada ao modo com que seres humanos aprendem. Os esforços de pesquisa nessa área têm sido realizados com o propósito de aproximar esta relação.

Como exemplo, ao mostrar a imagem de uma árvore para uma criança de três anos de idade, muito provavelmente ela saberá reconhecê-la, pois deve ter sido exposta a situações em que tenha visto imagens semelhantes e, sendo assim, foi treinada para dar tal resposta. Logo, ela aprendeu o que é uma árvore apenas olhando para elas, não necessariamente por meio de definições matemáticas precisas. Em outras palavras, o aprendizado foi feito com base em dados que, muitas vezes, são utilizados para obter-se soluções empíricas para determinados problemas onde não há a possibilidade de se criar uma solução analítica (Abu-Mostafa *et al.*, 2012).

Analogamente, um algoritmo pode ser treinado para diferenciar uma árvore de outros objetos com base em um conjunto de dados que possui descrições sobre as árvores, tais como, altura, cores, espessura, comprimento, etc. Tais descrições também são chamadas de atributos, propriedades ou características e são submetidas a um classificador que avalia as evidências apresentadas e toma uma decisão sobre o objeto sendo analisado (Duda *et al.*, 2012). Essa tarefa começa com a definição de um vetor de características em um espaço d -dimensional, da forma (Duda *et al.*, 2012):

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

onde $x \in X$ e X é o espaço de entrada, ou seja, todos os x vetores possíveis.

O conjunto de dados é formado por N destes vetores e, sendo assim, o problema agora está em particionar o espaço de características de tal maneira que uma fronteira de decisão é formada (Duda *et al.*, 2012). Pode-se, então, atribuir uma classe ou rótulo y específico para um dado vetor,

onde $y \in Y$ e Y é um conjunto finito de classes que, no caso binário, é da forma $Y = \{+1, -1\}$. A figura 3.6 mostra exemplos de particionamento de um espaço de características 2-dimensional de amostras de pele e não pele do conjunto de dados SFA².

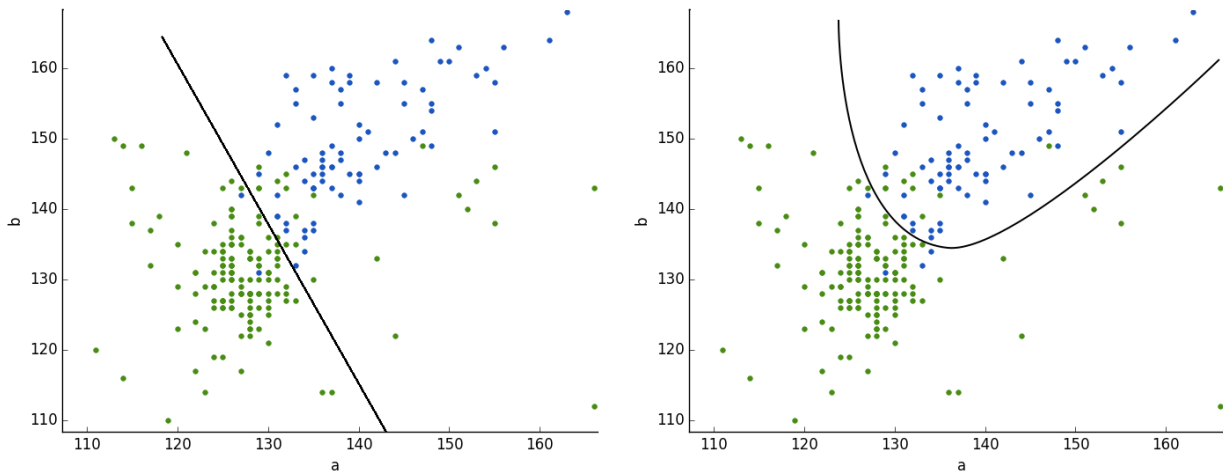


Figure 3.6: Fronteira de decisão no espaço de características 2-dimensional de amostras de pele e não pele do conjunto de dados SFA. Os eixos representam os canais a e b do modelo de cores Lab. As figuras à esquerda e à direita mostram a separação do espaço de características por funções linear e não linear, respectivamente. Vale ressaltar que, a complexidade computacional aumenta significativamente nos casos em que existem muitas características. Fonte: proposto pelo autor.

Note que a fronteira de decisão é uma função hipotética que o algoritmo de aprendizado deve produzir com base no treinamento dos dados. Ela deve ser mais próxima possível de uma função alvo ou função objetivo $f : X \mapsto Y$ que é desconhecida (Abu-Mostafa *et al.*, 2012).

Este tipo de abordagem é caracterizada como aprendizagem supervisionada, ou seja, os dados de treinamento contêm amostras explícitas de como deve ser a saída correta sobre um vetor de entrada (Abu-Mostafa *et al.*, 2012).

Na aprendizagem não supervisionada, também conhecida como clusterização, não se sabe a que classes pertencem os dados de treinamento. A missão do classificador, neste caso, é aglomerar os dados de entrada em agrupamentos naturais (Duda *et al.*, 2012). Logo, a aprendizagem não supervisionada pode ser vista como uma tarefa de encontrar padrões ou estruturas espontaneamente a partir dos dados de entrada (Abu-Mostafa *et al.*, 2012).

Uma outra abordagem é a aprendizagem reforçada que, da mesma forma que na aprendizagem não supervisionada, não usa dados de entrada rotulados, em vez disso, a saída proposta pelo treinamento é utilizada juntamente com uma medida de sua qualidade para melhorar os resultados do classificador (Abu-Mostafa *et al.*, 2012).

Ainda sobre os resultados do classificador, é importante salientar que aprender os parâmetros de uma função alvo e testá-la nos mesmos dados é um erro fundamental, pois o modelo repetiria os rótulos das amostras tão logo treinadas implicando em um ajuste perfeito dos dados, porém, não seria suficiente para prever qualquer coisa útil sobre novos dados de entrada. Este fenômeno é chamado de sobreajuste e para evitá-lo, o conjunto de dados é, então, particionado de forma a armazenar parte dos dados para uma etapa subsequente de teste, no intuito de encontrar o estimador com a melhor performance (Abu-Mostafa *et al.*, 2012).

A divisão dos dados em subconjuntos disjuntos de treinamento e de teste é uma abordagem eficaz quando uma grande quantidade de dados está disponível. Entretanto, quando os dados são limitados, a retenção de parte deles para o conjunto de teste reduz ainda mais o número de amostras disponíveis para treinamento (Mitchell, 1997). Logo, uma outra abordagem, conhecida como validação cruzada, pode ser aplicada de modo que todo o conjunto de dados é usado no treinamento e teste.

²Este conjunto de dados será detalhado na seção 4.1.

A validação cruzada consiste em dividir o conjunto de dados D em K subconjuntos disjuntos D_1, D_2, \dots, D_K , onde cada subconjunto tem tamanho aproximado N/K . O modelo é, então, treinado em cada um desses subconjuntos, exceto um que é mantido como um conjunto de validação, no qual a medida de erro é realizada. Este processo é repetido K vezes, de tal forma que cada um dos subconjuntos tem a oportunidade de agir como o conjunto de teste e, por isso, essa abordagem também é conhecida como *K-fold*. Ao final de todas as K iterações, a média do erro obtido por cada estimador é usado como a medida de performance do classificador (Abu-Mostafa *et al.*, 2012).

Alguns dos classificadores, que foram usados nos experimentos preliminares do capítulo 4, serão brevemente tratados nas seções 3.2.1 e 3.2.2.

3.2.1 Máquinas de vetores suporte

As Máquinas de Vetores Suporte (SVM) constituem uma técnica de aprendizagem computacional baseada na Teoria de Aprendizado Estatístico (Vapnik, 2013), cujo objetivo era de resolver problemas de classificação de padrões. Na prática, uma SVM tem a habilidade de gerar um hiperplano ou conjunto de hiperplanos num espaço de alta ou infinita dimensionalidade, que pode ser usado para tarefas de classificação, regressão ou outras (Duda *et al.*, 2012).

Derivado do próprio nome da técnica, os vetores de suporte são as amostras do conjunto de treinamento que definem o hiperplano ótimo e são os padrões mais difíceis de classificar (Duda *et al.*, 2012). Intuitivamente, eles são os padrões mais relevantes para a tarefa de classificação, pois uma mudança nestes vetores implicam diretamente no resultado do hiperplano ótimo³.

Seja o conjunto de dados de treinamento com N amostras da forma:

$$D = (x_1, y_1), (x_2, y_2), \dots, (x_N, y_N) \quad (3.13)$$

onde cada x_i é um vetor d -dimensional da forma dada em 3.12, $i = 1, 2, \dots, N$, $y_i \in Y$ e $Y = \{+1, -1\}$. Portanto, o conjunto de treinamento contém N observações com suas respectivas classes.

Assumindo que D é linearmente separável, pode-se separar os dados por meio de um hiperplano usando um classificador, também linear, definido pela equação (Lorena e Carvalho, 2003):

$$w \cdot x + b = 0 \quad (3.14)$$

onde $w \cdot x$ é um produto escalar, w é o vetor normal ao hiperplano e b é um termo compensador. O parâmetro $\frac{b}{\|w\|}$ determina o deslocamento do hiperplano em relação à origem.

A partir dessa definição, outros dois hiperplanos paralelos ao hiperplano ótimo podem ser obtidos, conforme as equações em 3.15, de forma que uma região delimitada, conhecida como margem, se forma entre eles.

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

Além disso, algumas restrições são definidas para evitar que não existam pontos entre $w \cdot x + b = 0$ e $w \cdot x + b = \pm 1$, formalmente tem-se (Lorena e Carvalho, 2003):

$$\begin{cases} w \cdot x_i + b \geq +1, & \text{se } y_i = +1 \\ w \cdot x_i + b \leq -1, & \text{se } y_i = -1 \end{cases} \quad (3.16)$$

ou, equivalentemente:

$$y_i(w \cdot x_i + b) \geq 1 \quad (3.17)$$

Segundo Campbell (2000), no sistema dado na equação 3.16, supõe-se que a margem é sempre maior que a distância entre $w \cdot x + b = 0$ e $|w \cdot x + b| = 1$ e, por essa razão, SVMs desta natureza

³O hiperplano de separação ótimo de uma SVM é aquele de uma classe de hiperplanos com a maior margem de separação entre os dois conjuntos de treinamento (Cortes e Vapnik, 1995).

são usualmente denominadas de SVMs com margens rígidas. A figura 3.7 mostra a representação gráfica destes conceitos.

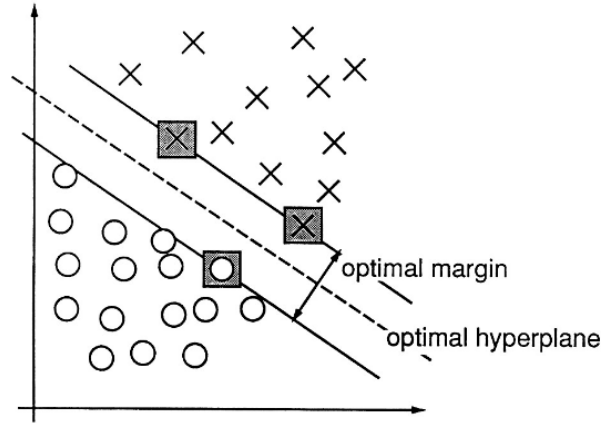


Figure 3.7: Exemplo de um problema de classes binárias separáveis num espaço 2-dimensional. Os vetores de suporte, marcados com quadrados cinza, definem a margem de maior separação entre as duas classes. Fonte: Cortes e Vapnik (1995).

Geometricamente, a distância entre os dois hiperplanos paralelos ao hiperplano ótimo é $\frac{2}{\|w\|}$. Consequentemente, pode-se deduzir que a distância entre o hiperplano ótimo e $w \cdot x + b = \pm 1$ é $\frac{1}{\|w\|}$. Portanto, a minimização de $\|w\|$ maximiza a margem e, sendo assim, tem-se um problema de otimização no qual deseja-se minimizar $\|w\|^2$ sujeito às restrições em 3.17 (Lorena e Carvalho, 2003). Logo, w e b ótimos que resolvem este problema definem o classificador e podem ser obtidos por multiplicadores de Lagrange (Campbell, 2000):

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

$$\text{Sujeito a: } \begin{cases} \alpha_i \geq 0 \\ \sum_{i=1}^N \alpha_i y_i = 0 \end{cases} \quad (3.19)$$

onde α são os multiplicadores de Lagrange.

É importante ressaltar que este tipo de SVM tem êxito em conjuntos de dados de treinamento linearmente separáveis. Nos casos em que os dados são não linearmente separáveis, é admitida a ocorrência de alguns erros de classificação para o conjunto de treinamento, pela inclusão de variáveis de relaxamento (Lorena e Carvalho, 2003). Essas alterações foram produzidas por Cortes e Vapnik (1995) e fazem parte de uma técnica denominada suavização de margens e, por conseguinte, SVMs dessa família também são chamadas de SVM com margens suaves. Logo, o problema de otimização torna-se (Lorena e Carvalho, 2003):

$$\text{Minimizar: } \|w\|^2 + C \sum_{i=1}^N \xi_i \quad (3.20)$$

$$\text{Sujeito a: } \begin{cases} \xi_i \geq 0 \\ y_i(w \cdot x_i + b) \geq 1 - \xi_i \end{cases} \quad (3.21)$$

onde C é uma constante de regularização determinada empiricamente que impõe um peso diferente para o treinamento em relação à generalização. Sendo assim, o problema pode ser resolvido da mesma maneira com a equação em 3.18, porém, com algumas alterações nas restrições (Lorena e Carvalho,

2003):

$$\text{Sujeito a: } \begin{cases} 0 \leq \alpha_i \leq C \\ \sum_{i=1}^N \alpha_i y_i = 0 \end{cases} \quad (3.22)$$

Em geral, a maioria dos problemas de classificação impedem que classificadores lineares sejam utilizados por não ser possível obter resultados satisfatórios na partição dos dados de treinamento por um hiperplano. Entretanto, as SVMs lineares podem ser estendidas para lidar com essa situação mapeando o espaço de entrada em um espaço de características de dimensão mais alta, tipicamente muito maior que o espaço original (Duda *et al.*, 2012), da forma:

$$D' = (\Phi(x_1), y_1), (\Phi(x_2), y_2), \dots, (\Phi(x_N), y_N) \quad (3.23)$$

A escolha apropriada de uma função Φ torna o conjunto de treinamento linearmente separável no espaço transformado (Lorena e Carvalho, 2003). A forma do hiperplano ótimo agora é definida por:

$$w \cdot \Phi(x) + b = 0 \quad (3.24)$$

Logo, os conceitos de vetores de suporte, margem, hiperplanos paralelos e, conseqüentemente, de problema de otimização para encontrar a solução para os parâmetros w e b se aplicam aqui de forma similar. Formalmente, o problema de otimização pode ser resolvido como (Lorena e Carvalho, 2003):

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

sujeito às mesmas restrições definidas em 3.22.

Agora, há a necessidade de se definir como o produto interno $\Phi(x_i) \cdot \Phi(x_j)$ entre dois vetores quaisquer $x_i, x_j \in D$ é realizado, cuja resposta está na introdução do conceito de *kernels* (Lorena e Carvalho, 2003).

Kernels são funções que têm a finalidade de projetar os vetores de entrada num espaço de características com número de dimensões exponencial ou infinito (Shawe-Taylor e Cristianini, 2004), da forma:

$$k(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (3.26)$$

Logo, é possível aplicar um *kernel* como descrito na equação 3.26 na etapa de otimização da SVM, calculando de maneira eficiente o produto interno a partir dos dados de entrada, sem nem mesmo computar explicitamente o mapeamento da função Φ (Shawe-Taylor e Cristianini, 2004).

Alguns dos *kernels* mais utilizados são (Lorena e Carvalho, 2003):

(i) *Kernel* linear

$$k(x_i, x_j) = (x_i \cdot x_j) \quad (3.27)$$

O *kernel* linear é o mais simples das funções *kernel* e se dá pelo produto interno de dois vetores x_i e x_j .

(ii) *Kernel* polinomial

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

onde g é o grau do polinômio e r é um termo constante. Para este *kernel*, os mapeamentos Φ também são funções polinomiais e, portanto, a complexidade é proporcional à escolha de g (Lorena e Carvalho, 2003).

(iii) *Kernel* gaussiano

$$k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (3.29)$$

Também conhecido como Função de Base Radial (RBF), o *kernel* gaussiano corresponde a um espaço de características de dimensão infinita (Lorena e Carvalho, 2003).

No *kernel* RBF e polinomial, o parâmetro γ pode ser visto como o inverso do raio de influência das amostras selecionadas pelo modelo como vetores de suporte (Pedregosa *et al.*, 2011).

3.2.2 k -vizinhos mais próximos

O k -Vizinhos Mais Próximos (k -NN) é um algoritmo baseado em instâncias, o que significa que a função alvo não é aprendida com base em amostras de treinamento; elas simplesmente são mantidas, de maneira que a relação entre uma nova amostra com as amostras de treinamento armazenadas é avaliada e uma função alvo é, então, atribuída a ela (Mitchell, 1997).

Como o próprio nome sugere, o k -NN classifica um novo ponto x atribuindo-lhe a classe de maior frequência dentre as k amostras mais próximas, ou seja, a decisão da classe de x é feita por maioria de votos dos k vizinhos mais próximos e, por essa razão, é interessante que a escolha de k seja um número ímpar para evitar empates (Duda *et al.*, 2012).

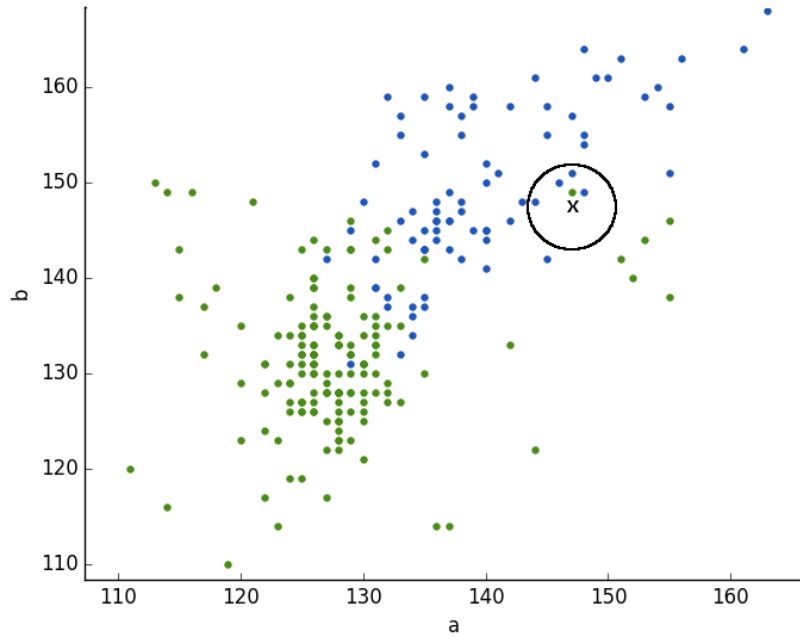


Figure 3.8: Exemplo de aplicação do k -NN na tarefa de classificação num espaço 2-dimensional. O algoritmo avalia as k amostras próximas de x , criando uma região esférica, e rotula x com a classe de maior frequência. Neste caso, $k = 5$ e a classe atribuída a x deve ser a mesma dos pontos em azul. Fonte: proposto pelo autor.

Seja D o conjunto de dados de treinamento com N amostras da forma apresentada em 3.13. Logo, a distância entre duas amostras x_i e x_j quaisquer pode ser obtida em termos da distância Euclidiana, definida como:

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

onde a_r refere-se ao r -ésimo atributo do vetor de entrada x e $i, j = 1, 2, \dots, N$. Outras métricas de distância podem ser atribuídas a $d(x_i, x_j)$, tais como, Manhattan, Chebyshev e Minkowski (Duda *et al.*, 2012).

Para, então, classificar uma nova amostra x_q , toma-se x_1, \dots, x_k instâncias do conjunto de treinamento, cujas distâncias dadas por 3.30, são dos k pontos mais próximos de x_q . Sendo assim, a função que estima a classe de x_q é dada por (Mitchell, 1997):

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

onde

$$\delta(y, f(x_i)) = \begin{cases} 1, & \text{se } y = f(x_i) \\ 0, & \text{caso contrário} \end{cases} \quad (3.32)$$

É importante ressaltar que $f(x_i)$ é conhecida, ou seja, é a classe da amostra x_i do conjunto de treinamento.

Uma variação óbvia da função dada na equação 3.31 é a atribuição de pesos de cada um dos k vizinhos, conforme sua distância, para um ponto x_q sendo classificado (Mitchell, 1997). Essa variação implica que pontos mais próximos de x_q têm maior influência na sua rotulação. Formalmente, tem-se que (Mitchell, 1997):

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

onde

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

No caso em que $d(x_q, x_i) = 0$, ou seja, x_q e x_i estão exatamente nas mesmas coordenadas, $g(x_q)$ pode assumir o mesmo valor de $f(x_i)$. Se há outras amostras de treinamento x_i com a mesma característica, então x_q pode assumir a classe da maioria delas (Mitchell, 1997).

3.2.3 Árvores de decisão

Árvore de decisão é um método para aproximação de funções alvo discretas, nas quais a função aprendida é representada por uma árvore de decisão ou, ainda, por um conjunto de regras do tipo *Se-Então* que são de fácil interpretação. É uma das técnicas de aprendizagem mais populares de inferência indutiva⁴ (Mitchell, 1997).

As amostras de um conjunto de dados são classificadas por uma árvore de decisão por um processo iterativo onde um atributo (nó) é escolhido como raiz até algum nó folha, onde a classe é atribuída à amostra. Cada ramo partindo de um nó representa um dos valores possíveis de um dado atributo (Mitchell, 1997).

Há uma família de algoritmos de árvore de decisão. Um deles é o Divisor Iterativo 3 (ID3) proposto por Quinlan (1986). O ID3 avalia cada atributo através de um teste estatístico para determinar o quão bem ele, por si só, classifica as amostras de treinamento. O melhor atributo é selecionado como o nó raiz da árvore. Um ramo descendente do nó raiz é criado para cada valor possível deste atributo e as amostras de treinamento são classificadas para o nó descendente adequado. Este processo é então repetido recursivamente utilizando as amostras associadas a cada nó descendente. Esse é um algoritmo de busca guloso, já que ele não retrocede para reconsiderar escolhas anteriores (Mitchell, 1997). O algoritmo para quando o subconjunto de amostras associado a um nó é da mesma classe ou quando não há relevância estatística para uma nova partição (Quinlan, 1986).

Para medir a impureza de uma partição, o ID3 usa o conceito de entropia, formalmente (Quinlan, 1986):

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

onde D é o conjunto de dados de treinamento com N amostras da forma apresentada em 3.13, y_{\oplus} e y_{\ominus} é a proporção de amostras positivas e negativas de D , respectivamente. É importante notar que a entropia é 0 quando todas as amostras de D pertencem à mesma classe, 1 quando D contém um número igual de amostras positivas e negativas e um valor entre 0 e 1 nos demais casos (Mitchell, 1997).

⁴A tarefa de indução é desenvolver uma regra de classificação que pode determinar a classe de qualquer objeto a partir dos valores de seus atributos (Quinlan, 1986).

Dada a entropia como uma medida de impureza de um conjunto de amostras de treinamento, pode-se definir o teste estatístico, conhecido como ganho de informação, que mede a efetividade de um atributo na classificação dos dados de treinamento, formalmente (Quinlan, 1986):

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

onde $V(a_r)$ é o conjunto de todos os possíveis valores do atributo a_r , D_v é o subconjunto de D no qual o atributo a_r tem o valor v .

Vale enfatizar que o primeiro termo da equação dado em 3.36 é a entropia do conjunto original D e o segundo termo é o valor esperado da entropia depois que D é particionado usando o atributo a_r (Mitchell, 1997). Outro aspecto importante é que o algoritmo inicia com o conjunto D original, que é substituído por D_v à medida que a recursão se aprofunda.

Batizado de C4.5, Quinlan (1993) estendeu o ID3 para possibilitar o uso de atributos contínuos, atributos com dados ausentes e melhoria na eficiência computacional. Além disso, essa versão cuida de questões como quão profunda a árvore deve ser para evitar que os dados de treinamento sejam perfeitamente classificados, ou seja, quando o conjunto de treinamento é particionado até que cada subconjunto contenha apenas amostras de uma única classe. A estratégia adotada por Quinlan (1993) foi podar a árvore posteriormente à geração da árvore ajustada. Esse processo, apesar de ser mais lento que a proposta anterior do ID3, tornou o algoritmo mais confiável e com maior capacidade de generalização.

Chapter 4

Experimentos Preliminares

Uma série de experimentos foram realizados com dois conjuntos de dados de pele de seres humanos. Os classificadores utilizados foram SVM, k -NN e a árvore de decisão *fuzzy* proposta por Cintra *et al.* (2013). Outro aspecto estudado é o efeito da escolha do modelo de cores no classificador. A formação dos conjuntos de dados é detalhada na seção 4.1 e os resultados dos experimentos podem ser vistos nas seções 4.2, 4.3 e 4.4.

4.1 Conjuntos de dados

Para os experimentos foram utilizadas duas bases de dados distintas. A primeira delas, denominada UCI neste trabalho, foi proposta por Bhatt e Dhall (2012) e obtida no repositório de aprendizagem de máquina da Universidade da Califórnia em Irvine (Lichman, 2013). A base é constituída por imagens de várias texturas de pele e não pele obtidas a partir de milhares de imagens arbitrárias de faces de diferentes idades, sexo e raças (Minear e Park, 2004; Phillips *et al.*, 1996).

A UCI contém 245.057 amostras, compostas por 3 atributos que constituem o vetor de entradas $x = [x_1, x_2, x_3]$, $x \in \mathbb{R}^d$, sendo d a dimensão do espaço, e que representam, respectivamente, os canais azul (B), verde (G) e vermelho (R) do modelo de cores RGB, além de uma quarta coluna que determina a classe a qual a amostra x pertence, denotada por y , sendo $y \in Y$ e $Y = \{+1, -1\}$.

Em outras palavras, cada amostra é um pixel RGB com um determinado rótulo. Como tem-se os dados rotulados com exemplos de qual é a saída correta para uma dada entrada, os experimentos subsequentes são categorizados como sendo de aprendizado supervisionado (Abu-Mostafa *et al.*, 2012).

A tabela 4.1 exemplifica um pequeno trecho da base de dados UCI. Vale ressaltar que do total das 245.057 amostras, 194.198 são de pixels não pele e 50.859 de pixels com diferentes tons de pele. Além disso, as imagens que foram utilizadas na extração do conjunto de dados não foi disponibilizada pelos autores.

Uma vez que os dados são pontos no espaço RGB, é possível plotar os dados para melhor interpretação dos mesmos, conforme mostra a figura 4.1.

O segundo conjunto de dados utilizado nos experimentos foi proposto por ao Paulo Brognoni Casati *et al.* (2013), intitulado pelos autores como banco de imagens de pele humana baseado no FERET e AR (SFA), cuja sigla também será utilizada neste trabalho. O SFA possui um conjunto de imagens de faces obtidas a partir de outros dois bancos de imagens coloridas: o FERET, criado por Phillips *et al.* (1996) e o AR, proposto por Martínez e Benavente (1998), tendo sido utilizadas 876 e 242 imagens de cada um, respectivamente. É importante ressaltar que as imagens do AR têm fundo branco e pequenas variações de cor da pele e, portanto, o ambiente é mais controlado que as imagens do FERET (ao Paulo Brognoni Casati *et al.*, 2013). A figura 4.2 mostra um exemplo das 1.118 imagens do banco.

As amostras de pele e não pele foram geradas aleatoriamente considerando a máscara *ground truth*¹ de cada imagem, sendo três amostras de pele e cinco de não pele. Cada amostra é uma

¹ *Ground truth* é o termo utilizado para denotar uma imagem cujo objeto de interesse está devidamente segmentado

B	G	R	Classe
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 4.1: Trecho com amostras da base de dados UCI. Cada uma das três primeiras colunas representam um canal do pixel do espaço de cores RGB e variam entre 0 e 255. A quarta coluna é a classe atribuída à amostra, que pode assumir +1 se for pele e -1, caso contrário. Originalmente, a classe que representa um pixel de não pele tinha valor 2, substituída por -1 para conformidade com os experimentos.

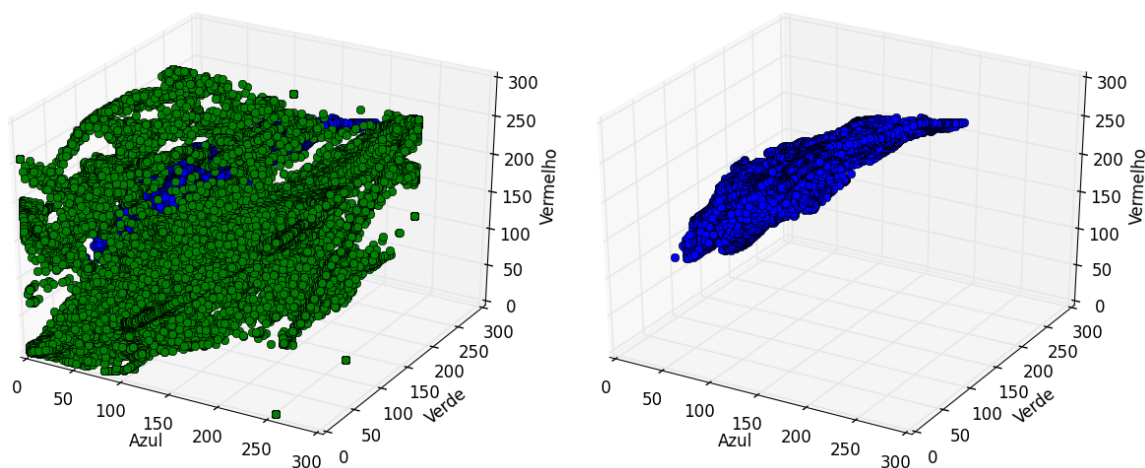


Figure 4.1: Visão 3-dimensional dos canais RGB do conjunto de dados UCI. Os pontos em azul são amostras de pele e os verdes de não pele. À esquerda tem-se todas as amostras do conjunto de dados; à direita apenas as amostras de pele. Fonte: proposto pelo autor.

janela de tamanho $n \times n$, sendo n ímpar, com um pixel central, a partir do qual outros tamanhos de amostra foram criados, que variam de 1×1 até 35×35 , como pode ser visto na figura 4.3 (ao Paulo Brognoni Casati *et al.*, 2013).

A partir das amostras criadas no SFA, um conjunto de dados foi extraído gerando uma base de dados de pele e não pele similar ao UCI na tabela 4.1. O tamanho da janela utilizado foi 9×9 e o espaço de cores também foi o RGB. Portanto, o total de amostras empregado nos experimentos é 724.464, sendo 271.674 de pele e 452.790 de não pele. A classe atribuída à cada amostra tem o mesmo valor que no UCI, ou seja, +1 se for pele e -1, caso contrário. A figura 4.4 mostra a representação gráfica dos dados plotados em 3 dimensões.

4.2 Primeiro experimento

O primeiro experimento foi realizado com k -NN e SVM, disponíveis no pacote *scikit-learn* (Pedregosa *et al.*, 2011). O espaço de cores usado originalmente foi o RGB, como citado na descrição dos conjuntos de dados em 4.1. Em ambos os casos, a estratégia escolhida de validação cruzada foi a *10-fold*, que é uma escolha comum desta abordagem na prática (Abu-Mostafa *et al.*, 2012). Além disso, a técnica de tabela de busca do *scikit-learn* também foi utilizada com o objetivo

e ressaltado, descartando os pixels remanescentes atribuindo-lhes cores uniformes.



Figure 4.2: Exemplos de imagens do banco de faces SFA. Na parte superior tem-se a imagem original e na inferior o ground truth com os pixels de cor de pele anotados manualmente, sendo que a cor preta $RGB = (0, 0, 0)$ foi atribuída a todos os pixels no fundo. Fonte: ao Paulo Brognoni Casati et al. (2013).

de encontrar os parâmetros mais adequados para cada classificador.

A tabela de busca é usada no *scikit-learn* para encontrar os parâmetros ótimos de um classificador quando eles não podem ser aprendidos pelo estimador, tais como, *kernel* e γ na SVM ou número de vizinhos no *k*-NN (Pedregosa et al., 2011). As tabelas de parâmetros utilizadas no treinamento da SVM e *k*-NN podem ser vistas em 4.2 e 4.3. Os parâmetros de cada linha são combinados na tentativa de encontrar o estimador ótimo. Por exemplo, uma escolha no treinamento da SVM seria *kernel*=rbf, $C=100$, *gamma*=1e-4. Todas as combinações possíveis são exploradas, sendo que a melhor delas é retornada (Pedregosa et al., 2011).

<i>kernel</i>	<i>C</i>				<i>gamma</i>			<i>degree</i>	
rbf	1	10	100	1000	1e-3	1e-4	1e-5		
poly	1	10	100	1000		1e-4	1e-5	3	4
linear	1	10	100	1000					

Table 4.2: Tabela de busca dos parâmetros do estimador ótimo na SVM. A coluna *kernel* refere-se aos kernels usados no treinamento que são gaussiano, polinomial e linear, respectivamente. *C* é um parâmetro de regularização que diz à SVM a quantidade de erro admitido no treinamento. *Gamma* é um parâmetro usado somente nos kernels gaussiano e polinomial, conforme citado em 3.2.1. *Degree* é o grau do polinômio; usado somente no kernel polinomial (Pedregosa et al., 2011).

Os resultados deste experimento podem ser vistos na tabela 4.4. Vale ressaltar que o treinamento foi executado com 10 tarefas em paralelo em ambos os classificadores e 30% dos dados, aleatoriamente, foram separados para teste.

Como pode ser visto na tabela 4.4, o *k*-NN teve resultados ligeiramente melhores que a SVM no conjunto de dados SFA. Ambos classificadores tiveram medidas de qualidade muito altas no conjunto de dados UCI, próximas a 100%, o que, provavelmente, é uma situação de sobreajuste. Uma causa possível está na separação dos dados de treinamento e teste. Há muitas repetições entre

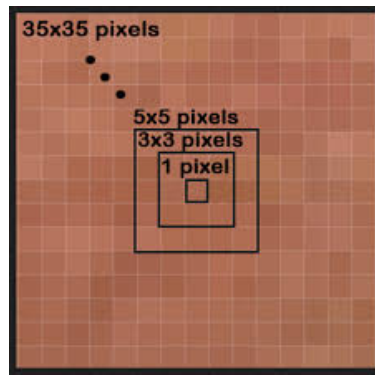


Figure 4.3: Estrutura das janelas que formam as amostras do SFA. No total, são 3.354 amostras de pele e outras 5.590 de não pele para cada tamanho de janela. Fonte: ao Paulo Brognoni Casati et al. (2013).

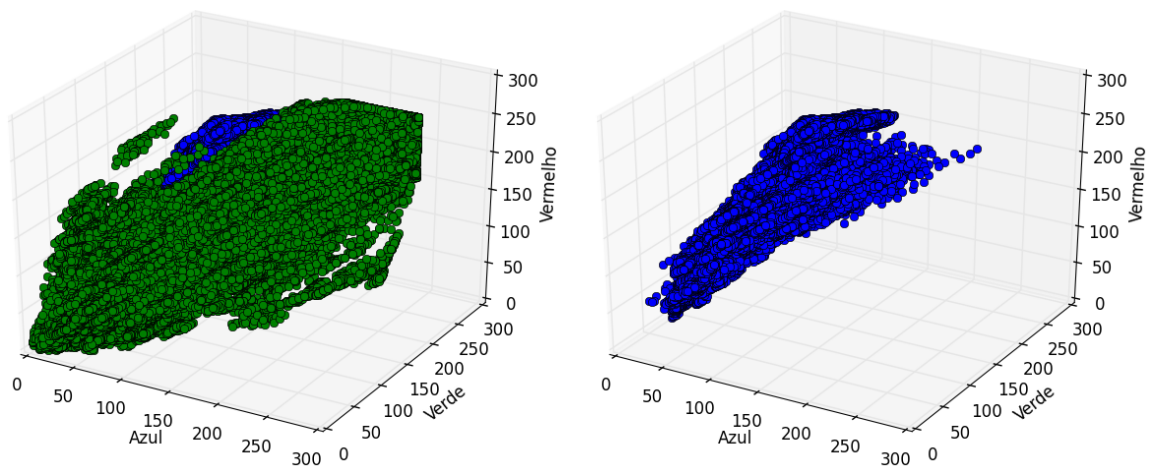


Figure 4.4: Visão 3-dimensional dos canais RGB do conjunto de dados SFA. Os pontos em azul são amostras de pele e os verdes de não pele. À esquerda tem-se todas as amostras do conjunto de dados gerado; à direita apenas as amostras de pele. Fonte: proposto pelo autor.

$n_neighbors$										$weights$		$algorithm$
3	5	9	15	25	50	100	200	400	800	distance	uniform	auto

Table 4.3: Tabela de busca dos parâmetros do estimador ótimo no k -NN. A coluna $n_neighbors$ refere-se ao número de vizinhos considerado no treinamento. $Weights$ é a função peso usada na predição, onde *uniform* indica que os pontos têm pesos iguais e *distance* indica que o inverso da distância é aplicado na classificação, conforme citado em 3.34. A terceira coluna indica qual algoritmo deve ser utilizado; *auto* significa que o algoritmo será decidido com base nos dados (Pedregosa et al., 2011).

Conjunto de dados	Classificador	Modelo de cores	$Precision$	$Recall$	$F1-score$
UCI	k -NN	RGB	0,9995	0,9995	0,9995
	SVM	RGB	0,9995	0,9995	0,9995
SFA	k -NN	RGB	0,9672	0,9669	0,9670
	SVM	RGB	0,9643	0,9628	0,9638

Table 4.4: Resultados dos experimentos com k -NN e SVM nos conjuntos de dados UCI e SFA. Os parâmetros ótimos do k -NN no treinamento com UCI foram $n_neighbors=3$, $weights=uniform$ e com SFA $n_neighbors=15$, $weights=uniform$. Os parâmetros ótimos da SVM no treinamento com UCI e SFA foram $kernel=rbf$, $C=100$ e $gamma=1e-3$.

as amostras, logo, é possível que amostras já vistas pelo classificador durante o treinamento sejam utilizadas na etapa de teste. Por essa razão, os experimentos subsequentes foram realizados somente com o SFA.

4.3 Segundo experimento

Este experimento, de maneira similar ao realizado na seção 4.2, também foi realizado com k -NN e SVM. Porém, o conjunto de dados SFA foi transformado para os espaços de cor HSV, Lab e YCbCr com o objetivo de compreender a influência nos resultados. Além disso, o atributo referente ao componente de luminância foi ignorado para que um teste somente com os componentes de cromaticidade fosse realizado. Novamente, a estratégia escolhida de validação cruzada foi a *10-fold*.

Como o objetivo deste experimento é avaliar a performance do classificador em espaços de cores distintos, os parâmetros ótimos obtidos no experimento 4.2 foram fixados aqui.

Os resultados podem ser vistos na tabela 4.5. Vale ressaltar que o treinamento foi executado com 10 tarefas em paralelo em ambos os classificadores e 30% dos dados, aleatoriamente, foram separados para teste.

Modelo de cores	Classificador	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
RGB	k -NN	0,9672	0,9669	0,9670
	SVM	0,9643	0,9628	0,9638
HSV	k -NN	0,9676	0,9673	0,9674
	SVM	0,9718	0,9677	0,9679
HS	k -NN	0,9215	0,9194	0,9199
	SVM	0,9305	0,9302	0,9302
Lab	k -NN	0,9671	0,9660	0,9670
	SVM	0,9675	0,9665	0,9672
ab	k -NN	0,9444	0,9439	0,9440
	SVM	0,9451	0,9447	0,9446
YCbCr	k -NN	0,9679	0,9677	0,9677
	SVM	0,9635	0,9633	0,9632
CbCr	k -NN	0,9487	0,9482	0,9483
	SVM	0,9496	0,9492	0,9493

Table 4.5: Resultados dos experimentos com k -NN e SVM no conjunto de dados SFA em espaços de cores distintos. As linhas com os modelos de cores HS, ab e CbCr indicam que o componente de luminância não foi utilizado no treinamento.

Os melhores resultados com a SVM e k -NN foram obtidos nos espaços de cores HSV e YCbCr, respectivamente. Dentre aqueles onde o componente de luminância foi ignorado, o CbCr teve a melhor performance.

É interessante observar que os indicadores de qualidade do modelo mostram que ambos classificadores tiveram taxas ao redor de 96% em todos os espaços de cor, quando aplicado com os 3 canais. Além disso, quando o canal de luminância foi ignorado, o desempenho foi inferior - aproximadamente 4 pontos percentuais no pior caso. Por outro lado, a perda de performance pode ser compensada pela redução da complexidade do modelo, já que uma dimensão pode ser descartada, mantendo, ainda, medidas de qualidade relativamente significativas.

4.4 Terceiro experimento

O último experimento foi realizado com a árvore de decisão *fuzzy* proposta por Cintra *et al.* (2013). O conjunto de dados SFA transformado para os espaços de cor HSV, Lab e YCbCr também foi utilizado aqui. Além disso, há um parâmetro que define o nível de confiança que deve ser aplicado no processo de poda da árvore. Por padrão, esse parâmetro assume o valor de 25%, o qual não compromete o desempenho da árvore e evita o fenômeno de sobreajuste. Outra questão relevante neste experimento é a escolha do método que estima a quantidade de conjuntos *fuzzy* por atributo, conforme citado na seção 1.2. A tabela de busca dos parâmetros do estimador ótimo no *FuzzyDT* pode ser vista em 4.6.

Para que os melhores parâmetros pudessem ser introduzidos no treinamento da árvore, foi desenvolvido um algoritmo que testa diversas combinações deles, avaliando a performance do modelo

Conjunto de dados	Nível de confiança				Método				# conjuntos <i>fuzzy</i>
RGB	10	15	20	25	infogain	wm	rf	fixed	2-9
HSV	10	15	20	25	infogain	wm	rf	fixed	2-9
Lab	10	15	20	25	infogain	wm	rf	fixed	2-9
YCbCr	10	15	20	25	infogain	wm	rf	fixed	2-9

Table 4.6: Tabela de busca dos parâmetros do estimador ótimo no *FuzzyDT*. A coluna conjunto de dados, conforme o nome sugere, refere-se aos conjuntos de dados oriundos do SFA em diferentes espaços de cores. A segunda coluna é o nível de confiança aplicado na poda da árvore. A coluna método indica o método de estimativa do número de conjuntos *fuzzy* para treinamento, seguida do número de conjuntos *fuzzy* que devem ser gerados por atributo. Este último parâmetro foi testado em um intervalo entre 2 e 9, conforme citado em Cintra et al. (2011) e é aplicável somente quando o método é *fixed*.

ajustado, de maneira similar à tabela de busca do *scikit-learn*. O algoritmo também tem a capacidade de executar várias instâncias em paralelo. A tabela 4.7 mostra os resultados com a taxa de erro em cada conjunto de dados para o respectivo espaço de cor, bem como os parâmetros que proporcionaram o melhor desempenho.

Os resultados mostram taxas de erros competitivas, do ponto de vista de classificação usando dados intervalares, em relação à modelagem com conjuntos de dados clássicos. Um ponto relevante é que o método fixo de geração dos conjuntos *fuzzy* foi absolutamente unânime dentre todas as quatro técnicas propostas por Cintra et al. (2011), sendo o tamanho dos conjuntos a única peculiaridade, independente do espaço de cor. Além disso, a taxa de erro obtida foi a mesma para todos os níveis de confiança testados para poda da árvore. Isso também ocorreu nas demais combinações de parâmetros, ou seja, naquelas onde a taxa de erro obtida foi maior. Portanto, é possível afirmar que, de fato, o nível de confiança pode ser mantido em 25% para o treinamento do *FuzzyDT*.

Modelo de cores	Taxa de erro	Método	# conjuntos <i>fuzzy</i>
RGB	3,00	fixed	4
HSV	3,23	fixed	3
Lab	2,84	fixed	6
YCbCr	2,72	fixed	8

Table 4.7: Resultados dos experimentos com árvore de decisão *fuzzy* no conjunto de dados SFA em espaços de cores distintos. O algoritmo também mostra o desvio padrão obtido com o estimador ótimo, porém, o valor é 0,00 na maioria dos casos e, por essa razão, não está explícito na tabela.

Além das taxas de erros produzidas pelo *FuzzyDT*, outra saída são as regras induzidas pelo algoritmo que ficam disponíveis para a inferência de novas amostras. Por exemplo, para o conjunto de dados no modelo de cores HSV, cuja taxa de erro foi de 3,23%, foram geradas 9 regras:

1. Se saturação é baixa **então** classe é -1
2. Se saturação é media e intensidade é baixa **então** classe é -1
3. Se matiz é baixa e saturação é media e intensidade é media **então** classe é 1
4. Se matiz é media e saturação é media e intensidade é media **então** classe é -1
5. Se matiz é alta e saturação é media e intensidade é media **então** classe é -1
6. Se matiz é baixa e saturação é media e intensidade é alta **então** classe é 1
7. Se matiz é media e saturação é media e intensidade é alta **então** classe é -1
8. Se matiz é alta e saturação é media e intensidade é alta **então** classe é -1
9. Se saturação é alta **então** classe é -1

Como citado na tabela 4.7, no espaço de cores HSV, o algoritmo gerou um total de 3 conjuntos *fuzzy* por atributo, que são: baixa, media e alta. As regras são arranjadas de tal forma que o usuário pode facilmente interpretá-las por meio de conjunções. O *FuzzyDT* demonstrou robustez e

bom desempenho quando comparado com classificadores clássicos, o que o credencia como um bom candidato na classificação de conjuntos de dados intervalares.

Chapter 5

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 5.1. 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 5.1.

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

5.1 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.

5.2 Cronograma proposto

As atividades listadas na seção 5.1 serão realizadas de acordo com o cronograma disposto na tabela 5.2. 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 5.2: *Cronograma das atividades previstas*

Bibliography

- Abu-Mostafa et al.(2012)** Yaser S. Abu-Mostafa, Malik Magdon-Ismael e Hsuan-Tien Lin. *Learning from data : a short course*. AMLBook.com. Citado na pág. [14](#), [15](#), [16](#), [23](#), [24](#)
- 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](#)
- ao Paulo Brognoni Casati et al.(2013)** Jo ao 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. [23](#), [24](#), [25](#), [26](#)
- 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. [6](#)
- Bezdek et al.(1984)** James C. Bezdek, Robert Ehrlich e William Full. FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2-3):191–203. Citado na pág. [2](#)
- 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. [23](#)
- Bhatt et al.(2009)** Rajen B. Bhatt, Abhinav Dhall, Gaurav Sharma e Santanu Chaudhury. Efficient skin region segmentation using low complexity fuzzy decision tree model. Em *2009 Annual IEEE India Conference*, páginas 1–4. IEEE. Citado na pág. [2](#)
- 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](#), [6](#)
- Campbell(2000)** Colin Campbell. *An Introduction to Kernel Methods*, páginas 155–192. Springer Verlag, Berlin. Citado na pág. [16](#), [17](#)
- 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. [6](#)
- Cintra et al.(2011)** Marcos E. Cintra, Maria C. Monard, Everton A. Cherman e Heloisa de Arruda Camargo. On the estimation of the number of fuzzy sets for fuzzy rule-based classification systems. Em *11th International Conference on Hybrid Intelligent Systems*, páginas 211–216. IEEE. Citado na pág. [2](#), [28](#)
- Cintra et al.(2013)** Marcos E. Cintra, Maria C. Monard e Heloisa de Arruda Camargo. A fuzzy decision tree algorithm based on C4.5. *Mathware & Soft Computing*, 20:56–62. Citado na pág. [2](#), [23](#), [27](#)

- 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
- 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. 31
- 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. 6
- 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. 5
- 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, 5, 6
- Kaur e Kranthi(2012)** Amanpreet Kaur e B. V. Kranthi. Comparison between YCbCr color space and CIE Lab color space for skin color segmentation. *International Journal of Applied Information Systems*, 3(4):30–33. Citado na pág. 6
- 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
- Kononenko(1994)** Igor Kononenko. Estimating attributes: analysis and extensions of RELIEF. Em *European conference on machine learning*, páginas 171–182. Springer. Citado na pág. 3
- Kovac et al.(2003)** Jure Kovac, Peter Peer e Franc Solina. *Human skin color clustering for face detection*, volume 2. IEEE. Citado na pág. 5, 6
- 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. 6
- Lichman(2013)** M. Lichman. UCI machine learning repository. <http://archive.ics.uci.edu/ml>, 2013. Citado na pág. 2, 23
- 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. 5
- Martínez e Benavente(1998)** Aleix Martínez e Robert Benavente. The AR face database. Relatório técnico, Purdue University. Citado na pág. 6, 23
- Miner e Park(2004)** Meredith Miner e Denise Park. Productive aging lab face database. <https://pal.utdallas.edu/facedb/>, 2004. Citado na pág. 23
- 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
- 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. 19, 24, 25, 26
- 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. 23
- 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. 2, 20, 21
- Quinlan(1993)** John Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann. Citado na pág. 2, 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. 6
- 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

- Umano et al.(1994)** Motohide Umano, Hirotaka Okamoto, Itsuo Hatono, Hiroyuki Tamura, Fumio Kawachi, Sukehisa Umedzu e Junichi Kinoshita. Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems. Em *Proceedings of the Third IEEE Conference on Fuzzy Systems. IEEE World Congress on Computational Intelligence.*, páginas 2113–2118. IEEE. Citado na pág. [2](#)
- 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](#), [5](#), [9](#)
- Wang(2003)** Li-Xin Wang. The WM method completed: a flexible fuzzy system approach to data mining. *IEEE Transactions on fuzzy systems*, 11(6):768–782. Citado na pág. [3](#)
- 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. [5](#)