

**UNIVERSIDADE DE CAXIAS DO SUL
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E
ENGENHARIAS**

LUIZ PEDRO ANDREETTA ZANELLA

**DESENVOLVIMENTO DE UM APLICATIVO MÓVEL PARA O
CÁLCULO DA ÁREA AFETADA PELA REQUEIMA DAS FOLHAS
EM VIDEIRAS**

CAXIAS DO SUL

2022

LUIZ PEDRO ANDREETTA ZANELLA

**DESENVOLVIMENTO DE UM APLICATIVO MÓVEL PARA O
CÁLCULO DA ÁREA AFETADA PELA REQUEIMA DAS FOLHAS
EM VIDEIRAS**

Trabalho de Conclusão de Curso
apresentado como requisito parcial
à obtenção do título de Bacharel em
Ciência da Computação na Área do
Conhecimento de Ciências Exatas e
Engenharias da Universidade de Caxias
do Sul.

Orientador: Prof. Dr. André Luis
Martinotto

Coorientador: Prof. Dr. Joséli Schwam-
bach

CAXIAS DO SUL

2022

LUIZ PEDRO ANDREETTA ZANELLA

**DESENVOLVIMENTO DE UM APLICATIVO MÓVEL PARA O
CÁLCULO DA ÁREA AFETADA PELA REQUEIMA DAS FOLHAS
EM VIDEIRAS**

Trabalho de Conclusão de Curso
apresentado como requisito parcial
à obtenção do título de Bacharel em
Ciência da Computação na Área do
Conhecimento de Ciências Exatas e
Engenharias da Universidade de Caxias
do Sul.

Aprovado em 01/07/2022

BANCA EXAMINADORA

Prof. Dr. André Luis Martinotto
Universidade de Caxias do Sul - UCS

Prof. Dr. Joséli Schwambach
Universidade de Caxias do Sul - UCS

Prof. Dr. Scheila de Ávila e Silva
Universidade de Caxias do Sul - UCS

RESUMO

A viticultura é um setor de destaque no agronegócio brasileiro e, assim como qualquer outra cultura agrícola, as videiras também estão suscetíveis a infestações de pragas e doenças. O monitoramento adequado dessas doenças é fundamental para a obtenção de bons resultados e frutos apropriados para comercialização. Esse monitoramento pode ser realizado com base no cálculo da área foliar afetada pela doença. No entanto, os métodos tradicionalmente utilizados para realizar esse cálculo podem apresentar um elevado custo e tempo de análise. Neste trabalho, foi desenvolvida uma solução para automatizar o processo de monitoramento da área foliar afetada pela requeima das folhas em videiras (*Alternaria Alternata*). Esta solução foi dividida em duas partes: um aplicativo móvel, nativo para a plataforma Android, e um *Web Service*. O aplicativo móvel possibilita que o usuário capture uma imagem de uma folha afetada pela doença a partir do seu *smartphone*. Essa imagem é enviada para o *Web Service* onde passa por um processamento digital, que inclui a remoção de ruídos, remoção do fundo da imagem, transformação do sistema de cores e segmentação da área afetada pela doença. Por fim, a estimativa da área afetada pela doença é realizada através do cálculo da área de um *pixel* e pela contagem dos *pixels* da imagem. Após, o resultado é enviado de para o aplicativo, onde é exibido para o usuário. Para validação, a área afetada pela requeima das folhas de videira foi calculada utilizando o software ImageJ e esses resultados foram comparados aos resultados obtidos utilizando o aplicativo desenvolvido. Os resultados obtidos foram satisfatórios, sendo que, para folhas com uma área foliar afetada superior a 3%, a diferença entre as áreas calculadas foi de no máximo 0,77%.

Palavras-chaves: Aplicativo móvel, Processamento digital de imagens, *Web Service*, Videiras, Requeima das folhas, *Alternaria Alternata*.

LISTA DE FIGURAS

Figura 1 – Sintomas do Míldio	13
Figura 2 – Folha afetada pela requeima foliar	14
Figura 3 – Manchas de ferrugem na face superior da folha	15
Figura 4 – Imagem obtida em um ambiente controlado	17
Figura 5 – Escala utilizada para o cálculo	17
Figura 6 – Segmentação da imagem	18
Figura 7 – Resultado da área calculada	19
Figura 8 – Modos de conectividade de um aplicativo	20
Figura 9 – Arquitetura Mobile Cloud Computing	21
Figura 10 – Participação no Mercado de Sistemas Operacionais Móveis no Mundo (Mar 2021)	22
Figura 11 – Imagem com ruído	25
Figura 12 – Exemplo de Filtro de Mediana 3x3	26
Figura 13 – Aplicação do Filtro de Mediana com diferentes matrizes de vizinhança	27
Figura 14 – Remoção do fundo de uma imagem	28
Figura 15 – Exemplo de limiarização	28
Figura 16 – Histograma sem picos bem definidos	29
Figura 17 – Histograma com dois picos definidos	30
Figura 18 – Sistema de cores RGB	31
Figura 19 – Sistema de cores CIELAB	32
Figura 20 – Conversão de RGB para CIELAB	32
Figura 21 – Segmentação da doença	33
Figura 22 – Distribuição dos grupos por K-means	34
Figura 23 – Algoritmo K-means com 2 centroides	35
Figura 24 – Folha com moeda como objeto de referência	36
Figura 25 – Imagem com escala como referência	37
Figura 26 – Imagem da doença segmentada	38
Figura 27 – Arquitetura de Comunicação entre o Aplicativo e o Web Service	39
Figura 28 – Fluxograma do processamento da imagem	40
Figura 29 – Resultado da remoção de ruídos	40
Figura 30 – Resultado da remoção de fundo	41
Figura 31 – Resultado da conversão para CIELAB	41
Figura 32 – Resultado da segmentação	42
Figura 33 – Seleção da imagem	43
Figura 34 – Informar escala	44
Figura 35 – Executar processamento da imagem	45

Figura 36 – Resultado da análise	46
Figura 37 – Imagens selecionadas para validação	47
Figura 38 – Folha com área afetada similar a cor da folha	49

LISTA DE TABELAS

Tabela 1 – Comparação dos resultados obtidos	48
--	----

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
CSS	<i>Cascading Style Sheets</i>
HTML	<i>HyperText Markup Language</i>
MCC	<i>Mobile Cloud Computing</i>
SDK	<i>Software Development Kit</i>
REST	<i>Representational State Transfer</i>
RGB	<i>Red, Green e Blue</i>
YCBCR	Sistema de cores definido por brilho e componentes cromáticos
HSI	<i>Hue, saturation e intensity</i>
CIELAB	Sistema de cores definido por <i>International Commission on Illumination</i>
CM	Centímetro

SUMÁRIO

1	INTRODUÇÃO	9
1.1	OBJETIVOS	10
1.2	ESTRUTURA DO TRABALHO	10
2	VITICULTURA BRASILEIRA	12
2.1	DOENÇAS FÚNGICAS DA VIDEIRA	12
2.1.1	Míldio	13
2.1.2	Requeima Foliar	14
2.1.3	Ferrugem	15
2.2	MONITORAMENTO DA ÁREA FOLIAR AFETADA	16
3	APLICATIVOS MÓVEIS	20
3.1	CONECTIVIDADE EM APLICATIVOS MÓVEIS	20
3.2	MOBILE CLOUD COMPUTING	21
3.3	DESENVOLVIMENTO DE APLICATIVOS MÓVEIS	22
3.3.1	Desenvolvimento Nativo	23
3.3.2	Desenvolvimento Multiplataforma	23
4	TÉCNICAS DE PROCESSAMENTO DIGITAL DE IMAGEM	25
4.1	REMOÇÃO DE RUÍDOS	25
4.2	REMOÇÃO DO FUNDO	27
4.3	TRANSFORMAÇÃO DO SISTEMA DE CORES	31
4.4	SEGMENTAÇÃO DA ÁREA DA DOENÇA	33
4.5	CÁLCULO DA ÁREA AFETADA	35
5	IMPLEMENTAÇÃO DESENVOLVIDA	39
5.1	WEB SERVICE	40
5.2	APLICATIVO MÓVEL	42
5.3	VALIDAÇÃO E AVALIAÇÃO DOS RESULTADOS	46
6	CONSIDERAÇÕES FINAIS	50
6.1	TRABALHOS FUTUROS	51
	REFERÊNCIAS	52

1 INTRODUÇÃO

A viticultura é um dos setores de grande destaque no agronegócio brasileiro. Essa produz, em média, 1,4 milhões de toneladas de uvas por ano, sendo que deste montante, 51% é destinado ao processamento de sucos, vinhos e derivados, e os outros 49% restante, para o consumo *in natura* (MELLO, 2019).

O cultivo da uva pode ser observado em todas as regiões brasileiras, cada uma das quais com suas características particulares. Porém, o Rio Grande do Sul concentra a maior área cultivada, sendo o principal produtor de uva do país. De fato, esse é responsável por mais da metade da produção nacional (MELLO, 2019).

Apesar de existirem grandes empresas e um alto investimento associado à viticultura, a maior parte da produção ainda é originada de pequenas propriedades de agricultura familiar. Desta forma, essa atividade é uma importante fonte de emprego e renda (MELLO, 2018).

Assim como, em qualquer cultura agrícola, as videiras também estão suscetíveis à infestação de pragas e doenças, que acabam afetando significativamente a produção da uva, causando perdas parciais ou total de rendimento e de qualidade. Desta forma, o aumento da produtividade da uva e seus derivados, está diretamente ligado ao controle dessas pragas e doenças (GARRIDO; GAVA, 2014).

Dentre as doenças que atacam as videiras, destaca-se a requeima das folhas, que é causada pelo fungo *Alternaria alternata*. Essa doença é originada nas folhas, sendo que os sintomas iniciais são manchas bem definidas de cor arroxeadas, evoluindo para um marrom-escuro. Rapidamente, essas lesões aumentam de tamanho cobrindo toda a área foliar, provocando a morte e a queda da folha. O monitoramento e controle adequado da requeima é imprescindível, uma vez que essa prejudica a maturação dos frutos, altera os níveis de açúcares e acidez, tornando os frutos impróprios para a comercialização (GARRIDO; GAVA, 2014).

Embora não exista a recomendação de um método exato para controle da requeima em videiras, o uso de fungicidas tem se demonstrado uma boa alternativa (KUHN *et al.*, 2003). No entanto, para que esse tratamento seja altamente eficiente, tem-se a necessidade de um acompanhamento periódico da evolução da doença nas folhas (GARRIDO, 2015).

Esse acompanhamento, pode ser realizado com base em imagens capturadas das folhas, onde é possível calcular a área foliar afetada pela doença. Nos métodos mais tradicionais esse cálculo é realizado manualmente ou através do uso de softwares específicos, tornando o processo demorado e com um elevado custo (MARTIN *et al.*, 2013).

Dentro deste contexto, neste trabalho foi desenvolvido um aplicativo móvel para a automatização desse processo de monitoramento. Esse aplicativo permite que o usuário selecione ou capture imagens das folhas afetadas pela doença diretamente de seu *smartphone*. Essas imagens são enviadas para um *Web Service*, onde são aplicadas técnicas de processamento digital de imagens, como, remoção de ruídos, remoção de fundo, transformação do sistema de cores e segmentação da área afetada. Por fim, é realizado o cálculo da área afetada pela doença, sendo que o resultado é enviado para o aplicativo móvel onde é exibido para o usuário.

1.1 OBJETIVOS

O objetivo deste trabalho consistiu no desenvolvimento de um aplicativo para dispositivos móveis, que permite o cálculo da área foliar afetada pela requeima das folhas em videiras. Para que o objetivo geral deste trabalho fosse atingido, os seguintes objetivos específicos foram realizados:

1. Criação de uma base de imagens de folhas afetadas pela requeima das folhas. Essas imagens foram utilizadas para a validação do aplicativo;
2. Desenvolvimento de uma aplicação *Web Service* capaz de calcular a área foliar afetada pela doença.
3. Desenvolvimento de um aplicativo móvel capaz de capturar ou selecionar fotos para que sejam processadas pela aplicação *Web Service*;
4. Validação dos resultados obtidos a partir da comparação com resultados obtidos através do software ImageJ (MD MAGALHÃES PJ, 2004).

1.2 ESTRUTURA DO TRABALHO

O presente trabalho está organizado da seguinte forma:

- No Capítulo 2 são apresentadas as principais doenças que afetam as videiras. Além disso, é apresentado o processo de cálculo da área foliar afetada pela requeima das folhas utilizando o *software* ImageJ.
- No Capítulo 3 são apresentadas as principais características e arquiteturas que são utilizadas para o desenvolvimento de aplicativos móveis. Além disso, é apresentada uma comparação entre o desenvolvimento nativo e multiplataforma.
- No Capítulo 4 são apresentadas as técnicas de processamento digital de imagens que podem ser utilizadas para o cálculo da área foliar afetada pela doença.

- No Capítulo 5 é apresentada a solução desenvolvida para automatizar o processo de monitoramento da área foliar afetada pela requeima das folhas em videiras. Nele é detalhado a arquitetura que foi utilizada para o desenvolvimento do aplicativo e do *web service*, bem como as técnicas de processamento digital de imagens que foram utilizadas para o cálculo da área foliar afetada pela doença. Além disso, são apresentados os principais resultados obtidos e uma síntese da análise dos mesmos.
- No Capítulo 6 são apresentadas as considerações finais deste trabalho e sugestões de trabalhos futuros.

2 VITICULTURA BRASILEIRA

Dados históricos indicam que a videira vem sendo cultivada no Brasil desde 1532, quando os colonizadores portugueses trouxeram as primeiras variedades de uva para o país. Essas eram uvas finas européias (*Vitis vinifera*) originadas de Portugal e da Espanha, e que foram introduzidas na então capitania de São Vicente, que hoje representa o estado de São Paulo (PROTAS; CAMARGO; MELLO, 2002).

A partir de então a atividade expandiu-se para as demais regiões brasileiras. Porém, essa atividade somente se consolidou em meados do século XIX, quando os imigrantes italianos introduziram o cultivo das variedades americanas (*Vitis labrusca*), que são conhecidas no Brasil como uvas rústicas. Essa variedade de uva é assim chamada devido a facilidade do cultivo, melhor adaptação às condições locais e por serem mais resistentes do que as uvas europeias (RITSCHER; SEBEN, 2010).

Desta forma, os vinhedos de uvas européias foram sendo substituídos pelo cultivo das uvas americanas. Em particular pelo cultivo da variedade Isabel, que passou a ser plantada em várias regiões do país, tornando-se fundamental para o crescimento e consolidação da viticultura brasileira, principalmente, no estado do Rio Grande do Sul (PROTAS; CAMARGO; MELLO, 2002).

A viticultura passou a ser totalmente difundida em território brasileiro a partir da década de 1960, quando surgiram os primeiros vinhedos de uvas finas no semiárido nordestino, mais precisamente no Vale do Rio São Francisco. Nas décadas seguintes, desenvolveram-se as demais regiões produtoras do país, baseadas especificamente no cultivo de uvas finas para consumo *in natura*. Nesse período, surgem os pólos vitivinícolas do norte do Paraná, em 1970, e do noroeste de São Paulo e norte de Minas Gerais, em 1980 (BOTELHO; PIRES, 2009).

Atualmente, o Brasil possui 18 estados produtores de uva, que somados chegam a uma produção média anual de 1,4 milhões de toneladas da fruta. Deste montante, 51% é destinado para a fabricação de sucos, vinhos e derivados. Os 49% restante é utilizado para o consumo *in natura* (MELLO, 2019)

2.1 DOENÇAS FÚNGICAS DA VIDEIRA

A viticultura está presente em todas as regiões do país, sendo que cada uma dessas regiões possui características climáticas distintas. Algumas dessas condições climáticas se mostram favoráveis ao desenvolvimento de patógenos, que disseminam doenças de origem fúngicas e viróticas.

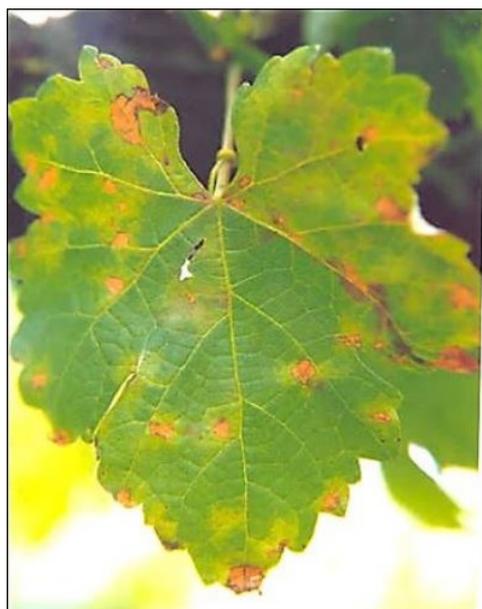
O controle e monitoramento adequado dessas doenças é essencial para a manutenção do padrão de qualidade dos frutos e evitar grandes perdas de produtividade. Dentre as doenças fúngicas de maior incidência em videiras, destacam-se o míldio (*Plasmopara viticola*), a requeima foliar (*Alternaria alternata*) e a ferrugem (*Phakopsora euvitis*) (KUNH *et al.*, 2003).

2.1.1 Míldio

O Míldio (*Plasmopara viticola*), também conhecido como mofo ou mufa, é muito comum em áreas de clima quente e úmido (GUERRA *et al.*, 2005). Essas condições climáticas, associadas a fatores como chuva, orvalho ou nevoeiro, aumentam o teor de água em torno da planta e favorecem o desenvolvimento da doença. A infecção do míldio ocorre pelas rachaduras naturais da planta, sendo fundamental a presença de água na superfície da videira para que ela aconteça (TAVARES; LIMA *et al.*, 2000).

Essa doença se caracteriza por afetar todas as partes da planta. Os primeiros sintomas podem ser observados nas folhas, onde surgem manchas amareladas com aspecto oleoso, conhecidas como "manchas de óleo" (Figura 1a). Nos cachos, nos ramos e na parte inferior das folhas, manifesta-se um mofo branco, originado pela alta umidade presente no ambiente. Aos poucos essas manchas evoluem, tornando-se necróticas e de cor avermelhadas, provocando a seca e a queda das folhas. As bagas passam a apresentar aspectos de podridão, ficando murchas e escurecidas, se destacando facilmente dos cachos (Figura 1b). Por fim, todo o desenvolvimento dos cachos e bagas são afetados (TAVARES; LIMA *et al.*, 2000).

Figura 1 – Sintomas do Míldio



(a) Folha com "manchas de óleo"



(b) Bagas com aspecto de podridão

Fonte: (GARRIDO; GAVA, 2014)

O controle do míldio deve ser iniciado, preventivamente, ainda na fase de plantio da videira. Deve-se optar por locais arejados, evitando áreas baixas e com um longo período de sombra. Além disso, deve-se manter um espaçamento apropriado entre os ramos, a fim de melhorar a ventilação da copa e diminuir a umidade presente em torno da planta (GUERRA *et al.*, 2005). Em casos de desenvolvimento do patógeno, o controle químico por uso de fungicidas é o mais indicado, alternando entre produtos de ação de contato e de ação sistêmica, durante todo o ciclo de desenvolvimento do fruto (KUHN *et al.*, 2003).

2.1.2 Requeima Foliar

A Requeima Foliar (*Alternaria alternata*) pode ser observada tanto nas variedades de uvas americanas (*Vitis labrusca*), como também em variedades de uvas finas européias (*Vitis vinifera*). Essa doença têm-se demonstrado altamente rigorosa, uma vez que acarreta na queda precoce das folhas, prejudica a maturação dos frutos, altera os níveis de açúcares e acidez, tornando os frutos impróprios para a comercialização, além de comprometer o desenvolvimento do ramos no ciclo seguinte (KUHN *et al.*, 2003).

A requeima é originada nas folhas, sendo que os sintomas iniciais, são manchas bem definidas de cor arroxeadas, evoluindo para um marrom-escuro (Figura 2). Rapidamente, essas lesões tornando-se necróticas, aumentando de tamanho e cobrindo toda a área foliar, provocando a morte e a queda da folha (GARRIDO; GAVA, 2014).

Figura 2 – Folha afetada pela requeima foliar



Fonte: (GARRIDO; GAVA, 2014)

Embora não exista um método exato para o controle da requeima foliar, o uso de fungicidas a base de tebuconazole e mancozeb tem se demonstrado uma boa alternativa. O uso desses fungicidas demonstram uma diminuição dos sintomas quando pulverizados no período de 30 dias após a poda, até 10 dias após o amolecimento das bagas (KUHN *et al.*, 2003).

2.1.3 Ferrugem

A Ferrugem (*Phakopsora euvitis*) caracteriza-se por ocorrer, principalmente, em áreas de climas tropicais e subtropicais. A severidade da doença nessas regiões é maior do que em regiões de clima mais frio, onde também é possível observar a contaminação, porém em fases finais do ciclo da cultura (GARRIDO; GAVA, 2014). As variedades de uvas americanas (*Vitis labrusca*) se mostram mais suscetíveis de contaminação pela doença, quando comparadas com variedades de uvas européias (*Vitis vinifera*) (GUERRA *et al.*, 2005).

Uma outra característica da ferrugem é o seu grande potencial de disseminação. De fato, os fungos são facilmente carregados pelo vento, podendo atingir vinhedos próximos, ou até mesmo mais distantes. Além disso, a disseminação pode ocorrer pela movimentação de pessoas e veículos, carregando material contaminado de uma área para outra (GARRIDO; GAVA, 2014).

Os sintomas da doença podem ser observados nas folhas da videira. Na face superior são produzidas manchas amareladas ou castanhas de inúmeros formatos, com aspecto semelhante a ferrugem (Figura 3). Já na face inferior, surgem massas de uredósporos de coloração amarelo-alaranjada, formando manchas necróticas sobre os mesmos na face superior da folha. A ferrugem, geralmente, afeta as folhas maduras e em casos mais severos podem levar a desfolha prematura da videira, comprometendo a produção e vigor da planta (KUHN *et al.*, 2003).

Figura 3 – Manchas de ferrugem na face superior da folha



Fonte: (GARRIDO; GAVA, 2014)

Uma vez que a desfolha causada pela ferrugem durante o período produtivo da videira é relativamente lenta, o controle químico dessa doença se mostra necessário em poucas áreas do cultivo. No entanto, este cenário costuma ser bem diferente na fase de repouso, que ocorre após a colheita. Neste período a velocidade da desfolha aumenta, chegando a desfolhar toda a cultura. Com base neste cenário, é recomendado a pulverização utilizando produtos dos grupos triazóis e estrobilurinas, em um período que deve ser iniciado próximo à colheita, estendendo-se até a fase inicial do repouso (KUHN *et al.*, 2003).

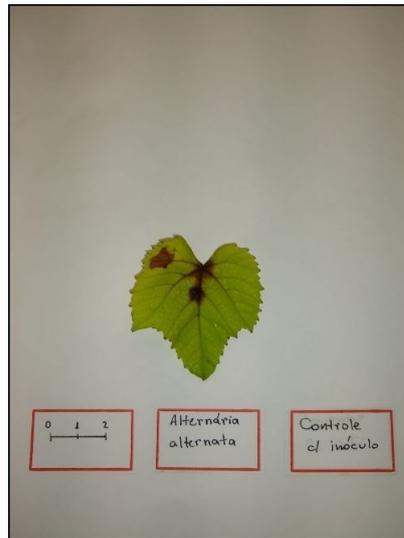
2.2 MONITORAMENTO DA ÁREA FOLIAR AFETADA

O monitoramento das doenças fúngicas em videira é fundamental para um maior controle sobre a aplicação de produtos químicos. Para a obtenção de um acompanhamento mais eficiente, pode-se efetuar um monitoramento periódico da evolução da área foliar afetada pela doença. A partir dessa informação é possível identificar os momentos certos para aplicação dos fungicidas, determinar áreas de aplicações mais específicas, ou então, optar pela diminuição da intensidade dos produtos aplicados (GARRIDO, 2015).

O cálculo da área foliar afetada pode ser realizado por meio de diferentes métodos, entretanto alguns desses métodos podem ser caros ou demandar um elevado tempo de análise. É o que ocorre, por exemplo, ao utilizar um integrador foliar (BOFF; ZAMBOLIM; VALE, 1991), onde os equipamentos são importados e de alto custo (MARTIN *et al.*, 2013). Em contrapartida, existem *softwares* gratuitos que possibilitam a realização do cálculo através de imagens de folhas infectadas, como é o caso do ImageJ (RAMOS *et al.*, 2015). Porém, neste caso tem-se um elevado tempo de análise, uma vez que todas as etapas do processo devem ser realizadas manualmente, além de exigir do usuário, um certo nível de conhecimento em métodos de processamento digital de imagem.

O processo da estimativa da área foliar afetada através do ImageJ é realizada a partir de imagens obtidas de folhas infectadas. Essas imagens geralmente são capturadas em um ambiente controlado (Figura 4). Por exemplo, a luminosidade do ambiente deve ser adequada de forma que não gere sombra na imagem. Neste caso a folha, geralmente, é posicionada sobre um fundo branco, para que exista um contraste entre as cores do fundo e da folha. Além de ser fundamental a existência de uma escala de tamanho conhecido presente na imagem.

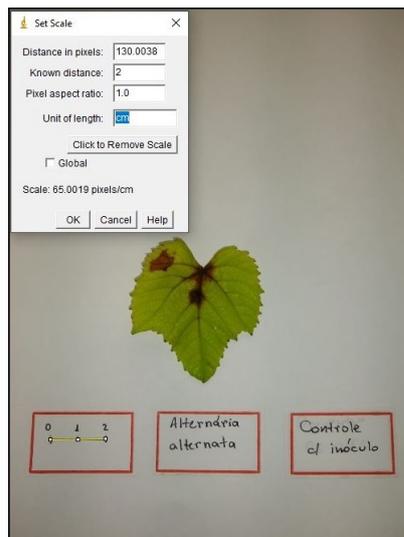
Figura 4 – Imagem obtida em um ambiente controlado



Fonte: Elaborado pelo autor

A escala presente na imagem é utilizada para estimar a área representada por cada *pixel* em uma unidade de medida conhecida. É com base nesta escala que é possível determinar a área foliar afetada em uma imagem. No caso do *software* ImageJ, deve-se traçar uma reta entre dois pontos de tamanho conhecido, e então informar a distância exata entre eles e a unidade de medida correspondente (Figura 5).

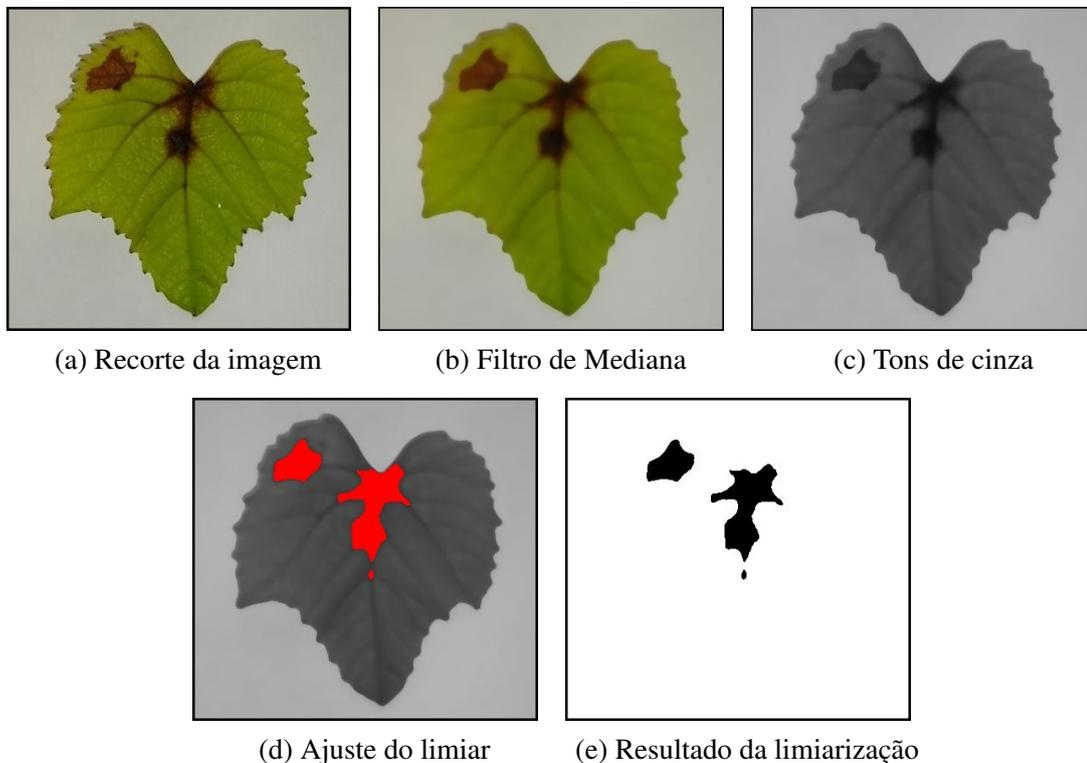
Figura 5 – Escala utilizada para o cálculo



Fonte: Elaborado pelo autor

Uma vez definida a escala, é possível remover da imagem, as partes que não representam a folha, mantendo somente a área necessária para a realização do cálculo (Figura 6a). Este processo é composto por algumas etapas para uma posterior segmentação da imagem¹. A primeira delas é a aplicação de um filtro de mediana, que é responsável por remover possíveis ruídos da imagem (Figura 6b). Após a imagem é convertida de um formato colorido para uma escala de tons de cinza (Figura 6c). Esta etapa é realizada de forma a facilitar a etapa de limiarização, que consiste em definir um valor limiar, em que todos os *pixels* acima desse valor são convertidos para preto e os pixels abaixo, para branco. No caso do *software* ImageJ, este valor limiar deve ser ajustado manualmente (Figura 6d). O resultado da limiarização é uma imagem contendo somente as áreas afetadas da folha (Figura 6e).

Figura 6 – Segmentação da imagem

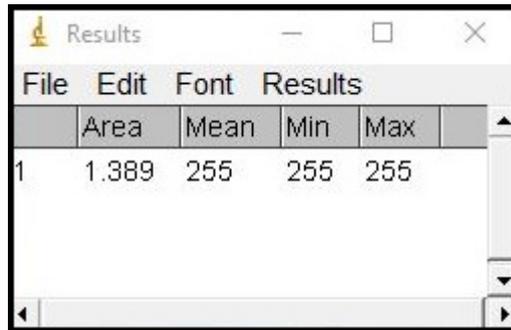


Fonte: Elaborado pelo autor

Por fim, o ImageJ realiza o cálculo das áreas que restaram na imagem (área preta da Figura 6e). Com base na definição de escala e da unidade de medida que foram informados no início do processo, este apresenta como resultado o tamanho da área foliar afetada pela doença (Figura 7).

¹ Segmentação é o processo utilizado em imagens, para isolar e extrair somente regiões relevantes para a análise e resolução de um problema. (YOUNG; GERBRANDS; VLIET, 1998)

Figura 7 – Resultado da área calculada



	Area	Mean	Min	Max
1	1.389	255	255	255

Fonte: Elaborado pelo autor

O processo descrito não necessita de nenhum investimento em equipamentos ou softwares, uma vez que o ImageJ é um software disponibilizado gratuitamente. Porém, quando se trata de um elevado número de imagens a serem analisadas, este processo torna-se lento e propício a erros. Dentro deste contexto, neste trabalho foi desenvolvido um aplicativo para dispositivos móveis, que possibilita automatizar esse processo.

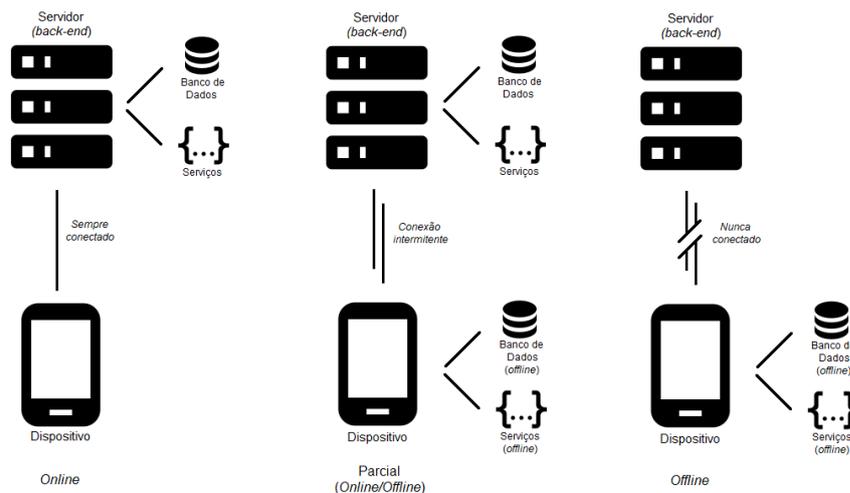
3 APLICATIVOS MÓVEIS

Para ser que um dispositivo seja considerado móvel, este deve possuir características específicas que representam a mobilidade. Essas características podem ser definidas como portabilidade, usabilidade, funcionalidade e conectividade. De fato, um dispositivo móvel deve ser portátil a ponto de ser facilmente transportado pelo usuário. Da mesma forma, esse dispositivo deve ser utilizável e funcional para que compreenda as inúmeras aplicações sendo acessadas por diferentes tipos de pessoas e em diversos ambientes. Por fim, este deve permitir uma facilidade de conexão com outros sistemas, proporcionando a troca de informações entre eles (LEE; SCHNEIDER; SCHELL, 2005).

3.1 CONECTIVIDADE EM APLICATIVOS MÓVEIS

Quando se trata de conectividade, existem três modos possíveis de comunicação entre uma aplicação móvel e servidores de *back-end*¹ (Figura 8). Um dispositivo ou aplicativo móvel, pode estar operando de maneira *online*, ou seja, em constantemente comunicação com o sistema de *back-end*. Em contrapartida, o mesmo pode estar operando totalmente *offline*, sem nenhuma conexão com o sistema de *back-end*. Por fim, pode haver uma comunicação intermitente entre ambos, mantendo a aplicação parcialmente conectada com os servidores (LEE; SCHNEIDER; SCHELL, 2005).

Figura 8 – Modos de conectividade de um aplicativo



Fonte: Elaborado pelo autor

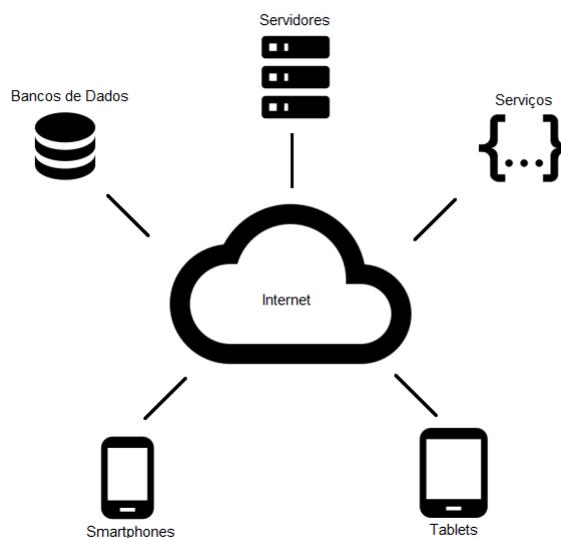
¹ *Back-end* é a camada de aplicação executada por trás da *interface*. Ou seja, essa não tem contato direto com o usuário, ficando responsável pelas regras de negócio, *webservices*, APIs e banco de dados.

Devido aos requisitos de mobilidade, principalmente pelos seus pesos e tamanhos reduzidos, os dispositivos móveis acabam sofrendo algumas limitações técnicas, como por exemplo, capacidade de processamento, consumo de bateria, espaço para armazenamento e disponibilidade de conexão (COSTA, 2015). Uma alternativa para a solução desses problemas é a utilização de uma arquitetura *Mobile Cloud Computing* (QURESHI *et al.*, 2011).

3.2 MOBILE CLOUD COMPUTING

Uma *Mobile Cloud Computing* consiste basicamente em centralizar e disponibilizar os recursos computacionais (armazenamento, processamento de dados, etc) como servidores de *back-end* (Figura 9). Assim, esses recursos são disponibilizados remotamente pela internet e podem ser acessados pelos aplicativos móveis através de uma API (*Application Programming Interface*) (COSTA, 2015). Desta forma, os aplicativos passam a ter uma conexão intermitente com serviços de *back-end*, realizando uma comunicação sempre que for necessário a utilização de um desses recursos.

Figura 9 – Arquitetura Mobile Cloud Computing



Fonte: Elaborado pelo autor

Um dos principais objetivos para a utilização de um modelo MCC é mover todo o processamento e armazenamento de informações do dispositivo móvel, para uma nuvem computacional que possui uma infraestrutura melhor e um maior poder de processamento. Neste caso, os dispositivos móveis são utilizados somente para a apresentação dos elementos visuais do *front-end*² e para a captura dos dados necessários para o funcionamento *offline*. A partir da utilização deste modelo é possível melhorar o desempenho e reduzir o consumo de bateria

² *Front-end* é a camada de aplicação responsável pelas interfaces e interação direta com o usuário. Nela estão disponíveis os elementos gráficos e visuais responsáveis pela captura e exibição dos dados que são processados pelo *back-end*.

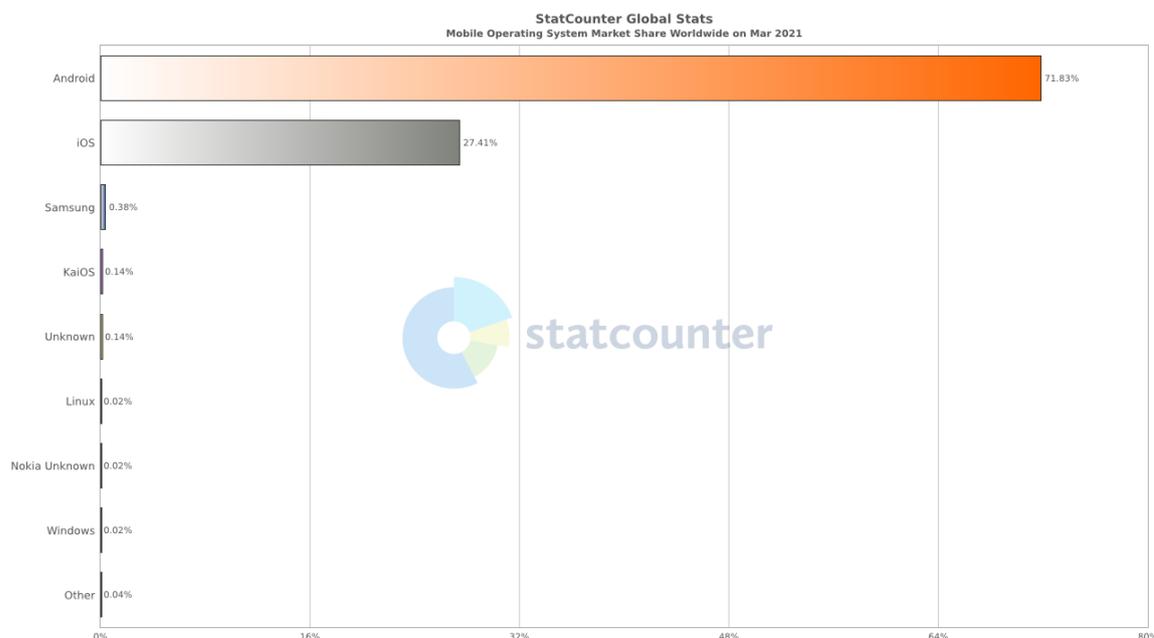
dos dispositivos móveis, principalmente de aplicações que estendam capacidades cognitivas do usuário, como aprendizado de máquina, realidade aumentada, reconhecimento de voz e visão computacional (SATYANARAYANAN *et al.*, 2009).

Uma outra vantagem do modelo MCC é uma maior capacidade de portabilidade (QURESHI *et al.*, 2011). Uma vez que o mercado de *smartphone* é compreendido por diversos sistemas operacionais, é fundamental que uma aplicação móvel esteja disponível para a maioria dessas tecnologias. Utilizando um modelo MCC, pode-se centralizar todo o desenvolvimento dos serviços de *back-end* em uma única aplicação disponível em uma nuvem computacional, onde os aplicativos desenvolvidos para diferentes sistemas operacionais podem acessá-la. A partir da utilização dessa arquitetura, tem-se única implementação *back-end*, sendo necessário somente desenvolver uma aplicação *front-end* para cada sistema operacional.

3.3 DESENVOLVIMENTO DE APLICATIVOS MÓVEIS

O mercado de *smartphones* mundial é dominado pelo sistema operacional Android, porém este está longe de ser uma unanimidade, uma vez que o sistema iOS da Apple detém uma grande fatia do mercado, como pode ser observado na Figura 10. Desta forma, torna-se atrativo o desenvolvimento de aplicativos que possam ser executados em ambos os sistemas operacionais. Para isso, existem dois modelos de desenvolvimento que são o nativo e o multiplataforma.

Figura 10 – Participação no Mercado de Sistemas Operacionais Móveis no Mundo (Mar 2021)



Fonte: (STATCOUNTER, 2021)

3.3.1 Desenvolvimento Nativo

O desenvolvimento nativo consiste em desenvolver uma aplicação exclusiva para uma determinada plataforma móvel, utilizando os seus próprios *frameworks*, *Software Development Kit* (SDK) e linguagens de programação. Os aplicativos desenvolvidos utilizando recursos nativos ficam diretamente atrelados a essa plataforma, tendo-se assim a necessidade de um novo desenvolvimento a cada nova plataforma a ser atendida (EL-KASSAS *et al.*, 2017).

Ao optar pelo desenvolvimento nativo tem-se acesso a todos os recursos, de baixo e alto nível, disponibilizados pelo sistema operacional. Desta forma, é possível realizar um melhor aproveitamento dos recursos e obter vantagens devido a facilidade de acesso às funcionalidades internas do sistema operacional, como câmera, áudio, geolocalização e microfone (EL-KASSAS *et al.*, 2017). Segundo ROUSE (2018) os aplicativos nativos oferecem uma experiência mais agradável e familiar para os usuários. Nestes, as interfaces gráficas seguem os padrões disponibilizados por cada sistema operacional e o desempenho da aplicação se torna mais rápido e intuitivo.

A grande desvantagem do desenvolvimento nativo, é o fato do aplicativo ficar restrito somente ao sistema operacional para o qual foi projetado. Para a utilização deste aplicativo em uma nova plataforma, torna-se necessário iniciar um novo desenvolvimento, elevando os custos de manutenção e desenvolvimento. Desta forma, tem-se um custo mais elevado para atender todas as plataformas de desenvolvimento (JUNTUNEN; JALONEN; LUUKKAINEN, 2013).

3.3.2 Desenvolvimento Multiplataforma

O desenvolvimento em multiplataformas possibilita a criação de uma única aplicação, que pode ser executada em mais de um sistema operacional. Neste caso, a aplicação deixa de estar diretamente vinculada a um único sistema operacional (EL-KASSAS *et al.*, 2017).

Uma das abordagens mais utilizada para o desenvolvimento de aplicações multiplataformas é denominada de *WebApps*, que são aplicações desenvolvidas para plataformas *web*, utilizando tecnologias como HTML, CSS e JavaScript. Essas, se caracterizam por não serem instaladas nos próprios dispositivos, sendo acessadas através de um navegador. A aparência e as funcionalidades dessas aplicações se assemelham a um aplicativo nativo, porém não possuem o mesmo desempenho (TAPPAREL, 2013).

Uma outra alternativa para o desenvolvimento de aplicativos multiplataformas são as aplicações híbridas. Os aplicativos híbridos também são desenvolvidos utilizando tecnologias *web*, porém são compilados para uma linguagem nativa de cada sistema operacional. Desta forma, têm-se um único código, a partir do qual os vários aplicativos são gerados. As aplicações híbridas não são totalmente nativas nem exclusivamente *web*, essas muitas vezes se caracterizam por serem uma aplicação *web* encapsulada em uma aplicação nativa (TAPPAREL, 2013).

Para o desenvolvimento multiplataforma de aplicativos móveis, existem diversos *frameworks* de desenvolvimento, sendo que entre esses destacam-se o Ionic (IONIC, 2020), o React Native (FACEBOOK, 2020), o Xamarin (MICROSOFT, 2020) e o Flutter (GOOGLE, 2020). As principais vantagens oferecidas pelo uso desses *frameworks* é uma maior velocidade de desenvolvimento. Isto se dá pelo fato de ser necessário desenvolver apenas um código fonte, através do qual é gerado o código para todas as demais plataformas. Assim, é necessário que os desenvolvedores possuam conhecimento somente sobre o *framework*, ao invés de possuir um conhecimento específico sobre o desenvolvimento para cada sistema operacional. Por fim, futuras implementações e possíveis manutenções de código, também se tornam mais ágeis pelo fato de existir somente um código fonte (MAHARANA, 2020).

Em contrapartida, existem algumas desvantagens na utilização de *frameworks*, sendo que uma das principais é uma maior dificuldade em criar interfaces que se adequem aos padrões e que funcionem em todas as plataformas. Além disso, os recursos de baixo nível do sistema operacional, como câmera, microfone e geolocalização, podem ser de difícil acesso, aumentando a complexidade do código. Por vezes, nem todas as funcionalidades dos dispositivos estão disponíveis para o *framework*, tendo-se a necessidade de utilizar ou desenvolver um *plugin* que realize essa funcionalidade. Por fim, o aplicativo gerado pode apresentar um desempenho inferior se comparado aos aplicativos desenvolvidos nativamente (MAHARANA, 2020).

4 TÉCNICAS DE PROCESSAMENTO DIGITAL DE IMAGEM

A proposta de solução adotada para este trabalho, consiste em uma sucessão de técnicas de processamento digital de imagens¹ com o objetivo de calcular a área foliar afetada pela doença. Nas próximas seções serão descritas as técnicas que foram utilizadas no desenvolvimento deste trabalho. Essas técnicas foram adotadas por já terem sido utilizadas com sucesso em trabalhos similares (JOAQUIM, 2019).

4.1 REMOÇÃO DE RUÍDOS

As imagens frequentemente são afetadas pela presença de ruídos, que podem ser causados por alterações na luminosidade e na utilização do *flash* no momento da captura, ou então, no caso específico deste trabalho, pela presença de veias nas folhas fotografadas (SINGH; MISRA, 2017). A remoção dos ruídos é importante uma vez que esses podem acabar afetando nas etapas posteriores e, conseqüentemente, nos resultados obtidos. Na Figura 11 tem-se um exemplo de uma imagem de uma folha onde é possível observar a presença de ruídos.

Figura 11 – Imagem com ruído



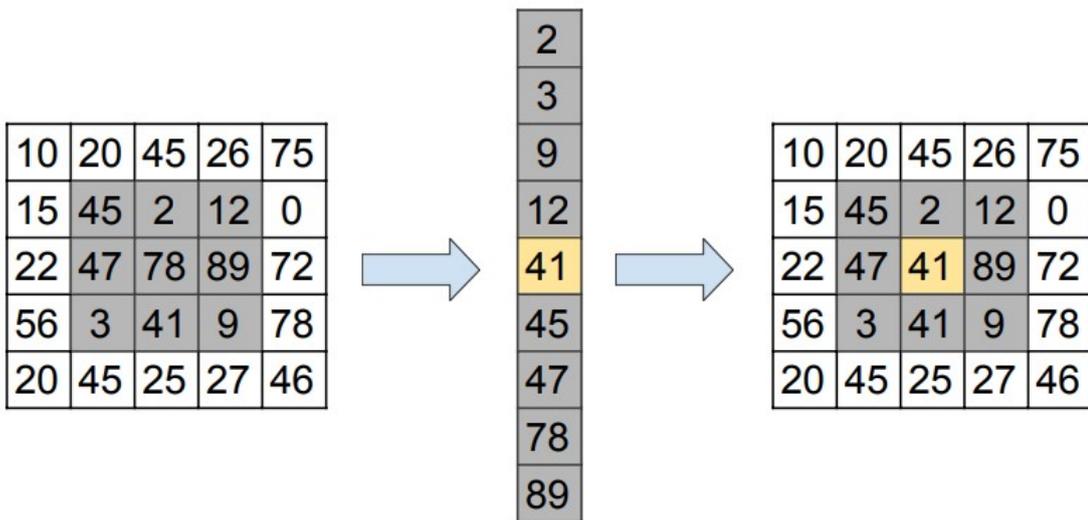
Fonte: Elaborado pelo autor

¹ Processamento digital de imagem é qualquer forma de manipulação, edição ou processamento de imagens realizadas por computador, onde a entrada do processo é uma imagem e a saída é essa mesma imagem alterada.

Para a remoção dos ruídos, optou-se pela utilização de um filtro de mediana, uma vez que essa técnica de filtragem mantém os aspectos importantes da imagem, como as bordas presentes no objeto, além de apresentar um borramento menor da imagem em comparação com outros métodos (SINGH; MISRA, 2017; VALLIAMMAL; GEETHALAKSHMI, 2012).

Um filtro de mediana consiste em substituir cada *pixel* da imagem, pelo valor da mediana² de seus vizinhos. O conjunto dos vizinhos é formado pelos *pixels* contidos em uma matriz de tamanho predeterminado, que possui como elemento central o *pixel* a ser substituído (GONZALEZ; WOODS, 2009). Na Figura 12 tem-se um exemplo de utilização de um filtro de mediana com uma matriz 3×3 .

Figura 12 – Exemplo de Filtro de Mediana 3x3

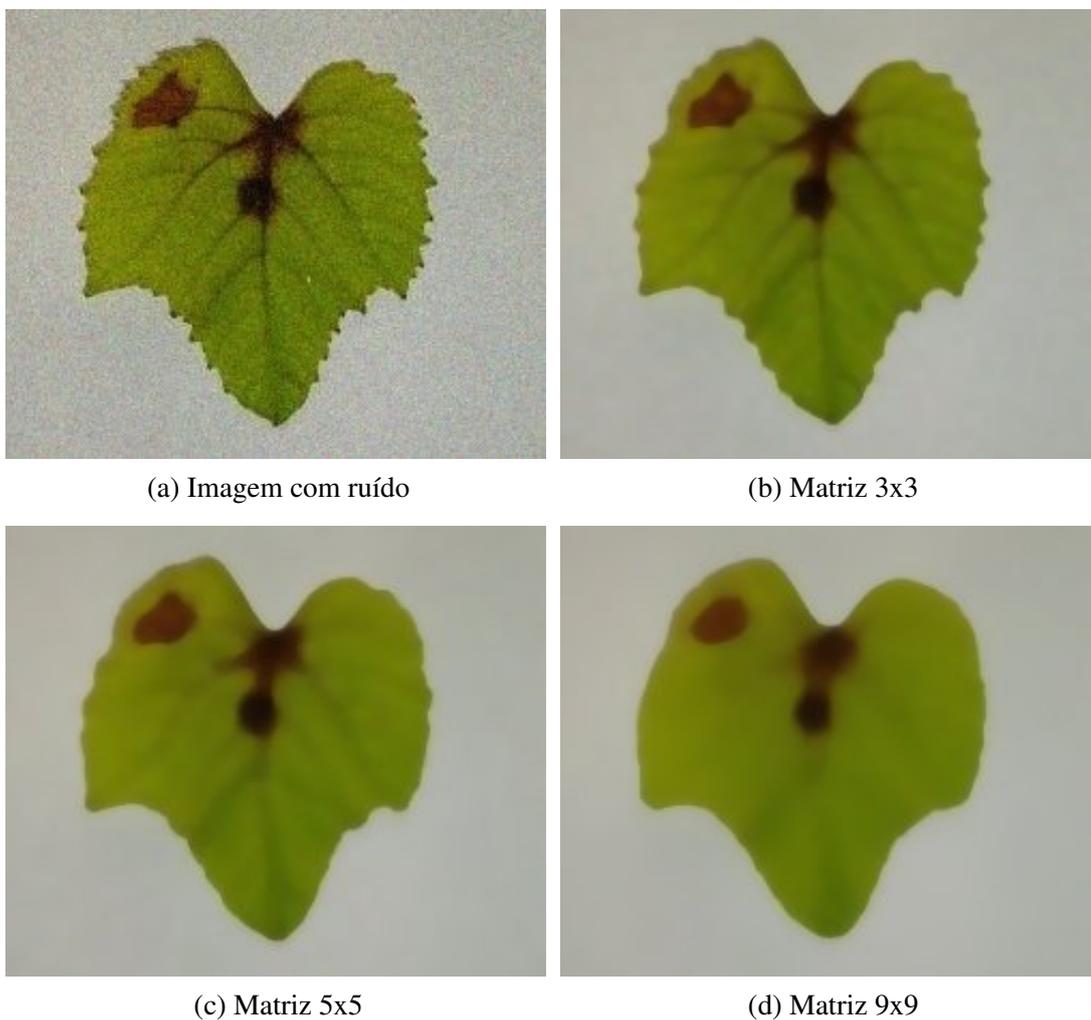


Fonte: Elaborado pelo autor

O tamanho da matriz de vizinhança utilizada tem uma influência direta nos resultados. Na Figura 13 tem-se o resultado da aplicação de um filtro mediana com matrizes de diferentes tamanhos. Como pode ser observado, a utilização de uma vizinhança muito pequena, preserva a qualidade da imagem, porém diminui a capacidade de filtragem. Já a utilização de uma vizinhança muito grande, pode ocasionar um borramento excessivo da imagem provocando uma perda de aspectos importantes, como por exemplo, a borda dos objetos (GONZALEZ; WOODS, 2009).

² A mediana de um conjunto corresponde ao valor onde, a metade dos valores do conjunto são menores ou iguais a mediana e a outra metade são maiores ou iguais a mediana (GONZALEZ; WOODS, 2009).

Figura 13 – Aplicação do Filtro de Mediana com diferentes matrizes de vizinhança



Fonte: Elaborado pelo autor

4.2 REMOÇÃO DO FUNDO

Em um processo de análise digital de imagens é fundamental identificar e extrair da imagem somente os objetos ou regiões que são relevantes para a solução do problema (YOUNG; GERBRANDS; VLIET, 1998). Desta forma, nesta etapa é removido o fundo da imagem, mantendo somente a folha afetada pela requeima foliar sobre um fundo totalmente branco (Figura 14). Para a remoção do fundo da imagem, optou-se pela utilização de uma técnica de limiarização (GONZALEZ; WOODS, 2009), uma vez que essa é uma técnica simples e já foi utilizada com sucesso em trabalhos similares (SINGH; MISRA, 2017; CHAUDHARY *et al.*, 2012; JOAQUIM, 2019).

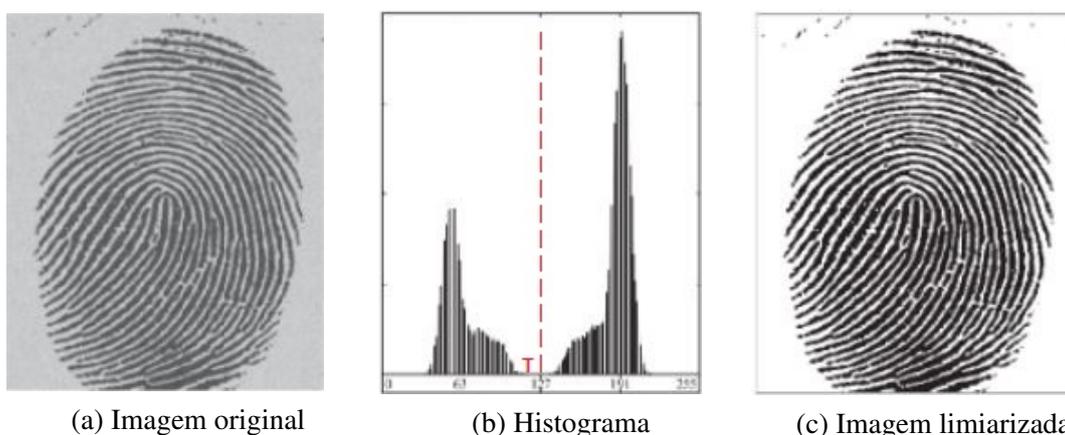
Figura 14 – Remoção do fundo de uma imagem



Fonte: Elaborado pelo autor

O método de limiarização é baseado no processamento do histograma³ de uma imagem em escala de cinza. A imagem de um objeto escuro sobre um fundo claro (Figura 15a), costuma gerar um histograma com dois picos bem definidos (Figura 15b), sendo possível determinar um valor limiar T entre esses dois picos. Desta forma, é possível separar os objetos do fundo da imagem determinando que os *pixels* com valores maiores que T são pertencentes aos objetos e os *pixels* com valores menores ou iguais a T são pertencentes ao fundo da imagem. O resultado da limiarização é uma imagem nas cores branca e preta (Figura 15c) (GONZALEZ; WOODS, 2009).

Figura 15 – Exemplo de limiarização

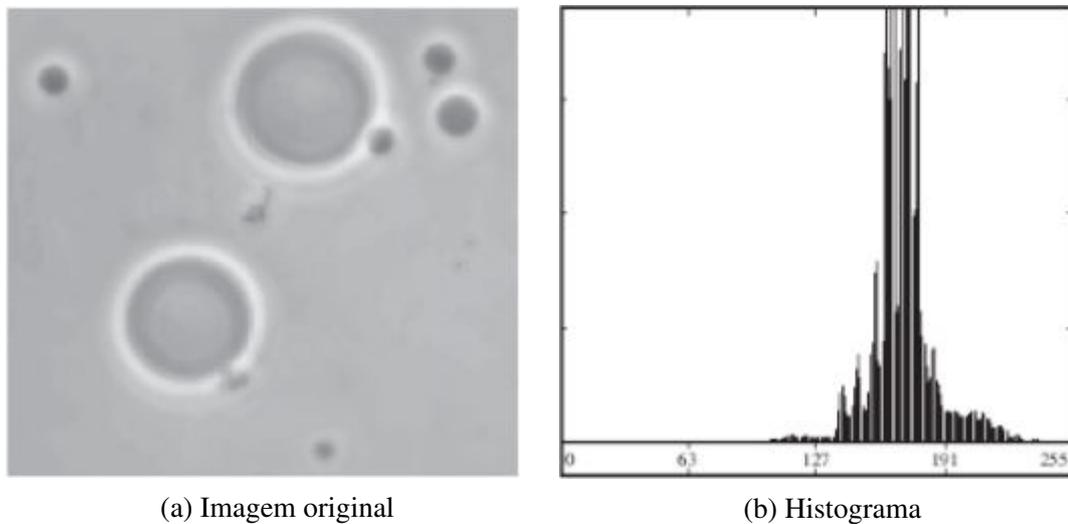


Fonte: Adaptado de (GONZALEZ; WOODS, 2009)

³ Em uma imagem em escala de cinza os *pixels* são representados por valores entre 0 e 255. O Histograma é a representação gráfica da quantidade de *pixels* para cada um desses valores presentes na imagem.

A determinação do limiar T e a distância entre os picos no histograma é muito importante para uma limiarização mais precisa. Porém, nem sempre os picos são bem definidos, o que pode representar um problema na escolha do limiar (GONZALEZ; WOODS, 2009). Isso ocorre principalmente em imagens que não apresentam um contraste entre o fundo e os objetos. Por exemplo, a Figura 16a apresenta uma imagem não segmentada e a Figura 16b o seu histograma. Neste caso, observa-se que o histograma não possui dois picos bem definidos e, conseqüentemente, tem-se uma maior dificuldade em se definir um limiar.

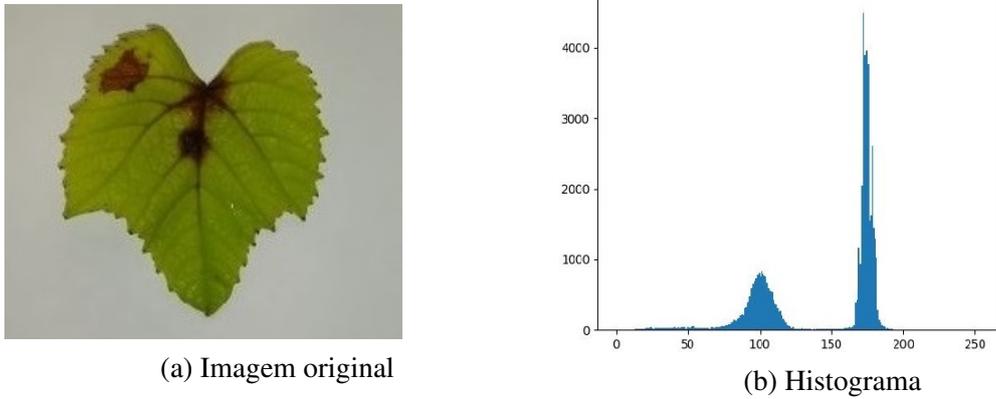
Figura 16 – Histograma sem picos bem definidos



Fonte: Adaptado de (GONZALEZ; WOODS, 2009)

As imagens utilizadas para o desenvolvimento deste trabalho foram capturadas em um ambiente controlado, onde as folhas foram posicionadas sobre um fundo claro (Figura 17a). Com isso, aumenta-se o contraste entre o fundo e a folha, proporcionando a geração de um histograma com dois picos bem definidos e facilitando na definição do limiar (Figura 17b).

Figura 17 – Histograma com dois picos definidos



Fonte: Elaborado pelo autor

Para a definição do limiar T e para a remoção do fundo da imagem optou-se pela utilização do método de limiarização de Otsu (CHAUDHARY *et al.*, 2012). Esse é um método que é executado sobre o histograma em escala de tons de cinza de uma imagem. Ou seja, considerando uma imagem com L tons de cinza, esse procedimento é realizado iterando sobre todos os possíveis valores de T entre 0 e $L - 1$, sendo escolhido o valor de T que minimiza a soma da variância intraclases da imagem⁴, onde uma das classes é formada pelos *pixels* com tons de cinza com valores entre 0 e T (fundo da imagem) e a outra formada pelos *pixels* com tons de cinza entre $T + 1$ e $L - 1$ (frente ou objetos) (TOROK; CONCI, 2016).

A variância intraclasse σ_W^2 para um determinado limiar T pode ser calculada através da equação

$$\sigma_W^2 = W_b(\mu_b - \mu_T)^2 + W_f(\mu_f - \mu_T)^2 \quad (4.1)$$

onde W_b e W_f são as probabilidades de um *pixel* pertencer a cada uma das classes, μ_b e μ_f são os valores médios das intensidades dos *pixels* de cada classe, e μ_T é a média total das intensidades da imagem. Os valores para W_b e W_f são calculados através das equações

$$W_b = \sum_{k=0}^{T-1} p_r(r_k) \quad (4.2)$$

$$W_f = \sum_{k=T}^{L-1} p_r(r_k) \quad (4.3)$$

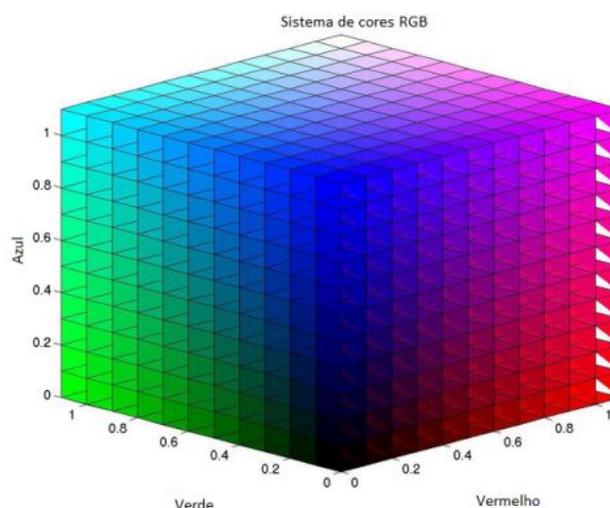
onde k é o nível de cinza e $p_r(r_k)$ é o percentual do k -ésimo nível de cinza (CORDEIRO, 2015).

⁴ Equivalente a maximizar a variância entre classes

4.3 TRANSFORMAÇÃO DO SISTEMA DE CORES

Após a remoção do fundo é realizado uma transformação do sistema de cores da imagem, que originalmente é representada pelo formato RGB, em um sistema que apresenta cores mais homogêneas, facilitando a segmentação (JOAQUIM, 2019). O sistema de cores RGB (Figura 18) é o formato de cores mais comum e utilizado, sendo composto por três cores, R (*red* - vermelho), G (*green* - verde) e B (*blue* - azul), de forma que, as demais cores do sistema, são geradas através de combinações dessas três (JOAQUIM, 2019).

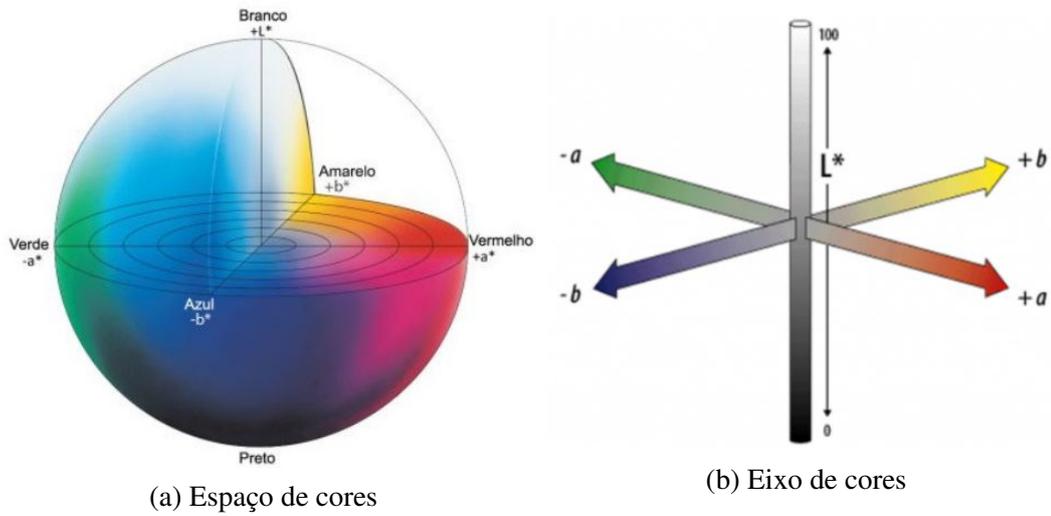
Figura 18 – Sistema de cores RGB



Fonte: (JOAQUIM, 2019)

No trabalho desenvolvido por CHAUDHARY *et al.* (2012) é realizada uma comparação entre os sistemas de cores YCbCr, HSI e CIELAB, para a identificação de áreas afetadas por doenças em folhas de plantas. Neste trabalho foi observado que o CIELAB é o sistema mais adequado para esse tipo de aplicação. Desta forma, esse foi o sistema de cores escolhido para o desenvolvimento deste trabalho. O sistema CIELAB (Figura 19) é representado em um espaço tridimensional, dividido em um eixo vertical e dois eixos horizontais. O eixo vertical l representa a iluminação ou escala de cinza da imagem. Já os eixos horizontais a e b , representam a variação entre as cores vermelha e verde e, azul e amarela, respectivamente (JOAQUIM, 2019).

Figura 19 – Sistema de cores CIELAB



Fonte: (JOAQUIM, 2019)

Ao ser realizada uma conversão do sistema de cores RGB para o sistema CIELAB, a imagem resultante apresenta uma variedade de cores menor e mais homogênea. De fato, esta conversão traduz as cores semelhantes no sistema RGB em uma mesma cor no sistema CIELAB. Desta forma, obtém-se uma imagem com uma maior uniformidade de cores, facilitando a etapa de segmentação (JOAQUIM, 2019). A Figura 20 apresenta um exemplo do resultado de uma conversão do sistema RGB para o sistema CIELAB.

Figura 20 – Conversão de RGB para CIELAB

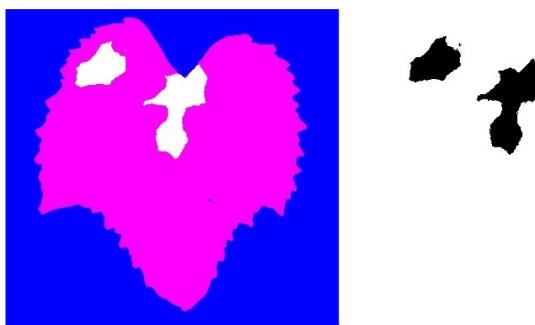


Fonte: (JOAQUIM, 2019)

4.4 SEGMENTAÇÃO DA ÁREA DA DOENÇA

Nesta etapa, é realizada uma segmentação da imagem, com o objetivo de separar a área afetada pela doença do restante da imagem. Ou seja, nesta etapa, são isolados somente os *pixels* relevantes para a realização do cálculo da área foliar afetada pela doença (Figura 21). Para a realização da segmentação optou-se pela utilização do método de clusterização *K-means* (YADAV; SHARMA, 2013; SAMAJPATI; DEGADWALA, 2015; JOAQUIM, 2019; BASHISH; BRAIK; BANI-AHMAD, 2011).

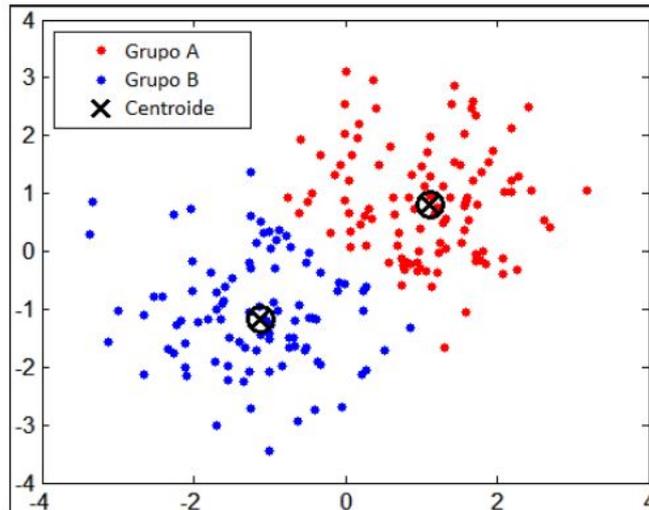
Figura 21 – Segmentação da doença



Fonte: Elaborado pelo autor

A clusterização tem como objetivo separar um determinado conjunto de dados em grupos conforme sua similaridade. Assim, os *pixels* de uma imagem que apresentam valores semelhantes são atribuídos a um mesmo grupo, e conseqüentemente, os *pixels* diferentes são atribuídos aos demais grupos (YADAV; SHARMA, 2013). Entre os métodos de clusterização existentes, um dos mais populares é o método *K-means*. Este se caracteriza por ser um método iterativo, de aprendizado não supervisionado, que tem por objetivo agrupar n dados em k grupos conforme suas similaridades. Cada um desses k grupos é representado por um centroide, sendo que o valor do centroide é definido pelo valor médio dos dados pertencentes ao grupo (Figura 22) (YADAV; SHARMA, 2013).

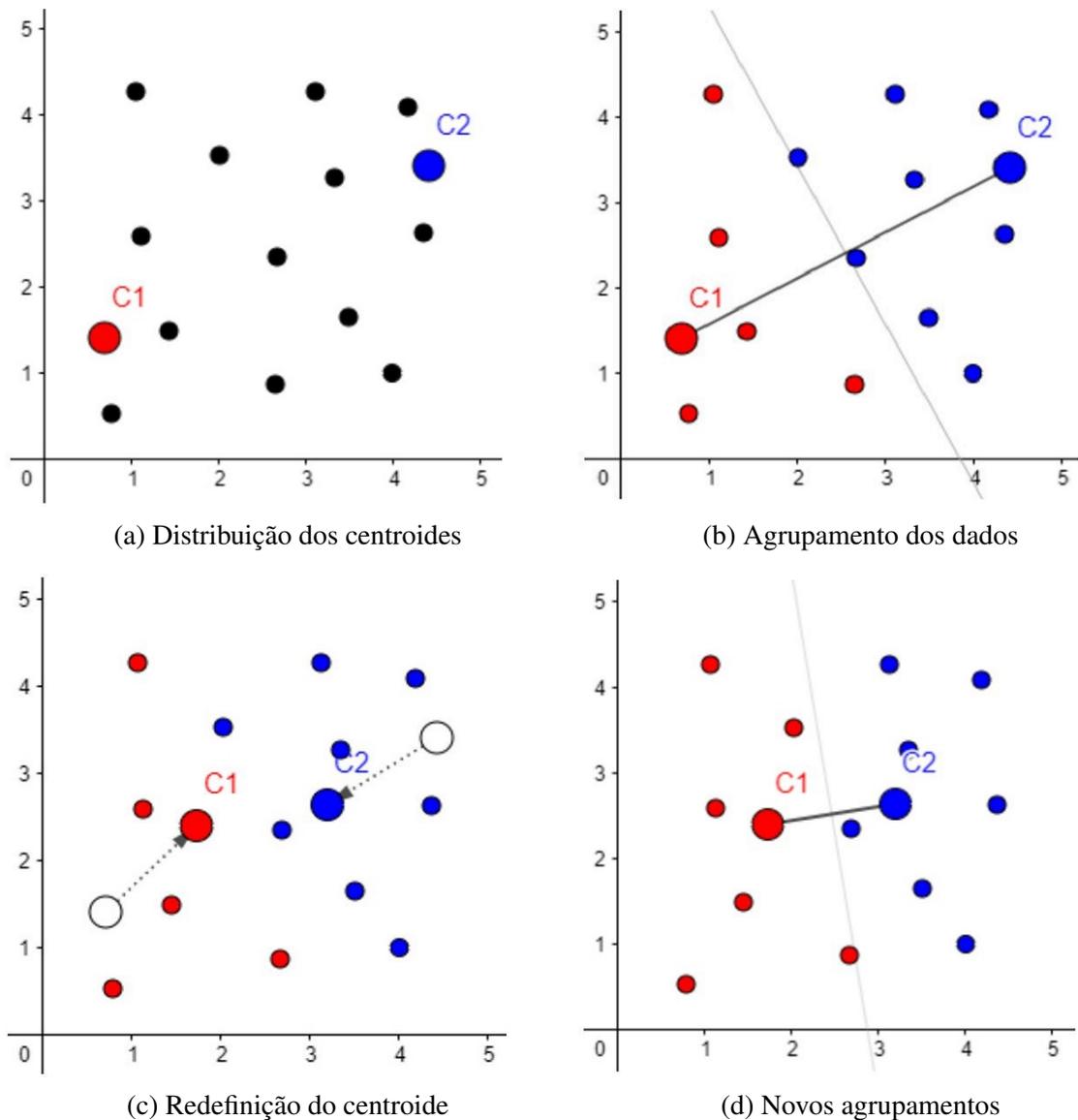
Figura 22 – Distribuição dos grupos por K-means



Fonte: (JOAQUIM, 2019)

Para a execução do método *K-means*, inicialmente deve-se distribuir aleatoriamente os k centroides no espaço representado pelo conjunto de dados (Figura 23a). Após esta distribuição, cada dado é associado ao grupo do centroide mais próximo, levando em consideração a distância *euclidiana*, por exemplo (MACHADO, 2017) (Figura 23b). Após, ao agrupamento dos dados, o valor de cada centroide é recalculado utilizando o valor médio dos dados pertencentes ao grupo (Figura 23c). Então, é realizada uma nova associação dos dados considerando os novos valores dos centroides (Figura 23d), e assim sucessivamente. Este processo é executado repetidamente até que nenhum dado seja associado a um grupo diferente (YADAV; SHARMA, 2013).

Figura 23 – Algoritmo K-means com 2 centroides



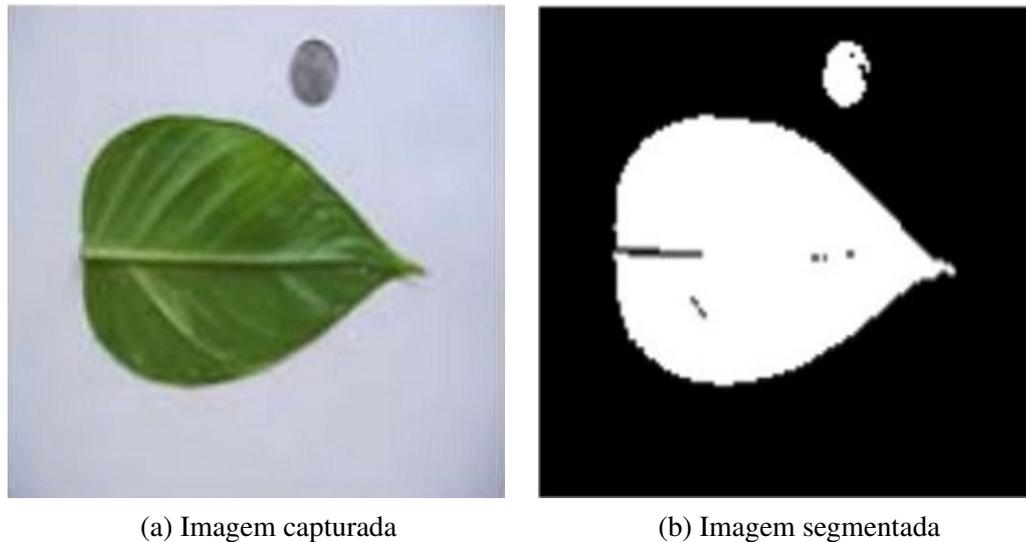
Fonte: Adaptado de (JOAQUIM, 2019)

4.5 CÁLCULO DA ÁREA AFETADA

Após a segmentação, restam na imagem, somente os *pixels* pretos que representam a área afetada pela doença. Nesta última etapa, o tamanho dessa área é determinado através da contagem desses *pixels*. Entretanto, para que seja possível calcular a área total afetada pela doença, é necessário, primeiro, estimar a área de um *pixel* da imagem (JOAQUIM, 2019).

No trabalho desenvolvido por PATIL; BODHE (2011), é apresentado um método para calcular a área de uma folha através da contagem de *pixels*, utilizando uma moeda como objeto de referência para determinar a área de um *pixel*. As imagens deste trabalho foram capturadas em um ambiente controlado, onde a folha foi posicionada juntamente com uma moeda de 2,5 cm de diâmetro sobre um fundo branco (Figura 24).

Figura 24 – Folha com moeda como objeto de referência



Fonte: (PATIL; BODHE, 2011)

Uma vez que o diâmetro da moeda é conhecido, é possível determinar a área (em cm^2) da moeda AM através da equação

$$AM = \pi(d/2)^2 \quad (4.4)$$

onde d é o diâmetro da moeda. A partir da área da moeda é possível estimar a área (em cm^2) de um *pixel* da imagem AP , através da equação

$$AP = AM/QPM \quad (4.5)$$

onde AM é a área da moeda e QPM é a quantidade de *pixels* que representam a moeda na imagem.

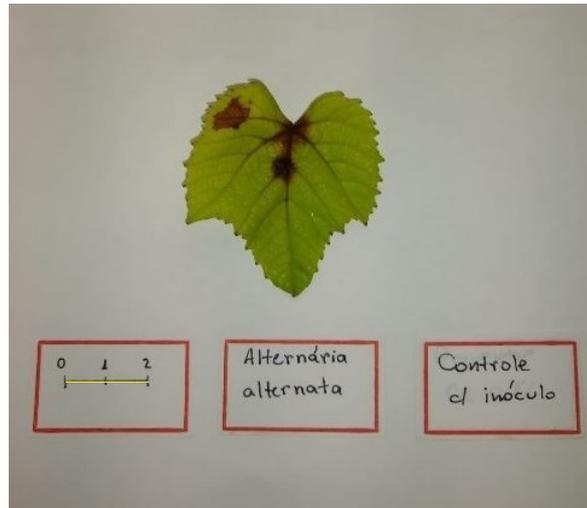
Após, determinada a área de um *pixel*, é possível então, calcular a área total (em cm^2) da folha AF através da equação

$$AF = AP * QPF \quad (4.6)$$

onde AP é a área estimada de um *pixel* e QPF é a quantidade de *pixels* pertencentes a folha (PATIL; BODHE, 2011);

As imagens utilizadas para o desenvolvimento deste trabalho foram capturadas em um ambiente controlado, onde foi inserido um escala de 2 cm em cada imagem (Figura 25). Essa escala pode ser usada como referência para determinar a área de um *pixel*. Através da seleção de dois pontos da escala e a indicação da distância entre eles é possível estimar a área do *pixel*.

Figura 25 – Imagem com escala como referência



Fonte: Elaborado pelo Autor

A partir da seleção de dois pontos na imagem, é possível estimar a quantidade de *pixels* existentes entre eles utilizando a distância Euclidiana através de

$$QPE = \sqrt{(P1X - P2X)^2 + (P1Y - P2Y)^2} \quad (4.7)$$

onde $P1$ e $P2$ são as coordenadas dos pontos e QPE é a quantidade de *pixels* entre os pontos.

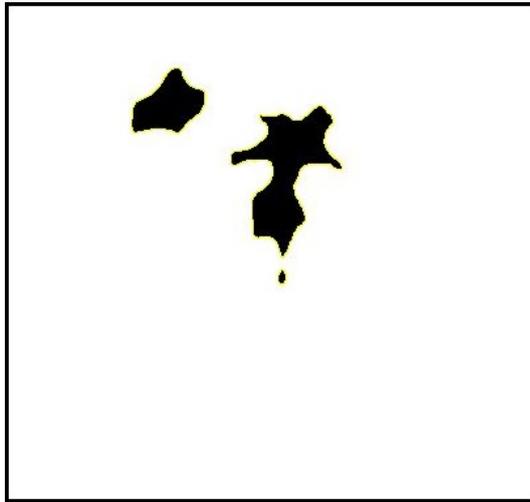
Uma vez conhecida a quantidade de *pixels* da escala, é possível determinar a área de um *pixel AP* através da equação

$$AP = (D/QPE)^2 \quad (4.8)$$

onde D é a distância informada da escala e QPE é a quantidade de *pixels* existentes no intervalo entre os dois pontos da escala (JOAQUIM, 2019).

O próximo passo para calcular a área foliar afetada é determinar a quantidade de *pixels* que fazem parte da área infectada. Como mencionado anteriormente, a imagem resultante da segmentação, é uma imagem somente com os *pixels* pretos que representam a área afetada pela doença (Figura 26). Dessa forma, a quantidade de *pixels* que representam a área afetada pela doença é determinada com a contagem dos *pixels* pretos presentes na imagem (JOAQUIM, 2019).

Figura 26 – Imagem da doença segmentada



Fonte: Elaborado pelo Autor

Por fim, é possível calcular a área afetada pela doença AAF através da equação

$$AAF = QPP * AP \quad (4.9)$$

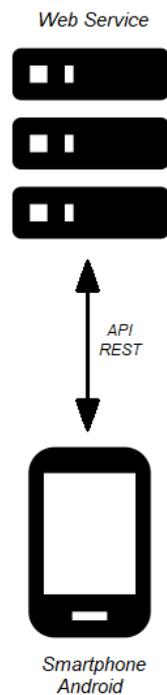
onde QPP é a quantidade de *pixels* pretos e AP é a área estimada de um *pixel* (JOAQUIM, 2019).

5 IMPLEMENTAÇÃO DESENVOLVIDA

Neste trabalho foi desenvolvido um aplicativo para dispositivos móveis, que realiza o cálculo da área foliar afetada pela requeima das folhas em videiras. Este aplicativo possibilita que o usuário selecione ou capture imagens das folhas afetadas através do seu *smartphone*. As imagens capturadas são enviadas a um servidor que executa as etapas de processamento da imagem para calcular a área foliar afetada pela doença. Após, os resultados são enviados para o *smartphone*, onde são apresentados para o usuário.

Para o desenvolvimento deste aplicativo optou-se por uma arquitetura onde a solução é dividida em duas partes. Um aplicativo móvel disponível para a plataforma Android, e um *Web Service*¹ sendo executado em um servidor na *web*. Esse *Web Service* é responsável pela execução de todas as etapas de processamento de imagens. Já o aplicativo móvel é responsável pela interface gráfica e toda interação com o usuário. As duas partes permanecem conectadas através de uma API REST (RICHARDSON; RUBY, 2008), sendo assim, é necessário que o dispositivo móvel possua uma conexão com a internet para ser utilizado. Na Figura 27 tem-se a arquitetura que foi utilizada para o desenvolvimento deste trabalho.

Figura 27 – Arquitetura de Comunicação entre o Aplicativo e o Web Service



Fonte: Elaborado pelo autor

¹ *Web Service* é um recurso utilizado para disponibilizar serviços de uma determinada aplicação, possibilitando a comunicação e troca de dados com outros sistemas, independente da plataforma ou linguagem de programação.

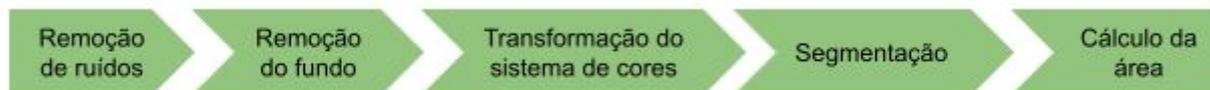
5.1 WEB SERVICE

O desenvolvimento de um *Web Service* proporciona a centralização do processamento digital das imagens em uma estrutura com um maior poder de processamento, melhorando assim o desempenho e reduzindo o consumo de energia dos dispositivos móveis. Além disso, como este serviço está disponível na *web*, é possível reaproveitá-lo em aplicativos para outras plataformas, como iOS, ou até mesmo em *softwares webs* ou *desktops*.

Para o desenvolvimento do *Web Service* optou-se pela linguagem de programação Python (LUTZ, 2013) devido, principalmente, à disponibilidade de um grande número de bibliotecas. Mais especificamente, foram utilizadas as bibliotecas: OpenCV (CULJAK *et al.*, 2012) para realizar o processamento das imagens, Numpy (OLIPHANT, 2006) para manipulação de matrizes, e Flask (GRINBERG, 2018) para disponibilizar a aplicação em um servidor *web*.

Como pode ser observado na Figura 28, para a obter o cálculo da área foliar afetada pela doença são realizadas as seguintes etapas de processamento de imagens: remoção de ruídos, remoção do fundo da imagem, transformação do sistema de cores, segmentação da área da doença e, por fim, o cálculo da área afetada.

Figura 28 – Fluxograma do processamento da imagem



Fonte: Elaborado pelo autor

Inicialmente, é utilizado um filtro mediana com o intuito de remover possíveis imperfeições e suavizar as bordas da imagem (GONZALEZ; WOODS, 2009). Para a remoção desses ruídos foi utilizado um filtro de mediana com matriz de vizinhança 3×3 . O resultado dessa etapa pode ser observado na Figura 29.

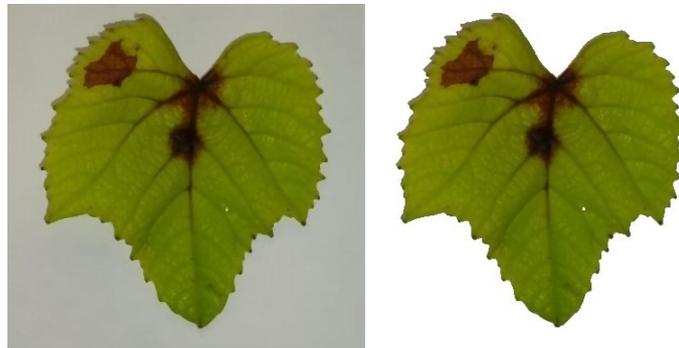
Figura 29 – Resultado da remoção de ruídos



Fonte: Elaborado pelo autor

Em uma próxima etapa é realizado um procedimento para remoção do fundo da imagem. Este processo tem por objetivo separar da imagem somente os objetos relevantes para a solução do problema. Para remoção do fundo, inicialmente, a imagem é convertida para uma escala de tons de cinza. Em seguida, é realizada a limiarização da imagem através do método de OTSU (CHAUDHARY *et al.*, 2012) fornecido pela biblioteca OpenCV (CULJAK *et al.*, 2012). O resultado desse é uma imagem binária, onde o fundo é representado pela cor branca e a folha pela cor preta. Por fim, todos os *pixels* que representam o fundo foram convertidos para branco na imagem original. Os *pixels* que representam a folha foram mantidos no seu formato colorido original. O resultado dessa etapa é uma imagem somente da folha sobre um fundo totalmente branco, como pode ser observado na Figura 30.

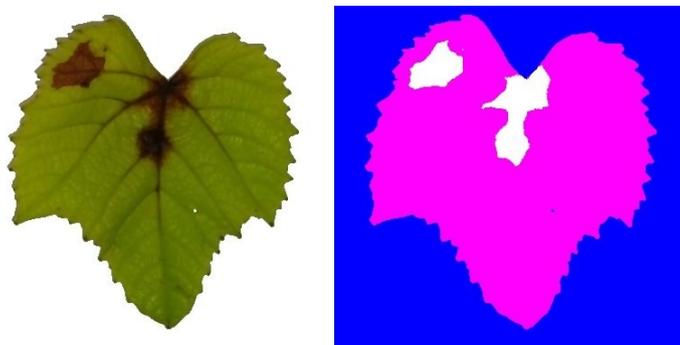
Figura 30 – Resultado da remoção de fundo



Fonte: Elaborado pelo autor

Após é realizado o processo de segmentação da doença. Neste caso, inicialmente é realizado uma transformação do sistema de cores, convertendo a imagem do formato RGB para o sistema de cores CIELAB. A imagem resultante apresenta cores mais homogêneas (Figura 31), facilitando a etapa de segmentação. O método de conversão utilizado foi desenvolvido utilizando as bibliotecas OpenCV (CULJAK *et al.*, 2012) e *Skimage* (WALT *et al.*, 2014).

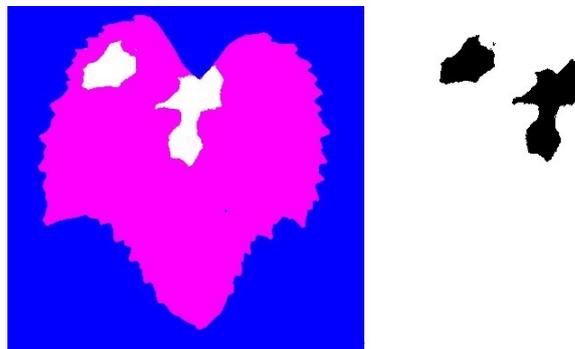
Figura 31 – Resultado da conversão para CIELAB



Fonte: Elaborado pelo autor

Posteriormente, sobre a imagem no formato CIELAB, é realizado o processo de segmentação da imagem extraindo somente a área representada pela doença. Para a segmentação da imagem foi utilizado o método *K-means* disponível na biblioteca OpenCV (CULJAK *et al.*, 2012). Inicialmente, foram definidos aleatoriamente 3 centroides, cada um representando uma cor da imagem. Então, o método foi executado com um critério de parada de no máximo 10 iterações. No final da execução, os *pixels* da imagem são agrupados em 3 grupos, sendo que os *pixels* do grupo que representa a área afetada são convertidos para preto, e os demais são convertidos para branco. Dessa forma, obtém-se uma imagem somente com a área afetada pela doença (Figura 32).

Figura 32 – Resultado da segmentação



Fonte: Elaborado pelo autor

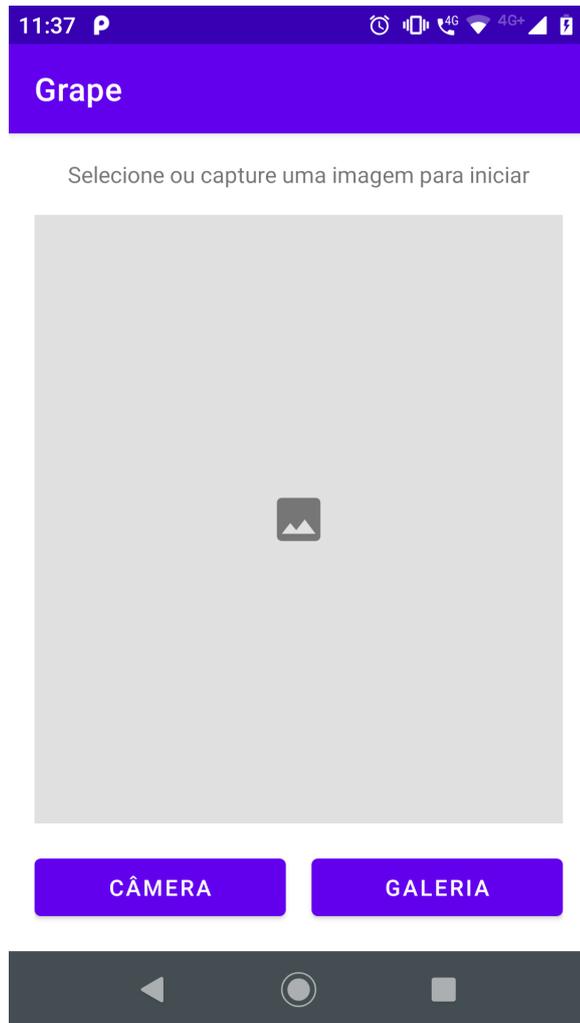
Uma vez definido os *pixels* que representam a área afetada pela doença, é possível determinar o tamanho dessa através da contagem desses *pixels*. Entretanto, primeiro é necessário estimar a área de um *pixel* (em cm^2). Esta estimativa é realizada através da escala que deve ser informada pelo usuário através do aplicativo. Para tanto, o usuário deve traçar uma linha reta entre dois pontos, informando a distância entre eles. A partir da definição da escala é possível determinar a área de um *pixel* e, posteriormente, a área afetada pela doença, através do método descrito na Seção 4.5.

5.2 APLICATIVO MÓVEL

O aplicativo foi desenvolvido utilizando o modo nativo para a plataforma Android. A escolha pelo desenvolvimento para a plataforma Android deve-se ao fato do sistema operacional Android predominar no mercado de dispositivos móveis, atingindo 71,83% dos usuários (STATCOUNTER, 2021), além de ser uma plataforma gratuita para desenvolvimento. Optou-se pelo desenvolvimento nativo pois esse proporciona um melhor aproveitamento dos recursos oferecidos pelo sistema operacional e uma facilidade de acesso às funcionalidades internas do dispositivo móvel (EL-KASSAS *et al.*, 2017), como por exemplo, a câmera que foi utilizada no desenvolvimento deste trabalho.

O objetivo do aplicativo móvel é possibilitar que o usuário selecione ou capture uma imagem da folha afetada pela doença diretamente do seu *smartphone*. Desta forma, a interface do aplicativo possibilita que o usuário acesse a galeria de fotos do usuário ou diretamente a câmera do *smartphone*, como pode ser observado na Figura 33.

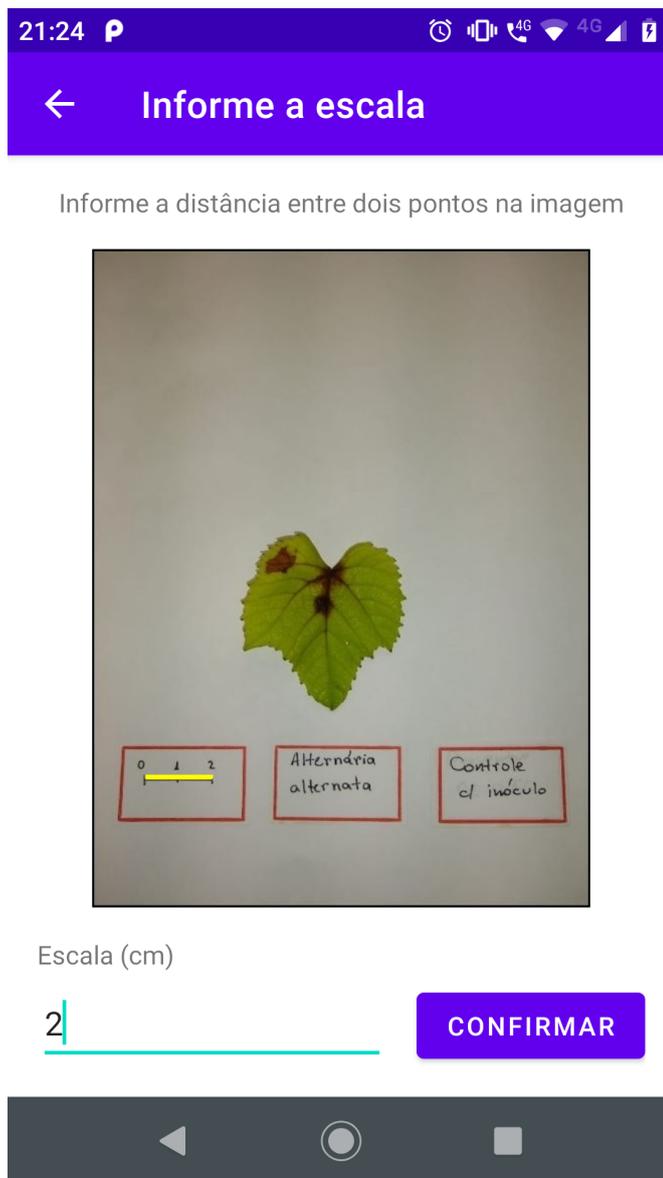
Figura 33 – Seleção da imagem



Fonte: Elaborado pelo Autor

Após, o usuário deve selecionar dois pontos na imagem e informar a distância (em *cm*) entre eles (Figura 34). Essa informação é necessária para que seja possível estimar a área de um *pixel*, que posteriormente será utilizado no cálculo da área afetada pela doença. As imagens utilizadas no desenvolvimento deste trabalho foram capturadas em um ambiente controlado, onde foi inserido uma escala de 2 cm em cada imagem. Neste caso, basta o usuário selecionar os dois extremos da escala e informar a distância de 2 cm. Qualquer outra imagem que venham a ser utilizada, deve ser capturada juntamente com um objeto de referência com tamanho conhecido, como por exemplo, uma régua, e informar a distância entre dois pontos quaisquer.

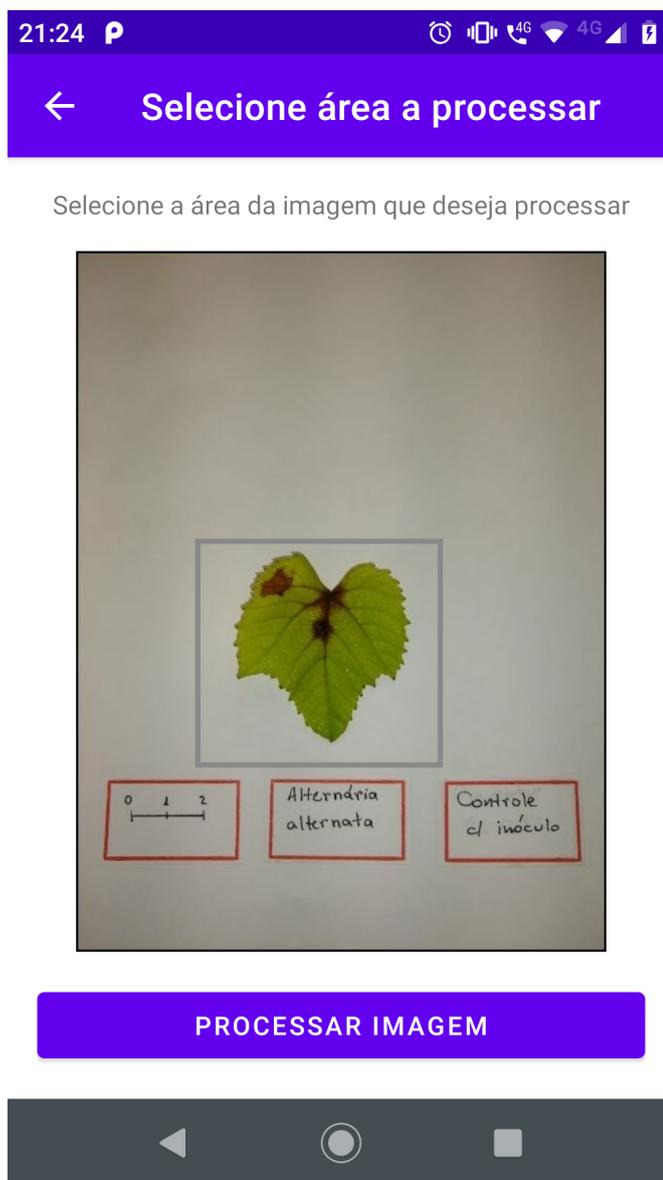
Figura 34 – Informar escala



Fonte: Elaborado pelo Autor

Após, o usuário pode selecionar uma sub-área da imagem que deseja processar e pressionar a ação "Processar Imagem" (Figura 35). Ao executar esta ação, as informações são enviadas para o *Web Service*, onde são realizadas as etapas de processamento da imagem. Ao final, os resultados obtidos são enviados para o aplicativo, onde são exibidos para o usuário.

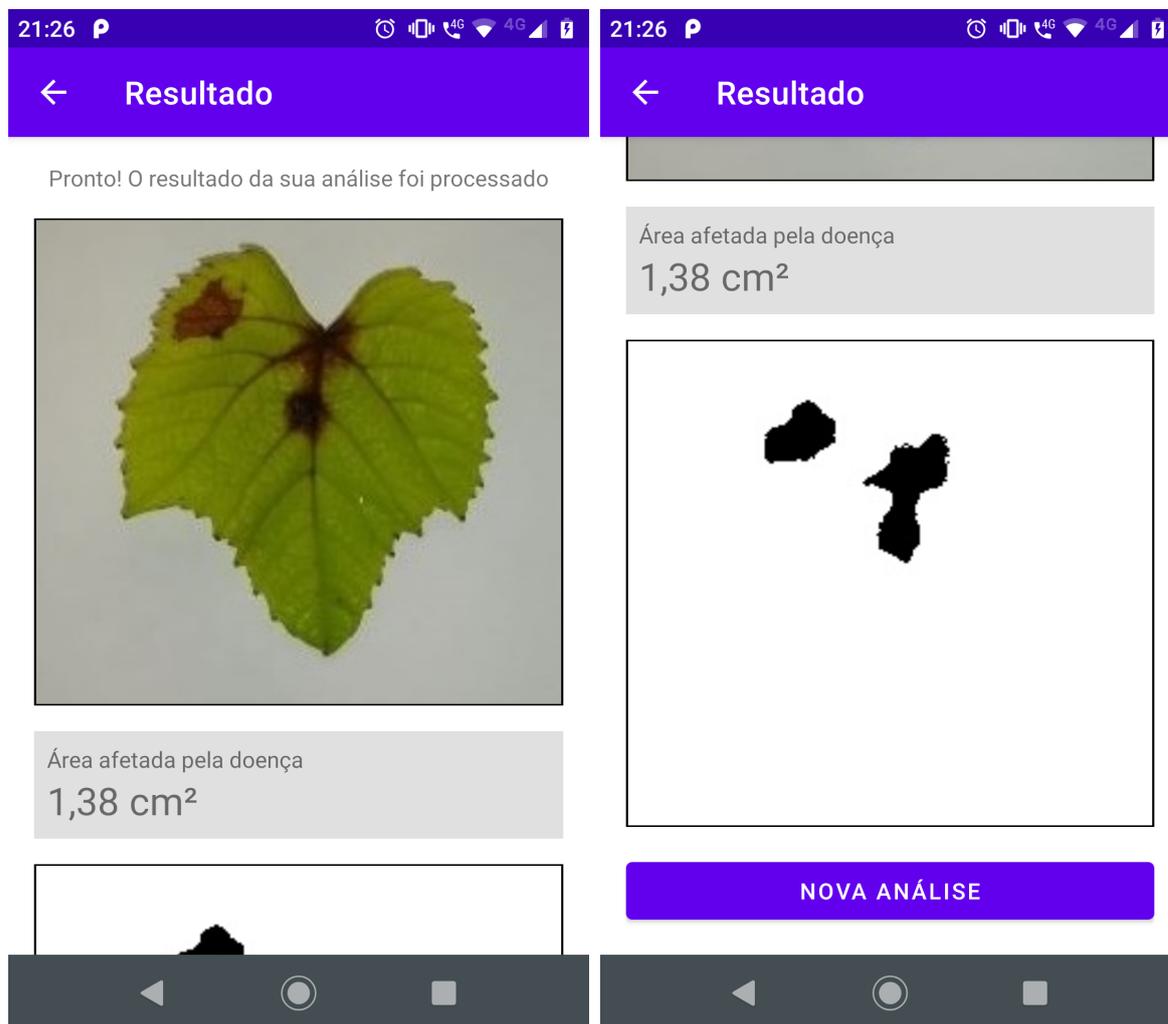
Figura 35 – Executar processamento da imagem



Fonte: Elaborado pelo Autor

Uma nova tela é criada para a apresentação dos resultados. Essa tela contém uma imagem representando a sub-área da imagem que foi processada, a imagem segmentada da doença (somente a parte que representa a doença na folha) e a área total (em cm^2) afetada pela doença. Na Figura 36 tem-se uma imagem com a tela de resultados. Nesta tela, é possível ainda que o usuário dê início ao processamento de uma nova imagem.

Figura 36 – Resultado da análise



Fonte: Elaborado pelo Autor

5.3 VALIDAÇÃO E AVALIAÇÃO DOS RESULTADOS

Para validação da implementação desenvolvida neste trabalho, foram utilizadas imagens disponibilizadas pela Prof. Dra. Joséli Schwambach do Laboratório de Controle Biológico Doenças de Plantas da Universidade de Caxias do Sul (UCS). Dentre as imagens disponibilizadas, foram selecionadas 41 imagens de folhas de videiras afetadas pela doença requeima foliar (*Alternaria Alternata*).

As imagens apresentam diferentes níveis de severidade da doença (entre 0 a 22%) e foram capturadas em um ambiente controlado. As folhas foram posicionadas sobre um fundo branco, gerando um contraste entre a folha e o fundo. Além disso, foi utilizada uma luminosidade adequada, de forma que não gere sombra sobre a imagem. As imagens possuem uma escala de 2 cm, que foi utilizada como referência para determinar a área de um *pixel*. Na Figura 37 é possível observar as imagens selecionadas para validação deste trabalho.

Figura 37 – Imagens selecionadas para validação



Fonte: Elaborado pelo autor

Inicialmente, a área afetada pela requeima das folhas de videira foi calculada utilizando o *software* ImageJ e esses resultados foram comparados aos resultados obtidos utilizando o aplicativo desenvolvido. Na Tabela 1 podem ser observados os resultados obtidos para cada uma das imagens ordenadas pelo nível de severidade da doença. Como pode ser observado, a diferença dos resultados está diretamente ligada ao nível de severidade da doença na folha. Em folhas com níveis de severidades menores (entre 0 e 1%), a quantidade de *pixels* que representam a área afetada é tão pequena que, qualquer variação mínima entre os métodos de validação, representa um percentual de diferença elevado. Neste caso, os resultados obtidos apresentam uma diferença de até 14,64%.

Tabela 1 – Comparação dos resultados obtidos

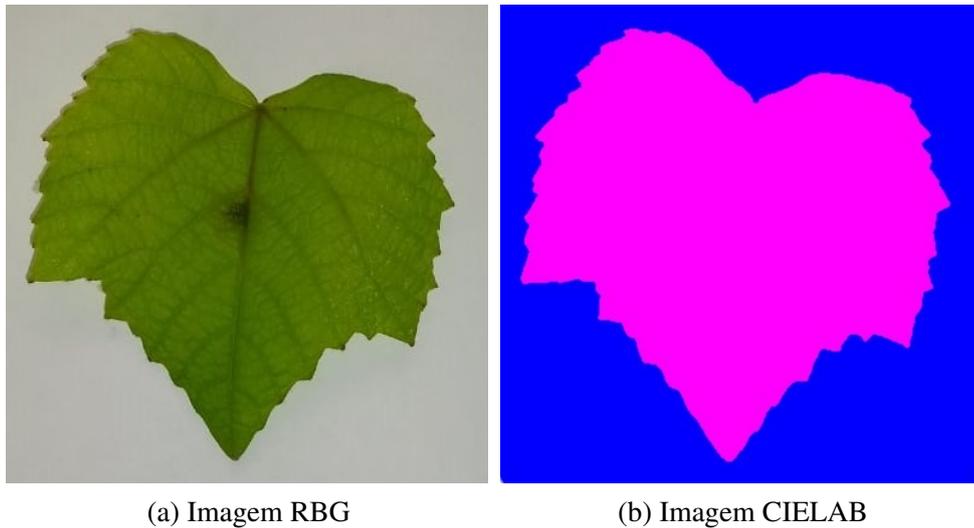
Folha	Severidade	Área obtida (cm ²)	Área ImageJ (cm ²)	Erro absoluto (cm ²)	Percentual de diferença
F01	0-1%	0,0170	0,019	0,0019	10,31%
F02	0-1%	0,0305	0,030	0,0005	1,66%
F03	0-1%	0,1464	0,170	0,0236	13,88%
F04	0-1%	0,0165	0,018	0,0015	8,33%
F05	0-1%	0,0394	0,042	0,0026	6,19%
F06	0-1%	0,1046	0,122	0,0174	14,26%
F07	0-1%	0,0405	0,043	0,0025	5,81%
F08	0-1%	0,0901	0,095	0,0049	5,15%
F09	0-1%	0,1139	0,133	0,0191	14,36%
F10	0-1%	0,0642	0,056	0,0082	14,64%
F11	0-1%	0,0913	0,107	0,0156	14,62%
F12	0-1%	0,2851	0,299	0,0139	4,64%
F13	0-1%	0,2719	0,276	0,0040	1,47%
F14	0-1%	0,1889	0,204	0,0150	7,37%
F15	0-1%	0,2044	0,194	0,0104	5,36%
F16	0-1%	0,2747	0,260	0,0147	5,65%
F17	1-2%	0,1793	0,178	0,0013	0,73%
F18	1-2%	0,2129	0,225	0,0121	5,37%
F19	1-2%	0,2524	0,258	0,0056	2,17%
F20	1-2%	0,2328	0,243	0,0102	4,19%
F21	1-2%	0,2916	0,293	0,0014	0,47%
F22	1-2%	0,1556	0,154	0,0016	1,03%
F23	1-2%	0,1254	0,122	0,0034	2,78%
F24	1-2%	0,1301	0,133	0,0028	2,12%
F25	1-2%	0,4753	0,458	0,0173	3,79%
F26	1-2%	0,1024	0,104	0,0016	1,53%
F27	1-2%	0,2495	0,262	0,0124	4,75%
F28	1-2%	0,3502	0,367	0,0167	4,55%
F29	1-2%	0,1257	0,123	0,0027	2,20%
F30	2-3%	0,4893	0,491	0,0016	0,32%
F31	2-3%	0,3290	0,324	0,0050	1,54%
F32	2-3%	0,2144	0,218	0,0036	1,65%
F33	2-3%	0,6709	0,659	0,0119	1,81%
F34	3-6%	0,8530	0,847	0,0060	0,70%
F35	3-6%	1,1482	1,148	0,0002	0,02%
F36	3-6%	0,6720	0,674	0,0020	0,29%
F37	6-10%	1,0260	1,021	0,0050	0,48%
F38	6-10%	0,9221	0,915	0,0071	0,77%
F39	6-10%	1,3590	1,367	0,0080	0,58%
F40	6-10%	2,4414	2,443	0,0015	0,06%
F41	10-25%	1,9280	1,935	0,0070	0,36%

Fonte: Elaborado pelo Autor

A diferença percentual diminui conforme o nível de severidade da doença vai aumentando. Por exemplo, para folhas com severidades entre 1 e 2%, a diferença máxima é de 5,37%. Já para severidades entre 2 e 3% a diferença não ultrapassa 1,81%. E para severidades acima de 3% a diferença é de no máximo 0,77%. Desta forma, pode-se considerar que os resultados obtidos são satisfatórios, principalmente, se for considerado o erro absoluto máximo, que em todos os casos não ultrapassa $0,0236 \text{ cm}^2$.

Por fim, observou-se que em imagens onde a cor da área afetada pela requeima é muito similar a cor da folha, ou então a área afetada é muito pequena (Figura 38a), não foi possível detectar a área afetada pela doença. Isso ocorre pois durante a transformação do sistema de cores RGB para o CIELAB (Figura 38b), não houve uma diferenciação entre as duas cores, impossibilitando a segmentação da área afetada pela doença.

Figura 38 – Folha com área afetada similar a cor da folha



Fonte: Elaborado pelo Autor

6 CONSIDERAÇÕES FINAIS

Neste trabalho foi desenvolvida uma solução para automatizar o processo de monitoramento da área foliar afetada pela requeima das folhas em videiras. O monitoramento e controle das doenças fúngicas em videiras é fundamental para a obtenção de frutos apropriados para comercialização. Os métodos tradicionalmente utilizados para realizar esse acompanhamento apresentam um elevado tempo de análise.

A solução desenvolvida utilizou-se de uma arquitetura dividida em duas partes: um aplicativo móvel para a plataforma Android e um *Web Service*. O *Web Service* proporcionou uma centralização de todas as operações de processamento de imagens em uma infraestrutura computacional com um poder de processamento maior. Além disso, é possível reaproveitar esse serviço no desenvolvimento de trabalhos futuros, como por exemplo, um aplicativo para a plataforma iOS ou até mesmo *softwares* para as plataformas *web* ou *desktop*.

Para o desenvolvimento do aplicativo móvel, optou-se pelo desenvolvimento nativo para a plataforma Android, uma vez que o desenvolvimento nativo proporciona um melhor aproveitamento e facilidade de acesso aos recursos internos do dispositivo móvel, como por exemplo, a câmera, que foi utilizada neste trabalho. Além disso, optou-se pelo desenvolvimento para a plataforma Android pois essa é gratuita para desenvolvimento e atinge 71,83% dos usuários do mercado.

Com relação às etapas de processamento digital da imagem, a utilização de um filtro de mediana, com uma matriz de vizinhança de 3x3, apresentou uma remoção adequada dos ruídos, mantendo os aspectos da imagem e evitando um borramento excessivo. Já para a remoção do fundo, o método de limiarização de Otsu apresentou resultados satisfatórios, principalmente pelo fato das imagens terem sido capturadas sobre um fundo totalmente branco, possibilitando a formação de dois picos bem definidos no histograma da imagem.

Para a segmentação da imagem, o método *K-means* foi utilizado pelo fato de ser um método de aprendizado não supervisionado, sem a necessidade de um treinamento prévio. Esse método demonstrou os melhores resultados após a imagem sofrer uma transformação para o sistema de cores CIELAB. De fato, a imagem resultante dessa transformação apresenta 3 cores bem distintas, facilitando a sua segmentação. Outros formatos de cores, como HSI, YCbCr e RGB, também foram testados, obtendo-se resultados inferiores ao CIELAB. No entanto, essa transformação mostrou-se ineficaz em imagens onde a cor da área afetada é muito similar a cor da folha.

Para a realização da contagem dos *pixels* e, conseqüentemente, determinação da área foliar afetada pela doença, foi utilizado um objeto (ou escala) de tamanho conhecido inserido nas imagens. Nas imagens selecionadas para a validação deste trabalho, foi utilizado, como

referência, uma escala de 2 cm inserida em cada imagem. Qualquer outra imagem que venha a ser utilizada, deve ser capturada juntamente com um objeto de tamanho conhecido como, por exemplo, uma régua. A determinação do tamanho desse objeto na tela do aplicativo, pode não ser tão exata, pelo fato do *touchscreen* não reconhecer exatamente o *pixel* que está sendo pressionado, mas sim, uma área envolvendo vários *pixels*. Desta forma, essa imprecisão na seleção dos *pixels* da escala pode provocar distorção no resultado.

Para a validação do aplicativo foram selecionadas 41 imagens de folhas de videiras afetadas pela requeima foliar, com diferentes níveis de severidade. Os resultados obtidos através do aplicativo desenvolvido neste trabalho foram comparados com os resultados obtidos pelo *software* ImageJ, obtendo-se uma diferença máxima de $0,0236 \text{ cm}^2$. A precisão dos resultados está diretamente ligado ao nível de severidade da doença. A diferença percentual entre os dois métodos diminui conforme o nível de severidade da doença vai aumentando. De fato, os melhores resultados foram obtidos em folhas com níveis de severidade acima de 3%, onde a diferença máxima foi de 0,77%. Embora este tenha sido o método utilizado para a validação, destaca-se que a solução desenvolvida neste trabalho pode ser aplicada para estimar a área foliar afetada por outras doenças. Por exemplo, essa solução poderia ser utilizada na estimativa da área afetada pelo Míldio e Ferrugem em videiras, além de doenças de outras plantas.

6.1 TRABALHOS FUTUROS

Como trabalhos futuros, sugere-se o desenvolvimento de um aplicativo para a plataforma iOS, abrangendo ainda mais o mercado de dispositivos móveis. Neste caso, somente o aplicativo iOS necessitaria ser desenvolvido, uma vez que o processamento digital das imagens está sendo disponibilizado por um *Web Service*.

Utilizar técnicas de *zoom* e posicionamento das imagens no aplicativo, com o objetivo de facilitar a definição da escala, aumentando a precisão do *touch* e, conseqüentemente, dos resultados. Ou ainda, desenvolver técnicas que permitam realizar uma estimativa automática da área de um *pixel*. Por exemplo, no caso de imagens com escala, poderia ser realizada uma detecção automática da escala.

Utilizar outras técnicas de processamento digital de imagem como, por exemplo, a detecção de bordas e remoção de sombras, a fim de possibilitar a realização da análise de imagens que não forem capturadas em um ambiente totalmente controlado.

REFERÊNCIAS

- BASHISH, D. A.; BRAIK, M.; BANI-AHMAD, S. Detection and classification of leaf diseases using k-means-based segmentation and. **Information technology journal**, v. 10, n. 2, p. 267–275, 2011.
- BOFF, P.; ZAMBOLIM, L.; VALE, F. Escalas para avaliação de severidade da mancha-de-estenfílio (*stemphylium solani*) e da pinta preta (*alternaria solani*) em tomateiro. **Fitopatologia Brasileira**, v. 16, n. 4, p. 280–283, 1991.
- BOTELHO, R. V.; PIRES, E. J. P. Viticultura como opção de desenvolvimento para os campos gerais. **Anais Encontro de Fruticultura dos Campos Gerais**, v. 1, p. 40–54, 2009.
- CHAUDHARY, P. *et al.* Color transform based approach for disease spot detection on plant leaf. **International journal of computer science and telecommunications**, Citeseer, v. 3, n. 6, p. 65–70, 2012.
- CORDEIRO, F. R. Desenvolvimento de um mecanismo semi-supervisionado para segmentação de tumores em imagens de mamografia digital. Universidade Federal de Pernambuco, 2015.
- COSTA, I. d. O. **Modelos par análise de disponibilidade em uma plataforma de mobile backend as a service**. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2015.
- CULJAK, I. *et al.* A brief introduction to opencv. In: IEEE. **2012 proceedings of the 35th international convention MIPRO**. [S.l.], 2012. p. 1725–1730.
- EL-KASSAS, W. S. *et al.* Taxonomy of cross-platform mobile applications development approaches. **Ain Shams Engineering Journal**, Elsevier, v. 8, n. 2, p. 163–190, 2017.
- FACEBOOK. **React Native - A framework for building native apps using React**. 2020. Disponível em: <<https://reactnative.dev/>>. Acessado em: 23 mai. 2020.
- GARRIDO, L. d. R. Cuidado com as doenças da videira. **Embrapa Uva e Vinho-Artigo em periódico indexado (ALICE)**, Revista Campo & Negócio HF, Uberlândia, p. 70-71, abr. 2015., 2015.
- GARRIDO, L. d. R.; GAVA, R. Manual de doenças fúngicas da videira. **Embrapa Uva e Vinho-Livro técnico (INFOTECA-E)**, Bento Gonçalves: Embrapa Uva e Vinho, 2014., 2014.
- GONZALEZ, R. C.; WOODS, R. C. **Processamento digital de imagens**. [S.l.]: Pearson Educación, 2009.
- GOOGLE. **Flutter - Beautiful native apps in record time**. 2020. Disponível em: <<https://flutter.dev/>>. Acessado em: 23 mai. 2020.
- GRINBERG, M. **Flask web development: developing web applications with python**. [S.l.]: "O'Reilly Media, Inc.", 2018.
- GUERRA, C. *et al.* **Sistema de produção de uvas rústicas para processamento em regiões tropicais do Brasil**. [S.l.]: Embrapa Uva e Vinho, 2005.

- IONIC. **Ionic - Cross-Platform Mobile App Development**. 2020. Disponível em: <<https://ionicframework.com/>>. Acessado em: 23 mai. 2020.
- JOAQUIM, B. M. Monitoramento da evolução da doença pinta preta em tomates através do processamento de imagens. 2019.
- JUNTUNEN, A.; JALONEN, E.; LUUKKAINEN, S. Html 5 in mobile devices—drivers and restraints. In: IEEE. **2013 46th Hawaii International Conference on System Sciences**. [S.l.], 2013. p. 1053–1062.
- KUHN, G. *et al.* **Cultivo da videira Niagara Rosada em regiões tropicais do Brasil**. [S.l.]: Embrapa Uva e Vinho, 2003.
- LEE, V.; SCHNEIDER, H.; SCHELL, R. **Aplicações móveis: arquitetura, projeto e desenvolvimento**. [S.l.]: Pearson Makron Books, 2005.
- LUTZ, M. **Learning Python: Powerful Object-Oriented Programming**. [S.l.]: O’Reilly Media, 2013. (Safari Books Online). ISBN 9781449355692.
- MACHADO, R. d. L. Desenvolvimento de um algoritmo imunológico para agrupamento de dados. 2017.
- MAHARANA, N. **Cross Platform Mobile Apps and Its Pros and Cons**. 2020. Disponível em: <<https://medium.com/andolasoft/cross-platform-mobile-apps-and-its-pros-and-cons-9c257ec64e94>>. Acessado em: 19 mai. 2020.
- MARTIN, T. N. *et al.* Uso do software imagej na estimativa de área foliar para a cultura do feijão. **Interciencia**, Asociación Interciencia, v. 38, n. 12, p. 843–848, 2013.
- MD MAGALHÃES PJ, R. S. A. Image processing with imagej. *biophotonics international* 11: 36-42. 2004.
- MELLO, L. M. R. de. Vitivinicultura brasileira: Panorama 2017. **Comunicado Técnico**, v. 207, n. 1, p. 11, 2018.
- _____. Vitivinicultura brasileira: Panorama 2018. **Comunicado Técnico**, v. 210, n. 1, p. 12, 2019.
- MICROSOFT. **Desenvolvimento de Aplicativos Xamarin com Visual Studio | Visual Studio**. 2020. Disponível em: <<https://visualstudio.microsoft.com/pt-br/xamarin/>>. Acessado em: 23 mai. 2020.
- OLIPHANT, T. E. **A guide to NumPy**. [S.l.]: Trelgol Publishing USA, 2006. v. 1.
- PATIL, S. B.; BODHE, S. K. Betel leaf area measurement using image processing. **International Journal on Computer Science and Engineering**, Citeseer, v. 3, n. 7, p. 2656–2660, 2011.
- PROTAS, J. d. S.; CAMARGO, U. A.; MELLO, L. de. A viticultura brasileira: realidade e perspectivas. In: IN: SIMPÓSIO MINEIRO DE VITICULTURA E ENOLOGIA, 1., 2002, ANDRADAS, MG . . . **Embrapa Uva e Vinho-Artigo em anais de congresso (ALICE)**. [S.l.], 2002.

- QURESHI, S. S. *et al.* Mobile cloud computing as future for mobile applications-implementation methods and challenging issues. In: IEEE. **2011 IEEE International Conference on Cloud Computing and Intelligence Systems**. [S.l.], 2011. p. 467–471.
- RAMOS, F. T. *et al.* Área do limbo foliar de diferentes plantas estimada por medidas lineares e matéria seca, calibradas com o software imagej. **Interciencia**, Asociación Interciencia, v. 40, n. 8, p. 570–575, 2015.
- RICHARDSON, L.; RUBY, S. **RESTful Web Services**. [S.l.]: O’Reilly Media, 2008. ISBN 9780596554606.
- RITSCHHEL, P.; SEBBEN, S. d. S. **Embrapa Uva e Vinho: novas cultivares brasileiras de uva**. [S.l.]: Bento Gonçalves: Embrapa Uva e Vinho, 2010., 2010.
- ROUSE, M. **What is native app?** 2018. Disponível em: <<https://searchsoftwarequality.techtarget.com/definition/native-application-native-app>>. Acessado em: 20 mai. 2020.
- SAMAJPATI, B. J.; DEGADWALA, S. D. A survey on apple fruit diseases detection and classification. **International Journal of Computer Applications**, Citeseer, v. 130, n. 13, p. 0975–8887, 2015.
- SATYANARAYANAN, M. *et al.* The case for vm-based cloudlets in mobile computing. **IEEE pervasive Computing**, IEEE, v. 8, n. 4, p. 14–23, 2009.
- SINGH, V.; MISRA, A. K. Detection of plant leaf diseases using image segmentation and soft computing techniques. **Information processing in Agriculture**, Elsevier, v. 4, n. 1, p. 41–49, 2017.
- STATCOUNTER. **Mobile Operating System Market Share Worldwide**. 2021. Disponível em: <<https://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-202103-202103-bar>>. Acessado em: 21 abr. 2021.
- TAPPAREL, Y. **Touch-Optimised Mobile Interface for Invenio Digital Library**. Tese (Doutorado) — Ecole Polytechnique, Lausanne, 2013.
- TAVARES, S. d. H.; LIMA, M. F. *et al.* Principais doenças da videira e alternativas de controle. **Embrapa Semiárido-Capítulo em livro científico (ALICE)**, In: LEAO, PC de S.; SOARES, JM (Ed.). A viticultura no Semi-Árido brasileiro . . . , 2000.
- TOROK, L.; CONCI, A. Método de otsu. **Instituto de Computação (UFF)**, 2016.
- VALLIAMMAL, N.; GEETHALAKSHMI, S. A novel approach for plant leaf image segmentation using fuzzy clustering. **International Journal of Computer Applications**, Citeseer, v. 44, n. 3, p. 10–20, 2012.
- WALT, S. van der *et al.* scikit-image: image processing in Python. **PeerJ**, v. 2, p. e453, 6 2014. ISSN 2167-8359. Disponível em: <<https://doi.org/10.7717/peerj.453>>.
- YADAV, J.; SHARMA, M. A review of k-mean algorithm. **International journal of engineering trends and technology**, Citeseer, v. 4, n. 7, p. 2972–2976, 2013.
- YOUNG, I. T.; GERBRANDS, J. J.; VLIET, L. J. V. Fundamentals of image processing. Delft University of Technology, 1998.