



**Universidade de Brasília**

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

**Aplicações para agricultura de precisão: Estimação de  
produção frutífera utilizando técnicas de visão  
computacional e aprendizagem de máquina**

Gustavo Ribeiro Teixeira  
Victor Mesquita Ferreira

Monografia apresentada como requisito parcial  
para conclusão do Bacharelado em Ciência da Computação

Orientador  
Prof. Dr. Flávio de Barros Vidal

Brasília  
2015

Universidade de Brasília — UnB  
Instituto de Ciências Exatas  
Departamento de Ciência da Computação  
Bacharelado em Ciência da Computação

Coordenador: Prof. Dr. Homero Luiz Piccolo

Banca examinadora composta por:

Prof. Dr. Flávio de Barros Vidal (Orientador) — CIC/UnB  
Prof. Dr. Juliana Martins de Mesquita Matos — CVTAAO/FAV/UnB  
Prof. Dr. Camilo Chang Dorea — CIC/UnB

### **CIP — Catalogação Internacional na Publicação**

Ribeiro Teixeira, Gustavo; Mesquita Ferreira, Victor.

Aplicações para agricultura de precisão: Estimação de produção frutífera utilizando técnicas de visão computacional e aprendizagem de máquina / Gustavo Ribeiro Teixeira, Victor Mesquita Ferreira. Brasília : UnB, 2015.

145 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2015.

1. agricultura de precisão, 2. contagem, 3. frutos, 4. processamento de imagens, 5. visão computacional

CDU 004.4

Endereço: Universidade de Brasília  
Campus Universitário Darcy Ribeiro — Asa Norte  
CEP 70910-900  
Brasília-DF — Brasil



# Dedicatória

Dedico este trabalho aos meus pais pelas lições ensinadas e repetidas ao longo dos anos até chegar nesta etapa da minha vida, pela compreensão e pelo suporte que me deram durante este período de muito aprendizado, de questionamentos e constantes mudanças de ideias, valores e virtudes. Dedico também a todos os alunos, professores e acadêmicos interessados no tema de pesquisa apresentado aqui e que porventura dêem continuidade a este projeto.

Gustavo Ribeiro Teixeira

# Dedicatória

Dedico este trabalho à minha família, principalmente aos meus pais, que me apoiaram e incentivaram durante todos estes anos e que proveram a educação, valores e virtudes que precisei para chegar até esta etapa da minha vida.

Victor Mesquita Ferreira

# Agradecimentos

Agradeço primeiramente ao professor Flávio de Barros Vidal pela paciência, empenho e conselhos oferecidos ao nos orientar neste trabalho. Ao professor André Costa Drummond pela orientação inicial, pelas experiências e lições compartilhadas durante nossas inúmeras conversas. Aos meus irmãos Gabriel e Ana Clara pelo bom humor e brincadeiras misturadas com seu apoio e compreensão. Aos amigos Felipe Viterbo e Lavousier Ferreira pela motivação e foco que me mantiveram firme neste trabalho. Ao meu amigo e parceiro de trabalho Victor, por sua ajuda, conhecimento e empenho para finalizar este trabalho. Aos familiares e amigos pelo apoio e compreensão dados durante esta fase em minha vida.

Gustavo Ribeiro Teixeira

# Agradecimentos

Agradeço ao professor Flávio de Barros Vidal por nos orientar e guiar neste trabalho árduo. Ao professor Díbio Leandro Borges por me orientar na etapa inicial do projeto. À minha irmã Vitória pelos momentos de descontração e risadas. Ao Evandro por me manter motivado durante todo esse tempo. Ao meu amigo e companheiro de trabalho Gustavo, por sua ajuda e dedicação para concluir este trabalho. Aos meus amigos e minha família pelo apoio e carinho que me deram durante esta etapa da minha vida.

Victor Mesquita Ferreira

# Resumo

A agricultura de precisão é o ramo da agricultura que utiliza diversos dados para a tomada de decisões em uma colheita. Uma informação importante obtida das colheitas é a estimativa da produção e uma das formas de se obter essa estimativa é realizando uma contagem dos frutos – uma tarefa árdua, que demanda muito tempo, de elevada taxa de imprecisão e com alto gasto financeiro.

Com a presença das tecnologias para auxílio no campo e a adoção da agricultura de precisão, vários meios de se acompanhar a colheita com um baixo custo vem sendo estudadas e estão se tornando disponíveis para uso na estimativa de produção, em especial a obtenção das sequências de imagens em série de tempo com sistemas de monitoramento.

Este trabalho teve por objetivo apresentar um método para realizar a detecção e contagem de frutos automática utilizando imagens retiradas da Internet – capturadas no campo de maneira não-destrutiva com máquinas fotográficas – fazendo uso de técnicas de processamento de imagens e visão computacional. A metodologia proposta segmenta as regiões que contém frutos nas imagens, e analisa a variação dos canais de cor para detectar pontos em que há sobreposição de frutos, separando as regiões segmentadas nestes pontos. Uma contagem do número final de regiões conectadas provê o número estimado de frutos na imagem. Os resultados são avaliados com a métrica *F-measure* a partir das anotações manualmente realizadas das imagens.

**Palavras-chave:** agricultura de precisão, contagem, frutos, processamento de imagens, visão computacional

# Abstract

Precision Agriculture is a field of agriculture which uses various data to decision-making in a harvest. An important information obtained from harvests is production estimatives and one way to obtain this estimate is making a production count – a hard and consuming task that has a high inaccuracy rate and high financial expense.

Using technologies to aid in the field and the adoption of precision agriculture, various ways of overseeing the harvest with low cost have been studied and are being made available to use in production estimative, in particular obtaining time-series image sequences with monitoring systems.

This work aims to introduce a method for automated fruit detection and counting using images acquired from the field in a non-destructive way using image processing and computer vision techniques. The proposed method segments the regions that contain fruits in the image, and analyses the variation in the color channels to detect points where there are fruits overlaps, then separates them in segmented regions. The estimated number of fruits in the image is obtained by counting the final number of connected regions. The results are evaluated by the F-measure using a created image ground truth.

**Keywords:** precision agriculture, count, fruits, image processing, computer vision

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Agricultura de precisão . . . . .	2
1.2	Justificativa . . . . .	3
1.3	Objetivos . . . . .	4
1.4	Metodologia proposta . . . . .	4
1.5	Organização dos capítulos . . . . .	4
<b>2</b>	<b>Revisão bibliográfica</b>	<b>5</b>
2.1	Segmentação de imagens de agricultura . . . . .	5
2.2	Contagem automática de frutos . . . . .	6
<b>3</b>	<b>Fundamentos teóricos</b>	<b>8</b>
3.1	Espaços de cor . . . . .	8
3.1.1	Espaço <i>RGB</i> . . . . .	9
3.1.2	Espaço <i>CIE L*a*b</i> . . . . .	10
3.2	Equalização de Histogramas . . . . .	11
3.3	Regiões conectadas . . . . .	12
3.3.1	Encontrando contornos de uma região . . . . .	13
3.3.2	Encontrando o centro de massa de uma região . . . . .	13
3.4	Segmentação de imagens . . . . .	13
3.4.1	<i>Thresholding</i> . . . . .	14
3.5	Árvores de decisão . . . . .	15
3.5.1	Estrutura interna e uso . . . . .	17
3.6	Transformação <i>Watershed</i> . . . . .	19
3.7	Transformação de distância . . . . .	20
3.8	Métrica de precisão <i>F-measure</i> . . . . .	21
<b>4</b>	<b>Metodologia Proposta</b>	<b>23</b>
4.1	Segmentação com árvores de decisão . . . . .	23
4.1.1	Extração de características de cor . . . . .	24
4.1.2	Geração da árvore de decisão . . . . .	25
4.1.3	Segmentação e geração de máscara . . . . .	26
4.2	Redução de ruído com transformação <i>Watershed</i> . . . . .	26
4.3	Transformação de distância . . . . .	27
4.4	Corte de regiões com frutos sobrepostos . . . . .	28
4.4.1	Encontrar contornos das regiões . . . . .	29
4.4.2	Encontrar centro de massa da região . . . . .	29

4.4.3	Encontrar variação de cor com conectividade oito . . . . .	30
4.4.4	Corte da região . . . . .	30
4.5	Contagem de regiões conectadas . . . . .	32
<b>5</b>	<b>Resultados e discussão</b>	<b>33</b>
5.1	Construção das bases de imagens . . . . .	33
5.2	Realização dos testes . . . . .	35
5.3	Resultados . . . . .	36
5.4	Discussão . . . . .	36
<b>6</b>	<b>Conclusão</b>	<b>40</b>
6.1	Perspectivas futuras . . . . .	41
<b>A</b>	<b>Resultados completos</b>	<b>42</b>
	<b>Referências</b>	<b>59</b>

# Lista de Figuras

1.1	Ciclo de agricultura de precisão. . . . .	3
3.1	Esquema do cubo de cores <i>RGB</i> . . . . .	9
3.2	Cubo de cores <i>RGB 24-bits</i> . . . . .	10
3.3	Desenho da esfera do espaço de cores <i>CIE L*a*b</i> . . . . .	11
3.4	Histogramas em nível de cinza . . . . .	14
3.5	Exemplo de uma árvore de decisão para segmentação de imagens de agricultura. . . . .	16
4.1	Fluxograma do algoritmo para contagem de frutos em imagens. . . . .	24
4.2	Seleção de regiões de interesse para treinamento da árvore de decisão. . . . .	25
4.3	Exemplo de uma imagem de acerola segmentada com árvores de decisão. . . . .	26
4.4	Máscara binária de segmentação após redução de ruído com <i>Watershed</i> . . . . .	27
4.5	Máscara binária após realização da transformação de distância. . . . .	28
4.6	Fluxograma do algoritmo de corte de regiões conectadas que contém frutos sobrepostos. . . . .	29
4.7	Máscara binária anterior com o centro de massa da região destacado em vermelho. . . . .	30
4.8	Imagem ilustrando os oito segmentos de reta na imagem onde será feita a análise de variação de cor para o corte de região. . . . .	31
4.9	Exemplo ilustrativo de corte realizado na máscara binária vista anteriormente. . . . .	31
5.1	Exemplos de fotos das bases de imagens de frutas coletadas da Internet. . . . .	34
5.2	Gráfico dos principais resultados de <i>F-measure</i> das bases de imagens. . . . .	37
5.3	Gráficos de alguns resultados de <i>F-measure</i> ilustrando seu comportamento. . . . .	38
A.1	Gráfico dos resultados de <i>F-measure</i> para a base de acerola, realizando treinamento com 30% e teste com 70% da base. . . . .	43
A.2	Gráfico dos resultados de <i>F-measure</i> para a base de acerola, realizando treinamento com 70% e teste com 30% da base. . . . .	44
A.3	Gráfico dos resultados de <i>F-measure</i> para a base de ameixa, realizando treinamento com 30% e teste com 70% da base. . . . .	45
A.4	Gráfico dos resultados de <i>F-measure</i> para a base de ameixa, realizando treinamento com 70% e teste com 30% da base. . . . .	46
A.5	Gráfico dos resultados de <i>F-measure</i> para a base de caju, realizando treinamento com 30% e teste com 70% da base. . . . .	47

A.6	Gráfico dos resultados de <i>F-measure</i> para a base de caju, realizando treinamento com 70% e teste com 30% da base. . . . .	48
A.7	Gráfico dos resultados de <i>F-measure</i> para a base de cereja, realizando treinamento com 30% e teste com 70% da base. . . . .	49
A.8	Gráfico dos resultados de <i>F-measure</i> para a base de cereja, realizando treinamento com 70% e teste com 30% da base. . . . .	50
A.9	Gráfico dos resultados de <i>F-measure</i> para a base de laranja, realizando treinamento com 30% e teste com 70% da base. . . . .	51
A.10	Gráfico dos resultados de <i>F-measure</i> para a base de laranja, realizando treinamento com 70% e teste com 30% da base. . . . .	52
A.11	Gráfico dos resultados de <i>F-measure</i> para a base de manga rosa, realizando treinamento com 30% e teste com 70% da base. . . . .	53
A.12	Gráfico dos resultados de <i>F-measure</i> para a base de manga rosa, realizando treinamento com 70% e teste com 30% da base. . . . .	54
A.13	Gráfico dos resultados de <i>F-measure</i> para a base de morango, realizando treinamento com 30% e teste com 70% da base. . . . .	55
A.14	Gráfico dos resultados de <i>F-measure</i> para a base de morango, realizando treinamento com 70% e teste com 30% da base. . . . .	56
A.15	Gráfico dos resultados de <i>F-measure</i> para a base de pêssego, realizando treinamento com 30% e teste com 70% da base. . . . .	57
A.16	Gráfico dos resultados de <i>F-measure</i> para a base de pêssego, realizando treinamento com 70% e teste com 30% da base. . . . .	58

# Lista de Tabelas

3.1	Matriz de confusão para avaliar a qualidade de uma classificação binária. . .	21
5.1	Informações sobre as bases de imagens de frutos utilizadas para os testes. . .	35
5.2	Principais resultados de <i>F-measure</i> para cada base de imagens. . . . .	36
A.1	Resultados de <i>F-measure</i> para a base de acerola, realizando treinamento com 30% e teste com 70% da base. . . . .	42
A.2	Resultados de <i>F-measure</i> para a base de acerola, realizando treinamento com 70% e teste com 30% da base. . . . .	43
A.3	Resultados de <i>F-measure</i> para a base de ameixa, realizando treinamento com 30% e teste com 70% da base. . . . .	44
A.4	Resultados de <i>F-measure</i> para a base de ameixa, realizando treinamento com 70% e teste com 30% da base. . . . .	45
A.5	Resultados de <i>F-measure</i> para a base de caju, realizando treinamento com 30% e teste com 70% da base. . . . .	46
A.6	Resultados de <i>F-measure</i> para a base de caju, realizando treinamento com 70% e teste com 30% da base. . . . .	47
A.7	Resultados de <i>F-measure</i> para a base de cereja, realizando treinamento com 30% e teste com 70% da base. . . . .	48
A.8	Resultados de <i>F-measure</i> para a base de cereja, com treinamento com 70% e teste com 30% da base. . . . .	49
A.9	Resultados de <i>F-measure</i> para a base de laranja, com treinamento com 30% e teste com 70% da base. . . . .	50
A.10	Resultados de <i>F-measure</i> para a base de laranja, realizando treinamento com 70% e teste com 30% da base. . . . .	51
A.11	Resultados de <i>F-measure</i> para a base de manga rosa, realizando treinamento com 30% e teste com 70% da base. . . . .	52
A.12	Resultados de <i>F-measure</i> para a base de manga rosa, com treinamento com 70% e teste com 30% da base. . . . .	53
A.13	Resultados de <i>F-measure</i> para a base de morango, realizando treinamento com 30% e teste com 70% da base. . . . .	54
A.14	Resultados de <i>F-measure</i> para a base de morango, com treinamento com 70% e teste com 30% da base. . . . .	55
A.15	Resultados de <i>F-measure</i> para a base de pêsego, realizando treinamento com 30% e teste com 70% da base. . . . .	56
A.16	Resultados de <i>F-measure</i> para a base de pêsego, realizando treinamento com 70% e teste com 30% da base. . . . .	57

# Capítulo 1

## Introdução

A agricultura de precisão é o ramo da agricultura que usa dados como clima, qualidade do solo e do ar, maturidade da plantação e até custos de mão-de-obra e de equipamentos, coletados em tempo real para realizar decisões mais inteligentes para a colheita, procurando-se aumentar o retorno econômico e minimizar os efeitos ambientais [27]. O campo de cultivo não é mais visto como uma unidade uniforme como na colheita tradicional e com os dados coletados é possível calcular a quantidade exata de insumos necessários para cada região do campo. De acordo com Zhang *et al.* [34], os métodos e tecnologias usados podem incluir o Sistema de Posicionamento Global (*GPS*), Sistemas de Informação Geográfica (*GIS*), componentes de computadores, sensoriamento remoto de campo, computação móvel, processamento de informações avançado e telecomunicações.

Dentro deste contexto, uma informação importante é a estimativa de produção da colheita. Com essa estimativa, adquire-se informações sobre polinização, estratégias de desbaste, tamanho da produção e planejamento do armazenamento, orçamento financeiro necessário, entre outros. Dentre as diversas formas de se obter/realizar a estimativa de colheita, uma forma simples e eficaz é realizando uma contagem da produção (por exemplo frutos) no campo. No entanto, isso é uma tarefa árdua, normalmente feita visualmente, requerendo que observadores treinados fiquem no campo várias horas por dia, demandando muito tempo, elevada imprecisão (o fator cansaço humano influencia em muito neste processo), além do elevado gasto financeiro para se realizar esta atividade.

Com o advento de tecnologias para auxílio na agricultura e a adoção da agricultura de precisão no campo, várias tecnologias estão se tornando disponíveis para auxílio na estimativa de produção. Em particular, a obtenção de sequências de imagens em série de tempo com sistemas de monitoramento de campo de baixo custo como o *Field Server* [15] facilitam estudos usando técnicas de visão computacional [31]. Essas imagens são capturadas automaticamente do campo, de forma não-destrutiva, permitindo assim que sejam feitas análises sobre essas imagens de todo o campo sem interferir na colheita.

Deseja-se a partir da utilização de técnicas, utilizando como única entrada a imagem capturada da produção, auxiliar os agricultores na estimativa de produção de colheita fornecendo uma solução automatizada que realize a contagem de frutos. Essa contagem deve

ser realizada de forma não-destrutiva, ou seja, sem interferir na colheita, e portanto deve ser robusta o suficiente para funcionar em condições naturais de iluminação solar e demais condições para o seu uso no campo. Este trabalho propõe uma metodologia para realizar a detecção e contagem de frutos por meio de imagens capturadas no campo de forma não-destrutiva utilizando técnicas de processamento de imagens e visão computacional.

## 1.1 Agricultura de precisão

De acordo com Pokrajac *et al.* [23], na agricultura tradicional, considera-se toda a área da plantação como homogênea e assim, para aplicar insumos como água, fertilizantes, defensivos e outros, deve-se considerar as necessidades médias da plantação. Logo, a mesma quantidade de insumos é aplicada em toda a área, ignorando necessidades específicas de cada parte do campo. Isso leva a lavouras com produtividade não uniforme.

A agricultura de precisão surgiu como uma forma de realizar decisões mais inteligentes na plantação com o objetivo de aumentar a produtividade, a eficiência do que é gerado, minimizar custos e os impactos ambientais. O campo é gerido diferentemente para inúmeras áreas. Em cada área, os insumos são aplicados no momento exato e na quantidade exata necessários à produção agrícola. O livro Agricultura de Precisão: Um Novo Olhar [27] da Embrapa, explica em mais detalhes a agricultura de precisão.

Entre as tecnologias, *softwares* e técnicas comumente utilizadas em agricultura de precisão estão imagens de satélite para levantamento georreferenciado no campo, mapeamento da condutividade elétrica do solo, colheita com máquinas com sensores de produtividade, mapeamento do plantio, produção e a aplicação de fertilizantes e defensivos agrícolas a taxas variáveis, entre outros. A Figura 1.1 mostra o ciclo geral de agricultura de produção para cada safra.

Para gerenciar a variabilidade espacial e temporal que existe no campo, em geral utiliza-se sistemas baseados em mapas, com a ajuda de tecnologias de *GPS* (Sistemas de Posicionamento Global) e *GIS* (Sistemas de Informação Geográfica), sensoriamento remoto, monitoramento de produção e amostragem de solo. Métodos avançados de geoestatística são usados para analisar a variabilidade espacial e temporal. Incorporando técnicas de modelagem da colheita, é possível desenvolver mapas de potencial produtivo como base para prescrição de fertilizantes. Baseando-se em condições climáticas projetadas, esses mapas podem ser usados para prever a variabilidade no crescimento da colheita e doenças na plantação.

A agricultura de precisão traz muitos benefícios para os produtores e para a população em geral. Há um ganho em lucratividade devido às análises econômicas que podem ser feitas mais precisamente, permitindo ao produtor obter avaliações exatas de risco e economizar insumos como fertilizantes com aplicação de somente quantidades exatas necessárias de acordo com a variabilidade espacial e temporal. Isso também leva a uso reduzido de agrotóxicos, levando a redução do impacto ao meio ambiente.



Figura 1.1: Ciclo de agricultura de precisão. Adaptado de Ciclo de Agricultura de Precisão [1].

## 1.2 Justificativa

Muitos produtores agrícolas no Brasil precisam e fazem uso de técnicas de agricultura de precisão para aumentar a produtividade da colheita de seus produtos. Neste trabalho abordaremos o desenvolvimento de técnicas a partir da extração da quantidade de frutos contidas em imagens. Técnicas como mapeamento do solo e aplicação variável de insumos, tecnologias como o posicionamento global por satélites (*GPS*) e sistemas de mapeamento variados são extremamente importantes para garantir a produtividade e uniformidade da produção.

Para alcançar esta meta, deve-se fazer uso de técnicas de estimativa de produção da colheita. A estimativa de produção ajuda no planejamento de estratégias de desbaste, que é a eliminação de plantas excedentes na colheita, provê informações sobre a polinização do campo e sobre o número de frutos estimados a serem produzidos, que ajuda a realizar planejamento de armazenamento da produção, orçamento e previsão de lucros.

Uma forma direta de se realizar estimativas de produção é fazendo uma contagem dos frutos no campo. No entanto, essa contagem de frutos demanda e mão-de-obra, visto que ela é feita em grande parte visualmente, necessitando de um observador no campo por várias horas. Há um gasto de recursos humanos, além de ser um parâmetro subjetivo dependendo do avaliador, o que pode prejudicar a estimativa. Este trabalho procura solucionar esse problema de estimativas com uma solução computacional desenvolvida para essa finalidade.

## 1.3 Objetivos

O objetivo geral deste trabalho é criar uma solução computacional que realize contagem de variados tipos de frutos automaticamente, auxiliando na estimativa de produção da colheita. A partir desta informação será possível no futuro o desenvolvimento de modelos que possam permitir uma estimativa precisa e confiável da produção agrícola.

Um objetivo específico é desenvolver uma metodologia em forma de software que realize a estimativa de contagem de frutos. A ferramenta deverá receber como entrada um lote de imagens do campo contendo frutos e retornar o número estimado de frutos naquela imagem. Deseja-se que a ferramenta possa ser aplicada a imagens de diversos frutos, com níveis de qualidade e de resolução variados, capturadas no campo com máquinas fotográficas em condições naturais de iluminação, em diversos ângulos, posicionamento, e diversas intempéries encontradas em situações reais no campo. Será usada uma métrica para avaliar a precisão do algoritmo, e deseja-se alcançar uma precisão de 70% de acertos da estimativa.

## 1.4 Metodologia proposta

A metodologia proposta neste trabalho para realizar a estimativa de número de frutos em imagem envolve realizar uma segmentação dos frutos na imagem utilizando árvores de decisão, realizar uma redução de ruído na máscara binária de segmentação utilizando transformação *Watershed*, redução das regiões na máscara com transformação de distância, análise da variação de cor e/ou luminância na imagem para separação de regiões com frutos sobrepostos, e contagem do número de regiões conectadas finais, que fornece a estimativa do número de frutos.

## 1.5 Organização dos capítulos

Este trabalho está dividido de acordo com a ordem a seguir: o Capítulo 2 realiza uma revisão de literatura de trabalhos recentes que aplicam o processamento de imagens em problemas de agricultura. O Capítulo 3 contém fundamentos teóricos de algoritmos e técnicas de processamento de imagens e visão computacional necessários para o entendimento da metodologia. No Capítulo 4 é apresentada a metodologia proposta por este trabalho para detecção e contagem automática de frutos por meio de imagens do campo. O Capítulo 5 apresenta os resultados experimentais da metodologia proposta e a análise e discussão dos resultados. O Capítulo 6 contém considerações finais e propostas para trabalhos e perspectivas futuras.

# Capítulo 2

## Revisão bibliográfica

Este capítulo dedica-se a realizar uma revisão de trabalhos recentes sobre técnicas de visão computacional e processamento de imagens aplicadas à agricultura, como técnicas de segmentação de imagens de agricultura e técnicas de estimativa de produção.

### 2.1 Segmentação de imagens de agricultura

Uma técnica muito utilizada para segmentação entre solo e planta em imagens de agricultura é chamada segmentação de *greeness* [24, 32]. Guijarro *et al.* [20] propuseram um método para melhorar essa técnica utilizando uma transformada *wavelet* discreta. O algoritmo extrai a informação de *greeness* da imagem e aplica a decomposição baseada em *wavelet*, obtendo imagens com coeficientes de aproximação, vertical, horizontal e diagonal. Descritores de textura são aplicados a essas imagens para obter a variabilidade espacial e então a transformada *wavelet* inversa é realizada, retornando uma imagem com informação de *greeness* melhorada. Finalmente, aplica-se o método de limiarização de Otsu para obter-se a imagem binária que discrimina o solo da parte verde da imagem.

O método obtém resultados melhores que os métodos de extração de *greeness* convencionais, quantificados em 4,5%. No entanto, sua aplicabilidade a este trabalho é limitada, pois o método destina-se a separar as regiões de planta em geral do solo, enquanto que neste processo de segmentação de frutos estamos interessados em separar os frutos de outros componentes.

Hung *et al.* [7] aplicam um método de aprendizado semi-supervisionado em imagens multiespectrais para segmentar frutos. O método utiliza aprendizado de características multiescalas não-supervisionado para extrair características multiescala e treinamento de campos aleatórios condicionais semi-supervisionado para segmentação da imagem em múltiplas classes.

O método possui a vantagem de ser em sua maior parte não-supervisionado, e os resultados mostram uma precisão de 88% na segmentação. No entanto, o método somente

marca as regiões correspondentes a frutos e não fornece uma estimativa do número de frutos. Seria necessária uma forma de dividir as regiões conectadas e segmentadas que contenham dois ou mais frutos parcialmente sobrepostos.

Guo *et al.* [31] desenvolveram um método para segmentação de imagens de agricultura utilizando um algoritmo de aprendizado de máquina chamado árvores de decisão. Esse algoritmo é usado para classificação ou regressão de dados criando uma estrutura em formato de árvore, em geral binária, e realiza decisões em cada nó até uma folha ser alcançada, onde o resultado é obtido. O método é particularmente eficiente com relação a outros métodos de segmentação de vegetação por ser robusto quanto a iluminação solar. Isso é extremamente importante no contexto deste trabalho, visto que um dos objetivos é criar um método para estimativa de frutos no campo em condições naturais de iluminação.

O método utiliza atributos de cor da imagem para classificar cada pixel em classes bem definidas, como por exemplo segmentar solo e folhas. Esses atributos de cores são os valores dos canais de vários espaços de cores diferentes, como os espaços *RGB*, *HSV*, *CIE L\*a\*b* e *CIE L\*u\*v*. A árvore é criada a partir dos atributos de cor de um subconjunto de dados previamente classificados.

Esse método possui a pequena desvantagem de ser supervisionado, necessitando de um conjunto de dados previamente classificados antes de realizar a segmentação. Como o procedimento utiliza somente características de cor, ele também está sujeito a erros decorrentes de cores similares entre classes diferentes ou variação de cores dentro de uma mesma classe.

## 2.2 Contagem automática de frutos

Yamamoto *et al.* [17] utilizam a técnica de segmentação de imagens de Guo *et al.* [31] para segmentar e contar frutos de tomates em imagens. Inicialmente é realizada uma segmentação por cor pixel a pixel seguindo o algoritmo de Guo *et al.* [31]. Com essa segmentação, são obtidas máscaras indicando regiões que contenham frutos de tomate. No entanto, essas regiões podem conter mais de um fruto sobrepostos, ou em alguns casos, por ruído da segmentação, podem não conter nenhum fruto.

Para resolver este problema, é realizada outra classificação com árvores de decisão, desta vez utilizando como entrada características de cor e formato de regiões que podem conter nenhuma fruta, somente uma fruta ou múltiplas frutas. Com um treinamento realizado, é possível classificar as regiões obtidas pela segmentação de cores em regiões sem frutas, contendo uma fruta ou contendo múltiplas frutas. Finalmente, para contar o número de frutas, basta separar as frutas das regiões com múltiplas frutas. Para isso, identifica-se regiões de superexposição dentro das regiões contendo múltiplas frutas. Essas regiões são causadas pelo *flash* da câmera em cada fruta. Com isso, é possível contar o número de tomates contidos em uma imagem.

Apesar dos bons resultados, este método depende de um ambiente controlado de iluminação para a obtenção dos pontos de superexposição e de um método de captura de imagens destrutivo.

Outro método que faz uso de pontos de superexposição para contagem de frutos é o método proposto por Font *et al.* [10]. Aqui, no entanto, a captura não é feita de forma

destrutiva, e sim com o uso de um sistema de captura com *flash* utilizado durante a noite, criando assim os picos de reflexão especular nas uvas que serão contadas. O método proposto realiza uma segmentação das uvas com uma limiarização de Otsu sobre o canal  $H$  do espaço de cor  $HSV$  e aplicação de filtros morfológicos para redução de ruído. Um filtro de média bidimensional é aplicado para suavizar a imagem, e então os picos de reflexão especular são detectados analisando-se a intensidade de um ponto central comparado à intensidade de pontos radiais em torno dele. Se a intensidade do ponto central for maior que a dos pontos radiais, ele é considerado um pico e é contabilizado.

Apesar de a captura de imagens e contagem de frutos ser feita de forma não-destrutiva, as condições de iluminação são altamente controladas ao se capturar imagens somente durante a noite com *flash*.

Payne *et al.* [5] estimam o número de mangas em um pomar realizando uma segmentação inicial como nos outros métodos, mas optando por uma regressão linear para estimar o número de frutos, compensando por possíveis erros no método. Pixels com predominância de vermelho (canal  $R$ ) sobre o verde (canal  $G$ ) do espaço  $RGB$ , são selecionados, a imagem é processada com um filtro de variância e limiarizada, removendo pixels de áreas com muitas bordas, como na grama ou folhagem, bem como pixels onde há pouca variação, como no céu. Os canais  $Cb$  e  $Cr$  do espaço de cor  $YCbCr$  também são limiarizados e os resultados anteriores são combinados, criando uma máscara binária dos frutos. O número de partículas na máscara é contado usando limitadores superior e inferior. Uma regressão linear é feita nos resultados para ajustar a contagem para um número o mais próximo do real.

O método mostra vantagens em seu passo que utiliza o filtro de variância. O artigo indica que este passo provê um mecanismo que contabiliza pela maior parte dos frutos sobrepostos. O método no entanto obtém uma taxa maior de erros com quantidades maiores de frutos. Também possui a desvantagem de ter vários ajustes específicos para a base testada.

# Capítulo 3

## Fundamentos teóricos

Este capítulo descreve os fundamentos teóricos de visão computacional e processamento de imagens necessários para o entendimento deste trabalho.

### 3.1 Espaços de cor

De acordo com Gonzalez [13], o propósito de um *color space* (também chamado de *color model* ou *color system*) é o de facilitar a especificação de cores em algum padrão geralmente aceito. Em essência, um espaço de cor é uma especificação de um sistema de coordenadas e um subespaço interior desse sistema onde cada cor é representada por um único ponto.

A maioria dos espaços de cor hoje estão voltados tanto para hardware (como para monitores coloridos e impressoras) ou voltados para aplicações onde a manipulação das cores é um objetivo (como na criação de desenhos coloridos para animações). Em termos de processamento de imagens, Gonzalez [13] menciona que os modelos voltados para hardware comumente usam na prática o modelo *RGB* (*Red, Green, Blue*) para monitores coloridos e uma vasta classe de câmeras de vídeo. Outro modelo também reconhecido para o uso no processamento de imagens é o *CIE L\*a\*b*, usado para trabalhar com imagens digitais de máquinas, impressoras e scanners.

Existem numerosos espaços de cor em uso atualmente devido ao fato de que a ciência das cores é um campo enorme que engloba várias áreas de aplicação. Logo, os modelos discutidos neste capítulo serão modelos voltados para o processamento de imagens e que foram utilizados na elaboração deste trabalho.

### 3.1.1 Espaço $RGB$

Gonzalez [13] explica que no espaço  $RGB$  cada cor aparece em seu componente primário de espectro vermelho, verde e azul. Esse modelo é baseado em um sistema de coordenadas cartesianas e o subespaço de interesse é o cubo apresentado na Figura 3.1, onde os valores  $RGB$  estão em três cantos; as cores ciano, magenta e amarelo estão em outros três vértices; a cor preta está na origem e a cor branca está na ponta mais distante da origem. Neste modelo, a escala de cinza (pontos com valores  $RGB$  equivalentes) se estende do preto ao branco ao longo da linha que une esses dois pontos. As diferentes cores nesse modelo são pontos no interior do cubo e são definidos por vetores que partem da origem. Por conveniência, assumimos que todos os valores de cor foram normalizados de maneira que o cubo mostrado na Figura 3.1 é um cubo unitário, ou seja, que todos os valores de  $R$ ,  $G$  e  $B$  estão dentro do intervalo  $[0,1]$ .

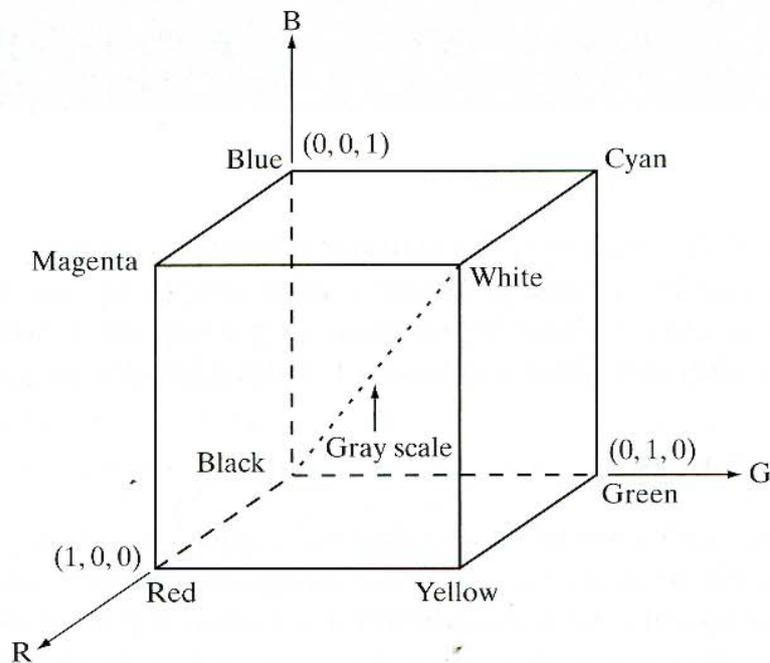


Figura 3.1: Esquema do cubo de cores  $RGB$ . Os pontos ao longo da diagonal principal tem valores de cinza, do preto na origem até o branco no ponto  $(1,1,1)$ . Adaptado de Gonzalez [13].

As imagens representadas no espaço  $RGB$  consistem em três componentes de imagem, uma para cada cor primária. Gonzalez [13] cita que, quando introduzidas em um monitor  $RGB$ , essas três imagens se combinam na tela para produzir uma imagem de cor composta. O número de bits usados para representar cada pixel no espaço  $RGB$  é chamado de *pixel depth*. Considere uma imagem  $RGB$  onde cada uma das imagens vermelha, verde e azul é uma imagem de  $8$ -bits. Sob essas condições, cada *cor* do pixel  $RGB$  [isto é, uma tripla dos valores  $(R,G,B)$ ] é considerada como tendo uma profundidade de  $24$ -bits (três planos de imagem vezes o número de bits por plano). O cubo mostrado na Figura 3.2 é um sólido, composto de  $(2^8)^3 = 16.777.216$  cores mencionado acima.

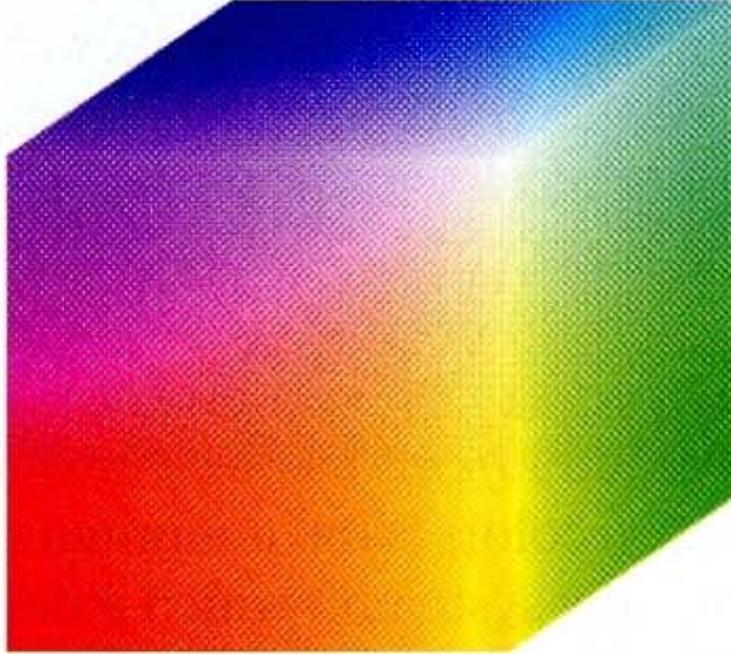


Figura 3.2: Desenho do cubo de cores *RGB 24-bits*. O cubo é composto de 16.777.216 cores, cada uma contendo um valor para a tripla (R,G,B). Adaptado de Gonzalez [13].

### 3.1.2 Espaço *CIE L\*a\*b*

De acordo com [2], o espaço de cor definido pela Comissão Internacional em Iluminação (*CIE*), baseado em um canal para a *luminância* (ou luminosidade, definida pela letra *L*) e dois canais de cores (*a* e *b*). Um problema encontrado no sistema de cores distribuído em coordenadas *x*, *y* e *z* (tal como o *RGB*) é que as distâncias colorimétricas entre as cores individuais não correspondem as diferenças de cores percebidas.

O *CIE* resolveu esse problema com o desenvolvimento do espaço de cores tridimensional *Lab* (ou espaço de cores *CIE L\*a\*b*). Nesse modelo, as diferenças de cor percebidas correspondem às distâncias quando mensuradas colorimetricamente. O eixo *a* se estende do verde ( $-a$ ) até o vermelho ( $+a$ ) e o eixo *b* do azul ( $-b$ ) até o amarelo ( $+b$ ). O brilho (*L*) aumenta da base até o topo nesse modelo tridimensional, que pode ser visualizado na Figura 3.3.

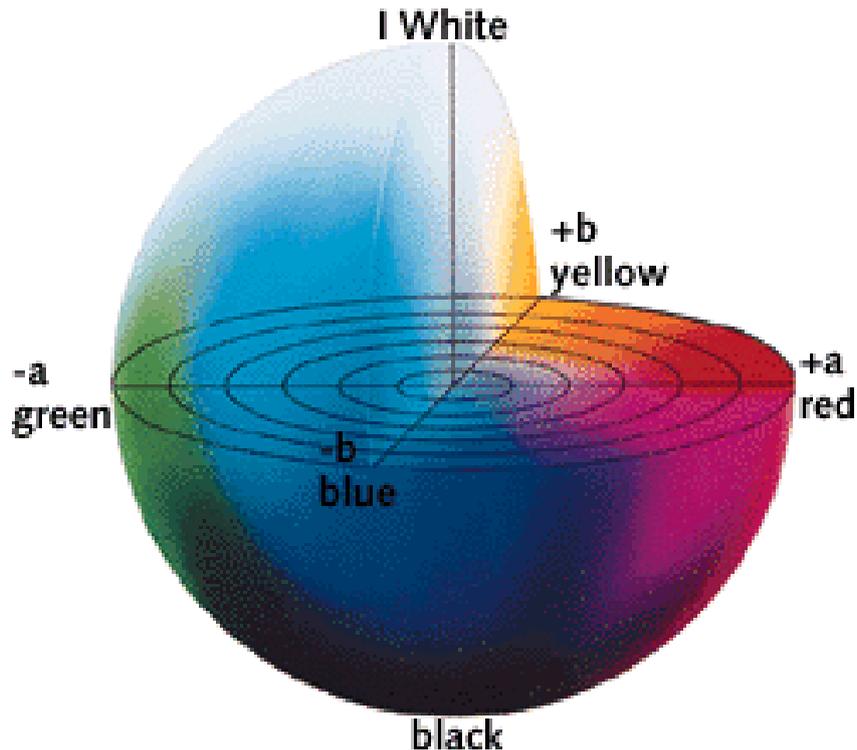


Figura 3.3: Desenho da esfera do espaço de cores  $CIE L^*a^*b$ , indicando as transições de cores do verde ao vermelho, do azul ao amarelo e a transição de luminosidade, que vai do preto ao branco. Adaptado de [2].

Esse espaço de cor é mais adequado para manipulações de imagens digitais que o espaço  $RGB$ , que é tipicamente utilizado em programas de edição de imagens. Por exemplo, o espaço  $CIE L^*a^*b$  é útil para avivar imagens e para remover artefatos em imagens  $JPEG$  ou em imagens de câmeras digitais e scanners.

## 3.2 Equalização de Histogramas

A equalização de histogramas (HE) é uma das principais técnicas utilizadas para aprimoramento do contraste (CE) em imagens. De acordo com o trabalho de Suh [29] para reconhecimento de discursos, a HE é uma eficiente transformação não-linear baseada na normalização de traços ou abordagem por modelos adaptados em razão da sua simplicidade algorítmica. A ideia básica da equalização de histogramas é a de normalizar as funções de densidade probabilística (PDFs) entre os dados de treinamento e de teste. Essa normalização pode ser alcançada ao converter o PDF dos traços de teste para o PDF dos traços de treinamento, e assim a HE pode compensar as discrepâncias entre os dados de treino e os de teste.

Ao explicar sobre o aprimoramento de imagens escuras em seu trabalho, Ling *et al.* [33] mencionam que a facilidade em implementar e produzir resultados satisfatórios em vários casos contribuiu na popularização da HE como método de aprimoramento de imagens. A equalização de histogramas uniformemente redistribui o alcance da escala de cinza para

aumentar a diferença média entre quaisquer dois níveis de cinza alterados. Por exemplo, o autor cita a equalização global de histogramas (GHE) de Gonzalez e Woods [13] baseado em uma função de distribuição cumulativa que redistribui um histograma original para um aprimoramento de contraste. Além disso, Ling cita a equalização de histogramas em duas dimensões (2DHE), um exemplo vindo do trabalho de Celik [9], que aumenta o contraste de uma imagem ao elevar as diferenças nos níveis de cinza entre os pixels vizinhos. Com base no 2DHE, uma representação em camadas da diferença dos histogramas 2D se faz presente para aprimorar as imagens, conforme complementa o trabalho de Lee *et al.* [8], citado por Ling *et al.* [33].

Segundo Gonzalez e Woods [13] citam em seu livro e conforme o trabalho de Kaur *et al.* [21] com imagens de tomografia computadorizada, se os níveis de intensidade são normalizados dentro do intervalo  $[0, L - 1]$ , onde  $L$  indica o total de níveis de cinza em uma imagem, e a função de densidade probabilística da imagem de entrada é  $P_r(w)$ , então o nível de intensidade de saída será  $s = \int_0^r P_r(w)dw$ , onde  $w$  é uma variável de integração. A função de densidade probabilística da imagem de saída será  $P_s(s)$  para  $0 \leq s \leq L - 1$  ou então  $P_s(s) = 0$  caso esteja fora do intervalo  $[0, L - 1]$ , o que expande o alcance desse intervalo e assim o contraste das imagens é aprimorado.

### 3.3 Regiões conectadas

De acordo com Gonzalez e Woods [13], o conceito de conectividade entre pixels é fundamental para simplificar a definição de imagens digitais, permitindo definir regiões e limites. Dois pixels estão conectados se são considerados vizinhos e se ambos satisfazem certos critérios de similaridade, como por exemplo se ambos possuem valores de cinza iguais.

Dois pixels podem ser considerados vizinhos se estão um imediatamente seguido do outro na vertical ou na horizontal, considerando uma vizinhança de 4, ou também na diagonal, considerando uma vizinhança de 8. A adjacência entre dois pixels que satisfazem esses critérios é definida por uma adjacência-4, que considera somente vizinhança de 4, uma adjacência-8, considerando uma vizinhança de 8, ou adjacência- $m$ , similar à adjacência-8, mas que não possui caminhos ambíguos.

Dois pixels são considerados conexos se existe um caminho entre eles, ou seja, uma sequência de pixels distintos que são adjacentes. Assim, uma região de uma imagem é um subconjunto de pixels que satisfazem um critério estabelecido e estão conectados, ou seja, existe um caminho entre qualquer um dos pixels dentro da região.

O conceito de regiões conectadas é importante para a segmentação de imagens, pois o objetivo da segmentação é separar a imagem em regiões conectadas que possuem alguma similaridade definida.

Com o conceito de regiões conectadas, é possível extrair informações das regiões, como extrair as bordas (ou contornos) de uma região conectada ou encontrar o centro de massa de uma região.

### 3.3.1 Encontrando contornos de uma região

Suzuki e Abe [30] propuseram uma técnica para encontrar contornos de regiões conectadas de uma imagem. Essa técnica é utilizada em imagens binárias, como as que são obtidas de uma segmentação de duas classes.

O algoritmo marca um pixel com um número quando ele está em uma borda entre duas regiões, ou seja, o pixel está exatamente entre uma região de pixels 0 e uma região de pixels 1. O algoritmo segue esses pixels de borda até conseguir retornar ao pixel inicial, concluindo o percorrimento da borda. A numeração das bordas determina se ela é borda externa ou uma borda de buraco e ainda determina a hierarquia de regiões e buracos na imagem.

### 3.3.2 Encontrando o centro de massa de uma região

Um momento de uma imagem é uma certa média ponderada particular (momentos) de intensidades de pixels da imagem ou uma função de tais momentos, escolhida para ter certa propriedade ou interpretação. Hu [16] utiliza esses momentos para reconhecimento de padrões visuais. Momentos permitem a extração de propriedades simples da imagem ou região, entre eles a posição do centro de massa.

De acordo com o algoritmo utilizado para calcular momentos de imagens bidimensionais em domínio discreto [3], os momentos espaciais  $m_{ji}$  de uma imagem ou polígono são dados por:

$$m_{ji} = \sum_{x,y} (I(x,y) \cdot x^j \cdot y^i), \quad (3.1)$$

onde  $I(x,y)$  é o valor de intensidade da imagem de entrada nas coordenadas  $(x,y)$ . Assim, as coordenadas  $(\bar{x}, \bar{y})$  do centro de massa da imagem podem ser obtidas por:

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}. \quad (3.2)$$

## 3.4 Segmentação de imagens

Em seu livro, Gonzalez e Woods [13] explicam que o processo de segmentação subdivide uma imagem em seus objetos ou regiões constituintes. O nível em que a subdivisão é perpetuada depende do problema a ser solucionado, ou melhor, a segmentação deve parar quando os objetos de interesse em uma aplicação forem isolados. Segmentar imagens não-triviais é uma das tarefas mais difíceis em processamento de imagens, pois a precisão nesse processo pode determinar o sucesso ou falha de procedimentos de análise computadorizados.

Os algoritmos de segmentação trabalham sobre imagens em escala de cinza e geralmente são baseados em uma das duas propriedades básicas de intensidade de valores: descontinuidade e similaridade. Na primeira categoria, a abordagem é feita ao participar uma imagem baseada nas súbitas mudanças de intensidade, como nas bordas da

imagem. As abordagens feitas na segunda categoria são baseadas em particionar uma imagem em regiões similares de acordo com um conjunto de critérios predefinidos, tais como o *thresholding*.

### 3.4.1 *Thresholding*

Gonzalez e Woods [13] citam que, por suas propriedades intuitivas e simplicidade no uso, o *thresholding* ocupa uma posição central nas aplicações de segmentação de imagens. Supondo que o histograma em escala de cinza apresentado no histograma *a*) da Figura 3.4 corresponda a uma imagem  $f(x, y)$  composta por objetos claros em um fundo escuro de tal forma que os pixels do objeto e do plano de fundo tenham níveis de cinza agrupados de dois modos dominantes. Uma maneira óbvia de extrair os objetos do fundo é selecionar um limiar (em inglês, *threshold*)  $T$  que irá separar esses modos. Logo, qualquer ponto  $(x, y)$  no qual  $f(x, y) > T$  é chamado de *object point*; do contrário, o ponto é chamado de *background point*.

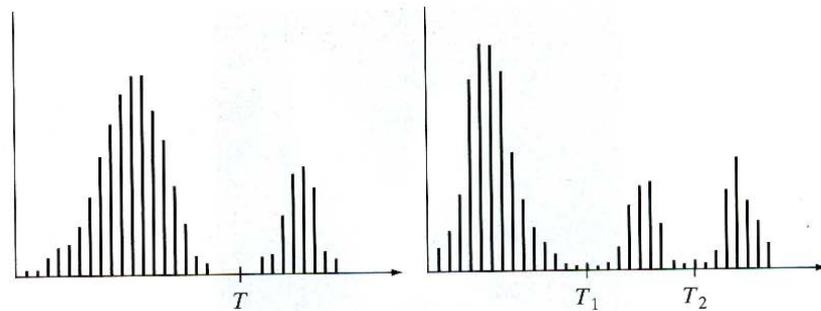


Figura 3.4: Histogramas em nível de cinza que podem ser particionados em a) um único *threshold* b) múltiplos *thresholds*. Adaptado de [13].

O histograma *b*) na Figura 3.4 mostra um caso mais geral dessa abordagem, onde três modos dominantes caracterizam o histograma (como exemplo, dois objetos claros em um fundo escuro). Aqui, de acordo com Gonzalez e Woods [13], o *multilevel thresholding* classifica o ponto  $(x, y)$  como pertencente a uma classe objeto se  $T_1 < f(x, y) \leq T_2$ , a outra classe objeto se  $f(x, y) > T_2$  e ao plano de fundo se  $f(x, y) \leq T_1$ .

Baseado no que foi dito acima, o *thresholding* pode ser visto como uma operação que envolve testes sobre uma função  $T$  da forma

$$T = T[x, y, p(x, y), f(x, y)] \quad (3.3)$$

onde  $f(x, y)$  é o nível de cinza no ponto  $(x, y)$  e  $p(x, y)$  denota alguma propriedade local nesse ponto – por exemplo, a média do nível de cinza nos pixels vizinhos com o centro em  $(x, y)$ . Uma imagem limiarizada  $g(x, y)$  é definida como

$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) > T \\ 0 & \text{se } f(x, y) \leq T. \end{cases} \quad (3.4)$$

Deste modo, os pixels rotulados como 1 (ou qualquer outro nível de cinza conveniente) correspondem a objetos, enquanto pixels nomeados como 0 (ou qualquer outro nível de cinza não atribuído a objetos) correspondem ao plano de fundo. Quando  $T$  depender apenas de  $f(x, y)$ , ou seja, dos valores do nível de cinza, o limiar é chamado de *global*. Se  $T$  depende tanto de  $f(x, y)$  quanto de  $p(x, y)$ , o limiar é chamado de *local*.

## 3.5 Árvores de decisão

Breiman *et al.* [18] propuseram um algoritmo de aprendizagem de máquina usando estruturas de dados em árvores, chamadas árvores de classificação e regressão (CART). Essas árvores de decisão são uma estrutura rápida e poderosa para classificação e regressão de dados. A árvore é construída a partir de atributos de um conjunto de dados de treino previamente classificados. De acordo com esses dados, a árvore faz uma decisão em cada nó, a partir de um de seus atributos, e escolhe em qual filho ela fará a próxima decisão. Ao chegar em uma folha da árvore, o algoritmo para e tem sua classificação. A Figura 3.5 mostra um exemplo de uma árvore de decisão e seu conjunto de dados de treino.

Árvores de decisão do algoritmo CART podem trabalhar com dois tipos de dados - numéricos e categóricos. Para trabalhar com dados numéricos, utilizam-se árvores de regressão, enquanto que para dados categóricos, usam-se árvores de classificação. A principal diferença entre os dois tipos de árvore é o resultado da predição, embora algumas estruturas e parâmetros internos da árvore também mudem. O foco desse trabalho é em árvores de classificação, porém árvores de regressão também serão analisadas.

A árvore é guardada internamente numa estrutura em árvore binária, como o próprio nome indica. Cada nó da árvore usa um certo atributo dos dados e define um limiar, no caso de uma árvore de regressão, ou um conjunto de valores, no caso de uma árvore de classificação, para decidir para qual dos nós filhos a decisão vai prosseguir. A predição de um dado de entrada começa na raiz da árvore, em cada nó uma decisão é feita e o algoritmo desce para o próximo nível da árvore. A folha da árvore indica a qual classe o dado pertence, no caso de uma árvore de classificação, ou uma constante aproximando o valor do dado, no caso de uma árvore de regressão.

Um problema do algoritmo de construção da árvore é que ele pode construir uma árvore muito grande, tornando as decisões muito sensíveis a ruídos nos dados de treino. Portanto, é necessária uma condição de parada que evite isso durante a construção da árvore de decisão ou deve-se remover os galhos da árvore após sua construção. Esposito *et al.* [14] descrevem e fazem uma análise comparativa de vários métodos conhecidos para poda de árvores de decisão.

Árvores de decisão são muito úteis para aplicações de visão computacional, podendo ser usadas para segmentar imagens em regiões definidas, como mostrado por Guo *et al.* [31]. Para isso, treina-se a árvore de decisão com atributos da imagem, como atributos de cor de diferentes espaços de cores, atributos de tamanho, formato, como a razão entre altura e largura de uma região e os atributos de textura da imagem.

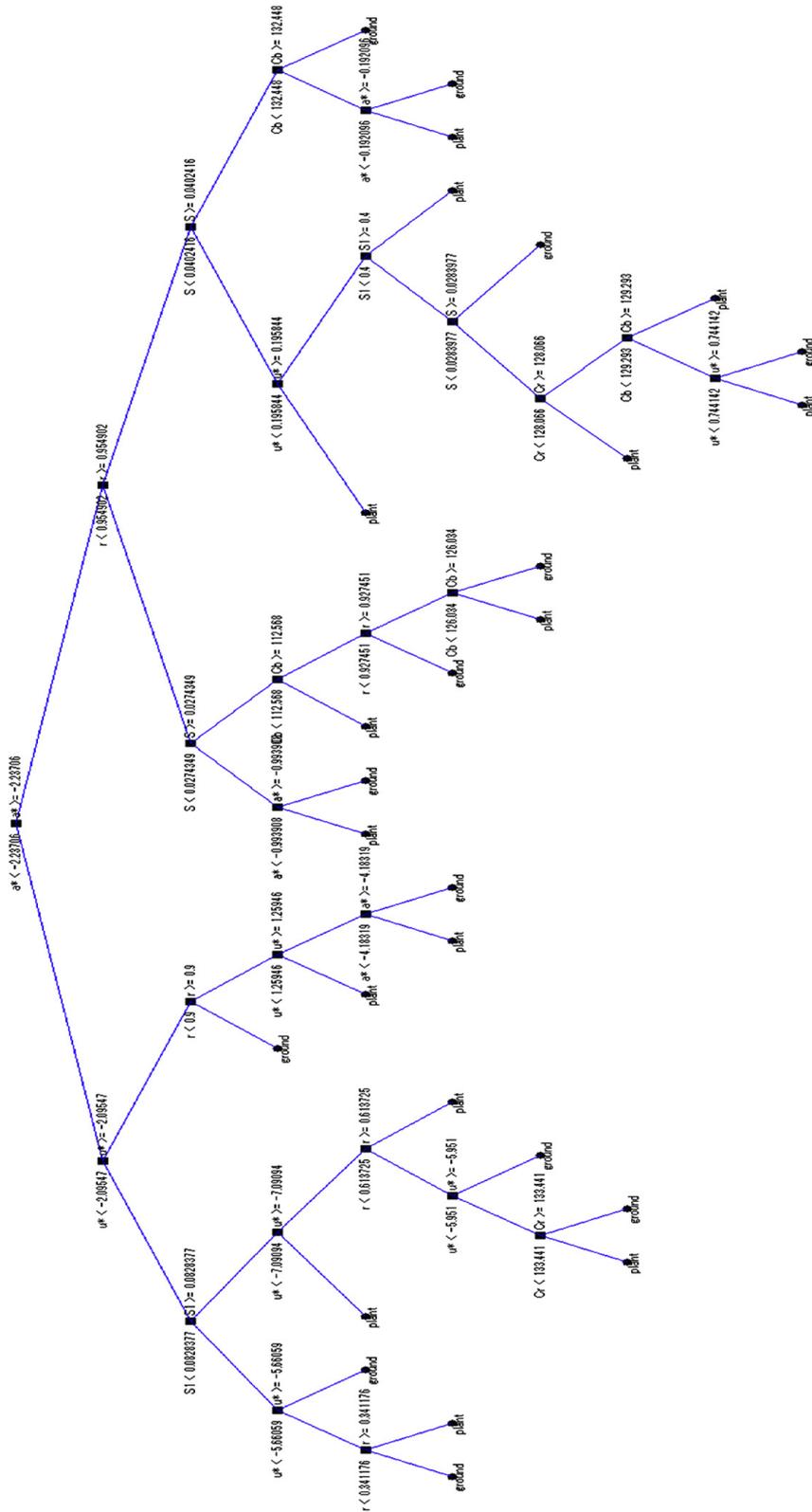


Figura 3.5: Exemplo de uma árvore de decisão para segmentação de imagens de agricultura. O nó raiz e os nós intermediários contém valores em ponto flutuante correspondentes a algum canal de cor, como o  $a^*$  do espaço  $CIE L^*a^*b^*$ , o  $u^*$  do espaço  $L^*u^*v^*$ ,  $Cb$  e  $Cr$  do espaço  $YCbCr$ . A decisão é feita a partir da comparação com o valor do nó – esquerda caso seja menor ou direita caso seja maior – e ao chegar ao nó folha o valor a ser classificado é definido como pertencente à planta ou ao solo. Adaptado de Guo *et al.* [31].

### 3.5.1 Estrutura interna e uso

A seguir é explicado como árvores de decisão são construídas e utilizadas de acordo com o algoritmo CART.

#### Construção

Antes de iniciar o procedimento de criação da árvore de decisão, é necessário selecionar os dados de treino para o algoritmo. Em geral, esses dados são um subconjunto dos dados a serem treinados, previamente classificados manualmente ou por algum outro algoritmo. A seleção desses dados de treino é muito importante, pois afeta diretamente o desempenho do algoritmo. O ideal é que esse subconjunto de dados esteja bem distribuído entre as classes do problema.

A árvore de decisão do algoritmo CART é constituída pela raiz, nós comuns e folhas, sendo uma árvore binária, ou seja, cada nó tem no máximo dois filhos, um à esquerda e um à direita. As folhas representam o resultado da classificação ou da regressão após rodar o algoritmo de predição para uma amostra. Cada nó comum, incluindo a raiz, contém uma divisão dos dados, que define se o algoritmo de predição continuará para a esquerda ou para a direita. Para realizar essa divisão, cada nó guarda um índice indicando qual variável é usada para a decisão naquele nó e os valores dessa variável que levam a decisão de qual nó será o próximo. No caso de árvores de regressão, uma variável numérica é suficiente para realizar a decisão. Para árvores de classificação, é necessário guardar um subconjunto de valores.

Um dos parâmetros de construção da árvore permite que divisões adicionais sejam criadas para cada nó, chamadas de divisões substitutas. Cada substituto define uma decisão adicional a ser tomada. O algoritmo de construção da árvore tenta criar a melhor divisão possível, mas para casos em que há valores faltando no treino, ter divisões adicionais ajuda na predição.

Para encontrar a melhor divisão, são utilizadas métricas que calculam a homogeneidade da variável de decisão do subconjunto de dados. Essas métricas são aplicadas em cada subconjunto candidato, e os resultados são combinados com uma média, o que provê uma medida da qualidade da divisão. Rokach *et al.* [26] descrevem algumas das métricas comumente usadas.

No algoritmo CART, para árvores de classificação, a métrica usada é a impureza do Gini, uma medida de quão provável um elemento aleatório do conjunto de dados seria incorretamente classificado se ele fosse agrupado aleatoriamente de acordo com a distribuição de classes no subconjunto. Ela pode ser calculada pela soma das probabilidades de cada item a ser escolhido vezes a probabilidade de um erro na categorização desse item. Ela alcança seu mínimo zero quando todos os casos no nó caem na mesma categoria. Se  $f_i$  é a fração de itens classificados com o valor  $i$  no conjunto, com  $i \in \{1, 2, \dots, m\}$ , então a impureza do Gini é dada por:

$$I_G(f) = 1 - \sum_{i=1}^m f_i^2 \quad (3.5)$$

A árvore de classificação é construída recursivamente, começando pelo nó raiz. De acordo com a métrica, a melhor divisão é encontrada para cada nó e, se necessário, divisões substitutas são criadas. Os nós filhos à esquerda e à direita são então divididos recursivamente. A cada passo, o algoritmo de construção pode parar caso a altura da árvore tenha alcançado o valor máximo definido pelos parâmetros, o número de amostras naquele nó seja muito pequeno estatisticamente para prosseguir com a divisão do nó, todas as amostras do nó pertençam à mesma classe ou a melhor divisão não ofereça nenhum ganho significativo comparado a uma escolha aleatória.

## Podando a árvore

O algoritmo de construção da árvore costuma criar árvores muito grandes. Isso afeta o desempenho da árvore, pois ela se torna mais sensível a erros nos dados de treino (dados ruidosos). Uma solução comumente usada é podar a árvore, removendo galhos que não adicionem informação relevante ou que incrementem o erro de predição. Esposito *et al.* [14] falam mais sobre poda de árvores de decisão e descrevem vários métodos para corte dos galhos.

O método usado pelo CART é a poda de custo-complexidade. Ela opera criando várias subárvores,  $T_0, T_1, \dots, T_k$  sobre o conjunto de treino, onde  $T_0$  é a árvore original, e  $T_k$  é a árvore contendo somente a raiz. Então, uma dessas árvores é escolhida como a árvore podada, de acordo com uma estimativa das taxas de erros reais das árvores.

Cada subárvore  $T_{i+1}$  é obtida substituindo uma subárvore da árvore  $T_i$  por uma folha adequada. A subárvore escolhida é aquela que obtém o menor aumento na taxa de erro aparente por folha podada,  $\alpha$ , dado por:

$$\alpha = \frac{\varepsilon(\text{podada}(T, t), S) - \varepsilon(T, S)}{|\text{folhas}(T)| - |\text{podada}(T, t)|} \quad (3.6)$$

onde

- $\varepsilon(T, S)$  é a taxa de erro da árvore  $T$  na amostra  $S$ ,
- $|\text{folhas}(T)|$  é o número de folhas da árvore  $T$ ,
- $\text{podada}(T, t)$  denota a árvore obtida trocando o nó  $t$  por uma folha adequada.

Então, a taxa de erro de cada subárvore  $T_i$  é estimada e a melhor árvore é selecionada em concordância com o resultado obtido.

É possível ainda selecionar dois parâmetros de poda durante a criação da árvore. Um dos parâmetros é o número de validações cruzadas que será feito para selecionar a melhor subárvore. O outro indica se o algoritmo de poda usará ou não uma regra adicional chamada 1SE (1one standard error), que faz a poda ser mais rigorosa, mas isso pode levar à poda de nós essenciais e a perda de informação.

## Previendo resultados

Após a árvore ter sido construída, é possível usá-la para classificar ou regredir dados. O algoritmo de predição recebe uma amostra do conjunto de dados no mesmo formato

que os dados de treino, mas sem a classificação prévia e inicia iterando pela árvore de decisão a partir da raiz. Em cada nó, o algoritmo compara um atributo da amostra, de acordo com o índice contido no nó, com os valores definidos na construção. Se os dados são numéricos, o valor de atributo é comparado com o limiar contido no nó. Caso seja menor que esse limiar, o algoritmo continua no nó à esquerda e, caso contrário, no nó à direita. Se os dados são categóricos, o algoritmo verifica se o valor está contido no subconjunto daquele nó, indo para a esquerda no caso afirmativo e para a direita caso contrário. Caso o atributo daquele nó esteja faltando ou seja desconhecido para a amostra, o teste será realizado novamente com a próxima divisão substituta ou, caso não hajam divisões substitutas, o próximo nó escolhido é o que estiver com o maior número de amostras.

Quando o algoritmo chega em um nó folha, o resultado contido nele é atribuído à amostra. Para árvores de classificação, esse resultado é um índice referente à classe à qual a amostra pertence, e para árvores de regressão uma constante que aproxima seu valor.

### 3.6 Transformação *Watershed*

Li [19], citado por Mustaqem *et al.* [4] menciona que a segmentação por watershed é um dos melhores métodos para agrupar os pixels de uma imagem baseado em suas intensidades, e de acordo com Kanitkar *et al.* [28] esse conceito é bem conhecido em topografia, sendo usado para extrair o mínimo valor de uma região presente em uma imagem e determinar o correspondente à linha divisória com o menor valor. Essa linha fornece uma rápida mudança nas fronteiras e faz com que essa transformação segmente depressões e linhas de borda em uma imagem.

Li [19] explica que pixels com intensidades similares são agrupados e Kanitkar *et al.* [28] detalham a transformação ao citar que o watershed trata a imagem como um plano, onde pixels claros tem valores altos e pixels escuros tem valores baixos. O watershed é uma ferramenta de operação matemática e morfológica normalmente utilizada para checar as saídas resultantes ao invés de ser aplicada como uma técnica de segmentação nas entradas porque ela possui a desvantagem de sempre sofrer com a pouca ou muita segmentação de uma imagem. Essa opinião também é compartilhada por Kanitkar *et al.* [28], que lidaram com os inconvenientes da técnica ao segmentar imagens baseando-se em marcadores e efetuar operações morfológicas de abertura e fechamento por reconstrução para limpar os ruídos e obter a versão segmentada final da imagem.

Em seu trabalho com detecção de tumores cerebrais, Mustaqem *et al.* [4] descrevem que por usar a transformação watershed diferentes métodos são utilizados. Dos métodos básicos, dois deles são dados a seguir: 1) o mínimo local computado na imagem de gradiente é escolhido como um marcador. Neste método ocorre uma supra segmentação e após a escolha do marcador o próximo passo é a junção de regiões; 2) a transformação watershed com os marcadores faz uso das posições definidas nesses marcadores. Esses pontos podem ser tanto definidas explicitamente por um usuário quanto determinadas automaticamente utilizando ferramentas morfológicas. Seguindo esses métodos, a técnica de watershed facilitou o destaque das regiões com o tumor e o uso eficiente das operações morfológicas. Kanitkar *et al.* [28] realizaram um trabalho com detecção de tumores

pulmonares utilizando a técnica de watershed controlada por marcadores e os resultados mostraram que essa segmentação apresenta uma performance melhor que a segmentação por limiares e demais algoritmos existentes, identificando com precisão a região de tumor no pulmão e fazendo com que o estágio do câncer seja classificado corretamente.

### 3.7 Transformação de distância

Gao e Lam [12] mencionam em seu trabalho de detecção de saliências com segmentação que a ideia básica desta transformação é calcular a distância para a mais borda próxima: o valor de um pixel em particular no domínio de transformação é a sua distância para a borda mais próxima quanto maior for seu valor de transformação. Geralmente, regiões salientes terão um forte contraste na intensidade ou transição de cores, logo uma região proeminente usualmente exibirá uma alta densidade nas bordas. Como a transformação de distância é uma medida da intensidade de bordas, ela permite a correlação com a detecção de saliências. Com o intuito de gerar um mapa de saliências usando a transformação de distâncias, os autores citam que o primeiro passo é aplicar os operadores de bordas, como os operadores de Sobel ou Canny, para produzir um mapa de bordas e levadas para a transformação de distância.

No entanto, a transformação usual é definida sobre imagens binárias em vez de imagens em escala de cinza ou imagens coloridas e por isso é necessário aplicar limiares nas imagens para formar as contrapartidas das imagens binárias. Gao e Lam [12] usam os limiares a partir de um valor intermediário para eliminar alarmes falsos, e o mapa de saliências final é obtido por simplesmente complementar o mapa de saliências inicial ou por reverter o sinal ao aplicar os limiares, já que um valor pequeno indica mais saliências, enquanto que um valor grande indica menos saliências.

A versão simples da transformação funciona bem para imagens com fundo homogêneo; imagens com cenas complexas ou com texturas mais elaboradas trazem resultados não satisfatórios, já que regiões com alta densidade de bordas nem sempre são salientes e por essa razão mais de uma técnica de segmentação é implementada em conjunto com a de transformação de distância. Gao e Lam [12] apresentam em seu trabalho uma comparação da performance entre os métodos de detecção de saliência existentes e seu modelo proposto que usa a Diferença de Gaussiana (DoG) adaptativa e a transformação de distância.

Em seu trabalho sobre métricas para imagens binárias, Baddeley [6], citado por Bouchot [11], introduz a noção dos mapas de distância locais considerando dois conjuntos abertos bem delimitados,  $A$  e  $B$ , em um espaço  $(\mathbb{R}^n, d)$ .

*Definição 1:* (Transformação de distância para o primeiro plano). A transformação de distância (TD) é definida como:

$$\forall x \in \mathbb{R}^n, TD_d[A](x) = d(x, A) = \min_{y \in A} (d(x, y)) \quad (3.7)$$

onde

- $x$  é o ponto pertencente ao espaço  $n$ -dimensional  $\mathbb{R}^n$ ;
- $A$  é um conjunto aberto bem delimitado no espaço  $\mathbb{R}^n$ ;

- $y$  é o ponto pertencente ao conjunto  $A$ ;
- $d(x, A)$  é a distância entre o ponto  $x$  e o conjunto  $A$ ;
- $\min_{y \in A}(d(x, y))$  é a distância mínima entre o ponto  $x$  e o ponto  $y$ .

*Observação 1:* o valor resultante da transformação de distância pode ser infinito para  $x \rightarrow \pm\infty$ . Bouchot [11] acrescenta que o domínio da definição de TD pode ser restringido para um domínio delimitado  $\Omega$  contendo  $A$ .

*Observação 2:* geralmente essa definição é chamada de transformação de distância para o primeiro plano, conforme explicitada acima. Algo similar, com a transformação de distância para o plano de fundo pode ser definido, onde o valor de TD para o fundo é finito o tanto quanto o conjunto  $A$  for delimitado.

A partir das observações, fica fácil definir a transformação de distância para funções binárias (i.e, funções indicadoras), enquanto essas funções tiverem o suporte de um domínio delimitado e considerarem a transformação de distância de seu suporte. Podemos escrever, de forma equivalente,

$$TD_d[A] = TD_d[\chi_A], \quad (3.8)$$

onde  $\chi_A$  representa a função indicadora do conjunto.

### 3.8 Métrica de precisão *F-measure*

Gao e Lam [12] mencionam que a *F-measure* é um indicador de performance global da precisão de um teste ou algoritmo. In [25] é proposto o uso da *F-measure* como uma média harmônica ponderada que combina outros dois parâmetros: a Precisão (*Precision*) e a Revocação (*Recall*). Em seu trabalho para detecção de saliências, os autores descrevem que a *Precision* mede a porcentagem dos pixels salientes verdadeiros detectados com relação ao número total de pixels com saliência nos *ground-truths*. O *Recall* é usado para medir a porcentagem de pixels salientes corretos em relação ao número total de pixels salientes nos *ground-truths*. Pela eficácia que apresentou, ela se tornou uma popular medida de performance em problemas de classificação binária onde uma classe é correspondente a outra.

Chinta [22], em seu trabalho de otimização de métricas, avalia a performance do problema binário de classificação com a matriz de confusão na Tabela 3.1, onde TP, TN, FP e FN são as métricas de contagem *True Positive*, *True Negative*, *False Positive* e *False Negative*, respectivamente.

Tabela 3.1: Matriz de confusão para avaliar a qualidade de uma classificação binária.

		Reconhecida como	
		Positiva	Negativa
Classe	Positiva	TP	TN
	Negativa	FP	FN

Como as classes de interesse em problemas de classificação binária são as positivas, os critérios de *Precision* e *Recall* são escritos utilizando os valores positivos dessa matriz como

$$\begin{aligned} Precision &= \frac{TP}{TP + FP} \text{ e} \\ Recall &= \frac{TP}{TP + FN}, \end{aligned}$$

e a F-measure é expressa como

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

# Capítulo 4

## Metodologia Proposta

O método para a contagem de frutos em imagens proposto neste trabalho utiliza as técnicas de processamento de imagens e visão computacional descritas nos capítulos anteriores. A Figura 4.1 mostra um fluxograma do algoritmo desenvolvido para a realização da proposta.

As próximas seções descrevem em detalhes cada técnica utilizada e explica como foram criadas as bases de imagens de frutos para teste do algoritmo.

### 4.1 Segmentação com árvores de decisão

Para que a contagem de frutos possa ser feita na imagem, inicialmente é realizada uma classificação da imagem para separar as regiões que contém frutos das regiões que não contém frutos. Essa classificação é feita pixel a pixel usando características de cor da imagem, resultando em uma classificação binária com uma classe indicando que a região contém frutos e outra indicando que a região não contém frutos.

Guo *et al.* [31] propuseram um algoritmo de segmentação de imagens de agricultura utilizando árvores de decisão, explicada em detalhes na Seção 3.5. Este algoritmo classifica imagens a partir de características de cor, pixel a pixel, em um número de classes previamente definido. As características de cores extraídas da imagem são os valores dos canais de cores de vários espaços de cores, como o *RGB*, *HSV*, *CIE L\*a\*b*, entre outros.

Este algoritmo de aprendizado de máquina é supervisionado e isso significa que ele deve ser treinado antes de realizar a classificação. O treinamento é feito em um subconjunto de imagens da base de teste usando suas características de cores e ao final é gerada uma árvore binária de classificação.

A metodologia proposta neste trabalho segue o algoritmo proposto por Guo *et al.* [31], com alguns ajustes. Os espaços de cor escolhidos para extrair as características de cor das imagens foram o *RGB* normal, o *CIE L\*a\*b* e o *RGB* após realizar uma equalização de

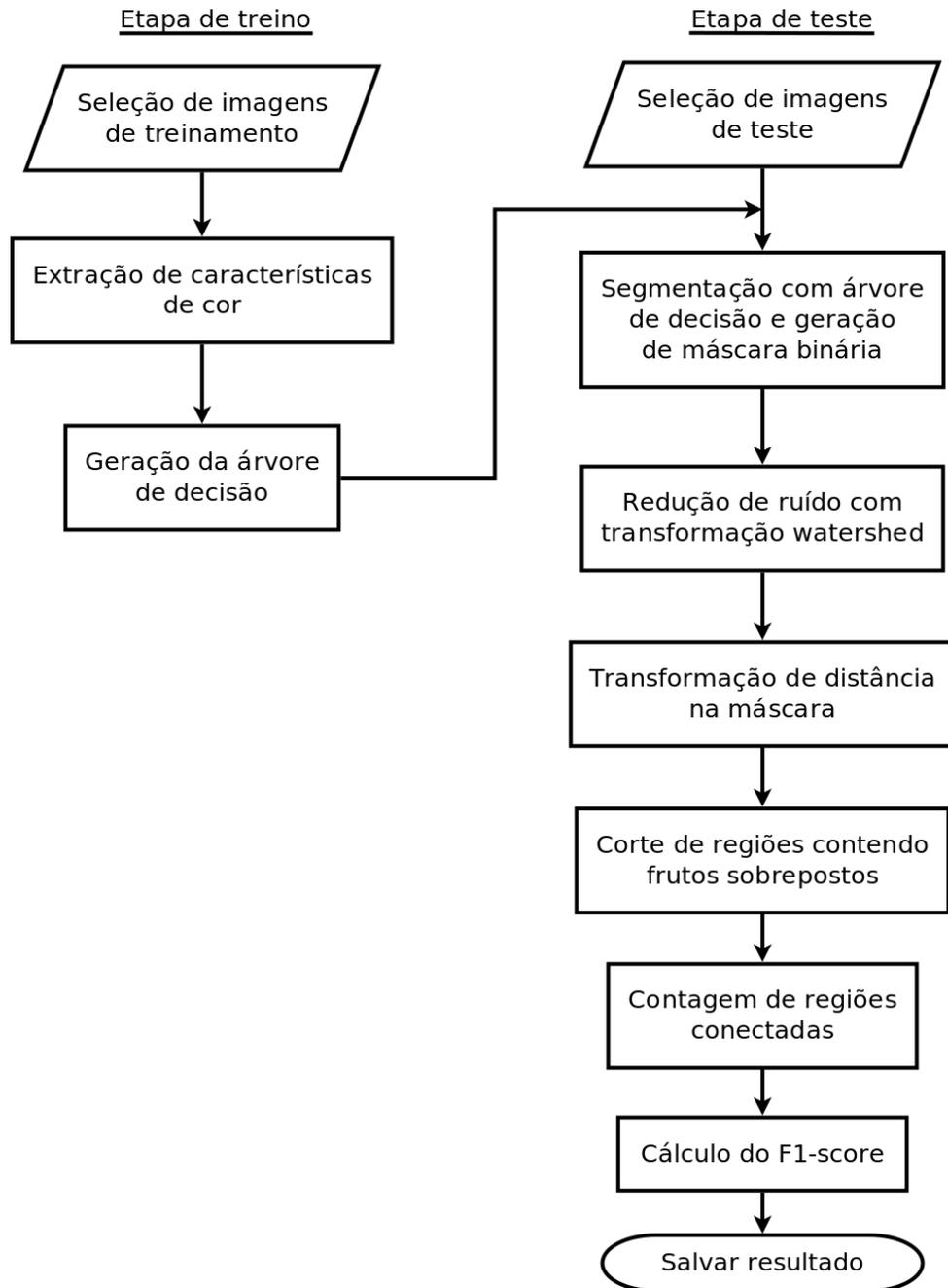


Figura 4.1: Fluxograma do algoritmo para contagem de frutos em imagens.

histograma. Após a classificação, o método retorna uma máscara binária, representando as duas classes do problema.

#### 4.1.1 Extração de características de cor

O treinamento da árvore de decisão é feito extraíndo características de cor de um subconjunto da base de imagens. As cores de interesse - no caso, as da fruta - serão

consideradas como parte de uma classe e cores referentes às folhas, caule, solo, objetos ou demais detalhes em uma foto farão parte de uma segunda classe. O usuário seleciona regiões de pixels nessas imagens de treino, informando a qual das duas classes aquela região de pixels pertence.

A Figura 4.2 mostra uma seleção de regiões realizada em uma imagem da base. O usuário pode mover o quadrado vermelho pela imagem, selecionando as regiões que deseja marcar e usando uma *trackbar* para selecionar a qual classe aquela região pertence. Cada região já marcada pelo usuário se torna azul para referência.

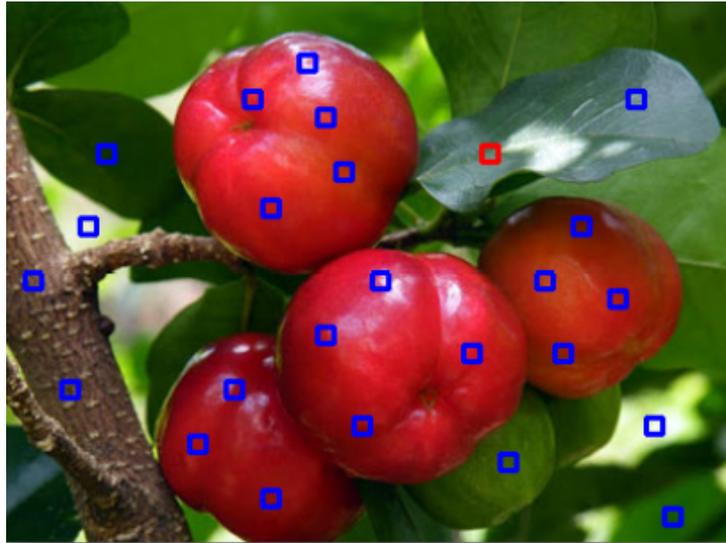


Figura 4.2: Seleção de regiões de interesse para treinamento do algoritmo. O quadrado em vermelho indica a localização do seletor de regiões e cada região marcada em azul terá seus atributos de cores extraídas para criar a árvore de decisão.

Cada pixel selecionado tem seus atributos de cor salvos em uma linha de uma matriz. Esses atributos são os valores de 9 canais de cores - todos os canais dos espaços de cores *RGB*, *CIE L\*a\*b* e *RGB* após realizada uma equalização de histograma, explicada na Seção 3.2. Uma segunda matriz, contendo o mesmo número de linhas da primeira, guarda o índice da classe a qual pertence o pixel na mesma linha da primeira matriz.

### 4.1.2 Geração da árvore de decisão

As duas matrizes contendo as informações sobre cada pixel selecionado são usadas para criar a árvore de decisão de acordo com o método do algoritmo CART explicado na Seção 3.5. O algoritmo faz uma ordenação dos atributos de cor para descobrir qual deles melhor separa as classes definidas. Esse atributo será usado para a seleção na raiz da árvore. A partir daí, cada novo nó filho faz uma ordenação dos atributos que melhor separam as classes no subconjunto que já foi separado pelo nó pai. Assim, a árvore será construída até que o ganho de informação pela separação não seja significativo. Após isso, é feita uma poda da árvore para remover nós com seleções que podem confundir o algoritmo de classificação ao levarem em conta ruído nos dados de entrada.

### 4.1.3 Segmentação e geração de máscara

Com a árvore de decisão criada, é possível segmentar as imagens de teste. Cada pixel da imagem de teste é comparado com os nós da árvore de decisão, a começar pela raiz. Compara-se o valor de uma das características de cor com um subconjunto de valores denotado pelo nó e o algoritmo desce para o nó filho à esquerda ou à direita de acordo com essa comparação. Ao chegar numa folha, o algoritmo obtém a classificação do pixel.

Ao classificar cada pixel, uma máscara é preenchida com cor branca em pixels cujo correspondente na imagem original foi classificado como fruto, e com cor preta caso contrário. A Figura 4.3 mostra um exemplo de uma imagem de acerola sendo segmentada pelo método descrito.



Figura 4.3: Exemplo de uma imagem de acerola segmentada com árvores de decisão. (a) Imagem original. (b) Máscara binária gerada pelo método de segmentação. A cor branca indica a classe "contém fruto" e a cor preta indica a classe "não contém fruto".

Este método de segmentação é robusto o suficiente para segmentar bem classes com alguma variação de cor, conforme acontece em frutos testados como o caju e o pêsego, e também possui um grau de invariância à iluminação solar, falhando somente em casos em que há extrema superexposição de uma região, como é possível ver nos pequenos buracos pretos dentro da máscara da Figura 4.3b.

## 4.2 Redução de ruído com transformação *Watershed*

É possível perceber algum ruído na máscara de segmentação na Figura 4.3b. Esse ruído pode ser removido com o uso da transformação *Watershed*, detalhada na Seção 3.6. Primeiramente, é realizada uma erosão e uma dilatação separadamente na máscara, com cada transformação sendo salva em uma matriz diferente. Na matriz com a dilatação da máscara realiza-se uma limiarização para atribuir a todos os pixels com valor menor que 1 o valor 128, cinza. A transformação *Watershed* é aplicada na soma das duas matrizes. Outra limiarização transforma os pixels na cor cinza em pixels brancos. O resultado é uma

máscara com menos ruído que a original, como demonstra a Figura 4.4 correspondendo ao resultado para a máscara da Figura 4.3b.

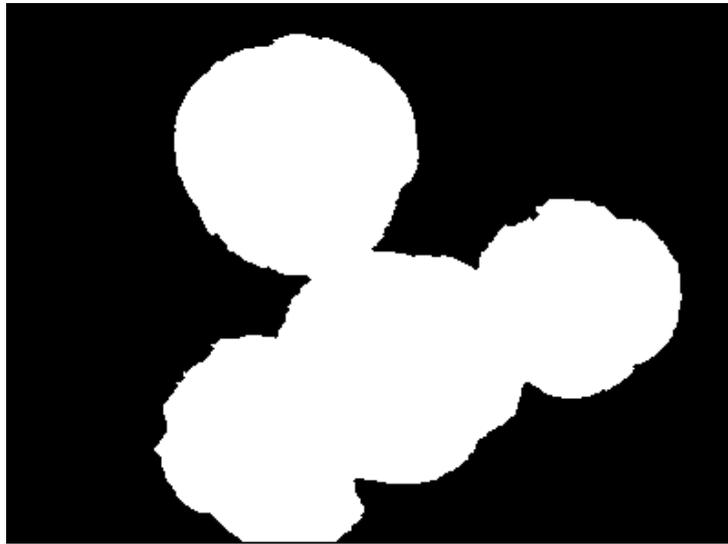


Figura 4.4: Máscara binária de segmentação após redução de ruído com *Watershed*.

### 4.3 Transformação de distância

Com a máscara binária criada, temos informação sobre onde estão os frutos na imagem e é possível contar o número de regiões conectadas para estimar de maneira grosseira o número de frutos. No entanto, essa máscara não fornece informações sobre possíveis sobreposições de frutos na imagem.

Em alguns casos, a sobreposição pode ser pequena o suficiente para que uma transformação de distância na máscara possa separar estes frutos. A transformação de distância, detalhada na Seção 3.7, reduz o tamanho das regiões brancas na máscara e quando essas regiões possuem um gargalo, que pode indicar uma sobreposição pequena de frutos, esse gargalo fica mais difuso. Com uma limiarização para tornar a máscara binária novamente o gargalo desaparece, deixando duas regiões em seu lugar. A limiarização é feita com um limiar de 0.4 após uma normalização dos valores retornados pela transformação de distância para valores entre 0 e 1.

A transformação nem sempre oferece um ganho experimental real (ou significativo) na contagem de frutos. Para cada base de frutos, foram realizados testes que usavam a técnica e outros que não usavam a transformação de distância. Em alguns casos, a transformação de distância leva a métrica de precisão a aumentar consideravelmente, e em outros casos fica claro que a ausência da transformação oferece os melhores resultados. Isso pode ocorrer por vários motivos que serão discutidos no Capítulo 5. No entanto, ainda existem claras vantagens de se utilizar essa técnica em algumas bases. Logo, a realização da transformação de distância é opcional (dependendo da base em questão) e por este

motivo todos os testes da metodologia foram feitos tanto utilizando a técnica como não a utilizando.

A Figura 4.5 mostra a mesma máscara da Figura 4.4 após sofrer a transformação de distância. É possível perceber que uma nova região conectada aparece, correspondendo a um fruto sobreposto a outros na imagem original, a Figura 4.3a. No entanto, os outros três frutos sobrepostos ainda correspondem a somente uma região conectada.

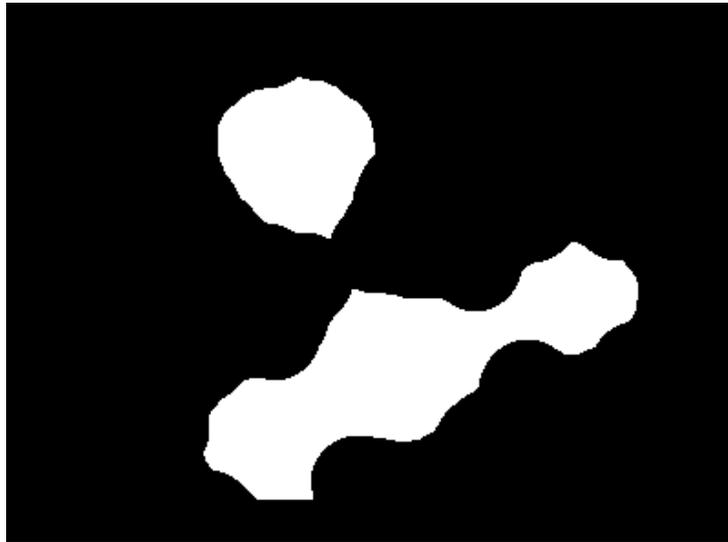


Figura 4.5: Máscara binária da Figura 4.4 após realização da transformação de distância. A transformação de distância é aplicada, realiza-se uma normalização da máscara para valores de 0 a 1 e então realiza-se uma limiarização com limiar de 0.4.

## 4.4 Corte de regiões com frutos sobrepostos

Em diversos casos, decorrem problemas em que os frutos não se separam devido à sobreposição no momento da captura. Nem todos eles são separados pela transformação de distância. Para isso, foi criado um método para detectar as regiões conectadas na máscara que contém frutos sobrepostos e separá-las. A Figura 4.6 mostra o fluxograma do algoritmo para corte de regiões.

Para todas as regiões conectadas da máscara, tenta-se encontrar pontos onde possivelmente há uma transição de um fruto para outro. Ao encontrar tal ponto, essa região é separada em duas e o algoritmo tenta encontrar novamente um ponto de corte nas duas regiões novas. Os pontos de transição entre frutos são encontrados a partir de uma análise da variação da luminância ou da cor na imagem. Quando a variação de luminância ou de cor é maior que um limiar estabelecido, aquele ponto é um candidato a um ponto de corte.

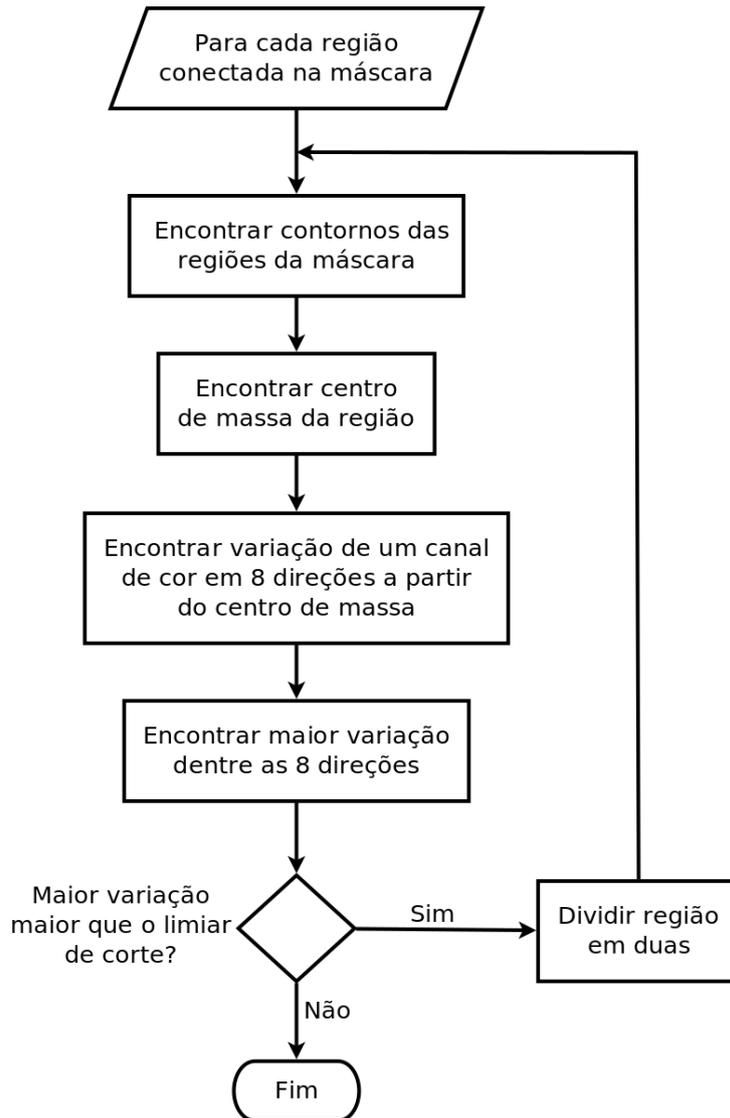


Figura 4.6: Fluxograma do algoritmo de corte de regiões conectadas que contém frutos sobrepostos.

#### 4.4.1 Encontrar contornos das regiões

A técnica para encontrar contornos das regiões conectadas da máscara, explicada na Subseção 3.3.1, é utilizada para obter as características de cada região conectada na máscara e permitir a obtenção do seu centro de massa. Utiliza-se o método que retorna somente os contornos externos da região, pois em geral os frutos não possuem buracos e a existência de espaços vazios na região deve indicar um erro no processo de segmentação.

#### 4.4.2 Encontrar centro de massa da região

A técnica mostrada na Subseção 3.3.2, que utiliza momentos de imagens bidimensionais, é usada para encontrar o centro de massa, a partir do qual é feita a análise da

derivada de um canal de cor em oito direções. A Figura 4.7 mostra a máscara anterior com o seu centro de massa destacado em vermelho.



Figura 4.7: Máscara binária anterior com o centro de massa da região destacado em vermelho.

#### 4.4.3 Encontrar variação de cor com conectividade oito

A partir do centro de massa, procura-se em oito direções (conectividade), sendo estas as direções verticais e horizontais positivas e negativas (bem como as diagonais entre elas) os valores de um canal de cor que serão salvos em vetores. Os canais de cores utilizados são os três canais do espaço de cor *CIE L\*a\*b*, onde o canal *L* corresponde à luminância e os canais *a* e *b* possuem informação de cor. Os vetores são preenchidos com os valores dos canais de cor da imagem original até que uma das direções atinja um ponto fora da região na máscara ou até que uma das bordas da imagem seja atingida. A Figura 4.8 ilustra essas oito retas a partir do centro de massa que serão analisadas.

Após encontrar todos os valores necessários para preencher os vetores, calcula-se a variação desses valores, que é um cálculo de derivada de função em um conjunto de valores discretos. Na prática, esse cálculo é uma simples subtração de um valor de cada vetor pelo valor anterior. Assim, obtém-se um novo vetor para cada direção com tamanho igual ao tamanho do vetor de valores originais da mesma direção menos um.

#### 4.4.4 Corte da região

O ponto de maior variação dentre os oito vetores de variação das oito direções conectadas é encontrado. Assim, garante-se que a região será cortada em no máximo um ponto para cada iteração do algoritmo. O ponto é considerado um ponto de transição entre frutos se a variação do canal de cor naquele ponto for maior que um limiar.

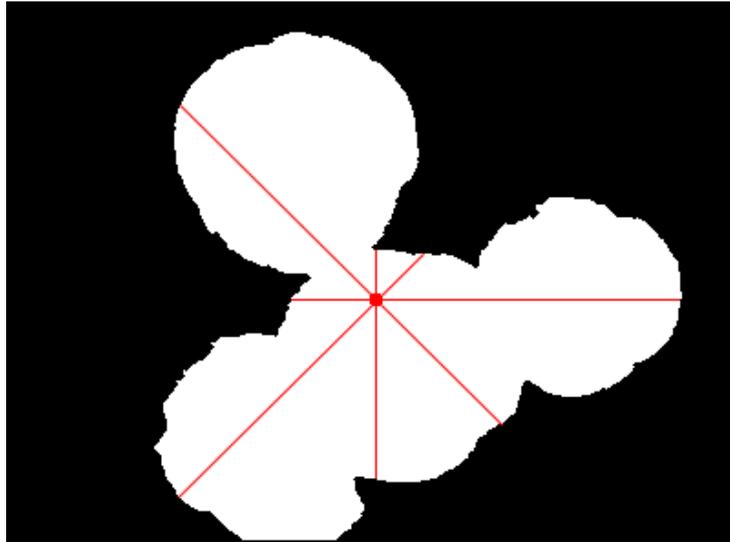


Figura 4.8: Imagem ilustrando os oito segmentos de reta na imagem onde será feita a análise de variação de cor para o corte de região.

Caso o ponto com maior variação não seja maior que o limiar, a região não é cortada e o algoritmo tenta novamente na próxima região conectada. Caso seja maior que o limiar, é dado início ao processo de divisão. Desenha-se na máscara um segmento de reta com a cor preta. Essa reta passa pelo ponto de corte, é perpendicular à direção na qual foi encontrado o ponto e está contida dentro da região conectada para que não corte outras regiões próximas. Com a divisão feita, o algoritmo executa novamente em cima das duas regiões novas, para tentar encontrar mais frutos sobrepostos na mesma região original. A Figura 4.9 ilustra uma possível divisão que o algoritmo pode realizar na máscara vista anteriormente.



Figura 4.9: Exemplo ilustrativo de corte realizado na máscara binária vista anteriormente.

Para determinar o limiar, foram realizados testes com valores variando entre 10 e 100, pois verificou-se que a maioria dos valores de variação do canal ficavam entre esses valores. Assim, é possível determinar um valor de limiar ótimo para cada base.

## 4.5 Contagem de regiões conectadas

Após o método de corte de regiões acima, obtém-se uma máscara com um número de regiões conectadas aproximadamente igual ao número de frutos da imagem. O método de encontrar contornos das regiões pode ser usado para contar o número de regiões encontradas na máscara. Assim, obtém-se o número de frutos reconhecidos em uma imagem.

O número de frutos encontrado pode ser comparado com as anotações, o *ground truth* de cada imagem, permitindo assim contar o número de frutos legítimos encontrados na imagem (*True Positives*), os frutos que não existem na imagem e foram contados pelo método (*False Positives*) e os frutos presentes na imagem e que não foram contados pelo programa (*False Negatives*). Verdadeiros negativos (*True Negatives*) não serão usados neste problema e os demais valores serão utilizados para o cálculo da *Precision*, do *Recall* e da *F-measure* do algoritmo. A forma como esses valores são contabilizados é melhor explicada no Capítulo 5.

# Capítulo 5

## Resultados e discussão

Este capítulo apresenta os resultados experimentais do método proposto no Capítulo 4. A Seção 5.1 explica como a base de imagens de teste foi construída. A Seção 5.2 detalha a forma de realização dos testes experimentais. A Seção 5.3 mostra os resultados em forma de tabelas e gráficos. E a Seção 5.4 contém uma análise e discussão dos resultados obtidos.

### 5.1 Construção das bases de imagens

Para testar o algoritmo desenvolvido, foram construídas várias bases de imagens, cada uma contendo imagens de um fruto específico. As imagens foram retiradas da Internet, com o único critério de que as fotos sejam capturadas em condições naturais no campo e contenham poucos objetos que não sejam as plantas e possivelmente outras características comuns nesta situação, como por exemplo o céu. Assim, as bases possuem imagens de tamanhos e resoluções variados, de forma a testar a independência da solução proposta à esta característica. Esta característica foi definida como um critério que torna a técnica proposta independente do modelo de câmera de captura, incluindo aquelas de mais baixa resolução. Procurou-se também adquirir somente imagens com o formato .jpeg ou .jpg. Cada base possui entre aproximadamente 50 e 100 imagens, nomeadas com uma ou duas letras referentes ao nome da fruta e um número para a contagem.

Além das imagens em si, cada base possui uma pasta correspondente contendo arquivos de texto, um para cada imagem, com o mesmo nome da imagem correspondente e extensão .txt. Esses arquivos contém um número de anotação da imagem, o *ground truth*, representando o número real de frutos contidos na imagem correspondente. O número de frutos da imagem foi contado manualmente para cada imagem e desenvolvido pelo membros do projeto. Essas anotações são usadas para calcular a métrica de precisão e analisar os resultados.



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)

Figura 5.1: Exemplos de imagens das bases de imagens de frutas coletadas da internet. (a) Acerola. (b) Ameixa. (c) Caju. (d) Cereja. (e) Laranja. (f) Manga Rosa. (g) Morango. (h) Pêssego.

Tabela 5.1: Informações sobre as bases de imagens de frutos utilizadas para os testes.

Base de frutos	Quantidade de imagens	Tamanho da maior imagem	Tamanho da menor imagem
Acerola	113	4288 x 3216	150 x 137
Ameixa	60	3488 x 2616	275 x 150
Caju	65	2848 x 4272	236 x 157
Cereja	47	3648 x 2053	200 x 160
Laranja	108	2592 x 1936	135 x 100
Manga Rosa	38	2304 x 3072	134 x 184
Morango	114	3888 x 2592	250 x 187
Pêssego	34	3000 x 2250	180 x 134

Na Figura 5.1 são apresentados uma amostra de cada uma das imagens que formam o banco de dados utilizado para os testes da metodologia proposta. A Tabela 5.1 mostra a quantidade de imagens e o tamanho da maior e da menor imagem de cada uma das bases.

## 5.2 Realização dos testes

Os testes foram realizados em todas as oito bases de imagens de frutos. Para cada base, foram seguidos dois critérios: um treinamento com 30% das imagens da base e os testes realizados em cima dos outros 70%, e também foram realizados testes com treinamento usando 70% das imagens e execução feita sobre os outros 30%.

Três parâmetros do algoritmo são alterados para a realização dos testes: a realização ou não da distância de transformação, o canal de análise da variação de cor para corte de região, podendo este ser o canal  $L$ ,  $a$  ou  $b$  do espaço de cor  $CIE L^*a^*b$ , e o limiar de variação de cor para corte de região, variando entre 10 e 100, com passos de 10 unidades cada.

O algoritmo é executado para cada base de imagens de frutos. Os resultados de cada imagem são comparados ao *ground truth*, e o número de verdadeiros positivos (TP), falsos positivos (FP) e falsos negativos (FN) é contabilizado para todas as imagens de uma base. Ao final da execução do algoritmo para uma base, esses valores são usados para o cálculo da métrica de precisão *F-measure*, explicada na Seção 3.8. Os valores da *F-measure* são mostrados em tabelas e gráficos, indicando a fruta correspondente, o canal, o limiar de corte e se a transformação de distância foi realizada.

Para cada imagem da base, compara-se o número de frutos obtidos e o valor contido nas anotações. Se esses valores são iguais, adiciona-se esse valor ao número de verdadeiros positivos. Caso o número de frutos obtidos seja maior que o número contido nas anotações, a diferença entre eles é somada ao número de falsos positivos e o *ground truth* é somado aos verdadeiros positivos. Caso o número de frutos obtidos seja menor que as anotações, a diferença entre eles é acrescentada ao número de falsos negativos e o valor obtido pelo método é somado aos verdadeiros positivos. Como não há imagens sem frutos na base, o número de falsos negativos não é contabilizado.

Tabela 5.2: Principais resultados de  $F$ -measure para cada base de imagens.

Limiar	$F$ -measures							
	Acerola (canal $L$ , sem TD)	Laranja (canal $L$ , com TD)	Cereja (canal $a$ , sem TD)	Caju (canal $a$ , com TD)	Ameixa (canal $L$ , com TD)	Morango (canal $L$ , sem TD)	Pêssego (canal $a$ , com TD)	Manga (canal $a$ , com TD)
10	0.116741	0.388919	0.231613	0.231969	0.0463065	0.0915758	0.660393	0.241117
20	0.260279	0.613146	0.557225	0.578249	0.136986	0.157907	0.610961	0.361949
30	0.433974	0.646355	0.668596	0.722408	0.369637	0.248445	0.591815	0.482972
40	0.570847	0.658141	0.665639	0.772563	0.582418	0.313268	0.583803	0.557143
50	0.640428	0.670206	0.659341	0.778182	0.71831	0.344688	0.578799	0.524345
60	0.689732	0.677801	0.657233	0.778182	0.746269	0.357548	0.578296	0.524345
70	0.691756	0.686225	0.655118	0.778182	0.763359	0.368397	0.578296	0.524345
80	0.684564	0.674315	0.655118	0.778182	0.752	0.377264	0.578296	0.524345
90	0.679827	0.663385	0.655118	0.778182	0.721311	0.377029	0.578296	0.524345
100	0.669178	0.662064	0.655118	0.778182	0.721311	0.378248	0.578296	0.524345

### 5.3 Resultados

No Apêndice A, as Tabelas A.1 a A.15 e as Figuras A.1 a A.15 contêm todas as tabelas e gráficos com os valores de  $F$ -measure resultantes dos testes do algoritmo para cada base de imagens de frutos. As tabelas contêm os valores obtidos em formato numérico e os gráficos facilitam a comparação visual dos vários resultados. Os resultados para cada base variam de acordo com a quantidade de imagens utilizadas para o treinamento da árvore de decisão, o limiar de corte, o canal de corte e a realização ou não da transformação de distância. Todas essas informações estão adequadamente indicadas nos gráficos e tabelas.

A Tabela 5.2 mostra os melhores resultados de  $F$ -measure para todas as bases de imagens de frutos, utilizando 30% da base para o treinamento e 70% para os testes. A métrica  $F$ -measure muda de acordo com as variações do limiar de corte entre 10 e 100 e os outros parâmetros, como o canal de corte e a realização ou não da transformação de distância (TD) foram fixados para cada base de acordo com os melhores resultados obtidos e são ilustrados na tabela. A Figura 5.2 ilustra o gráfico correspondente à tabela para efeitos de comparação.

### 5.4 Discussão

Os melhores resultados obtidos foram para a base de imagens de caju, com o maior valor de  $F$ -measure sendo 0.778 para o treinamento em 30% da base e 0.849 para o treinamento em 70% da base. A base com os piores resultados no geral foi a de morango, com o maior valor de  $F$ -measure chegando somente a 0.38. Somente as bases de caju e ameixa obtiveram resultados acima de 0.7 para o treinamento com 30% da base, embora as bases de acerola e laranja terem chegado bem próximas a esse resultado também. Fica clara a dificuldade de se obter resultados altos e estáveis através de bases diferentes e não controladas, repletas de detalhes de cenários ou objetos e com uma alta variação de cores nos frutos nas fases verde, madura e os que estão na transição dessas duas fases.

Principais resultados de F-measure das bases de imagens

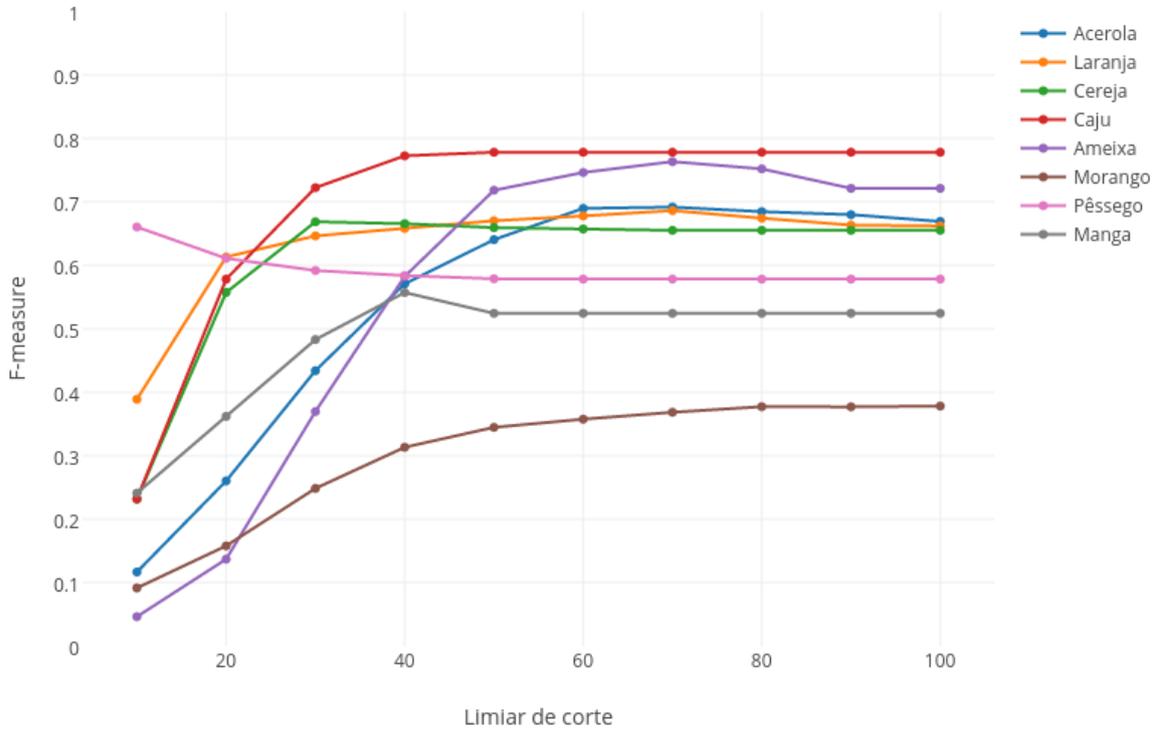


Figura 5.2: Gráfico dos principais resultados de  $F$ -measure das bases de imagens.

É possível perceber que houve uma melhoria significativa nos resultados quando o treinamento é feito com 70% das bases comparado ao treinamento em 30%. Isso é algo positivo, pois demonstra que o método reage como esperado ao melhorar o treinamento. Essa etapa de treinamento influencia altamente os resultados, como foi possível perceber experimentalmente, pois são coletados um grande volume de dados sobre as cores de cada pixel que compõe as frutas e os que fazem parte do caule, folhas e demais detalhes em uma foto. Testes iniciais feitos com um treinamento de poucas imagens forneceram resultados muito aquém do valor aceitável de 0.7 de  $F$ -measure.

O limiar de corte de região influencia bastante os resultados. Em geral, como é visto nos gráficos das Figuras 5.3a e 5.3b, um aumento do limiar leva a um aumento da  $F$ -measure até um pico claramente definido, a partir do qual os resultados diminuem. Após um certo limiar, os resultados se estabilizam e não se alteram. Isto ocorre pois, a partir deste ponto, a variação do canal de cor selecionado nunca é maior que o limiar, ou seja, a partir desse ponto o algoritmo não realiza mais o corte de regiões. Em alguns casos, o ponto de estabilização pode ser atingido antes de ocorrer um pico claramente definido. O limiar no qual o valor máximo da  $F$ -measure é atingido e o valor a partir do qual ele se estabiliza variam bastante para cada base, pois esses valores dependem de propriedades das imagens da base. Imagens de qualidade baixa, por exemplo, podem ter grande variação nos canais de cor devido aos ruído e artefatos de compressão. Propriedades intrínsecas

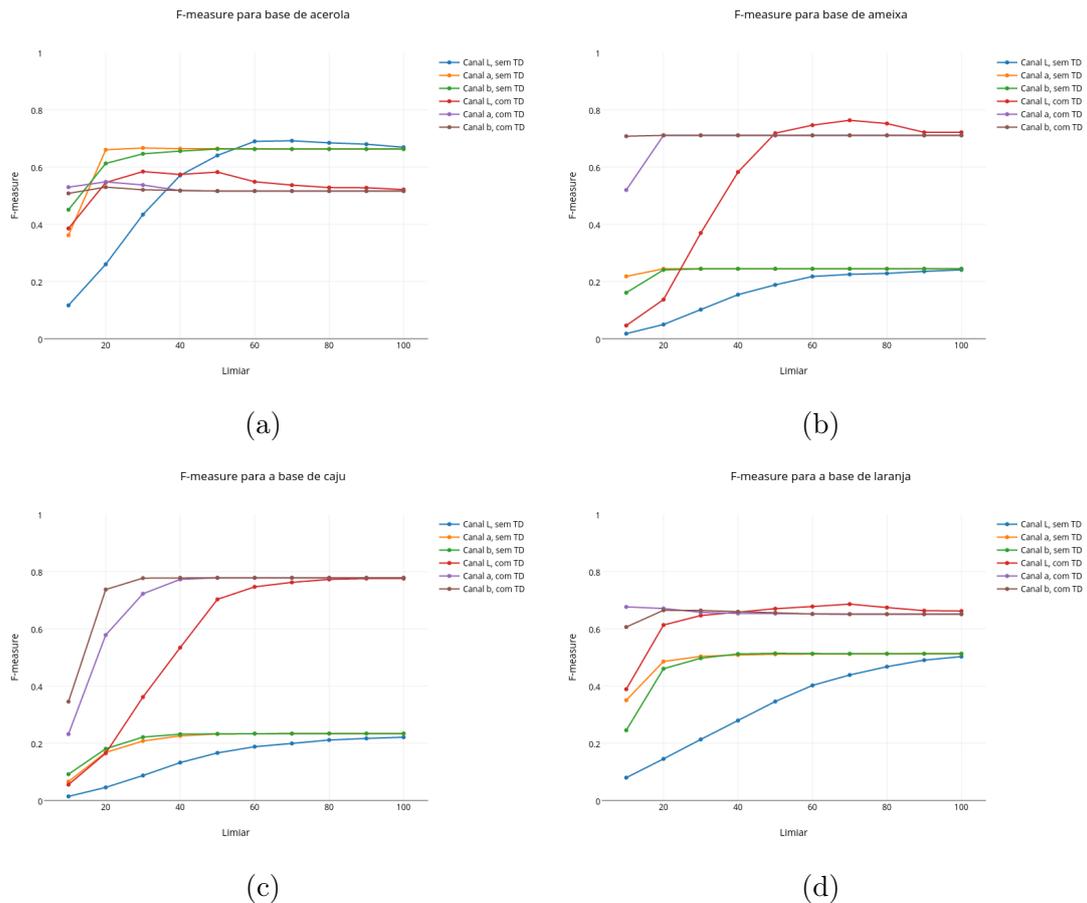


Figura 5.3: Gráficos de alguns resultados de  $F$ -measure ilustrando seu comportamento. Os resultados são das bases de (a) acerola, (b) ameixa, (c) caju e (d) laranja.

dos frutos também causam essa variação. O morango, por exemplo, possui regiões escuras dentro da região de frutas, as sementes, o que leva a uma maior variação de cor e luminância na imagem. Isso pode ser uma das causas dos valores baixos de  $F$ -measure obtidos para essa base.

A escolha de canal e a realização da transformação de distância fornecem resultados variados para cada base. Enquanto algumas bases obtêm resultados melhores realizando o corte de região no canal  $L$  do espaço de cores  $CIE L^*a^*b$ , outros conseguem melhor resultado cortando em cima do canal  $a$ . O mesmo pode ser dito sobre a realização da transformação de distância, que em alguns casos ajuda a separar as regiões de frutos e em outros pode introduzir erros na máscara. Isso pode ocorrer por propriedades das imagens da base, que foram coletadas aleatoriamente ou por propriedades dos frutos em si. Analisando os resultados completos, percebe-se que os dados para os canais  $a$  e  $b$  são similares, porém na maioria das situações o canal  $a$  possui uma pequena vantagem. Ambos esses canais modelam a variação de cor da imagem e por este motivo podem ter resultados parecidos. A vantagem do canal  $a$  pode decorrer do fato dele modelar a variação entre verde e vermelho. Os frutos que apresentam vantagens no canal  $a$  possuem coloração avermelhada - cereja, caju, pêssego e manga.

A segmentação realizada por cor impõe algumas limitações na segmentação. Frutos com muita variação de cor na superfície do fruto ou entre frutos são difíceis de serem segmentados. A base de morangos mostra isso, pois há uma grande variação de cor entre os morangos, alguns sendo vermelhos, outros tendendo mais ao rosa e ainda alguns que, por não estarem muito maduros, tendem ao verde ou ao branco. Além disso, as sementes e a alta refletância do fruto criam variação de cor na superfície do fruto, criando regiões de superexposição que não são segmentadas corretamente. A base de manga também demonstra um problema com a segmentação - o talo próximo ao fruto possui uma coloração avermelhada, o que leva o algoritmo a marcar erroneamente essas regiões como frutos.

Nas Figuras 5.3c e 5.3d, é possível ver uma grande disparidade entre os resultados em que foi realizada a transformação de distância e os resultados que não a realizaram. Neste caso específico, bem como em alguns outros, a transformação de distância melhorou bastante os resultados, justificando assim o seu uso, apesar de nem sempre essa melhoria acontecer. Isso pode acontecer pois a transformação de distância em alguns casos remove regiões incorretas na segmentação ou mesmo limita a influência de ruído nos canais de cor no momento do corte de regiões. Em alguns casos, a transformação de distância piora os resultados, como no caso da cereja. Isso pode ocorrer pois o fruto da cereja é relativamente pequeno, e a redução do tamanho das regiões que a transformação de distância faz remove regiões corretas, aumentando o erro da estimativa.

O corte de regiões de frutos sobrepostos nem sempre funciona como esperado. Variações de cor na superfície do fruto, como acontece na base de morango, levam a mudanças bruscas nos valores de cor e luminância, aumentando a chance daquela região ser dividida, mesmo que contenha apenas um fruto. A forma como a divisão é realizada não é precisa, mesmo que o ponto de corte seja escolhido corretamente. Raramente a fronteira entre frutos sobrepostos é uma reta, e isso pode ocasionar erros onde o corte é realizado novamente por uma parte de um fruto ainda estar contido na região do outro. Algo desejável para trabalhos futuros é melhorar o método de corte de regiões contendo múltiplos frutos ou encontrar um novo método para isso.

As bases de imagens utilizadas para testar a metodologia proposta contém imagens aleatórias obtidas de diferentes fontes de domínio público. Essas imagens possuem características diferentes umas das outras decorrentes dos aparelhos de captura, das condições de iluminação no momento da captura, dos ângulos diferentes em que foram capturadas e outros fatores. Tudo isso contribui para que estas sejam bases complexas e com alta variação. Isso pode ter contribuído para alguns dos baixos resultados obtidos pelo método. No entanto, resultados bons foram obtidos também, mostrando que a abordagem proposta reage bem mesmo em situações complexas e imprevisíveis, obtendo resultados consideráveis. Uma estratégia para o futuro seria testar o método em bases de imagens controladas, capturadas com um mesmo aparelho, em locais similares e possivelmente com baixa variação de condições de iluminação, para verificar melhorias nos resultados.

# Capítulo 6

## Conclusão

Foi desenvolvido um método para detecção e contagem de frutos em imagens voltado para aplicações na área de agricultura de precisão. O método segmenta os frutos da imagem e realiza uma análise de variação de um canal para dividir as regiões que possuem múltiplos frutos. Assim, o método pode contar o número de regiões conectadas na máscara final e obter o número de frutos da imagem.

O método pode ser aplicado para imagens de frutos diversos, como foi descrito e apresentado neste trabalho (testes de avaliação). O método consegue contar os frutos de imagens obtidas da Internet sem muito critério e ainda assim conseguir índices relativamente bons de precisão dependendo dos parâmetros selecionados para o teste. A métrica *F-measure* foi usada para medir a precisão do algoritmo. O método obteve resultados satisfatórios, acima de 0.7, para as classes de frutos de ameixa e caju. Em outras classes, como acerola, laranja e cereja, apesar de não ter atingido essa meta, muitas vezes os resultados chegaram próximos, passando de 0.6. Isso indica que ainda existe espaço para melhorias do algoritmo, podendo este alcançar resultados mais elevados para a maioria das bases.

Os testes para o método proposto foram realizados em bases de imagens não controladas, retiradas com o critério retratando situações reais e de domínio público em sítios *Web*. Essas imagens possuem tamanhos (em *bytes*), resoluções espaciais e tipos de compressões altamente variados. Este fator pode ter influenciado negativamente os resultados. A variação nas imagens pode causar problemas tanto na etapa de segmentação dos frutos quanto na etapa de corte de regiões de frutos sobrepostos. Entretanto se faz necessária que as características desta base de dados para comprovar a robustez e acurácia da técnica proposta.

O método de segmentação utiliza atributos de cor das imagens para segmentar os frutos e por essa razão pode falhar quando há variação de cor num mesmo fruto, entre os frutos ou quando há cores similares entre classes diferentes. A segmentação é supervisionada, requerendo um treinamento manual realizado antes da segmentação. Os resultados da segmentação dependem altamente desse treinamento. Para garantir bons resultados, é preciso selecionar regiões da imagem e classificá-las cuidadosamente, garantindo que re-

giões variadas serão incluídas no treinamento. O treinamento deve englobar o máximo de características variadas possível, por isso deve-se utilizar um número grande de imagens variadas para garantir bons resultados.

O corte de região também pode falhar quando há grande variação de cor num mesmo fruto. A análise pode perceber grande variação e cortar a região num ponto em que haja uma grande variação de cor ou luminância, como no caso de picos de reflexão especular ou mesmo em pontos com artefatos de compressão. O corte não é preciso, pois a região é dividida com um segmento de reta e isso também pode levar a erros. O formato das frutas não é levado em conta para realizar a divisão, embora isso possa levar o algoritmo a uma especialização para cada fruto diferente.

## 6.1 Perspectivas futuras

Ainda há a oportunidade de melhorar o método proposto e incluir a realização de mais testes. Primeiramente, existem parâmetros que ainda podem ser analisados com mais profundidade e critério, como parâmetros da transformação de distância e espaços de cores utilizados para extração de atributos de cor para treinamento da árvore de decisão, bem como verificando sua influência nos resultados finais.

Subsequentemente, o método pode ser aprimorado, especialmente a etapa de corte de regiões. Apesar de uma base controlada ajudar na busca de pontos de corte, o método ainda pode escolher pontos como picos de reflexão especular para o corte que não são realmente pontos de transição entre frutos sobrepostos. Uma possibilidade é encontrar vales seguidos de picos bruscos nos valores de luminância em cada uma das oito direções, ao invés de simplesmente selecionar qualquer ponto que varie mais que um limiar. A divisão em si também pode ser melhorada. Pode-se utilizar informações de direções adicionais além da direção do ponto que mais variou para selecionar um segmento de reta mais próximo da linha real de separação ou encontrar uma forma de utilizar informações de bordas da imagem, utilizando um filtro de bordas.

Existem outras formas de detectar frutos sobrepostos na imagem, algumas delas descritas na revisão bibliográfica, no Capítulo 2, ou mesmo outros procedimentos. Um método a ser considerado no início da pesquisa foi utilizar uma segmentação com superpixels. Essa forma de segmentação separaria a imagem em muitas regiões pequenas e o problema se tornaria juntar regiões contíguas ao invés de separar regiões. Assim, esse método substituiria tanto a segmentação com árvores de decisão quanto a separação de regiões de frutos sobrepostos.

# Apêndice A

## Resultados completos

Tabela A.1: Resultados de  $F$ -measure para a base de acerola, realizando treinamento com 30% e teste com 70% da base.

	Acerola					
Limiar	Sem transformação de distância			Com transformação de distância		
	Canal $L$	Canal $a$	Canal $b$	Canal $L$	Canal $a$	Canal $b$
10	0.116741	0.361829	0.450852	0.385472	0.52968	0.507937
20	0.260279	0.660321	0.612542	0.545746	0.548173	0.529813
30	0.433974	0.666667	0.646192	0.584192	0.537313	0.520499
40	0.570847	0.664146	0.655717	0.574032	0.517857	0.517457
50	0.640428	0.664541	0.662821	0.582382	0.516129	0.516129
60	0.689732	0.662821	0.662821	0.548552	0.516129	0.516129
70	0.691756	0.662821	0.662821	0.536842	0.516129	0.516129
80	0.684564	0.662821	0.662821	0.528369	0.516129	0.516129
90	0.679827	0.662821	0.662821	0.527531	0.516129	0.516129
100	0.669178	0.662821	0.662821	0.521429	0.516129	0.516129

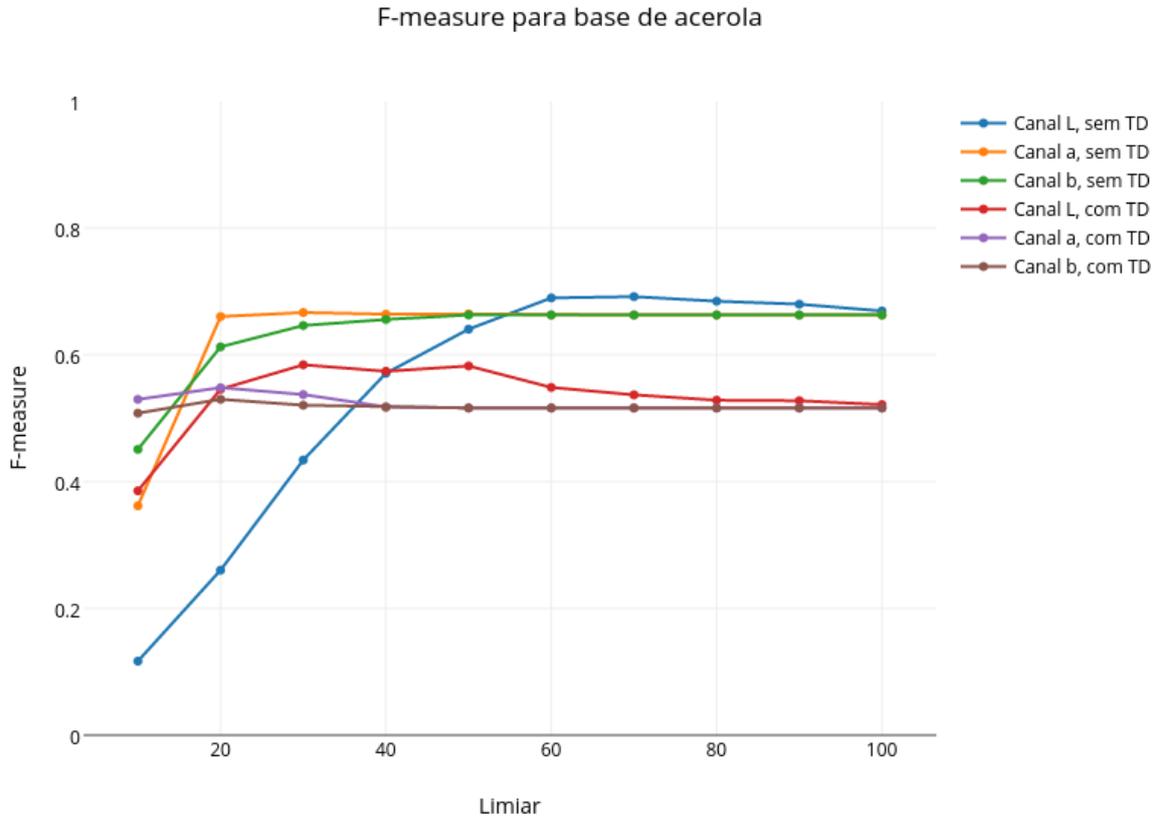


Figura A.1: Gráfico dos resultados de  $F$ -measure para a base de acerola, realizando treinamento com 30% e teste com 70% da base.

Tabela A.2: Resultados de  $F$ -measure para a base de acerola, realizando treinamento com 70% e teste com 30% da base.

Limiar	Acerola					
	Sem transformação de distância			Com transformação de distância		
	Canal $L$	Canal $a$	Canal $b$	Canal $L$	Canal $a$	Canal $b$
10	0.115758	0.354098	0.442404	0.386646	0.567568	0.575251
20	0.278173	0.651811	0.664463	0.568356	0.598214	0.615741
30	0.47846	0.704082	0.697509	0.59501	0.621027	0.593516
40	0.604819	0.694097	0.68705	0.630197	0.586466	0.586466
50	0.71137	0.691756	0.684685	0.64977	0.582915	0.582915
60	0.717781	0.684685	0.684685	0.620853	0.582915	0.582915
70	0.716007	0.684685	0.684685	0.610169	0.582915	0.582915
80	0.705263	0.684685	0.684685	0.593137	0.582915	0.582915
90	0.696113	0.684685	0.684685	0.595533	0.582915	0.582915
100	0.692857	0.684685	0.684685	0.586466	0.582915	0.582915

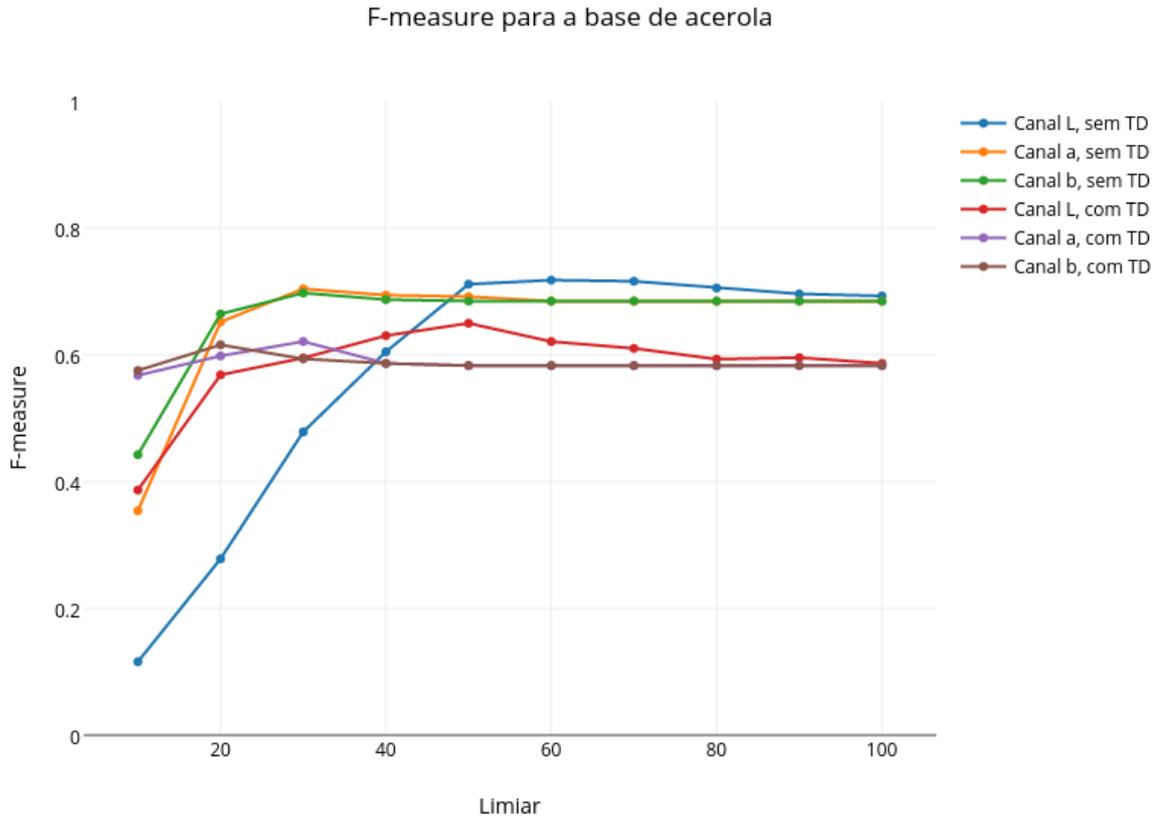


Figura A.2: Gráfico dos resultados de  $F$ -measure para a base de acerola, realizando treinamento com 70% e teste com 30% da base.

Tabela A.3: Resultados de  $F$ -measure para a base de ameixa, realizando treinamento com 30% e teste com 70% da base.

Limiar	Ameixa					
	Sem transformação de distância			Com transformação de distância		
	Canal $L$	Canal $a$	Canal $b$	Canal $L$	Canal $a$	Canal $b$
10	0.0179232	0.217993	0.16095	0.0463065	0.519774	0.707692
20	0.0499208	0.244489	0.240631	0.136986	0.710744	0.710744
30	0.102024	0.244489	0.244489	0.369637	0.710744	0.710744
40	0.154034	0.244489	0.244489	0.582418	0.710744	0.710744
50	0.188341	0.244489	0.244489	0.71831	0.710744	0.710744
60	0.217617	0.244489	0.244489	0.746269	0.710744	0.710744
70	0.225	0.244489	0.244489	0.763359	0.710744	0.710744
80	0.228464	0.244489	0.244489	0.752	0.710744	0.710744
90	0.235521	0.244489	0.244489	0.721311	0.710744	0.710744
100	0.240631	0.244489	0.244489	0.721311	0.710744	0.710744

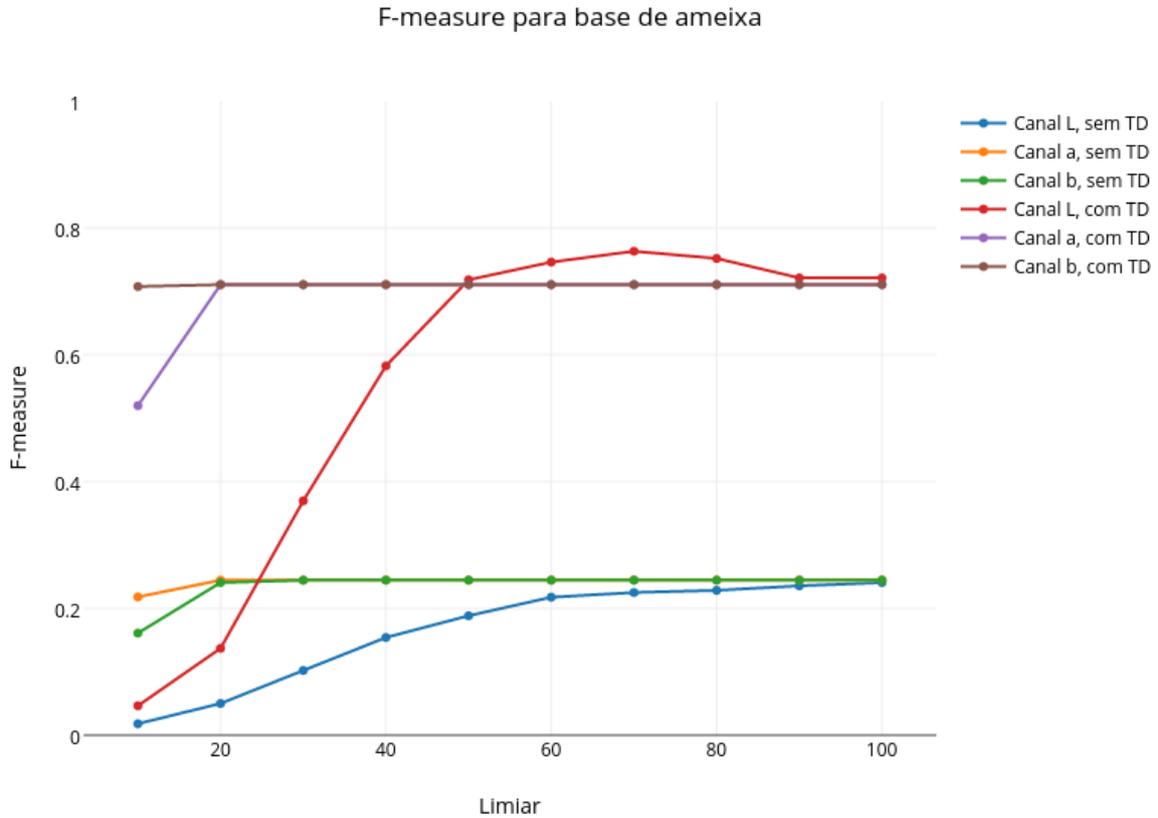


Figura A.3: Gráfico dos resultados de  $F$ -measure para a base de ameixa, realizando treinamento com 30% e teste com 70% da base.

Tabela A.4: Resultados de  $F$ -measure para a base de ameixa, realizando treinamento com 70% e teste com 30% da base.

Limiar	Ameixa					
	Sem transformação de distância			Com transformação de distância		
	Canal $L$	Canal $a$	Canal $b$	Canal $L$	Canal $a$	Canal $b$
10	0.0138097	0.190317	0.174157	0.050501	0.460784	0.625
20	0.0430328	0.22869	0.229039	0.143885	0.716667	0.716667
30	0.103194	0.231579	0.23431	0.408922	0.716667	0.716667
40	0.153846	0.231579	0.231579	0.601093	0.716667	0.716667
50	0.198738	0.231579	0.231579	0.689655	0.716667	0.716667
60	0.225	0.231579	0.231579	0.720588	0.716667	0.716667
70	0.236842	0.231579	0.231579	0.746032	0.716667	0.716667
80	0.246032	0.231579	0.231579	0.741935	0.716667	0.716667
90	0.242424	0.231579	0.231579	0.737705	0.716667	0.716667
100	0.241803	0.231579	0.231579	0.727273	0.716667	0.716667

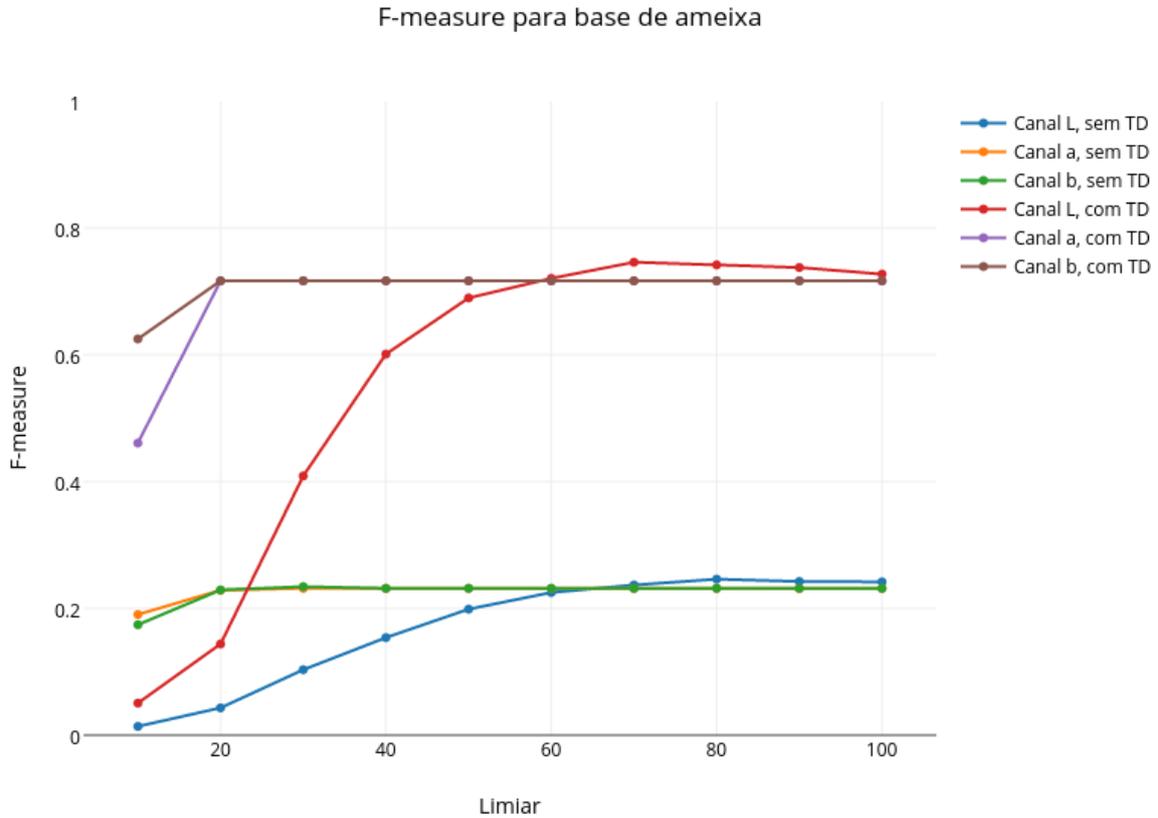


Figura A.4: Gráfico dos resultados de  $F$ -measure para a base de ameixa, realizando treinamento com 70% e teste com 30% da base.

Tabela A.5: Resultados de  $F$ -measure para a base de caju, realizando treinamento com 30% e teste com 70% da base.

Limiar	Caju					
	Sem transformação de distância			Com transformação de distância		
	Canal $L$	Canal $a$	Canal $b$	Canal $L$	Canal $a$	Canal $b$
10	0.0143829	0.0661538	0.0919871	0.0555556	0.231969	0.345566
20	0.0459608	0.168011	0.18089	0.165278	0.578249	0.737201
30	0.0874576	0.208054	0.221626	0.361635	0.722408	0.776978
40	0.132512	0.225895	0.231857	0.534279	0.772563	0.778182
50	0.166448	0.232734	0.232734	0.702875	0.778182	0.778182
60	0.18806	0.234063	0.23384	0.746575	0.778182	0.778182
70	0.199521	0.234063	0.23384	0.762238	0.778182	0.778182
80	0.211506	0.234063	0.234063	0.772563	0.778182	0.778182
90	0.216973	0.234063	0.234063	0.775362	0.778182	0.778182
100	0.221231	0.234063	0.234063	0.775362	0.778182	0.778182

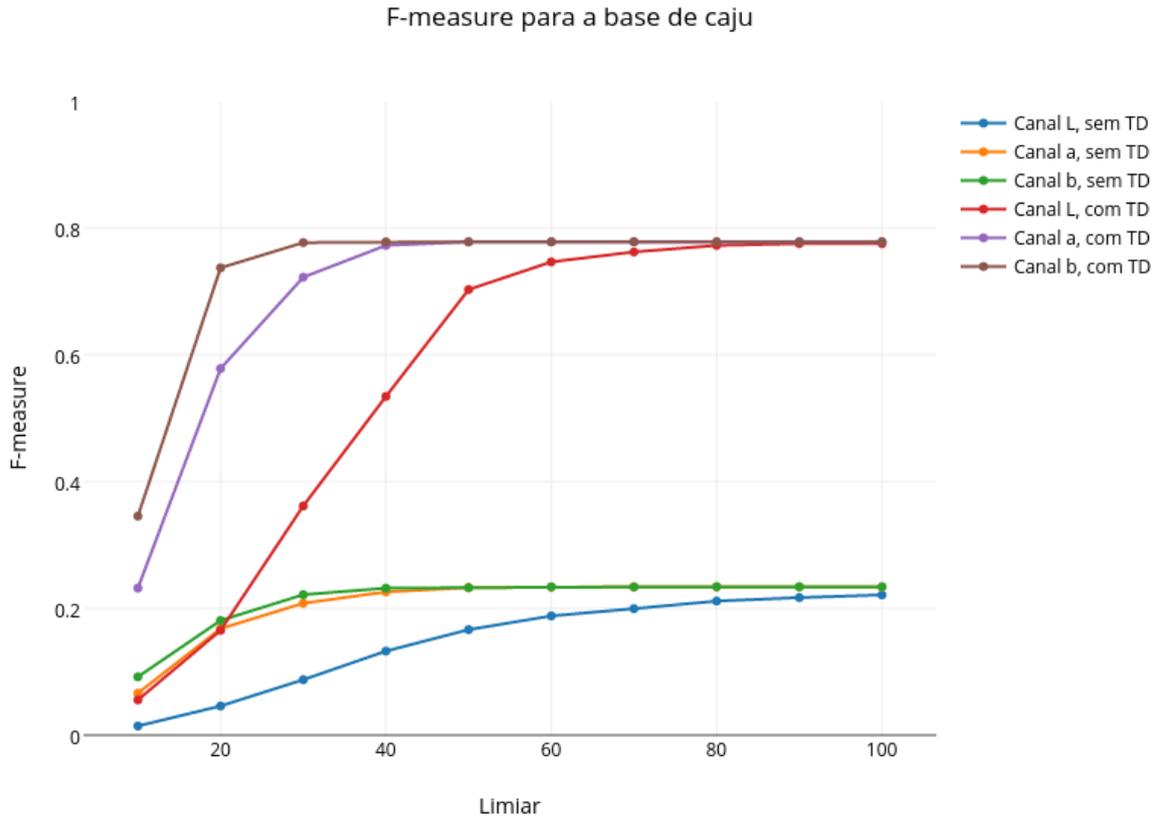


Figura A.5: Gráfico dos resultados de  $F$ -measure para a base de caju, realizando treinamento com 30% e teste com 70% da base.

Tabela A.6: Resultados de  $F$ -measure para a base de caju, realizando treinamento com 70% e teste com 30% da base.

Limiar	Caju					
	Sem transformação de distância			Com transformação de distância		
	Canal $L$	Canal $a$	Canal $b$	Canal $L$	Canal $a$	Canal $b$
10	0.00991736	0.0500759	0.0781065	0.0344828	0.161111	0.347826
20	0.029438	0.122449	0.147651	0.123711	0.383562	0.811594
30	0.052506	0.154567	0.156398	0.26484	0.7	0.835821
40	0.0815822	0.157143	0.157518	0.5	0.848485	0.848485
50	0.106109	0.157518	0.157518	0.682927	0.848485	0.848485
60	0.123364	0.157518	0.157518	0.848485	0.848485	0.848485
70	0.130435	0.157518	0.157518	0.848485	0.848485	0.848485
80	0.136646	0.157518	0.157518	0.848485	0.848485	0.848485
90	0.144737	0.157518	0.157518	0.848485	0.848485	0.848485
100	0.148984	0.157518	0.157518	0.848485	0.848485	0.848485

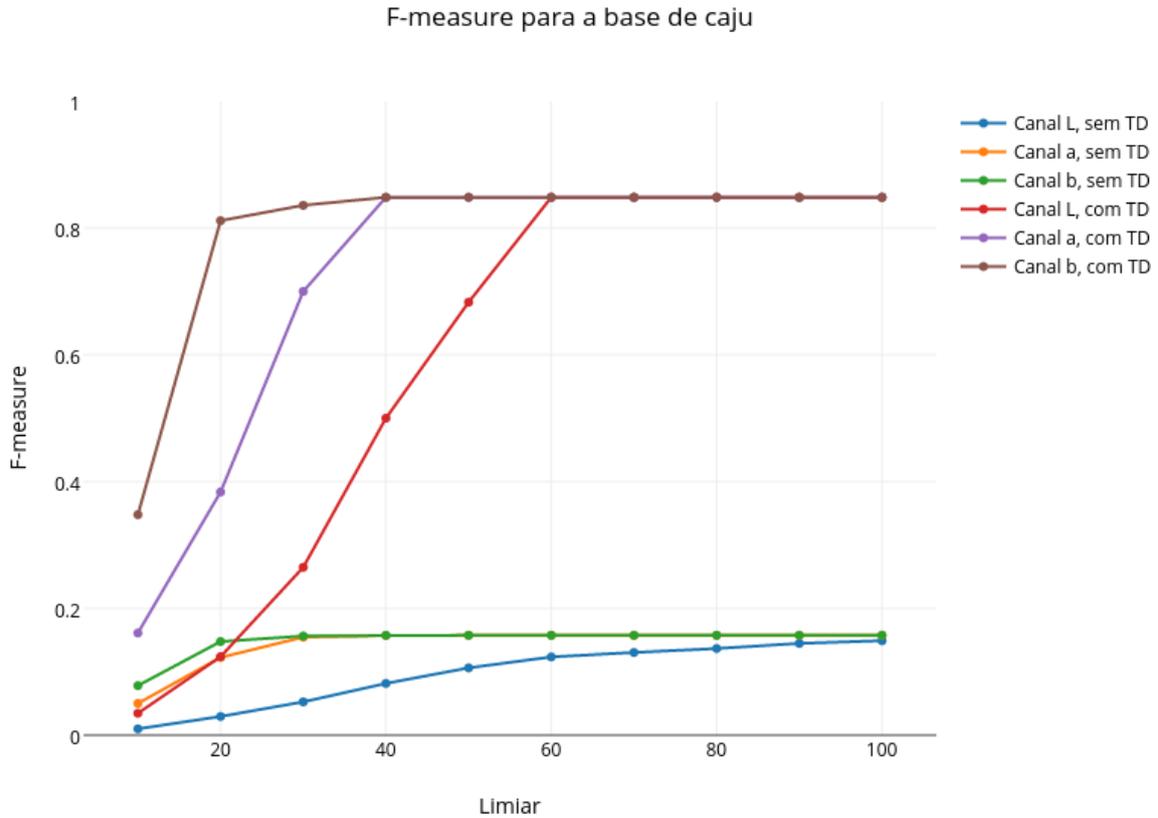


Figura A.6: Gráfico dos resultados de  $F$ -measure para a base de caju, realizando treinamento com 70% e teste com 30% da base.

Tabela A.7: Resultados de  $F$ -measure para a base de cereja, realizando treinamento com 30% e teste com 70% da base.

Limiar	Cereja					
	Sem transformação de distância			Com transformação de distância		
	Canal $L$	Canal $a$	Canal $b$	Canal $L$	Canal $a$	Canal $b$
10	0.0582969	0.231613	0.239552	0.237113	0.460641	0.379437
20	0.118102	0.557225	0.542212	0.367836	0.544248	0.444444
30	0.184941	0.668596	0.613603	0.447115	0.545012	0.504587
40	0.262264	0.665639	0.652106	0.492625	0.54321	0.534314
50	0.348866	0.659341	0.655118	0.531952	0.527638	0.527638
60	0.430141	0.657233	0.655118	0.529524	0.527638	0.527638
70	0.486085	0.655118	0.655118	0.544715	0.527638	0.527638
80	0.541254	0.655118	0.655118	0.555789	0.527638	0.527638
90	0.553341	0.655118	0.655118	0.563319	0.527638	0.527638
100	0.578065	0.655118	0.655118	0.556291	0.527638	0.527638

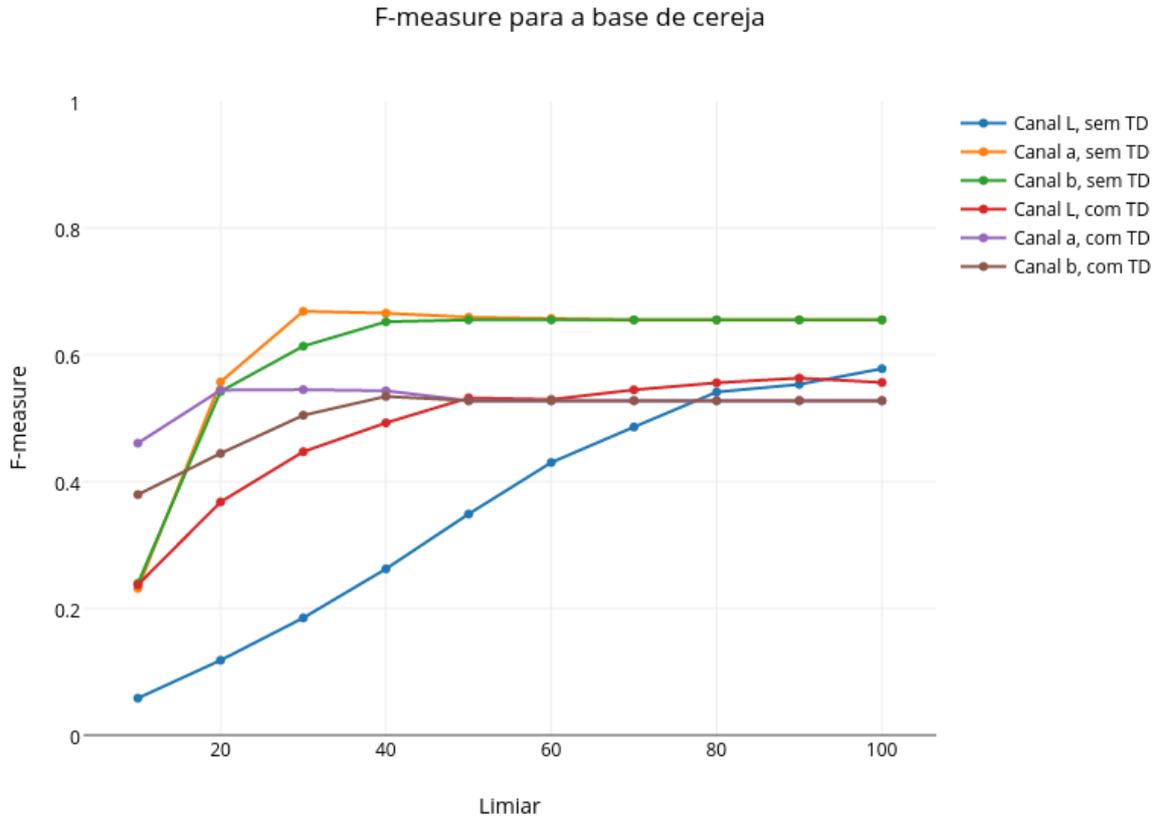


Figura A.7: Gráfico dos resultados de  $F$ -measure para a base de cereja, realizando treinamento com 30% e teste com 70% da base.

Tabela A.8: Resultados de  $F$ -measure para a base de cereja, com treinamento com 70% e teste com 30% da base.

Limiar	Cereja					
	Sem transformação de distância			Com transformação de distância		
	Canal $L$	Canal $a$	Canal $b$	Canal $L$	Canal $a$	Canal $b$
10	0.104078	0.364592	0.431579	0.339343	0.516129	0.522255
20	0.193214	0.721174	0.759091	0.4846	0.42963	0.374101
30	0.318542	0.790476	0.764706	0.569106	0.390805	0.392308
40	0.458716	0.768473	0.76808	0.557185	0.393822	0.393822
50	0.593472	0.763092	0.765	0.503311	0.381323	0.381323
60	0.725953	0.765	0.765	0.447552	0.381323	0.381323
70	0.795082	0.765	0.765	0.434164	0.381323	0.381323
80	0.768559	0.765	0.765	0.440299	0.381323	0.381323
90	0.77027	0.765	0.765	0.441948	0.381323	0.381323
100	0.767059	0.765	0.765	0.418251	0.381323	0.381323

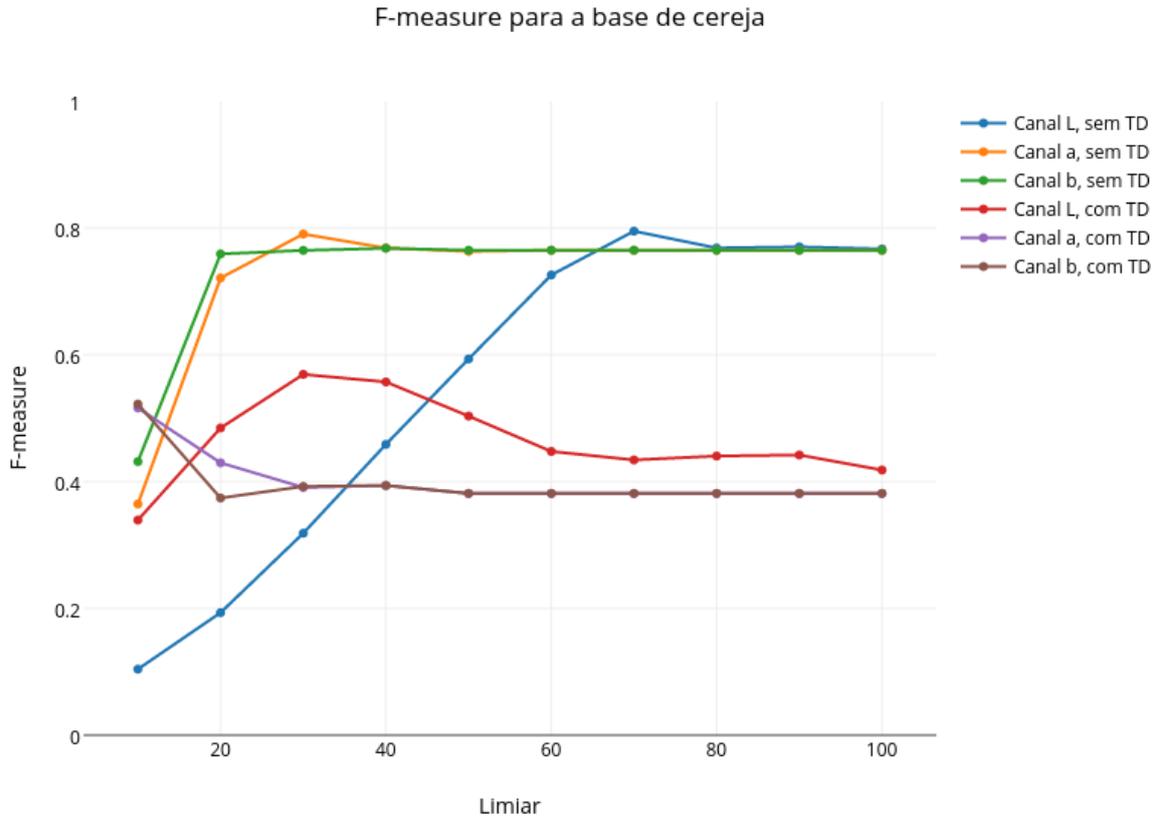


Figura A.8: Gráfico dos resultados de  $F$ -measure para a base de cereja, realizando treinamento com 70% e teste com 30% da base.

Tabela A.9: Resultados de  $F$ -measure para a base de laranja, com treinamento com 30% e teste com 70% da base.

Limiar	Laranja					
	Sem transformação de distância			Com transformação de distância		
	Canal $L$	Canal $a$	Canal $b$	Canal $L$	Canal $a$	Canal $b$
10	0.0798743	0.350506	0.245488	0.388919	0.676423	0.606144
20	0.14552	0.485962	0.460449	0.613146	0.671027	0.664695
30	0.213526	0.503543	0.497309	0.646355	0.657643	0.664066
40	0.279446	0.5085	0.512135	0.658141	0.653616	0.659802
50	0.345926	0.511221	0.514177	0.670206	0.653339	0.6563
60	0.40225	0.512204	0.512879	0.677801	0.651721	0.651721
70	0.438627	0.512204	0.512889	0.686225	0.651181	0.651181
80	0.467585	0.512693	0.512889	0.674315	0.651181	0.651181
90	0.49049	0.512889	0.512889	0.663385	0.651181	0.651181
100	0.502819	0.512889	0.512889	0.662064	0.651181	0.651181

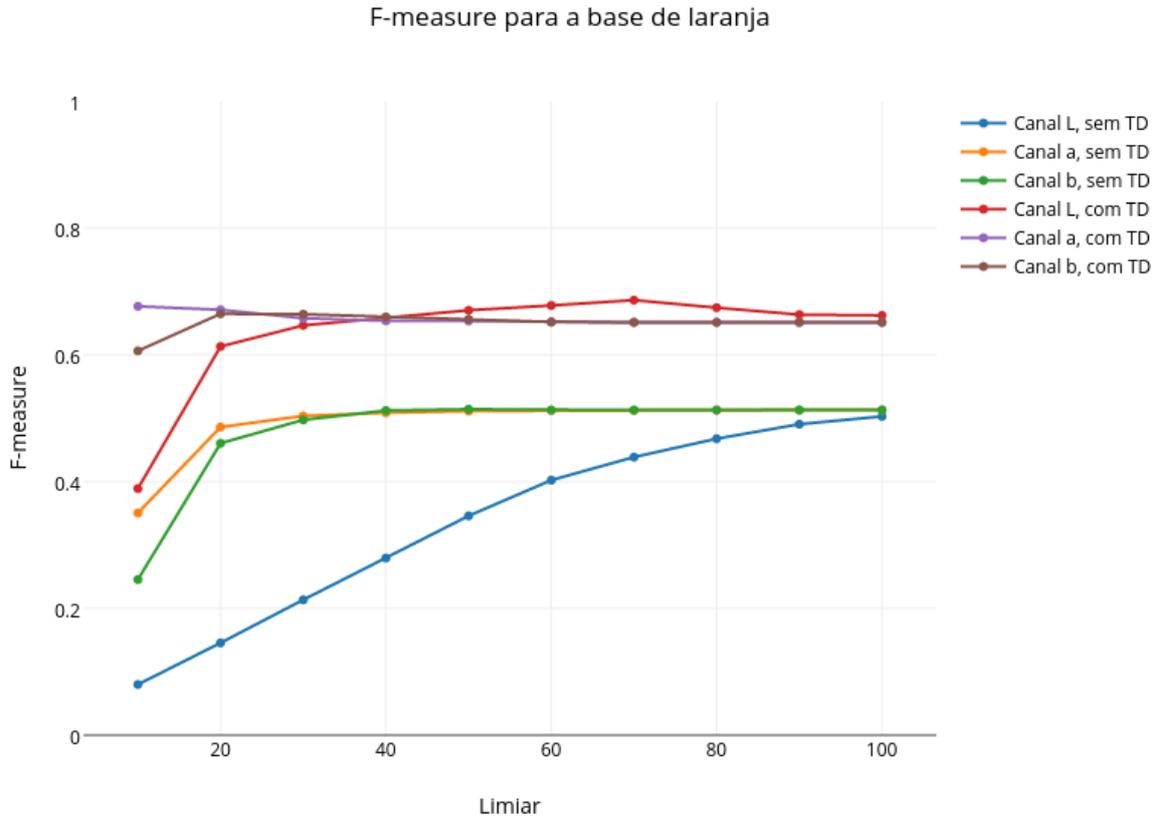


Figura A.9: Gráfico dos resultados de  $F$ -measure para a base de laranja, realizando treinamento com 30% e teste com 70% da base.

Tabela A.10: Resultados de  $F$ -measure para a base de laranja, realizando treinamento com 70% e teste com 30% da base.

Limiar	Laranja					
	Sem transformação de distância			Com transformação de distância		
	Canal $L$	Canal $a$	Canal $b$	Canal $L$	Canal $a$	Canal $b$
10	0.129294	0.507776	0.3626	0.423521	0.607985	0.503569
20	0.293131	0.550989	0.556213	0.630376	0.557377	0.56654
30	0.412371	0.551255	0.558653	0.615748	0.548982	0.550921
40	0.495406	0.55409	0.560768	0.606061	0.548105	0.545278
50	0.560241	0.556087	0.561666	0.610864	0.548105	0.545278
60	0.586629	0.558761	0.559957	0.58473	0.545278	0.545278
70	0.582104	0.560257	0.560257	0.567644	0.545278	0.545278
80	0.573034	0.560257	0.560257	0.555024	0.545278	0.545278
90	0.564424	0.560257	0.560257	0.552124	0.545278	0.545278
100	0.563127	0.560257	0.560257	0.54845	0.545278	0.545278

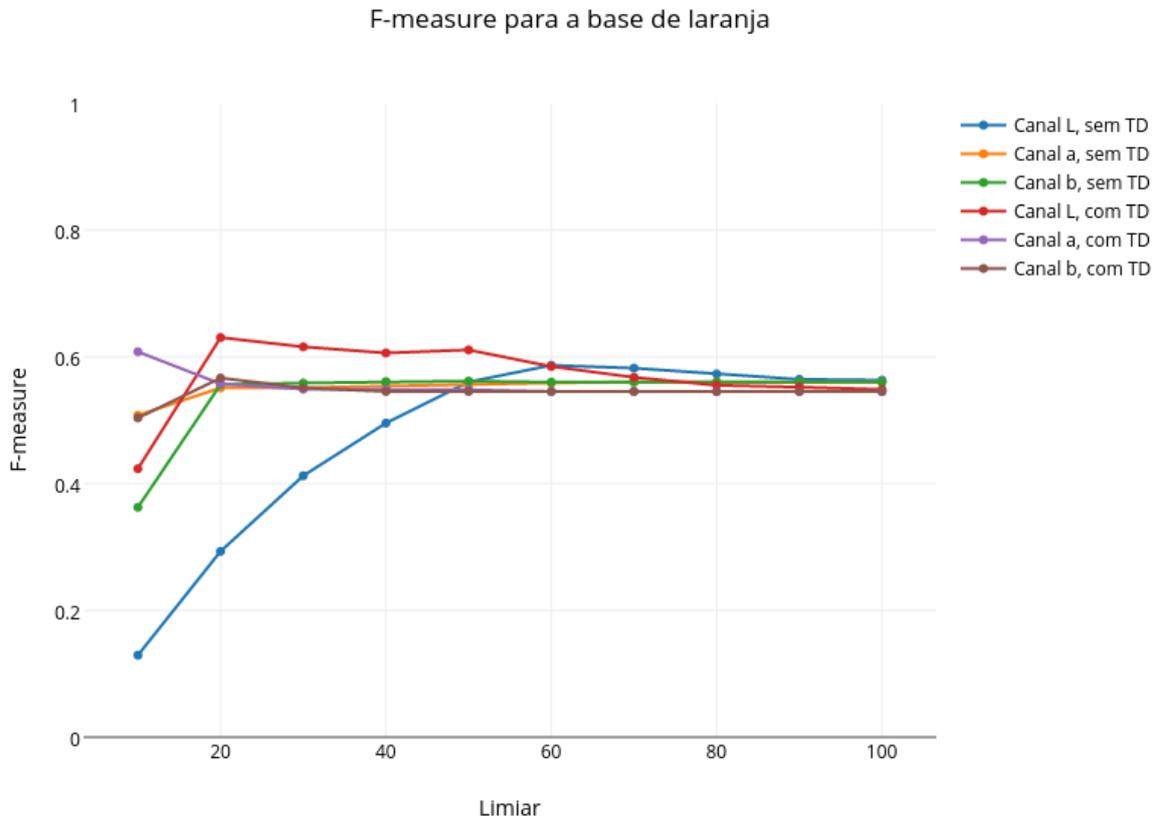


Figura A.10: Gráfico dos resultados de  $F$ -measure para a base de laranja, realizando treinamento com 70% e teste com 30% da base.

Tabela A.11: Resultados de  $F$ -measure para a base de manga rosa, realizando treinamento com 30% e teste com 70% da base.

Limiar	Manga					
	Sem transformação de distância			Com transformação de distância		
	Canal $L$	Canal $a$	Canal $b$	Canal $L$	Canal $a$	Canal $b$
10	0.0258966	0.114603	0.134078	0.0810267	0.241117	0.278922
20	0.0526115	0.235533	0.264706	0.143109	0.361949	0.382353
30	0.0878906	0.325386	0.369427	0.19697	0.482972	0.55516
40	0.113416	0.41953	0.46748	0.259918	0.557143	0.524345
50	0.142077	0.485356	0.486258	0.308748	0.524345	0.524345
60	0.18	0.488323	0.491453	0.354582	0.524345	0.524345
70	0.210621	0.491453	0.493562	0.4038	0.524345	0.524345
80	0.243496	0.493562	0.493562	0.425316	0.524345	0.524345
90	0.274326	0.493562	0.493562	0.438503	0.524345	0.524345
100	0.295541	0.493562	0.493562	0.452722	0.524345	0.524345

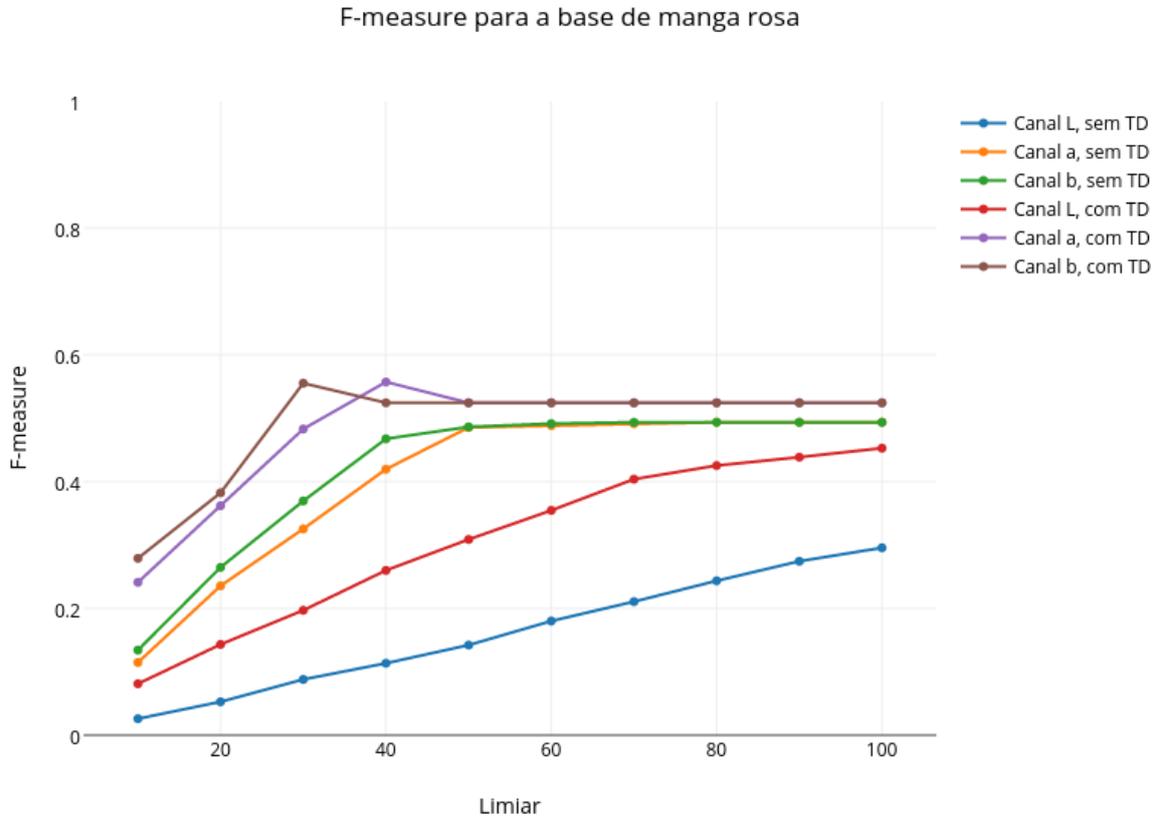


Figura A.11: Gráfico dos resultados de  $F$ -measure para a base de manga rosa, realizando treinamento com 30% e teste com 70% da base.

Tabela A.12: Resultados de  $F$ -measure para a base de manga rosa, com treinamento com 70% e teste com 30% da base.

Limiar	Manga					
	Sem transformação de distância			Com transformação de distância		
	Canal $L$	Canal $a$	Canal $b$	Canal $L$	Canal $a$	Canal $b$
10	0.0317593	0.105971	0.0742605	0.0996068	0.16185	0.190291
20	0.0474407	0.141439	0.124183	0.156057	0.264151	0.294737
30	0.0561941	0.210721	0.234568	0.185776	0.407767	0.439791
40	0.066702	0.306452	0.323864	0.197761	0.461538	0.431138
50	0.0744544	0.368932	0.365385	0.235849	0.431138	0.431138
60	0.0876133	0.371336	0.371336	0.289941	0.431138	0.431138
70	0.100885	0.371336	0.372549	0.34507	0.431138	0.431138
80	0.117526	0.372549	0.372549	0.428571	0.431138	0.431138
90	0.15	0.372549	0.372549	0.44	0.431138	0.431138
100	0.170149	0.372549	0.372549	0.457447	0.431138	0.431138

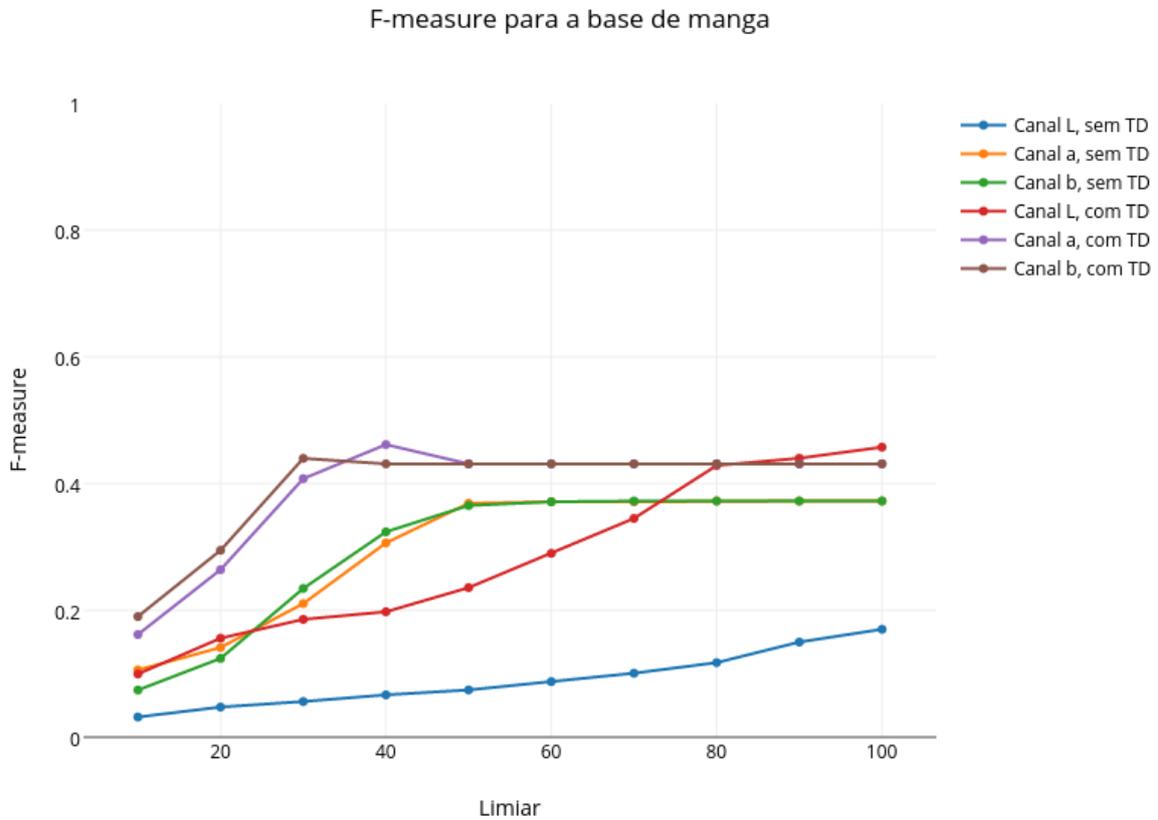


Figura A.12: Gráfico dos resultados de  $F\text{-measure}$  para a base de manga rosa, realizando treinamento com 70% e teste com 30% da base.

Tabela A.13: Resultados de  $F\text{-measure}$  para a base de morango, realizando treinamento com 30% e teste com 70% da base.

Limiar	Morango					
	Sem transformação de distância			Com transformação de distância		
	Canal $L$	Canal $a$	Canal $b$	Canal $L$	Canal $a$	Canal $b$
10	0.0915758	0.299619	0.268872	0.263759	0.31308	0.283263
20	0.157907	0.374938	0.342437	0.312536	0.310259	0.281094
30	0.248445	0.37792	0.358072	0.341676	0.291741	0.273848
40	0.313268	0.375669	0.360637	0.350491	0.284356	0.273409
50	0.344688	0.377722	0.365759	0.349982	0.281718	0.273538
60	0.357548	0.373869	0.367468	0.341998	0.278942	0.27448
70	0.368397	0.370253	0.368744	0.320634	0.274096	0.272362
80	0.377264	0.367994	0.367347	0.306207	0.271493	0.271493
90	0.377029	0.367347	0.367347	0.29867	0.271059	0.271059
100	0.378248	0.367347	0.367347	0.293262	0.271059	0.271059

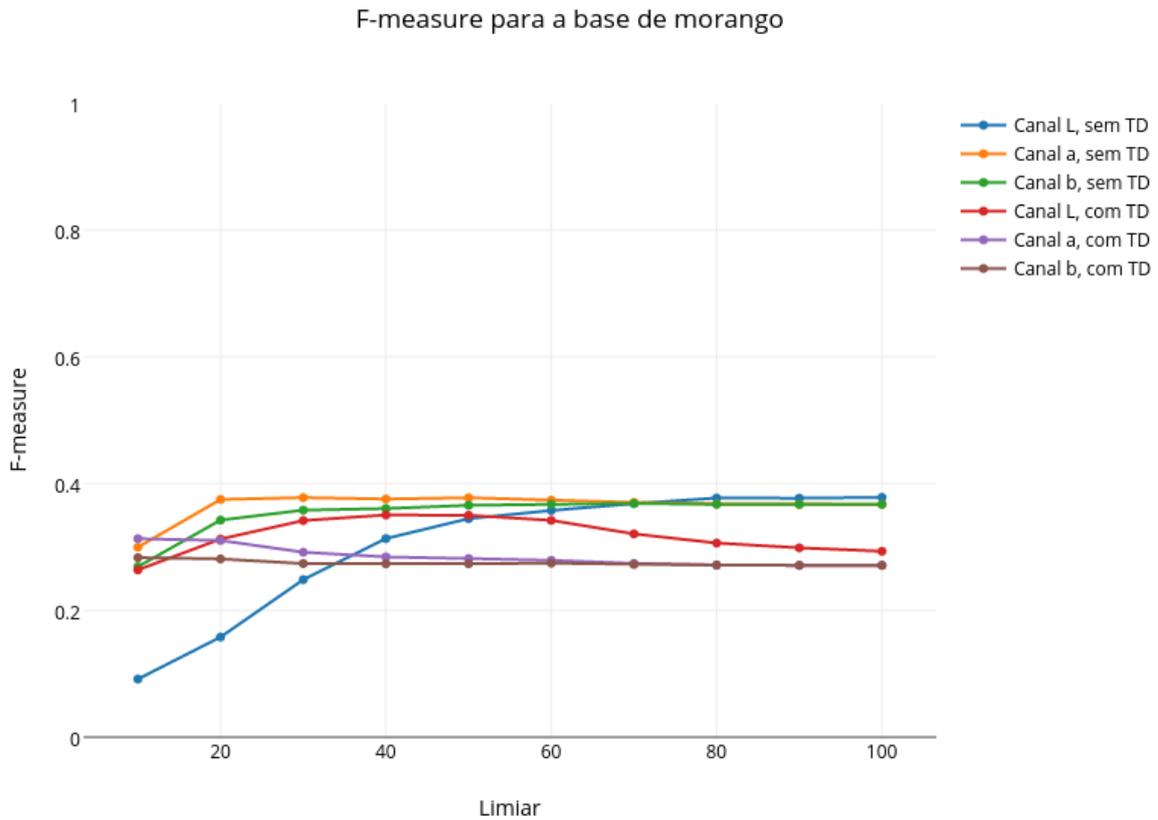


Figura A.13: Gráfico dos resultados de  $F$ -measure para a base de morango, realizando treinamento com 30% e teste com 70% da base.

Tabela A.14: Resultados de  $F$ -measure para a base de morango, com treinamento com 70% e teste com 30% da base.

Limiar	Morango					
	Sem transformação de distância			Com transformação de distância		
	Canal $L$	Canal $a$	Canal $b$	Canal $L$	Canal $a$	Canal $b$
10	0.160468	0.411332	0.371876	0.307524	0.32148	0.308656
20	0.271989	0.443687	0.3776	0.365767	0.308646	0.279303
30	0.362222	0.411111	0.351194	0.357128	0.286355	0.250882
40	0.407767	0.372294	0.329658	0.354337	0.263591	0.242942
50	0.408704	0.345231	0.328291	0.335438	0.248083	0.24008
60	0.381471	0.333895	0.327324	0.311543	0.243397	0.239866
70	0.371072	0.328257	0.325897	0.29381	0.239866	0.239866
80	0.358425	0.326842	0.325424	0.278233	0.239866	0.239866
90	0.357331	0.325424	0.325424	0.271581	0.239866	0.239866
100	0.353481	0.325424	0.325424	0.263743	0.239866	0.239866

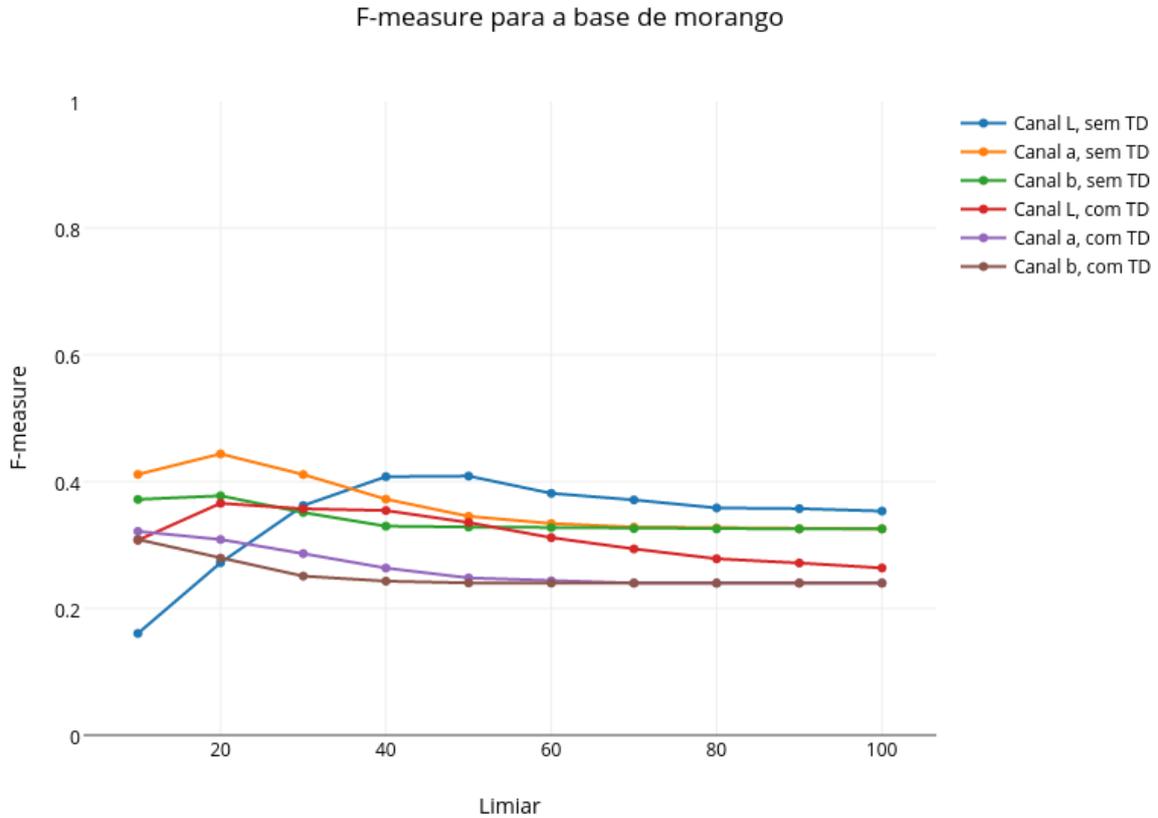


Figura A.14: Gráfico dos resultados de  $F$ -measure para a base de morango, realizando treinamento com 70% e teste com 30% da base.

Tabela A.15: Resultados de  $F$ -measure para a base de pêsego, realizando treinamento com 30% e teste com 70% da base.

Limiar	Pêssego					
	Sem transformação de distância			Com transformação de distância		
	Canal $L$	Canal $a$	Canal $b$	Canal $L$	Canal $a$	Canal $b$
10	0.0992338	0.391468	0.310689	0.355281	0.660393	0.612964
20	0.186395	0.546419	0.53999	0.58465	0.610961	0.640984
30	0.267115	0.585163	0.586403	0.628313	0.591815	0.622603
40	0.342423	0.599468	0.598865	0.648092	0.583803	0.584799
50	0.40566	0.60552	0.603299	0.646707	0.578799	0.578799
60	0.466522	0.606035	0.606401	0.645217	0.578296	0.578799
70	0.508787	0.60654	0.606578	0.655416	0.578296	0.578296
80	0.551046	0.606667	0.606667	0.641558	0.578296	0.578296
90	0.581869	0.606667	0.606667	0.634691	0.578296	0.578296
100	0.600738	0.606667	0.606667	0.620202	0.578296	0.578296

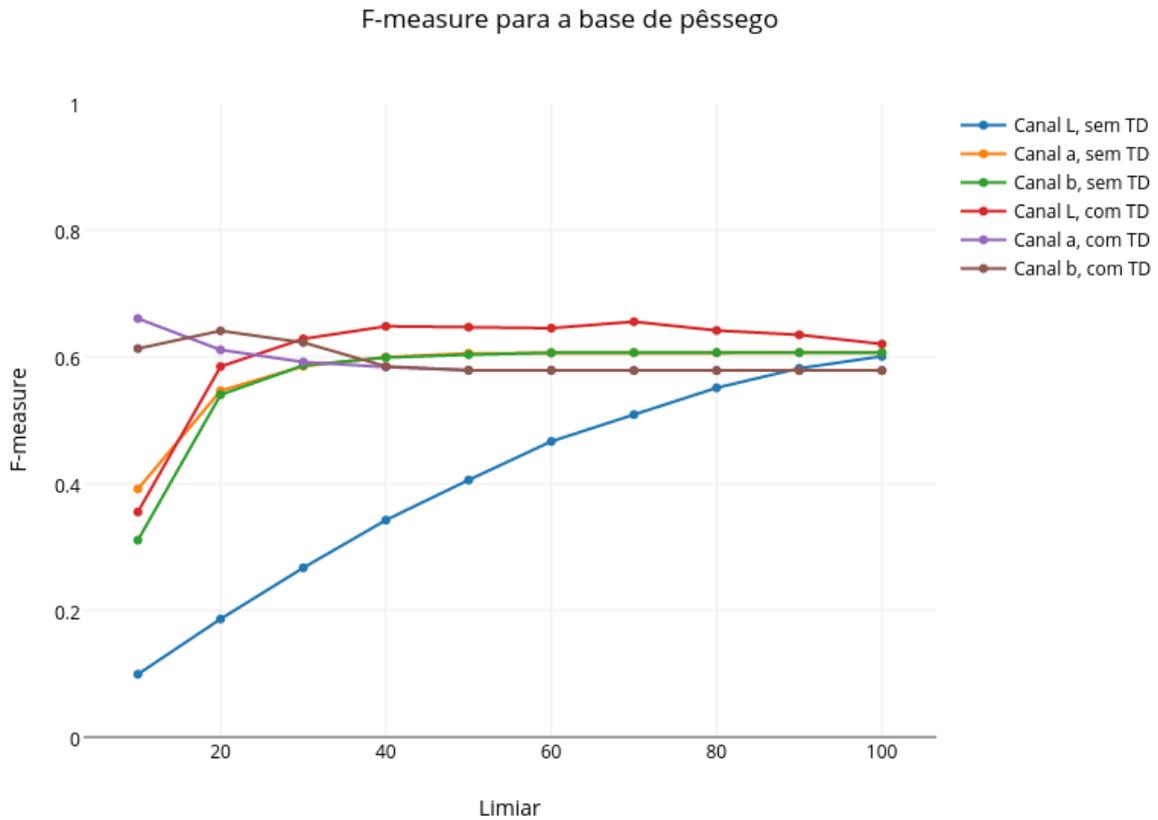


Figura A.15: Gráfico dos resultados de  $F$ -measure para a base de pêsego, realizando treinamento com 30% e teste com 70% da base.

Tabela A.16: Resultados de  $F$ -measure para a base de pêsego, realizando treinamento com 70% e teste com 30% da base.

Limiar	Pêssego					
	Sem transformação de distância			Com transformação de distância		
	Canal $L$	Canal $a$	Canal $b$	Canal $L$	Canal $a$	Canal $b$
10	0.125009	0.464576	0.383758	0.39268	0.631763	0.567244
20	0.266103	0.620479	0.614311	0.601406	0.567071	0.58738
30	0.387082	0.6335	0.628557	0.631995	0.548708	0.559496
40	0.479613	0.638341	0.63877	0.63172	0.54813	0.549285
50	0.55098	0.638945	0.640467	0.623596	0.546975	0.547553
60	0.590591	0.639594	0.638741	0.612868	0.546975	0.546975
70	0.613951	0.638903	0.638903	0.605017	0.546975	0.546975
80	0.632859	0.638557	0.638557	0.586154	0.546975	0.546975
90	0.638856	0.638557	0.638557	0.577087	0.546975	0.546975
100	0.644217	0.638557	0.638557	0.570646	0.546975	0.546975

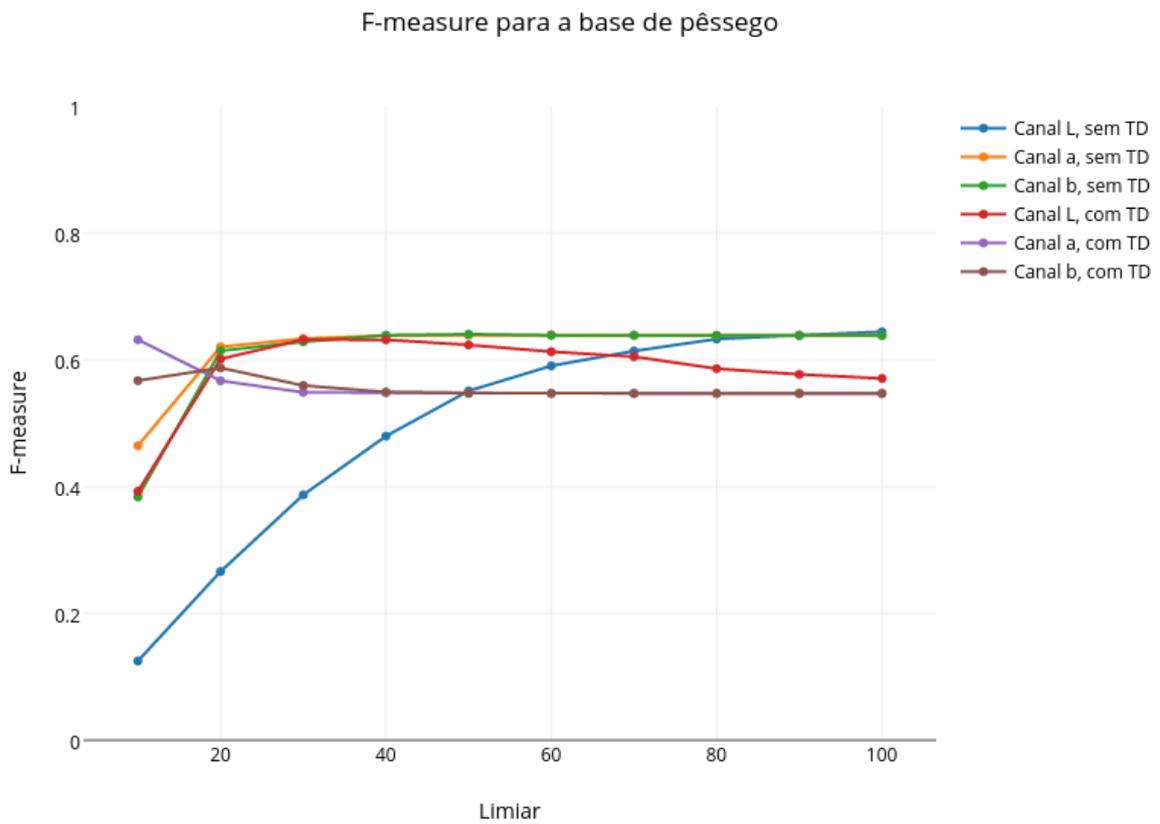


Figura A.16: Gráfico dos resultados de  $F$ -measure para a base de pêssego, realizando treinamento com 70% e teste com 30% da base.

# Referências

- [1] Ciclo de agricultura de precisão. <http://www.agrolink.com.br/georreferenciamento/AgriculturaPrecisao.aspx>. Acessado em junho de 2015. 3
- [2] Explanation of the lab color space. [http://www.aces.edu/dept/fisheries/education/pond\\_to\\_plate/documents/ExplanationoftheLABColorSpace.pdf](http://www.aces.edu/dept/fisheries/education/pond_to_plate/documents/ExplanationoftheLABColorSpace.pdf). Acessado em novembro de 2015. 10, 11
- [3] Moments – structural analysis and shape descriptors – OpenCV 2.4.12 documentation. [http://docs.opencv.org/2.4/modules/imgproc/doc/structural\\_analysis\\_and\\_shape\\_descriptors.html?highlight=moments#moments](http://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?highlight=moments#moments). Acessado em novembro de 2015. 13
- [4] A. Javed e T. Fatima A. Mustaqem. An efficient brain tumor detection algorithm using watershed & thresholding based segmentation. *I.J. Image, Graphics and Signal Processing*, pages 34–39, 2012. 19
- [5] P. Subedi e D. Jarvis A. Payne, K. Walsh. Estimation of mango crop yield using image analysis–segmentation method. *Computers and electronics in agriculture*, 91:57–64, 2013. 7
- [6] A. J. Baddeley. An error metric for binary images. *Proc. Int. Workshop on Robust Computer Vision*, pages 59–78, 1992. 20
- [7] Z. Taylor J. Underwood e S. Sukkarieh C. Hung, J. Nieto. Orchard fruit segmentation using multi-spectral feature learning. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 5314–5320. IEEE, 2013. 5
- [8] C. Lee e C. S. Kim C. Lee. Contrast enhancement based on layered difference representation of 2d histograms. *IEEE Trans. Image Process*, 22(12):5372–5384, 2013. 12
- [9] T. Celik. Two-dimensional histogram equalization and contrast enhancement. *Pattern Recognit*, 45(10):3810–3824, 2012. 12
- [10] M. Tresanchez M. Teixidó D. Martinez J. Moreno D. Font, T. Pallejà and J. Palacín. Counting red grapes in vineyards by detecting specular spherical reflection peaks in rgb images obtained at night with artificial illumination. *Computers and Electronics in Agriculture*, 108:105–111, 2014. 6

- [11] J. Bouchot e F. Morain-Nicolier. Scaled-distance-transforms and monotonicity of autocorrelations. *IEEE Signal Processing Letters*, 21(10):59–78, 1992. 20, 21
- [12] Hong-Yun Gao e Kin-Man Lam. Segmentation-enhanced saliency detection modell based on distance transform and center bias. *2014 IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 2803–2807, 2014. 20, 21
- [13] R. C. Gonzalez e R. E. Woods. *Digital Image Processing*. Tata McGraw Hill Education Private Limited, 3rd edition edition, 2010. 8, 9, 10, 12, 13, 14
- [14] G. Semeraro e J. Kay F. Esposito, D. Malerba. A comparative analysis of methods for pruning decision trees. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(5):476–491, 1997. 15, 18
- [15] M. Hirafuji, H. Yoichi, M. Wada, T. Fukatsu, T. Kiura, H. Hu, S. Ninomiya, and H. Shimamura. Demo abstract: Field server: Multi-functional wireless sensor network node for earth observation. 2005. 1
- [16] M. Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, 1962. 13
- [17] Y. Yoshioka e S. Ninomiya K. Yamamoto, W. Guo. On plant detection of intact tomato fruits using image analysis and machine learning methods. *Sensors*, 14(7):12191–12206, 2014. 6
- [18] C.J. Stone e R.A. Olshen L. Breiman, J. Friedman. *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984. 15
- [19] G. Li. Improved watershed segmentation with optimal scale based on ordered dither halftone and mutual information. *Computer Science and Information Technology (ICCSIT)*, pages 296–300, 2010. 19
- [20] G. Pajares e P. Zitinski M. Guijarro, I. Riomoros. Discrete wavelets transform for improving greenness image segmentation in agricultural images. *Computers and Electronics in Agriculture*, 118:396–407, 2015. 5
- [21] J. Kaur e J. Kaur M. Kaur. Survey of contrast enhancement techniques based on histogram equalization. *(IJACSA) International Journal of Advanced Computer Science and Applications*, 2(7):137–141, 2011. 12
- [22] S. Shevade e M. N. Murty P. M. Chinta, P. Balamurugan. Optimizing f-measure with non-convex loss and sparse linear classifiers. *Neural Networks (IJCNN), The 2013 International Joint Conference*, pages 1–8, 2013. 21
- [23] D. Pokrajac, A. Lazarevic, S. Vucetic, T. Fiez, and Z. Obradovic. Image processing in precision agriculture. *proc. TELSIKS'99*, pages 616–619, 1999. 2

- [24] A. Ribeiro, C. Fernández-Quintanilla, J. Barroso, M.C. García-Alegre, J.V. Stafford, et al. Development of an image analysis system for estimation of weed pressure. In *Precision agriculture'05. Papers presented at the 5th European Conference on Precision Agriculture, Uppsala, Sweden.*, pages 169–174. Wageningen Academic Publishers, 2005. 5
- [25] C. J. Rijsbergen. Information retrieval. *Butterworth*, 2013. 21
- [26] O. Rokach, L. e Maimon. Top-down induction of decision trees classifiers-a survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 35(4):476–487, 2005. 17
- [27] A.V. Resende L.H. Bassoi e A.C.C. Bernardi R.Y. Inamasu, J.M. Naime. *Agricultura de precisão: um novo olhar*. São Carlos: Embrapa Instrumentação, 2011. 1, 2
- [28] N. D. Thombare e S. S. Lokhande S. S. Kanitkar. Detection of lung cancer using marker-controlled watershed transform. *2015 International Conference on Pervasive Computing (ICPC)*, 2015. 19
- [29] Y. Suh. Probabilistic class histogram equalization based on posterior mean estimation for robust speech recognition. *IEEE Signal Processing Letters*, 22(12):2421–2424, 2015. 11
- [30] S. Suzuki and K. Abe. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985. 13
- [31] U.K. Rage e S. Ninomiya W. Guo. Illumination invariant segmentation of vegetation for time series wheat images based on decision tree model. *Computers and Electronics in Agriculture*, 96:58–66, August 2013. 1, 6, 15, 16, 23
- [32] D.M. Woebbecke, G.E. Meyer, K. Von Bargen, and D.A. Mortensen. Shape features for identifying young weeds using image analysis. *Transactions of the ASAE*, 38(1):271–281, 1995. 5
- [33] Y. Wang H. Shen e X. Lu Z. Ling, Y. Liang. Adaptive extended piecewise histogram equalisation for dark image enhancement. *IET Image Process*, 9(11):1012–1019, 2015. 11, 12
- [34] N. Zhang, M. Wang, and N. Wang. Precision agriculture—a worldwide overview. *Computers and electronics in agriculture*, 36(2):113–132, 2002. 1