

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Informática

Um Método Automático para o Ajuste de
Segmentadores de Imagens Baseados em Informação
de Textura e Cor

Fernando Henrique Bezerra Cardoso

Dissertação submetida à Coordenação do Curso de Pós-Graduação em
Ciência da Computação da Universidade Federal de Campina Grande
como parte dos requisitos necessários para obtenção do grau de Mestre
em Ciência da Computação.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Modelos Computacionais e Cognitivos

Herman Martins Gomes
(Orientador)

Campina Grande, Paraíba, Brasil

©Fernando Henrique Bezerra Cardoso, 13 de agosto de 2009

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

C268m

Cardoso, Fernando Henrique Bezerra

Um método automático para o ajuste de segmentadores de imagens baseados em informação de textura e cor / Fernando Henrique Bezerra Cardoso – Campina Grande, 2009.

110 f.

Dissertação (Mestrado em Ciência da Computação)- Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.

Referências.

Orientador: Prof. Dr. Herman Martins Gomes.

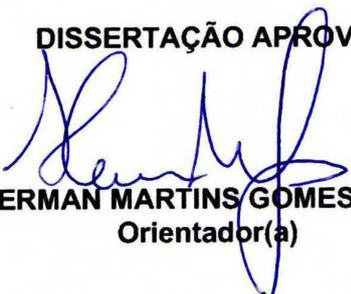
1. Visão Computacional 2. Processamento Digital de Imagens
3. Detecção de Materiais em Fotografia I. Título.

CDU 004.932 (043)

**'UM MÉTODO AUTOMÁTICO PARA O AJUSTE DE SEGMENTADORES DE IMAGENS
BASEADOS EM INFORMAÇÃO DE TEXTURA E COR''**

FERNANDO HENRIQUE BEZERRA CARDOSO

DISSERTAÇÃO APROVADA EM 13.08.2009



HERMAN MARTINS GOMES, PH.D
Orientador(a)



MARCUS COSTA SAMPAIO, DR.
Examinador(a)



JOÃO MARQUES DE CARVALHO, PH.D
Examinador(a)



GEORGE DARMITON DA CUNHA CAVALCANTI, DR.
Examinador(a)

CAMPINA GRANDE - PB

Aos meus pais, Sérgio e Lina.

Agradecimentos

A realização deste trabalho só foi possível graças:

Ao Grande Arquiteto do Universo, criador de todas as coisas, por me dar a vida e força e iluminação para mais esta empreitada.

Aos meus pais, Sérgio e Lina, pela educação, criação, amor e dedicação. Ao meu irmão Maurício, melhor companheiro de república que pude ter, e às minhas avós, Silvia e Tecla, pelo carinho e pelas orações.

À minha noiva – e esposa daqui a bem poucos dias – Fernanda, por ter me apoiado em todos os momentos e por ter permanecido ao meu lado quando as coisas ficaram realmente difíceis. Tenho certeza de que encontrei a pessoa certa e que serei feliz, bem feliz ao seu lado.

Ao meu orientador Herman, pelos ensinamentos e pela compreensão ao me compreender e oferecer sua orientação à distância nos horários em que poderia estar descansando ao lado de sua família.

Aos meus amigos de república, Felipe e Gilson, companheiros de estudos e lazer desde a graduação. E também os amigos que fiz em Campina Grande, Yuri, Flávio, Carlos Augusto, Welflen, Guilherme, Chico, João Bosco, Paulo de Tarso, Bruno Brito. Um obrigado especial a Marcos e Pablo, e ao meu velho amigo Augusto Fontes.

Aos meus queridos colegas de laboratório, Saulo de Tarso e Bruno Alexandre, pelas pesquisas que fizemos juntos, e pelo empenho em juntos desenvolver a Visão Computacional. Também a Camilla Falconi, pela importante ajuda ainda no início do projeto da dissertação.

Aos professores Fábio Soares, Domingos Machado, Montserrat Fortuny, Alexandre Santos, José Maria Rodrigues e Christiane Leal, pela preparação que me deram durante minha graduação.

Ao professor Marcus Sampaio, pela diversificação acadêmica que me proporcionou ao me convidar para trabalharmos juntos no campo da Mineração de Dados.

Aos colegas do curso de formação e da TIC-SEAL da Petrobras, pela compreensão por minhas ausências devido à escrita da dissertação e às reuniões de orientação.

Ao CNPq e à HP, pelo apoio financeiro.

Resumo

O ajuste de segmentadores para a detecção de materiais específicos que compõem o conteúdo retratado por uma imagem digital é usualmente executado de forma manual, ou com pouco grau de automatização. A obtenção de segmentadores com base em aprendizagem normalmente envolve os seguintes passos: construir um conjunto de imagens segmentadas manualmente; analisar os melhores parâmetros - espaço de cores, profundidade de cor, tamanho da janela de *pixels* a ser considerada, dentre outras; ajustar um classificador que decide se cada *pixel* faz parte ou não da superfície do material. Uma vez que esta segmentação é geralmente um passo intermediário de inúmeras aplicações das áreas de Visão Computacional e de Processamento de Imagens, uma fraca automatização do processo faz com que esforço seja investido em uma tarefa secundária. Neste trabalho, é proposto um método de automatização do ajuste de segmentadores de materiais baseados em informação de cor e textura, utilizando redes neurais artificiais como estruturas de classificação, filtros de Gabor como descritores de textura e um algoritmo *floodfill* adaptado como etapa de pós-processamento. A descoberta dos melhores parâmetros dos filtros de Gabor e do algoritmo *floodfill* é realizada através da aplicação de um algoritmo genético. Uma avaliação experimental envolvendo a detecção de 3 classes de materiais (pele humana, grama e céu) foi realizada, e os segmentadores obtidos com o método proposto apresentaram resultados melhores (com relação à Medida-F) do que o método geral *Naïve Bayes* bem como métodos específicos para cada tipo de material, tendo sido demonstrada a generalidade do método proposto.

Abstract

The tuning of image segmenters to detect different materials captured in a digital image is usually performed in a manual fashion or taking advantage of little automation. Obtaining such segmenters using a machine learning technique usually involves the following tasks: building an manually segmented image set; analyzing the best parameters – color space, color depth, window size, among others; training a classifier that decides whether a pixel belongs or not to the surface of the material. Since segmentation is usually an intermediate step to many applications of Computer Vision and Image Processing, the lack of automation leads to effort lost in a secondary task. In this work, we propose a method to automatically adjust image segmenters based on color and texture information, using artificial neural networks as classifiers, Gabor filters as texture descriptors and an adapted floodfill algorithm as a post-processing step. The parameters of the Gabor filters and of the adapted floodfill are determined using a genetic algorithm. An experimental evaluation has been conducted, using our method to adjust segmeters for three different materials (human skin, grass and blue sky) and the results obtained are better (regarding the F-Measure) than than the results of general Naive Bayes method, as well as the results of specific segmenters for each material, thus supporting the generality of the proposed method.

Conteúdo

1	Introdução	1
1.1	Descrição do Problema	3
1.2	Motivação	3
1.3	Objetivos	5
1.4	Organização do Documento	5
2	Revisão Bibliográfica	7
2.1	Trabalhos Relacionados	7
2.2	Discussão sobre Segmentação de Imagens e Detecção de Materiais	9
2.3	Considerações Finais	16
3	Método Proposto	18
3.1	Visão Geral do Método Proposto	18
3.2	Segmentador Neural Baseado em Informação de Cor	19
3.2.1	Treinamento de um Segmentador Neural	20
3.2.2	Obtenção do Segmentador Neural Baseado em Informação de Cor	27
3.3	Segmentador Baseado em Informação de Textura e Cor	30
3.4	Expansão de Regiões para Recuperação da Medida-F	34
3.5	Detalhes de Implementação do Protótipo	37
3.6	Considerações Finais	39
4	Avaliação Experimental	40
4.1	Detecção de Grama em Quadros de Vídeos Desportivos	40
4.1.1	Conjunto de Imagens de Treinamento	41
4.1.2	Obtenção Automática do Segmentador	42

4.1.3	Teste do Segmentador de Grama	50
4.2	Detecção de Pele Humana em Fotografias	52
4.2.1	Conjunto de Imagens de Treinamento	53
4.2.2	Obtenção Automática do Segmentador	53
4.2.3	Teste do Segmentador de Pele	58
4.3	Detecção de Céu Azul em Fotografias	60
4.3.1	Conjunto de Imagens de Treinamento	62
4.3.2	Obtenção Automática do Segmentador	63
4.3.3	Teste do Segmentador de Céu Azul	67
4.4	Considerações Finais	68
5	Conclusão	72
	Referências Bibliográficas	82
A	Conceitos Fundamentais	83
A.1	Definições Iniciais	83
A.2	Métricas de Avaliação dos Segmentadores	84
A.2.1	Taxa de Detecção (<i>Recall</i>)	85
A.2.2	Precisão	86
A.2.3	Medida-F (<i>F-measure</i>)	87
A.3	Técnicas Utilizadas no Método Proposto	87
A.3.1	Redes Neurais Artificiais	88
A.3.2	Algoritmos Genéticos	95
A.3.3	Filtros de Gabor	103
B	Transformações entre Espaços de Cores	107
B.1	De RGB para CIE XYZ	107
B.2	De RGB para YCbCr	107
B.3	De RGB para HSV	108
B.4	De RGB para HSL	108
B.5	De RGB para CIE L*a*b*	109
B.6	De RGB para CIE L*u*v	110

B.7 De RGB para tons de cinza	110
---	-----

Lista de Figuras

2.1	Exemplo de segmentação das regiões correspondentes a grama.	10
2.2	Exemplo de imagem (ao centro) decomposta em canais de espaços de cores diferentes (RGB acima e HSV abaixo).	12
2.3	Histograma dos pixels da base de imagens de céu azul, com respeito à distância relativa ao topo das imagens.	14
2.4	Comparação visual entre diferentes texturas.	15
2.5	Comparativo entre aplicações isoladas e combinadas de Filtros de Gabor . . .	16
3.1	Visão geral do método proposto	18
3.2	Distribuições dos pixels de grama (verdes) e não-grama (vermelhos), considerando possíveis separações a partir do uso de canais de cores verde, azul e vermelho.	20
3.3	Representação da arquitetura das redes MLP no âmbito desta dissertação. . .	21
3.4	Exemplo de imagem no conjunto de treinamento.	22
3.5	Efeito do filtro de borragem gaussiana (janela de 9x9 pixels)	23
3.6	Distâncias consideradas quando se utiliza o posicionamento dos pixels . . .	25
3.7	Principais passos da obtenção do segmentador neural baseado em informação de cor	28
3.8	Resultados parciais da segmentação baseada em informação de textura e cor.	32
3.9	Algoritmo <i>floodfill</i> adaptado e demonstração dos passos intermediários . . .	35
3.10	Algoritmo <i>floodfill</i> executado para detecção de grama.	36
4.1	Exemplo de imagem do conjunto de treinamento para detecção de grama, com sua respectiva segmentação manual.	41

4.2	Histograma dos pixels de grama base de imagens de partidas de futebol, com respeito à distância relativa ao topo das imagens.	47
4.3	Histograma dos pixels de grama de todas as imagens da base de partidas de futebol, com respeito à distância relativa à margem esquerda das imagens.	48
4.4	Resultados da aplicação segmentador obtido utilizando o método proposto ao problema da detecção de grama (grama em branco e demais regiões em preto).	50
4.5	Histograma dos pixels de pele na base de imagens de treinamento, com respeito à distância relativa ao topo das imagens.	56
4.6	Histograma dos pixels de pele na base de imagens de treinamento, com respeito à distância relativa à margem esquerda das imagens.	56
4.7	Resultados da aplicação segmentador obtido utilizando o método proposto ao problema da detecção de pele (pele em branco e demais regiões em preto).	59
4.8	Exemplo de imagem do conjunto de treinamento para detecção de céu azul, com sua respectiva segmentação manual.	62
4.9	Histograma dos pixels de céu na base de imagens de treinamento, com respeito à distância relativa à margem esquerda das imagens.	66
4.10	Resultados da aplicação segmentador obtido utilizando o método proposto ao problema da detecção de céu (céu em branco e demais regiões em preto).	71
4.11	Evolução dos resultados da segmentação antes e depois da aplicação do <i>floodfill</i> adaptado	71
A.1	Exemplo de imagem e sua respectiva segmentação manual para composição da base de treinamento para o segmentador de grama.	84
A.2	Exemplo de imagem e sua respectiva segmentação manual para composição da base de treinamento para o segmentador de grama.	85
A.3	Exemplo da utilização de segmentadores hipotéticos na imagem da Figura A.1(a).	86
A.4	Neurônio Biológico - Fonte: Adaptado de [33]	90
A.5	Neurônio Artificial - Fonte: Adaptado de [16]	91
A.6	Algumas das funções de ativações utilizadas nas redes neurais MLP.	92

A.7 Rede Neural do tipo <i>Perceptron</i> de Múltiplas Camadas.	93
A.8 Aprendizado Supervisionado. Fonte: adaptado de [16]	93
A.9 Representação do Método da Roleta	100
A.10 Cruzamento genético de dois indivíduos	102
A.11 Partes real e imaginária e magnitude um filtro de Gabor	104
A.12 Exemplo de imagem e seu resultado após a aplicação do filtro de Gabor . . .	105
A.13 Diferentes orientações de filtros de Gabor	105
A.14 Diferentes escalas de filtros de Gabor	106

Lista de Tabelas

3.1	Métricas extraídas dos resultados parciais da segmentação baseada em informação de textura e cor.	32
3.2	Resultados parciais da segmentação baseada em informação de textura e cor.	36
4.1	Evolução das estratégias de treinamento.	42
4.2	Determinação do melhor espaço de cor para segmentação de grama.	44
4.3	Determinação da melhor profundidade de cor para segmentação de grama (YCbCr).	45
4.4	Determinação do melhor tamanho da janela de <i>pixels</i> para segmentação de grama (YCbCr, 8 <i>bits</i> por canal).	45
4.5	Determinação da melhor quantidade de neurônios na camada oculta para segmentação de grama (YCbCr, 8 <i>bits</i> por canal, janela de 7×7 <i>pixels</i>).	46
4.6	Determinação dos parâmetros do segmentador baseado em informação de textura e cor para detecção de grama.	49
4.7	Determinação dos parâmetros para expansão de regiões na detecção de grama.	49
4.8	Resultados obtidos para detecção de grama	51
4.9	Determinação do melhor espaço de cor para detecção de pele.	53
4.10	Determinação da melhor profundidade de cor para detecção de pele (YCbCr).	54
4.11	Determinação do melhor tamanho da janela de <i>pixels</i> para detecção de pele (YCbCr, 6 <i>bits</i> por canal).	54
4.12	Determinação da melhor quantidade de neurônios na camada oculta para detecção de pele (YCbCr, 6 <i>bits</i> por canal, janela de 3×3 <i>pixels</i>).	55
4.13	Determinação dos parâmetros do segmentador baseado em informação de textura e cor para detecção de pele.	57

4.14	Determinação dos parâmetros para expansão de regiões na detecção de pele.	58
4.15	Resultados obtidos para detecção de pele	60
4.16	Determinação do melhor espaço de cor para detecção de céu.	63
4.17	Determinação da melhor profundidade de cor para detecção de céu ($L^*a^*b^*$).	63
4.18	Determinação do melhor tamanho da janela de <i>pixels</i> para detecção de céu ($L^*a^*b^*$, 7 <i>bits</i> por canal).	64
4.19	Determinação da melhor quantidade de neurônios na camada oculta para detecção de céu ($L^*a^*b^*$, 7 <i>bits</i> por canal, janela de 7×7 <i>pixels</i>).	65
4.20	Determinação dos parâmetros do segmentador baseado em informação de textura e cor para detecção de céu.	66
4.21	Determinação dos parâmetros para expansão de regiões na detecção de céu.	67
4.22	Resultados obtidos para detecção de céu azul	68
4.23	Resumo dos Resultados	69
4.24	Tempos médios de treinamentos dos segmentadores	70
A.1	Comparativo entre as computações convencional a e neural. Fonte: Adaptado de [51].	88
A.2	Emparelhamento de Reprodutores	101

Lista de Algoritmos

3.1	Exploração do melhor espaço de cores	28
3.2	Exploração da melhor profundidade de cores	29
A.1	Algoritmo genético básico	99

Capítulo 1

Introdução

A segmentação de imagens em regiões compostas por determinados tipos de material (pele humana, cabelos, grama, asfalto, céu, entre outros) é um passo importante na resolução de uma série de problemas da Visão Computacional.

A detecção de pele humana, por exemplo, pode restringir o espaço de busca de algoritmos de localização de faces [14,22,37] ou fornecer informação semântica de imagens que indica a presença de pornografia [26]. Já a detecção de céu em fotografias é um passo intermediário para aplicações de categorização entre imagens ao ar livre e imagens em ambientes fechados e de determinação da posição do horizonte [13]. Um outro exemplo é a detecção de grama em quadros de vídeos desportivos, que é utilizada para facilitar a sua sumarização [11,58].

A obtenção de segmentadores de imagens capazes de detectar materiais tipicamente envolve a seguinte sequência de passos:

1. Construir um conjunto de imagens segmentadas manualmente [26,13,11] ou, eventualmente, utilizar uma base pública de imagens [47,56];
2. Analisar os melhores parâmetros – espaço de cores, profundidade de cor, tamanho da janela de *pixels* a ser considerada, dentre outras – para detectar o material em questão [37];
3. Ajustar um classificador que decide se cada *pixel* faz parte ou não da superfície do material. Diversas técnicas de classificação podem ser aplicadas para separar as regiões de um determinado material, tais como os classificadores Bayesianos [24, 42], Gaussianos [42,46], o algoritmo *Expectation/Maximization* [10] e redes neurais

artificiais do tipo *Perceptron* de Múltiplas Camadas (MLP - *Multi Layer Perceptron*) [42]; e

4. Testar o segmentador para verificar a sua qualidade.

Estas tarefas normalmente são feitas com pouca ou nenhuma automatização, uma vez que, após extensiva revisão da literatura, não foi encontrado um método automático para obtenção e ajuste de segmentadores desta natureza. Na falta de tal método, é necessário investir tempo e esforço para uma atividade secundária durante um projeto em Visão Computacional, que poderiam ser investidos em outras atividades que contribuem mais diretamente para o objetivo final do projeto.

Desta maneira, a automatização dos processos de obtenção e ajuste de segmentadores introduz uma maior agilidade para alcançar resultados nos projetos de pesquisa em Visão Computacional que necessitam da detecção de materiais como passo intermediário, não sendo mais necessário desviar esforços para resolver um problema periférico, nem repetir os trabalhos de treinamento e de ajuste a cada problema que se deseja resolver.

A fim de permitir a automatização dessas tarefas, este trabalho tem como principal objetivo a proposta de um método genérico de obtenção de segmentadores de regiões baseados em informação de cor e textura para detecção de materiais.

O único subsídio utilizado para originar o segmentador de materiais com o método proposto neste trabalho é o conjunto de características de textura e cor extraído de um conjunto de imagens manualmente rotuladas. Isto retira do usuário a responsabilidade de parametrizar os segmentadores para cada tipo de material que se deseja detectar.

Vale ressaltar que automatização não exime o desenvolvedor de sistemas de visão computacional da necessidade de construção de uma base de dados de imagens com distribuições de amostras positivas e negativas de materiais que sejam estatisticamente significativas, do rotulamento de regiões das imagens dessa base de dados e da implementação de processos que garantam a qualidade do rotulamento (uma vez que tipicamente é efetuado de forma manual).

1.1 Descrição do Problema

O problema abordado por este trabalho pode ser enunciado da seguinte maneira: “Que passos devem ser seguidos para que seja obtido um bom segmentador de imagens para detecção de materiais a partir de um conjunto de imagens em que as regiões do material em questão estão marcadas de maneira confiável?”

Este enunciado apresenta algumas particularidades. A primeira delas, e que deve ser muito bem enfatizada, é que a resposta à pergunta é uma sequência de passos a serem seguidos – um método – e não o segmentador em si. Desta forma o objetivo desta dissertação não é obter um segmentador genérico, mas propor um método genérico para obtenção de segmentadores específicos.

A segunda particularidade é que esta pergunta não necessariamente exige que os segmentadores obtidos através da utilização do método proposto sejam ótimos, ou os melhores dentre os seus concorrentes. Bons segmentadores – cujos resultados sejam compatíveis com os resultados dos métodos específicos para cada tipo de material – já são suficientes para satisfazer o enunciado do problema.

A verificação da qualidade dos segmentadores obtidos através da utilização do método proposto é realizada por meio de três estudos de caso: detecção de grama em quadros desportivos, detecção de pele humana em fotografias e detecção de céu azul em fotografias. Nestes estudos de caso, tanto o segmentador fornecido pelo método quanto outros segmentadores extraídos da literatura científica são testados em uma mesma base de testes e os resultados são comparados.

Por fim, há a exigência de que os conjuntos sejam marcados manualmente de maneira confiável, ou seja, a qualidade dos segmentadores depende diretamente da qualidade da segmentação manual realizada na base de imagens de treinamento dos segmentadores .

1.2 Motivação

Por serem normalmente executadas manualmente e não serem o objetivo final da maioria das aplicações de visão computacional em que se inserem, as tarefas necessárias à obtenção de segmentadores de materiais utilizam recursos humanos que poderiam ter sido melhor

aplicados na resolução do problema central, na hipótese de ser possível a automatização do ajuste do segmentador.

Desta forma, a principal motivação para a realização deste trabalho é diminuir o tempo exigido nas tarefas que compõem o processo de obtenção de segmentadores de imagens considerando diferentes tipos de materiais, exceto pela segmentação da base de treinamento, que inevitavelmente deve ser feita de maneira manual.

A automatização introduzida pelo método proposto permitirá uma maior agilidade na obtenção dos segmentadores, uma vez que os processos manuais se tornarão desnecessários. Desta forma, os recursos que antes estariam destinados à obtenção de segmentadores podem ser utilizados de outra forma mais produtiva, aumentando o foco no problema principal e, por consequência, as suas chances de sucesso.

Sem um método de automatização, é necessário realizar sob supervisão humana a investigação, por exemplo, de diversos espaços de cor em que podem ser codificadas as fotografias digitais, verificando a concentração dos padrões de cor pertencentes ao material em regiões específicas do espaço, aumentando a separabilidade das regiões pertencentes ou não ao material.

Outros aspectos da amostragem de *pixels* que influem na separabilidade das regiões, como a profundidade de cor em que a amostragem será realizada, além do tamanho da janela de *pixels* que será utilizada como subsídio extra para a composição da amostragem, também precisam ser verificados sob supervisão humana.

Todas estas tarefas passarão a ser desnecessárias caso se verifique a possibilidade de automatizar a obtenção dos segmentadores de materiais, além de outras como a parametrização da técnica de aprendizado de máquina empregada, análise da influência da textura do material na separabilidade de suas regiões e a partição do conjunto total de amostras em conjuntos homogêneos e representativos do todo para treinamento, validação e teste.

Além disto, a própria natureza do trabalho realizado exige que sejam produzidos segmentadores para alguns tipos de material, um para cada estudo de caso, uma vez que a validação será realizada através deste mecanismo. A disponibilização destes segmentadores também justifica a produção deste estudo.

1.3 **Objetivos**

O objetivo geral desta dissertação é propor um método de automatização da obtenção e do ajuste de segmentadores de imagens baseados em informação de cor e textura para detecção de materiais em fotografias digitais.

A proposta consiste em uma sequência bem definida de passos a serem seguidos – e que possam ser implementados na forma de um programa de computador – para que se possa obter um segmentador com métricas de avaliação aceitáveis a partir de um conjunto de imagens rotulado manualmente para o material que se deseja detectar.

Os objetivos específicos são:

- Levantar na bibliografia científica os aspectos das imagens que são relevantes para promover a detecção de materiais em fotografias;
- Implementar um protótipo de programa de computador que automatiza a obtenção e o ajuste dos segmentadores, seguindo o método proposto nesta dissertação;
- Validar o método proposto, através da aplicação do protótipo para obter segmentadores para os materiais dos estudos de caso desta dissertação e da comparação dos seus resultados com técnicas presentes na literatura; e
- Obter segmentadores para cada material dos estudos de caso, que podem ser utilizados posteriormente em aplicações de Visão Computacional.

1.4 **Organização do Documento**

O restante deste documento está organizado em mais quatro capítulos. O Capítulo 2 apresenta o resultado da revisão bibliográfica realizada a fim de fundamentar o método proposto nesta dissertação. Trata-se, portanto, da descrição de trabalhos relacionados e com objetivos semelhantes aos desta dissertação e da reunião de aspectos das imagens que já foram utilizados na literatura como insumo para detecção de materiais.

Em seguida, o método proposto nesta dissertação é descrito em mais detalhes no Capítulo 3 e avaliado experimentalmente na forma de estudos de casos relatados no Capítulo 4.

Finalizando o documento, o Capítulo 5 expõe as conclusões do trabalho realizado e as sugestões de trabalhos futuros que agregarão valor à proposta apresentada nesta dissertação.

Há ainda mais dois apêndices: no Apêndice A são discutidos os aspectos mais relevantes da fundamentação teórica em que se baseia o método proposto nesta dissertação e definidos os termos e métricas utilizados no decorrer do texto. No Apêndice B estão detalhadas as equações utilizadas para realizar transformação entre espaços de cores no protótipo implementado.

Capítulo 2

Revisão Bibliográfica

Neste capítulo, é apresentado o resultado da revisão bibliográfica realizada a fim de fundamentar o método proposto nesta dissertação. Para organizar melhor a leitura, o capítulo está dividido em uma seção de trabalhos relacionados e, em seguida, uma seção para discussão de aspectos relativos às tarefas de segmentação de imagens e detecção de materiais, apresentando trabalhos relacionados a estas tarefas. Por fim, são apresentadas as considerações finais sobre a revisão bibliográfica realizada.

2.1 Trabalhos Relacionados

Devido à especificidade do objetivo geral deste estudo, que é propor um método para automatizar a obtenção e o ajuste de segmentadores de imagens baseados em informação de textura e cor, não foi encontrado nenhum trabalho com exatamente o mesmo fim, mesmo após a realização de exaustiva revisão da bibliografia.

Entretanto, foram encontrados trabalhos contendo alguns aspectos que se assemelham de alguma forma com o tema abordado nesta dissertação, conduzidos pelos pesquisadores do projeto PERMM [2], que propõem uma série de aplicações de recuperação de imagens baseada em conteúdo.

Em um relatório técnico intitulado “*Content Based Image Retrieval Using Semantic Visual Categories*” [54], apresenta-se uma técnica de recuperação de imagens a partir de primitivas semânticas de alto nível, como “a imagem possui grama na parte esquerda e areia na parte direita”. As principais contribuições deste relatório são demonstrar que características

visuais de alto nível são importantes como fator determinante da relevância na recuperação das imagens, além de permitirem a construção de interfaces amigáveis de busca.

A recuperação de imagens utilizando essa técnica é dividida em três estágios: (i) segmentação das imagens, (ii) classificação utilizando redes neurais artificiais e (iii) composição das chaves de busca.

A segmentação das imagens nessa técnica aplica a abordagem de similaridade de cores utilizando a técnica de diagramas de Voronoi [49]: cria-se um diagrama de Voronoi a partir das bordas detectadas da imagem e os pontos de pico deste diagrama são utilizados como sementes para o crescimento de regiões similares. São utilizados extratores de informação de cor e textura nestas regiões para compor os respectivos vetores de características.

Em seguida, são treinadas várias redes neurais artificiais do tipo MLP, cada uma especializada na detecção de um tipo de material – e também do tipo RBF (*Radial Basis Function*) [16], estas apresentando maiores taxas de erro nas classificações. Para compor os conjuntos de treinamentos das redes neurais, amostras dos materiais em que as redes são especializadas foram rotuladas manualmente. Os materiais escolhidos foram: tijolos, nuvens, pelagem animal, grama, gelo, estradas, rochas, areia, pele, árvores e água. Os percentuais de regiões corretamente classificadas pelas redes neurais variou de aproximadamente 86% para tijolos a aproximadamente 98% para grama.

Para facilitar o uso da ferramenta, é apresentada uma interface que permite a composição de chaves de busca a partir de um esboço do que se deseja recuperar: o usuário indica a posição dos materiais que deseja encontrar nas imagens a partir de elementos de textura que compõem um “tesauro de materiais”. Desta forma, é possível fazer um esboço de uma imagem que contenha “grama na parte esquerda e areia na parte direita”. Além disto são apresentadas as formas tradicionais de busca de imagens em sistemas de recuperação: envio de uma imagem e recuperação de imagens similares, ou indicação textual das características que se deseja encontrar.

Esta aplicação ainda é auxiliada pela linguagem Oquel [53], especializada em consultas a bases de imagens, que permite a construção de consultas com uma sintaxe baseada no conceito de ontologias, o que a torna bastante flexível, permitindo também consultas como “*some sky which is close to trees in upper corner, size at least 20%*” (imagens que mostram o céu perto de árvores nos cantos superiores, ocupando ao menos 20% da área da imagem).

O que diferencia o projeto PERMM da solução proposta nesta dissertação é o fato de que esta consiste em um método genérico para a obtenção e ajuste de segmentadores de imagens, enquanto o projeto PERMM, por sua vez, apresenta uma aplicação composta de diversos segmentadores, que são capazes de detectar os materiais citados, mas não oferece mecanismos conhecidos de indução de segmentadores para novos tipos de materiais a partir de um conjunto de treinamento previamente segmentado de forma manual.

Além disto, a técnica de segmentação aplicada no projeto PERMM se utiliza de uma abordagem para detectar bordas e leva em consideração as formas das regiões antes de extrair as informações de cor e textura [54]. Isto torna mais robusta a segmentação de regiões, porém adiciona uma maior complexidade ao processo.

Por fim, apesar de serem utilizadas técnicas complexas de extração de características, como descritores de densidade, matrizes de co-variância e momentos da imagem (características de forma), não são explorados aspectos básicos como diferentes espaços de cores (apenas o RGB – *Red, Green, and Blue* – é considerado) e redução da profundidade de cor que, comprovadamente, ajudam na separabilidade das regiões de alguns materiais, conforme explicado a seguir.

2.2 Discussão sobre Segmentação de Imagens e Detecção de Materiais

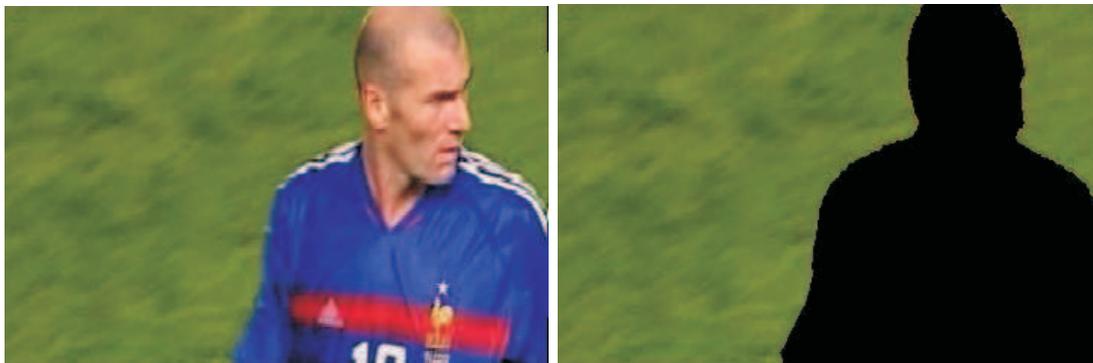
Diversas aplicações de visão computacional, em especial na área de processamento de imagens digitais, exigem uma prévia segmentação da imagem com que se pretende trabalhar. Segmentar uma imagem significa subdividi-la em regiões ou objetos constituintes da mesma [15]. Isto é importante para reduzir o espaço de busca na execução da análise da imagem, além de permitir tratar a imagem como um conjunto de regiões separadas em vez de um único objeto.

A segmentação de uma imagem pode ser executada através da detecção de descontinuidades ou por similaridade entre regiões. Na primeira abordagem, mudanças bruscas nos tons de cinza da imagem são localizadas, em busca de pontos e linhas isoladas para permitir a composição das bordas das regiões encontradas. Após este processo, algoritmos de ligação de bordas são utilizados para compor fronteiras e finalizar, desta maneira, a segmentação em

regiões.

Na abordagem por similaridade, são utilizados algoritmos de crescimento de regiões, como o *floodfill* [15] para separar a imagem em regiões de cor similar, de acordo com uma determinada métrica preestabelecida de distância. Também podem ser utilizadas técnicas de divisão e fusão de regiões [20], a fim de melhorar os resultados do processo de segmentação.

A segmentação que se pretende executar com o método proposto neste trabalho consiste em identificar e isolar, na imagem fornecida como entrada, as regiões que correspondem a um determinado material (como, por exemplo, pele, grama, céu) a partir de informações de cor e textura (a Figura 2.1 é um exemplo de segmentação por grama).



(a) Imagem original.

(b) Imagem segmentada.

Figura 2.1: Exemplo de segmentação das regiões correspondentes a grama.

Aspectos Relevantes à Detecção de Materiais

A seguir, serão apresentadas as características das imagens que são relevantes para a detecção de materiais, por promoverem uma maior separabilidade entre as regiões do material que se deseja detectar e o restante da imagem.

Estas características serão utilizadas no método proposto por esta dissertação, a fim de se obter um segmentador que utiliza a melhor combinação das características apresentadas.

Espaço de Cor

Uma vez que os segmentadores induzidos pelo método proposto neste trabalho executarão a classificação de cada um dos *pixels* de uma imagem de maneira isolada, como é feito nos segmentadores de materiais *pixel a pixel* [48,58] – em contraste com os segmentadores

baseados em regiões [63,64] –, uma parte dos vetores de características utilizados como entrada da Rede Neural MLP associada ao segmentador será composta pelos valores de componentes de cor no espaço de cores adotado.

De acordo com Gonzales e Woods, um espaço de cores é “uma especificação de um sistema de coordenadas tridimensionais e um subespaço dentro deste sistema onde cada cor é representada por um único ponto” [15]. Em outras palavras, é uma forma de codificar espacialmente os fenômenos físicos da intensidade e do comprimento das ondas de luz que compõem a imagem em questão.

As imagens capturadas para segmentação são codificadas no espaço de cores RGB, originário dos sistemas de exibição de imagens que utilizavam tubos de raios catódicos, que é utilizado em diversas aplicações de detecção de materiais [26,11,10,42].

No entanto, este espaço de cores apresenta problemas, como a grande correlação entre os valores dos canais de cor e a mistura entre crominância e luminância nas 3 componentes. Diante do exposto, a depender do tipo de material que se queira segmentar, seria de grande utilidade efetuar uma transformação de espaço de cores para aumentar a separabilidade das regiões correspondentes à superfície do material em questão e às que não correspondem à mesma [48]. A Figura 2.2 exibe uma imagem (ao centro) decomposta nos espaços de cores RGB e HSV [42,65].

Outro motivo para executar a transformação de espaço de cores é desconsiderar o canal de luminância com o objetivo de anular o efeito das sombras e das ilhas de iluminação. Para tanto, espaços de cor que contêm um canal específico para luminância – YCbCr (*Luma, Blue-difference, and Red-difference*) [42,65], YES, YIQ e YUV (os dois últimos são sistemas de cor empregados na codificação de transmissões de televisão nos padrões NTSC e PAL) – ou espaços baseados em matiz e saturação – HSV, HSL e HSI (*Hue, Saturation, and Value, Lighthness, Intensity*) – podem ser utilizados, analisando apenas as componentes de cor, desconsiderando os canais de luminância, valor, brilho ou intensidade (Y, L, I ou V), conforme o espaço adotado.

Além disto, podem ser utilizados canais de cores menos usuais e que surgiram com o intuito de modelar a percepção das cores pelos seres humanos, como o CIE XYZ e suas transformações perceptualmente uniformes CIE L*a*b* [42,65] e CIE L*u*v* [56].

Vale ressaltar que não é para todos os tipos de materiais que a utilidade da transformação



Figura 2.2: Exemplo de imagem (ao centro) decomposta em canais de espaços de cores diferentes (RGB acima e HSV abaixo).

de espaço de cores se verifica [1,18,24,48,63]. Por este motivo, opta-se nestes casos por não realizar a transformação de espaço de cor para poupar o processamento necessário para tal tarefa.

No método genérico de obtenção de segmentadores aqui proposto, além de serem explorados segmentadores que tratam a informação de cor dos *pixels* no espaço de cores RGB, serão explorados outros espaços de cores, a fim de otimizar a segmentação. Os espaços de cores utilizados são (além do RGB): HSV , YCbCr, CIE XYZ, HSL, CIE $L^*a^*b^*$ e CIE $L^*u^*v^*$.

Profundidade de Cor

Além das transformações de espaços de cores, seus valores podem ser quantizados em diferentes profundidades de cor, isto é, diferentes quantidades de *bits* por canal de cor. Em geral, as cores no espaço RGB são codificadas com 8 *bits* para cada canal. Entretanto, quando a profundidade é reduzida, há uma tendência de homogeneização nas cores (cores similares em 8 *bits* são consideradas iguais com codificações de menor profundidade). Jones e Regh

[26], em seu trabalho de detecção de pele humana em fotografias, utilizaram de 4 a 8 *bits* para cada canal do espaço de cores RGB e obtiveram melhores resultados com 5 *bits* por canal.

Jayaram *et al.* [24], como parte de seu estudo acerca dos efeitos da modelagem de cor na capacidade de detectar pele humana em fotografias também exploraram profundidades para cada um dos canais dos espaços de cores abordados. Em seu estudo, foram obtidos segmentadores ótimos utilizando de 5 a 7 *bits* por canal, a depender do espaço de cores utilizado.

Posicionamento dos *Pixels*

Em geral, cada material presente em uma fotografia tem uma certa tendência a estar posicionado em uma determinada região da imagem. Este fato está ligado à natureza de cada material: nuvens tendem a estar na parte superior; grama, água e asfalto nas regiões inferiores das imagens.

Zafarifar *et al.* [64,62,63] propõem a geração de um modelo probabilístico da posição vertical dos *pixels* para detecção de regiões de céu azul em fotografias. Este modelo se baseia no processamento de um conjunto de imagens, e atribui 100% de probabilidade aos *pixels* do topo da imagem e decresce verticalmente, com taxa de variação determinada pela coordenada vertical média dos *pixels* de céu.

Isto faz sentido, ao se pensar que o céu provavelmente estará no topo da maioria das imagens que o contém, e pode ser observado na Figura 2.3, que apresenta um histograma dos *pixels* de céu com respeito à distância relativa ao topo das imagens, computado a partir da base de treinamento utilizada no estudo de caso de detecção de céu apresentado na Seção 4.3

No histograma apresentado, verifica-se que há uma concentração da área correspondente a céu mais perto do topo das imagens, e que as ocorrências vão diminuindo conforme a distância até o topo vai aumentando. Isto indica que incluir o posicionamento vertical no vetor de características apresentado ao segmentador ajuda a obter melhores resultados neste caso.

No entanto, é surpreendente notar que não foi encontrada, no âmbito da revisão bibliográfica realizada, nenhum outro relato de trabalho que levasse em consideração posiciona-

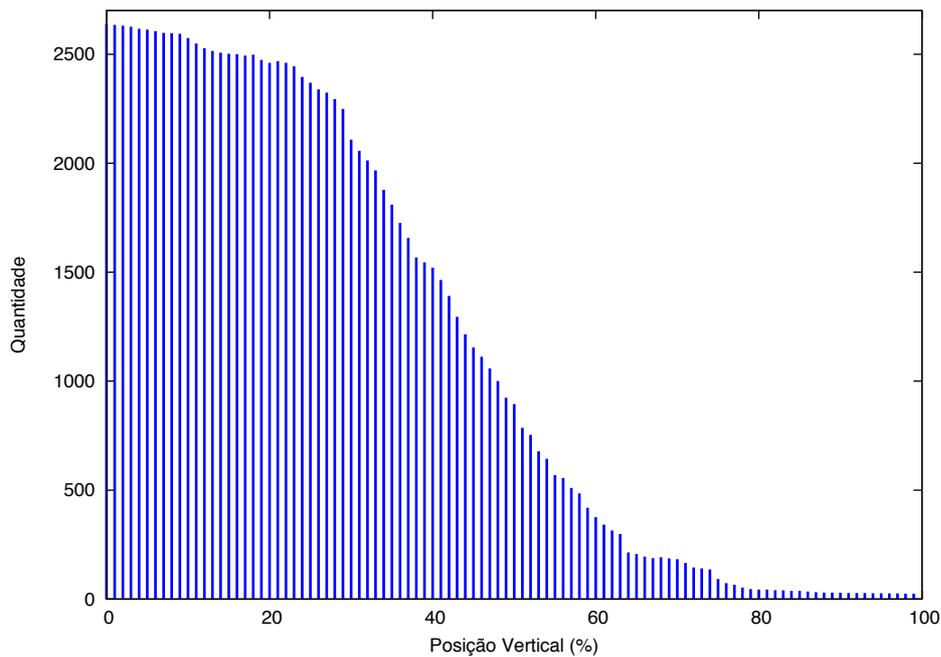


Figura 2.3: Histograma dos pixels da base de imagens de céu azul, com respeito à distância relativa ao topo das imagens.

mento dos *pixels* para detectar materiais, uma vez que há outros materiais que apresentam tal característica.

Nesta dissertação, é proposto um critério sistemático para decidir pela utilização ou não do posicionamento vertical ou horizontal – ou ambos – a depender do material que se deseja detectar. Com isto é verificada a relevância deste aspecto para a separabilidade dos materiais.

Textura

Apesar de não haver um consenso na literatura técnica acerca da definição clara e formal de textura, a maioria dos autores concorda que ela pode ser definida como o efeito visual causado pelas mudanças de intensidade nos *pixels* da imagem [55]. Em outras palavras, a textura é um padrão homogêneo resultante da presença de pequenas variações de uma intensidade de cor que caracterizam certos tipos de materiais.

No contexto da segmentação de imagens por características de materiais, é de extrema necessidade levar em consideração a textura: tons de verde, por exemplo, podem confundir um segmentador de grama que leva em consideração apenas a cor, mas dificilmente farão o mesmo, caso a textura seja levada em consideração (ver Figura 2.4).

Para a visão humana, reconhecer texturas é um processo fácil, diferentemente do que



(a) Tom de verde sem textura.

(b) Tom de verde com textura de grama.

Figura 2.4: Comparação visual entre diferentes texturas.

ocorre nos processos automáticos, uma vez que as máquinas precisam executar técnicas computacionais complexas e custosas. No projeto PERMM [2], é usado um modelo de textura baseado na conectividade dos *pixels* que têm picos de valor na segunda derivada de uma versão em tons de cinza da imagem (*ridge map*) [49]. Esta abordagem, no entanto, não é invariante à escala, tampouco é capaz de mapear a textura em valores numéricos, de modo que não é indicada para utilização com rede neurais artificiais MLP.

Para que haja invariância de escala e rotação na análise de texturas em imagens, devem ser utilizadas técnicas que suportem estas propriedades, dentre as quais se destacam os *Wavelets* [32] - em especial Daubechies e Haar [59] -, a técnica LBP (*Local Binary Patterns*) [41].

A técnica dos Filtros de Gabor, (apresentada no Apêndice A, página 103), pode ser adaptada para ser invariante à escala e à rotação, conforme sugerido por Jiang *et al.* [25]. Para isto, é necessário utilizar-se dos resultados da aplicação de diversos filtros, variando a sua escala e a sua orientação, em seguida os resultados são somados para compor uma única imagem, na qual a luminância de cada *pixel* corresponde ao valor da característica de textura do *pixel* de mesmas coordenadas na imagem original.

Para exemplificar esta adaptação, a Figura 2.5 apresenta duas versões da aplicação de filtros de Gabor na imagem de exemplo. A Figura 2.5(b) é o resultado da aplicação de um Filtro de Gabor com núcleo medindo 35 *pixels* de lado e com orientação de 0 radiano, enquanto a 2.5(c) é a composição dos resultados de quatro Filtros de Gabor, variando as combinações entre as escalas 25 e 35 *pixels* e as orientações de 0 e $\frac{\pi}{2}$ radianos.

A resposta de cada região ao filtro de Gabor é percebida pela intensidade da região corres-

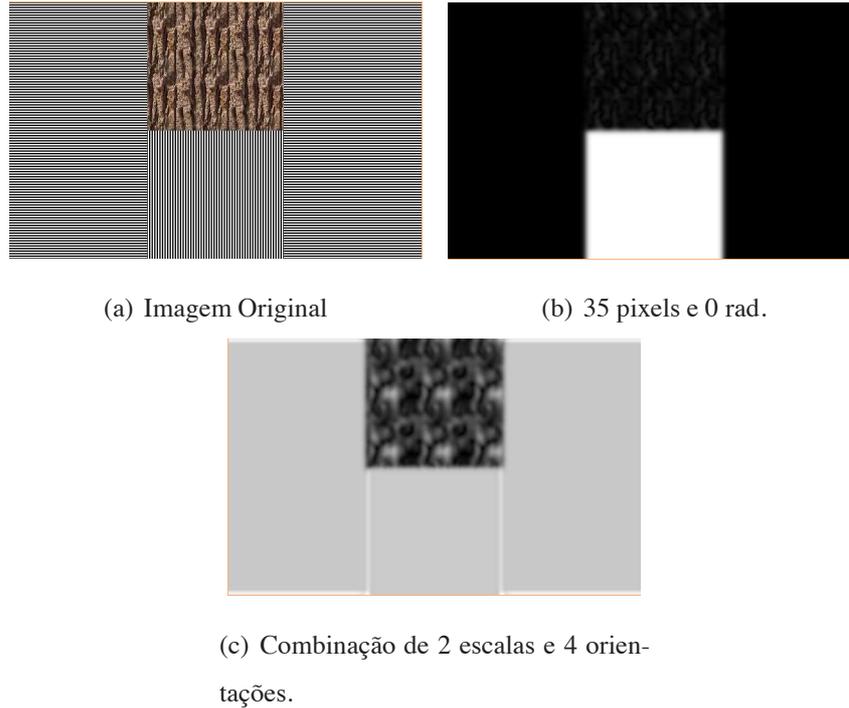


Figura 2.5: Comparativo entre aplicações isoladas e combinadas de Filtros de Gabor

pondente na imagem de saída: quanto mais forte a resposta, mais clara a região. Percebe-se que, na Figura 2.5(b), as regiões de listras na horizontal não produzem resposta e aparecem apagadas, uma vez que apenas as texturas verticais são detectadas pelo único Filtro de Gabor aplicado à imagem. Já na Figura 2.5(c), todas as regiões produzem respostas, mais fortes ou mais fracas conforme forem a textura da região.

Quando utilizados dessa maneira, os filtros de Gabor podem ser considerados uma técnica de extração de informação de textura invariante a escala e a rotação, além de prover um único valor real para cada *pixel* da imagem, o que a torna a melhor candidata à utilização para compor o método proposto nesta dissertação. A descrição deste método – Capítulo 3 – traz mais detalhes acerca da utilização dos filtros de Gabor no contexto deste estudo.

2.3 Considerações Finais

Apesar de ser um tema pouco explorado na literatura acadêmica, a revisão bibliográfica realizada pôde fornecer informações muito importantes acerca de tópicos relevantes para o ajuste e obtenção de segmentadores de detecção de materiais.

Destacam-se as constatações de que há diversas características das imagens e dos mate-

riais que já foram estudados e comprovadamente ajudam a promover a separabilidade entre as regiões do material que se deseja detectar e o restante da imagem. Dentre essas características estão o espaço de cor adotado, a profundidade de cor, o posicionamento dos *pixels* na imagem e a textura apresentada pelo material.

Além disso, muitas referências consultadas serviram como fontes de técnicas cujos resultados foram comparados aos alcançados pelos segmentadores obtidos usando o método proposto no próximo capítulo.

Capítulo 3

Método Proposto

Neste capítulo é apresentado o método proposto para a automatização do ajuste de segmentadores de imagens baseados em informação de textura e cor, principal contribuição desta dissertação. Inicialmente, é fornecida uma visão geral das principais fases do método proposto, seguida do detalhamento de cada uma delas e da implementação de protótipo de *software* que o implementa.

3.1 Visão Geral do Método Proposto

O diagrama apresentado na Figura 3.1 contém uma visão geral do método proposto.

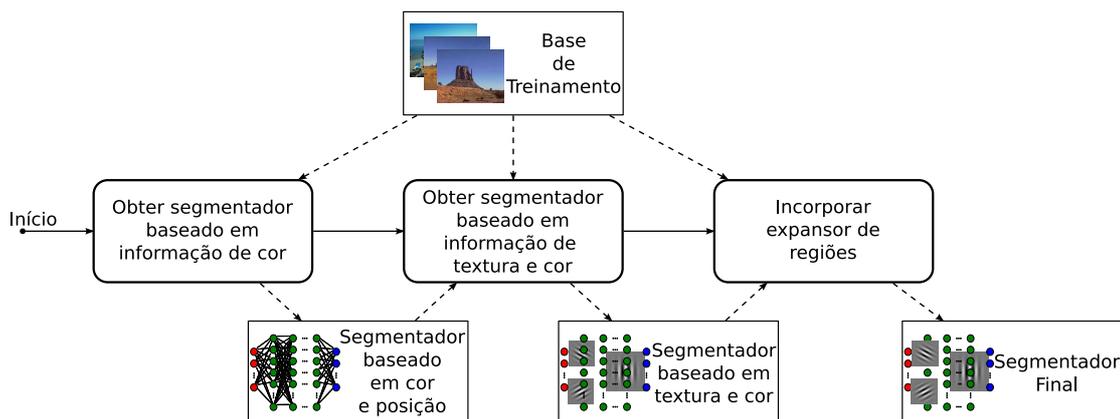


Figura 3.1: Visão geral do método proposto

Pode-se observar que o método é dividido em, basicamente, três fases principais:

1. Obtenção do segmentador baseado em informação de cor: nesta fase apenas são utilizadas características de cor e do posicionamento dos *pixels* nas imagens, como será descrito a seguir;
2. Obtenção do segmentador baseado em informação de cor e textura: utilizando como insumo o segmentador gerado na fase anterior, incorporam-se as características de textura para obter um segundo segmentador, em geral de menor qualidade que o primeiro;
3. Obtenção do segmentador final: utilizando como insumo o segundo segmentador, é obtido um terceiro, que consiste em um expansor das regiões fornecidas pelo segmentador baseado em cor e textura, de modo a recuperar a perda na medida-F (Apêndice A, página 87) provocada na segunda fase do algoritmo.

A abordagem de gerar um segmentador de pior qualidade para depois recuperar a sua qualidade no passo seguinte foi sugerida por Jiang *et al.* [25] para o âmbito da detecção de pele humana em fotografias e generalizada para satisfazer o requisito de detectar outros tipos de materiais. Vale ressaltar que cada fase utiliza como insumo o resultado da fase anterior, ou seja, quanto melhor for o resultado de uma fase, melhor também será o da fase adiante.

3.2 Segmentador Neural Baseado em Informação de Cor

O objetivo da primeira fase do método proposto nesta dissertação é obter e ajustar um segmentador de imagens baseado em informação de cor, levando também em consideração o posicionamento dos *pixels* nas imagens.

Para promover a classificação dos *pixels* como pertencentes às regiões *P* e *N* (Apêndice A, página 83) das imagens, foram utilizadas Redes Neurais Artificiais do tipo MLP, uma vez que o problema abordado é uma instância de problemas de classificação de padrões não-linearmente separáveis. Isto pode ser observado na Figura 3.2, em que são exibidas as distribuições dos *pixels* pertencentes às regiões de grama (em verde) e de não-grama (em vermelho) da base de treinamento utilizada no estudo de caso da Seção 4.1, considerando os componentes de cor Verde \times Vermelho e Verde \times Azul.

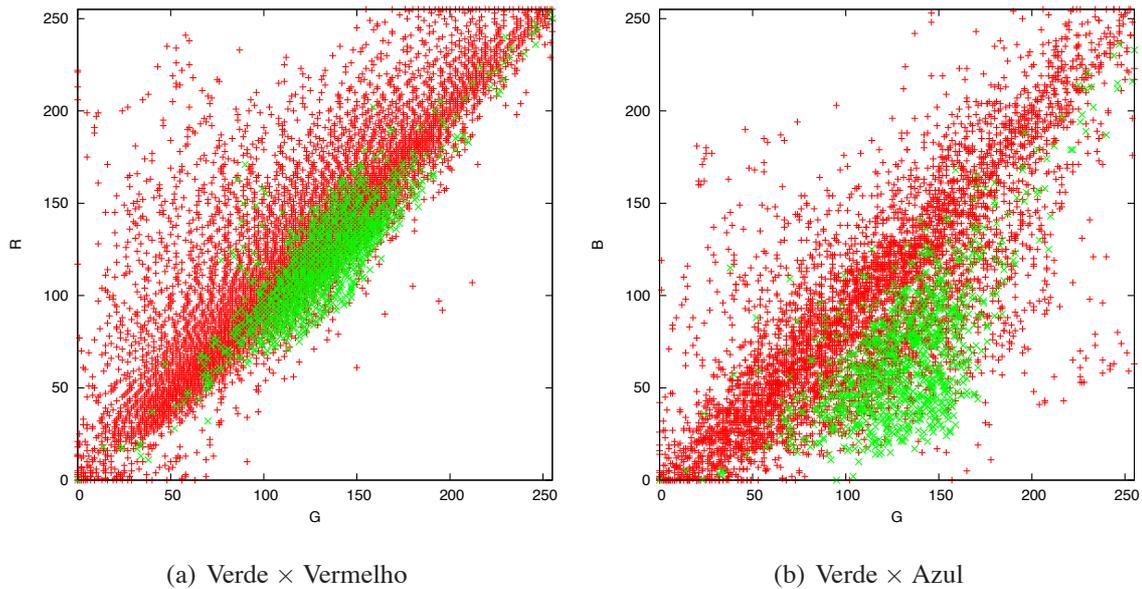


Figura 3.2: Distribuições dos pixels de grama (verdes) e não-grama (vermelhos), considerando possíveis separações a partir do uso de canais de cores verde, azul e vermelho.

Uma vez que existem materiais cuja separação é não linear, se faz necessária a utilização de uma técnica que seja capaz de lidar com este tipo de massa de dados. Além disto, como é possível observar nas figuras, há um pouco de sobreposição entre as duas regiões. Isto faz com que as redes MLPs sejam uma técnica adequada, pois elas são capazes de moldar uma superfície de separação não-linear que maximiza a separabilidade e, por consequência, a medida-F do segmentador final.

Dentre as técnicas capazes de separar padrões não-linearmente separáveis, como classificadores Bayesianos e Máquinas de Vetores de Suporte (SVM), as redes MLP foram utilizadas por que, apesar de apresentarem um treinamento bastante lento [16], possuem as vantagens de necessitarem de pouca memória para execução [42] e de não exigirem que nenhuma premissa seja satisfeita com relação às distribuições de probabilidades dos padrões a serem classificados.

3.2.1 Treinamento de um Segmentador Neural

No âmbito do trabalho apresentado nesta dissertação, o treinamento de uma rede neural MLP requer que uma série de tarefas sejam executadas antes que o algoritmo de aprendizagem seja efetivamente executado, a fim de se compor o conjunto de padrões de treinamento com base em que a rede neural MLP será treinada.

A arquitetura das redes neurais treinadas neste estudo está apresentada na Figura 3.3: uma camada de entrada com 3 a 5 neurônios, a depender dos eixos de posicionamento utilizados no treinamento, uma camada oculta com número de neurônios diferente a cada treinamento e dois neurônios de saída, um referente à região P (resultados positivos) e outro à região N (resultados negativos). Nesta figura, $ch_i, i \in \{1,2,3\}$ são os valores dos canais de cor (Equação 3.3, página 24), v é a coordenada vertical do *pixel* relativa à margem inferior da imagem (Equação 3.5, página 24) e h é a coordenada horizontal do *pixel* relativa à margem direita da imagem (Equação 3.4, página 24).

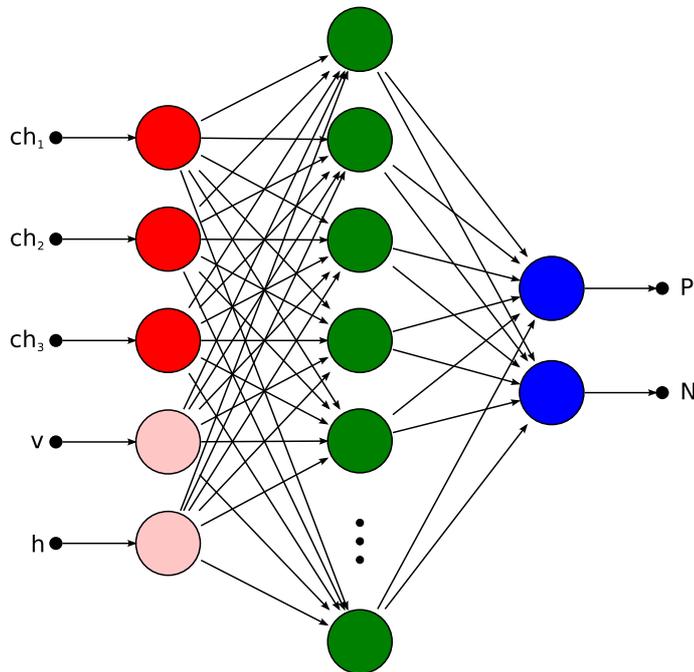


Figura 3.3: Representação da arquitetura das redes MLP no âmbito desta dissertação.

O único insumo exigido para utilizar o método proposto é uma base de imagens que contenha o material que se deseja detectar. Cada imagem é acompanhada de uma máscara que representa sua segmentação manual. Por exemplo, caso se queira detectar pele humana, o conjunto de imagens de treinamento deve ser segmentado como na Figura 3.4

Notar que, para efeito de treinamento, as regiões correspondentes ao material que se quer segmentar são marcadas na cor branca e as demais em cor negra. Além disto, as regiões de interface entre superfícies do material em questão e demais regiões das imagens devem ser marcadas em cor cinza, para serem desconsideradas durante a fase de treinamento. Esta

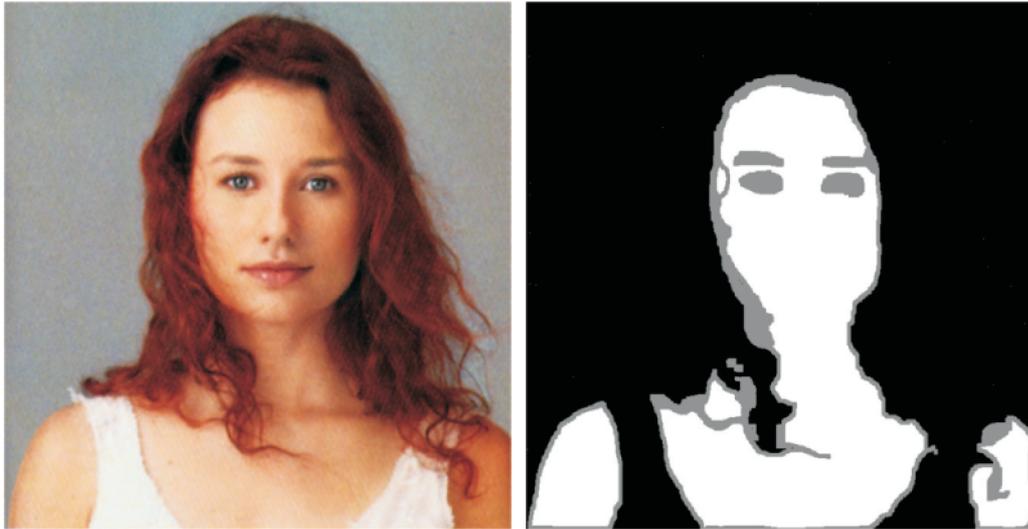


Figura 3.4: Exemplo de imagem no conjunto de treinamento.

estratégia, já utilizada em trabalhos revisados [48], é importante, pois o potencial de erro na marcação destas regiões é muito alto, uma vez que é difícil de se determinar se os *pixels* correspondem à superfície do material que se deseja segmentar.

No início da estratégia de amostragem para composição do conjunto de treinamento, as imagens sofrem uma filtragem de borragem (*blur*) gaussiana. O filtro aplicado é configurado de modo a ser quadrado e ter soma de seus pesos iguais a um.

Isto serve para diminuir a característica de localidade da operação de detecção do material: caso seja considerado apenas o valor de um *pixel*, todos os outros de mesma cor terão sempre os mesmos resultados, independente do “contexto” em que se encontram na imagem. O efeito da aplicação deste filtro pode ser observado na Figura 3.5.

No início da preparação do conjunto de treinamento, todos os *pixels* das imagens são amostrados, à exceção daqueles que estiverem marcados em cinza, o que indica que devem ser desconsiderados. Cada *pixel* amostrado tem o valor de seus canais convertido para o espaço de cor designado para o treinamento da rede neural. As funções de transformação do espaço de cores RGB para os diferentes espaços de cores analisados nesta dissertação encontram-se no Apêndice B.

Em seguida, é efetuada a normalização parcial dos valores extraídos. Esta normalização tem por objetivo mapear para o intervalo $[0..1]$ os valores dos canais do espaço de cores designado. Isto se consegue aplicando as transformações apresentadas na Equação 3.1, em



(a) Imagem Original

(b) Imagem Borrada

Figura 3.5: Efeito do filtro de borragem gaussiana (janela de 9x9 pixels)

que p_i é o valor do *pixel* no i -ésimo canal e c_i^{min}, c_i^{max} são os valores mínimo e máximo do domínio do canal, respectivamente.

$$p_i^{norm} = \frac{p_i - c_i^{min}}{c_i^{max} - c_i^{min}} \quad (3.1)$$

A seguir é efetuada a redução da profundidade de cor. Apesar de os canais dos espaços de cores possuírem domínios em intervalos completamente diferentes, o que determina a profundidade de cor é a quantidade *bits* utilizada para codificar os canais. Ou seja, independente do espaço de cores utilizado, inicialmente todos os canais têm profundidade de 8 *bits* por canal.

Este fato permite que a redução da profundidade de cor seja feita após a normalização dos canais: considera-se o valor da normalização parcial como originária de um domínio no intervalo $[0..255]$ usando 256 níveis de quantização, transforma-se para este intervalo, executa-se a redução de níveis e retorna-se ao intervalo da normalização parcial ($[0..1]$). A Equação 3.2 é empregada para esta finalidade.

$$p_i^{reduced} = \frac{\lfloor \frac{p_i^{norm} \cdot 255 \cdot 2^l}{256} \rfloor}{2^l - 1} \quad (3.2)$$

em que l é a profundidade de cores após a redução.

Por fim é realizada uma segunda normalização, desta vez para mapear os valores encontrados após a redução da profundidade de cor para o intervalo de valores de entrada da rede neural MLP, $[-1..1]$. Os limites deste intervalo foram adotados para serem consistentes com os valores de saída da função de ativação tangente hiperbólica (Figura A.6, na página 92). Para este fim utiliza-se a Equação 3.3.

$$ch_i = 2p_i^{reduced} - 1 \quad (3.3)$$

em que ch_i é o valor de entrada da rede neural para o i -ésimo canal de cor, no espaço de cores considerado.

Além da informação de cor, o posicionamento dos *pixels* durante o treinamento das redes MLPs também pode ser considerado. Neste caso, pode-se considerar o posicionamento no eixos horizontal (distância relativa à margem esquerda), vertical (distância relativa ao topo da imagem), ambos, ou nenhum. Isto significa que as redes neurais terão de 3 a 5 elementos de entrada.

Quando se define que a rede vai levar em consideração um eixo de posicionamento, o padrão de entrada incorpora mais uma dimensão, cujo valor é calculado utilizando as Equações 3.4 ou 3.5 (para as eixos horizontal e vertical, respectivamente), em que x é a coordenada do *pixel* relativa à margem esquerda, y é a coordenada do *pixel* relativa ao topo da imagem, w é a largura da imagem e h é a sua altura (Figura 3.6).

$$h = \frac{x}{w} \quad (3.4)$$

$$v = \frac{y}{h} \quad (3.5)$$

As redes neurais treinadas no âmbito desta dissertação possuem dois neurônios de saída: o primeiro deles o_1 representa a saída que indica a pertinência do *pixel* à região P e o seguinte, o_2 , indica a pertinência à região N . Desta maneira, a saída esperada para os padrões

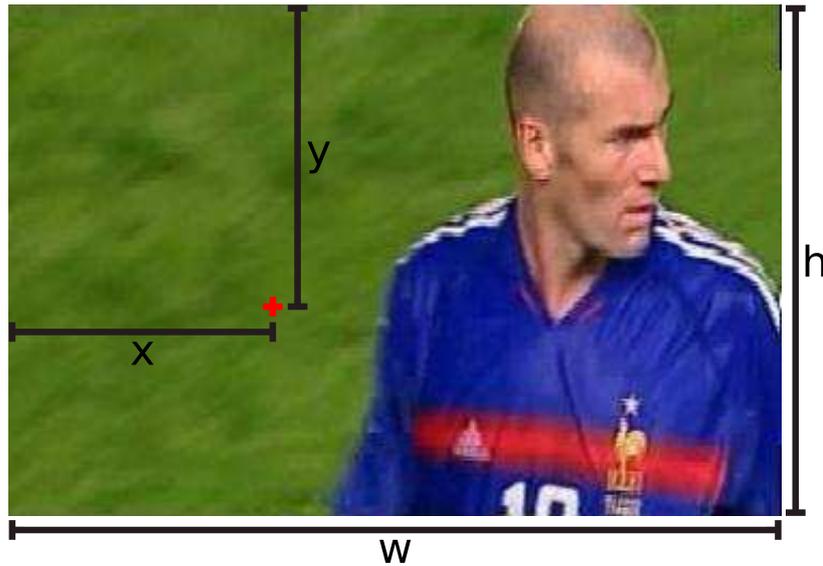


Figura 3.6: Distâncias consideradas quando se utiliza o posicionamento dos pixels

que representam *pixels* pertencentes ao material que se quer detectar é $o_1 = +1, o_2 = -1$ e para os demais $o_1 = -1, o_2 = +1$.

As amostras são ordenadas de acordo com os valores dos canais de cor, do posicionamento dos *pixels* – quando forem utilizados – e da região em que se encontram (P ou N), nesta ordem de critérios.

Desta lista, são selecionadas todas as k -ésimas amostras tais que $k \equiv \eta \pmod{[\eta + 1]}$ para compor o conjunto de validação do treinamento, com base em que serão calculadas as métricas de avaliação das redes em tempo de treinamento. Nesta equação, o parâmetro η é a *relação de holdout*, ou seja, a relação aproximada entre as quantidades de *pixels* para validação e para treinamento.

No entanto, é sabido que a ocorrência de ambiguidades – isto é, de padrões iguais de entrada com resultados esperados diferentes – no conjunto de treinamento de uma rede neural degradam a sua capacidade de aprendizado, uma vez que o mapeamento entrada-saída deixa de ser uma função matemática [47]. Portanto, torna-se necessário filtrar as amostras da seguinte maneira:

1. são removidas todas as ocorrências da classe minoritária e uma quantidade igual de ocorrências da classe majoritária de todos os padrões que apresentam sobreposição com números diferentes de ocorrências em cada classe;

2. são removidas todas as ocorrências dos padrões que apresentam sobreposição com número igual de ocorrências em cada classe;
3. permanecem as amostras dos padrões que não apresentam repetição.

Por exemplo, se o padrão $p_1 = (ch_{11}, ch_{21}, ch_{31})$ ocorrer 120 vezes na classe P e apenas 32 vezes na classe N , ao final da filtragem este padrão apresentará $120 - 32 = 88$ ocorrências na classe P , de acordo com a regra de filtragem 1. Já um padrão $p_2 = (ch_{12}, ch_{22}, ch_{32})$, se ocorrer 120 vezes na classe P e apresentar o mesmo número de ocorrências na classe N , será completamente extinto do conjunto de treinamento, de acordo com a regra de filtragem 2.

Após a filtragem, o conjunto de treinamento das redes composto a partir da seleção de uma quantidade $2s$ de amostras, de tal forma que s é o maior múltiplo de h (relação de *holdout*) que seja menor que o número total de amostras da classe (P ou N) minoritária na lista filtrada, conforme Equação 3.6.

$$s = S_{min} - S_{min} \bmod k \quad (3.6)$$

em que S_{min} é a quantidade de amostras da classe (P ou N) que apresentar menor quantidade de ocorrências na lista filtrada.

Esta quantidade é dividida igualmente entre amostras equiespaçadas das duas classes (P e N), de modo a fazer com que o conjunto de padrões para treinamento seja equilibrado e contenha exatamente a mesma quantidade de amostras de cada uma das classes.

A técnica de validação adotada neste trabalho foi o *holdout validation* [60]. Apesar de existirem técnicas mais sofisticadas de validação – a exemplo das diversas variantes do *crossvalidation* –, o *holdout* foi adotado devido às diferentes naturezas dos conjuntos de treinamento (filtrado, sem ambiguidades) e de validação (sem filtragem).

Esta técnica define que o treinamento da rede neural deve ser feito com um subconjunto da massa de dados original de treinamento, que não deve ser usado para treinamento do modelo preditivo. Isto motivou a separação das amostras da base de imagens em dois grupos diferentes de amostras. Para que sejam mitigados os efeitos da aleatoriedade da inicialização dos pesos da rede neural, é definida uma quantidade i de iterações para cada um dos treinamentos executados.

Cada uma das iterações dos treinamentos tem como critério de parada uma quantidade

máxima de épocas de treinamento. No entanto, uma vez que a validação da rede neural é feita com um conjunto separado, a rede final não necessariamente tem os pesos sinápticos apresentados na última época, mas os pesos apresentados na época em que teve menor erro de validação. Isto é importante para evitar o *overfitting* da rede e aumentar seu poder de generalização.

Depois da aplicação do algoritmo de aprendizado da rede neural é feito o cálculo da medida-F, obtendo a classe de cada padrão de acordo com a seguinte regra: para um padrão ser considerado da classe P , a condição $o_1 \geq o_2 + t$ deve ser satisfeita.

O parâmetro t é denominado “limiar de aceitação” e tem seu valor descoberto a partir de uma varredura no intervalo contínuo $[0..2]$ (o valor máximo ocorre quando $o_1 = 1, o_2 = -1$), em busca do valor que ajusta os resultados para a maior medida-F.

3.2.2 Obtenção do Segmentador Neural Baseado em Informação de Cor

O ajuste do segmentador neural baseado em informação de cor consiste em encontrar o valor mais adequado a cinco parâmetros: espaço de cores, profundidade de cor, tamanho da janela de *pixels*, número de neurônios na camada oculta e a configuração dos eixos de posicionamento.

Os passos seguidos na fase de obtenção do segmentador neural baseado em informação de cor são apresentados na Figura 3.7. Observa-se que os parâmetros são ajustados em série, um após o outro. A partir do momento em que um parâmetro já for ajustado, ele já será utilizado para decidir o valor do próximo parâmetro, e assim por diante.

Os valores padrão (iniciais) dos parâmetros são:

- Espaço de cores: RGB, por ser o espaço de cores mais comum para codificações digitais [15];
- Profundidade de cor: 8 *bits* por canal de cor;
- Tamanho da janela: 1, obtendo o mesmo efeito de analisar somente o *pixel* central, sem levar em consideração a influência dos *pixels* vizinhos;

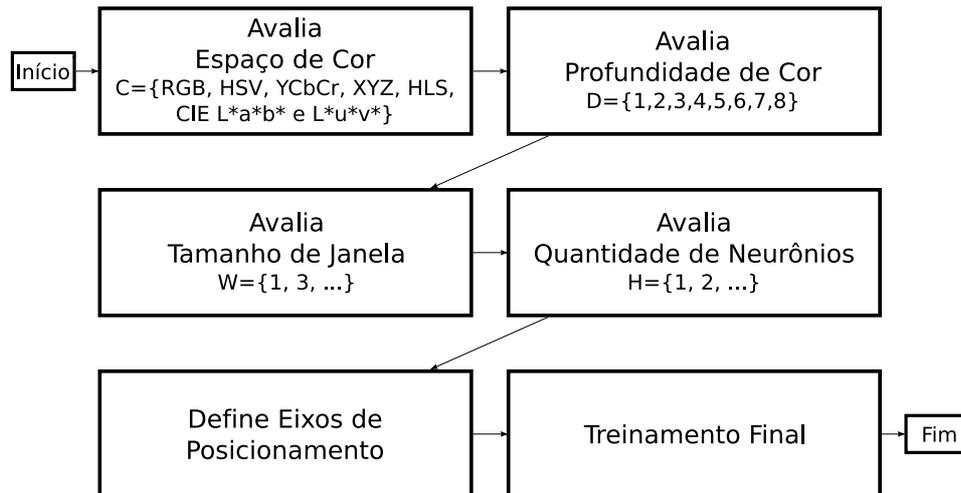


Figura 3.7: Principais passos da obtenção do segmentador neural baseado em informação de cor

- Neurônios na camada oculta: 7, a partir da aplicação da heurística de Hecht-Nielsen [17], considerando 3 entradas para a rede (sem utilizar os eixos de posicionamento); e
- Eixos de posicionamento: por padrão, não é levado em consideração nenhum dos eixos de posicionamento.

O primeiro passo é decidir o espaço de cores que possibilita a maior separabilidade entre as regiões P e N das imagens pertencentes ao conjunto de treinamento para o material que se deseja detectar. Para tanto, é treinada uma rede neural MLP para cada um dos espaços de cores a serem explorados (RGB, HSV, YCbCr, XYZ, HLS, $L^*a^*b^*$ e $L^*u^*v^*$), conforme o Algoritmo 3.1. É adotado o espaço de cores que apresentar maior medida-F entre seus concorrentes.

Algoritmo 3.1: Exploração do melhor espaço de cores

```

1 exploraEspacoDeCores( baseDeTreinamento , baseDeTeste ) {
2   melhorMedidaF ← 0;
3   melhorEspacoDeCores ← ∅;
4   para cada espacoDeCores em (RGB, HSV, YCbCr, XYZ, HLS, L*a*b* e L*u*v
   *) {
5     redeMLP ← treinaRede( baseDeTreinamento , espacoDeCores , niveis=256,
       janela=1, neuronios=6, ∅);
6     medidaF = testaRede( redeMLP, baseDeTeste );
7     se (medidaF > melhorMedidaF ) {

```

```

8     melhorMedidaF ← medidaF;
9     melhorEspacoDeCores ← espacoDeCores;
10    }
11   }
12   retorna melhorEspacoDeCores;
13  }
```

Em seguida decide-se a profundidade de cores a ser considerada durante a detecção do material. Para tanto é treinada uma rede neural MLP para cada uma das profundidades (1 a 8 *bits*, em ordem decrescente), com o espaço de cores encontrado no passo anterior, conforme o Algoritmo 3.2. É adotada a profundidade de cores que apresentar maior medida-F entre suas concorrentes.

Algoritmo 3.2: Exploração da melhor profundidade de cores

```

1  exploraProfundidadeDeCores ( baseDeTreinamento , baseDeTeste ) {
2    melhorMedidaF ← 0;
3    melhorProfundidadeDeCores ← ∅;
4    para cada profundidadeDeCor em ( 8 , 7 , ... , 1 ) {
5      redeMLP ← treinaRede ( baseDeTreinamento , melhorEspacoDeCores ,
6        profundidadeDeCor , janela=1 , neuronios=6 , ∅ );
7      medidaF = testaRede ( redeMLP , baseDeTeste );
8      se ( medidaF > melhorMedidaF ) {
9        melhorMedidaF ← medidaF;
10       melhorProfundidadeDeCores ← profundidadeDeCor;
11     }
12   }
13   retorna melhorProfundidadeDeCores;
14 }
```

É necessário decidir também sobre os melhores valores para o tamanho da janela de *pixels* e o número de neurônios na camada oculta. Isto é realizado de forma idêntica ao processo de descoberta dos parâmetros anteriores: inicialmente decide-se sobre o tamanho da janela, treinando uma rede neural MLP para cada um dos tamanhos de janela (1,3,5,7,9) e verificando o maior valor de medida-F, com o espaço e a profundidade de cores já encontrados. Em seguida, com os parâmetros já ajustados, são treinadas redes neurais com diferentes números de neurônios na camada oculta (1,2,...,14) e verificado o maior valor de medida-F.

Os eixos de posicionamento a serem utilizados são determinados logo em seguida. Para cada um dos eixos é computado um histograma a partir da base de treinamento que conta a frequência das ocorrências do material que se deseja detectar com respeito à distância relativa à margem esquerda ou superior, a depender do eixo analisado. Deste histograma é extraído o coeficiente de variação (razão entre o desvio padrão e a média), e caso esse valor seja igual ou superior a 0,5, o eixo passa a ser utilizado como entrada da rede neural. Este valor foi determinado empiricamente a partir de experimentos realizados com versões preliminares do método proposto.

Por fim é realizado um treinamento final, levando em consideração todos os parâmetros já ajustados, para obter o segmentador neural baseado em informação de cor que será utilizado como base para obtenção dos próximos segmentadores. A partir do treinamento final, a medida-F considerada para comparação é calculada com respeito a todos os *pixels* da base de treinamento e não com respeito apenas ao conjunto de validação, como feito até então. Isto deve-se ao fato de que nas fases seguintes da obtenção do segmentador a aplicação dos classificadores não pode ser efetuada em pontos isolados, devendo ser aplicados a imagens como um todo.

3.3 Segmentador Baseado em Informação de Textura e Cor

Até então o segmentador obtido só leva em consideração informação de cor. Entretanto, conforme mencionado na Seção 2.2 (página 14), é necessário levar em consideração informação de textura para que sejam obtidas melhores segmentações de detecção de material.

Este é o objetivo da segunda fase do método proposto nesta dissertação: incorporar informação de textura ao segmentador fornecido como resultado da primeira fase. Isto é feito a fim de se minimizar a taxa de falsas aceitações do segmentador ζ , medida com respeito a uma base de imagens B através da Equação 3.7.

$$F_{Pr}^B(\zeta) = \frac{\sum_{I \in B} S(F_P(I))}{\sum_{I \in B} S(N(I))} \quad (3.7)$$

em que I é uma imagem pertencente à base B , $S(x)$ é a área da região x , $N(I)$ é a região da imagem que não corresponde ao material e a região $F_P(I)$ segue a definição do Apêndice A

(página 84).

Para tanto é utilizada uma técnica semelhante à proposta por Jiang *et al.* [25] para detecção de pele, que consiste em utilizar uma combinação de diversas escalas e orientações de filtros de Gabor a fim de obter um descritor de textura centrado em cada *pixel* da imagem.

Inicialmente define-se $\dot{W}_{m\theta}(I)$ como sendo a magnitude do resultado da convolução de um filtro de Gabor com escala $e = \left[24(\sqrt{2})^m + 1\right]$ e orientação θ com a imagem I . Ressalta-se que esta operação só faz sentido se for considerada apenas a intensidade de cada *pixel*, ou seja, deve-se primeiramente converter a imagem para tons de cinza.

Utilizando esta definição é calculado o descritor de textura da imagem, que tem seu valor obtido da utilização da Equação 3.8,

$$T(x,y) = \frac{\sqrt{\sum_{m=0}^M \sum_{n=0}^N \dot{W}_{m\theta_n}^2(x,y)}}{MN} \quad (3.8)$$

em que T é a imagem resultante contendo os descritores de textura, x e y são as coordenadas dos *pixels*, $\theta_n = \frac{n\pi}{N}$ e M e N são o número de escalas e orientações de filtros de Gabor utilizados na composição.

Em seguida é realizada uma binarização de T , utilizando limiares inferior e superior para definir quais valores de intensidade são considerados como pertencentes ao material que se deseja detectar. A Equação 3.9, que formaliza o processo de binarização.

$$B(x,y) = \begin{cases} P, & \text{se } l \leq T(x,y) \leq u \\ N, & \text{caso contrário} \end{cases} \quad (3.9)$$

em que $B(x,y)$ é o resultado da binarização do *pixel* na coordenada horizontal x e na coordenada vertical y .

Por fim é efetuada uma operação de conjunção entre os resultados do segmentador neural baseado em informação de cor $R_c = \sigma_c(I)$ com o resultado da binarização B , a fim de se obter o resultado R_{tc} da segmentação baseada em informação de textura e cor (Equação 3.10).

$$R_{tc}(x,y) = \begin{cases} P, & \text{se } R_c(x,y) \in P(R_c) \text{ e } B(x,y) \in P(B) \\ N, & \text{caso contrário} \end{cases} \quad (3.10)$$

Para visualizar o efeito de cada um dos passos da segmentação baseada em textura e cor, a Figura 3.8 exibe os resultados parciais das operações ($M = 4$, $N = 1$, $l = 0$, $u = 7$). Os respectivos resultados parciais da segmentação são apresentados na Tabela 3.1. Notar a redução no valor da taxa de falsas aceitações apresentada a cada resultado parcial.

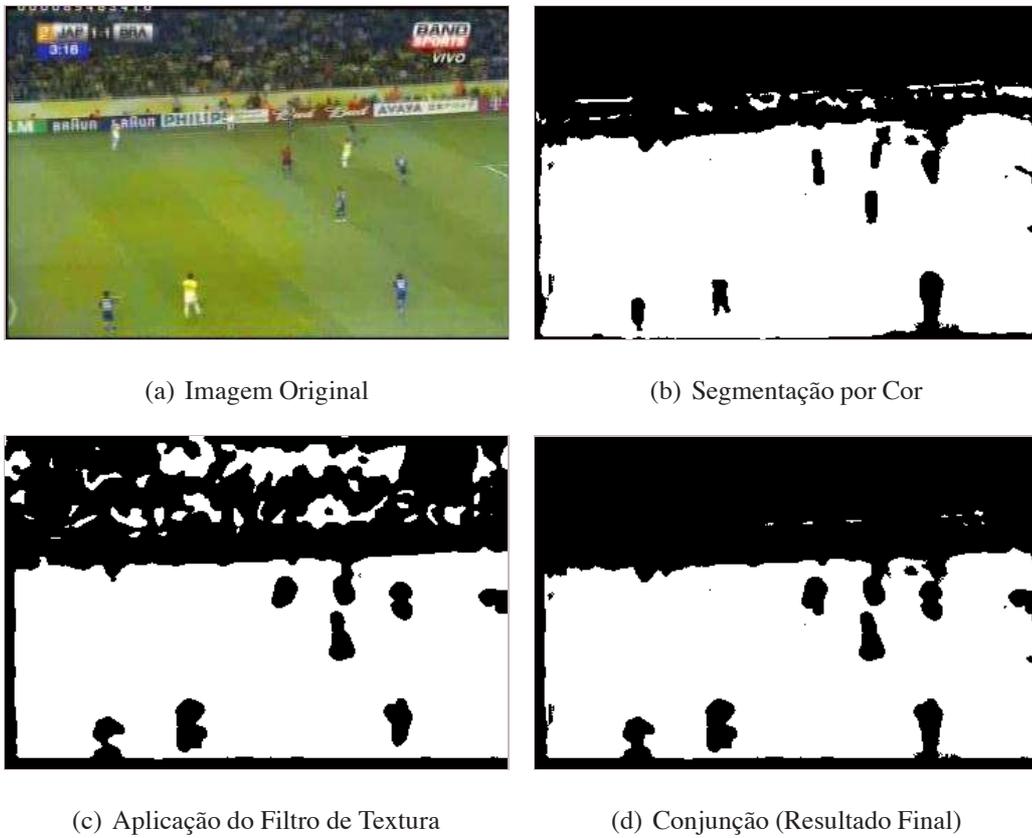


Figura 3.8: Resultados parciais da segmentação baseada em informação de textura e cor.

Figura - Etapa	Medida-F	F_{Pr}
3.8(b) - Segmentação por Cor	0,9455	0,0363
3.8(c) - Aplicação do Filtro de Textura	0,8769	0,1800
3.8(d) - Conjunção (Resultado Final)	0,9138	0,0028

Tabela 3.1: Métricas extraídas dos resultados parciais da segmentação baseada em informação de textura e cor.

Nota-se que a obtenção do segmentador baseado em informação de textura e cor envolve determinar o valor de quatro parâmetros: M , N , l e u . De maneira diferente da proposta por Jiang *et al.* [25], que simplesmente estimou empiricamente um valor para tais parâmetros, isto é realizado no âmbito desta dissertação através da busca da combinação de valores que resulta no segmentador capaz de obter a maior medida-F dentre os demais que também consideram o mesmo segmentador obtido na Fase 1 do método proposto.

Uma vez que o espaço de busca tornaria ineficaz a realização de uma busca exaustiva por todas as combinações possíveis de valores de parâmetros, e aproveitando-se de o problema caracterizar-se como um problema de otimização, é utilizado um algoritmo genético para determinar os valores dos parâmetros.

Neste caso, um cromossomo é uma quádrupla $C = \{M_C, N_C, l_C, u_C\}$, que corresponde aos quatro parâmetros que se deseja otimizar. Este cromossomo é codificado como uma cadeia de *bits*, de modo a tornar o cromossomo mais longo na perspectiva do algoritmo, o que aumenta a quantidade de pontos de corte no momento da operação de *crossover*, permitindo uma melhor exploração do espaço de busca de soluções.

A avaliação de um cromossomo consiste em obter um segmentador utilizando seus parâmetros e determinar a sua medida-F ($\beta = 5$) com respeito à base de imagens de treinamento fornecida como insumo para utilização do método proposto. A medida-F ($\beta = 5$), neste caso, atua como função objetivo do Algoritmo Genético.

O espaço de soluções a ser explorado pelo algoritmo é o produto cartesiano dos seguintes intervalos inteiros:

- [1..10] para o parâmetro M , respeitando a restrição de que os filtros devem caber na menor imagem do conjunto de treinamento apresentado;
- [1..10] para o parâmetro N ;
- [0..64] para o parâmetro l ; e
- [1..64] para o parâmetro u , respeitando a restrição de que $u > l$. Quando a restrição não pode ser satisfeita, diz-se que o cromossomo é inválido.

Os parâmetros l e u foram limitados até 64 para agilizar a execução do algoritmo genético, uma vez que percebeu-se que valores muito altos de descritores de textura não ocorrem

nos resultados dos filtros de Gabor aplicados nas imagens.

Os detalhes de execução dos algoritmos genéticos serão apresentados mais adiante, juntamente com a descrição de toda a configuração dos testes, na Seção 3.5.

3.4 Expansão de Regiões para Recuperação da Medida-F

A última fase do método proposto nesta dissertação consiste em incorporar ao segmentador um expensor de regiões baseado em informação de cor, a fim de recuperar a perda de medida-F provocada pela fase anterior (ver Figura 3.8 e Tabela 3.1 como referência).

Para efetuar a expansão de regiões, uma versão adaptada do algoritmo *floodfill* [15] é executada nos resultados da fase 2 do método proposto. A adaptação consiste em (i) mudar a definição da área conectada e (ii) utilizar mais de uma semente ao executar o algoritmo.

A mudança na definição da conectividade é que, em vez de considerarem conectados apenas *pixels* de cores que respeitem um certo limite de distância da cor da semente, são considerados conectados os *pixels* que respeitem, no espaço de cores determinado no primeiro passo da primeira fase do método proposto:

1. Um limite b de distância Euclidiana de cor com respeito ao *pixel* da borda da região já marcada; e
2. Um limite a de distância Euclidiana de cor com respeito à média de cor da região já marcada.

A Figura 3.9(a) exibe uma imagem (ampliada para facilitar a visualização), sobre a qual será executada o algoritmo *floodfill* adaptado ($a = b = 50$), e a semente marcada com um círculo vermelho. No passo exibido na Figura 3.9(b), a média da área marcada é 172 e o *pixel* candidato à marcação (em verde) possui intensidade 92, o que não respeita o limite a .

Já no passo exibido na Figura 3.9(c), o *pixel* da área marcada conectado candidato à marcação tem intensidade 136, enquanto ele próprio (em verde) possui intensidade 50, o que não respeita o limite b . O resultado final da execução do algoritmo é exibido na Figura 3.9(d).

A utilização de múltiplas sementes para uma mesma execução do algoritmo *floodfill*, por sua vez, é facilmente obtida pela simples iteração por todas as sementes executando o algo-

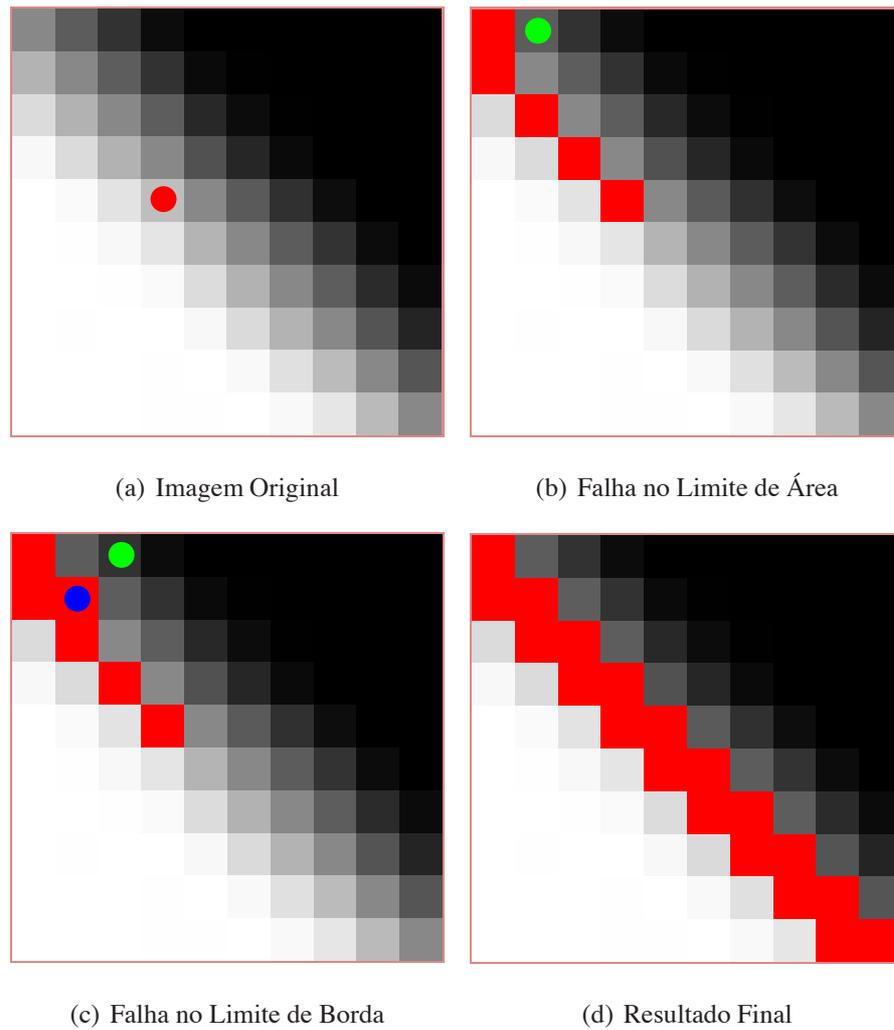


Figura 3.9: Algoritmo *floodfill* adaptado e demonstração dos passos intermediários

Figura	Medida-F
3.10(b) - Resultado Parcial	0,9138
3.10(c) - Resultado Final	0,9644

Tabela 3.2: Resultados parciais da segmentação baseada em informação de textura e cor.

ritmo para cada uma delas. São consideradas sementes para o algoritmo *floodfill* adaptado todos os *pixels* marcados como P no resultado da segunda fase do método proposto.

Para um exemplo real da execução deste algoritmo no âmbito da automatização proposta nesta dissertação, a Figura 3.10 exibe o resultado da aplicação do *floodfill* adaptado ao resultado parcial da Figura 3.8(d). Para esta execução, os parâmetros utilizados foram $a = 18,5753$ e $b = 232,634$, no espaço de cores $YCbCr$.

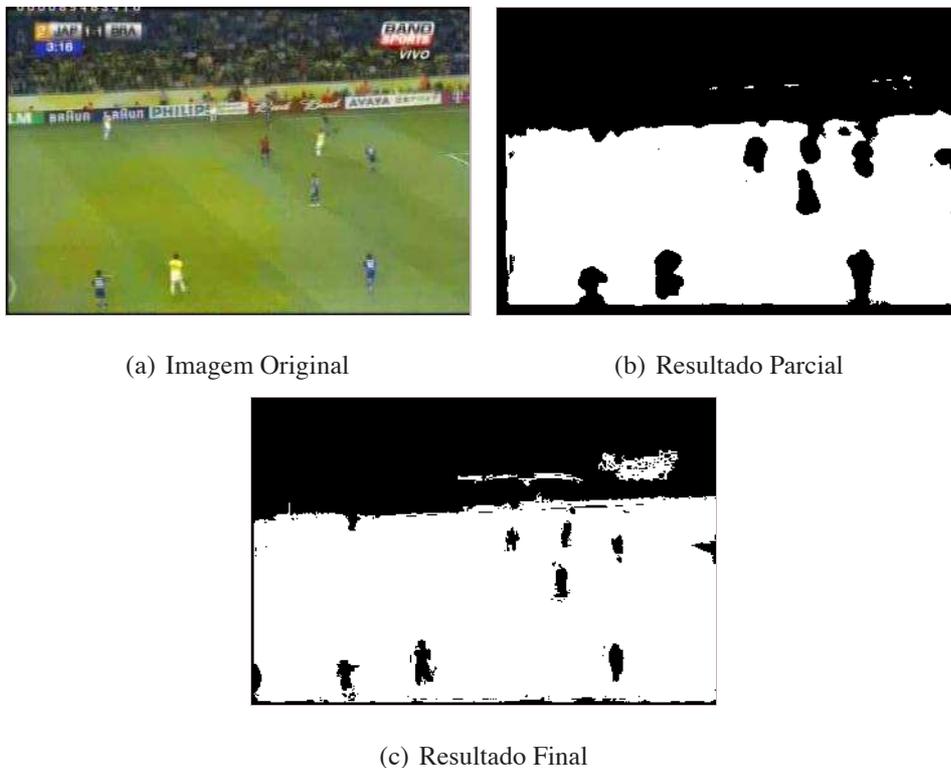


Figura 3.10: Algoritmo *floodfill* executado para detecção de grama.

Os respectivos resultados parciais da segmentação são apresentados na Tabela 3.2. Notar a recuperação da medida-F após a incorporação do expansor de regiões, que aumentou em relação à segmentação apenas baseada em informação de cor (Figura 3.8(b)).

Nota-se que a incorporação do expansor de regiões envolve determinar o valor dos parâmetros a e b . Neste caso, também é utilizado um algoritmo genético para determinar os

valores mais adequados.

Um cromossomo deste algoritmo é um par ordenado $C = \{a_C, b_C\}$, que corresponde aos dois parâmetros que se deseja otimizar. Este cromossomo, a exemplo do utilizado para determinar os parâmetros dos filtros de textura, também é codificado como uma cadeia de *bits* a fim de tornar o cromossomo mais longo na perspectiva do algoritmo.

A avaliação deste cromossomo consiste em obter um segmentador incorporando o expansor de regiões obtido com seus parâmetros e determinar a sua medida-F com respeito à base de imagens de treinamento fornecida como insumo para utilização do método proposto. A medida-F também atua neste caso como função objetivo do Algoritmo Genético.

O espaço de soluções a ser explorado pelo algoritmo é produto cartesiano dos seguintes intervalos inteiros $[0..452]$ para ambos os parâmetros. Este limite superior foi adotado por ser a maior distância euclidiana possível entre dois pontos em um mesmo espaço de cores (CIE $L^*u^*v^*$), dentre os espaços adotados.

Os detalhes de execução dos algoritmos genéticos serão apresentados mais adiante, juntamente com a descrição de toda a configuração dos testes, na Seção 3.5.

Neste ponto é realizada a decisão de que segmentador considerar como sendo o ideal para o material que se deseja detectar: o que considera apenas informação de cor (Fase 1), ou o segmentador final que considera informação de cor e textura com a aplicação do algoritmo *floodfill* adaptado. Considera-se o segmentador que obtiver maior medida-F com respeito a todos os *pixels* da base de treinamento.

3.5 Detalhes de Implementação do Protótipo

O desenvolvimento de um protótipo de *software* implementando o método proposto nesta dissertação também faz parte do escopo do trabalho.

Todos os módulos do protótipo foram implementados na linguagem de programação C++. Esta linguagem foi adotada por ser orientada a objetos, o que o que facilita importantes aspectos no âmbito da engenharia de *software* (reusabilidade, extensibilidade, manutenibilidade, entre outros), e por ser capaz de gerar programas com maior velocidade de execução quando comparados com programas gerados em outras linguagens como Java, Fortran etc. [6].

Além disto, na linguagem C++ estão escritas as três bibliotecas em que se baseiam o protótipo: Intel OpenCV [23], `libFann` [40] e `galib` [57]. A primeira delas trata dos aspectos relacionados ao processamento de imagens digitais, desde as estruturas de dados que representam as imagens até rotinas de manipulação de mais alto nível. A `libFann` fornece a implementação das funcionalidades de treinamento e execução de redes neurais artificiais do tipo MLP com o algoritmo de treinamento por retropropagação. Por fim, a `galib` se trata de uma biblioteca que fornece um *framework* parametrizável para execução de algoritmos genéticos e suas diversas variações. Com a sua utilização, é suficiente fornecer a definição da estrutura do cromossomo, uma vez que a implementação da função objetivo e o *framework* trata de encapsular os detalhes de implementação específicos dos algoritmos genéticos.

Os testes realizados para compor a avaliação experimental foram executados em um computador equipado com processador Intel Pentium IV, 2 GB de memória de acesso aleatório (RAM), 80 GB de capacidade de armazenamento em disco rígido, que executa o sistema operacional Linux – distribuição Slackware – com núcleo na sua versão 2.6.17.

Conforme descrito neste capítulo, o método proposto envolve o treinamento de Redes Neurais Artificiais do tipo MLP. No âmbito deste método, todos os treinamentos foram executados durante 100 épocas de aprendizado, no regime de *holdout validation*, com parâmetros $h = 10$ (relação de *holdout*) e $i = 10$ (quantidade de iterações para um mesmo treinamento). As redes MLP treinadas eram completamente conectadas e nenhuma variação foi aplicada no algoritmo padrão de retropropagação. Todos os neurônios, exceto os da camada de saída, apresentam uma entrada de *bias*, conforme explicado no Apêndice A, Seção A.3.1.

O desvio padrão σ dos filtros de borragem gaussiana é calculado de acordo com o tamanho l do lado do quadrado usando a implementação padrão do OpenCV, de acordo com a Equação 3.11.

$$\sigma = 0,5 + \frac{0,6}{l} \quad (3.11)$$

Quanto à parametrização dos algoritmos genéticos executados no âmbito desta dissertação, foram utilizadas as recomendações de Deb e Agrawal [9]: os algoritmos foram baseados primordialmente em operadores de *crossover* por terem função objetivo de comportamento

desconhecido. Desta forma, a probabilidade de *crossover* foi sempre alta (0,9) e a probabilidade de mutação bastante baixa (0,01). O tamanho da população foi de 100 indivíduos para todas as execuções e metade da população sempre era substituída em cada uma das 500 gerações permitidas ao algoritmo.

3.6 Considerações Finais

Este capítulo descreveu em detalhes o método proposto nesta dissertação para automatizar o ajuste de segmentadores de detecção de material baseados em informação de textura e cor.

A principal característica do método é levar em consideração vários aspectos que, segundo trabalhos contidos na revisão bibliográfica, são úteis para aumentar a separabilidade das regiões P e N das imagens de um determinado material.

Além disto, há uma sistematização para o ajuste de todos os parâmetros, ou seja, nada fica definido de maneira *ad hoc* pelo usuário do método, o que dá maior confiabilidade aos resultados fornecidos pelos segmentadores.

No próximo capítulo, são descritos os resultados experimentais da utilização do método proposto em problemas reais de detecção de materiais.

Capítulo 4

Avaliação Experimental

Com o intuito de avaliar experimentalmente o método de automatização do ajuste de segmentadores proposto nesta dissertação, foram desenvolvidos estudos de caso envolvendo a detecção de três materiais diferentes: grama a partir de quadros de vídeos desportivos, pele humana em fotografias, e céu azul em fotografias.

Este capítulo é dedicado a descrever os estudos de caso, apresentando e discutindo os resultados obtidos.

4.1 Detecção de Grama em Quadros de Vídeos Desportivos

O problema da detecção de grama em quadros de vídeos desportivos foi o primeiro dos estudos de caso escolhidos para fazer parte desta avaliação experimental. Este problema consiste em utilizar como entrada os quadros de vídeos de partidas de esportes praticados em campos gramados para detectar as regiões que retratem a grama do campo.

A detecção de grama tem a sua utilidade na sumarização de vídeos desportivos (geração automática de resumos das partidas, contendo apenas os melhores momentos), através da separação de tomadas de longa distância e de curta distância e a posição da bola com relação ao campo de jogo, entre outros [31].

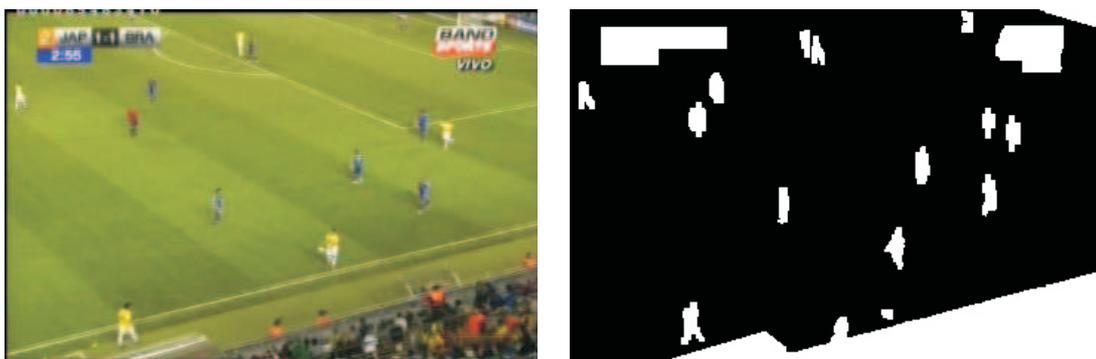
As tomadas de longa distância são, em geral, mais interessantes para o espectador do que as de curta distância, como *replays* e tomadas da torcida, por exemplo. Já a posição da bola com relação ao campo de jogo pode revelar lances de proximidade do gol que também são de alto interesse.

Wang *et al.* [58] propuseram um método de detecção de grama baseado na aplicação de limiares nos valores dos componentes de cor da imagem RGB: inicialmente cada imagem é dividida em janelas com tamanho 16×16 *pixels* e logo após se realiza a análise de cor dominante em busca das janelas que contenham maior presença das tonalidades esverdeadas. Isto consiste em verificar se as diferenças entre a componente G com os demais canais ($G - B$ e $G - R$) são maiores que um limiar T . De acordo com as características da base de dados empregada no artigo, caso o limiar seja extrapolado, tem-se que o *pixel* é considerado pertencente à área de grama do quadro em questão.

Outra técnica, proposta por Firmino Jr. e Gomes [11], consiste em extrair amostras aleatórias de uma base de imagens de treinamento para compor protótipos de *pixels* grama e não-grama. Para decidir se um *pixel* de uma imagem que se quer segmentar faz parte da área de grama, computam-se as menores distâncias entre o *pixel* e os protótipos extraídos da base de treinamento.

4.1.1 Conjunto de Imagens de Treinamento

Para utilizar o método de obtenção de segmentadores apresentado nesta dissertação, a base de dados de treinamento utilizada no trabalho de Firmino Jr. e Gomes [11] foi reaproveitada, por ter sido cuidadosamente obtida de forma manual. Ela, porém, não apresenta regiões de indeterminação, o que possivelmente prejudicou um pouco os resultados obtidos. A Figura 4.1 é um exemplo de imagem de treinamento com sua respectiva segmentação manual.



(a) Imagem original

(b) Imagem Segmentada Manualmente

Figura 4.1: Exemplo de imagem do conjunto de treinamento para detecção de grama, com sua respectiva segmentação manual.

Ao todo, a base contém 50 quadros de diversas partidas de futebol disputadas durante

Estratégia	Filtragem	Ambiguidade	Repetições	Proporcionalidade	Medida-F
E_0					93,52%
E_1					85,35%
E_2					66,51%
E_3					85,51%
E_4					97,38%

Tabela 4.1: Evolução das estratégias de treinamento.

a Copa do Mundo FIFA de 2006. Todas as imagens foram capturadas de transmissões de televisão, e apresentam resolução de 360×240 *pixels*.

Destas imagens, 25 – exatamente a metade – foram destinadas a compor o conjunto de treinamento a ser utilizado como insumo para obtenção automática do segmentador de grama, e as 25 restantes utilizadas como conjunto de testes, para ser viabilizada a medição das medida-F do segmentador obtido com o método proposto e dos demais segmentadores usados para comparação.

4.1.2 Obtenção Automática do Segmentador

Evolução da Estratégia de Amostragem

A estratégia de amostragem para construção dos conjuntos de treinamento e teste descrita no Capítulo 3 foi consolidada através de experimentos preliminares conduzidos durante a aplicação do método proposto no estudo de caso da detecção de grama em quadros de vídeos desportivos. Os resultados destes experimentos estão apresentados na Tabela 4.1.

Inicialmente, a estratégia E_0 utilizada para amostrar os *pixels* garantia que fossem retiradas quantidades iguais de amostras das classes P e N de cada imagem. Isto garantia que seria utilizada para treinamento a mesma quantidade de padrões para cada classe ao final da amostragem de todas as imagens.

Entretanto, a estratégia não contava com mecanismos de filtragem de sobreposições entre amostras de classes diferentes, isto é: caso um *pixel* fosse amostrado para uma classe, e depois o mesmo *pixel* ocorresse em uma classe diferente, ele seria incluído em duplicidade no conjunto de treinamento, e associado a classes diferentes. Isto provocava com que a rede

neural fosse treinada com padrões ambíguos, o que significa que o conjunto de treinamento não se tratava de uma amostra de pontos de uma função matemática. Isto vai de encontro com os preceitos teóricos das Redes Neurais Artificiais, que são aproximadores universais de funções [16], e causa problemas em seu treinamento [47].

Suspeitou-se, desta forma, que a medida-F obtida, apesar de expressiva, estava sendo prejudicada com a presença de ambiguidades no conjunto de treinamento. A estratégia E_1 foi concebida para retirar este fenômeno da estratégia anterior.

A retirada da ambiguidade funcionou da seguinte maneira: caso um mesmo padrão de entrada seja encontrado em classes diferentes no conjunto de treinamento, será considerado apenas uma vez e na classe em que aparecer majoritariamente. Por exemplo, se o padrão $\{12, 34, 56\}$ aparecer nas imagens como P 150 vezes, e como N 20 vezes, ele será utilizado apenas uma vez na classe P .

No entanto, foram retiradas também todas as repetições intraclasse, o que provocou uma drástica redução na quantidade de *pixels* amostrados: 1.271.840 amostras em E_0 e apenas 31.700 em E_1 , uma redução de mais de 97%.

Esta redução da quantidade de amostras provocou a queda da medida-F da estratégia E_1 com relação à medida-F da estratégia E_0 e evidenciou a necessidade de maior volume de padrões no conjunto de treinamento, já observada anteriormente [28].

Com o objetivo de aumentar a quantidade de amostras obtidas no conjunto de treinamento, foi retirada também a filtragem de amostras nas imagens na estratégia E_2 , ou seja, todos os *pixels* das imagens passaram a ser amostrados, passando a não ser mantida a mesma quantidade de padrões de cada classe ao final da amostragem de todas as imagens.

Isto fez com que 137.430 passassem a ser amostrados, mas provocou uma redução ainda maior na medida-F, levando à conclusão de que o balanceamento do número de padrões entre as classes era ainda mais importante para a manutenção da medida-F dos segmentadores.

Foi reintroduzida na estratégia E_3 a equalização do número de amostras para cada classe. No entanto, de maneira diferente da E_1 , esta equalização só ocorria no final da montagem do conjunto de treinamento, em vez de ser feita a cada imagem, o que permitiu aumentar o número de amostras (totalizando 40.190) e consequente recuperação da medida-F.

Finalmente, de modo a permitir um grande aumento no número de amostras, mantendo a não ambiguidade no conjunto, a retirada de ambiguidade passou a ser efetuada de um modo

Espaço de Cor	D_r	$\sigma(D_r)$	P_r	$\sigma(P_r)$	F_m	$\sigma(F_m)$
HLS	89,09%	0,0163	88,61%	0,0099	88,84%	0,0118
HSV	94,38%	0,0053	91,50%	0,0064	92,92%	0,0049
L*a*b*	96,39%	0,0022	94,79%	0,0020	95,58%	0,0011
L*u*v*	96,21%	0,0022	95,14%	0,0013	95,67%	0,0005
RGB	96,50%	0,0029	95,14%	0,0026	95,81%	0,0004
XYZ	96,16%	0,0024	95,19%	0,0016	95,67%	0,0008
YCbCr	96,59%	0,0028	95,21%	0,0022	95,89%	0,0004

Tabela 4.2: Determinação do melhor espaço de cor para segmentação de grama.

diferente do realizado na E_1 : caso um mesmo padrão de entrada seja encontrado em classes diferentes no conjunto de treinamento, será utilizado uma quantidade $q = \|n(P) - n(N)\|$ de vezes na classe majoritária, onde $n(P)$ e $n(N)$ são as quantidades de ocorrências do padrão nas classes P e N , respectivamente.

Por exemplo, se desta vez o padrão $\{12, 34, 56\}$ aparecer nas imagens 150 vezes como P e 20 vezes como N , ele será replicado apenas 130 ($150 - 20$) vezes na classe P .

Esta estratégia permitiu aumentar a quantidade de amostras (totalizando 1.240.140) consideradas no conjunto de treinamento e conseqüentemente a medida-F do segmentador, tornando-a a mais elevada entre as obtidas com as diversas estratégias experimentadas. Desta forma, a estratégia E_4 passou a ser considerada nos experimentos subsequentes desta dissertação, compondo o método de automatização de ajuste e obtenção de segmentadores proposto.

Obtenção do Segmentador Neural Baseado em Informação de Cor

As Tabelas 4.2 a 4.5 exibem os resultados obtidos para cada valor de parâmetro durante a determinação dos melhores parâmetros para a rede neural que segmenta as imagens com base apenas em informação de cor.

O espaço de cores YCbCr obteve um destaque sensível em relação aos demais espaços de cores avaliados, e foi observado um comportamento monotônico no crescimento da medida-F com respeito à quantidade de *bits* por canal de cor. Os demais parâmetros não se mostraram

Profundidade de Cor	D_r	$\sigma(D_r)$	P_r	$\sigma(P_r)$	F_m	$\sigma(F_m)$
1 bit por canal (2 níveis)	93,97%	0,0003	73,41%	0,0127	82,42%	0,0163
2 bits por canal (4 níveis)	98,77%	0,0190	77,50%	0,0154	86,82%	0,0163
4 bits por canal (16 níveis)	98,57%	0,0006	92,07%	0,0013	95,21%	0,0163
5 bits por canal (32 níveis)	96,16%	0,0033	94,66%	0,0024	95,40%	0,0163
6 bits por canal (64 níveis)	96,61%	0,0034	94,85%	0,0029	95,72%	0,0163
7 bits por canal (128 níveis)	96,35%	0,0009	95,16%	0,0009	95,75%	0,0163
8 bits por canal (256 níveis)	96,59%	0,0028	95,21%	0,0022	95,89%	0,004

Tabela 4.3: Determinação da melhor profundidade de cor para segmentação de grama (YCbCr).

Tamanho da Janela	D_r	$\sigma(D_r)$	P_r	$\sigma(P_r)$	F_m	$\sigma(F_m)$
1 pixels (1 × 1)	96,34%	0,0028	95,37%	0,0022	95,85%	0,0004
9 pixels (3 × 3)	96,56%	0,0033	95,06%	0,0038	95,81%	0,0007
25 pixels (5 × 5)	96,52%	0,0029	95,18%	0,0029	95,85%	0,0003
49 pixels (7 × 7)	96,59%	0,0026	95,21%	0,0029	95,89%	0,0003
81 pixels (9 × 9)	96,85%	0,0026	94,81%	0,0037	95,82%	0,0009

Tabela 4.4: Determinação do melhor tamanho da janela de pixels para segmentação de grama (YCbCr, 8 bits por canal).

Neurônios	D_r	$\sigma(D_r)$	P_r	$\sigma(P_r)$	F_m	$\sigma(F_m)$
1	97,92%	0,0035	92,92%	0,0069	95,35%	0,0023
2	96,92%	0,0063	94,46%	0,0134	95,67%	0,0044
3	97,02%	0,0029	94,73%	0,0039	95,86%	0,0008
4	96,77%	0,0036	94,97%	0,0046	95,86%	0,0008
5	96,61%	0,0036	95,19%	0,0038	95,90%	0,0005
6	96,43%	0,0029	95,33%	0,0031	95,88%	0,0005
7	96,59%	0,0026	95,21%	0,0029	95,89%	0,0003
8	96,37%	0,0029	95,35%	0,0028	95,86%	0,0003
9	96,47%	0,0031	95,33%	0,0026	95,90%	0,004
10	96,48%	0,0033	95,31%	0,0033	95,89%	0,006
11	96,42%	0,0019	95,39%	0,0013	95,90%	0,0005
12	96,26%	0,0031	95,48%	0,0022	95,87%	0,0006
13	96,41%	0,0038	95,35%	0,0033	95,88%	0,0003
14	96,14%	0,0021	95,57%	0,0017	95,85%	0,0002

Tabela 4.5: Determinação da melhor quantidade de neurônios na camada oculta para segmentação de grama (YCbCr, 8 bits por canal, janela de 7×7 pixels).

tão determinantes para a separabilidade das regiões P e N , uma vez que não houve grandes variações da medida- F com a mudança dos mesmos.

Após estes experimentos, foi conduzida a investigação acerca da influência do posicionamento dos *pixels* na separabilidade das regiões de grama das imagens do conjunto de treinamento.

Intuitivamente, a grama em um quadro de vídeo desportivo tende a se localizar na parte inferior, uma vez que ela compõe o solo em que os atletas estarão disputando as suas partidas. Isto pode ser comprovado observando o histograma apresentado na Figura 4.2, em que a concentração de ocorrências de grama se dá nos maiores valores da coordenada y dos *pixels*, uma vez que a coordenada $(0,0)$ das imagens está localizada no seu canto superior esquerdo.

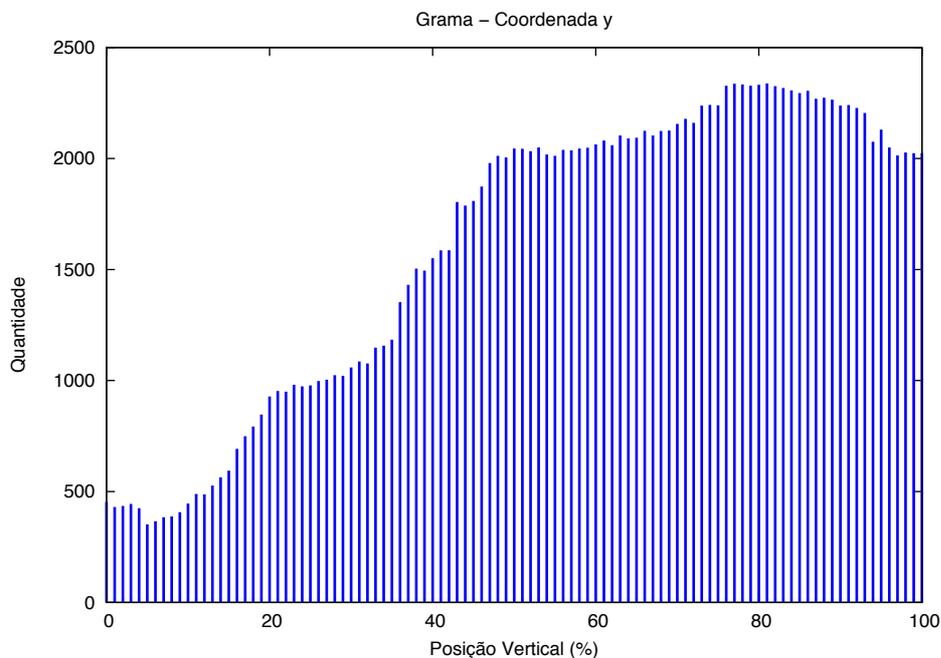


Figura 4.2: Histograma dos pixels de grama base de imagens de partidas de futebol, com respeito à distância relativa ao topo das imagens.

A média dos valores no histograma é $\mu^g = 1587,25$ e o desvio padrão $\sigma^{g_v} = 685,52$, o que leva a um coeficiente de variação $c_v^{g_v} = 0,4329$. Desta maneira, apesar de estarem distribuídos com tendência na parte de baixo da imagem, a coordenada vertical não é utilizada na segmentação de grama, uma vez que seu coeficiente de variação é inferior ao limiar de 0,5 utilizado nesta dissertação.

No que diz respeito à coordenada horizontal, a grama em um quadro de vídeo desportivo tende a se espalhar de maneira praticamente uniforme ao longo da extensão. A Figura 4.3

corroborar com esta intuição.

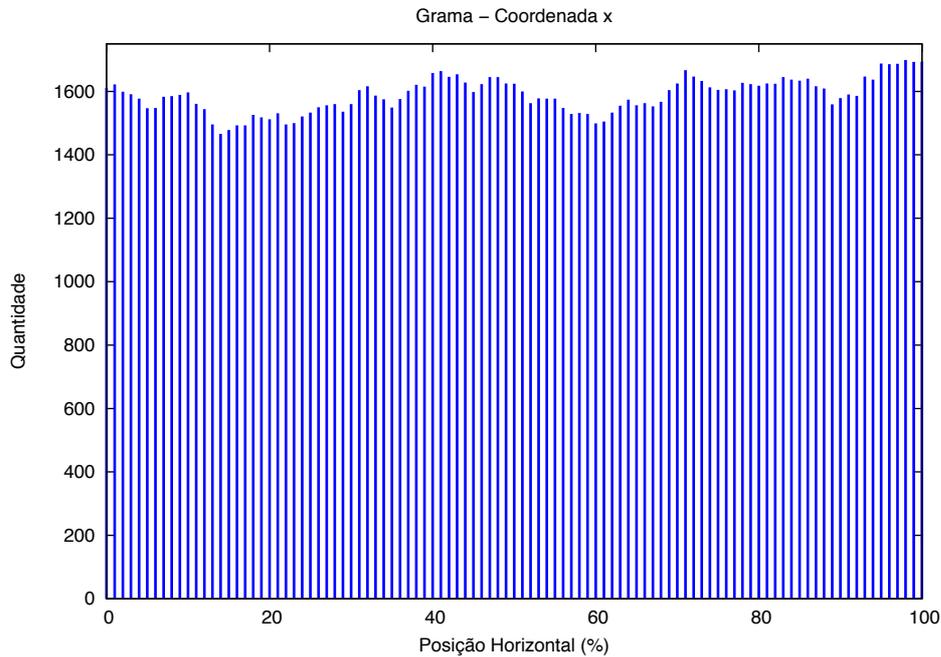


Figura 4.3: Histograma dos pixels de grama de todas as imagens da base de partidas de futebol, com respeito à distância relativa à margem esquerda das imagens.

A média dos valores no histograma é $\mu^g = 1587,25$ – deve ser idêntica à média da coordenada vertical – e o desvio padrão $\sigma^{g_h} = 52,49$, o que leva a um coeficiente de variação $c_v^{g_h} = 0,0331$. Como intuitivamente esperado, a coordenada horizontal também não é utilizada na segmentação de grama.

Deste modo, o treinamento final da rede neural artificial que compõe o segmentador baseado em informação de cor consistiu de uma repetição do treinamento da rede neural para YCbCr, amostrando em 8 bits por canal de cor e com uma janela de 7×7 pixels.

O treinamento final foi executado até que se completassem 1000 épocas e o segmentador obtido apresentou taxa de detecção de 96,70% e precisão de 96,82%, totalizando uma medida-F de 96,76%, com respeito à base de imagens de treinamento.

Obtenção do Segmentador Baseado em Informação de Textura e Cor

A execução do algoritmo genético descrito na Seção 3.3 resultou na determinação dos valores para os parâmetros do segmentador baseado em informação de textura e cor que constam na Tabela 4.6.

Parâmetro	Descrição	Valor
M	Número de escalas dos filtros de Gabor	1
N	Número de orientações dos filtros de Gabor	4
l	Limiar inferior da binarização	0
u	Limiar superior da binarização	7

Tabela 4.6: Determinação dos parâmetros do segmentador baseado em informação de textura e cor para detecção de grama.

Após a introdução da informação de textura como insumo para segmentação, ocorreu a esperada redução no valor da medida-F do segmentador para $F^{gct} = 89,32\%$ ($D_r^{gct} = 81,05\%$ e $P^{gct} = 99,46\%$), com respeito ao conjunto de imagens de treinamento.

Expansão de Regiões para Melhoria da Medida-F

O último passo da obtenção do segmentador de grama foi a execução do algoritmo genético descrito na Seção 3.4, que resultou na determinação dos valores para os parâmetros para a expansão de regiões com o algoritmo *floodfill* adaptado que constam na Tabela 4.7.

Parâmetro	Descrição	Valor
a	Limiar de distância de cor da região já marcada.	18,5753
b	Limiar de distância de cor da borda da região marcada	232,634

Tabela 4.7: Determinação dos parâmetros para expansão de regiões na detecção de grama.

Após a expansão das regiões com o algoritmo *floodfill* adaptado, ocorreu a esperada recuperação do valor da medida-F do segmentador para $F^{gct} = 95,61\%$ ($D_r^{gct} = 95,36\%$ e $P^{gct} = 95,85\%$), com respeito ao conjunto de imagens de treinamento.

No entanto, a recuperação não foi suficiente para atingir um valor de medida-F superior ao do segmentador baseado apenas em informação de cor, que obteve $F^{gct} = 96,76\%$. Por este motivo, considera-se o segmentador baseado apenas em informação de cor como o resultado da aplicação do método proposto nesta dissertação para a detecção de grama.

4.1.3 Teste do Segmentador de Grama

O segmentador de grama obtido na aplicação do método proposto nesta dissertação foi executado nas 25 imagens que compunham o conjunto de testes para este material.

O segmentador obtido foi capaz de detectar 96,68% da área correspondente a grama, com precisão de 97,55%. Isto resulta em um medida-F com o valor de 97,11%. A Figura 4.4 apresenta exemplos de quadros de vídeos desportivos processadas pelo segmentador obtido.

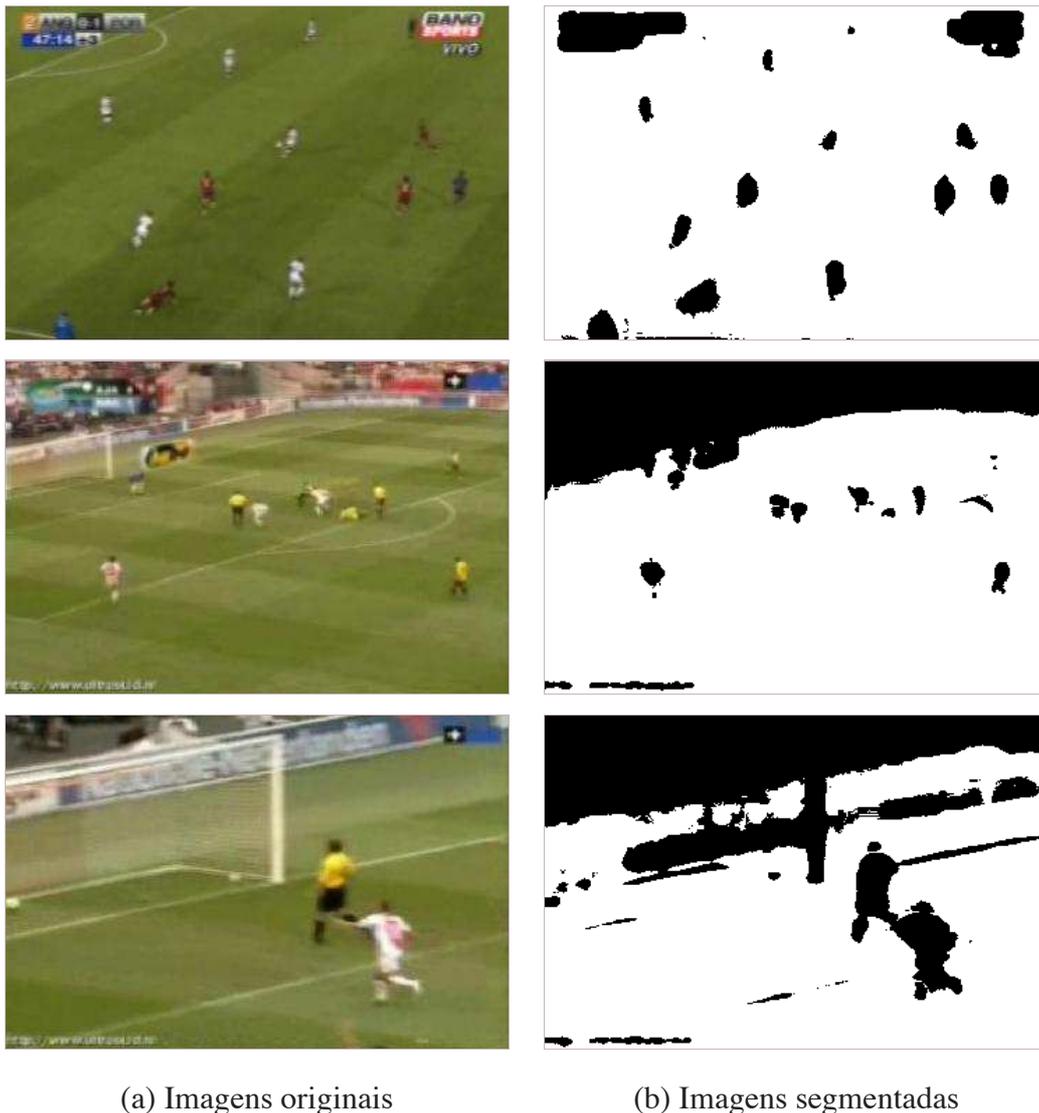


Figura 4.4: Resultados da aplicação segmentador obtido utilizando o método proposto ao problema da detecção de grama (grama em branco e demais regiões em preto).

Nesta figura, a primeira imagem apresentada é o elemento do conjunto de teste que obteve maior medida-F entre os demais. A segunda imagem é a que obteve medida-F mais próxima do valor obtido pelo conjunto e a terceira é a que obteve a menor medida-F de todas.

Percebe-se que o segmentador de grama não é capaz de fazer uma separação clara das marcações do campo de jogo. Além disto, em fundos complexos como na terceira imagem na região em que a grama aparece através da rede, o segmentador apresenta degradação nos resultados.

Também foi testado o segmentador da fase 3 do método proposto, que foi capaz de detectar 95,97% da área correspondente a grama, com precisão de 97,01%, resultando em uma medida-F com o valor de 96,49%, verificando também no conjunto de testes que o segmentador da fase 1 obtém melhores resultados.

Para permitir uma comparação, também foram testados um classificador *bayesiano* – gerado através da plataforma de mineração de dados Weka [60] – e com os segmentadores de grama propostos por Firmino Jr. e Gomes [11] (*template matching*) e por Wang *et al.* [58] (*limiarização de cores*), em um conjunto de imagens de teste composto pelas demais imagens da base de dados da qual foi extraído o conjunto de treinamento. A Tabela 4.8 exhibe os resultados obtidos entre os segmentadores comparados.

Segmentador	Tx. Detecção	Precisão	Medida-F
Automatizado	96,68%	97,55%	97,11%
<i>Naïve Bayes</i>	96,64%	96,93%	96,78%
Firmino Jr. e Gomes [11]	98,12%	92,79%	95,38%
Wang <i>et al.</i> [58]	44,33%	97,81%	61,01%

Tabela 4.8: Resultados obtidos para detecção de grama

O segmentador de Wang *et al.* [58] teve baixa taxa de detecção devido ao seu limiar ter sido fixado empiricamente em vez de ajustado com a utilização de técnicas de aprendizado de máquina. Como o conjunto de imagens utilizado para treinamento foi diferente, sugere-se que houve *overfitting*. Os demais segmentadores apresentam resultados bem próximos aos da utilização do método proposto.

Observa-se que o método foi capaz de dar origem a um segmentador cujos resultados além de serem compatíveis com os resultados dos demais segmentadores testados, chegam a ser superiores a todos os outros.

4.2 Detecção de Pele Humana em Fotografias

O problema da detecção de pele humana em fotografias é uma instância da segmentação de imagens para detecção de materiais que consiste em indicar quais *pixels* de uma determinada imagem correspondem à pele humana.

A detecção de pele humana é de grande importância em uma série de domínios de aplicação por ser um passo intermediário para possibilitar a redução do espaço de busca para tarefas de detecção de nudez em fotografias/vídeos [12], reconhecimento de faces [37, 22] e detecção de seres humanos para fins de segurança [22].

Para resolver o problema da detecção de pele humana em fotografias, Jones e Regh [26] propuseram uma técnica baseada em histogramas utilizando um classificador *Bayesiano*. Três histogramas são gerados a partir de um conjunto de imagens de treinamento (previamente segmentado de maneira manual): um correspondente à distribuição de cores no conjunto completo, outro levando em consideração os *pixels* correspondentes à pele humana, e um terceiro que representa o modelo de cores para não-pele. O conjunto de imagens fornecido pelos autores, apesar de conter sérios problemas de segmentação, é largamente utilizado para teste (e treinamento) de classificadores para segmentação de pele humana [5, 10], porém a técnica dos mesmos não utiliza informações de textura.

Também foram propostas soluções baseadas em limiarização em cada canal de cor [37, 12]. Em outras palavras, determina-se uma região contígua no espaço de cores utilizado, de modo a maximizar a acurácia de testes do segmentador gerado. Esta técnica, além de não levar em consideração informações de textura, limita o formato da região encontrada a hiper-paralelogramos no hiperespaço formado pelos canais do espaço de cores.

As próprias redes neurais artificiais, do tipo *Multilayer Perceptron*, já foram utilizadas para este propósito, em comparação com outras técnicas de classificação, com grande sucesso [42], bem como também foi utilizado o algoritmo Expectation/Maximization com restrições espaciais [10]. Os resultados foram mais satisfatórios que os obtidos por Jones e Regh [26], porém as informações de textura também não foram utilizadas nesta última solução.

4.2.1 Conjunto de Imagens de Treinamento

Para utilizar o método de segmentação apresentado nesta dissertação, um subconjunto da base de imagens de treinamento fornecida por Jones e Regh [26] foi re-rotulado de maneira totalmente manual, formando um total de 119 imagens de treinamento. A Figura 3.4 (página 22) é um exemplo de imagem de treinamento com sua respectiva segmentação manual, já levando em consideração as regiões de indeterminação, que serão desconsideradas pelo segmentador.

Destas imagens, 60 foram destinadas a compor o conjunto de treinamento a ser utilizado como insumo para obtenção automática do segmentador de pele, e as 59 restantes utilizadas como conjunto de testes, para ser viabilizada a medição das medidas-F do segmentador obtido com o método proposto e dos demais segmentadores usados para comparação.

4.2.2 Obtenção Automática do Segmentador

Obtenção do Segmentador Neural Baseado em Informação de Cor

As Tabelas 4.9 a 4.12 exibem os resultados obtidos para cada valor de parâmetro durante a determinação dos melhores parâmetros para a rede neural que segmenta as imagens baseada apenas em informação de cor.

Espaço de Cor	D_r	$\sigma(D_r)$	P_r	$\sigma(P_r)$	F_m	$\sigma(F_m)$
HLS	80,46%	0,0211	30,74%	0,0265	44,40%	0,0254
HSV	74,26%	0,0136	42,08%	0,0217	53,68%	0,0161
L*a*b*	76,37%	0,0095	42,04%	0,0218	54,19%	0,0183
L*u*v*	72,73%	0,0463	42,64%	0,0174	53,72%	0,0221
RGB	73,86%	0,0213	45,66%	0,0204	56,38%	0,0142
XYZ	74,14%	0,0129	38,94%	0,0083	51,05%	0,0069
YCbCr	74,52%	0,0387	46,27%	0,0182	57,02%	0,0133

Tabela 4.9: Determinação do melhor espaço de cor para detecção de pele.

O espaço de cores YCbCr obteve um destaque sensível em relação aos demais espaços de cores avaliados, e foi observado um comportamento crescente da medida-F até 6 bits por

Profundidade de Cor	D_r	$\sigma(D_r)$	P_r	$\sigma(P_r)$	F_m	$\sigma(F_m)$
1 bit por canal (2 níveis)	20,00%	0,4216	1,65%	0,0348	3,05%	0,0643
2 bits por canal (4 níveis)	10,00%	0,3162	0,83%	0,0261	1,52%	0,0482
3 bits por canal (8 níveis)	38,77%	0,0080	53,34%	0,0195	44,87%	0,0028
4 bits por canal (16 níveis)	57,08%	0,0121	48,90%	0,0159	52,60%	0,0060
5 bits por canal (32 níveis)	68,49%	0,0242	49,47%	0,0157	57,42%	0,0129
6 bits por canal (64 níveis)	72,72%	0,0090	49,57%	0,0106	58,95%	0,0073
7 bits por canal (128 níveis)	73,76%	0,0416	48,04%	0,0145	58,11%	0,0128
8 bits por canal (256 níveis)	74,52%	0,0387	46,27%	0,0182	57,02%	0,0133

Tabela 4.10: Determinação da melhor profundidade de cor para detecção de pele (YCbCr).

Tamanho da Janela	D_r	$\sigma(D_r)$	P_r	$\sigma(P_r)$	F_m	$\sigma(F_m)$
1 pixel (1 × 1)	72,72%	0,0090	49,57%	0,0106	58,95%	0,0073
9 pixels (3 × 3)	73,67%	0,0096	49,27%	0,0151	59,03%	0,0099
25 pixels (5 × 5)	73,78%	0,0124	48,47%	0,0225	58,46%	0,0144
49 pixels (7 × 7)	73,75%	0,0094	48,94%	0,0134	58,82%	0,0087
81 pixels (9 × 9)	71,01%	0,0717	49,29%	0,0127	57,99%	0,0234

Tabela 4.11: Determinação do melhor tamanho da janela de *pixels* para detecção de pele (YCbCr, 6 bits por canal).

canal de cor, seguido de uma queda nesta métrica. A redução na profundidade de cor já havia sido verificada por Jones e Rehg [26] (em seu estudo, foi determinada uma profundidade de 5 bits por canal de cor, provavelmente devido às diferenças entre as bases de imagens utilizadas) e foi confirmada neste experimento.

O tamanho da janela de *pixels* não se mostrou determinante para a separabilidade das regiões P e N , uma vez que não houve grandes variações da medida-F com a mudança deste parâmetro.

A quantidade de neurônios na camada escondida, por sua vez, apresentou comportamento crescente seguido de estabilização a partir de 10 neurônios na camada escondida, o que sugere que aumentar este parâmetro não trará melhora à medida-F.

Após estes experimentos, foi conduzida a investigação acerca da influência do posicionamento dos *pixels* na separabilidade das regiões de pele das imagens do conjunto de treina-

Neurônios	D_r	$\sigma(D_r)$	P_r	$\sigma(P_r)$	F_m	$\sigma(F_m)$
1	76,72%	0,0578	41,97%	0,0753	53,52%	0,0679
2	73,03%	0,0348	45,52%	0,0291	55,97%	0,0201
3	69,33%	0,0523	48,40%	0,0203	56,86%	0,0150
4	72,90%	0,0313	47,09%	0,0282	57,15%	0,0229
5	71,82%	0,0369	48,00%	0,0231	57,46%	0,0186
6	71,93%	0,0455	48,00%	0,0250	57,46%	0,0202
7	72,43%	0,0383	48,94%	0,0133	58,37%	0,0169
8	68,92%	0,0751	50,13%	0,0121	57,86%	0,0290
9	72,90%	0,0217	49,60%	0,0156	59,01%	0,0139
10	72,97%	0,0254	49,70%	0,0116	59,11%	0,0145
11	73,59%	0,0119	49,91%	0,0078	59,47%	0,0033
12	72,51%	0,0327	50,17%	0,0072	59,28%	0,0140
13	73,44%	0,0108	49,99%	0,0093	59,48%	0,0052
14	73,09%	0,0321	49,98%	0,0087	59,34%	0,0131

Tabela 4.12: Determinação da melhor quantidade de neurônios na camada oculta para detecção de pele (YCbCr, 6 bits por canal, janela de 3×3 pixels).

mento.

A princípio, a pele humana em uma fotografia tende a se localizar em todas as extensões do eixo superior, uma vez que depende da distância do enquadramento das fotos, do traje utilizado, da quantidade das pessoas e de sua faixa etária, dentre outros. Isto pode ser comprovado observando o histograma apresentado na Figura 4.5, em que as ocorrências de pele variam pouco ao longo do eixo vertical.

A média dos valores no histograma é $\mu^p = 504,75$ e o desvio padrão $\sigma^{pv} = 212,90$, o que leva a um coeficiente de variação $c_v^{pv} = 0,4218$. Desta maneira, apesar de estarem distribuídos com tendência na parte de baixo da imagem, a coordenada vertical não é utilizada na segmentação de pele, de acordo com o método proposto nesta dissertação.

No que diz respeito à coordenada horizontal, a pele em um quadro de vídeo desportivo tende a se concentrar na parte central das imagens, devido à centralização das pessoas pelos fotógrafos. A Figura 4.6 corrobora esta intuição.

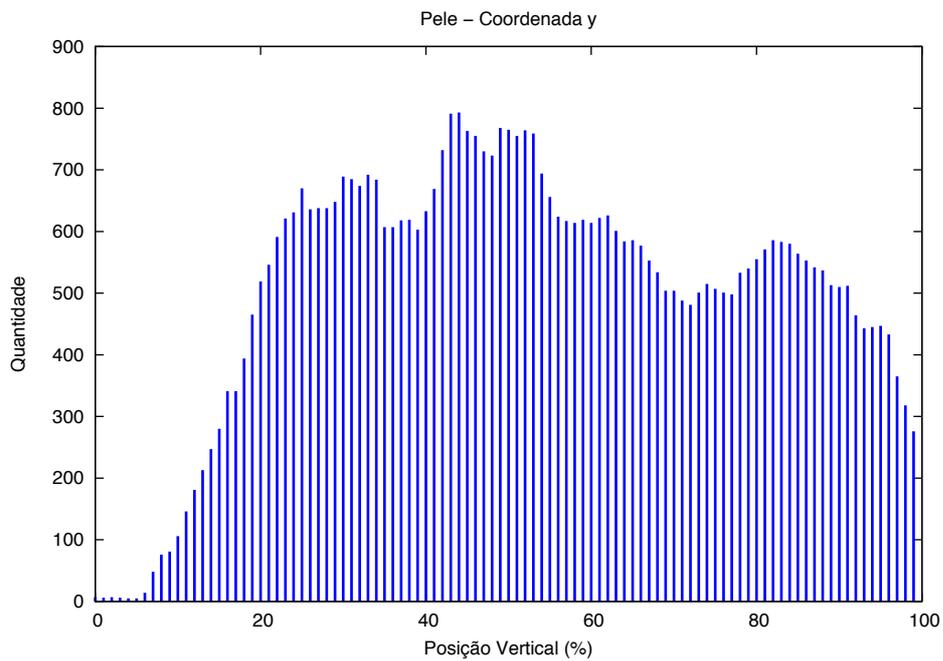


Figura 4.5: Histograma dos pixels de pele na base de imagens de treinamento, com respeito à distância relativa ao topo das imagens.

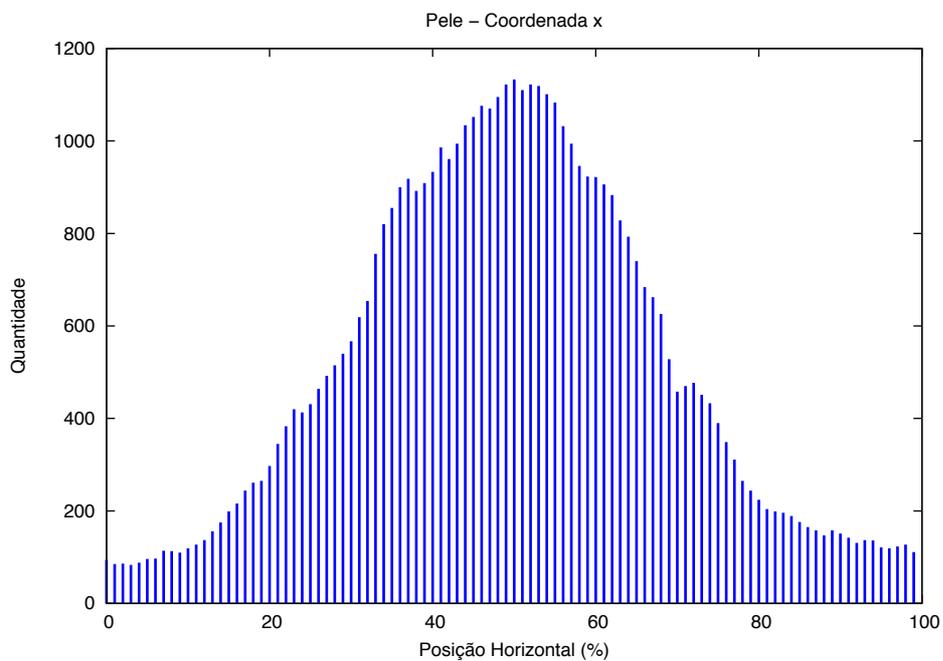


Figura 4.6: Histograma dos pixels de pele na base de imagens de treinamento, com respeito à distância relativa à margem esquerda das imagens.

A média dos valores no histograma é $\mu^p = 504,75$ e o desvio padrão $\sigma^{ph} = 364,24$, o que leva a um coeficiente de variação $c_v^{ph} = 0,7216$, maior que o limiar estabelecido em 0,5. Como intuitivamente esperado, a coordenada horizontal deve, então, ser utilizada na segmentação de pele.

O treinamento final foi realizado utilizando o espaço de cores YCbCr, com 6 *bits* por canal de cor, usando uma janela de 3×3 *pixels* e 13 neurônios na camada escondida da rede neural. Conforme verificado na análise do posicionamento dos *pixels*, a coordenada horizontal foi incluída como entrada da rede neural neste treinamento.

O segmentador neural baseado em informação de cor obtido apresentou medida-F $F = 68,81\%$ ($D_r = 74,77\%$ e $P = 63,72\%$) com respeito ao conjunto de imagens de treinamento. Percebeu-se um aumento de 9,33 pontos percentuais na medida-F em relação ao segmentador com os mesmos parâmetros sem considerar a coordenada horizontal, devido unicamente ao aumento de 13,73% que foi introduzido na precisão do segmentador.

Obtenção do Segmentador Baseado em Informação de Textura e Cor

A execução do algoritmo genético descrito na Seção 3.3 resultou na determinação dos valores para os parâmetros do segmentador baseado em informação de textura e cor que constam na Tabela 4.13.

Parâmetro	Descrição	Valor
M	Número de escalas dos filtros de Gabor	2
N	Número de orientações dos filtros de Gabor	1
l	Limiar inferior da binarização	3
u	Limiar superior da binarização	19

Tabela 4.13: Determinação dos parâmetros do segmentador baseado em informação de textura e cor para detecção de pele.

Após a introdução da informação de textura como insumo para segmentação, ocorreu a esperada redução no valor da medida-F do segmentador para $F^{gct} = 49,88\%$ ($D_r^{gct} = 40,53\%$ e $P^{gct} = 64,85\%$).

Expansão de Regiões para Melhoria da Medida-F

O último passo da obtenção do segmentador de pele foi a execução do algoritmo genético descrito na Seção 3.4, que resultou na determinação dos valores dos parâmetros para a expansão de regiões com o algoritmo *floodfill* adaptado. Os resultados obtidos encontram-se na Tabela 4.14.

Parâmetro	Descrição	Valor
a	Limiar de distância de cor da região já marcada.	9,72294
b	Limiar de distância de cor da borda da região marcada	43,3425

Tabela 4.14: Determinação dos parâmetros para expansão de regiões na detecção de pele.

Após a expansão das regiões com o algoritmo *floodfill* adaptado, ocorreu a esperada recuperação do valor da medida-F do segmentador para 57,84% ($D_r^{gct} = 63,37\%$ e $P^{gct} = 53,20\%$), com respeito ao conjunto de imagens de treinamento.

No entanto, a recuperação não foi suficiente para atingir um valor de medida-F superior ao do segmentador baseado apenas em informação de cor, que obteve $F^{gct} = 68,81\%$. Por este motivo, considera-se o segmentador baseado apenas em informação de cor como o resultado da aplicação do método proposto nesta dissertação para a detecção de pele humana em fotografias.

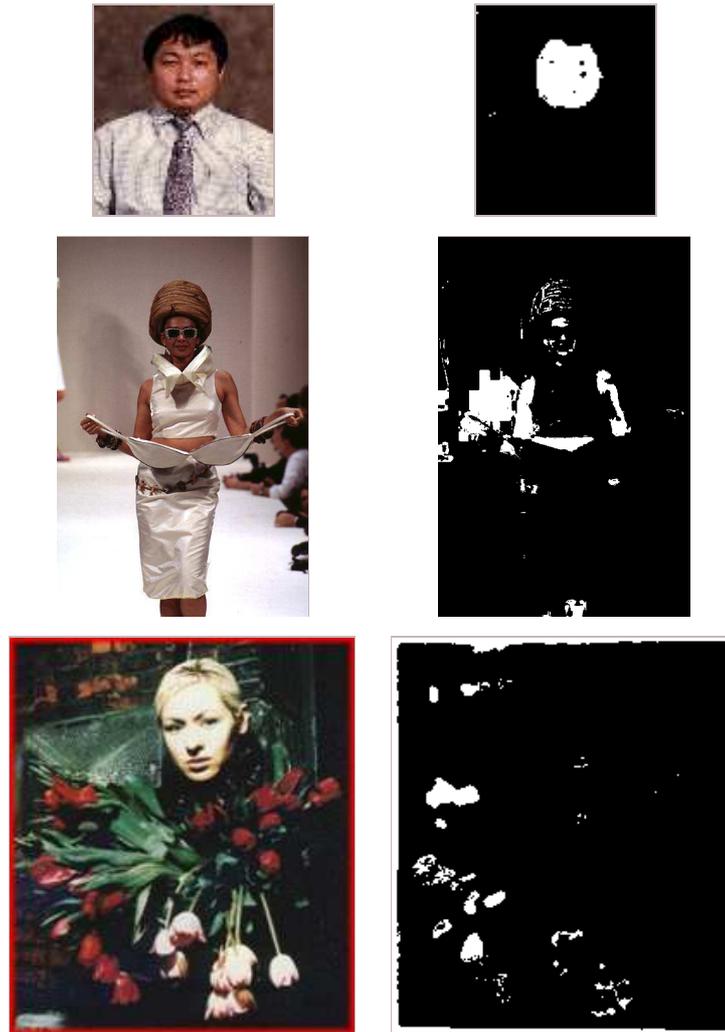
4.2.3 Teste do Segmentador de Pele

O segmentador de pele obtido na aplicação do método proposto nesta dissertação foi testado nas 59 imagens que compunham o conjunto de testes para este material.

O segmentador obtido foi capaz de detectar 52,82% da área correspondente a pele, com precisão de 49,31%. Isto resulta em uma medida-F com o valor de 51,01%. A Figura 4.7 apresenta exemplos de fotografias processadas pelo segmentador obtido.

Nesta figura, a primeira imagem apresentada é o elemento do conjunto de teste que obteve maior medida-F entre os demais. A segunda imagem é a que obteve medida-F mais próxima do valor obtido pelo conjunto e a terceira é a que obteve a menor medida-F de todas.

Percebe-se que o comportamento do segmentador apresenta grandes variações entre as imagens. No primeiro exemplo, a separação é feita com sucesso, mesmo se tratando de um



(a) Imagens originais

(b) Imagens segmentadas

Figura 4.7: Resultados da aplicação segmentador obtido utilizando o método proposto ao problema da detecção de pele (pele em branco e demais regiões em preto).

fundo que pode ser confundido com tons de pele humana. Já no último exemplo, apenas uma pequena região do lábio da modelo é detectada, ao passo em que as regiões vermelhas da borda e das rosas são confundidas com regiões de pele.

Também foi testado o segmentador da fase 3 do método proposto, que foi capaz de detectar 57,44% da área correspondente a pele, com precisão de 43,46%, resultando em um medida-F com o valor de 49,49%, verificando também no conjunto de testes que o segmentador da fase 1 obtém melhores resultados.

Para permitir uma comparação com abordagens existentes, também foram testados um classificador *bayesiano* gerado através da plataforma de mineração de dados Weka, um de-

tecor de pele que utiliza o modelo de pele RGB-H-CbCr proposta por Rahman *et al.* [44] e, por fim, um segmentador baseado na técnica de *cluster* no espaço de cores HSV, proposto por Herodotou *et al.* [19], no mesmo conjunto de testes em que o segmentador obtido automaticamente foi testado. A Tabela 4.15 exhibe os resultados obtidos entre os segmentadores comparados.

Segmentador	Tx. Detecção	Precisão	Medida-F
Automatizado	52,82%	49,31%	51,01%
<i>Naïve Bayes</i>	84,50%	35,17%	49,67%
Rahman <i>et al.</i> [44]	66,56%	29,40%	40,78%
Herodotou <i>et al.</i> [19]	83,34%	29,97%	44,09%

Tabela 4.15: Resultados obtidos para detecção de pele

Observa-se que o método foi capaz de dar origem a um segmentador cujos resultados são compatíveis com os resultados dos demais segmentadores testados além de serem superiores a todos os demais testados. Isto se deve principalmente ao fato de a precisão do segmentador de pele obtido pela utilização do método proposto ser bastante superior à precisão de seus concorrentes, o que compensa a sua baixa taxa de detecção, que é a menor de todos os concorrentes.

4.3 Detecção de Céu Azul em Fotografias

O terceiro estudo de caso em que foi aplicado o método proposto nesta dissertação trata-se do problema da detecção de céu azul em fotografias. Este problema consiste em utilizar como entrada fotografias diversas, preferencialmente ao ar livre, para que sejam localizadas as regiões que retratem o céu azul, ou seja, excetuando as nuvens e em cenas diurnas, onde o coloração azulada resultante do espalhamento da luz solar é visível aos olhos humanos.

A detecção de céu azul é utilizada como passo intermediário para diversas aplicações, dentre as quais destacam-se a categorização entre imagens ao ar livre e imagens em ambientes fechados [13], determinar a posição do horizonte e permitir a rotação automática de fotografias capturadas com a câmera na posição vertical [13] e para viabilizar tratamento especializado a regiões de céu em transmissões de visuais – televisão e filmes, dentre outros

[18]. Além disto, como no caso de todos os tipos de materiais, permite o rotulamento automático das imagens, o que é especialmente útil em sistemas de recuperação por conteúdo [13, 63].

Zafarifar e de With [64,63] apresentaram um método de detecção de regiões de céu azul em imagens baseado em modelos probabilísticos compostos a partir da informação de cor e de textura, além da análise da posição vertical dos *pixels* de cada imagem. Foi utilizado o espaço de cores *YUV*, uma vez que a crominância dos *pixels* de céu tendem a variar linearmente no plano *UV* com respeito à sua coordenada vertical, desde o azul saturado até tons avermelhados (passando pelo cinza).

Por se tratar de uma aplicação concebida para tratamento de imagens de televisão, foi elaborado usando exclusivamente o espaço de cores *YUV*, evitando a sobrecarga da transformação de espaços de cor.

O resultado da técnica apresentada não é a binarização da imagem em regiões pertencentes e não pertencentes a céu, mas um mapa contendo as probabilidades de cada *pixel* pertencer à região de céu. Contudo, avaliação da técnica foi realizada de forma subjetiva, o que impediu que fosse apresentada uma comparação objetiva com os resultados desta dissertação.

Gallagher *et al.* [13] utilizaram uma complexa técnica de ajuste de modelos polinomiais de gradientes de cor para melhorar a técnica neural (usando redes MLP) apresentada por Luo e Etz [34]. Desta forma, o resultado obtido pela técnica neural – um conjunto de regiões e suas probabilidades de serem céu – era usado como ponto de partida para a seleção da maior região conectada de céu e para a geração de um modelo polinomial de gradiente vertical de cor a partir das informações obtidas pela região selecionada. As demais regiões eram comparadas com esta região maior, e as que tivessem ajuste com o modelo polinomial gerado maior que um determinado limiar l também seriam consideradas céu.

O resultado desse algoritmo, apesar de ser uma binarização das imagens de entrada, não realiza contagem de área detectada da maneira que é proposta por esta dissertação. Isto faz com que a detecção de regiões pequenas e grandes tenham o mesmo peso no cômputo das métricas e também inviabiliza uma comparação com o resultado obtido pela utilização do método aqui proposto. Cabe também a observação de que o algoritmo apresentado tem a capacidade de exclusivamente adicionar regiões de céu aos resultados de Luo e Etz [34], não

permitindo a contestação de regiões como falsas positivas.

4.3.1 Conjunto de Imagens de Treinamento

Para utilizar o método de obtenção de segmentadores apresentado nesta dissertação, foi utilizada a base de imagens de treinamento gentilmente fornecida por Zafarifar, que trata-se de um subconjunto público da base utilizada para treinamento dos trabalhos de Zafarifar e de With [64,63]. Esta base passou por um tratamento automatizado que permitiu a adequação do seu rotulamento manual às diretrizes estabelecidas na Seção 3.2.1. A Figura 4.8 é um exemplo de imagem de treinamento com sua respectiva segmentação manual.

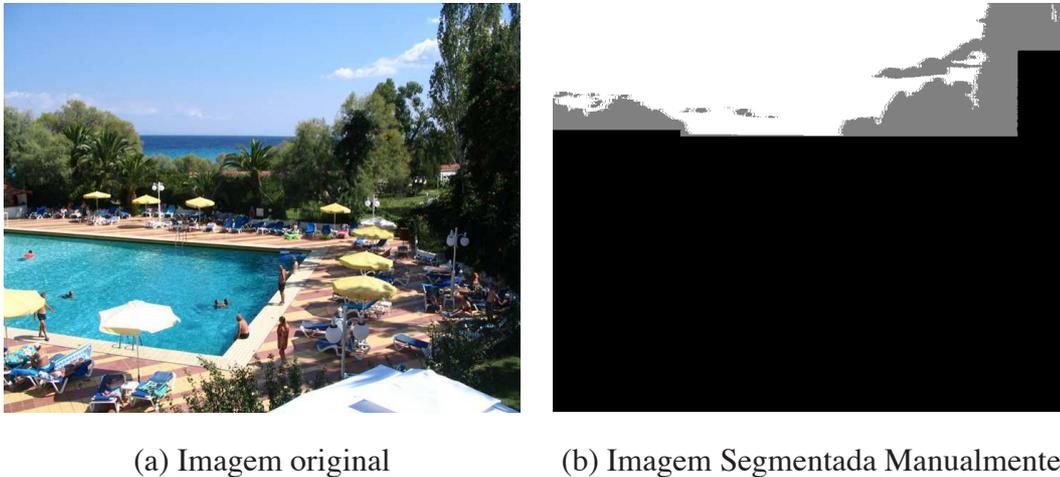


Figura 4.8: Exemplo de imagem do conjunto de treinamento para detecção de céu azul, com sua respectiva segmentação manual.

Ao todo, a base contém 57 fotografias e quadros de transmissões de televisão de conteúdo variado. As imagens apresentam resoluções variadas e muitas delas contêm regiões que podem ser facilmente confundidas com céu, como reflexos do céu em água, piscinas e diversos objetos de coloração azulada.

Destas imagens, 29 foram destinadas a compor o conjunto de treinamento a ser utilizado como insumo para obtenção automática do segmentador de céu, e as 28 restantes utilizadas como conjunto de testes, para ser viabilizada o cálculo da medida-F do segmentador obtido com o método proposto e dos demais segmentadores usados para comparação.

4.3.2 Obtenção Automática do Segmentador

Obtenção do Segmentador Neural Baseado em Informação de Cor

As Tabelas 4.16 a 4.19 exibem os resultados obtidos para cada valor de parâmetro durante a determinação dos melhores parâmetros para a rede neural que segmenta as imagens baseada apenas em informação de cor.

Espaço de Cor	D_r	$\sigma(D_r)$	P_r	$\sigma(P_r)$	F_m	$\sigma(F_m)$
HLS	96,13%	0,0023	85,43%	0,0088	90,46%	0,0051
HSV	96,17%	0,0017	86,11%	0,0012	90,86%	0,0010
$L^*a^*b^*$	97,09%	0,012	87,31%	0,0011	91,94%	0,0011
$L^*u^*v^*$	96,86%	0,0035	87,29%	0,0008	91,83%	0,0019
RGB	95,83%	0,0031	87,00%	0,0023	91,20%	0,0022
XYZ	96,27%	0,0025	87,12%	0,0006	91,46%	0,0014
YCbCr	96,30%	0,0041	87,54%	0,0042	91,71%	0,0040

Tabela 4.16: Determinação do melhor espaço de cor para detecção de céu.

Profundidade de Cor	D_r	$\sigma(D_r)$	P_r	$\sigma(P_r)$	F_m	$\sigma(F_m)$
1 bit por canal (2 níveis)	98,57%	0,0012	73,45%	0,0012	84,18%	0,0012
2 bits por canal (4 níveis)	93,55%	0,0485	73,60%	0,0014	82,34%	0,0194
3 bits por canal (8 níveis)	98,55%	0,0050	76,59%	0,0000	86,15%	0,0021
4 bits por canal (16 níveis)	94,34%	0,0007	87,38%	0,0193	90,72%	0,0110
5 bits por canal (32 níveis)	96,07%	0,0232	86,36%	0,0085	90,95%	0,0139
6 bits por canal (64 níveis)	96,62%	0,0023	87,13%	0,0027	91,63%	0,0023
7 bits por canal (128 níveis)	97,69%	0,0047	86,98%	0,0037	92,03%	0,0017
8 bits por canal (256 níveis)	97,09%	0,0012	87,31%	0,0011	91,94%	0,0011

Tabela 4.17: Determinação da melhor profundidade de cor para detecção de céu ($L^*a^*b^*$).

O espaço de cores $L^*a^*b^*$ forneceu a maior medida-F em relação aos demais espaços de cores avaliados, e foi observado um comportamento crescente da medida-F até 7 bits por canal de cor, seguido de uma queda nesta métrica.

Tamanho da Janela	D_r	$\sigma(D_r)$	P_r	$\sigma(P_r)$	F_m	$\sigma(F_m)$
1 pixels (1 × 1)	97,69%	0,0047	86,98%	0,0037	92,03%	0,0017
9 pixels (3 × 3)	97,01%	0,0023	87,36%	0,0000	91,93%	0,0011
25 pixels (5 × 5)	97,12%	0,0019	87,36%	0,0001	91,98%	0,0008
49 pixels (7 × 7)	98,61%	0,0025	86,90%	0,0012	92,39%	0,0016
81 pixels (9 × 9)	97,22%	0,0032	87,68%	0,0024	92,20%	0,0024

Tabela 4.18: Determinação do melhor tamanho da janela de *pixels* para detecção de céu ($L*a*b^*$, 7 bits por canal).

O tamanho da janela de *pixels* não se mostrou determinante para a separabilidade das regiões P e N , uma vez que não houve grandes variações da medida-F com a mudança deste parâmetro.

A quantidade de neurônios na camada escondida, por sua vez, apresentou comportamento crescente seguido de rápida estabilização logo a partir de 3 neurônios na camada escondida, o que sugere que aumentar este parâmetro também não trará melhora à medida-F.

Após estes experimentos, foi conduzida a investigação acerca da influência do posicionamento dos *pixels* na separabilidade das regiões de céu azul das imagens do conjunto de treinamento.

Intuitivamente, o céu tende a se localizar na parte superior das fotografias, e isto pode ser comprovado observando o histograma apresentado na Figura 2.3 (página 14), em que a concentração de ocorrências de céu se dá nos menores valores da coordenada y dos *pixels*.

A média dos valores no histograma é $\mu^c = 1161,2$ e o desvio padrão $\sigma^{c_v} = 1042,46$, o que leva a um coeficiente de variação $c_v^{c_v} = 0,8977$. Desta maneira, a coordenada vertical é utilizada na segmentação de céu, por ter suficiente variabilidade ($c_v^{c_v} > 0,5$) de acordo com o método proposto nesta dissertação.

No que diz respeito à coordenada horizontal, o céu tende a se espalhar de maneira praticamente uniforme ao longo da extensão da fotografia. A Figura 4.9 corrobora esta intuição.

A média dos valores no histograma é $\mu^c = 1161,2$ e o desvio padrão $\sigma^{c_h} = 65,11$, o que leva a um coeficiente de variação $c_v^{c_h} = 0,0561$. Como intuitivamente esperado, a coordenada horizontal não é utilizada na segmentação de céu.

O treinamento final foi realizado utilizando o espaço de cores $L*a*b^*$, com 7 bits por

Neurônios	D_r	$\sigma(D_r)$	P_r	$\sigma(P_r)$	F_m	$\sigma(F_m)$
1	95,87%	0,0178	85,05%	0,0308	90,09%	0,0280
2	97,59%	0,0145	86,84%	0,0108	91,89%	0,0047
3	97,73%	0,0098	87,13%	0,0046	92,12%	0,0018
4	97,76%	0,0082	87,25%	0,0057	92,20%	0,0009
5	97,30%	0,0036	87,43%	0,0010	92,10%	0,0012
6	97,43%	0,0060	87,51%	0,0026	92,20%	0,0015
7	98,61%	0,0044	86,90%	0,0025	92,39%	0,0009
8	97,29%	0,0027	87,50%	0,0024	92,13%	0,0002
9	97,58%	0,0017	87,32%	0,0019	92,16%	0,0003
10	97,24%	0,0018	87,52%	0,0018	92,12%	0,0002
11	96,81%	0,0009	87,55%	0,0005	91,94%	0,0001
12	97,29%	0,0021	87,58%	0,0024	92,18%	0,0004
13	96,83%	0,0013	87,57%	0,0020	91,97%	0,0005
14	97,78%	0,0058	87,27%	0,0019	92,23%	0,0015

Tabela 4.19: Determinação da melhor quantidade de neurônios na camada oculta para detecção de céu ($L^*a^*b^*$, 7 bits por canal, janela de 7×7 pixels).

canal de cor, usando uma janela de 7×7 pixels, 7 neurônios na camada escondida da rede neural, e desta vez utilizando a coordenada vertical dos pixels.

Com este treinamento, obteve-se um segmentador neural baseado em informação de cor com medida-F de 78,42% ($D_r = 64,39\%$ e $P = 99,07\%$), medida com respeito à base de imagens de treinamento.

Obtenção do Segmentador Baseado em Informação de Textura e Cor

A execução do algoritmo genético descrito na Seção 3.3 resultou na determinação dos valores para os parâmetros do segmentador baseado em informação de textura e cor que constam na Tabela 4.20.

Após a introdução da informação de textura como insumo para segmentação, ocorreu a esperada redução no valor da medida-F do segmentador para $F^{gct} = 78,11\%$ ($D_r^{gct} = 64,32\%$ e $P^{gct} = 99,43\%$), medida com respeito à base de treinamento. Para detecção de céu azul, no

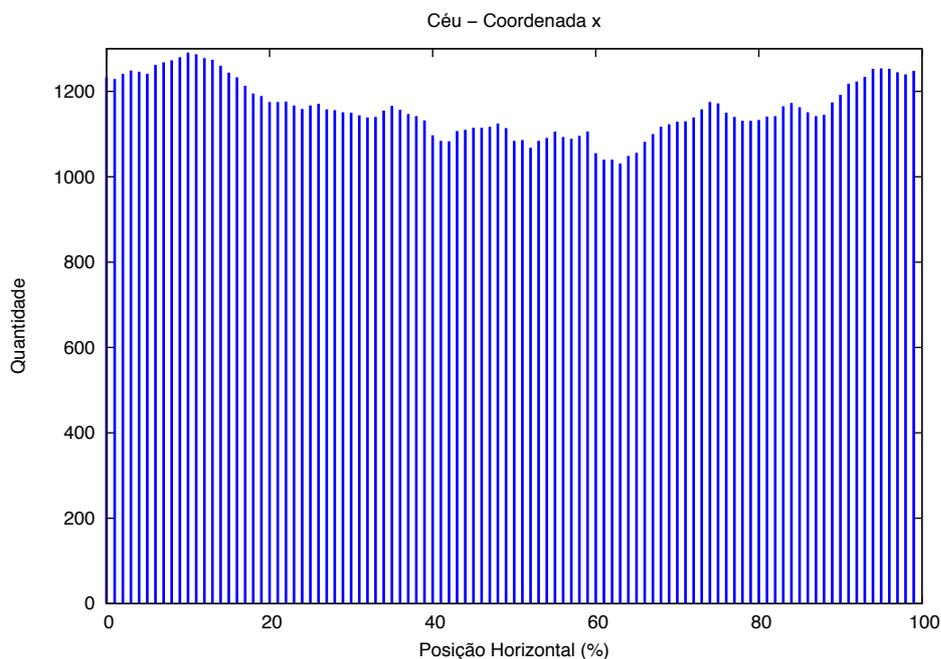


Figura 4.9: Histograma dos pixels de céu na base de imagens de treinamento, com respeito à distância relativa à margem esquerda das imagens.

Parâmetro	Descrição	Valor
M	Número de escalas dos filtros de Gabor	8
N	Número de orientações dos filtros de Gabor	2
l	Limiar inferior da binarização	0
u	Limiar superior da binarização	19

Tabela 4.20: Determinação dos parâmetros do segmentador baseado em informação de textura e cor para detecção de céu.

Parâmetro	Descrição	Valor
a	Limiar de distância de cor da região já marcada	237,941
b	Limiar de distância de cor da borda da região marcada	2,65362

Tabela 4.21: Determinação dos parâmetros para expansão de regiões na detecção de céu.

entanto, esta redução não foi tão significativa quanto a observada para os demais materiais.

Expansão de Regiões para Melhoria da Medida-F

O último passo da obtenção do segmentador de céu foi a execução do algoritmo genético descrito na Seção 3.4, que resultou na determinação dos valores para os parâmetros para a expansão de regiões com o algoritmo *floodfill* adaptado que constam na Tabela 4.21.

Após a expansão das regiões com o algoritmo *floodfill* adaptado, ocorreu a esperada recuperação do valor da medida-F do segmentador para $F^{gct} = 96,18\%$ ($D_r^{gct} = 97,49\%$ e $P^{gct} = 94,90\%$).

Para céu azul, a recuperação do valor da medida-F com a aplicação do algoritmo *floodfill* adaptado foi extremamente importante, e possibilitou o aumento da medida-F em 17,76%. Isto fez com que o segmentador para detecção de céu azul fosse o obtido na última fase do treinamento, levando em consideração informação de cor e textura.

4.3.3 Teste do Segmentador de Céu Azul

O segmentador de céu azul obtido na aplicação do método proposto nesta dissertação foi executado nas 28 imagens que compunham o conjunto de testes para este material.

O segmentador obtido foi capaz de detectar 94,52% da área correspondente a céu azul, com precisão de 91,21%. Isto resulta em um medida-F com o valor de 92,84%. A Figura 4.10 apresenta exemplos de fotografias processadas pelo segmentador obtido.

Também foi testado o segmentador da fase 1 do método proposto, que foi capaz de detectar 62,26% da área correspondente a grama, com precisão de 98,00%, resultando em um medida-F com o valor de 76,14%, verificando também no conjunto de testes que o segmentador da fase 3 obtém melhores resultados.

Para permitir uma comparação com abordagens existentes, também foram testados um

classificador *bayesiano* gerado através da plataforma de mineração de dados Weka. Não foram testados mais segmentadores, uma vez que os trabalhos revisados usavam abordagens de análise de resultados subjetivas, incompatíveis com a proposta desta dissertação. A Tabela 4.22 exibe os resultados obtidos entre os segmentadores comparados.

Segmentador	Tx. Detecção	Precisão	Medida-F
Automatizado	94,52%	91,21%	92,84%
<i>Naïve Bayes</i>	93,47%	89,60%	91,49%

Tabela 4.22: Resultados obtidos para detecção de céu azul

O segmentador de céu azul foi o único entre os três estudos de caso apresentados nesta dissertação que se utilizou das três fases do método proposto. Nele, os benefícios do algoritmo *floodfill* adaptado puderam ser aproveitados de maneira bastante satisfatória juntamente com o posicionamento vertical dos *pixels*.

Percebeu-se que, no caso do material específico, os resultados do segmentador ainda na primeira fase eram em geral restritos à parte superior das imagens, uma vez que a rede neural foi treinada levando em consideração a coordenada vertical dos *pixels*.

Isto já resolvia uma parte do problema: o reflexo de regiões de céu em volumes de água, como lagos, piscinas, mar, entre outros, era desconsiderado da segmentação. O algoritmo *floodfill* complementou a segmentação ao expandir a região de céu e aumentar a taxa de detecção do segmentador. Observar que, com respeito ao conjunto de treinamento, a taxa de detecção subiu de 64,39% para 97,49% com a aplicação do algoritmo *floodfill* adaptado.

A Figura 4.11 exemplifica uma das imagens do conjunto de treinamento em que o resultado parcial e o final se comportam exatamente da maneira como descrita.

4.4 Considerações Finais

Observando os resultados obtidos (sumarizados na Tabela 4.23) com a aplicação do método proposto para obtenção dos segmentadores de grama, pele e céu, percebe-se que é possível automatizar a obtenção e o ajuste de bons segmentadores para detecção de materiais.

Todos os segmentadores obtidos com o método automatizado alcançaram medida-F melhor que os seus concorrentes, quando testados na mesma base de imagens, o que sugere

Material	Tx. Detecção	Precisão	Medida-F
Gramma	96,68%	97,55%	97,11%
Pele	52,82%	49,31%	51,01%
Céu	94,52%	91,21%	92,84%

Tabela 4.23: Resumo dos Resultados

que os resultados obtidos são compatíveis com técnicas específicas de cada material, o que constitui um dos princípios para uma solução ao problema proposto (Seção 1.1).

No que diz respeito aos parâmetros avaliados, observou-se que o espaço de cor é a informação mais importante a ser ajustada para os segmentadores na primeira fase, quando ainda se baseiam exclusivamente em informação de cor. A profundidade de cor também tem importância, uma vez que, nos três materiais que compuseram o estudo de caso, se demonstrou crescente até atingir um valor máximo. O tamanho da janela de *pixels* para a borragem Gaussiana não se mostrou tão importante quanto os demais parâmetros de cor.

Para os três materiais, observou-se também um comportamento bastante parecido no que diz respeito à quantidade de neurônios na camada escondida da rede neural: rápida elevação para baixas quantidades de neurônios com estabilização logo em seguida. Isto sugere que, uma vez que a quantidade máxima de parâmetros a serem analisadas é 5 (3 canais de cor e, no máximo, 2 coordenadas de *pixels*), não são necessários muitos neurônios para que a rede neural alcance bons resultados.

No que diz respeito à análise de textura: percebe-se que o mecanismo de redução da medida-F para aumento da precisão com posterior aplicação do algoritmo *floodfill* adaptado só tem eficácia em condições bastante peculiares, como o caso do céu azul: muita homogeneidade nas cores das regiões *P* e fronteiras entre regiões muito bem definidas. Nos demais casos o algoritmo *floodfill* não foi capaz de aumentar o valor da medida-F em comparação com o segmentador que se baseia apenas em informação de cor.

Por fim, cabe ressaltar que, apesar de ter sido possível medir o tempo necessário para treinamento dos segmentadores usando o método proposto e dos segmentadores Bayesianos (Tabela 4.24), uma avaliação de desempenho não pode ser efetuada com rigor, uma vez que os métodos comparados (i) já foram implementados levando em consideração os resultados de treinamento publicados pelos autores ou (ii) foram treinados em computadores diferentes

do utilizado para executar o método proposto, no caso dos classificadores Bayesianos.

Segmentador	Gramma	Pele	Céu
Automatizado	3710s	2413s	48145s
Bayesiano	6,65s	6,90s	104,73s

Tabela 4.24: Tempos médios de treinamentos dos segmentadores

Além disto, os segmentadores Bayesianos treinados levaram em consideração a melhor combinação de parâmetros encontrada durante o treinamento dos segmentadores obtidos pelo método, o que tira o rigor de uma comparação direta. Entretanto, consta como trabalho futuro desta dissertação realizar uma avaliação mais rigorosa do custo computacional em se utilizar o método genérico de ajuste de segmentadores apresentado neste estudo.

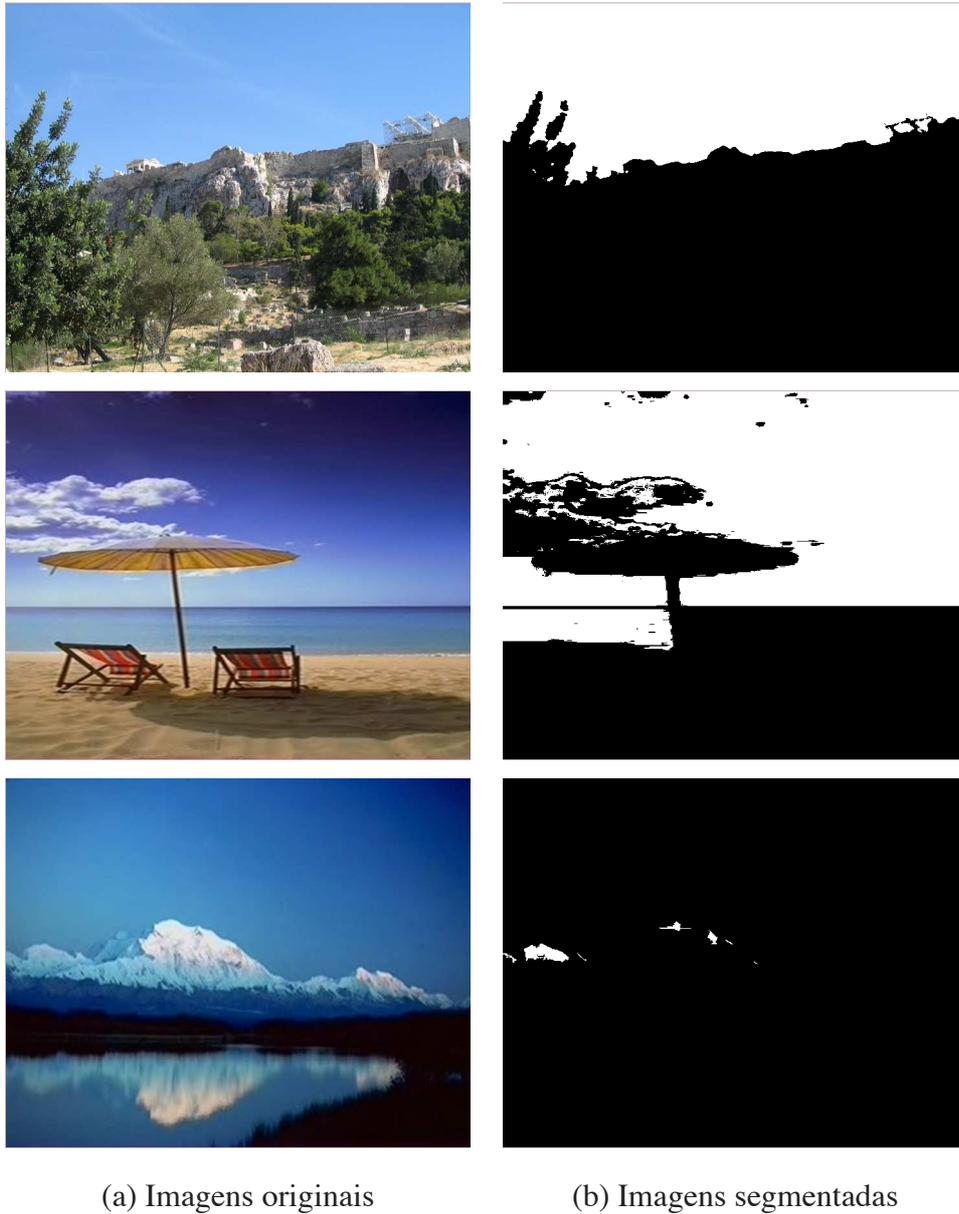


Figura 4.10: Resultados da aplicação segmentador obtido utilizando o método proposto ao problema da detecção de céu (céu em branco e demais regiões em preto).

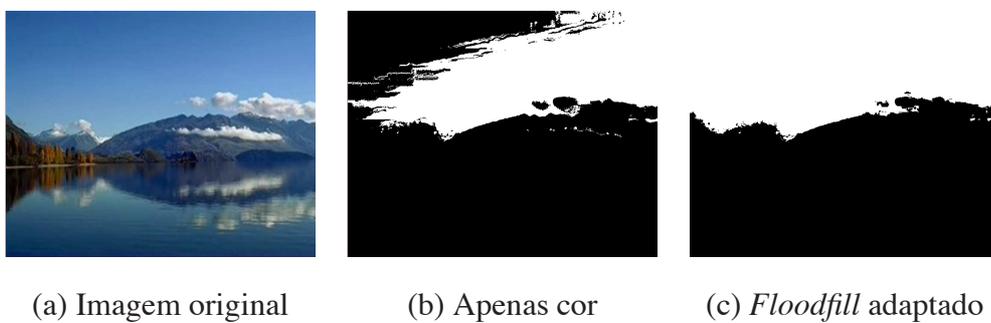


Figura 4.11: Evolução dos resultados da segmentação antes e depois da aplicação do *floodfill* adaptado

Capítulo 5

Conclusão

Nesta dissertação, foi proposto um método para automatização da obtenção de segmentadores de imagens para detecção de materiais, baseados em informação de cor e textura.

Através do método proposto, principal objetivo e contribuição desta dissertação, a responsabilidade pela execução das tarefas de analisar as melhores combinações de parâmetros e de treinar e ajustar tais segmentadores passou a não pertencer a quem deseja apenas usufruir o segmentador como meio para conseguir um determinado objetivo.

O Capítulo 3 descreve passo a passo todas as operações necessárias para que a automatização seja alcançada.

Pode-se afirmar que o objetivo central da dissertação foi atingido, já que: (i) o método proposto é genérico, servindo para gerar segmentadores para diferentes materiais, a depender exclusivamente da definição de um conjunto de treinamento por parte do usuário, contendo regiões rotuladas de onde aparece o material que se deseja detectar; e (ii) os resultados obtidos pela implementação do método proposto são compatíveis com os resultados obtidos com métodos de detecção desenvolvidos especificamente para cada material.

Outra contribuição do trabalho foi a busca na literatura técnica pelos aspectos das imagens que podem ser utilizados para ajudar na detecção de materiais. Reunir estes aspectos em um único segmentador é tarefa importante, pois permite que a detecção seja baseada em um maior número de fatores – representação das cores, profundidade de cores, tamanho da área analisada, posicionamento dos *pixels*, dentre outros –, tornando a fundamentação dos resultados da detecção ainda mais robusta.

Também figuram como contribuições desta dissertação a geração de:

1. Um segmentador de grama em quadros de vídeos desportivos, que pode ser utilizado para as aplicações de sumarização de vídeos de partidas disputadas em campos gramados;
2. Um segmentador de pele humana em fotografias que pode ser utilizado para aplicações como reconhecimento facial, detecção de pornografia e processamento de imagens em dispositivos fotográficos;
3. Um segmentador de céu azul em fotografias, cuja aplicabilidade é encontrada em categorizadores de imagens “ao ar livre” e “ambientes fechados”, e na determinação da orientação de fotografias.

Cabe ressaltar que o segmentador de pele humana desenvolvido nesta dissertação foi utilizado na dissertação de mestrado “Detecção de Olhos em Imagens com Faces Humanas” [29], que trata-se da proposição de uma nova abordagem para detecção de olhos humanos em fotografias, com diversas aplicações no âmbito da fotografia.

Neste estudo, realizado na Universidade Federal de Campina Grande (UFCG), o papel do segmentador de pele foi fornecer critérios para eliminação de falso candidatos a olhos.

Estes segmentadores estão disponíveis na página do Laboratório de Visão Computacional da UFCG [35]. Todo o material pode ser obtido e utilizado livremente em projetos de pesquisa, desde que não tenham fins comerciais.

Além disto, a realização da dissertação resultou também na implementação do gerador de segmentadores de materiais usado para conduzir os experimentos dos estudos de caso apresentados no Capítulo 4. Com a utilização deste gerador de segmentadores, a tarefa da automatização está ao alcance de qualquer pessoa que necessite de um segmentador de materiais.

Todos os códigos fontes, conjuntos de imagens de treinamento, *scripts* de execução e documentos necessários para a reprodução dos resultados apresentados neste trabalho também se encontram na página do Laboratório de Visão Computacional da UFCG. Também são disponibilizados todos os resultados obtidos pelos segmentadores dos estudos de caso desta dissertação, permitindo uma análise particular para cada imagem.

Por fim, também pode ser considerada uma contribuição do trabalho o enfoque dado ao problema de detecção de materiais como problemas de recuperação da informação o que

pode ser evidenciado pelas métricas utilizadas nesta dissertação (medida-F, taxa de detecção e precisão).

Ter o enfoque de recuperação de informação é importante, principalmente para materiais que figuram em pequenas regiões da imagem. Por exemplo, um segmentador de um material que aparece em 5% dos *pixels* das imagens, que rejeita todos os *pixels* alcança taxa de erro de apenas 5%, mas uma vez que sua taxa de detecção é 0, terá medida-F nula e portanto não poderá ser considerado um bom segmentador.

Dentre os pontos positivos da proposta apresentada neste trabalho, destaca-se o fato de ela não exigir que o usuário tenha conhecimento prévio de técnicas de segmentação ou de detecção de materiais para que consiga obter um segmentador para qualquer material.

Além disto, o emprego de redes neurais artificiais – um método de execução rápida e pouco consumo de memória – faz com que seja possível embutir os segmentadores gerados usando o método proposto em dispositivos de baixa capacidade de memória como PDAs, Máquinas Fotográficas, dentre outros [42].

Entretanto, há a desvantagem de o método proposto exigir uma grande quantidade de tempo para ser capaz de gerar um segmentador para um material. Isto se deve à grande quantidade de pontos de treinamento para cada rede neural treinada, e ao fato de que cada combinação de parâmetros exige um treinamento diferente.

A estratégia de serializar a determinação dos valores de cada parâmetro surgiu deste fenômeno, e trouxe consigo o risco de não estar sendo coberto todo o espaço de buscas de combinações de valores de parâmetros.

Trabalhos Futuros

Apesar de todos os objetivos do trabalho apresentado nesta dissertação terem sido alcançados, ainda há uma série de propostas de trabalhos futuros que agregarão muito valor ao método aqui apresentado.

O primeiro destes trabalhos seria realizar análises matemática e estatística mais formais, *Principal Component Analysis* [60] por exemplo, para quantificar a contribuição que cada aspecto analisado provê ao segmentador final. Isto servirá para que (i) pesos diferenciados sejam atribuídos a cada aspecto no momento da amostragem do conjunto de treinamento e

(ii) sejam descartados aspectos que tenham pouca contribuição para a separabilidade entre as regiões, tornando mais rápida a detecção dos segmentadores.

Esta análise matemática permitirá também uma prova da generalidade do método, uma vez que a realização dos estudos de caso é apenas um forte indício de que ele funciona para vários materiais, mas não necessariamente comprova que valerá para todos.

Em seguida, algumas modificações no modo de treinamento podem ser testadas para que o seu efeito nos resultados dos segmentadores obtidos seja analisado. A primeira modificação é utilizar redes neurais com apenas um neurônio de saída, com a aplicação de um limiar de aceitação também aprendido através de varredura. Isto trará maior simplicidade ao treinamento, e permitirá a aplicação de análise de curva ROC (*Receiver Operating Characteristic*).

Também futuramente, técnicas de otimização global serão incorporadas no protótipo apresentado neste trabalho, com o objetivo de facilitar a descoberta da arquitetura ideal da rede neural artificial a ser utilizada para cada um dos materiais para os quais se deseja obter segmentadores.

Assim, a exploração da quantidade de neurônios da camada escondida será restrita a menos valores, além de serem ampliadas as possibilidades de outros aspectos arquiteturais das redes neurais: número de camadas escondidas, conectividade entre neurônios, dentre outros.

Isto também permitirá que seja testada uma maior quantidade de combinações de parâmetros, em vez de serializá-los e assumir que são independentes, além de exigir um menor tempo de treinamento, uma vez que as técnicas de otimização global desconsideram arranjos de parâmetros com menor probabilidade de sucesso. Desta forma, a serialização da determinação dos valores dos parâmetros passará a ser desnecessária, e deixará de existir a desvantagem da fixação da ordem em que os parâmetros são ajustados (não há garantia de que a ordem adotada é a melhor para qualquer material).

Por fim, serão introduzidos outros mecanismos generalizar ainda mais o método de obtenção de segmentadores: inclusão de mais espaços de cores e da possibilidade de prover interfaces para implementação de espaços definidos pelo usuário; tornar genérico o motor de classificação, permitindo que sejam utilizadas técnicas diferentes das redes neurais artificiais e a inclusão técnicas de pré e pós processamento nas imagens para ampliar as possibilidades de utilização do protótipo disponibilizado.

Referências Bibliográficas

- [1] A. Albiol, L. Torres, and E. J. Delp. Optimum color spaces for skin detection. In *Proceedings of the 2001 International Conference on Image Processing*, volume 1, pages 122–124 vol.1, 2001.
- [2] AT&T Laboratories. Permm: Image analysis. World Wide Web Project at http://www.cl.cam.ac.uk/research/dtg/attarchive/permm/image_analysis.html, 2007.
- [3] S. C. Avila and S. R. Reillo. Two Different Approaches for Iris Recognition Using Gabor Filters and Multiscale Zero-crossing Representation. *Pattern Recognition*, 38(2):231–240, 2005.
- [4] I. A. Basheer and M. Hajmeer. Artificial neural networks: Fundamentals, computing, design, and application. *Journal of Microbiological Methods*, 43:3–31, 2000.
- [5] J. Brand and J. S. Mason. A comparative assessment of three approaches to pixel-level human skin-detection. In *Proceedings of the 15th International Conference on Pattern Recognition*, volume 1, pages 1056–1059 vol.1, 2000.
- [6] F. H. B. Cardoso, F. M. Cardoso, G. P. Santos Jr, M. Fortuny, and A. F. Santos. Estudo comparativo de linguagens de programação aplicadas a simuladores de processos químicos. In *Anais do XV Congresso de Iniciação Científica da Universidade Federal de Sergipe*, 2005.
- [7] C. Darwin. *On the Origin of Species: By Means of Natural Selection*. Dover Publications, June 2006.
- [8] J. G. Daugman. Uncertainty Relation for Resolution in Space, Spatial Frequency, and

- Orientation Optimized by Two-dimensional Visual Cortical Filters. *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, 2(7):1160–1169, 1985.
- [9] K. Deb and S. Agrawal. Understanding interactions among genetic algorithm parameters. In *Foundations of Genetic Algorithms 5*, volume 5, pages 265–286, 1998.
- [10] A. Diplaros, T. Gevers, and N. Vlassis. Skin detection using the em algorithm with spatial constraints. In *Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 3071–3075 vol.4, 2004.
- [11] P. T. Firmino Jr and H. M. Gomes. Detecção de lances no campo de ataque em uma partida de futebol. In *Anais do Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens 2007*, 2007.
- [12] P. Fuangkhon and T. Tanprasert. Nipple detection for obscene pictures. In *Proceedings of WSEAS International Conference on Signal, Speech and Image Processing*, pages 315–320. WSEAS, August 2005.
- [13] A. C. Gallagher, J. Luo, and W. Hao. Improved blue sky detection using polynomial model fit. In *Proceedings of the 2004 International Conference on Image Processing - ICIP '04*, volume 4, pages 2367–2370 Vol. 4, 2004.
- [14] C. Garcia and G. Tziritas. Face detection using quantized skin color regions merging and wavelet packet analysis. *IEEE Transactions on Multimedia*, 1(3):264–277, 1999.
- [15] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, 1987.
- [16] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 2nd edition, July 1998.
- [17] R. Hecht-Nielsen. *Neurocomputing*. Addison-Wesley, September 1990.
- [18] S. Herman and E. Bellers. Locally-adaptive processing of television images based on real-time image segmentation. In *Digest of Technical Papers of the 2002 International Conference on Consumer Electronics - ICCE 2002*, pages 66–67, 2002.
- [19] N. Herodotou, K. N. Plataniotis, and A. N. Venetsanopoulos. *Image Processing Techniques for Multimedia Processing*. CRC Press, 2000.

- [20] S. Ho and K. Lee. Efficient image segmentation using a generic and non-parametric approach. In *Proceedings of the Fourth International Conference/Exhibition on High Performance Computing in the Asia-Pacific Region*, volume 2, pages 777–781 vol.2, 2000.
- [21] J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The MIT Press, April 1992.
- [22] R. Hsu, M. Abdel-Mottaleb, and A. K. Jain. Face detection in color images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):696–706, 2002.
- [23] Intel Corporation. The opencv library. World Wide Web Project Page at <http://opencv.willowgarage.com>, 2008.
- [24] S. Jayaram, S. Schmugge, M. C. Shin, and L. V. Tsap. Effect of colorspace transformation, the illuminance component, and color modeling on skin detection. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - CVPR 2004*, volume 2, pages II–813–II–818 Vol.2, 2004.
- [25] Z. Jiang, M. Yao, and W. Jiang. Skin detection using color, texture and space information. In *Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery - FKSD 2007*, volume 3, pages 366–370, 2007.
- [26] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, 2002.
- [27] C. C. Klimasauskas. Neural networks: an engineering perspective. *Communications Magazine, IEEE*, 30(9):50–53 vol.2, 1992.
- [28] S. Lawrence, I. Burns, A. Back, A. Tsoi, and C. Giles. Neural network classification and prior class probabilities. In *Neural Networks: Tricks of the Trade*, volume 1524/1998 of *Lecture Notes in Computer Science*, pages 545–559. Springer Berlin / Heidelberg, 1998.
- [29] B. B. Leite. Detecção de olhos em imagens com faces humanas. Master’s thesis, Universidade Federal de Campina Grande - UFCG, 2008.

- [30] V. Levesque. Texture Segmentation Using Gabor Filters. *Center For Intelligent*, 2000.
- [31] E. O. Lima. Sumarização de vídeos de futebol com baixo custo computacional utilizando ritmo visual: Descritores de cor e movimento de câmera. Master's thesis, UNIFOR, 2006.
- [32] S. Livens, P. Scheunders, G. van de Wouwer, and D. van Dyck. Wavelets for texture analysis, an overview. In *Proceedings of the Sixth International Conference on Image Processing and Its Applications*, volume 2, pages 581–585 vol.2, 1997.
- [33] C. Loesch and S. T. Sari. *Redes Neurais Artificiais: Fundamentos e Modelos*. Blumenau: Editora da FURB, 1996.
- [34] J. Luo and S. P. Etz. A physical model-based approach to detecting sky in photographic images. *Image Processing, IEEE Transactions on*, 11(3):201–212, 2002.
- [35] LVC-UFCG. Laboratório de visão computacional. Página Institucional na World Wide Web em <http://lvc.ufcg.edu.br>, 2009.
- [36] M. Mitchell. *An Introduction to Genetic Algorithms (Complex Adaptive Systems)*. The MIT Press, February 1998.
- [37] R. Montufar-Chaveznaza, F. H. Gallardo, and S. P. Hernandez. Face detection by polling. In *Proceedings of the 2005 IEEE International Workshop on Intelligent Signal Processing*, pages 292–297, 2005.
- [38] J. R. Movellan. Tutorial on Gabor Filters. *Tutorial paper* <http://mplab.ucsd.edu/tutorials/pdfs/gabor.pdf>, 2008.
- [39] M. U. Munir and M. Y. Javed. Fingerprint Matching Using Gabor Filters. In *National Conference on Emerging Technologies*, pages 147–151, 2004.
- [40] S. Nissen. Implementation of a Fast Artificial Neural Network Library (fann). *Report, Department of Computer Science University of Copenhagen (DIKU)*, 31, 2003.
- [41] T. Ojala and M. Pietikäinen. Unsupervised texture segmentation using feature distributions. In *Proceedings of the 9th International Conference on Image Analysis and Processing - ICIAP '97*, pages 311–318, London, UK, 1997. Springer-Verlag.

- [42] S. L. Phung, A. Bouzerdoum, and D. Chai. Skin segmentation using color pixel classification: Analysis and comparison. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(1):148–154, January 2005.
- [43] L. Prechelt. Proben1 - a set of neural network benchmark problems and benchmarking rules, 1994.
- [44] N. A. Rahman, Kit C. Wei, and J. See. Rgb-h-cbcr skin colour model for human face detection. In *Proceedings of The MMU International Symposium on Information & Communications Technologies - M2USIC 2006*, 2006.
- [45] G. J. E. Rawlins, editor. *Foundations of Genetic Algorithms*. Morgan Kaufmann, 1st edition, July 1991.
- [46] J. Ruiz-Del-Solar and R. Verschae. Skin detection using neighborhood information. In *Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 463–468, 2004.
- [47] Y. Shi, M. Xiao, and J. Yang. Pixel-based skin color detection considering overlap region. In *Proceedings of the 2007 International Symposium on Intelligent Signal Processing and Communication Systems - ISPACS 2007*, pages 256–259, 2007.
- [48] M. C. Shin, K. I. Chang, and L. V. Tsap. Does colorspace transformation make any difference on skin detection? In *Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision - WACV 2002*, pages 275–279, 2002.
- [49] D. Sinclair. Voronoi seeded colour image segmentation. Technical report, AT&T Laboratories, 1999.
- [50] N. E. Sondak and V. K. Sondak. Neural networks and artificial intelligence. *SIGCSE Bull.*, 21(1):241–245, 1989.
- [51] M. H. A. Tafti. Neural networks: a new dimension in expert systems applications. In *Proceedings of the 1990 ACM SIGBDP Conference on Trends and Directions in Expert Systems - SIGBDP '90*, pages 423–433, New York, NY, USA, 1990. ACM.

-
- [52] L. Tarassenko. *Guide to Neural Computing Applications*. John Wiley's Sons, Inc., New York, NY, USA, 1998.
- [53] C. Town and D. Sinclair. Ontological query language for content based image retrieval. In *Proceedings of the 2001 IEEE Workshop on Content-Based Access of Image and Video Libraries - CBAIVL 2001*, pages 75–80, 2001.
- [54] C. P. Town and D. Sinclair. Content based image retrieval using semantic visual categories. Technical report, AT&T Laboratories Cambridge, 2000.
- [55] M. Tuceryan and A. K. Jain. Texture analysis. *Handbook of pattern recognition & computer vision*, pages 235–276, 1993.
- [56] V. Vezhnevets, V. Sazonov, and A. Andreeva. A survey on pixel-based skin color detection techniques. In *Proceedings of the GraphiCon 2003*, pages 85–92, 2003.
- [57] M. Wall. GALib: A C++ Library of Genetic Algorithm Components. *Mechanical Engineering Department, Massachusetts Institute of Technology*, 1996.
- [58] J. Wang, C. Xu, E. Chng, X. Yu, and Q. Tian. Event detection based on non-broadcast sports video. In *Proceedings of the 2004 International Conference on Image Processing - ICIP '04*, volume 3, pages 1637–1640 Vol. 3, 2004.
- [59] J. Z. Wang, G. Wiederhold, O. Firschein, and X. S. Wei. Content-based image indexing and searching using daubechies' wavelets. *International Journal on Digital Libraries*, 1(4):311–328, 1998.
- [60] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, June 2005.
- [61] Y. Yoon and L. L. Peterson. Artificial neural networks: an emerging new technique. In *Proceedings of the 1990 ACM SIGBDP Conference on Trends and Directions in Expert Systems - SIGBDP '90*, pages 417–422, New York, NY, USA, 1990. ACM.
- [62] B. Zafarifar and P. H. N. de With. Adaptive modeling of sky for video processing and coding applications. In *27th Symposium on Information Theory in the Benelux*, page 58, 2006.

-
- [63] B. Zafarifar and P. H. N. de With. Blue sky detection for picture quality enhancement. In *Advanced Concepts for Intelligent Vision Systems*, pages 522–532. Springer Berlin / Heidelberg, 2006.
- [64] B. Zafarifar and P. H. N. de With. Blue sky detection for content-based television picture quality enhancement. In *Digest of Technical Papers of the 2007 International Conference on Consumer Electronics - ICCE 2007*, pages 1–2, 2007.
- [65] B. D. Zarit, B. J. Super, and F. K. H. Quek. Comparison of five color models in skin pixel classification. In *Proceedings of the 1999 International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 58–63, 1999.

Apêndice A

Conceitos Fundamentais

Neste capítulo, são discutidos os aspectos mais relevantes sobre os quais se baseia o método proposto nesta dissertação. Inicialmente, são apresentadas definições para os termos utilizados no decorrer do texto, a fim de uniformizar o tratamento e extinguir qualquer ambiguidade.

Em seguida, são apresentadas as métricas utilizadas para medição de qualidade de um segmentador para detecção de materiais e para comparação de segmentadores.

Por fim, o capítulo contém uma discussão sucinta das técnicas utilizadas pelo método, seus parâmetros ajustáveis e a influência desses parâmetros no processo de segmentação.

A.1 Definições Iniciais

As imagens I submetidas a um segmentador ζ têm duas regiões sem intersecção: (i) a região $P(I)$, que abrange todos os *pixels* p correspondentes ao material que se deseja detectar, e (ii) a região $N(I)$ que corresponde aos demais *pixels* da imagem. Na Figura A.1, estão apresentadas uma imagem original (A.1(a)) a suas respectivas regiões P e N (A.1(b)), caso o material a detectar seja grama. Os símbolos P e N foram inspirados nos conceitos de amostras *Positivas* e *Negativas* em um problema de classificação de padrões.

Os resultados das segmentações $R = \zeta(I)$, por sua vez, são divididos em quatro regiões: (i) a região T_P , correspondente às aceitações verdadeiras; (ii) a região T_N , correspondente às rejeições verdadeiras; (iii) a região F_P , correspondente às falsas aceitações e (iv) a região F_N , correspondente às falsas rejeições. As definições formais destas regiões são apresentadas nas



(a) Imagem original

(b) Regiões P (Branco) e N (Preto)

Figura A.1: Exemplo de imagem e sua respectiva segmentação manual para composição da base de treinamento para o segmentador de grama.

Equações A.1 a A.4.

$$T_P = \{R(x,y) \mid I(x,y) \in P(I) \text{ e } R(x,y) \in P(R)\} \quad (\text{A.1})$$

$$T_N = \{R(x,y) \mid I(x,y) \in N(I) \text{ e } R(x,y) \in N(R)\} \quad (\text{A.2})$$

$$F_P = \{R(x,y) \mid I(x,y) \in N(I) \text{ e } R(x,y) \in P(R)\} \quad (\text{A.3})$$

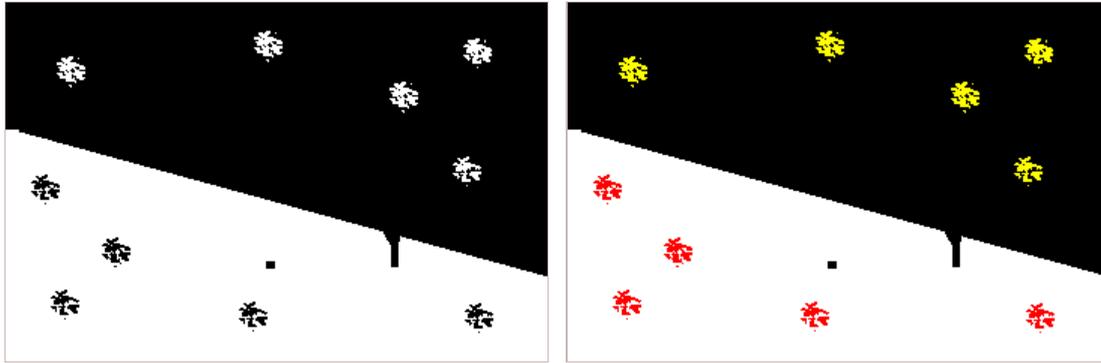
$$F_N = \{R(x,y) \mid I(x,y) \in P(I) \text{ e } R(x,y) \in N(R)\} \quad (\text{A.4})$$

em que $P(R)$ e $N(R)$ denotam respectivamente as regiões de R consideradas como pertencentes ao material em que se deseja detectar e as demais regiões da imagem.

A Figura A.2(a) é o resultado de uma segmentação hipotética da imagem original da Figura A.1(a). Para ilustrar o significado das regiões dos resultados, elas foram marcadas em cores diferentes (A.2(b)).

A.2 Métricas de Avaliação dos Segmentadores

Métricas de avaliação precisam ser definidas para possibilitar a medição quantitativa da qualidade dos segmentadores obtidos utilizando-se o método proposto neste trabalho. Uma vez que os segmentadores para detecção de materiais são classificadores dos *pixels* da imagem



(a) Resultado de uma segmentação hipotética (b) Regiões T_P (Branco), T_N (Preto), F_P (Amarelo) e F_N (Vermelho)

Figura A.2: Exemplo de imagem e sua respectiva segmentação manual para composição da base de treinamento para o segmentador de grama.

nas classes (i) pertencente ou (ii) não pertencente à área do material, as métricas utilizadas para avaliar os segmentadores são as mesmas que se empregam para classificadores. Nesta seção, são definidas as métricas de avaliação de classificadores [60] e apresentadas as justificativas para a utilização de cada uma delas.

A.2.1 Taxa de Detecção (*Recall*)

A Taxa de Detecção de um segmentador é a medida que indica a relação entre a área da imagem corretamente classificada como sendo pertencente ao material que se deseja detectar e a área total deste material. Formalmente, a medida da taxa de detecção D_r^B de um segmentador ζ com respeito a uma base B de imagens de teste é definida pela Equação A.5.

$$D_r^B(\zeta) = \frac{\sum_{I \in B} S(T_P(I))}{\sum_{I \in B} S(P(I))} \quad (\text{A.5})$$

em que S denota a função área de uma região. No âmbito desta região, a unidade de área é o *pixel*.

Intuitivamente, pode-se dizer que um segmentador com alta taxa de detecção consegue detectar a maioria dos dos *pixels* da área em questão. Para ilustrar o significado desta medida, a Figura A.3 exhibe os resultados da aplicação de dois segmentadores hipotéticos de grama, um com alta taxa de detecção (85,05%) e outro com baixa taxa de detecção (18,08%), à

imagem da Figura A.1(a).

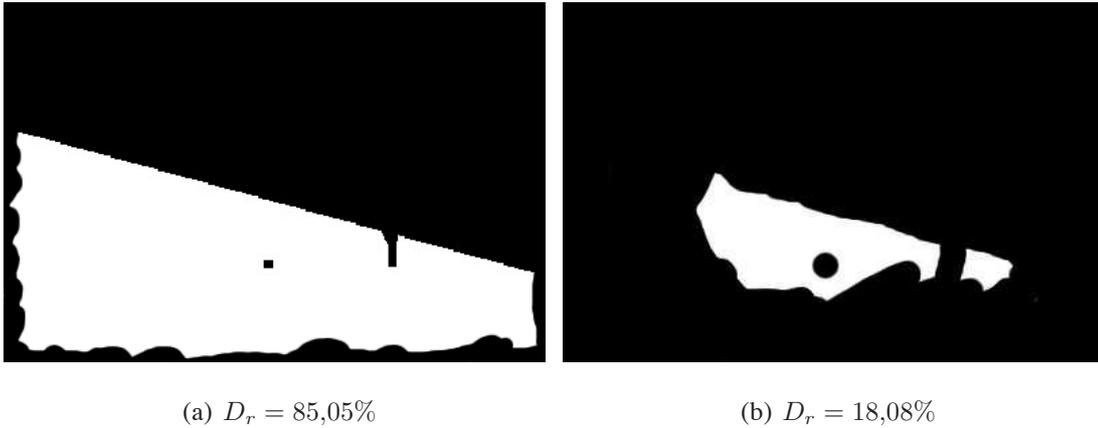


Figura A.3: Exemplo da utilização de segmentadores hipotéticos na imagem da Figura A.1(a).

Utilizar esta medida isoladamente não é suficiente para medir a qualidade de um segmentador, uma vez que não considera a região $N(I)$ das imagens. Observando a Equação A.5, percebe-se que um segmentador que simplesmente aceita todas as amostras apresenta 100% de taxa de detecção, mas erra em todas as regiões N das imagens.

A.2.2 Precisão

A Precisão de um segmentador é a medida que indica a relação entre a área da imagem corretamente classificada como sendo pertencente ao material que se deseja detectar e a área total aceita pelo segmentador. Formalmente, a medida da Precisão P_r^B de um segmentador ζ com respeito a uma base B de imagens de teste é definida pela Equação A.6.

$$P_r^B(\zeta) = \frac{\sum_{I \in B} S(T_P(I))}{\sum_{I \in B} S(T_P(I)) + \sum_{I \in B} S(F_P(I))} \quad (\text{A.6})$$

Intuitivamente, pode-se dizer que um segmentador de alta precisão acerta a maioria dos *pixels* que considerou fazerem parte do material que se deseja detectar.

Esta medida soluciona o problema apresentado pela utilização da taxa de detecção, pois passa a considerar a região $N(I)$ (verificar que a Equação A.3 leva em consideração a região $N(I)$). No entanto, utilizar esta medida isoladamente ainda não é suficiente para medir a qualidade de um segmentador, uma vez que não considera a região $N(R)$ dos resultados.

Observando a Equação A.6, percebe-se que um segmentador que simplesmente aceita poucas amostras, acertando em todas elas, apresenta 100% de precisão, mas deixa de detectar regiões pertencentes ao material em questão.

A.2.3 Medida-F (*F-measure*)

A solução para o problema da insuficiência das medidas apresentadas é combiná-las em uma única medida, passando assim a considerar todos os aspectos dos segmentadores, e viabilizando uma comparação mais adequada.

Para tanto, adota-se neste trabalho a Medida-F (média harmônica entre a taxa de detecção e a precisão) ao se comparar dois segmentadores para um mesmo material em uma mesma base de imagens de teste. Formalmente a Medida-F (F_m) é calculada empregando-se a Equação A.7.

$$F_m^B(\zeta) = (1 + \beta^2) \frac{P_r^B(\zeta) \cdot D_r^B(\zeta)}{\beta^2 P_r^B(\zeta) + D_r^B(\zeta)} \quad (\text{A.7})$$

Nesta equação, o parâmetro β é a relação entre os pesos da taxa de detecção e da precisão, de modo que $\beta = 3$, por exemplo, significa que o peso da taxa de detecção é 3 vezes maior que o da precisão.

Nesta dissertação, todos os valores de medida-F foram calculados usando o parâmetro $\beta = 1$, exceto quando um valor diferente foi explicitamente informado.

Desta forma, afirmar que um segmentador ζ_1 é melhor que um outro segmentador ζ_2 é equivalente à seguinte relação $F_m(\zeta_1) > F_m(\zeta_2)$.

A.3 Técnicas Utilizadas no Método Proposto

Durante a elaboração do método apresentado nesta dissertação, foi necessário resolver três principais problemas: separar os *pixels* correspondentes a um material dos demais *pixels* das imagens, otimizar os parâmetros do extrator de informação de textura e associar cada *pixel* da imagem a um valor de característica de textura.

Nesta seção, são apresentadas as fundamentações teóricas das técnicas utilizadas para resolver cada um dos problemas apresentados: as Redes Neurais Artificiais, os Algoritmos Genéticos e os Filtros de Gabor, respectivamente.

A.3.1 Redes Neurais Artificiais

A corrente conexionista da Inteligência Artificial defende a ideia de que não é possível reduzir certos problemas a passos lógicos (algoritmos) para sua resolução. Em vez disto, é necessária uma simulação dos processos relacionados à inteligência natural a fim de se atingir uma verdadeira inteligência artificial. Com base neste pensamento de simular processos de raciocínio e na tentativa de reproduzir o funcionamento do cérebro humano, foi proposto o modelo computacional das Redes Neurais Artificiais [16].

Elas introduziram na Ciência da Computação um novo paradigma de resolução de problemas que antes, com a computação convencional, não eram resolvidos por serem de alta complexidade e proibitivos, considerando os recursos computacionais existentes [61].

Estas diferenças entre a computação neural e a convencional se devem ao fato de que os modelos computacionais convencionais são incapazes de tratar dados contínuos, incompletos, ruidosos ou nebulosos, de modo que problemas que envolvam estes tipos de dados não serão tratados de maneira tão eficiente por um computador que segue o modelo de Von Neumann. A Tabela A.1 exhibe as principais diferenças entre a computação convencional e a neural.

Característica	Computação Convencional	Computação Neural
Sinal	Digital	Analógico
Dados	Completos e Estruturados.	Incompletos e Nebulosos.
Programas	Dependentes de instruções precisas.	Formulam métodos de processamento de maneira independente.
Processamento	Matemático e Lógico.	Sinais ponderados.
Resultados	Previsíveis	Em geral imprevisíveis.
Memória	Separada do processamento.	Em geral, memória e processamento estão combinados.

Tabela A.1: Comparativo entre as computações convencional e neural. Fonte: Adaptado de [51].

Ainda nos anos 40, mais precisamente em 1943 e 1949, surgiram os primeiros trabalhos no sentido de simular o comportamento de um cérebro biológico no que diz respeito ao raciocínio e aprendizado. Tais trabalhos, o primeiro de McCulloch e Pitts e o segundo de

Hebb, tratam respectivamente da primeira proposta de uma rede neural artificial, o Psychon, e da organização e comportamento das redes neurais naturais [16].

Em 1958, dois trabalhos foram publicados e tiveram grande importância: o primeiro deles, de von Neuman, relacionava o funcionamento de um computador digital com o funcionamento do cérebro biológico. O outro, de Rosemblat, trazia o design de um *hardware*, denominado Mark I Perceptron, considerado o primeiro da computação neural [4].

Logo em seguida, nas décadas de 60 e 70, houve uma considerável pausa nas pesquisas desta área da inteligência artificial, que somente foi retomada nos anos 80 com as publicações de Hopfield, Kohonen e Rumelhart, que introduziram as redes de Hopfield, os Mapas Auto-Organizáveis e o algoritmo *backpropagation*.

Desde então, vários grupos de pesquisa têm se dedicado a estudar e desenvolver as redes neurais artificiais, fazendo inúmeros progressos tanto na teoria quanto na aplicação desta técnica de computação nos mais variados campos da ciência.

A unidade fundamental do sistema nervoso biológico são as células denominadas neurônios. Tais células, como as demais células eucariontes do corpo, podem ser divididas em membrana celular, citoplasma e núcleo [16]. Porém, devido ao fato de estas células serem especializadas em processar impulsos nervosos, a sua forma é diferente das demais células dos organismos.

Funcionalmente, um neurônio biológico pode ser dividido em 3 partes, conforme apresentado na Figura A.4: os dendritos, o corpo celular e o axônio. Os dendritos – em geral mais numerosos e mais curtos que os axônios [16] – são as terminações dos neurônios responsáveis por receberem os impulsos nervosos enviados por seus vizinhos. O corpo celular, por sua vez, é o responsável pelo processamento deste impulsos, computando um novo impulso de saída para o neurônio. Este impulso computado é enviado ao axônio – em geral mais longo e apresentando-se uma única vez para cada neurônio – para ser transmitido aos neurônios vizinhos através de suas terminações.

O processamento de informação em um neurônio biológico se inicia nos dendritos, que são responsáveis por coletar os impulsos nervosos – na forma de sinais elétricos – dos neurônios vizinhos e enviá-los ao corpo celular. O corpo celular emite então um novo novo sinal elétrico cuja intensidade depende tanto de um limiar do próprio neurônio receptor, quanto da intensidade dos sinais recebidos de seus vizinhos.

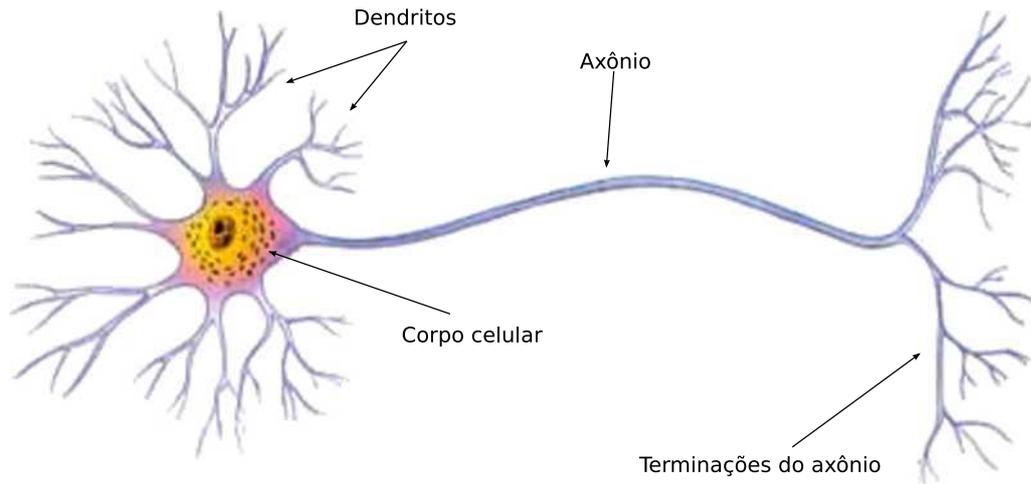


Figura A.4: Neurônio Biológico - Fonte: Adaptado de [33]

Os impulsos chegam a um neurônio através de conexões denominadas sinapses, e podem atuar tanto no sentido de contribuir quanto no de inibir a geração de um novo sinal, de modo que nem sempre um novo impulso é gerado e diz-se que o neurônio não foi ativado [16].

Caso o neurônio seja ativado, o novo sinal elétrico, que corresponde ao impulso nervoso de saída do neurônio em questão, passa através do axônio e suas terminações até chegar às sinapses: espaço entre dois neurônios, pelo qual o impulso nervoso é passado do neurônio gerador para o receptor. Como os neurônios não se tocam nas sinapses, a transmissão é feita pela emissão de substâncias chamadas neurotransmissores, cuja quantidade emitida depende, dentre uma série de outros fatores, principalmente da eficiência da sinapse e da intensidade do impulso nervoso do neurônio gerador

Nos neurônios seguintes, o processamento é feito de maneira idêntica, fazendo com o que os impulsos nervosos sejam transmitidos através da rede de neurônios do cérebro – a rede neural biológica – e transformados em cada um dos neurônios, que podem ser considerados os nós de processamento da rede.

Como foi mencionado anteriormente, as sinapses por si só podem ser mais fortes ou mais fracas que as demais, fazendo com que umas produzam mais neurotransmissores que as outras, e com que o sinal transmitido por elas seja mais intensificado do que o mesmo sinal quando transmitido pelas demais.

Desta maneira, os sinais que chegam através das sinapses mais fortes têm predominância na composição do sinal que chega ao corpo celular de um neurônio, ou seja, a sua contribui-

ção para o processamento do impulso nervoso é maior do que a das demais sinapses da rede neural biológica.

A fim de simular o comportamento de um conjunto de neurônios humanos, foi proposto o modelo matemático das redes neurais artificiais. Em problemas de classificação, são utilizadas principalmente as redes neurais do tipo *Perceptron* de Múltiplas Camadas (MLP). Uma rede neural artificial deste tipo é composta de várias entidades semelhantes, denominadas neurônios artificiais. Um neurônio artificial pode ser abstraído conforme a Figura A.5 [16].

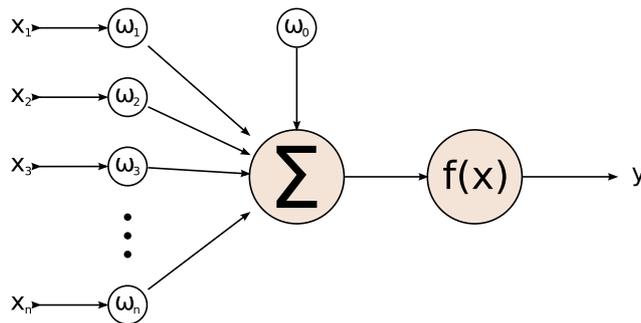


Figura A.5: Neurônio Artificial - Fonte: Adaptado de [16]

Como pode ser observado, um neurônio artificial é uma entidade que recebe do ambiente externo, por meio de suas entradas, impulsos (x_i) que são combinados através do somatório dos produtos de todos os impulsos com o peso de sua respectiva entrada (ω_i) de modo a se determinar um sinal de entrada ($s = \sum_{i=0}^n x_i \omega_i$) para o neurônio. Há uma entrada x_0 a mais em todos os neurônios, de valor fixo 1, e cujo peso ω_0 serve para ajuste da saída independente das entradas. Este peso tem o nome especial de *bias* [16].

A entrada s é submetida a uma função f que determina o sinal de saída do neurônio, denominada função de ativação. Diversas funções podem ser utilizadas, dentre as quais se destacam as logarítmico-sigmoidais e as tangentes hiperbólicas (Figura A.6).

Desta forma, podem ser feitas algumas analogias com os neurônios biológicos: as entradas de um neurônio artificial correspondem aos dendritos dos neurônios biológicos, o peso das entradas é análogo à força de cada sinapse, e a função de ativação corresponde à composição e geração do novo sinal por parte do corpo celular. Há ainda a analogia entre o valor do sinal de entrada de um neurônio artificial e a intensidade do impulso nervoso que chega em um biológico.

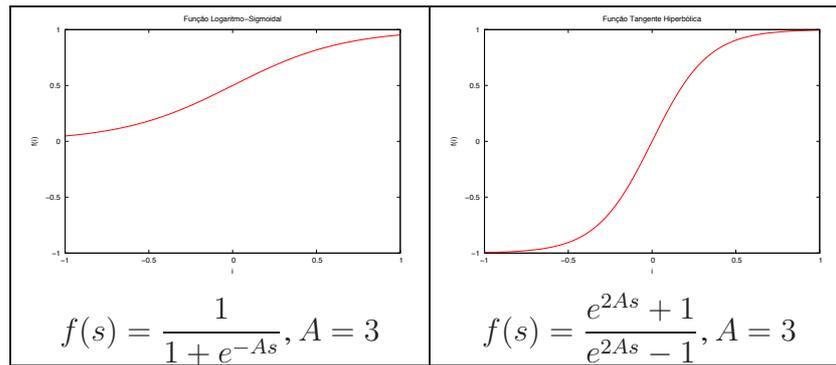


Figura A.6: Algumas das funções de ativações utilizadas nas redes neurais MLP.

A partir do arranjo de neurônios artificiais é possível montar uma rede neural, interligando saídas de uma camada de neurônios às entradas dos neurônios de uma próxima camada. Conforme pode-se observar na Figura A.7, uma rede neural pode ter diversas camadas entre a entrada e a saída (camadas intermediárias) e quantidade variável de neurônios em cada uma das camadas.

O aprendizado, nas redes neurais MLP, segue o paradigma supervisionado: um conjunto de dados é previamente computado de forma manual e é utilizado para ensinar a rede e torná-la capaz de generalizar e inferir regras. A Figura A.8 representa o esquema de funcionamento do aprendizado supervisionado. Para cada passo do treinamento, um vetor de características é apresentado à rede neural artificial, que gera uma resposta para o mesmo. Em contrapartida, o professor (ou supervisor) determina a resposta que se deseja que a rede gere. A partir da comparação entre as respostas desejada e real, é possível gerar um sinal de erro.

De acordo com o sinal de erro gerado, procede-se à correção dos pesos, o que deve ser feito de modo que, quando apresentado à rede novamente o mesmo conjunto de dados de entrada, a diferença entre a resposta desejada e a resposta real seja menor possível, ou seja, mais perto da perfeição.

Segundo Prechelt [43], recomenda-se que os dados sejam divididos em conjunto de treinamento, que é utilizado para ajustar os pesos da rede; de validação, que é utilizado para determinar o erro médio quadrático da rede e evitar que a rede “decore” os dados de treinamento; e conjunto de testes, utilizado para medir a precisão da rede após o seu treinamento.

O algoritmo de aprendizado deve ser repetido até que uma condição de parada seja satisfeita. Estabelecem-se, geralmente, como condição de parada a quantidade de iterações nos dados de treinamento, ou quando se alcança um erro médio quadrático aceitável na compu-

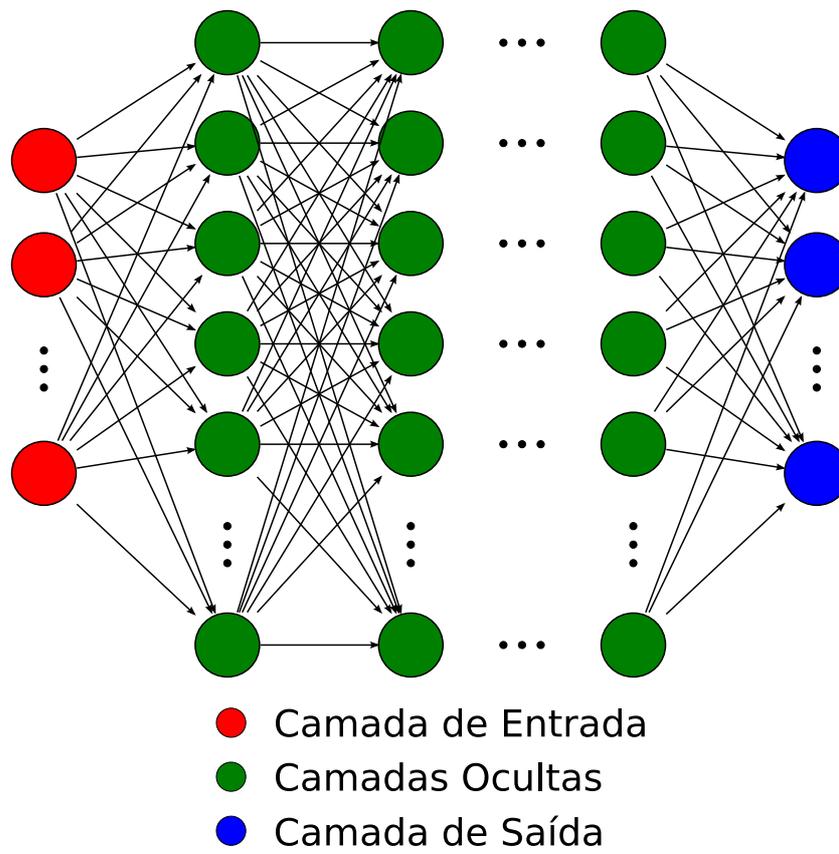


Figura A.7: Rede Neural do tipo *Perceptron* de Múltiplas Camadas.

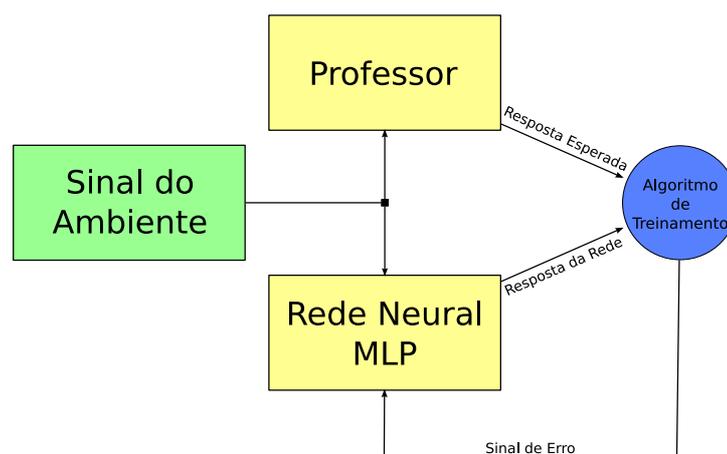


Figura A.8: Aprendizado Supervisionado. Fonte: adaptado de [16]

tação dos dados do conjunto de treinamento.

O mais famoso algoritmo de treinamento de Redes Neurais MLP é o da retropropagação. Ele recebe esse nome porque propaga o ajuste dos pesos das entradas do neurônio de trás para frente pelas camadas da rede neural, ou seja, os erros são corrigidos primeiramente na camada de saída, depois na última camada oculta, seguindo-se pela penúltima, até que se atinja a camada de entrada da rede.

Para cada padrão j do conjunto de dados de treinamento, a resposta da rede é computada da maneira como foi descrita, e os erros dos neurônios da camada de saída são determinado de acordo com a Equação A.8, na qual φ é a função de ativação da rede, $v(j)$ é o resultado do somatório dos produtos entre os pesos e as entradas no neurônio, d_j é a saída desejada para os padrão de treinamento e y_j é a saída obtida pela rede.

$$\delta_j = \varphi'(v(j)) (d_j - y_j) \quad (\text{A.8})$$

Já os erros dos neurônios das camadas ocultas são determinados de acordo com a Equação A.9, na qual k é o índice da camada que se está computando o erro, h_j é a saída gerada pelo neurônio e $w_{k+1,i,n}$ é o peso da i -ésima entrada do n -ésimo neurônio da k -ésima camada.

$$\delta_{k,j} = \varphi'(v(j)) \sum_{n=1}^N \delta_{k+1,i,j} w_{k+1,i,n} \quad (\text{A.9})$$

Depois de calcular os erros, aplica-se a Equação A.10 para ajustar os pesos de um determinado neurônio, na qual i é o índice da entrada cujo peso será corrigido, j é o índice dos dados de treinamento, η é o fator de aprendizado da rede, x_i é o valor da entrada em questão e δ_j é o erro do neurônio para a j -ésima entrada.

$$\Delta\omega_{i,j} = \eta\delta_j x_i \quad (\text{A.10})$$

Devido à não-linearidade das funções de ativação utilizadas nas Redes Neurais MLP, e principalmente à presença de várias camadas de neurônios dispostos de maneira conveniente, elas são utilizadas como solução para classificar padrões não-linearmente separáveis. A principal aplicação das redes MLP é a aproximação de funções, para a qual podem ser mapeadas diversas classes de aplicações, como a previsão de séries temporais e o reconhecimento de padrões.

A aproximação de funções utilizando Redes Neurais Artificiais se dá através do treinamento de uma rede MLP de modo a associar os elementos do domínio da função aos seus valores correspondentes no conjunto imagem da mesma. Quando o valor da função não é conhecido para todos os elementos do domínio, a rede neural MLP é capaz de aproximar ou generalizar essa regra e estimar os mapeamentos desconhecidos.

Estas redes neurais têm utilidade quando os dados são coletados através de experimentos, e necessitam ser extrapolados, ou nos casos em que computar analiticamente estas funções se torna muito custoso. Com frequência estas redes neurais são utilizadas no campo da microbiologia [4], controle de processos e em processamento adaptativo de sinais [27].

Para abordagens mais detalhadas sobre os assuntos referentes à redes neurais artificiais, seu projeto, utilização e aplicações, recomenda-se a leitura das referências [16,50,33]. Para informações acerca da Engenharia de Software envolvida em um projeto de redes neurais, recomenda-se [52].

A.3.2 Algoritmos Genéticos

Uma outra corrente da Inteligência Artificial – a corrente Evolucionista – simula um outro fenômeno biológico observado na natureza para resolução de uma miríade de problemas: a Evolução das Espécies. Alguns problemas de otimização podem ser resolvidos a partir de uma série de soluções candidatas – uma “população” – que sucessivamente vão evoluindo até que se encontre uma solução mais adequada ao problema – o “indivíduo” ótimo.

Estes algoritmos evolucionistas são chamados Algoritmos Genéticos (AG), e esta seção é dedicada a uma breve revisão sobre os principais tópicos relacionados a este tema.

A Teoria da Evolução das Espécies

“Dei a este princípio, em virtude do qual uma variação, por insignificante que seja, se conserva e se perpetua, se for útil, o nome de Seleção Natural, para indicar as relações desta seleção com a que o homem pode operar.” [7]

Com estas palavras, o naturalista britânico Charles Darwin nomeou o principal fenômeno de sua Teoria sobre a Evolução das Espécies. Esta teoria explicava algumas das questões em

aberto àquela época acerca da origem da enorme quantidade de espécies que habitam a Terra e inspirou, depois de décadas, a técnica dos algoritmos genéticos.

Segundo a biologia, todos os seres vivos possuem suas propriedades físicas armazenadas nas suas células em estruturas orgânicas denominadas cromossomos. Cada cromossomo é uma cadeia de Genes que define todas as características de um indivíduo. Quando se reproduzem, os indivíduos misturam seus genes de uma forma que cada um dos descendentes possui metade dos genes de cada um dos pais.

Em uma população com diversos indivíduos, alguns deles possuem características que os tornam mais adaptados ao ambiente em que vivem que outros indivíduos da mesma geração, como força física, facilidade em conseguir alimento, sentidos mais aguçados, entre outros.

Estes indivíduos mais adaptados, segundo a Teoria da Evolução das Espécies, possuem mais chances de se reproduzirem, uma vez que serão mais atrativos aos parceiros para acasalamento, bem como terão maior probabilidade de ter uma vida mais longa, acasalando mais vezes.

Uma vez que os membros das gerações seguintes terão características genéticas herdadas de seus pais, elas preservarão estas características que os tornaram mais adaptados ao ambiente. É desta maneira que os indivíduos mais aptos de cada geração sofrem a Seleção Natural.

Além da Seleção Natural, as mutações aleatórias dos genes – devido a razões externas ao organismo, como radioatividade ou defeitos na propagação dos genes – também são um fator de muita importância no processo evolutivo. Estas mutações podem eventualmente produzir melhoras na adaptabilidade do indivíduo mutante, aumentando as suas probabilidades de sobrevivência e reprodução, aumentando, desta forma, as suas chances de ser naturalmente selecionado para preservar características genéticas.

Problemas de Otimização

Questões como “Que caminho deve seguir um carteiro para minimizar a distância percorrida e ainda assim entregar todas as suas correspondências?” e “Qual é a melhor distribuição de peso da carga de um caminhão para que a degradação de suas peças seja minimizada?” têm uma similaridade: ambas questionam a melhor solução para um dado problema, dadas todas as soluções possíveis para o mesmo. Estes problemas recebem a denominação de Problemas

de Otimização.

Um Problema de Otimização consiste em determinar a melhor solução entre todas as soluções possíveis a um outro problema. Formalmente, um problema P de otimização é uma quadrupla (I, f, m, g) em que:

- I é um conjunto de instâncias do problema a ser otimizado;
- dada uma instância $x \in I$, $f(x)$ é o conjunto de todas as soluções que resolvem a instância x ;
- dada uma instância x e uma solução $y \in f(x)$ a esta instância, $m(x, y)$ é a medida de y , em geral, $m : I, f \rightarrow \mathbb{R}$
- g é a função objetivo, em geral as funções “máximo” ou “mínimo”.

A solução ao problema de otimização é determinar para uma instância x uma solução ótima, isto é, uma solução y tal que $m(x, y) = g \{m(x, y') \mid y' \in f(x)\}$.

Um exemplo deste tipo de problema foi apresentado nesta dissertação: “dado um segmentador com filtros de Gabor que utiliza a técnica proposta por Jiang et al. [25], que combinação de parâmetros produz a maior medida-F (com $\beta = 5$)?”

Uma solução bem sucedida para este tipo de problema depende, basicamente, de três fatores:

- Codificação do Problema: o problema deve estar eficiente e formalmente codificado, sem inconsistências. Para que isto seja possível, é necessário modelar matemática e computacionalmente todos os aspectos do problema, bem como modelar suas soluções;
- Função Objetivo: é a função que se deseja otimizar – maximizar ou minimizar. No exemplo citado, a função objetivo é a medida-F ($\beta = 5$) que, de acordo com o enunciado do problema, deve ser maximizada; e
- Espaço das Soluções: é o conjunto de todas as possíveis soluções para o problema. Este conjunto deve ser modelado adequadamente e reduzido, se possível, para desconsiderar soluções absurdas, de modo a facilitar a busca pela solução ótima.

De modo geral, resolver problemas de otimização é bastante complicado devido às dificuldades na codificação do problema, uma vez que nem todos os aspectos têm uma modelagem matemática trivial. Além disto, existem problemas com alta dimensionalidade e que possuem inúmeras soluções possíveis, o que faz a busca pela solução ótima uma tarefa bastante demorada.

Ciclo de Funcionamento dos Algoritmos Genéticos

Na tentativa de propor uma maneira diferente de resolver problemas de otimização, e simulando o processo natural de “otimização das espécies”, os algoritmos genéticos foram propostos por John Holland na década de 1970 [21]. Estes algoritmos, consideram cada uma das possíveis soluções de um dado problema como indivíduos – cada uma deles possuindo seus próprios genes – que se reproduzem de modo a compor novas gerações de soluções. A probabilidade de reprodução de cada indivíduo é proporcional à sua aptidão, ou seja, a qualidade da solução que representa.

É necessário conhecer os conceitos básicos relacionados aos algoritmos genéticos antes da explanação de seu funcionamento. Os conceitos são:

- **Cromossomo:** computacionalmente, um cromossomo é um elemento do espaço de soluções, ou seja, uma solução válida para o problema de otimização dado. No exemplo do problema da otimização dos filtros de Gabor, os cromossomos são cada uma das combinações de parâmetros possíveis;
- **Gene:** é cada parâmetro que compõe um cromossomo. Pode ser considerado a porção mínima em que uma solução é dividida. Os genes das soluções do problema da otimização dos filtros de Gabor são cada um dos quatro parâmetros a otimizar;
- **Fenótipo:** biologicamente, o fenótipo de um indivíduo é o conjunto de suas características físicas. Computacionalmente, pode ser considerado como o conjunto de características que descrevem a solução do problema. O fenótipo das soluções do problema da otimização dos filtros de Gabor é o resultado produzido pelos filtros nas imagens em que são aplicados;

- População: é o conjunto de indivíduos que está sendo analisado na iteração corrente do algoritmo genético em execução.
- Geração: é o conjunto de indivíduos da mesma “idade”, ou seja, que nasceram no mesmo ciclo de reprodução;
- Aptidão: é a quantificação da adaptabilidade de um indivíduo ao seu *habitat*. Computacionalmente, este conceito está associado a quão distante uma solução está da solução ótima – sua qualidade, em outras palavras. Um bom exemplo de função de aptidão para o problema da otimização dos filtros de Gabor é a medida-F ($\beta = 5$) conforme a Equação A.7: quanto maior a medida-F, melhor é a aptidão do cromossomo.

O Algoritmo A.1 é o pseudocódigo de um algoritmo genético básico, e mostra a sua característica iterativa – assim como a Seleção Natural. Percebe-se que, a cada geração, a população de indivíduos passa pelos mesmos operadores genéticos, obtendo a próxima geração. A população inicial é obtida aleatoriamente, baseando-se no fato de que no início as espécies ainda não passaram por nenhum tipo de evolução.

Algoritmo A.1: Algoritmo genético básico

```

1 populacao ← solucoesAleatorias ();
2 melhorSolucao ← melhorDaPopulacao ( populacao );
3 enquanto ( !ehOtima( melhorSolucao ) ) {
4     reprodutores ← roleta ( populacao );
5     novaGeracao ← reproduz ( reprodutores );
6     mutacao ( novaGeracao );
7     populacao ← populacao ∪ novaGeracao;
8     poda ( populacao );
9     melhorSolucao ← melhorDaPopulacao ( populacao );
10 }
11 retorna melhorSolucao ;

```

Para cada população, a função de aptidão de todos os indivíduos é calculada e normalizada, a fim de se suavizarem discrepâncias entre valores muito altos e muito baixos.

Em seguida a condição de parada pode ser testada. De acordo com a listagem apresentada, o algoritmo para de executar quando a população possui um indivíduo com aptidão suficiente para ser considerado “quase-ótimo” – a tolerância deve ser definida de acordo

com o problema em questão. Outra possível condição de parada é restringir a quantidade de iterações. Pode-se também combinar as duas condições de parada, de maneira ponderada.

Caso a condição de parada não seja satisfeita, o algoritmo prossegue selecionando os reprodutores. Analogamente ao que acontece na natureza, esta seleção é baseada na aptidão normalizada de cada indivíduo da população: os mais aptos são os que têm mais chance de reproduzir.

A maneira mais comum de implementar esta seleção de reprodutores de um algoritmo genético é usando o Método da Roleta. Este método modela uma roleta em que cada indivíduo corresponde a uma “fatia” cuja largura é proporcional à sua aptidão.

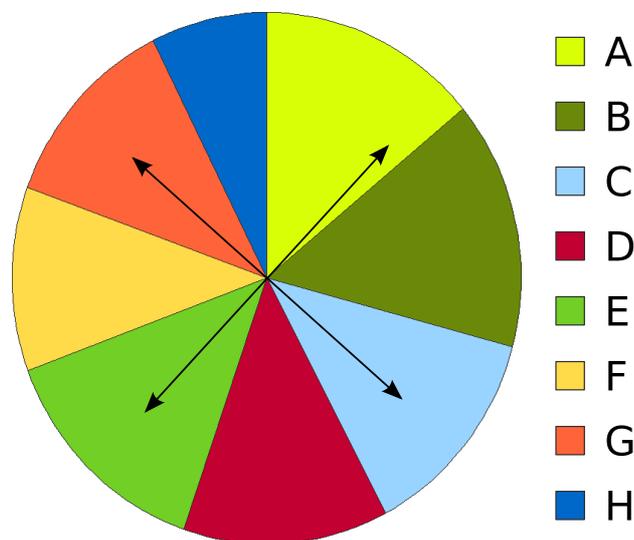


Figura A.9: Representação do Método da Roleta

Como se pode observar na representação da Figura A.9, ponteiros equidistantes são posicionados no centro da roleta – tantos quantos forem os reprodutores a serem selecionados – e postos a rodar livremente. Os indivíduos selecionados são aqueles que tiverem suas “fatias” apontadas.

É fácil notar que a probabilidade de seleção de um indivíduo é proporcional à sua aptidão, e que o mesmo indivíduo pode, se for muito apto em relação aos demais, ser selecionado mais de uma vez para se reproduzir. Esta situação também ocorre na natureza, pois indivíduos podem se reproduzir mais vezes a depender de sua aptidão.

No exemplo da Figura A.9, está representada uma roleta para um algoritmo genético que tem oito candidatos a reprodução (A a H). Depois de girarem os ponteiros, os indivíduos A,

Tabela A.2: Emparelhamento de Reprodutores

Emparelhamento	
Reprodutor	Par
A	E
C	G
E	C
G	A

C, E e G são selecionados para compor o conjunto de reprodutores que comporão a próxima geração.

Os reprodutores vão efetivamente se reproduzir na fase de cruzamento genético. A primeira parte desta fase é o emparelhamento dos reprodutores, para que sejam decididos os pares de cada um dos indivíduos. Isto é feito clonando-se a lista de reprodutores e reordenando aleatoriamente a lista clonada. Após o reordenamento, o n-ésimo indivíduo da lista original é emparelhado ao n-ésimo indivíduo da nova lista. Um exemplo deste emparelhamento está exibido na Tabela A.2.

O processo de cruzamento genético de dois elementos é muito similar com o que ocorre com os cromossomos naturais. O primeiro passo deste processo é determinar o ponto de cruzamento: ponto aleatório na sequência genética que funciona como pivô do cruzamento. As sequências genéticas dos novos indivíduos gerados consiste na parte inicial do primeiro genitor – do início da sequência até o ponto de cruzamento – e a parte final do segundo genitor – do ponto de cruzamento até o final. A Figura A.10 é uma ilustração do operador de cruzamento genético.

Por fim, acontece a simulação do fenômeno da mutação genética. Este fenômeno ocorre quando genes sofrem alterações acidentais, causadas por fatores externos, e que modificam o seu fenótipo. Na natureza, as mutações úteis – as que promovem uma maior aptidão ao indivíduo mutante – têm maior probabilidade de serem perpetuadas. Os algoritmos genéticos modelam este fenômeno ao alterar o valor de um gene aleatoriamente. De modo análogo, os indivíduos podem, eventualmente, se tornar mais aptos, aumentando a sua probabilidade de reprodução e, conseqüentemente, de perpetuação de sua carga genética.

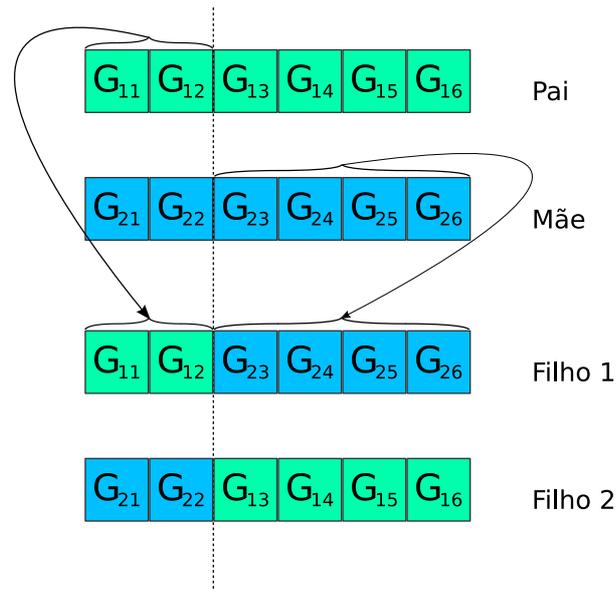


Figura A.10: Cruzamento genético de dois indivíduos

O passo final de cada iteração é a poda da população: os indivíduos que apresentam menor aptidão são removidos da população, simulando assim, a morte natural dos indivíduos menos adaptados ao ambiente.

Parâmetros de um Algoritmo Genético

A rápida convergência de um algoritmo genético e a boa confiabilidade de seus resultados dependem diretamente da escolha dos valores para os seguintes parâmetros:

- **Tamanho da População:** se um algoritmo genético possui uma população pequena, ela não é capaz de cobrir todas as regiões do espaço de soluções, e a confiabilidade dos resultados fica comprometida. Em contrapartida, uma população exageradamente grande torna a execução demanda mais capacidade de processamento, tornando a computação um tanto lenta;
- **Número de Gerações:** para atingir um bom nível evolutivo, ou seja, para que a convergência ocorra para um mínimo ou máximo global da superfície da função de aptidão, um algoritmo genético deve ser executado com o máximo de gerações possível, assim como na natureza, onde uma alteração em um indivíduo requer um grande número de gerações para ser perpetuada;

- Taxa de Cruzamento: O número de cruzamento entre indivíduos de uma geração é bastante importante, uma vez que não pode ser muito baixo, a ponto de tornar o algoritmo lento, nem muito grande, pois isto pode provocar o desaparecimento de estruturas genéticas úteis; e
- Probabilidade de Mutação: mutações também são importantes na execução de um algoritmo genético, pois permitem avaliar regiões do espaço de soluções que ainda não estão contempladas pela população. A estabilidade da população é abalada, e caso a convergência esteja se dirigindo para um mínimo – ou máximo – local, isto será evitado. Por outro lado, muita mutação em uma execução de algoritmo genético pode também provocar a perda de estruturas úteis.

Os valores ótimos destes parâmetros dependem do problema a ser resolvido. Entretanto há um certo consenso de que os reprodutores devem ser entre 60% e 65% da população inicial, bem como a taxa de mutação deve ficar entre 0,1% e 5%. Os demais parâmetros são ainda mais específicos, e devem ser ajustados pelo programador do algoritmo a cada caso.

Considerações Finais

O ajuste de parâmetros de segmentadores é um problema que envolve a busca de uma combinação ótima de valores de diversos parâmetros. Isto faz uma busca exaustiva se tornar muito dependente de poder de processamento.

Esta dependência justifica a utilização de um método de otimização global, como os algoritmos genéticos, para a descoberta dos valores ótimos de parâmetros a serem ajustados na obtenção dos segmentadores para detecção de materiais.

Sugere-se a leitura de Mitchell, 1998 [36] e FOGA, 1991 [45] para maiores detalhes acerca do funcionamento de algoritmos genéticos.

A.3.3 Filtros de Gabor

Para extrair informação de textura dos materiais que se deseja detectar com os segmentadores obtidos utilizando a técnica proposta nesta dissertação, foi adotada a técnica de filtragem utilizando Filtros de Gabor de Duas Dimensões.

Esta técnica – proposta por Daugman, em 1986 [8] – é baseada na aplicação de um filtro de duas dimensões a uma imagem com o intuito de analisar a resposta obtida. A Equação A.11 é a definição de um filtro de Gabor de duas dimensões,

$$g(x,y) = s(x,y)w_r(x,y) \quad (\text{A.11})$$

Nesta Equação, $s(x,y)$ é uma portadora senóide complexa de duas dimensões e $w_r(x,y)$ é uma gaussiana de duas dimensões, chamada também envelope [38]. A Figura A.11 mostra as partes Real (Figura A.11(a)), Imaginária (Figura A.11(b)) e a Magnitude (Figura A.11(c)) de um Filtro de Gabor.

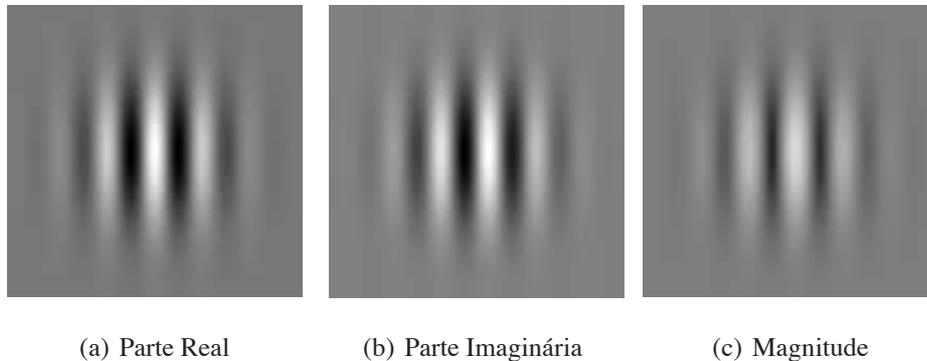


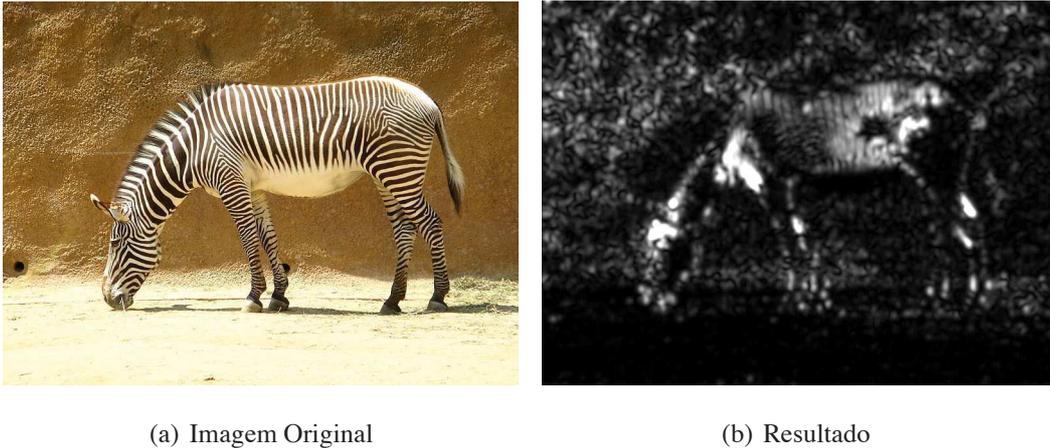
Figura A.11: Partes real e imaginária e magnitude um filtro de Gabor

A resposta da aplicação destes filtros (em especial a Magnitude, por caracterizar a “força” da resposta) pode ser utilizada para detectar as porções da imagem que têm textura similar à do filtro. Conforme se pode observar na Figura A.12 (imagem original de autoria de Chris Valle ¹), são ressaltadas as regiões de listras verticais quando se aplica o filtro da Figura A.11(c) à imagem exibida.

No entanto, em uma mesma imagem existem diversos tipos de textura que podem estar dispostas em diversas orientações, e em diversas frequências. Para se ter uma noção exata da intensidade da textura em todas as regiões da imagem, é necessário combinar o resultado de diversos Filtros de Gabor, alterando os parâmetros dos mesmos.

No âmbito desta dissertação, serão considerados os seguintes parâmetros:

¹Imagem copiada sob autorização da GNU Free Documentation License, obtida na URL <http://commons.wikimedia.org/wiki/File:LAzooZebra.jpg>

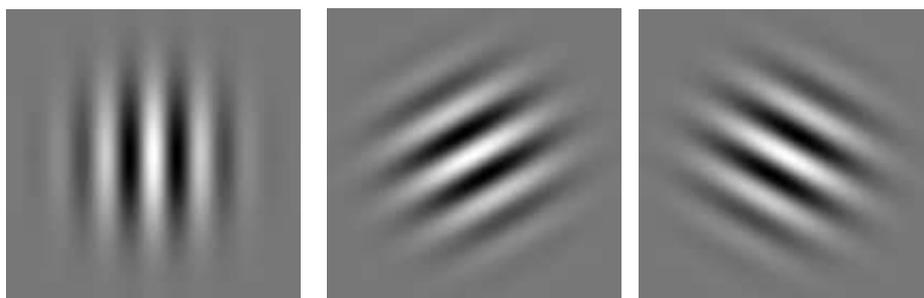


(a) Imagem Original

(b) Resultado

Figura A.12: Exemplo de imagem e seu resultado após a aplicação do filtro de Gabor

- **Orientação:** indica, em radianos, a rotação que será sofrida pelo Filtro de Gabor antes de ser aplicado à imagem. Alterando este parâmetro, é possível capturar texturas em diferentes orientações na imagem, não se restringindo apenas às ocorrências de uma orientação (a exemplo da Figura A.12, que teve no resultado apenas as texturas verticais). A Figura A.13 exibe as partes reais de três Filtros de Gabor em diferentes orientações.
- **Escala:** altera o tamanho do Filtro de Gabor, e por consequência a frequência da senoíde associada. Ajustando este parâmetro, é possível detectar texturas de maior ou menor frequência, conforme a necessidade do filtro. A Figura A.14 exibe as partes reais de três Filtros de Gabor em diferentes escalas.

(a) Orientação: 0 radiano (b) Orientação: $\frac{2\pi}{3}$ radianos (c) Orientação: $\frac{4\pi}{3}$ radianos**Figura A.13:** Diferentes orientações de filtros de Gabor

Aplicações práticas dos Filtros de Gabor, além do método proposto nesta dissertação, incluem detecção de materiais [25], segmentação por textura [30] e biometria, ao se realizar

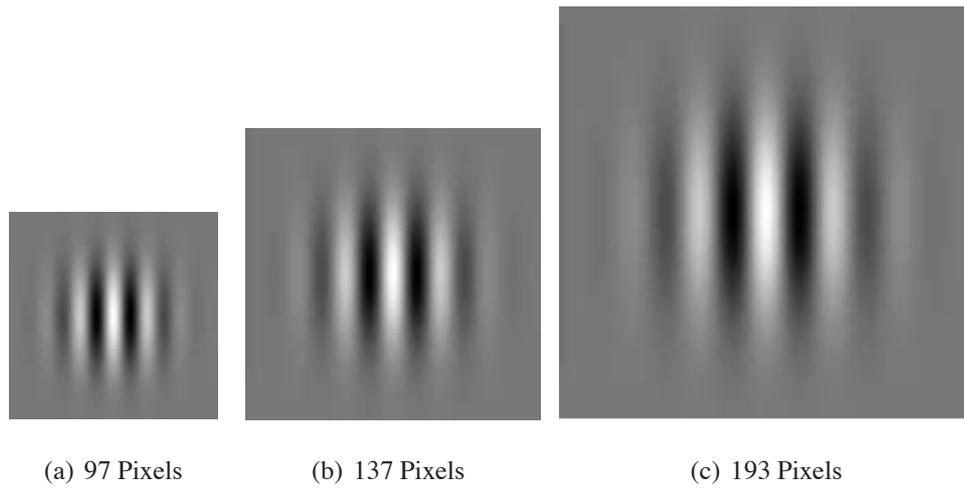


Figura A.14: Diferentes escalas de filtros de Gabor

identificação de impressões digitais [39], e de íris [3], dentre outras.

Apêndice B

Transformações entre Espaços de Cores

Neste apêndice, encontram-se todas as equações utilizadas para converter do espaço de cores RGB para os demais espaços de cores analisados pelo protótipo que implementa o método proposto nesta dissertação. Estas equações são utilizadas pela biblioteca de Visão Computacional OpenCV [23].

B.1 De RGB para CIE XYZ

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0,412453 & 0,357580 & 0,180423 \\ 0,212671 & 0,715160 & 0,072169 \\ 0,019334 & 0,119193 & 0,950227 \end{pmatrix} \times \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (\text{B.1})$$

Os valores de X , Y , e Z se encontram no intervalo inteiro $[0..255]$, com profundidade de 8 *bits* por canal de cor.

B.2 De RGB para YCbCr

$$Y = 0,299R + 0,587G + 0,114B \quad (\text{B.2})$$

$$Cb = (B - Y)0,564 + 128 \quad (\text{B.3})$$

$$Cr = (R - Y)0,713 + 128 \quad (\text{B.4})$$

Os valores de Y , Cb , e Cr se encontram no intervalo inteiro $[0..255]$, com profundidade de 8 *bits* por canal de cor.

B.3 De RGB para HSV

Para transformar do espaço RGB para o HSV, inicialmente todas as componentes – R, G e B – são normalizadas para o intervalo $[0..1]$. Após a normalização, são aplicadas as seguintes equações:

$$V = \max(R, G, B) \quad (\text{B.5})$$

$$S = \begin{cases} \frac{V - \min(R, G, B)}{V}, & \text{se } V \neq 0 \\ 0, & \text{caso contrário} \end{cases} \quad (\text{B.6})$$

$$H' = \begin{cases} \frac{60(G - B)}{S}, & \text{se } V = R \\ \frac{120 + 60(B - R)}{S}, & \text{se } V = G \\ \frac{240 + 60(R - G)}{S}, & \text{se } V = B \end{cases} \quad (\text{B.7})$$

Em seguida, caso o valor de H' seja negativo, ele é somado a 360, de modo que $H = H' + 360$. Nos casos contrários, $H = H'$. Assim, os valores de H estão no intervalo inteiro $[0..360]$ e os valores de S e V estão no intervalo contínuo $[0..1]$.

B.4 De RGB para HSL

Para transformar do espaço RGB para o HSL, inicialmente todas as componentes – R, G e B – são normalizadas para o intervalo $[0..1]$. Após a normalização, são aplicadas as seguintes

equações:

$$V_{max} = \max(R, G, B) \quad (\text{B.8})$$

$$V_{min} = \min(R, G, B) \quad (\text{B.9})$$

$$L = \frac{V_{max} + V_{min}}{2} \quad (\text{B.10})$$

$$S = \begin{cases} 0, & \text{se } V_{min} = V_{max} \\ \frac{V_{max} - V_{min}}{V_{max} + V_{min}} = \frac{V_{max} - V_{min}}{2L}, & \text{se } L \leq \frac{1}{2} \\ \frac{V_{max} - V_{min}}{2 - (V_{max} + V_{min})} = \frac{V_{max} - V_{min}}{2 - 2L}, & \text{se } L > \frac{1}{2} \end{cases} \quad (\text{B.11})$$

$$H' = \begin{cases} \frac{60(G - B)}{S}, & \text{se } V_{max} = R \\ \frac{120 + 60(B - R)}{S}, & \text{se } V_{max} = G \\ \frac{240 + 60(R - G)}{S}, & \text{se } V_{max} = B \end{cases} \quad (\text{B.12})$$

Em seguida, caso o valor de H' seja negativo, ele é somado a 360, de modo que $H = H' + 360$. Nos casos contrários, $H = H'$. Assim, os valores de H estão no intervalo inteiro $[0..360]$ e os valores de S e L estão no intervalo contínuo $[0..1]$.

B.5 De RGB para CIE L*a*b*

Para transformar do espaço RGB para o CIE L*a*b*, são realizadas a transformação de RGB para XYZ e a normalização dos valores XYZ para o intervalo contínuo $[0..1]$ como passos intermediários. Em seguida:

$$L^* = \begin{cases} 116\sqrt[3]{Y} - 16, & \text{se } Y > 0,008856 \\ 903,3Y, & \text{se } Y \leq 0,008856 \end{cases} \quad (\text{B.13})$$

$$a^* = 500 \left[f\left(\frac{X}{0,950456}\right) - f(Y) \right] + 128 \quad (\text{B.14})$$

$$b^* = 200 \left[f(Y) - f\left(\frac{Z}{1,088754}\right) \right] + 128 \quad (\text{B.15})$$

$$\text{onde } f(t) = \begin{cases} \sqrt[3]{t}, & \text{se } t > 0,008856 \\ 7,787t + 0,137931034, & \text{se } t \leq 0,008856 \end{cases} \quad (\text{B.16})$$

Os valores de L^* e os de a^* , e b^* se encontram nos intervalos inteiros $[0..100]$ e $[-127..127]$, respectivamente, com profundidade de 8 *bits* por canal de cor.

B.6 De RGB para CIE L*u*v

Primeiro são realizadas a transformação de RGB para XYZ seguida de normalização para o intervalo contínuo $[0..1]$. Em seguida:

$$L^* = \begin{cases} 116\sqrt[3]{Y} - 16, & \text{se } Y > 0,008856 \\ 903,3Y, & \text{se } Y \leq 0,008856 \end{cases} \quad (\text{B.17})$$

$$u^* = 13L \left(4 \frac{X}{X + 15Y + 3Z} - 0,19793943 \right) \quad (\text{B.18})$$

$$v^* = 13L \left(9 \frac{Y}{X + 15Y + 3Z} - 0,46831096 \right) \quad (\text{B.19})$$

Os valores de L^* , u^* e v^* se encontram nos intervalos inteiros $[0..100]$, $[-134..220]$ e $[-140..122]$, respectivamente, com profundidade de 8 *bits* por canal de cor.

B.7 De RGB para tons de cinza

Para transformar do espaço RGB para tons de cinza, com profundidade de 8 *bits*, é aplicada a seguinte equação:

$$G = 0,299R + 0,587G + 0,114B \quad (\text{B.20})$$

Os valores de G se encontram no intervalo inteiro $[0..255]$.