

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**SISTEMA DE VISÃO COMPUTACIONAL PARA  
RECONHECIMENTO E CLASSIFICAÇÃO DE PADRÕES  
DE FAMÍLIAS DE PLANTAS INVASORAS**

**BRUNO MORAES MORENO**

**ORIENTADOR: PROF. DR. PAULO ESTEVÃO CRUVINEL**

São Carlos

Junho, 2023

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**SISTEMA DE VISÃO COMPUTACIONAL PARA  
RECONHECIMENTO E CLASSIFICAÇÃO DE PADRÕES  
DE FAMÍLIAS DE PLANTAS INVASORAS**

**BRUNO MORAES MORENO**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Visão Computacional.

Orientador: Prof. Dr. Paulo Estevão Cruvinel

São Carlos

Junho, 2023



# UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia  
Programa de Pós-Graduação em Ciência da Computação

---

## Folha de Aprovação

---

Defesa de Dissertação de Mestrado do candidato Bruno Moraes Moreno, realizada em 08/06/2023.

### Comissão Julgadora:

Prof. Dr. Paulo Estevão Cruvinel (EMBRAPA)

Prof. Dr. Nelson Delfino D'Avila Mascarenhas (UFSCar)

Prof. Dr. Roberto de Alencar Lotufo (UNICAMP)

## AGRADECIMENTOS

---

---

Aos meus Pais, Deurivan Moreno Rodrigues e Morgana Gomes de Moraes Moreno, por todo o apoio e amor recebido e nas oportunidades a mim oferecidas, que me permitiram seguir na carreira acadêmica até este momento.

Ao Prof. Dr. Paulo Estevão Cruvinel, pela dedicação na orientação deste trabalho. Por apoiar e acreditar na realização deste projeto de pesquisa, e por me motivar em momentos que eu mesmo me duvidava. Pelos ensinamentos compartilhados.

A Dr. Augusto Guerreiro Fontoura Costa, Dr. Pedro Luis da Costa Aguiar Alves e Bruno Felipe da Silva, pelo auxílio na realização do experimento de campo. A Nícolas dos Santos Rosa, pelos encaminhamentos iniciais e recomendação para a pesquisa.

Aos meus colegas da Pós-Graduação, com quem tive a oportunidade de conviver durante o período do curso, sobretudo, os colegas Alex Bisetto Bertolla e Wilbur Naike Chiuyari Veramendi.

À Universidade Federal de São Carlos (UFSCar) e ao Programa de Pós-Graduação em Ciência da Computação (PPGCC), pela oportunidade de desenvolvimento profissional e pessoal. Aos Professores do programa, sempre dedicados ao ensino. Aos colaboradores do setor administrativo, sobretudo, ao Ivan sempre solícito à frente da secretaria do programa.

À Embrapa Instrumentação, pela oportunidade de desenvolvimento da pesquisa.

À CAPES (Processo 88887.488562/2020-00) pelo auxílio e apoio durante o curso.

À todos que, de alguma forma, contribuíram para a realização deste trabalho.

# RESUMO

A Visão Computacional, além de envolver técnicas do reconhecimento de padrões e classificação de objetos, vem se caracterizando como um campo emergente e de fundamental importância na contextualização da computação inteligente. Sua aplicação tem envolvido diferentes áreas do conhecimento, o que inclui a área agrícola. Desde os seus primórdios que envolveram a interpretação de manuscritos e textos digitados, atualmente encontra papel fundamental no que tange a produção de alimentos e energia sustentável. Esta pesquisa busca utilizar e desenvolver a Visão Computacional para a identificação e o reconhecimento localizado de plantas invasoras, as quais demandam controle para que se possa desenvolver adequadamente as culturas agrícolas. Tendo isto em perspectiva, pode-se utilizar algoritmos e conceitos de Visão Computacional e Internet das Coisas no auxílio de tal tarefa. É realizado então a construção de um sistema estéreo capaz de adquirir imagens digitais em campo, obter informação de profundidade, segmentar, extrair características e classificar plantas por intermédio de métodos supervisionados de aprendizado de máquina. Assim, o reconhecimento de padrões de famílias de plantas invasoras pode auxiliar no uso racional de insumos e minimizar impactos ambientais, entrando no rol de tecnologias da agricultura de precisão. Para o desenvolvimento do sistema foi validado *hardware* e *software* de controle embarcado, permitindo ação em tempo real e comunicação sem fio. As ações de processamento de sinais e imagens foram aplicadas em dois bancos de dados referente a culturas que são impactadas pela presença de plantas invasoras, a do milho (*Zea mays*) e a do amendoim (*Arachis hypogaea*). Para a etapa de pré-processamento, parâmetros de calibração de câmeras foram obtidos, para a retificação de imagens estéreo. Para a etapa de segmentação, foram performadas uma limiarização no espaço de cor HSV e operações morfológicas. As métricas de acurácia e área de sobreposição foram utilizadas para a validação do processo. Os mapas de disparidades foram obtidos a partir da avaliação de algoritmos de pareamento locais e semi-globais. Para a etapa de extração de características, foram aplicados descritores de Padrões Binários Locais e momentos de Haralick. Para a classificação das plantas invasoras, foram utilizados classificadores baseados em Máquinas de Vetores de Suporte. O sistema desenvolvido se mostrou capaz de gerar informação de profundidade e classificação de plantas, o que contribui na tomada de decisão e na aplicação em taxa variada de insumos. Os resultados obtidos comprovam a eficácia do método desenvolvido para reconhecimento, qualificação e classificação para suporte à decisão no controle de famílias de plantas invasoras.

**Palavras-chave:** Visão computacional; Visão estéreo; Reconhecimento de padrões; Classificação de plantas invasoras; Aprendizado de máquina.

# ABSTRACT

Computer Vision, in addition to involving pattern recognition and object classification techniques, has been characterized as an emerging field of fundamental importance in the context of intelligent computing. Its application has involved different areas of knowledge, which includes the agricultural area. Since its beginnings, which involved the interpretation of manuscripts and typed texts, it currently plays a fundamental role in the production of food and sustainable energy. This research seeks to use and develop Computer Vision for the identification and localized recognition of invasive plants, which require control in order to properly develop agricultural crops. With this in perspective, algorithms and concepts of Computer Vision and Internet of Things can be used to help with this task. Then, the construction of a stereo system capable of acquiring digital images in the field, obtaining depth information, segmenting, extracting features and classifying plants through supervised machine learning methods is carried out. Thus, the recognition of patterns of invasive plant families can help in the rational use of inputs and minimize environmental impacts, entering the list of precision agriculture technologies. For the development of the system, embedded control hardware and software were validated, allowing real-time action and wireless communication. Signal and image processing actions were applied to two databases referring to crops that are impacted by the presence of invasive plants, maize (*Zea mays*) and groundnut (*Arachis hypogaea*). For the pre-processing step, camera calibration parameters were obtained for stereo image rectification. For the segmentation step, thresholding in the HSV color space and morphological operations were performed. Accuracy and overlapping area metrics were used to validate the process. The disparity maps were obtained from the evaluation of local and semi-global matching algorithms. For the feature extraction step, descriptors of Local Binary Patterns and Haralick moments were applied. For the classification of invasive plants, classifiers based on Support Vector Machines were used. The developed system proved to be capable of generating depth information and plant classification, which contributes to decision-making and the application of different input rates. The results obtained prove the effectiveness of the method developed for recognition, qualification and classification for decision support in the control of families of invasive plants.

**Keywords:** Computer Vision; Stereo vision; Pattern recognition; Plant classification; Machine learning.

## LISTA DE SIGLAS

---

---

AUC	Área Debaixo da Curva
CNN	Rede Neural Convolucional
ESF	Função de Espalhamento de Borda
GPS	Sistemas de Posicionamento Global
HOG	Histograma de Gradientes Orientados
HSV	Intensidade Saturação Valor
IoT	Internet das Coisas
kNN	k-Vizinhos mais Próximos
LBP	Padrões Binários Locais
LSF	Função de Dispersão de Linha
MTF	Função de Transferência Modular
OTF	Função de Transferência Óptica
PSF	Função de Dispersão de Ponto
RGB	Vermelho Verde Azul
RBF	Função de Base Radial
ROC	Característica de Operação do Receptor
RPi	Raspberry Pi
SVM	Máquina de Vetor de Suportes

## LISTA DE FIGURAS

---

---

Figura 1 – Exemplo de imagens de folhas presentes no Flavia Leaf <i>dataset</i> . . . . .	22
Figura 2 – Exemplo de imagens de folhas presentes no Swedish Leaf <i>dataset</i> . . . . .	23
Figura 3 – Exemplo de imagens de folhas presentes no ICL Leaf <i>dataset</i> . . . . .	23
Figura 4 – Exemplo de imagem no Aberystwyth Leaf Evaluation <i>dataset</i> . . . . .	24
Figura 5 – Exemplo de imagem no Komatsuna <i>dataset</i> . . . . .	24
Figura 6 – Exemplo de imagens de folhas presentes no MEW2012 <i>dataset</i> . . . . .	25
Figura 7 – Esquemático do sistema. . . . .	48
Figura 8 – Raspberry Pi 3 model B+. . . . .	50
Figura 9 – Pi Camera v1. . . . .	52
Figura 10 – Conexão entre a câmera e a placa embarcada. . . . .	53
Figura 11 – Posicionamento da câmera (representação 3D e 2D). . . . .	53
Figura 12 – Detalhes do inversor AC/DC. . . . .	54
Figura 13 – Fonte de alimentação com retificação em onda completa e estabilização de tensão. . . . .	55
Figura 14 – Ciclo do milho e período ideal para aplicação de herbicidas. . . . .	56
Figura 15 – Plantículas do milho. . . . .	57
Figura 16 – Plantícula da leiteira. . . . .	58
Figura 17 – Plantículas do capim-carrapicho. . . . .	58
Figura 18 – Ciclo do amendoim. . . . .	59
Figura 19 – Plantícula do amendoim. . . . .	60
Figura 20 – Plantas de mucuna-preta. . . . .	60
Figura 21 – Plantícula do capim-braquiária. . . . .	61
Figura 22 – Obtenção da MTF via método gume de faca inclinado. . . . .	66
Figura 23 – <i>Dataset 1</i> : cultura do milho. . . . .	68
Figura 24 – Campo experimental Jaboticabal: vista geral. . . . .	69
Figura 25 – Campo experimental Jaboticabal: vista lateral. . . . .	70
Figura 26 – Modelo de Visão Estéreo. . . . .	71
Figura 27 – Estrutura típica de plantas dicotiledôneas (como a mucuna) e monocotiledôneas (como o capim). . . . .	77
Figura 28 – Tabuleiro de xadrez para calibração. . . . .	81
Figura 29 – Espaço de cor HSV. . . . .	83



Figura 30 – Definição geométrica do SVM. . . . .	91
Figura 31 – Espaço de características original e depois de transformado por um <i>kernel</i> . . . . .	92
Figura 32 – Exemplo de curvas ROC, com anotação da linha diagonal e da AUC. . . . .	95
Figura 33 – Diagrama da conexão entre os componentes do sistema. . . . .	99
Figura 34 – Visão frontal do sistema. . . . .	100
Figura 35 – Visão traseira do sistema. . . . .	101
Figura 36 – Interior da caixa protetora, com RPis e retificador. . . . .	101
Figura 37 – Detalhes da câmera, equipamento estéreo e lâmpada. . . . .	102
Figura 38 – Experimentos de tamanho de arquivo de imagem. . . . .	103
Figura 39 – Imagens do tabuleiro de xadrez, capturadas de forma síncrona e não proces- sadas pelas câmera esquerda e câmera direita. . . . .	104
Figura 40 – Disparidade da distância da linha de base e avaliação do erro de distorção. . . . .	106
Figura 41 – ESF e LSF de uma amostra da câmera esquerda e direita. . . . .	107
Figura 42 – MTF de cada sensor de câmera e do sistema. . . . .	107
Figura 43 – Localização geográfica e divisão da área de estudo. . . . .	109
Figura 44 – Tomada em tempo real do <i>Dataset 2</i> (imagens em escala ilustrativa). . . . .	110
Figura 45 – Exemplo de imagens adquiridas para o <i>Dataset 2</i> , com tomadas respectiva- mente da câmera esquerda e da câmera direita. . . . .	111
Figura 46 – Exemplo de imagens de valor verdade da segmentação do <i>Dataset 1</i> . . . . .	112
Figura 47 – Exemplo de imagens de valor verdade da segmentação do <i>Dataset 2</i> . . . . .	113
Figura 48 – Exemplo de imagens com anotação manual de classes do <i>Dataset 1</i> , com a leiteira em verde, o capim em azul e outras plantas em vermelha. . . . .	114
Figura 49 – Exemplo de imagens com anotação manual de classes do <i>Dataset 2</i> , com a mucuna em verde, o capim em azul e o amendoim em vermelha. . . . .	115
Figura 50 – Pares de imagens retificadas escolhidas para exemplificar a avaliação dos algoritmos de visão estéreo. . . . .	117
Figura 51 – Mapa de disparidade do primeiro algoritmo estéreo, pré e pós processado. . . . .	118
Figura 52 – Mapa de disparidade do segundo algoritmo estéreo, com diferentes resoluções de entrada. . . . .	119
Figura 53 – Exemplar da segmentação do <i>Dataset 1</i> . . . . .	120
Figura 54 – Exemplar da segmentação do <i>Dataset 2</i> . . . . .	121
Figura 55 – Exemplar da segmentação do <i>Dataset 2</i> . . . . .	123
Figura 56 – Exemplar das janelas de decisão para cálculo dos descritores, com o canal H e o resultado do algoritmo de Canny. . . . .	124
Figura 57 – Matriz de confusão do SVM Linear C=1000: <i>Dataset 1</i> . . . . .	129
Figura 58 – Curva ROC: capim-carrapicho. . . . .	130
Figura 59 – Curva ROC: leiteira. . . . .	130
Figura 60 – Curva ROC: micro-média das curvas de cada classe do <i>Dataset 1</i> . . . . .	133
Figura 61 – Matriz de confusão do classificador de capim-carrapicho. . . . .	134

Figura 62 – Curva ROC classificador específico: capim-carrapicho. . . . .	134
Figura 63 – Matriz de confusão do classificador de leiteira. . . . .	135
Figura 64 – Curva ROC classificador específico: leiteira. . . . .	135
Figura 65 – Matriz de confusão do SVM Linear C=1000: <i>Dataset 2</i> . . . . .	136
Figura 66 – Curva ROC: braquiária. . . . .	136
Figura 67 – Curva ROC: mucuna-preta. . . . .	137
Figura 68 – Curva ROC: amendoim. . . . .	137
Figura 69 – Curva ROC: micro-média das curvas de cada classe do <i>Dataset 2</i> . . . . .	138
Figura 70 – Matriz de confusão do classificador de braquiária. . . . .	138
Figura 71 – Curva ROC classificador específico: braquiária. . . . .	139
Figura 72 – Matriz de confusão do classificador de mucuna-preta. . . . .	139
Figura 73 – Curva ROC classificador específico: mucuna-preta. . . . .	140
Figura 74 – Exemplos de obtenção da altura das plantas. . . . .	140
Figura 75 – Mapa taxa de ocupação mucuna-preta. . . . .	141
Figura 76 – Mapa taxa de ocupação mucuna-preta: fatia de profundidade de até 15 cm. . . . .	141
Figura 77 – Mapa taxa de ocupação mucuna-preta: fatia de profundidade entre 15 e 30 cm. . . . .	142
Figura 78 – Mapa taxa de ocupação mucuna-preta: fatia de profundidade acima de 30 cm. . . . .	142
Figura 79 – Mapa taxa de ocupação capim braquiária. . . . .	143
Figura 80 – Mapa taxa de ocupação capim braquiária: fatia de profundidade de até 15 cm. . . . .	143
Figura 81 – Mapa taxa de ocupação capim braquiária: fatia de profundidade entre 15 e 30 cm. . . . .	144
Figura 82 – Mapa taxa de ocupação capim braquiária: fatia de profundidade acima de 30 cm. . . . .	144

## LISTA DE TABELAS

---

---

Tabela 1 – Visão geral de exemplos de aplicação do SVM . . . . .	38
Tabela 2 – Visão geral de exemplos de aplicação do kNN . . . . .	39
Tabela 3 – Visão geral de exemplos de aplicação de CNN . . . . .	43
Tabela 4 – Características da Raspberry Pi 3 model B+ . . . . .	50
Tabela 5 – Características Câmera Pi. . . . .	52
Tabela 6 – Variáveis utilizadas nos descritores. . . . .	87
Tabela 7 – Teste de velocidade do cartão SD. . . . .	102
Tabela 8 – Energia do sistema em diferentes configurações. . . . .	103
Tabela 9 – Variação dos parâmetros da segmentação. . . . .	122
Tabela 10 – Exemplos do retorno dos descritores para 20 amostras. . . . .	125
Tabela 11 – Resultado da validação cruzada no <i>Dataset 1</i> (anotado em negrito o escolhido) do <i>kernel</i> RBF e Linear. . . . .	127
Tabela 12 – Resultado da validação cruzada no <i>Dataset 1</i> do <i>kernel</i> Sigmoide. . . . .	128
Tabela 13 – Resultado do classificador SVM Linear C=1000: <i>Dataset 1</i> . . . . .	129
Tabela 14 – Resultado do classificador de capim-carrapicho. . . . .	133
Tabela 15 – Resultado do classificador de leiteira. . . . .	133
Tabela 16 – Resultado do classificador SVM Linear C=1000: <i>Dataset 2</i> . . . . .	133
Tabela 17 – Resultado do classificador de braquiária. . . . .	133
Tabela 18 – Resultado do classificador de mucuna-preta. . . . .	134
Tabela 19 – Volume das plantas invasoras na 1ª fileira do campo experimental. . . . .	135

# SUMÁRIO

---

---

<b>CAPÍTULO 1–INTRODUÇÃO</b> . . . . .	<b>14</b>
1.1 Motivação . . . . .	14
1.2 Objetivos . . . . .	18
1.3 Contribuições do trabalho . . . . .	18
1.4 Organização do trabalho . . . . .	18
<b>CAPÍTULO 2–REVISÃO DA LITERATURA E PATENTES</b> . . . . .	<b>19</b>
2.1 Estado da arte . . . . .	19
2.1.1 Metadados dos artigos . . . . .	19
2.1.2 <i>Datasets</i> e aquisição de imagens . . . . .	20
2.1.3 Pré-processamento . . . . .	26
2.1.4 Segmentação . . . . .	27
2.1.5 Pós-processamento . . . . .	29
2.1.6 Extração de características . . . . .	30
2.1.6.1 Descritores de forma . . . . .	31
2.1.6.2 Descritores de textura . . . . .	34
2.1.6.3 Descritores de cor . . . . .	36
2.1.7 Classificadores . . . . .	37
2.1.7.1 <i>Support Vector Machines</i> (SVM) . . . . .	37
2.1.7.2 k-Vizinhos mais Próximos (kNN) . . . . .	38
2.1.7.3 Representação esparsa . . . . .	39
2.1.7.4 Rede Neural Artificial (ANN) . . . . .	40
2.1.7.5 Árvore de Decisão e <i>Random Forest</i> (RF) . . . . .	41
2.1.7.6 Outros métodos . . . . .	41
2.2 Estado da técnica . . . . .	44
2.3 Considerações finais . . . . .	45
<b>CAPÍTULO 3–MATERIAIS E MÉTODOS</b> . . . . .	<b>47</b>
3.1 Abordagem geral . . . . .	47
3.2 Materiais . . . . .	48
3.2.1 Raspberry Pi . . . . .	49
3.2.1.1 Gerenciamento de memória . . . . .	51
3.2.2 Câmera . . . . .	51
3.2.3 Bateria . . . . .	52
3.2.3.1 Potência do sistema . . . . .	53
3.2.3.2 Tempo de operação do sistema . . . . .	54

3.2.4	Inversor e retificador . . . . .	54
3.2.5	Lâmpada LED . . . . .	55
3.2.6	Fenologia e fenotipagem das espécies presentes . . . . .	55
3.2.6.1	Cultura do milho e suas plantas invasoras . . . . .	56
3.2.6.2	Cultura do amendoim e suas plantas invasoras . . . . .	57
3.3	Técnica de IoT para transmissão de dados . . . . .	60
3.4	Aquisição de imagens . . . . .	62
3.4.1	Estabelecimento da comunicação Bluetooth . . . . .	62
3.4.2	Captura de imagens . . . . .	62
3.5	Função de transferência modular . . . . .	63
3.6	Datasets para validação do método . . . . .	67
3.7	Visão estéreo computacional . . . . .	70
3.7.1	Geometria e princípios do sistema estéreo . . . . .	70
3.7.2	Métodos de correspondência estéreo . . . . .	73
3.7.3	Cálculo de volume a partir do mapa de disparidade . . . . .	76
3.7.4	Pseudocódigo do algoritmo estéreo . . . . .	78
3.8	Calibração das câmeras . . . . .	79
3.9	Espaço de cor . . . . .	81
3.10	Segmentação . . . . .	84
3.10.1	Limiarização . . . . .	84
3.10.2	Operações morfológicas . . . . .	85
3.10.3	Métricas de qualidade . . . . .	86
3.10.4	Pseudocódigo da segmentação . . . . .	86
3.11	Extração de Características . . . . .	86
3.11.1	Descritores de forma . . . . .	87
3.11.1.1	Detecção de bordas de Canny . . . . .	88
3.11.1.2	Padrões Binários Locais das bordas . . . . .	88
3.11.2	Descritores de textura . . . . .	89
3.11.3	Pseudocódigo da extração de características . . . . .	90
3.12	Classificação . . . . .	90
3.12.1	Métricas de qualidade . . . . .	93
3.12.1.1	Matriz de confusão . . . . .	94
3.12.1.2	Curva ROC . . . . .	95
3.12.1.3	Validação cruzada . . . . .	95
3.12.2	Pseudocódigo do classificador . . . . .	96

**CAPÍTULO 4—RESULTADOS E DISCUSSÕES . . . . . 98**

4.1	Qualificação do método e do sistema . . . . .	98
4.1.1	Gerenciamento de memória . . . . .	99
4.1.2	Gerenciamento de consumo de energia . . . . .	102

4.2	Validação das câmeras	103
4.2.1	Definição da linha de base	104
4.2.2	Obtenção da função de transferência modular	104
4.2.3	Calibração das câmeras	105
4.3	Resultados em campo experimental	108
4.4	Resultados do algoritmo de visão estéreo	116
4.5	Resultados do algoritmo de segmentação	116
4.6	Resultado dos descritores	123
4.7	Resultados dos classificadores	126
4.7.1	Dataset 1	126
4.7.1.1	Classificador geral	126
4.7.1.2	Classificador específico: capim-carrapicho	126
4.7.1.3	Classificador específico: leiteira	127
4.7.2	Dataset 2	129
4.7.2.1	Classificador geral	129
4.7.2.2	Classificador específico: capim braquiária	131
4.7.2.3	Classificador específico: mucuna-preta	131
4.8	Cálculo de volume	131
<b>CAPÍTULO 5 – CONCLUSÕES</b>		<b>146</b>
5.1	Principais conclusões	146
5.2	Sugestões trabalhos futuros	147
<b>REFERÊNCIAS</b>		<b>148</b>

# Capítulo 1

## INTRODUÇÃO

---

---

Este Capítulo apresenta as motivações contextualizadas com base em diversos estudos e revisão sistemática da literatura. Também apresenta o objetivo geral e os objetivos específicos, os quais incluem etapas laboratoriais e validação em nível de campo com operação em tempo real. Adicionalmente, também são apresentadas as principais contribuições decorrentes do desenvolvimento da pesquisa.

### 1.1 Motivação

A agricultura é definida como o cultivo de plantas e fungos para produção de alimentos, suprimentos animais, fibras, combustíveis e outras matérias primas (MACDONALD, 2020). Ela é um dos fatores importantes para o desenvolvimento da civilização humana. Apesar de haver relatos anteriores sobre o cultivo de algumas plantas e até o uso de ferramentas para auxiliar o plantio e colheita, os indícios do surgimento da agricultura podem ser verificados entre 9.000 e 8.500 anos A.C., na região em que hoje se encontram Israel, Iraque, Líbano e Síria (TAUGER, 2010). Com o cultivo de trigo e centeio, os hábitos nômades foram sendo deixados de lado, enquanto as pessoas passaram a viver em assentamentos maiores. Nas Américas, a agricultura teve o início de seu desenvolvimento tardiamente, por volta de 3.000 anos A.C., a partir da cultura de milho e batata, originados respectivamente do México e dos Andes.

Em 2020, no mundo, 1,56 bilhões de hectares de terra agricultáveis foram usadas como área de cultivo, sendo que somente no Brasil essa área foi de 63,5 milhões ha (4,08% do mundo), que o tornou o 5º país em área de cultivo (FAO, 2020). Em 2021, a agropecuária, junto com a silvicultura e pesca, foi responsável por 4,3% do PIB mundial, o que equivale a uma receita de 4,15 trilhões de dólares (IBRD, 2021). Em abril de 2023, a produção em lavouras em um período de um ano no país tinha como valor bruto 869 bilhões de reais (MAPA, 2023).

Tamanho crescimento da agricultura só foi possível com o advento de novas tecnologias e aprimoramento de técnicas já existentes. Entre técnicas que aumentaram o rendimento de grãos por hectares no século 20, pode-se citar o desenvolvimento de novas variedades, a mecanização e automação agrícola e o uso de técnicas para o controle de pragas, como aquelas que contemplam

o uso de herbicidas, inseticidas e fungicidas (RASMUSSEN et al., 1998). Além disso, a forte industrialização no campo e o melhoramento genético redirecionaram a mudança da agricultura do século 21, com técnicas necessárias frente a futuras mudanças climáticas (MIFLIN, 2000).

Soma-se a este cenário de realidades a melhoria e inovações das técnicas de manejo e dentre elas encontra destaque o manejo baseado em fundamentos da Agricultura de Precisão. A Agricultura de Precisão é um conjunto de técnicas e metodologias que tem como objetivo otimizar o manejo de cultivos e a utilização dos insumos agrícolas, aumentando a eficiência econômica da colheita, garantindo uma maior segurança alimentar (MAPA, 2009). É possível observar que no final do século 20 e início do século 21, ocorreu no setor agrícola não só demanda por alimentos e fontes de energia, mas também por uso racional de insumos, gestão de risco e sustentabilidade. Para se atingir os patamares desejados e outros que necessitam ser considerados, se fez necessário ampliar o uso do conhecimento científico e tecnológico, o que passou a ser fundamental. Neste contexto, as tecnologias avançadas, os métodos avançados e a interação interdisciplinar de áreas passou a ser parte fundamental deste novo paradigma. Outro aspecto importante a ser observado foi o uso expressivo de métodos e processos da Ciência da Computação, envolvendo processamento de sinais, dados e imagens, arquiteturas dedicadas e uso massivo de estatística multivariada, visando a melhoria das técnicas de reconhecimento de padrões e respectivas classificações, para o auxílio a tomada de decisão na aplicação das boas práticas agrícolas. Assim, o uso de grandes bases de dados decorrentes de Sistemas de Posicionamento Global (GPS), de imagens digitais obtidas em diferentes espectros e de sistemas inteligentes possibilitaram evoluir as práticas de amostragem do sistema solo e planta e seu georreferenciamento (GEBBERS; ADAMCHUK, 2010).

Inseridos nestas áreas, pesquisadores têm buscado desenvolver algoritmos para tarefas de percepção visual, incluindo o reconhecimento de objetos (determinar a presença ou não na imagem), a detecção deles e entendimento de cena (compreender o que cada elemento na imagem é e significa). Dado a dificuldade da natureza dos problemas, é necessário tanto estudo teórico como prático ao desenvolver algoritmos. Um problema clássico deste segmento envolve classificação de imagens, que é o agrupamento automático de padrões contidos no ambiente em classes pré-definidas (FENG et al., 2019). Outro é a visão estéreo, em que duas ou mais câmeras recordam diferentes visões de uma mesma cena e a partir dos dados recolhidos pode reconstruí-la, obter contorno de objetos, orientação de superfícies ou profundidade. Tais métodos são pertencentes aos estudos da Visão Computacional, que tenta reproduzir mecanismos realizados no sistema visual e no cérebro humano (GOSTA; GRGIC, 2010).

A Visão Computacional é uma área que tem como objetivo a captura de imagens para a análise ou interpretação de cenas no mundo real. Ela estuda os métodos e algoritmos assim como suas implementações. As áreas de aplicação dessa disciplina tem expandindo recentemente devido ao seu progresso, avanços em câmera e tecnologias computacionais (KLETTE, 2014). Uma forma de compreender a performance de algoritmos de Visão Computacional é hierarquizar



as tarefas deles em três níveis: baixo, em que se é extraída características de apenas uma imagem, como borda, cantos, regiões homogêneas e fragmentos de curvas; médio, em que se extrai de um grupo de imagens características semânticas relevantes, como padrões em comum, profundidade e movimento; e alto, em que se interpreta tais informações (MEER, 2004).

A tecnologia funciona então em duas etapas: aquisição e processamento de imagens. A aquisição é feita exclusivamente a partir de sensores, captando o ambiente e os padrões presentes nas imagens digitais. Esses sensores podem então capturar imagens no espectro visível ou térmico e serem acoplados a veículos, dispositivos, robôs, drones e até satélites, com o auxílio de placas embarcadas, que se tornaram mais acessíveis e eficientes, permitindo o processamento de imagens nos computadores de bordo (SADIQ et al., 2021).

Já o processamento de imagens pode ser resumido em cinco partes. Na primeira, os dados brutos são pré-processados, removendo ruídos e selecionando apenas o objeto de interesse. Em outra etapa, há a extração de características e na terceira etapa, as características passam por um processo de seleção, diminuindo a dimensionalidade dos dados. Em seguida, os dados são classificados, agrupando-os com base em suas semelhanças. Finalmente, na etapa de tomada de decisão, novos dados de entrada podem ser classificados a partir do modelo treinado, identificando assim a que grupo ele pertence (WÄLDCHEN et al., 2018; RAJORIYA, 2020).

Outra aplicação da Visão Computacional é em estereoscopia, em que dados de profundidade são coletados a partir de combinação de imagens capturadas de forma síncrona a partir de duas câmeras, permitindo então a obtenção de informação volumétrica. A visão estéreo é um método de medição eficaz, que utiliza o princípio da triangulação para obter as informações tridimensionais dos pontos da superfície do objeto, onde aliado com a segmentação permite o cálculo de volume de diversos objetos (SHENG et al., 2020b). Com o auxílio de elementos que permitam a obtenção de uma escala de referência, a reconstrução de estruturas 3D estima então com boa acurácia o volume (BÁNDI et al., 2020; KONSTANTAKOPOULOS et al., 2021).

A área agrícola é beneficiada por tais técnicas, em que o uso de sistemas de inspeção visual é bem amplo e assim pode-se substituir tarefas exaustivas e repetitivas realizadas por humanos, reduzindo erros causados por cansaço, subjetividade e demora de ação. Tais sistemas necessitam então de fontes de luzes, dispositivos para a aquisição das imagens e de um sistema computacional para a extração de características e processamento, oferecendo resultados com tomadas de decisão com maior acurácia, menor tempo, repetíveis e sem necessitar de contato físico com o objeto em análise. Os sistemas podem então realizar tarefas como minimização de uso de herbicidas, orientar veículos em campo, inspecionar e sortear frutas e sementes, monitorar rebanhos e analisar saúde de plantas de cultivo (GOMES; LETA, 2012).

Deste modo, a Visão Computacional pode contribuir no controle de plantas invasoras, importante para um correto manejo do cultivo. Considera-se atualmente que das 374.000 espécies de plantas conhecidas (sendo 295.383 plantas capazes de gerar flores), cerca de 250 espécies se tornaram importantes para serem controladas quando presente em área de cultura

(CHRISTENHUSZ; BYNG, 2016; HOLM et al., 1977). Essas espécies reduzem o rendimento da cultura em mais de 50% apenas por meio da competição pela umidade presente no solo, causando danos maiores do que os decorrentes da presença de animais invasores, doenças e outras pragas (ABOUZIENA; HAGGAG, 2016). Portanto, as plantas invasoras são um problema devido a sua interação competitiva ou parasítica com a planta de cultivo, podendo então disputar por recursos presentes no solo, espaço, luz solar e até impedindo o crescimento das raízes da planta de interesse (BHATLA; LAL, 2018).

Entre os métodos de segurança desenvolvidos para o controle das plantas invasoras, se tem a solarização do solo, cobertura morta e cobertura morta biodegradável, uso de água quente e o uso de herbicidas naturais e químicos (ABOUZIENA; HAGGAG, 2016). Entretanto, a aplicação controlado destes insumos demanda tecnologia de aplicação adequada e pode ser definida emprego de conhecimentos científicos e técnicos que proporcionam a correta colocação do produto biologicamente ativo no alvo de interesse (ANTUNIASSI et al., 2004; CUNHA; CARVALHO, 2010). O Ministério da Agricultura, Pecuária e Abastecimento (MAPA) define todos esse insumos e técnicas para o controle como agrotóxicos (Lei nº 7802). Tais procedimentos devem envolver somente as quantidades necessárias, de forma econômica e de forma a apresentar o mínimo de deriva para evitar atingir outras áreas localizadas na vizinhança do alvo planejado. As aplicações de agrotóxicos buscam um resultado biológico esperado, que normalmente é o controle de pragas ou plantas invasoras presentes nas culturas.

O uso indiscriminado de agrotóxicos, entretanto, gera danos indesejados ao meio ambiente. Os herbicidas, quando usados indevidamente, são capazes de se espalharem para regiões indesejadas, diminuindo a qualidade da água presente em rios, lagos e até mesmo da água subterrânea, podendo então afetar na variedade de espécies fora da região de cultivo e até mesmo podendo gerar impactos negativos na saúde humana (LIEBMAN; DYCK, 1993). Outra consequência é a perda da efetividade, uma vez que se usado de forma excessiva acaba por favorecer que espécies de plantas invasoras desenvolvam resistências à certos herbicidas, levando-as a se proliferarem mais rapidamente nas áreas das culturas agrícolas. Logo, várias medidas capazes de reduzir o uso dessas substância são alvo de estudos. Por exemplo, o uso das substâncias no tempo ideal de controle já diminui tais riscos, em vez de aplicá-las de forma profilática ou periódica. De modo semelhante, definir valores limites de doses ou frequência também reduzem a quantidade de material usado (RAJCAN; SWANTON, 2001).

Tendo em vista tal cenário de realidades, os vários pontos mencionados se configuraram como elementos motivadores para o desenvolvimento de um método de reconhecimento e classificação de famílias de plantas invasoras, visando melhorar a eficiência do controle, bem como a aplicação localizada com base no uso racional de herbicidas.

## 1.2 Objetivos

O objetivo geral consiste em desenvolver e consolidar um sistema de visão computacional para reconhecimento e classificação de padrões de famílias de plantas invasoras, baseado em métodos de processamento de sinais e imagens e aprendizado de máquina e abordagem a partir do uso de método de visão estereoscópica.

Os objetivos específicos consistem em: desenvolvimento de estrutura mecânica para o alojamento de *hardware* e métodos de *software* de Visão Computacional embarcados; consolidação de uma base de imagens adquirida em ambiente de laboratório e de campo em tempo real; segmentação, extração de características e classificação de famílias de plantas com base no uso de aprendizado de máquina, utilizando máquinas de vetores de suporte (SVM); evolução do sistema de leitura 2D para 3D, a partir de dados de profundidade obtidos pelo desenvolvimento de coleta de imagens estéreo, visando melhoria na definição do mapa de prescrição para auxílio do volume de calda a ser aplicado para ganhos da eficácia e eficiência no controle de plantas invasoras.

## 1.3 Contribuições do trabalho

A principal contribuição deste trabalho está relacionada ao desenvolvimento de método e de um sistema estéreo móvel, baseado nos conceitos de Visão Computacional, capaz de identificar e classificar padrões em famílias de plantas invasoras inteiras e seu controle localizado em taxa variada. Esta pesquisa agrega valor a área da Visão Computacional, bem como auxilia no uso racional de insumos, na efetividade do controle de plantas invasoras e na minimização de impactos ambientais.

## 1.4 Organização do trabalho

Este trabalho está estruturado em cinco capítulos onde, a partir desta Introdução, o Capítulo 2 apresenta uma revisão da literatura em estado da arte e da técnica, o Capítulo 3 trata sobre os materiais e métodos, incluindo a fundação teórica e pseudocódigos dos modelos e algoritmos desenvolvidos. Adicionalmente, o Capítulo 4 apresenta resultados e discussões, assim como e finalmente, no Capítulo 5 são expostas as conclusões e sugestões de trabalhos futuros.

# Capítulo 2

## REVISÃO DA LITERATURA E PATENTES

---

---

Este Capítulo apresenta uma descrição dos principais trabalhos relacionados ao tema desta pesquisa, associados ao estado da arte e ao estado da técnica. A revisão bibliográfica foi baseada no uso de palavras-chaves nas seguintes bases bibliográficas, referendadas e atualizadas nas áreas de Visão Computacional e Ciência da Computação: *IEEE Xplore*, *Science Direct*, *Scopus*, *Web of Science*, *Google Scholar*. Além disso, também foi realizado um monitoramento tecnológico em base de patentes a partir de palavras-chaves específicas.

### 2.1 Estado da arte

Foi realizada uma revisão sistemática com o objetivo geral de identificar e avaliar quais são os métodos de reconhecimento de padrões mais utilizados em imagens digitais de plantas, afim de conhecer, comparar, avaliar, melhorar e adaptar os algoritmos já usados. Além disso, foi analisado a evolução temporal e demográfica do interesse de pesquisadores sobre a área.

Para auxiliar na execução e análise da revisão sistemática, foi utilizado a plataforma StArt (*State of the Art through Systematic Review*). Essa ferramenta foi desenvolvida pelo Lapes (Laboratório de Pesquisa em Engenharia de Software) da UFSCAR com o intuito de auxiliar pesquisadores no gerenciamento e processamento de revisões sistemáticas. A *String* de busca  $S_{busca}$  utilizada foi:

$S_{busca}$ : **"pattern recognition"AND ("weed"OR "plant"OR "agriculture"OR "agricultural") AND ("computer vision"OR "image processing")**

Também foram feitas pesquisas com palavras-chaves pontuais, com o intuito de aprofundar melhor em tópicos de interesse. A partir de um total de 741 artigos, foram selecionados 75 artigos pertinentes e de interesse para a pesquisa.

#### 2.1.1 Metadados dos artigos

A partir dos artigos selecionados foi possível analisar dados compartilhados entre eles. Na distribuição temporal, 14 artigos são de 2017, 8 de 2018, 15 de 2019, 9 de 2020, 12 de

2021, 9 de 2022 e 5 de 2023. Na análise de origem, a maioria dos textos são da China (17 artigos), Índia (11 artigos) e Estados Unidos da América (10 artigos). No total e considerando as pesquisas feitas de forma conjunta entre mais de um país, 40 artigos foram produzidos por países Asiáticos, 19 nas Américas, 11 na Europa, 4 na Oceania e 1 na África. Não foi possível identificar a localização de produção de apenas um artigo.

Dos artigos selecionados, 9 são revisões sistemáticas relacionadas diretamente ao tema da pesquisa e portanto apresentam um compilado de dados de outros trabalhos além dos aqui analisados. Entre estas revisões, [Pereira et al. \(2017\)](#) avaliaram 60 artigos do período de 2000 até 2015, focando em detecção e classificação de frutos, apesar de também compilarem dados de detecção de folhas e plantas. Já [Sabu e Sreekumar \(2017\)](#) analisaram 13 artigos entre 2000 e 2016, [Bojamma e Shastry \(2019\)](#) analisaram 11 entre 1998 e 2018 e [Wäldchen et al. \(2018\)](#) compilaram a informação de 79 artigos; as três revisões focaram nos aspectos gerais da identificação de espécies de plantas, desde o pré-processamento das imagens digitais até a processo de classificação. Já [Sadiq et al. \(2021\)](#) analisaram o uso de imagens digitais, visão computacional e sensores na agricultura, a partir de dados recolhidos de 39 artigos publicados entre 2001 e 2021, enquanto [Dokania e Yadav \(2022\)](#) compilaram 20 artigos entre 2011 e 2022 que utilizaram métodos de visão computacional na análise de vegetação em situação de estresse.

[Wäldchen e Mäder \(2018\)](#) apresentaram a revisão sistemática mais detalhada entre as analisadas, em busca do perfil dos pesquisadores da área, dos métodos de aquisição de imagens utilizados, as características mais detectadas e utilizadas na identificação de plantas, quais destes fatores são mais relevantes e se há aplicações práticas já desenvolvidas. No total, 120 artigos publicados em repositórios científicos de alto impacto durante a década de 2005 a 2015 foram encontrados e estudados.

[Wang et al. \(2017\)](#) fizeram pesquisa semelhante, porém adicionando também uma análise dos classificadores mais utilizados de um conjunto de 84 artigos publicados entre 2000 ao início de 2016. De modo similar, [Rehman et al. \(2019\)](#) também fizeram uma análise dos classificadores utilizados em diversas aplicações na agricultura, dividindo entre métodos supervisionados e não supervisionados, de um grupo de 131 artigos publicados desde 1990 até 2017.

### 2.1.2 *Datasets* e aquisição de imagens

O primeiro passo para treinar algoritmos para a classificação de espécies de plantas diferentes é definir o banco de imagens, isto é, o *dataset*, que será utilizado. Devido a natureza do problema, as imagens tem que possuir informações suficientes para a identificação das plantas individualmente e possuir uma qualidade mínima para ser capaz de delimitar a região de interesse na imagem. Assim, o autores analisados se restringiram a lidar com imagens obtidas próximas ao objeto, rejeitando então casos como o de imagens obtidas via satélite ou de uma distância muito alta com o auxílio de UAVs (como por exemplo em [Sheng et al. \(2020a\)](#)), onde se trabalham com grandes regiões em vez de de objetos individuais. Também, esses autores desejaram não

limitar a aplicação do sistema a imagens obtidas por equipamentos muito específicos, imagens coloridas (RGB) ou na escala de cinza são preteridas, não considerando imagens 3D obtidas por um grande conjunto de câmeras (como em [Li et al. \(2020\)](#)) ou imagens fora da frequência da luz visível, como imagens espectrais, infra-vermelho e termais ([DOKANIA; YADAV, 2022](#)). Embora tais equipamentos sejam capazes de discriminar a morfologia de plantas, como [Braga et al. \(2023\)](#) que utilizaram um espectrômetro infravermelho próximo para diferenciar três espécies de plantas invasoras, eles fogem do escopo desta pesquisa.

Um problema comum na área de identificação de espécies de planta é a falta de um *dataset* padrão para avaliação de diferentes métodos amplamente adotado e de *datasets* que apresentam imagens em condições reais ao que seria observado na natureza. A maioria dos estudos focam na identificação das plantas a partir das folhas, pois estão presentes durante boa parte da vida da planta, ao contrário de flores e frutos e são mais fáceis de se registrar e guardar. [Wäldchen et al. \(2018\)](#) e [Bojamma e Shastry \(2019\)](#) também perceberam como dificuldade na área um alto número de espécies para identificação, sendo que há variações morfológicas entre folhas de plantas da mesma espécie ao mesmo tempo que folhas de espécies diferentes podem apresentar pequenas variações entre si. Muitos estudos também não têm métodos para rejeição de dados ao qual não foram treinados, isto é, espécies não utilizadas no treinamento do modelo preditivo. Outro problema ao adquirir as imagens das plantas são variações de qualidade e iluminação.

[Wäldchen e Mäder \(2018\)](#) notaram que dos 120 estudos analisados, a maioria usaram imagens digitais escaneadas ou pseudo-escaneadas, com imagens de folhas em ambiente controlado e fundo uniforme, facilitando o trabalho de obter a região de interesse e evitando dificuldades como variações de iluminação, sobreposição de folhas e folhas danificadas. Apenas 25 artigos utilizaram imagens adquiridas em ambiente natural, sendo que a maioria destes foram obtidos pelos próprios pesquisadores e não disponibilizados para uso posterior.

Entre os *datasets* mais utilizados na área, há três que são mais consolidados: o Flavia Leaf *dataset*, Swedish Leaf *dataset* e o ICL Leaf *dataset*. O Flavia Leaf *dataset* ([WU et al., 2007](#)) é composto de 1907 imagens de 32 espécies, com 50 a 77 imagens de folhas por espécies. A resolução das imagens é de 1600 por 1200 *pixels* e possuem uma folha por imagem sobre um fundo totalmente branco. Exemplo das imagens para cada espécie pode ser visto na Figura 1. [Wang et al. \(2019\)](#), [Hirasen e Viriri \(2020\)](#), [Zhang et al. \(2020\)](#), [Reddy et al. \(2021\)](#), [Yang \(2021\)](#) e [Preethi et al. \(2023\)](#) usaram este *dataset*.

Já o Swedish Leaf *dataset* ([SÖDERKVIST, 2001](#)) disponibiliza 1125 imagens de 15 espécies mais semelhantes entre si, com 75 imagens de folha por espécie. Desenvolvido pela Universidade de Linköping, na Suécia, ele contém imagens de folhas escaneadas em fundo uniforme, como pode ser visto na Figura 2. Novamente, [Wang et al. \(2019\)](#), [Hirasen e Viriri \(2020\)](#), [Zhang et al. \(2020\)](#), [Reddy et al. \(2021\)](#) e [Yang \(2021\)](#) utilizaram este *dataset*.

O ICL Leaf *Dataset* foi coletado pelo Laboratório de Computação Inteligente da Acade-

mia Chinesa de Ciências e contém 16848 imagens de folhas de 220 plantas, com alguns exemplos presentes na Figura 3. Utilizaram em seus trabalhos este *dataset* Zhang et al. (2017a), Zhang et al. (2017b), Wang et al. (2019) e Zhang et al. (2020).

Entre os *datasets* menos populares, tem-se o Aberystwyth Leaf Evaluation *dataset* (BELL; DEE, 2016), Komatsuna (UCHIYAMA et al., 2017), COCO (LIN et al., 2014) e MEW2012 (NOVOTNY; SUK, 2013). O Aberystwyth Leaf Evaluation *dataset* consiste em imagens do crescimento de 4 conjuntos de 20 plantas da espécie *Arabidopsis thaliana* adquiridas em uma estufa durante o período de 35 dias, com exemplo de uma planta em duas datas diferentes presente na Figura 4. Este *dataset* foi utilizado por Al-Shakarji et al. (2017), Dobrescu et al. (2019), Giuffrida et al. (2019), Kuznichov et al. (2019) e Lin et al. (2023). Já o Komatsuna *dataset* foi adquirido a partir de uma câmera RGB-D, possuindo então imagens digitais RGB e imagens com informação de profundidade, de mudas de *Brassica rapa var. perviridis* em diferentes estágios de crescimento, como pode ser visto na Figura 5. Utilizaram o *dataset* Dobrescu et al. (2019), Giuffrida et al. (2019) e Lyu et al. (2020).

Figura 1 – Exemplo de imagens de folhas presentes no Flavia Leaf *dataset*.



Fonte: Wu et al. (2007).

Figura 2 – Exemplo de imagens de folhas presentes no Swedish Leaf *dataset*.



Fonte: [Söderkvist \(2001\)](#).

Figura 3 – Exemplo de imagens de folhas presentes no ICL Leaf *dataset*.



Fonte: [Hu et al. \(2012\)](#).



Figura 4 – Exemplo de imagem no Aberystwyth Leaf Evaluation *dataset*.



Fonte: [Bell e Dee \(2016\)](#).

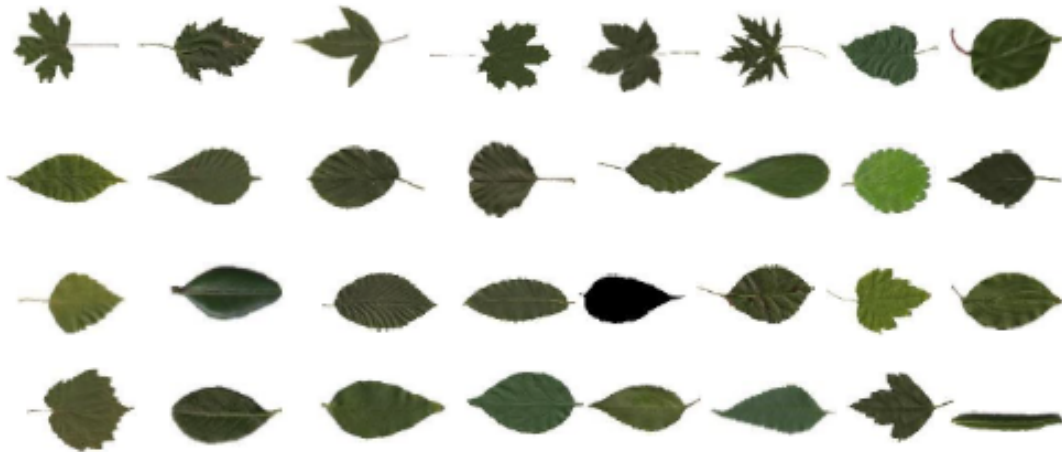
Figura 5 – Exemplo de imagem no Komatsuna *dataset*.



Fonte: [Uchiyama et al. \(2017\)](#).

O Microsoft COCO *dataset*, utilizado por [Huang et al. \(2017\)](#) e [Higgs et al. \(2019\)](#), consiste em 328 mil imagens de diversos objetos em situações reais divididos em 91 tipos, sendo um destes grupos plantas em vasos. Já o MEW2012 *dataset* (*Middle European Woods*) possui 9745 imagens de folhas com fundo simples, representando 151 espécies com no mínimo 50 imagens para cada. Na Figura 6 é mostrado exemplos das imagens presentes no *dataset*, utilizado por [Huang et al. \(2017\)](#), [Higgs et al. \(2019\)](#) e [Zhang et al. \(2020\)](#).

Figura 6 – Exemplo de imagens de folhas presentes no MEW2012 *dataset*.



Fonte: [Novotny e Suk \(2013\)](#).

Foi possível perceber que os *datasets* mais utilizados apresentam apenas imagens de folhas isoladas e com fundo simples de ser retirado. Dos artigos analisados, 18 desenvolveram *datasets* próprios, em que a maioria não foram disponibilizados. [Wäldchen e Mäder \(2018\)](#) também perceberam que a maioria dos *datasets* utilizados são pequenos, com menos de 100 espécies e de 1000 imagens disponíveis. Dos 120 estudos por eles analisados, 12 utilizaram o Swedish Leaf *dataset*, 19 o Flavia Leaf *dataset*, 12 o ICL Leaf *dataset*, 20 o ImageCLEF11 e ImageCLEF12 *dataset* ([GOËAU et al., 2011](#); [GOËAU et al., 2013](#)) e 6 utilizaram o Leafsnap *dataset* ([KUMAR et al., 2012](#)). Notou-se que apesar de não ser o foco do banco de imagens, os dois últimos *datasets* possuem algumas imagens registradas em ambiente natural.

Parte dos autores estudados também utilizaram técnicas de aumento do número de imagens usadas para o treinamento de métodos de classificação (*data augmentation*) a partir de transformações como mudança de escala, rotação, translação e mudança de contraste e iluminação nas imagens base. [Kuznichov et al. \(2019\)](#) desenvolveram um algoritmo capaz de gerar novas imagens a partir de folhas de imagens pré-existentes, as segmentando e selecionando as melhores para criar imagens simuladas. Essas novas imagens criadas puderam ser geradas de forma desordenada, espalhando as folhas em um fundo, rotacionando-as, mudando a escala e simulando até mesmo a sobreposição. Essa folhas também puderam estar estruturadas, formando

novas imagens a partir da seleção de um fundo de um grupo específico, definindo aleatoriamente onde foi o centro da planta e adicionando as folhas pré-selecionadas de forma que o tamanho e ângulo delas sejam condizente com uma muda real. Outros métodos de aumento do banco de imagens foram analisados por [Xu et al. \(2023\)](#).

### 2.1.3 Pré-processamento

Antes da etapa de seleção da região de interesse da imagem, ações podem ser realizadas tanto para melhorar tanto a qualidade da imagem, para lidar com casos de presença de ruído, iluminação irregular, presença de sombras e baixa resolução, quanto para diferenciar as características entre o objeto e o fundo ([DOKANIA; YADAV, 2022](#)). Outra ação comumente realizada é a mudança do espaço de cor da imagem para trabalhar apenas com a informação necessária.

Entre as ações de melhoramento, [Sabu e Sreekumar \(2017\)](#) notaram que alguns pesquisadores procuram o eixo maior das folhas e o colocam na horizontal para tornar os seus métodos invariantes a rotação. Outras técnicas também utilizadas são a conversão para a escala de cinza, remoção de ruído a partir de filtro passa-baixa, Gaussiano ou filtro de média, ajuste de contraste e equalização de histograma.

Como exemplos, pode-se citar [Gayatri et al. \(2021\)](#) e [Hameed e Üstündağ \(2021\)](#), que realizaram o aprimoramento de contraste, enquanto [Zhang et al. \(2020\)](#) equalizaram o histograma da imagem e utilizaram um filtro passa-alta para evidenciar as bordas da folha. Já [Patrón et al. \(2018\)](#) rotacionaram a imagem garantindo que ela fique perpendicular, enquanto [Zhang et al. \(2017b\)](#) mudaram a escala de todas as imagens para que tivessem resolução de 32x32, similar a [Zhang et al. \(2017a\)](#), que redimensionaram para 32x16. Com o intuito de eliminar ruídos, [Raut e Bhalchandra \(2018\)](#) utilizaram um filtro de mediana 5x5.

Apesar de ser comum manter a imagem no espaço de cor RGB ou realizar uma conversão simples para a escala de cinza, pode-se converter para outros espaços como o HSV ([JYE et al., 2017](#); [CHEN et al., 2019](#); [ARUNACHALAM; ANDREASSON, 2021](#)), o L\*a\*b\* ([ALSHAKARJI et al., 2017](#); [GAYATRI et al., 2021](#); [ZHANG et al., 2021](#)) e YCbCr ([KAMATH et al., 2020](#)). [Turhal e Dagdelen \(2022\)](#) utilizaram dados da imagem em três espaços de cor, o HSV, L\*a\*b\* e CIELUV. [Pereira et al. \(2017\)](#) notaram que dos 60 artigos por eles analisados, 7 usaram o espaço de cor HSV, 9 o HSI, 2 o CIELUV, 3 o L\*a\*b\* e 2 o YCbCr.

Outras transformações utilizando os valores dos *pixels* RGB são comumente realizadas, como os Índices de Vegetação, com o intuito de acentuar a presença de vegetação nas imagens digitais. Os mais comumente utilizado na área de identificação de plantas é o ExG (*excess green*) ([ANDREA et al., 2017](#); [HIGGS et al., 2019](#); [KHAN et al., 2020](#); [LE et al., 2020](#)). [Le et al. \(2020\)](#) também utilizaram junto com o ExG o ExR (*excess red*). [Bakhshipour e Jafari \(2018\)](#) usaram tanto o ExG, como as transformações *GMR* (*green minus red*) e *Cg* (*green colour difference*). [Santiago et al. \(2019\)](#) aplicaram outra transformação, chamada de *PCD*, utilizando os canais

vermelho e verde normalizados para calcularem uma distância de cor dos *pixels*.

Um dos objetivos dessa mudança do espaço de cor é, além de facilitar uma posterior segmentação das plantas, diminuir os efeitos da variação de iluminação na cena. Assim, enquanto alguns pesquisadores preferiram normalizar o valor de intensidade dos *pixels* verde (ANDREA et al., 2017; LYU et al., 2020), outros preferiram usar canais específicos de tais espaços (H para o HSV, \*a e \*b no L\*a\*b\* e Y no YCbCr). Outras ações também podem ser realizadas com o mesmo objetivo. Higgs et al. (2019), por exemplo, nivelaram a luminância da imagem para facilitar a detecção das bordas das plantas, enquanto Jimenez et al. (2017) usaram um filtro em HSL para saturar a imagem antes de convertê-la para a escala de cinza. Já Lin et al. (2023) realizaram uma correção de cor com auxílio de uma rede neural.

Ações para eliminar elementos constantes presentes nas imagens também puderam ser realizadas nesta etapa. Khan et al. (2020) aproveitaram que o fundo das imagens era fixo para subtraí-lo de todas as amostras. De modo semelhante, Higgs et al. (2019) observaram que a sombra do veículo em que a câmera estava posicionada aparecia nas imagens capturadas e somaram a mudança de *pixels* entre todas as imagens para computar o gradiente deles. Com esse dado, foram capazes de criar uma máscara com os elementos estáticos para a subtração. Jye et al. (2017) converteram a imagem para HSV apenas para facilitar a eliminação da sombra da folha escaneada com o uso de uma máscara específica e posteriormente a converteu para a escala de cinza. Todas estas técnicas têm como objetivo final preparar as imagens e facilitar a etapa de segmentação, eliminando dados desnecessários.

### 2.1.4 Segmentação

A segmentação de uma imagem tem como objetivo separar o objeto de interesse, no caso plantas e folhas, do fundo. O processo mais comum parte do uso de características gerais da imagem para gerar uma máscara binária, capaz de remover o fundo e isolar apenas a região em que extrairá as características utilizadas para a classificação.

Um método de clássico para tal tarefa é o método de limiarização por Otsu (OTSU, 1979), que calcula de forma automática o valor no histograma da imagem ideal para separar a imagem em dois grupos com menor soma de variância intraclasses. Wang et al. (2019) e Preethi et al. (2023) aplicaram o algoritmo na imagem em escala de cinza, enquanto Lyu et al. (2020) realizaram a mesma operação apenas no canal verde do RGB. Bakhshipour e Jafari (2018) utilizaram Otsu nas imagens geradas pelas transformações ExG, GMR e Cg e obtiveram melhor resultado com o GMR, com taxa de segmentação correta (número de *pixels* corretamente segmentados por número de *pixels* do objeto) igual a 99,32%. Andrea et al. (2017) usaram apenas na imagem ExG. De forma semelhante, Santiago et al. (2019) aplicaram também Otsu na imagem gerada pela transformação PCD. Khan et al. (2020) aplicaram primeiramente o método de Otsu e delimitaram um valor mínimo de limiar, mas para garantir um melhor resultado utilizou a imagem transformada em ExG para encontrar um segundo valor de limiar, limitado entre os

dois anteriores (o valor mínimo e o obtido por Otsu).

Outros pesquisadores preferiram simplesmente, a partir de experimentos, encontrar o melhor valor para realizarem uma limiarização simples. Zhang et al. (2017b) encontraram o valor de limiar igual a 30 na imagem na escala de cinza, enquanto Jimenez et al. (2017) também fizeram a limiarização na imagem cinza previamente saturada. Chen et al. (2019), Arunachalam e Andreasson (2021) e Wang et al. (2022) realizaram uma limiarização baseado nos tons de verde no canal H do HSV, porém Chen et al. (2019) também aplicaram o detector de bordas de Canny (CANNY, 1986), segmentando a imagem e utilizando a intersecção dos dois resultados. O operador de Canny utiliza um algoritmo com várias etapas capaz de detectar bordas a partir de um gradiente de magnitude da intensidade dos *pixels*. Depois desta etapa, eles previram e verificaram a partir dos segmentos de folhas identificados a folha em sua totalidade, levando em consideração a inclinação das bordas e o esqueleto dela, obtendo como resultado uma segmentação com uma acurácia SBD (*Symmetric Best Dice*) de 37%. Moreno e Cruvinel (2022) também realizaram a limiarização em um intervalo no canal H do HSV, aliado com um algoritmo de *watershed* e operações morfológicas para segmentação individual das folhas, já que as imagens apresentavam diversidade em famílias e tipos de plantas invasoras.

É possível também realizar a segmentação a partir de métodos de agrupamento em *clusters*, como o k-means (MACQUEEN et al., 1967), separando os *pixels* da imagem em 2 ou mais grupos de modo que cada *pixel* é definido participante do *cluster* com distância mais próxima. Gayatri et al. (2021) por exemplo utilizaram tal método para separar na imagem o fundo, a folha e a área na folha afetada por doenças. Hameed e Üstündağ (2021) também segmentaram folhas com tal método. Algoritmos k-means podem então ser modificados ou aliados com outros métodos para se obter um melhor resultado da segmentação. Por exemplo, Zhang et al. (2021) removeram a maior parte do fundo da imagem utilizando um Grabcut (ROTHER et al., 2004) modificado e depois utilizaram um algoritmo adaptativo dinâmico *fuzzy* k-means para finalizar a separação da única planta presente em cada imagem do fundo.

Al-Shakarji et al. (2017) também utilizaram o k-means para gerar uma máscara binária da região onde se encontra a planta (região interna do pote presente na imagem no espaço  $L*a*b^*$ ), enquanto um algoritmo de maximização de expectativa (DEMPSTER et al., 1977) gerava uma máscara mais detalhada destacando a presença da planta na imagem em RGB. As máscaras então eram interseccionadas, segmentando corretamente apenas as plantas. K-means então foi utilizado com a intenção de remover os ruídos resultantes da maximização de expectativa. Após essa etapa, também foi aplicado Watershed (VINCENT; SOILLE, 1991) para segmentar as folhas das plantas, obtendo então cada folha de forma individual, com uma acurácia SBD entre 72,1% e 74,6%.

Outro método utilizado para a segmentação é o baseado em redes neurais, algoritmos que simulam as conexões entre neurônios de tal modo que os dados de entrada passam por nós treinados de diferentes pesos em várias camadas, produzindo o resultado esperado. Krause et al.

(2019) abordaram o problema a partir da análise de cena (*scene parsing*), algoritmo desenvolvido pelo MIT a partir de uma rede neural convolucional de 3 camadas aplicada em diversas imagens comuns de pessoas e objetos, procurando o maior objeto correspondente a planta na imagem (descartando regiões menores) e o utilizando na segmentação. Como consequência, apenas uma espécie pôde ser analisada posteriormente, já que a segmentação ignora as plantas menores.

O uso de rede neural convolucional (CNN) tem se tornado comum na segmentação de folhas, porém exigem alto teor de processamento para o seu treino, como pode ser visto em Parashar et al. (2021), Li et al. (2022) e Lin et al. (2023). Higgs et al. (2019) utilizaram uma rede *Faster R-CNN* para detectar a presença de plantas individuais na imagem, possibilitando a detecção de várias plantas pequenas em uma mesma imagem, enquanto similarmente Almazaydeh et al. (2022) utilizaram uma rede *Mask R-CNN*.

Há diferenciados métodos utilizados por pesquisadores no trabalho de segmentação de plantas, porém são métodos novos ou pouco utilizados, com resultado insuficientes que garantam a eficácia para aplicações mais gerais. Por exemplo, Pengcheng et al. (2018) utilizaram os valores dos *pixels* presentes em uma imagem, obtida em campo a partir de uma câmera hiper-espectral, como entrada de um classificador SVM com *kernel* obedecendo uma função *grey sigmoid*, segmentando-a em dois grupos (solo e planta), alcançando uma acurácia de 83,7%, entretanto com um custo computacional muito alto. Já Chouhan et al. (2019) segmentaram folhas a partir de um algoritmo de otimização de pilhagem bacterial (*Bacterial Foraging Optimization*) aliado com uma rede neural artificial, obtendo uma precisão de 86,01% na segmentação, porém utilizando apenas duas imagens na avaliação do método. Turhal e Dagdelen (2022) criaram tensores de cor de 3º grau na segmentação pelo método de *Common Vector Approach*, com precisão de 99,5%, enquanto Trombini et al. (2023) aplicaram para a segmentação um método não supervisionado baseado em grafos e nos campos aleatórios de Markov.

### 2.1.5 Pós-processamento

O pós-processamento tem como objetivo principal melhorar os resultados da segmentação, preparando melhor as amostras para as próximas etapas. Sabu et al. (2017) por exemplo redimensionaram todas as imagens segmentadas para terem 1200 *pixels* de altura, padronizando as imagens que seriam usadas como entrada em futuros algoritmos. Boa parte das técnicas utilizadas focam em operações morfológicas nas imagens binárias, com o intuito de eliminar pequenos artefatos como buracos ou objetos pequenos fora da região de interesse. Le et al. (2020) realizaram operação de abertura e fechamento usando um elemento estruturante quadrado 5x5, ao passo que Jye et al. (2017), Wang et al. (2019) e Arunachalam e Andreasson (2021) apenas aplicaram o fechamento com um elemento 2x2. Preethi et al. (2023) também removeram buracos a partir de operações morfológicas.

Outro procedimento comum no preparo das imagens para a fase de extração de características é a de criar duas imagens a partir da imagem binária: uma em que a utiliza como

máscara sobre a imagem original, recuperando a cor original do objeto (ARUNACHALAM; ANDREASSON, 2021); e uma em que se é obtido o contorno da imagem binária, a partir do algoritmo de Canny Edge (JYE et al., 2017) ou de filtros específicos de passa-alta (ZHANG et al., 2020). Zhang et al. (2017b) aproveitaram a informação do contorno para identificarem e removem o caule da folha e depois as redimensionaram para 32x32, enquanto Kamath et al. (2020) utilizaram as informações de cor do objeto para aplicarem um algoritmo de equalização de histograma adaptativo com o intuito de diminuir os efeitos de iluminação destoantes. Já Zhang et al. (2017a) realçaram as regiões de borda na imagem a partir de um operador Robert cross.

Também é comum a aplicação de filtros com o intuito de minimizar os ruídos da limiarização. Jye et al. (2017) utilizaram um filtro de mediana 3x3 e Zhang et al. (2017b) um de raio 10. Com o objetivo de suavizar as imagens limiarizadas, Andrea et al. (2017) preferiram eliminar as folhas com dimensões menores do que 64x64, enquanto Chen et al. (2019) utilizaram um algoritmo para completar as folhas a partir do caule detectado e da curvatura presente nas bordas das folhas.

### 2.1.6 Extração de características

As principais características utilizadas no trabalho de identificação de espécies de plantas a partir das folhas podem ser separadas em três tipos: forma, cor e textura. Apesar disso, outras características como as nervuras, simetria, tipo de margem, ponta e base das folhas também podem ser utilizadas, porém são mais incomuns. As nervuras, apesar de apresentarem grande informação para a diferenciação entre diferentes espécies, são difíceis de serem analisadas em imagens digitais devido ao fato de geralmente não terem um contraste muito alto, o que faz necessário então que a demarcação delas na imagem sejam feitas manualmente após a captura da imagem, ou previamente a partir de substâncias químicas capazes de circular pelas veias da planta e possuam coloração diferente da cor natural da folha. Já as margens são mais relevantes ao utilizar para distinguir entre espécies de plantas em que as folhas são dentadas, isto é, não lisas. Pode-se então criar descritores a partir de um ou mais destes fatores.

De acordo com Wäldchen e Mäder (2018), o fator mais utilizado é a forma, sendo usado como único parâmetro em quase metade dos estudos analisados. Se considerarmos o uso com outros fatores, 89 dos 120 estudos analisados utilizaram a forma. Os descritores ideais são os invariantes a transformações geométricas como translação, rotação, escala e reflexão e podem ser separados em três grupos. O primeiro grupo é o mais simples, baseado somente nas informações morfológicas e dimensões da borda. Neste grupo, os descritores mais utilizados são os de circularidade, razão de aspecto, retangularidade, área, perímetro, solidez, diâmetro e tamanho do eixo maior e menor. O segundo grupo de descritores são baseados na região do objeto, como descritores de momento de Hu, Zernike e Tchebichef e descritores de características locais como o SIFT, HOG e SURF. O terceiro grupo são os baseados no contorno, como os descritores de assinatura, de Fourier e de dimensão fractal.

O segundo fator mais usado é a textura, geralmente obtido a partir de descritores que utilizam os filtros de Gabor, FracDim, matriz de co-ocorrência e histograma de orientação de borda. Já a cor é o fator menos utilizado, sendo usado descritores de momentos de cor (média, desvio padrão, distorção e curtose) e histogramas de cor. Para fatores mais incomuns como a nervura, se constrói descritores como a razão entre a área da nervura e a área da folha e outros baseados na morfologia das nervuras; e para a margem, desenvolve-se descritores como assinatura e representação sequencial dos dentes.

Sabu e Sreekumar (2017) também perceberam um uso maior de descritores baseado em forma, como comprimento, largura, razão de aspecto, diâmetro, envoltória convexa, perímetro, retangularidade, área, compacidade e excentricidade. Para os descritores de textura mais comum se tem o que analisam a energia, contraste e entropia, enquanto que para os de cor os momentos. Os algoritmos de SURF, SIFT e HOG também são bastante utilizados. Já Bojamma e Shastry (2019) perceberam apenas descritores de características baseado em formas, enquanto Wäldchen et al. (2018) concluíram que descritores de cor eram os menos relevantes entre os estudados.

Wang et al. (2017) decidiram comparar o peso de parâmetros gerados por 31 descritores de forma, textura e cor analisados e classificar espécies de plantas utilizando um SVM simples, realizando 20 experimentos aleatórios a partir de cada parâmetro. Concluíram então que os 5 melhores foram o HOG, EnS (sequência de entropia), GFD (descriptor genérico de Fourier), momentos de Hu e CDS (sequência de distância do centro). Como tais descritores geram mais de um parâmetro e isto oferecia uma vantagem com mais entradas para o classificador, eles resolveram analisar os descritores que geravam apenas um valor. Os mais significantes então foram a excentricidade, o fator de estreitamento, correlação de textura, razão entre comprimento e perímetro, distorção de cor, irregularidade e média de luminância da textura.

### 2.1.6.1 Descritores de forma

Os descritores de forma são em geral obtidos a partir da imagem binária obtida pela segmentação, utilizando toda a região ou apenas a borda. Alguns descritores podem ser construídos a partir de informações simples dada por variáveis relacionadas ao formato morfológico da folha e elas então podem ser manipuladas para se obter novos descritores, com informações mais complexas. Entre os descritores mais utilizados na literatura, se tem:

1. Eixo maior (comprimento): segmento de linha conectando a base e a ponta da folha, utilizado por Jye et al. (2017), Li et al. (2020) e Preethi et al. (2023);
2. Eixo menor (largura): maior segmento de linha que é perpendicular ao eixo maior, usado por Jye et al. (2017), Li et al. (2020) e Preethi et al. (2023);
3. Área: número de *pixels* dentro da região da folha, empregado por Jye et al. (2017), Raut e Bhalchandra (2018), Li et al. (2020) e Preethi et al. (2023);



4. Perímetro: soma da quantidade de *pixels* pertencentes à borda do objeto, aplicado por [Jye et al. \(2017\)](#), [Raut e Bhalchandra \(2018\)](#) e [Preethi et al. \(2023\)](#);
5. Razão de aspecto ou excentricidade: define o quão estreito ou largo é uma folha, utilizado por [Raut e Bhalchandra \(2018\)](#) e [Preethi et al. \(2023\)](#);
6. Excentricidade: refere-se à excentricidade da elipse que possui os mesmos segundo momentos que o objeto, descritor aplicado por [Jye et al. \(2017\)](#);
7. Circularidade: ilustra a diferença entre a planta e um círculo, que foi um dos quatro descritores de forma utilizados por [Bakhshipour e Jafari \(2018\)](#);
8. Retangularidade: representa o quão retangular um objeto é, aplicado por [Bakhshipour e Jafari \(2018\)](#) e [Preethi et al. \(2023\)](#);
9. Envoltória convexa (*convex hull*): menor região convexa que contém em seu interior o objeto em análise, utilizado por [Khan et al. \(2020\)](#);
10. Momentos de Hu: descreve o contorno da região a partir de 7 descritores estatísticos que são invariantes a translação, rotação e escala, porém sensíveis a ruído. Os descritores de Hu são formados a partir dos momentos centrais da região em análise e foram usados por [Jye et al. \(2017\)](#) e [Bakhshipour e Jafari \(2018\)](#);
11. Técnicas de características locais: no geral, estas técnicas buscam pontos significativos na imagem (geralmente na escala de cinza) que possuem alto valor de informação para representar o objeto. Assim, pontos com alto contraste e de borda são detectados e eles podem ser comparados com o de outras imagens. Como descritor então, pode-se agrupar as imagens que possuem muitos pontos em comuns como objetos similares. Entre eles, pode-se citar:
  - a) *Scale-invariant feature transform* (SIFT): seleção de pontos de interesse no objeto invariantes a escala, extraído então um descritor local por ponto, permitindo então compará-los com os pontos de interesse de outra imagem, apresentando similaridades caso os objetos sejam também similares. Utilizado por [Qu et al. \(2019\)](#) e [Santiago et al. \(2019\)](#);
  - b) *Speed up robust features* (SURF): este algoritmo segue a mesma estrutura do SIFT, porém com algumas diferenças em suas etapas, criando um descritor que é mais rápido, sendo utilizado por [Sabu et al. \(2017\)](#) e [Hameed e Üstündağ \(2021\)](#);
  - c) *Accelerated KAZE Features* (AKAZE): também similar ao SURF e SIFT, ganhou popularidade por ser de código livre. Sua maior diferença é que, ao analisar os pontos de interesse em diferentes escalas, não utiliza métodos lineares com o intuito de preservar melhor as bordas. Também aplicado por [Qu et al. \(2019\)](#);

- d) *Histograma de Gradientes Orientados* (HOG): também é uma técnica de características locais, criando blocos a partir da contagem de ocorrências de um gradiente de orientações formado a partir do contraste normalizado na vizinhança do ponto. Como o método cria muitos dados redundantes, se faz necessário a aplicação de técnicas capazes de reduzir a dimensionalidade dos dados. Este método foi aplicado por [Jye et al. \(2017\)](#), [Pawara et al. \(2017\)](#), [Sabu et al. \(2017\)](#) e [Lyu et al. \(2020\)](#).
12. **Descritores de Fourier:** métodos que convertem o formato espacial da folha para o domínio da frequência (tem como entrada a imagem binária), a partir da transformada de Fourier. A partir disso, cria-se um vetor a partir do histograma de harmônicos de frequência. São no geral invariantes à escala, rotação e translação, porém perdem informações de forma locais. Entre suas variações, há o Descritor Genérico de Fourier (GFD), baseado em uma transformação de Fourier polar ([ZHANG; LU, 2002](#)), gerando um vetor com dados referentes a frequências radiais e angulares presentes na forma do objeto. Foi um dos descritores também aplicados por [Bakhshipour e Jafari \(2018\)](#);
13. **Assinaturas:** descritores que representam a forma do objeto a partir de uma função 1-dimensional obtida a partir do contorno. Entre uma das assinaturas, há a Sequência Distância Centro (CDS), vetor formado a partir da distância euclidiana entre os *pixels* que compõem o perímetro e o centroide da folha. Como gera muitos dados que possuem bastante redundância, é necessário depois reduzir a dimensionalidade do descritor. Logo, pode apresentar alto custo computacional, além de ser sensível a ruído. [Lyu et al. \(2020\)](#) desenvolveram uma assinatura, chamada de descritor radial de objeto, em que em vez de usarem o centroide como âncora eles utilizaram uma semente que representa o centro aproximado de objetos com formatos radiais, o que forma uma curva.

Os pesquisadores também desenvolveram ou aplicaram descritores menos utilizados, como é o caso de [Li et al. \(2020\)](#), que trabalhando com uma imagem 3D também utilizaram o ângulo de inclinação entre a folha e o plano horizontal na extração de características. Já [Bakhshipour e Jafari \(2018\)](#) adotaram conjuntamente descritores de razão não linear entre comprimento e área. [Wang et al. \(2019\)](#) utilizaram apenas a borda e analisaram pontos côncavos e convexos de forma diferenciadas, extraindo informações locais de contorno, contando inclusive o número de padrões carrilhados presentes na margem.

[Yang \(2021\)](#) também utilizou a borda da folha, operando com um descritor de forma baseado em triângulos de multi-escalas. Para cada região, foi analisado propriedades dos triângulos que podem ser realizados com os *pixels* vizinhos que pertencem ao contorno, que são a curvatura da ponta, orientação e a distância dela para o centroide do triângulo (o número de triângulos possíveis para cada ponto e suas escalas é calculado previamente). O dados puderam então passar por uma DFT, selecionando apenas um número fixo de coeficiente de baixa ordem, para que se possa reduzir a sua dimensionalidade.

Também utilizando os pontos presentes no contorno, [Zhang et al. \(2020\)](#) obtiveram o descritor contexto de forma (*shape context*), no qual o contorno é fragmentado em pequenos pontos e é construído para cada ponto um histograma, em que se conta quanto dos outros pontos do perímetro estão em cada região, mapeada pelo logaritmo da distância normalizada e pelo ângulo das coordenadas polares.

[Chen et al. \(2021\)](#) e [Chen et al. \(2022\)](#) criaram descritores semelhantes, em que a partir de um *pixel* fixo na extremidade do eixo maior, desenvolve-se árvores (semicírculos) ou triângulos de diferentes escalas a partir de dois *pixels* equidistantes no contorno, sendo que o ângulo de abertura formado por tais figuras é armazenada em um vetor descritor de forma.

Por outro lado, [Khan et al. \(2020\)](#) analisaram a planta por completo utilizando um algoritmo de esqueletização, capaz de identificar os ramos e detectar assim corretamente o número total de folhas, processo semelhante ao realizado por [Chen et al. \(2019\)](#). Já [Patrón et al. \(2018\)](#) utilizaram um filtro de Sobel na folha para detectar a borda e as nervuras da folha, aplicando depois a transformada de Wavelet em nível 1 e construindo um descritor como um vetor de dimensão 8 com dados estatísticos da transformada (média e desvio padrão).

### 2.1.6.2 Descritores de textura

Ao contrário dos descritores de forma, descritores de textura e de cor consideram a informação de cor ou níveis de cinza dentro da região de interesse. Enquanto os descritores de cor utilizam o valor de cor de cada *pixel* individual com foco maior, os de textura consideram a composição do *pixel* com sua vizinhança, procurando padrões por toda a imagem. Assim, a imagem binária obtida pela segmentação é aplicada como uma máscara na imagem original. Entre os mais utilizados na literatura, se tem:

1. Matriz de coocorrência: é uma matriz composta pelo número de combinações diferentes dos valores de intensidade que ocorrem em uma imagem, formando uma tabela de probabilidades possíveis. A partir destas medidas, é possível obter outros descritores, porém pode-se montar um vetor com os próprios valores da matriz. Foi utilizada por [Jye et al. \(2017\)](#) e [Gayatri et al. \(2021\)](#). [Chen et al. \(2021\)](#) e [Chen et al. \(2022\)](#) também a empregaram, calculando os valores para as diferentes escalas da intersecção de suas figuras geométricas com a folha. Entre estes descritores, também conhecidos como momentos de Haralick, se tem:
  - a) Energia: também chamado de segundo momento, representa o grau de uniformidade da imagem;
  - b) Entropia: reflete a complexidade ou não-uniformidade da região;
  - c) Contraste: também chamado de momento inercial, representa a clareza da folha;
  - d) Homogeneidade: conhecido como momento da diferença inversa, resulta em altos valores para imagens com níveis de intensidade de cor que variam pouco;

- e) Média de luminância;
  - f) Variância: obtém a dispersão dos dados em torno da média da matriz, obtendo um valor para a interação entre *pixels* vizinhos na horizontal e outro para a vertical;
  - g) Correlação: utilizada para calcular a similaridade entre os elementos da matriz de coocorrência, medindo a dependência linear de *pixels* em relação a sua vizinhança.
2. Sequência de Entropia (EnS): descritor baseado na aplicação de uma rede neural acoplada a pulsos (PCNN) no objeto, gerando uma imagem binária para cada pulso. A fim de transformar estes dados 2D para 1D, desenvolve-se um vetor com a somatória da entropia deles (MA et al., 2002). Este descritor é invariante a rotação, escala e translação. Zhang et al. (2020) utilizaram tal método, porém utilizando pulsos de saída dupla (DPCNN) para obter os vetores de entropia;
  3. Padrões Binários Locais (LBP): algoritmo que gera uma informação de textura para cada *pixel*, a partir da comparação se o seu valor é maior ou menor do que o dos seus vizinhos. A partir deste dado, é possível gerar um número seguindo a vizinhança em determinada ordem. O descritor então é gerado com base no histograma contendo a frequência que tais números aparecem por toda a imagem. Em seu desenvolvimento, pode-se adicionar etapas para torná-lo mais invariante a escala e rotação e mudanças uniformes de iluminação e contraste. Le et al. (2020) o utilizaram com vizinhança 8, 16 e 24, aplicando tanto na imagem em escala de cinza quanto na imagem referente apenas ao contorno da planta. Já Yang (2021) usaram uma versão do descritor com histogramas de Fourier, tornando-o mais robusto a rotação, enquanto Hirasen e Viriri (2020) adaptaram para o *Local Directional Patterns*, que é mais robusto a ruído e mudanças não uniformes de iluminação. Florindo e Metze (2021) criaram um descritor próprio com base no LBP.

Há também como aplicar outros métodos para obter descritores de textura diferentes. Um exemplo é o uso do FracDim para análise de textura, adicionando a informação do nível de intensidade de cor como uma nova dimensão da imagem. Assim, enquanto como descritor de forma se trabalha no  $\mathbb{R}^2$ , ao descrever a textura pelo FracDim Volumétrico se trabalha no  $\mathbb{R}^3$ , computando a dimensão fractal Bouligand–Minkowski da imagem cinza (BACKES et al., 2009) ou pelo algoritmo de dimensão multi-escala (BACKES; BRUNO, 2010). O descritor de Fourier também pode ser utilizado para analisar texturas, aceitando como entrada a imagem em escala de cinza em vez da imagem binária (ABDESSELAM, 2009).

Por outro lado, Kamath et al. (2020) aplicaram os filtros de textura de Laws, que são obtidos a partir da convolução de 25 máscaras 5x5 obtidos a partir da combinação de cinco vetores de dimensão 5. Em seguida, as imagens passaram por um filtro de média 15x15 e foram normalizadas e combinadas em catorze imagens texturizadas. Os descritores finais são obtidos a partir das propriedades estatísticas de cada imagem, i.e., a média, desvio padrão, distorção, curtose e entropia.

Florindo e Bruno (2017) propuseram um novo descritor de textura baseado na transformada discreta de Schroedinger. O descritor foi comparado com outros de textura, como matriz de coocorrência, a partir de uma Análise de Discriminante Linear e mostrou resultados melhores. Já Imanov e Alzouhbi (2019) utilizaram como descritor de textura parâmetros estatísticos retirados da matriz de codificação *run-length*. Essa matriz tem dimensão equivalente ao número de níveis de cinza possíveis versus a dimensão máxima da imagem. Assim, dado uma direção, percorre-se a imagem e anota o tamanho e a intensidade das sequências de *pixels* iguais que aparecerem. Esse procedimento então é aplicado em cada um dos canais RGB e é calculado cinco parâmetros para cada um, gerando como descritor final um vetor de tamanho 15.

### 2.1.6.3 Descritores de cor

A maioria dos descritores são baseados no primeiros momentos estatísticos de cor, podendo tanto ser aplicados em canais individuais dos espaços de cor como na imagem em escala de cinza. A maior vantagem dos momentos de cor é que possuem baixo custo computacional e pouca dimensionalidade, o que os faz serem bastante utilizados em aplicações em tempo real. Os descritores mais utilizados então são:

1. Média: valor médio em um canal de cor da imagem;
2. Desvio padrão: relação da variação de distribuição do valores em um canal de cor;
3. Distorção: descreve a informação sobre a distribuição de cor no objeto, em relação ao quão simétrico ou não ela é;
4. Curtose: também descreve a distribuição da informação, mas referente a quão próximo é de uma distribuição normal e se é mais acentuada ou aplanada;
5. Histogramas de cor: descreve a distribuição de cores no espaço de cor escolhido, contando a frequência que um intervalo de cor aparece na imagem. Apesar de ser invariante a translação e rotação, não possui informação espacial da distribuição de cor e dependendo do número de intervalos, podem gerar dados com alta dimensionalidade. Zhang et al. (2021) utilizaram tal descritor, concatenando os três histogramas de  $L^*$ ,  $a^*$  e  $b^*$  em um só vetor e obtendo o logarítmico da FFT (*Fast Fourier Transform*) dele.

Outros descritores de cor podem ser desenvolvidos utilizando critérios menos usuais. Arunachalam e Andreasson (2021) utilizaram seis índices de vegetação (índice verde de folha, somatória de refletância verde, índice normalizado de diferença verde-vermelho, razão vermelho para verde e índice normalizado ExG ) e dois relacionados a vitalidade da plantação, sendo um para estimar o teor de clorofila e outro o teor de nitrogênio da planta.

## 2.1.7 Classificadores

Com as características extraídas, comumente seleciona-se um classificador para realizar o trabalho de identificação das espécies de plantas a partir delas. Os classificadores podem então ser separados em duas classes: os supervisionados, em que tem como entrada os vetores de características e um rótulo a qual dos grupos a amostra pertence; e os não supervisionados, em que estes dados não são demarcados previamente e espera que o classificador os distinga a partir de similaridades presentes entre eles (RAJORIYA; TIWARI, 2020).

Quando há muitos dados de entrada, reduzir a dimensionalidade deles a partir da busca de redundância de informação se torna desejável, sendo que métodos variados como a Análise de Componentes Principais (PCA) e a Eliminação Recursiva de Características (RFE) podem ser utilizados (FLORINDO; BRUNO, 2017; KAMATH et al., 2020). Com o vetor de características de tamanho desejado, a última etapa é o treinamento dos classificadores, em que se separa parte do *dataset* para ser usado no treino e outra para o teste do classificador treinado. A maioria dos classificadores utilizados pelos pesquisadores da área de identificação da espécie de plantas são métodos supervisionados, descritos a seguir:

### 2.1.7.1 *Support Vector Machines* (SVM)

Classificadores binários capazes de separar os dados em duas classes distintas, a partir da busca de um hiperplano capaz de separar os dados de amostra em duas regiões diferentes. Caso os dados não sejam linearmente separáveis, funções de kernel são aplicadas para transformar o espaço de dados em um com dimensão maior, em que eles sejam agora lineares. Este classificador é altamente dependente das amostras utilizadas pelo treinamento. Depois de treinado, os dados são classificados a partir da região em que pertencerem (BOSER et al., 1992).

Ele pode ser adaptado para lidar com mais de uma classe, geralmente aplicando uma abordagem em que se separa uma classe de todas as outras com o classificador, repetindo tal operação com as classes agrupadas anteriormente até que haja a situação trivial de duas classes. As maiores vantagens do SVM é que ele permite uma boa capacidade de generalização, versátil devido a diferentes funções de *kernel*, capacidade de controle das soluções por otimização de margem e robusto a enviesamento. Porém, entre suas desvantagens principais há o treinamento lento, limitações em velocidade e tamanho tanto no treinamento quanto no teste, tendência a sobreajuste quando o tamanho da amostras é bem menor que o número de características e não gera por si só uma estimativa de probabilidade (SABU; SREEKUMAR, 2017).

Entre os estudos analisados, o SVM foi o mais utilizado. Os trabalhos relacionados que o usaram pode ser visto na Tabela 1, em que a primeira coluna apresenta os estudos analisados, enquanto na segunda se tem o *dataset* do trabalho e a porcentagem dele que foi utilizado para o treino do classificador. A terceira coluna apresenta os tipos de características que foram extraídas para o método, ao passo que na quarta coluna há o tipo de *kernel* utilizado, ou seja, linear,

polinomial, Gaussiano ou uma Função de Base Radial (RBF), com os valores dos parâmetros caso os pesquisadores tenham explicitado. Por fim, na última coluna mostra a acurácia dos SVMs desenvolvidos e aplicados nos diferentes *datasets* (representados pela letra inicial deles). Como muitos estudos utilizaram *datasets* diferentes, comparar os classificadores se torna uma tarefa não intuitiva.

Tabela 1 – Visão geral de exemplos de aplicação do SVM.

Estudo	Dataset (% treino)	Tipo de características	Kernel	Acurácia
Jye et al. (2017)	KLU University (80%)	Forma e textura	RBF, com $\gamma = 0,01$ e $C = 3$	83,3%
Pawara et al. (2017)	AgrilPlant, LeafSnap e Folio dataset (80%)	Forma	Linear	79,43% (A), 72,63% (L) e 92,78% (F)
Bakshpour e Jafari (2018)	Dataset próprio (60%)	Forma	RBF, com $\gamma = 0,68$ e $C = 51$	93,33%
Imanov e Alzouhbi (2019)	Dataset próprio	Textura	Linear	96,7%
Santiago et al. (2019)	Dataset próprio (60%)	Textura	RBF	71,6%
Zhang et al. (2019)	Dataset próprio A (70%) e B (75%)	Forma, textura e margem	-	83,33% (A) e 79,83% (B)
Le et al. (2020)	"bccr-segset"dataset	Textura	Polinomial de 2ª ordem e RBF, com $\gamma = 10^{-5}$ e $C = 30$	RBF 98,63% (melhor entre os dois kernels)
Hirasen e Viriri (2020)	Flavia (35%) e Swedish dataset (26%)	Textura	-	83,21% (F) e 97,70% (S)
Zhang et al. (2020)	Flavia (49,5%), Swedish (49,3%), MEW2012 (49,7%) e ICL (49,9%) datasets	Forma e textura	Linear	98,53% (F), 97,93% (S), 94,19% (M) e 94,22% (I)
Florindo e Metzger (2021)	KTH-TIPS2b (25%), UIUC (50%) e UMD (50%) datasets	Textura	Linear	60% (KTH), 95,7% (UIUC), 98,3% (UMD)
Gayatri et al. (2021)	-	Textura	-	98,13%

### 2.1.7.2 k-Vizinhos mais Próximos (kNN)

Método de classificação não paramétrico baseado na distância mais próxima da entrada em relação aos dados de treino no espaço de características, em que o treinamento baseia em guardar os dados de entrada e o rótulo de classe correspondente a cada um. No processo de classificação, a imagem não catalogada tem seu vetor de características comparado com o  $k$  vizinhos mais próximos no espaço, recebendo o mesmo rótulo igual ao da maioria da vizinhança. Se  $k = 1$ , o objeto é designado como da classe do ponto com menor distância (ALTMAN, 1992).

Como o resultado do classificador está intimamente relacionado aos dados de treino, quantos maior a quantidade de dados utilizados nesta etapa, melhor será a qualidade dele.

Ao contrário do SVM, para o caso de múltiplas classes não é necessário adaptar o algoritmo. Porém, devido a sua simplicidade, ele é sensível ao ruído originado pelo acréscimo de entradas irrelevantes, é considerado um aprendiz preguiçoso e exige alto custo computacional pois precisa calcular a distância para cada amostra (SABU; SREEKUMAR, 2017). Para evitar isto, um vetor de características pequeno é desejável, o que também previne sobreajuste (HIRASEN; VIRIRI, 2020). Entre os estudos analisados, o kNN foi o segundo mais utilizado, como os estudos apresentados na Tabela 2, em que a quarta coluna especifica o tipo de distância que foi utilizada pelo classificador.

Tabela 2 – Visão geral de exemplos de aplicação do kNN.

Estudo	Dataset (% treino)	Tipo de características	Distância	Acurácia
Pawara et al. (2017)	AgrilPlant, LeafSnap e Folio dataset (80%)	Forma	-	38,13% (A), 58,51% (L) e 84,30% (F)
Sabu et al. (2017)	Dataset próprio (50%)	Forma	Distância euclidiana	99,6%
Imanov e Alzouhbi (2019)	Dataset próprio	Textura	Similaridade de cossenos	96,7%
Zhang et al. (2019)	Dataset próprio A (70%) e B (75%)	Forma, textura e margem	-	76,28% (A) e 93,42% (B)
Hirasen e Viriri (2020)	Flavia (35%) e Swedish (26%)	Textura	Distância de Hamming	96,94% (F) e 97,22% (S)
Yang (2021)	Flavia, Swedish e MEW2012 dataset (70%)	Forma e textura	Distância ponderada normalizada	99,16% (F), 98,48% (S) e 95,64% (M)
Chen et al. (2022)	Datasets de Soja e Amendoim	Forma e textura	Distância obtida por algoritmo <i>Deep Fusion</i>	81,69% (S) e 50,12% (A)

### 2.1.7.3 Representação esparsa

Também conhecido como aproximação esparsa ou do inglês *sparse representation based classification* (SRC), este método aplica uma solução esparsa para um sistema de equações lineares. A partir dos dados de entrada, constrói-se um dicionário supercompleto. Assim, cada imagem de teste pode ser representada como uma combinação linear das imagens utilizadas no treino do classificador, utilizando o menor número possível de imagens deste conjunto. Logo, procura representar de forma mais esparsa uma amostra pelo menor número de coeficientes não nulos em termos do dicionário e determina a classe ao qual ela pertence pelo menor erro possível de reconstrução, em que se espera que tais coeficientes correspondam apenas a imagens de mesma classe (YANG et al., 2007). Zhang et al. (2019) aplicaram este classificador em dois *datasets* e obtiveram uma acurácia de 87,18% no dataset A e 96% no B. Já Zhang et al. (2021) conseguiram como resultado 94,51% ao aplicarem em um dataset próprio.

Ao mesmo tempo que o SRC assegura a esparsidade enquanto perde informação local dos dados de treino, pode-se adicionar um fator de peso para preservar a similaridade entre os



dados de teste e os vizinhos nos dados de treino, diminuindo esta perda. Esta versão ponderada do classificador (WSRC) foi utilizado por [Zhang et al. \(2017a\)](#) no ICL *dataset*, com 20% das imagens separadas para treinamento, em que a função de peso é baseada na distância de Jaccard, produzindo uma acurácia final de 93,42%.

De modo semelhante, [Zhang et al. \(2017b\)](#) aplicaram o WSRC no ICL *dataset*, utilizando 1100 imagens para o treino de 21 classes (aproximadamente 6,53% do dataset), resultando em uma acurácia de 88,27% na classificação de todas as outras imagens, com tempo de execução total de 1218 segundos (cerca de 20 minutos). Eles também adaptaram o classificador para uma versão ponderada discriminante (DWSRC), em que se integra a esparsidade, as informações globais e as locais da estrutura dos dados de treino. Nesta nova versão, obtiveram uma acurácia relativamente maior de 91,12% em um tempo de execução 36% mais rápido.

#### 2.1.7.4 Rede Neural Artificial (ANN)

Sistema computacional de classificação baseado nas redes biológicas dos neurônios. Esta rede é composta de nós conectados, que recebem um sinal, o processam por funções não-lineares e enviam o resultado para os nós conectados a eles, formando camadas. A cada nó (neurônio) ou conexão é associado um peso, que é ajustado durante o processo de treinamento do classificador. Pode-se também ser associado valores de limiares, ao qual permite que um sinal só seja transmitido caso for maior que um valor estabelecido.

Pode-se também utilizar retro-alimentação (*feed-forward back-propagation*), em que os estímulos nas camadas finais são utilizados também para reajustar os nós das camadas iniciais. No fim, realiza sucessivas iterações até diminuir o erro da saída da rede neural e o rótulo correto dos dados de treinamento. Entre as maiores vantagens deste tipo de classificador estão o requerimento de menos testes estatísticos e a habilidade de detectar implicitamente relações entre variáveis dependentes e independentes, mesmo as não-lineares. Já como desvantagem se apresenta o alto custo computacional e uma tendência de sobreajuste ([SABU; SREEKUMAR, 2017](#)).

[Patrón et al. \(2018\)](#) utilizaram uma rede neural com duas camadas ocultas com funções de ativação sigmoide tangencial, cada uma com 50 neurônios, além de uma camada de saída com 31, com funções lineares. Os pesos foram ajustados pelo método de treinamento gradiente conjugado escalonado. A acurácia do classificador foi de 95,70%, em que foram usado 70% das imagens de dataset próprio para o treinamento da rede.

[Bakhshipour e Jafari \(2018\)](#) avaliaram o uso de uma ANN, entretanto utilizando somente duas camadas, sendo uma com função de ativação do tipo sigmoide tangencial e outra do tipo sigmoide logarítmica. Adicionalmente, estes autores utilizaram para controlar essa ANN um algoritmo de Levenberg-Marquardt ([MARQUARDT, 1963](#)). Assim, verificaram que, após o treinamento utilizando 60% de um dataset próprio, uma acurácia final igual a 92,50% foi encontrada. É importante observar, com base nessa revisão da literatura, que nesse caso a rede

avaliada foi uma rede do tipo ANN e não uma rede CNN, o que pode ter levado a esse resultado verificado de acurácia inferior ao obtido com o uso de um classificador do tipo SVM que proporcionou uma acurácia de 93,33%.

Por fim, [Jye et al. \(2017\)](#) aplicaram uma rede neural de 3 camadas, sendo apenas uma delas oculta. Na camada de entrada há 158 nós, 10 na oculta e 3 na de saída, todos treinados também com o algoritmo gradiente conjugado escalonado. Eles utilizaram 70% das imagens do dataset da KLU University para o treinamento, obtendo acurácia de 83,3%. Ademais, [Pawara et al. \(2017\)](#) construíram uma rede bem semelhante com 512 neurônios e taxa de aprendizado de 0,001, obtendo acurácia de 74,63% quando aplicado no *AgrilPlant dataset*, 79,27% no *LeafSnap dataset* e 92,37% no *Folio dataset*.

#### 2.1.7.5 Árvore de Decisão e *Random Forest* (RF)

Uma Árvore de Decisão é um classificador que é modelado a partir de sua contraparte biológica, em que uma árvore é criada para todos os dados e tem como objetivo minimizar o erro em cada folha. As folhas representam classes distintas enquanto os ramos representam as características que levam o dado a cada classe. Apesar de que pequenas alterações nos dados de entrada possam causar grande mudanças na árvore, seu uso é simples e tem predição rápida ([SABU; SREEKUMAR, 2017](#)). [Zhang et al. \(2019\)](#) utilizaram dois *datasets* para avaliar tal método, obtendo acurácia de 79,49% no dataset A e 71% no B.

O RF é um método de classificação baseado em várias árvores de decisão, capaz de realizar tanto classificação como regressão de dados. Ele lida bem com relações não-lineares e dados discrepantes, porém ao diminuir a variância que as árvores de decisão introduzem, aumentam um pouco o enviesamento. Múltiplas árvores são construídas, uma para cada descritor, em que cada árvore fornece uma classificação; ao dado de entrada então é atribuído a classe do voto majoritário. [Kamath et al. \(2020\)](#) trabalharam com um dataset pequeno com apenas 24 imagens para teste e obtiveram 95% de acurácia ao utilizar o RF. Já [Zhang et al. \(2019\)](#) obtiveram uma acurácia de 90,38% quando aplicaram o RF no dataset A e 98,92% no dataset B, valores superiores aos obtidos por eles utilizando apenas uma Árvore de Decisão. [Florindo e Metze \(2021\)](#) implementaram o RF com no máximo 500 árvores em três *datasets*, obtendo acurácia de 64,5% no *dataset* KTH-TIPS2b, 96,5% no UIUC e 98,5% no UMD.

#### 2.1.7.6 Outros métodos

Entre outros métodos utilizados, nota-se a Análise Discriminante Linear (LDA), método que avalia a combinação estatística de características de forma limiar, separando os objetos em duas ou mais classes. [Florindo e Bruno \(2017\)](#) aplicaram este classificador em no dataset 1200Tex, que possui imagens de folhas escaneadas com foco na obtenção de textura, e obtiveram uma acurácia de 85,33%. Já [Florindo e Metze \(2021\)](#) obtiveram acurácia de 66,7% no *dataset* KTH-TIPS2b, 98,3% no UIUC e 99,4% no UMD utilizando tal método.

Zhang et al. (2019), na tarefa de comparar diferentes classificadores, também utilizaram a LDA, com acurácia de 87,18% no dataset A e 97,08% no dataset B. Outros classificadores avaliados por eles e ainda não citados foram o classificador Naïve Bayes, classificador probabilístico que aplica o teorema de Bayes, e a regressão logística, que pode ser usada como um classificador binário caso se determine um valor de probabilidade limiar de um dado pertencer ou não a uma classe. O classificador Naïve Bayes, para esse caso, apresentou acurácia de 83,97% no dataset A e 78,17% no B, enquanto a regressão logística apresentou 87,18% no dataset A e 96,58% no B.

Sabu e Sreekumar (2017) notaram o uso do classificador *Probabilist Neural Network* (PNN), rede neural comumente de quatro camadas. Uma camada é responsável pelo cálculo de distância do vetor de entrada em comparação com os utilizados durante o treinamento, enquanto outra avalia a contribuição de cada classe no dado de entrada e resulta em um vetor de probabilidades, que será utilizado para determinar a qual grupo a entrada pertence. Este classificador se apresentou tolerante a entradas destoantes e adaptativo a mudanças nos dados, porém tem tempo de treinamento longo, alta complexidade estrutural e muitos atributos podem resultar em sobreajuste.

Wang et al. (2017) observaram um uso predominante de classificadores kNN e SVM no processamento de identificação de plantas, com o PNN sendo o terceiro classificador mais utilizado. De forma semelhante, Rehman et al. (2019) analisaram que entre os métodos supervisionados comuns se tinha o Naïve Bayes, LDA, kNN e SVM. Eles notaram também um uso de métodos não-supervisionados, mesmo que em menor proporção, como o *K-means*, agrupamento por algoritmo de lógica *fuzzy* e modelos de mistura Gaussiana.

Eles concluíram que, entre os supervisionados, o Naïve Bayes é bom para dados que se já tem conhecimento prévio da distribuição no espaço, porém tem performance pior quando há muitas características a se analisar. Já o LDA é melhor ao analisar vetores de entrada maiores, sendo que o uso de análises discriminantes não-lineares é capaz de aumentar a acurácia deste tipo de classificador. Para esse trabalho, os autores observaram que o SVM possui bom potencial na tarefa de identificação de espécies de plantas, enquanto o kNN é melhor para dados sem conhecimento prévio, apesar do tempo alto de processamento. Ao analisar os métodos não-supervisionados, eles concluíram que o *K-means* é melhor quando os dados são bem separáveis em grupos distintos, porém os classificadores de lógica *fuzzy* e de mistura de Gaussiana lidam melhor com sobreposição.

Atualmente, também são encontradas pesquisas que utilizam CNN e suas variações, voltadas a *Deep Learning*, que não realizam a parte de extração de características, utilizando a própria imagem como entrada da rede. Os trabalhos que a usaram podem ser vistos na Tabela 3. CNN é um algoritmo composto de múltiplas camadas convolucionais, camadas de *pooling*, responsáveis por extrair as características, junto com camadas totalmente conectadas, para realizarem a classificação (GUAN et al., 2022). A disponibilidade em grande escala de máquinas capazes de realizar programação paralela em GPU faz com que a aplicação desta técnica de

*Deep Learning* seja mais viável (WÄLDCHEN et al., 2018). Outras redes neurais de maior complexidade também são utilizadas, como visto em Zielinski et al. (2022), aos quais aplicaram tal rede em dados de textura de folha e obtiveram uma acurácia entre 87% e 95,81%.

Tabela 3 – Visão geral de exemplos de aplicação de CNN.

Estudo	Dataset (% treino)	Tipo	Arquitetura	Acurácia
Andrea et al. (2017)	Dataset próprio (76%)	CNN	cNET 16 filtros	97,26%
Pawara et al. (2017)	AgrilPlant, LeafSnap e Folio datasets (80%)	CNN	GoogleNet pré-afinada	98,33% (A), 97,66% (L) e 97,66% (F)
Pearlstein et al. (2017)	Dataset próprio	Deep CNN	AlexNet	95%
Dobrescu et al. (2019)	Aberystwyth Leaf Evaluation dataset	Deep CNN	VGG-16	41%
Giuffrida et al. (2019)	Komatsuna e Aberystwyth Leaf Evaluation datasets	CNN	ResNet-50	26%
Krause et al. (2019)	BJFU100 (80%) e UHManoa100 datasets (93,3%)	CNN	ResNet18 (B) e Inception-ResNet-v2 pré-afinada (U)	97,80% (B) e 89,21% (U)
Vit et al. (2019)	Dataset próprio	Region-based CNN	ResNet-101	-
Chen et al. (2021)	Datasets de Soja e Amendoim	CNN	ResNet-34 pré-afinada	68,28% (S) e 72,92% (A)
Hameed e Üstündağ (2021)	Plant Village dataset (70%)	Deep CNN	GoogleNet, ResNet, AlexNet e Inception	93,1 - 99,9 %
Kritsis et al. (2021)	Dataset próprio (90%)	CNN	EfficientNet-B4	92%
Parashar et al. (2021)	Dataset próprio (60%)	CNN	ImageNet e ResNet101	86,1%
Reddy et al. (2021)	Leafsnap, UCI Leaf, PlantVillage, Flavia e Swedish datasets (80%)	CNN	-	100% (F), 100% (S), 100% (U), 90% (P) e 98% (L)
Ren et al. (2021)	Sorghum-100 dataset	CNN	ResNet50 pré-afinada	78,78%
Almazaydeh et al. (2022)	Mendeley (40%)	Region-based CNN	ResNet-101	97,5%
Li et al. (2022)	Dataset próprio (80%)	CNN	CSPDarknet53	92,6%
Malik et al. (2022)	PlantCLEF2015 (80%) e UBD dataset (80,5%)	CNN	ImageNet pré-afinada e EfficientNet-B1	87% (P) e 84% (U)
Preethi et al. (2023)	Cannabis (88,2%) e Aberystwyth Leaf Evaluation dataset (78,8%)	CNN	-	94,8% (C) e 96,1% (A)

Apesar de obter bons resultados comparados a técnicas tradicionais, é necessário muitos dados para treinamento deste tipo de rede neural (ZHU et al., 2018). Adicionalmente, as características aprendidas de uma rede neural profunda são muito específicas para o conjunto de dados de treinamento, o que significa que, se não forem bem construídas, provavelmente não terão um desempenho adequado para imagens diferentes do conjunto de treinamento (O'MAHONY et al.,

2020). Inclusive, o processo de treinamento apresenta um custo computacional alto, porém há medidas de diminuir o tempo deste processo sem alterar tanto a acurácia (HUANG et al., 2017). Portanto, técnicas tradicionais são mais usadas quando o problema pode ser simplificado para ser utilizado com microcontroladores de baixo custo.

Outros autores observaram que um dos maiores problemas ao se comparar diferentes classificadores e métodos utilizados entre pesquisadores é a falta de um padrão de testagem, de um *dataset* a ser sempre utilizado (WÄLDCHEN; MÄDER, 2018). Tendo em vista essa realidade, Malach e Pomenkova (2018) propuseram um método de comparação baseado no intervalo de confiança da curva Característica de Operação do Receptor (curva ROC). A curva ROC é uma representação gráfica do desempenho de um classificador binário, em que se analisa a relação entre a taxa de falsos positivos e a de verdadeiros positivos. As duas formas mais comuns de usá-la para avaliar diferentes classificadores é analisando a área debaixo da curva (AUC), que quanto maior melhor, e outra é avaliando a sobreposição dos intervalos de confiança. Outras medidas utilizadas para avaliação de classificadores podem ser obtidas a partir da matriz de confusão, que é feita a partir do resultado da classificação dos dados separados para teste ao passar pelo classificador já treinado (POWERS, 2008).

## 2.2 Estado da técnica

O estado da técnica tem como objetivo analisar patentes e protótipos em documentos registrados em órgãos oficiais de registro que utilizam imagens digitais para classificar espécies de plantas, principalmente plantas invasoras. Foi realizado um monitoramento tecnológico em base de patentes. As bases de dados utilizadas na busca foram a Derwent Innovations Index (DII - mundial) e Instituto Nacional da Propriedade Industrial (base de programa de computador), a partir das seguintes palavras-chaves: *invasive plant\**, *pattern recognition*, *image process\**, *decision making*, *weed*, *smart comput\**. Entre os 19 resultados, sete foram considerados relevantes a esta pesquisa.

McLoughlin (2000) desenvolveu um veículo para atuação pré-plantio, em que se usou herbicidas em plantas invasoras já presentes no solo e extraiu como informação a altura da vegetação, densidade e o tamanho das folhas e caule. O aparato tem um invólucro com iluminação própria, com a finalidade de eliminar a luminosidade não constante do ambiente na captura das imagens. Igualmente, Zhipeng et al. (2022) construíram um veículo capinador, que remove fisicamente a plantas invasoras. Ele é alimentado por bateria própria e possui controle sem fio. Similarmente, Jianmin e Zhipeng (2022) maquinaram um veículo biônico de quatro pernas feito para identificar a presença de plantas invasoras a partir de uma câmera frontal e removê-las com uma capinadeira localizada na traseira do aparelho.

Diferentemente dos anteriores, Xiaojun et al. (2021) desenvolveram um veículo para trabalhar em pasto que de fato realiza o reconhecimento de plantas invasoras. O sistema captura

imagens e as transmite para um servidor externo, no qual será realizado a identificação das plantas invasoras e a resolução das instruções para pulverização. O servidor então devolve o comando para o veículo executar. De modo paralelo, é coletado e organizado um banco de dados de plantas invasoras, para futuras aplicações de *Big Data*. [Chunlong et al. \(2022\)](#) também segmentaram as plantas invasoras a partir de imagens capturas por câmeras e com os cálculos realizados em um servidor, para depois aplicar o herbicida. Um modelo de *deep learning* é usado para segmentar com precisão e para evitar pulverização excessiva, insuficiente ou incorreta de herbicidas.

Das patentes selecionadas, [Elgart \(2020\)](#) desenvolveu um dispositivo com função de capturar imagens e diferenciar as plantas presentes em relação ao solo. As informações por ele coletadas puderam ser disponibilizadas considerando transmissão de dados para um controlador e atuador sobre as plantas invasoras. [Tavares et al. \(2022\)](#) programaram um algoritmo capaz de distinguir as plantas invasoras presentes na cultura da cana-de-açúcar utilizando imagens de satélite.

## 2.3 Considerações finais

Uma das maiores dificuldades observadas para identificação de plantas, conforme apresentado na literatura considerada, é a minimização do tempo computacional. Este é um fator limitante quando há o intuito de aplicar este processo de identificação e classificação de espécies de plantas em tempo real, junto com a obtenção de bons resultados com imagens obtidas a partir de câmeras de baixa resolução e com a presença de ruído ([PEREIRA et al., 2017](#)). [Andrea et al. \(2017\)](#), por exemplo, avaliaram o tempo de processamento do método desenvolvido em três situações: em uma máquina utilizando a CPU com o auxílio da unidade de processamento gráfico (GPU), apenas a CPU e em um sistema embarcado Raspberry Pi (RPI). Eles perceberam que enquanto com a GPU uma amostra era processada em 1,92 ms, com somente a CPU esse tempo subiu para 10,92 ms e com a RPI foi para 150,8 ms. Entretanto, é possível também observar que há dificuldades para se embarcar uma GPU ou mesmo uma CPU tradicional para aplicações em campos agrícolas, razões para qual deve-se atentar com o compromisso entre a disponibilidade real e o tempo envolvido para o reconhecimento de plantas e suas classificações. Assim, quando se tem como intuito uma aplicação prática de um algoritmo de classificação de espécies de plantas, tanto a acurácia quanto o tempo se tornam fatores importantes.

Tendo estes critérios acima mencionados como pontos de estudo, [Garbay et al. \(2021\)](#) avaliaram os impactos de redes CNN em microcontroladores, a respeito da latência, do consumo de energia e do espaço de memória necessário no *hardware*, para catorze frequências de operação distintas. Esses autores chegaram na conclusão de que o conhecimento sobre tais parâmetros é essencial para escolher técnicas de redução do tamanho das redes a fim de adaptar os algoritmos neurais dentro dos limites da aplicação. Também, [Elsts e McConville \(2021\)](#) levantaram tais

questionamentos do uso de microcontroladores de baixo custo. Neste contexto, foi notado que as estruturas de aprendizado de máquina de CNN requerem um ambiente de linguagem de programação de alto nível e várias bibliotecas, o que necessita um direcionamento mais preciso dos modelos de execução no ambiente operacional mais limitado dos microcontroladores embarcados. Esses autores concluíram também que é recomendado empregar redes CNN mais simples pois os ganhos de acurácia de modelos mais complexos não compensam os impactos de consumo de energia e custo computacional e a frequência de operação do sistema é um fator muito significativo

Dos trabalhos da literatura, poucos utilizaram mais de uma câmera para a obtenção das imagens em análise. Os que utilizam tiveram foco maior na reconstrução 3D das plantas. Por exemplo, [Nguyen et al. \(2016\)](#) utilizaram uma estrutura com 10 câmeras acopladas para reconstrução 3D de uma planta posicionada em uma plataforma, capaz de rotacionar garantindo uma melhor coleta de dados. [Li et al. \(2017\)](#) também capturaram imagens estéreo de plantas a partir de uma estrutura apoiada em um tripé, com o intuito de gerar um mapa de profundidade e a partir dele gerar uma reconstrução das plantas em nuvem de pontos. Já [Lin et al. \(2011\)](#) utilizaram imagens estéreo, com o intuito de obter características da cultura como altura, comprimento, largura e diâmetro, permitindo até uma estimativa de volume da planta, enquanto [Ge et al. \(2019\)](#) aplicaram um algoritmo binocular com um modelo de mistura Gaussiana para o reconhecimento de plantículas de brócolis em ambiente real.

[Moreno e Cruvinel \(2018\)](#), em trabalho premiado no congresso internacional SENSOR-DEVICES (*best paper*), apresentaram os princípios básicos e limitações teóricas na qualificação de um sistema de visão estéreo para uso na agricultura, enquanto [Moreno e Cruvinel \(2023\)](#) expandiram tal discussão a partir de uma abordagem de utilização de sistemas embarcados, ampliando as especificações no desenvolvimento de sistema estereoscópico voltado a caracterização de plantas invasoras.

Desta forma, tomando por base a revisão em estado da arte e em estado da técnica, foram selecionados e inferidos as principais técnicas que foram avaliadas para o desenvolvimento desta pesquisa conforme apresentado no próximo Capítulo, que trata do item Materiais e Métodos.

# Capítulo 3

## MATERIAIS E MÉTODOS

---

---

Neste Capítulo são apresentados os materiais e métodos utilizados no desenvolvimento da pesquisa, bem como etapas e técnicas utilizadas, as quais envolvem *hardware* e *software*, além de protocolos utilizados para a validação dos resultados.

### 3.1 Abordagem geral

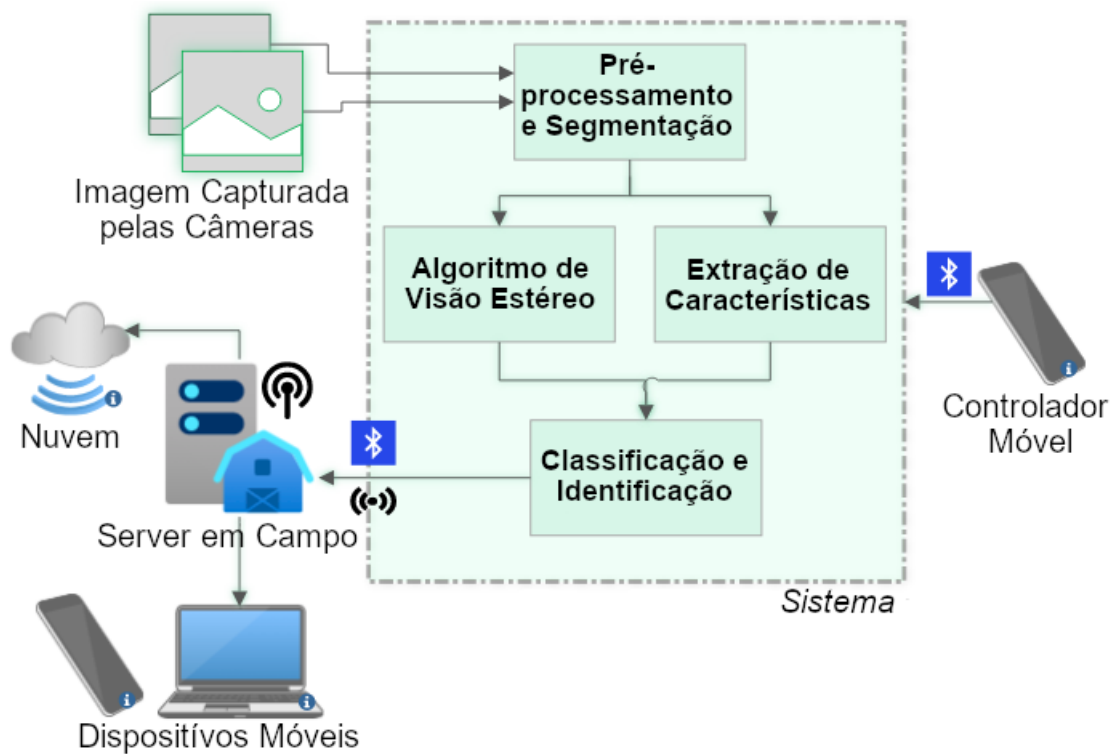
O sistema desenvolvido considera para a sua operação a captura de imagens estereoscópicas em um ambiente real de culturas agrícolas, com particular ênfase na captura das informações de plantas invasoras presentes na região de interesse, de forma a serem identificadas e classificadas a partir do uso de um algoritmo embarcado. A Figura 7 ilustra esquematicamente a estrutura do sistema (*hardware* e *software*) que viabiliza a operação de reconhecimento e classificação de padrões de famílias de plantas invasoras. Para a captura de imagens estéreo, foram utilizados dois sensores de câmeras, cujas imagens são integradas para formarem aos pares imagens estereoscópicas.

Essas imagens são pré-processadas, inicialmente considerando a filtragem de ruídos e enviesamentos derivados que constituem características intrínsecas de cada um dos sensores. Após tal etapa, o objeto de interesse é selecionado a partir de uma segmentação (considerando a imagem do lado direito e a do lado esquerda para fins da obtenção do mapa de disparidade ou informação de profundidade). A partir deste ponto, é realizada a extração de características das plantas imageadas, ao mesmo tempo que um algoritmo de visão estéreo obtém os dados de profundidade dos objetos presentes na cena. Esses dados são agora agrupados em vetores, que são utilizados para o treinamento de um classificador que auxilie na separação das espécies das famílias de plantas invasoras.

O sistema tem controle operacional realizado a partir de um dispositivo móvel, no caso um celular Android, a partir do uso de comunicação Bluetooth. Os dados processados pelo método podem ser enviados para módulos externos ao sistema, ou ainda serem armazenados em servidores locais ou em Nuvens, para usos posteriores, como por exemplo sistemas pulverizadores de agrotóxicos ou defensivos agrícolas.



Figura 7 – Esquemático do sistema.



## 3.2 Materiais

Para o desenvolvimento do sistema, foram utilizados diversos materiais mecânicos, elétricos, eletrônicos e computacionais. Dentre eles, os mais significativos incluem:

- 2 Raspberry Pi;
- 2 cartões micro SD de 32 GB;
- Adaptador cartão SD para micro SD;
- Retificador CA-CC de 127-220 V para 5 V/2 A;
- 2 cabos micro USB;
- Bateria Automotiva de 12 V e 60 Ah;
- Inversor CC-CA de 12 V para 220 V;
- 2 lâmpadas LED, uma de 4,5 W e outra de 18 W;
- Soquete de lâmpada;
- Monitor LCD (entrada HDMI);

- Cabo HDMI;
- Cabo de alimentação do monitor;
- Teclado USB;
- Mouse USB;
- 2 Câmeras Pi;
- 2 Cabos flats 30 cm para Câmera Pi;
- *Stereo Rig* ajustável;
- Cabo Ethernet;
- Pen Drive de 4 GB;
- Caixa de polietileno para proteção do sistema;
- Estrutura metálica para fixação do sistema;
- Cabos de energia, conectores, parafusos, roscas e borboletas;
- Bancada de trabalho;
- Dispositivos de medição de corrente e tensão;
- GPS Portátil modelo Garmin;
- Aparelho de celular Android.

### 3.2.1 Raspberry Pi

Raspberry Pi é uma série de mini-computadores embarcados desenvolvido no Reino Unido pela *Raspberry Pi Foundation* em associação com a Broadcom e lançados a partir de 2012. A *Raspberry Pi Foundation* também disponibiliza periféricos que podem ser utilizados com suas placas, como displays e câmeras.

O uso de RPi na agricultura vem sendo observado, devido a ser um computador de última geração com uma grande quantidade de aplicações práticas em todas as áreas de domínio (SAEED et al., 2022). O embarcado, aliado com sensores, possui várias aplicações na agricultura, como em fenotipagem, detecção de doenças, proteção de plantação, manejo de água, monitoramento de cultura, medição de parâmetros de solo e nutriente, entre outros (MATHE et al., 2022). Com a placa também é possível capturar e processar as imagens em um mesmo módulo, permitindo aplicações em sistemas estéreo (NEVES; MATOS, 2013), além de possibilitar uso abrangente de algoritmos embarcados para o uso na Agricultura de Precisão (SADDIK et al., 2022).

Para a pesquisa foi escolhido a placa da família RPi 3, modelo B+ (Figura 8), lançado em 2018, devido a uma combinação de disponibilidade de funções e menor custo. As características de *hardware* desta placa podem ser conferidas na Tabela 4, que contém as informações dos componentes integrados e a descrição dos conectores disponíveis. Este modelo tem temperatura de operação entre 0°C e 50°C e necessita de um cartão de memória micro SD com o sistema operacional instalado (recomendando pelo fabricante o Raspberry Pi OS, conhecido como Raspbian).

O modelo tem um processador BCM2837B0 Cortex-A53 (ARMv8) de 64 bits, 1 GB de memória SDRAM e velocidade de processador de 1,4 GHz. Ele deve ser alimentado por uma tensão contínua nominal de 5 V e fonte capaz de fornecer até 2,5 A de corrente. O tamanho da memória interna é definida pela capacidade do cartão micro SD escolhido, sendo recomendado no mínimo um de 8 GB. A RPi 3 +, diferentemente de modelos anteriores da família, permite comunicação sem fio *Local Area Network* (LAN) e *Bluetooth Low Energy* (BLE), possibilitando transferência de dados sem a necessidade de conexão física, a partir de um chip Cypress CYW43455.

Figura 8 – Raspberry Pi 3 model B+.

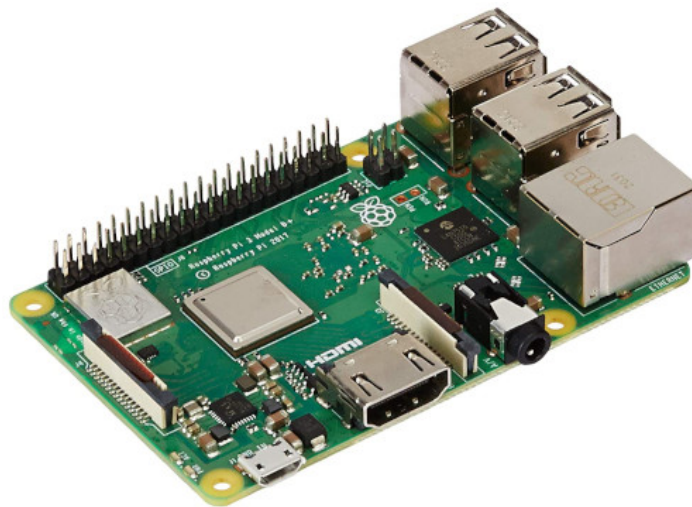


Tabela 4 – Características da Raspberry Pi 3 model B+

Processador	BCM2837B0 Cortex-A53 (ARMv8) 64-bit		
Clock	1,4 GHz	GPIO	40 pinos
Memória	1 GB SDRAM	Gigabit Ethernet	1 conector
Porta USB	4 USB 2.0	HDMI	1 conector
Camera serial interface (CSI)		Display serial interface (DSI)	
Wireless (dual band)		Bluetooth 4.2/BLE	
3,5mm 4 polos jack para saída de áudio		Slot para cartão micro SD	
Suporte Power-over-Ethernet		Entrada DC 5V/2,5A	

### 3.2.1.1 Gerenciamento de memória

Como o sistema captura e armazena muitas imagens digitais, foi necessário considerar a capacidade de armazenamento dos dados e seu limite de operação. Para isso, duas informações foram obtidas, sendo elas:

- **Taxa de escrita e leitura de dados:** tempo que demorou para o sistema escrever e ler uma quantidade grande de dados na memória. Definiu-se assim quanto tempo se deve esperar entre essas operações para que elas ocorram sem erros ou corrompimento de arquivos, podendo então inserir *delays* no código caso necessário.
- **Capacidade de memória:** como os arquivos de imagens são os que possuem maior tamanho, os usou como limitante da memória. O número de imagens possível de armazenamento então foi estimado a partir de um procedimento de captura de um número fixo de imagens em diferentes resoluções. Com essas imagens, observou o tamanho dos arquivos e calculou a média, a fim de eliminar a influência da compressão da imagem em diferentes cenários. Também foi calculado a capacidade para o pior caso, isto é, caso as imagens não sejam comprimidas.

### 3.2.2 Câmera

Foi utilizado no desenvolvimento da pesquisa duas câmeras, a Pi Camera v1 (Figura 9), cujas especificações são mostradas na Tabela 5. Entre os parâmetros mais importantes, destacam-se a distância focal fixa de 3,60 mm, a resolução máxima do sensor de 2592 x 1944 *pixels* e o ângulo de abertura da câmera de 53,50° na horizontal e 41,41° na vertical. Além disso, o foco ideal da câmera é de 1 m - ∞ e a relação sinal ruído dela é de 36 dB. Outro detalhe importante é que o RPi ajusta automaticamente o brilho e o balanço de branco da câmera, mas caso seja necessário é possível corrigir tais valores via *software*.

A conexão do sensor com a placa é dada via conector serial próprio, como poder ser visto na Figura 10. O cabo flat padrão da câmera é de 15 cm, mas foi substituído por outro, de 30 cm, para permitir que o sensor fique mais distante da placa. Com o dado do campo de visão horizontal e vertical da câmera, foi possível estimar a área do mundo real que a câmera captura, i.e, quanto de área do solo estará presente na imagem digital. Com o sensor posicionado perpendicular ao solo (Figura 11) a uma altura  $h$  e com os dados de campo de visão  $VFOV$  e  $HFOV$ , a área capturada  $A_{cap}$  pode ser calculada pela Equação 3.1.

$$\begin{cases} A_{cap} = \text{lado horizontal} \times \text{lado vertical} \\ A_{cap} = 2h \tan\left(\frac{HFOV}{2}\right) \times 2h \tan\left(\frac{VFOV}{2}\right) \\ A_{cap} = 4h^2 \tan\left(\frac{HFOV}{2}\right) \tan\left(\frac{VFOV}{2}\right) \end{cases} \quad (3.1)$$

Figura 9 – Pi Camera v1.

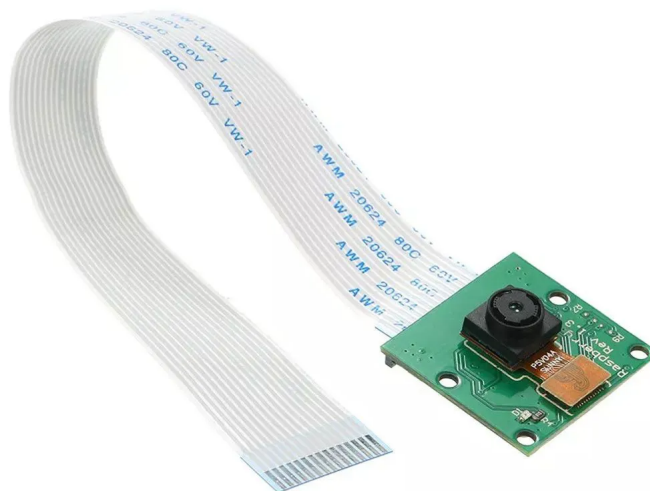


Tabela 5 – Características Câmera Pi.

Tamanho	25 x 24 x 9 mm
Resolução	5 MP
Módulos de vídeo	1080p30, 720p60, 640x480p60/90
Sensor	OmniVision OV5647
Resolução do sensor	2592 x 1944 <i>pixels</i>
Área de imagem do sensor	3,76 x 2,74 mm
Tamanho do <i>pixel</i>	1,4 $\mu\text{m}$ x 1,4 $\mu\text{m}$
Tamanho óptico	1/4"
Full-frame SLR equivalente	35 mm
Razão S/N	36 dB
Alcance dinâmico	67 dB @ 8 vezes de ganho
Foco fixo	1 m - $\infty$
Comprimento focal	3,60 $\pm$ 0,01 mm
Campo de visão horizontal (HFOV)	53,50° $\pm$ 0,13°
Campo de visão vertical (VFOV)	41,41° $\pm$ 0,11°
Razão focal (F-stop)	2,9

### 3.2.3 Bateria

Para que o sistema possa ser usado em campo de cultivo de forma independente, é necessário um sistema de alimentação de energia portátil. Foi então selecionada uma bateria automobilística Pioneira F60DG, com peso de 14,6 kg, 242 mm de comprimento, 175 mm de largura e 175 mm de altura.

Figura 10 – Conexão entre a câmera e a placa embarcada.

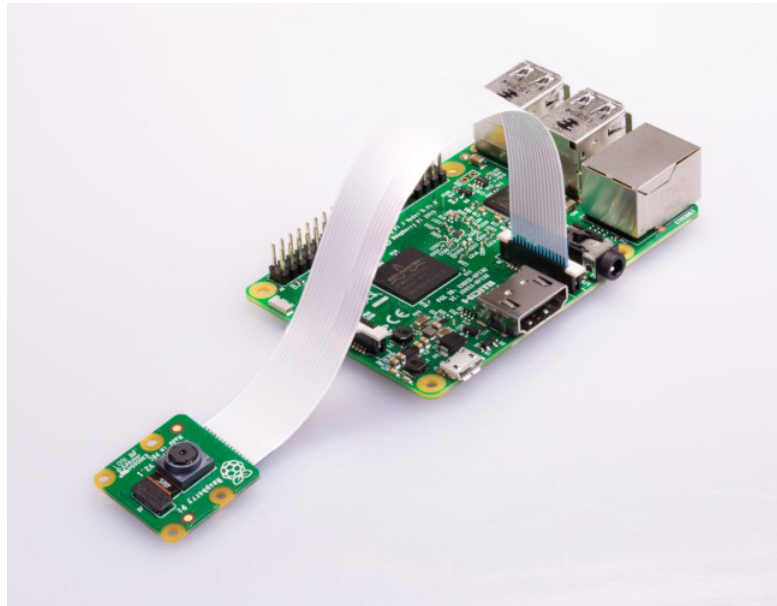
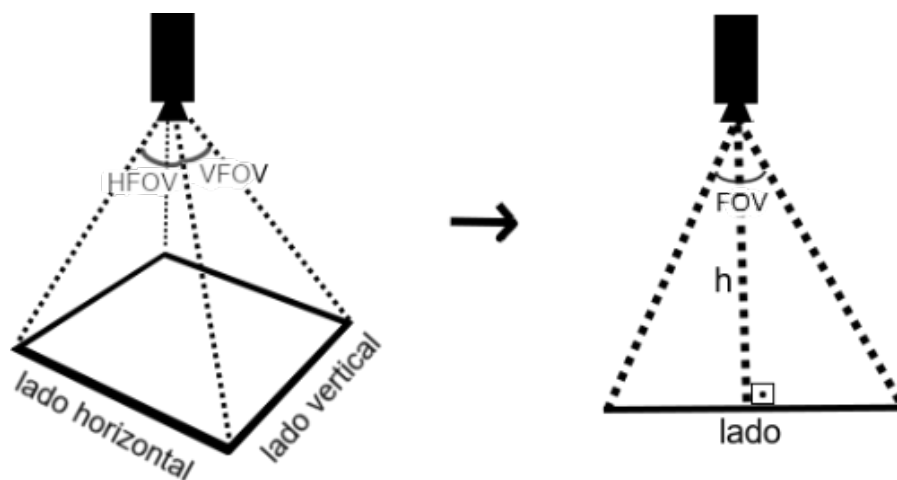


Figura 11 – Posicionamento da câmera (representação 3D e 2D).



Ela possui tensão nominal de 12 Vdc e 60 Ah de capacidade, capaz de fornecer a potência suficiente para alimentar todos os componentes. O exterior da bateria é feito de polipropileno injetado, oferecendo uma alta proteção mecânica da bateria e tornando-a mais resistente a elementos externos, o que é importante considerando o ambiente em que ela será utilizada.

### 3.2.3.1 Potência do sistema

Para calcular a potência do sistema, é medido a tensão nos terminais da bateria e a corrente no fio no polo positivo, multiplicando então os valores. Foram medidos os valores para diferentes situações de operação: o sistema em *stand-by*, com as RPis e câmeras ativas e

com cada lâmpada LED ligada ou não. Também foi mensurado a corrente de pico, registrada comumente logo após inicializar o sistema, para garantir que seu valor não esteja fora do suporte da fonte.

### 3.2.3.2 Tempo de operação do sistema

Para se obter o tempo de operação do sistema e assim saber a auto-suficiência dele, na situação em que se teve maior gasto de energia, ele foi mantido em atividade ininterrupta até desligamento automático por corrente insuficiente. O tempo então foi medido de forma externa a partir de um cronômetro e de forma interna a partir de um programa simples que registrava o tempo do sistema de 15 em 15 segundos em um arquivo de texto.

### 3.2.4 Inversor e retificador

Para poder alimentar a lâmpada LED e a RPi, escolheu utilizar um inversor para transformar a tensão da bateria em uma apropriada para os dispositivos e um retificador para estabilizar a tensão para o valor necessário para as placas.

Um inversor é um dispositivo elétrico capaz de converter um sinal elétrico contínuo para alternado, como apresentado na Figura 12. Foi utilizado o inversor Hayonik 400 W de 12 Vdc para 220 Vac  $\pm$  10% RMS, que gera uma senoide modificada de 60 Hz de frequência. A temperatura de proteção deve ser menor de 65 °C e possui 95 mm de comprimento, 40mm de largura e 164 mm de altura. O aparelho possui proteção para alta tensão de entrada maior do que 15 Vdc, alarme sonoro de baixa tensão para  $10,5 \pm 0,5$  Vdc e desligamento automático quando esta tensão diminui para  $10 \pm 0,5$  Vdc. A eficiência do dispositivo é maior do que 80 %, o que indica que pode ter perdas de energia até 20 % na transformação da energia contínua para alternada.

Figura 12 – Detalhes do inversor AC/DC.



Já o retificador utilizado é um dispositivo que realiza o trabalho contrário do inversor, transformando o sinal elétrico alternado para um contínuo. O retificador utilizado (Figura 13) permite entrada bivolt e tem como saída uma tensão fixa de 5 V, capaz de fornecer até 2 A de corrente.

Figura 13 – Fonte de alimentação com retificação em onda completa e estabilização de tensão.



### 3.2.5 Lâmpada LED

A função das lâmpadas LED é de iluminar de forma uniforme a área em que será capturada as imagens, garantindo que sombras influenciem menos na leitura dos dados presentes. Foram analisadas duas lâmpadas, uma de 4,5 W e outra de 18W, porém o sistema funciona com apenas uma instalada. Ambas são bivolt, possuem vida útil nominal de 25000 horas e temperatura de trabalho entre -10 °C e 40 °C. A lâmpada de 4,5 W, quando conectada a uma tensão de 220 V, possui corrente nominal de 0,046 A, enquanto a de 18 W é de 0,07 A.

### 3.2.6 Fenologia e fenotipagem das espécies presentes

O conhecimento da fenologia (estudo de como a planta se desenvolve ao longo de suas diferentes etapas) e fenotipagem (caracterização de recursos genéticos para atributos diversos da planta) das espécies em análise permite saber quais são as características visíveis mais importantes no processo de identificação e o momento ideal para realizar as análises de reconhecimento

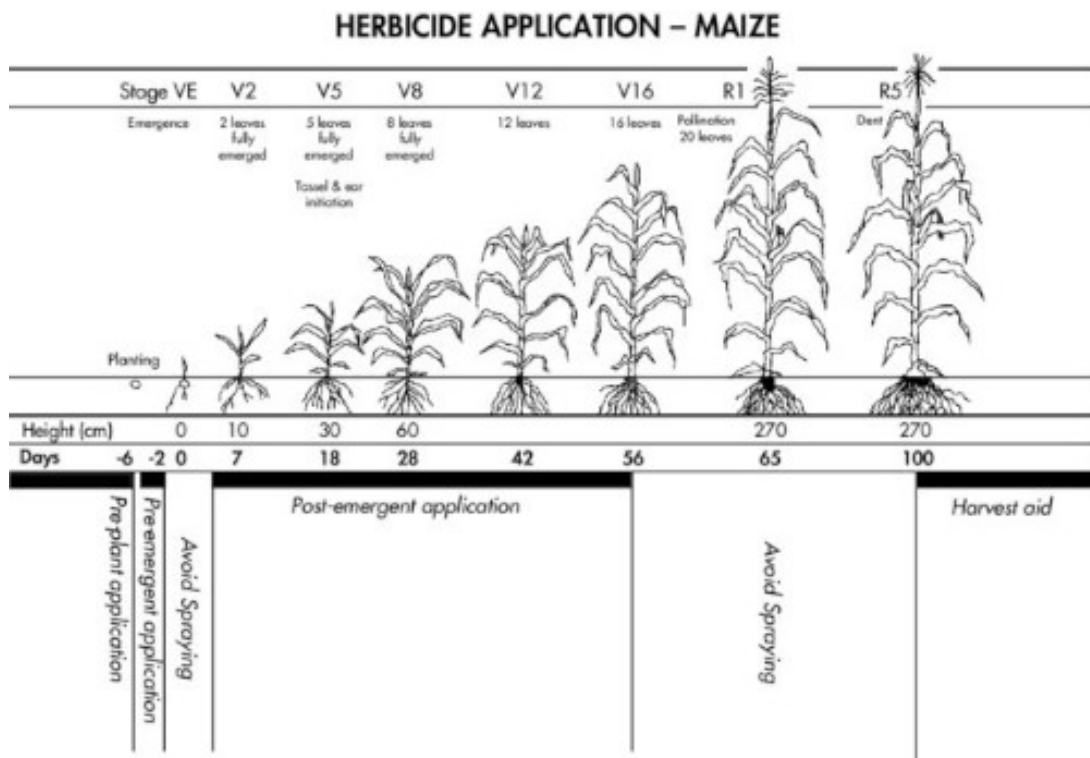


de padrões, isto é, quando deve ser feita a captura das plantas e o que deve levar em consideração na identificação das espécies.

### 3.2.6.1 Cultura do milho e suas plantas invasoras

A Figura 14 apresenta os vários estágios (vegetativos e reprodutivos) de crescimento do milho e o manejo da aplicação de herbicidas de acordo com a etapa da planta. O período crítico de interferência para a prevenção de plantas invasoras começa em torno de 15 dias após a emergência e vai até 45 dias. Nesse período, se houver plantas invasoras, a lavoura terá sua produtividade reduzida. Observa-se que o na fase em que o milho tem entre 2 folhas a 8 folhas desenvolvidas, a planta tem entre 10 e 60 cm de altura.

Figura 14 – Ciclo do milho e período ideal para aplicação de herbicidas.



Fonte: [Belfield e Brown \(2008\)](#).

A planta do milho (Figura 15) usualmente apresenta um caule principal e entre um e dois ramos laterais, folhas verdes grandes (0,3 - 1,5 m de comprimento por 5 - 15 cm de largura), lineares (retas com lados paralelos) e em formas de lança, com nervura central proeminente. Cada caule produz entre 8 a 21 folhas, que ocorrem isoladamente em lados alternados do caule, formando uma estrutura radial ([NPARKS FLORA & FAUNA WEB, 2023](#)).

Entre as plantas invasoras da cultura do milho, há a Leiteira (Figura 16), também conhecida como amendoim-bravo, que tem altura entre 40 e 60 cm, com folhas verdes largas e

mais arredondadas que ocorrem ao longo do caule, medindo de 4 a 10 cm de comprimento. As folhas inferiores são alternadas e lanceadas, enquanto as superiores são opostas ou verticiladas (EMBRAPA - Empresa Brasileira de Pesquisa Agropecuária, 2014b). Outra planta invasora é o capim-carrapicho (Figura 17), que desenvolve entre 20 e 60 cm de altura, com folhas em grande quantidade e distribuídas sobre os colmos do caule, em forma de lâmina foliar plana. As folhas medem de 10 a 30 cm de comprimento por 5 a 10 cm de largura (EMBRAPA - Empresa Brasileira de Pesquisa Agropecuária, 2014a).

Figura 15 – Plantículas do milho.



### 3.2.6.2 Cultura do amendoim e suas plantas invasoras

A Figura 18 apresenta os vários estágios de crescimento do amendoim, em DAS (dias depois da sementeira). Diferentemente do milho, o amendoim é uma planta que não cresce muito em altura, o que o faz mais suscetíveis aos impactos negativos das plantas invasoras durante todo o seu cultivo. Mesmo assim, é usual definir os 45 dias após a sementeira como o período crítico para lidar com a competição das outras plantas, para que não haja grandes impactos na produção, apesar de que dependendo das condições climáticas da região esse tempo pode subir até 100 dias (JAT et al., 2011).

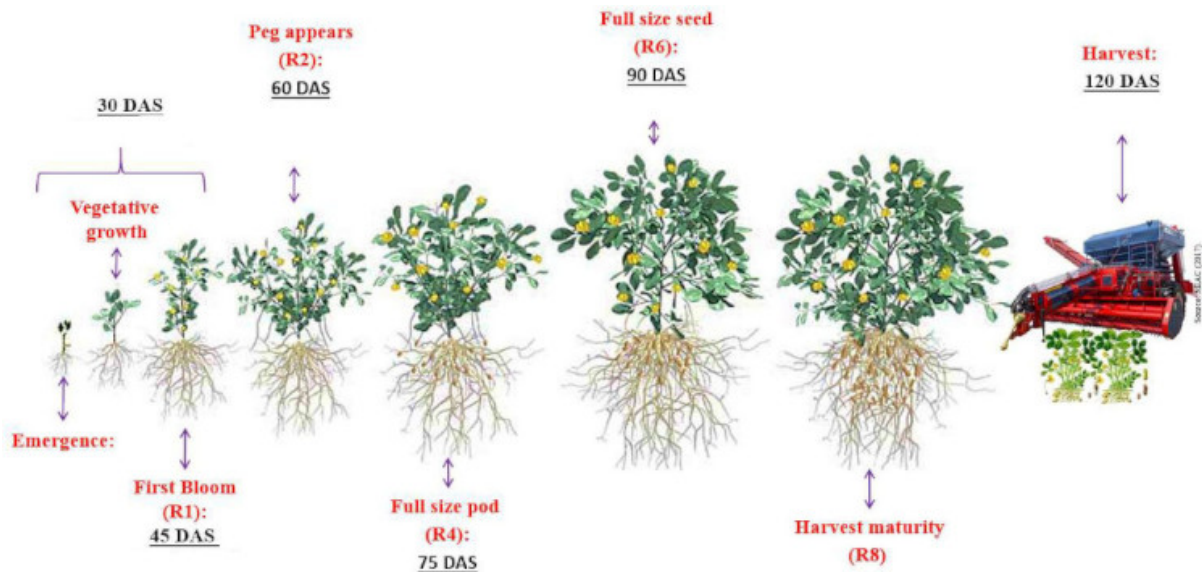
Figura 16 – Plantícula da leiteira.



Figura 17 – Plantículas do capim-carrapicho.



Figura 18 – Ciclo do amendoim.



Fonte: Carneiro et al. (2019).

A planta do amendoim (Figura 19) pode ter entre 23 a 46 cm de altura, possui folhas verdes, compostas (mais de uma folha saindo do mesmo ponto), de forma elípticas ou ovais, de 3 - 7 cm de comprimento e 2 - 3 cm de largura. As folhas têm um número par de folíolos, cada um com uma nervura central proeminente (NORTH CAROLINA STATE UNIVERSITY EXTENSION, 2023).

Entre as plantas invasoras do amendoim, se tem a mucuna-preta (Figura 20), que possui folhas largas verdes escuras, de 9 - 18 cm de comprimento e 8 - 11 cm de largura, com um nervuras bem marcadas, formato elipsoidal e levemente assimétricas. Ela possui característica trepadeira, cada folha consiste em três folíolos ligados a um pecíolo comum e pode crescer mais do que 100 cm quando adulta, apresentando uma espiga floral característica da inflorescência (SILVA et al., 2013; SANTANA et al., 2018).

Outra planta invasora é o capim-braquiária (Figura 21), que é composto por folhas que consistem em uma bainha, que envolve o caule e folhas em formato de lâminas estreitas que se afunilam nas pontas. Cada lâmina tem entre 5 - 25 cm de comprimento e 7 - 20 mm de largura e as folhas no geral tem uma cor verde pálido ou verde amarelado. A planta adulta apresenta entre 60 a 100 cm de altura (ENVIRONMENTAL WEEDS OF AUSTRALIA, 2016).

Figura 19 – Plantícula do amendoim.



Figura 20 – Plantas de mucuna-preta.



### 3.3 Técnica de IoT para transmissão de dados

Foi utilizado a técnica de Internet das Coisas (IoT), que é uma infraestrutura global de rede usada pela sociedade de informação, com uma combinação de Internet, comunicação próxima e sensores conectados, para que dispositivos possam trocar informações no âmbito de informações do sistema (SORRI et al., 2022). Seu uso permitem a coleta dados com sensores, os

processa e entrega uma saída para outro sistema a partir do uso da Internet, podendo ser acessado em diferentes locais. A arquitetura IoT utilizada três elementos principais: itens de informação que geram dados, uma rede independente e aplicação inteligente, isto é, processamento e troca de dados pela rede (ALI et al., 2015). A comunicação sem fio da RPi a torna capaz de trabalhar com projetos IoT, permitindo que objetos sejam detectados ou controlados remotamente através da infraestrutura de rede existente e reduzindo a intervenção humana (BALAMURUGAN; SATHEESH, 2017). Os sistemas IoT na agricultura podem ser separados em três módulos: setor da fazenda, setor do servidor e setor do cliente. O setor da fazenda geralmente consiste em detectar parâmetros agrícolas locais, identificar a localização e dados do sensor, transferir dados de campos de cultivo para tomada de decisão, suporte à decisão e análise de risco precoce com base em dados recentes e ação e controle com base no monitoramento da cultura. No setor do servidor, a camada de rede é responsável pela transformação confiável para a camada de aplicação. O último módulo, o setor do cliente ou camada de aplicação, coleta e processa informações, proporcionando um ambiente onde os usuários podem monitorar os dados processados pelo sistema através de um navegador web, em qualquer lugar e a qualquer hora. (PATIL; KALE, 2016).

Figura 21 – Plantícula do capim-braquiária.



O sistema nesta pesquisa considera a coleta de dados de imagem no campo, seu processamento, classificação e interpretação, o que é realizada a partir de monitoramento via comunicação serial Bluetooth por um aplicativo móvel. Uma rede Wireless Personal Area Networks (WPAN) foi montada em uma única RPi, com controle unificado e console de monitoramento para vários dispositivos sem fio. O transporte e armazenamento de dados tornam-se essenciais, com dados que podem também ser salvos em um servidor externo ou na nuvem e depois transferidos para outros dispositivos, inclusive para os equipamentos responsáveis pela pulverização do produto na lavoura.

## 3.4 Aquisição de imagens

Para a aquisição de imagens, foram utilizados protocolos de comunicação corretamente configurados e seguros (dispositivos desconhecidos não podem interagir com a rede em atuação), montou um algoritmo para a aquisição das imagens. A Camera Pi não produz dados de imagem no formato *RAW*, que são os dados da imagem não processados e não comprimidos. Assim, foi escolhido salvar as imagens capturadas no formato *PNG*. *PNG*, ou *Portable Network Graphics*, é um formato de imagem bitmap que utiliza de compressão sem perdas pelo algoritmo de Phil Katz, conhecido como DEFLATE. Ele reduz em menor escala a qualidade da imagem quando comparado com o formato JPEG (AGUILERA, 2006). A aquisição de imagens é dividida em na etapas de estabelecimento da comunicação via Bluetooth entre todos os dispositivos e a etapa de captura em si.

### 3.4.1 Estabelecimento da comunicação Bluetooth

A comunicação entre todos os dispositivos é realizada via protocolo Bluetooth, que suporta até 7 dispositivos conectados simultaneamente. A conexão do Bluetooth 4.2 pode chegar até um limite de transferência de 1 Mbs e o sinal pode alcançar 10 m de distância da placa em ambientes internos e 50 m em ambientes externos.

Um dos protocolos utilizados é o de comunicação por radiofrequência (RFCOMM). O protocolo RFCOMM é uma camada importante que oferece uma interface serial para a camada de transporte do Bluetooth, emulando um cabo de interconexão RS-232. O RFCOMM é baseado no padrão ETSI 07.10, que permite a emulação e multiplexação de várias portas seriais em um único transporte (BISDIKIAN, 2001). Também é utilizado o protocolo OBEX (*OBject EXchange*) para a transferência de arquivos, que é uma implementação em *software* do protocolo de rede FTP (File Transfer Protocol), que é executado em cima de RFCOMM (MORENO; OKAMOTO, 2011). Para garantir a segurança do sistema, ele se conecta apenas a equipamentos confiáveis e em portas específicas. Uma RPi então é definida como Mestre, responsável por receber os comandos enviados por um aplicativo no celular Android e a partir de tal comando realizar suas ações e informar a outra placa o que ela também deve fazer. Já a outra RPi é definida como Escravo, recebendo os comandos do Mestre e obedecendo-os.

A comunicação do Mestre pode ser visto no pseudocódigo 1, enquanto o da Escravo se encontra no pseudocódigo 2. Nota-se que ambos os algoritmos foram inicializados automaticamente junto com sistema e caso apresentem alguma falha de conexão, são reiniciados automaticamente. Os endereços utilizados se referem ao número MAC dos dispositivos.

### 3.4.2 Captura de imagens

Com a comunicação estabelecida, a captura de imagens é controlada via celular, como apresentado nos pseudocódigos 3 e 4. O aplicativo Android possui botões que ao serem pres-

sionados mandam de forma serial e via Bluetooth uma *string* de comando para a RPi Mestre. A partir de tal comando, é realizado a captura das imagens, a transmissão dos arquivos para o celular ou o desligamento do sistema.

---

**Algorithm 1** Pseudocódigo para comunicação: Mestre

---

**Entrada:** Endereço do escravo *end\_slave*, endereço do celular *end\_celular*, portas de comunicações *port\_1* e *port\_2*

```

1 início
2   s1 = criar_socket_bluetooth(RFCOMM)
3   s2 = criar_socket_bluetooth(OBEXFTP)
4   conectar(s1,(end_slave, port_1))
5   vincular(s2,(end_celular, port_2))
6   enquanto conexão celular não for interrompida faça
7     |   cmd = receber_dado(celular)
8     |   enviar(cmd,Escravo)
9   fim
10 fim
Saída: String de comando cmd

```

---



---

**Algorithm 2** Pseudocódigo para comunicação: Escravo

---

**Entrada:** Endereço do mestre *end\_master*, porta de comunicação *port\_1*

```

11 início
12   s = criar_socket_bluetooth(RFCOMM)
13   vincular(s,(end_master, port_1))
14   aceitar_conexao(s)
15   enquanto conexão Mestre não for interrompida faça
16     |   cmd = receber_dado(Mestre)
17   fim
18 fim
Saída: String de comando cmd

```

---

## 3.5 Função de transferência modular

Para avaliar a qualidade das imagens adquirida pelas câmeras, é utilizada a Função de Transferência Modular (MTF), a partir de cada uma delas e do conjunto. A MTF expressa quão bem um sistema óptico preserva o contraste das frequências espaciais do objeto na imagem e é um método de desempenho bem estabelecido (ZWANENBERG et al., 2019).

Projetistas de lentes e câmeras encontram desafios no desenvolvimento de sistemas com alta qualidade de imagem. O problema de maior preocupação é como otimizar os parâmetros da lente, como curvaturas e espessuras, para obter uma alta qualidade de imagem, como por exemplo, uma alta resolução de imagem. Um conjunto de otimizações foram propostas para melhorar as aberrações dos sistemas de lentes, utilizando como métrica a MTF, o termo de amplitude da Função de Transferência Óptica (OTF) que é semelhante à função de transferência do sistema linear (FANG et al., 2007; FANG; TSAI, 2008; CHEN et al., 2009).



**Algorithm 3** Pseudocódigo captura de imagens no Mestre**Entrada:** Comando *cmd*, resolução da câmera *resol*, diretório de armazenamento *dir*

```

19 início
20   para cada cmd faça
21     caso cmd== 'captr' faça
22       enviar(cmd,Escravo) // gatilho de sincronização
23       img = capturar_imagem(resol)
24       salvar_imagem(img,dir)
25       img2 = receber_dado(Escravo)
26       esperar_operação()
27       salvar_imagem(img2,dir)
28     fim
29     caso cmd== 'send1' faça
30       enviar(img,celular)
31       esperar_operação()
32     fim
33     caso cmd== 'send2' faça
34       enviar(img2,celular)
35       esperar_operação()
36     fim
37     caso cmd== 'shutd' faça
38       enviar(cmd,Escravo)
39       desligar_sistema()
40     fim
41   fim
42 fim

```

**Algorithm 4** Pseudocódigo captura de imagens no Escravo**Entrada:** Resolução da câmera *resol*, diretório de armazenamento *dir*

```

43 início
44   cmd = receber_dado(Mestre)
45   para cada cmd faça
46     caso cmd== 'captr' faça
47       img2 = capturar_imagem(resol)
48       salvar_imagem(img2,dir) // opcional
49       enviar(img2,Mestre) // envio via protocolo OBEXFTP
50       esperar_operação()
51     fim
52     caso cmd== 'shutd' faça
53       desligar_sistema()
54     fim
55   fim
56 fim

```

Assim, a função de transferência é considerada uma das principais características do sistema linear. Existem algumas propostas que apresentam seus métodos para abordar a função de transferência (BADDOUR, 2011; CHEN; YAN, 2010; LI et al., 2011). Um simples é receber

a resposta ao impulso na saída como entrada sendo um sinal de impulso puro, obtendo portanto a resposta ao impulso da função de transferência. Usando o mesmo procedimento, uma fonte pontual é considerada como o sinal de impulso para ajudar a estimar a resposta da imagem em um sistema de lentes.

A imagem da fonte pontual mostrada no plano da imagem é chamada de Função de Dispersão de Ponto (PSF), que é a transformada inversa de Fourier da OTF. Portanto, a MTF aplicada para determinar a resolução da imagem pode ser derivado do termo de amplitude da transformada de Fourier da PSF. A MTF é, em princípio, classificado em três categorias: método de varredura, autocorrelação e métodos de relação cruzada. Destes, o método de varredura parece ser mais comumente usado em muitos campos e a medição de Função de Dispersão de Linha (LSF) é igualmente preferível porque pode ser obtida de modo simples.

Então, o sensor de câmera pode ser definido levando em consideração o cálculo da LSF da lente da câmera e da MTF, que representa a resposta de magnitude do sistema óptico a senoides de diferentes frequências espaciais, ou seja, recuperadas pela transformada de Fourier da LSF. Vários aspectos-chave da instrumentação óptica estão relacionados com a implementação de uma fonte linear para um determinado sistema óptico, o impacto do tamanho finito da fonte na medição e a escolha dos elementos ópticos para a imagem da resposta de padrões específicos e sua relação com a lente utilizada no sensor da câmera.

Tomando uma fonte linear a solução para medir a MTF é em 1D ortogonalmente à direção da linha. Isso pode ser comprovado considerando uma dada fonte  $S(x, y) = \delta(x).C$  e uma lente de diâmetro igual a  $a$ , se obtenha a resposta da objetiva  $R(k_x, k_y)$ , descrita na Equação 3.2.

$$R(k_x, k_y) = \int \int_{-a/2}^{a/2} \delta(x) C e^{j(k_x x + k_y y)} dx dy \quad (3.2)$$

Assim, as frequências espaciais associadas com o coordenador espacial  $(x, y)$  podem ser expressas como o quadrado da transformada de Fourier do produto da fonte com a abertura da lente  $R^2(k_x, k_y)$ , com  $(k_x, k_y)$ . Portanto, procurando a solução de (3.2) e resolvendo a integral por partes é possível chegar a:

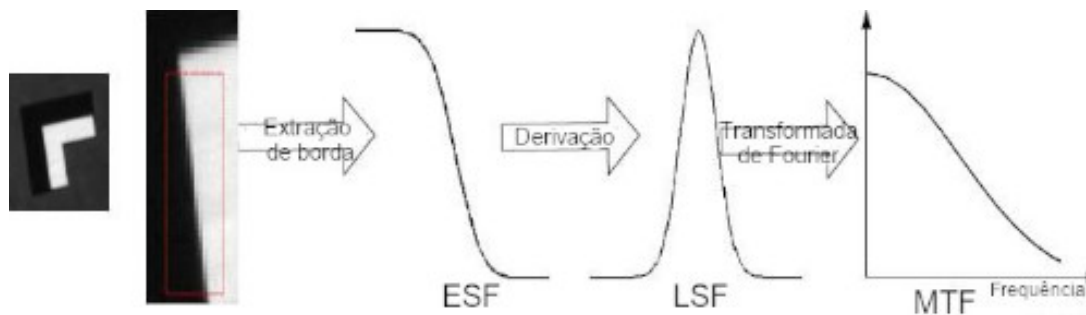
$$R^2(k_x, k_y) \propto \frac{\sin^2(ak_y)}{(ak_y)^2} \quad (3.3)$$

A Equação (3.3) corresponde à LSF. A Transformada de Fourier da LSF então fornece a MTF 1D na direção  $yy$ . Considerando que a lente possua simetria circular, a partir de tal função já é possível caracterizar toda a lente.

Uma forma popular de estimar a curva MTF para frequência espacial é chamada de método do gume de faca inclinado, no qual a curva é obtida a partir de uma região da imagem onde há uma transição de um tom muito escuro para um tom muito claro (KAWAGISHI et al., 2020).

Uma Função de Espalhamento de Borda (ESF) é calculada a partir do fio da navalha registrado, dando a resposta unidirecional do sistema de imagem a um objeto de borda, substituindo a PSF. A LSF então pode ser obtida de mesmo modo a partir da derivada da ESF e por fim a MTF é calculada a partir da Transformada de Fourier. Esse processo pode ser visualizado na Figura 22.

Figura 22 – Obtenção da MTF via método gume de faca inclinado.



Fonte: adaptado de Han et al. (2017).

Em sistemas estéreo, geralmente a MTF do sistema é resumida como um conjunto de curvas para cada sensor utilizado, ou apenas a curva do sensor de menor qualidade (NAUMOV et al., 2019). Nesta pesquisa, é considerada a resposta de todos os sensores, realizando a convolução das respostas dos sensores, com base na multiplicação das MTFs no domínio da frequência, conforme ilustra a Equação 3.4.

$$MTF_{sistema} = \mathcal{F}(LSF_1 \otimes LSF_2) = MTF_1 \times MTF_2 \quad (3.4)$$

Para qualificar um sensor, usualmente é analisado três pontos da MTF: a frequência em que ela decai 50% (na qual o contraste da imagem é degradado pela metade), a frequência em que ela decai 10% (na qual o contraste da imagem é degradado em 90%) e o valor da MTF na frequência de Nyquist, que preferencialmente deve ser maior que 0 (ALIÓ et al., 2005). Considerando estes aspectos, a MTF se torna fundamental na análise do contraste da imagem, de forma que seja analisado o impacto da resolução espacial e de variações de iluminação. Caso o contraste seja comprometido, detalhes de textura e borda das plantas podem ser prejudicados ao ponto de impossibilitar a extração correta de características. Para tais situações, pode-se utilizar a própria MTF em processos de aprimoramento da imagem, a partir da deconvolução do sinal baseado em filtro de Wiener (OH; CHOI, 2014). A caracterização da MTF então é útil para se definir a resposta espacial do sistema de visão considerando sua capacidade de detecção a partir de uma dimensão mínima em *pixels* do objeto de interesse.

O pseudocódigo 5 representa processo de obtenção da MTF do sistema. Nota-se que se utilizou imagens capturadas pelos sensores em que havia várias transições inclinadas entre quadrados brancos e pretos.

**Algorithm 5** Pseudocódigo para cálculo da MTF do sistema

**Entrada:** Imagem câmera esquerda ( $I_L$ ), Imagem câmera direita ( $I_R$ ), número de amostras a ser considerada  $n$

```

57 início
58   para cada imagem  $I_L$  e  $I_R$  faça
59     Formar  $n$  subimagens a partir de regiões em que há um gume de faca inclinado
60     para cada subimagem  $n$  faça
61       |  $ESF(n) = \text{ler\_valor\_pixels}(\text{linha horizontal centralizada})$ 
62     fim
63      $ESF = \text{média}(ESF(n))$ 
64      $ESF = \text{normalizar}(ESF)$ 
65      $LSF = \text{derivada}(ESF)$ 
66      $MTF = \text{transformada\_fourier}(LSF)$ 
        // a partir de  $I_L$  se obtém  $MTF_L$ , e de  $I_R$  obtém  $MTF_R$ 
67   fim
68    $MTF\_sistema = MTF_R \times MTF_L$ 
69 fim
Saída: MTF de cada câmera  $MTF_R$  e  $MTF_L$  e MTF final do sistema  $MTF\_sistema$ 

```

### 3.6 Datasets para validação do método

Para a validação do método, foram utilizadas nesta pesquisa *datasets* de imagens, sendo um em campo de milho e outra em campo de amendoim. O primeiro, que é denominado como *Dataset 1*, sendo composta por 34 arquivos georreferenciados de imagens adquiridas em um campo experimental da Embrapa Milho e Sorgo (Rodovia MG 424 km 45 - Sete Lagoas, MG, Brasil) de 38 ha de área, dividida em 41 parcelas espaçadas de 100,0 x 100,0 m e com área de 12 m<sup>2</sup> (4,0 x 3,0 m). As imagens possuem resolução de 600 x 600 *pixels* e apresenta em seu conteúdo plantas da espécie *Zea mays* (milho), *Euphorbia heterophylla* (leiteira, que é uma planta invasora com folhas largas) e *Cenchrus echinatus* (capim-carrapicho, planta invasora com folhas compridas), em um fundo com terra e palha, todos dentro de um quadro de madeira medindo 0,5 x 0,5 m. Exemplos de imagens deste banco de dados podem ser observadas na Figura 23 (a partir de experimentos de projetos desenvolvidos pela EMBRAPA).

O segundo, denominado como *Dataset 2*, é composto com plantas invasoras presentes na cultura do amendoim. Foi gerado no campo experimental localizado na Faculdade de Ciências Agrárias e Veterinárias da Unesp (Via de Acesso Prof. Paulo Donato Castellane - Jaboticabal, SP, Brasil), em uma área de 72 m<sup>2</sup> (5 x 14,4 m), como pode ser visto nas Figuras 24 e 25, em experimentos desenvolvidos com parceria entre a UNESP e a EMBRAPA. Para esses experimentos conduzidos com participação Dr. Augusto Guerreiro Fontoura Costa. Para esta pesquisa, a abordagem de validação considera exclusivamente as três espécies de plantas que foram semeadas, ou seja: *Arachis hypogaea* (amendoim), *Urochloa decumbens* (capim-braquiária, planta invasora com folhas estreitas) e *Mucuna aterrima* (mucuna-preta, planta invasora de folhas largas). A escolha das famílias de plantas invasoras se deu a serem as de maior interesse no

manejo do amendoim, pois elas são mais resistentes a insumos de controle. Também nasceu na área de cultivo outras plantas, como por exemplo a *Alternanthera ficoidea* (apaga-fogo, planta invasora de folhas largas), que já faziam parte do banco de sementes no solo. O banco de sementes descreve o montante de sementes presentes no solo ou nos restos vegetais, sendo que cerca de 95% da composição chega a ser de plantas invasoras (CARMONA, 1992).

Figura 23 – Dataset 1: cultura do milho.



Fonte: Experimento de projeto conduzido por P. E. Cruvinel e D. Karam na Embrapa Milho e Sorgo.

Foram plantadas fileiras lineares amendoim de 5 m de comprimento contendo cerca de 12 plantas por metro, espaçadas entre si a 0,9 m. O amendoim foi plantado por volta do dia 10 de Outubro, as plantas invasoras foram semeadas no dia 26 de Outubro e a aquisição das imagens foi realizada no dia 7 de Dezembro. A aquisição foi realizada da seguinte maneira:

- Selecionou as fileiras em que tivessem semeadas mais de uma espécie de planta;
- Para cada fileira, foram tomadas imagens que varressem todo seu comprimento;

- Antes de capturar as imagens, nivelou o aparelho com auxílio de um nível de bolha;
- Para cada bloco, foram capturadas 5 imagens por câmera, com o intuito de poder descartar imagens de má qualidade (desfocadas, erro de captura ou com muito ruído);
- Para cada bloco, os dados de georreferenciamento foram anotados com auxílio de um GPS;
- Repetiu as operações por bloco e por fileira, até varrer toda a área.

Figura 24 – Campo experimental Jaboticabal: vista geral.



Fonte: Experimento de projeto conduzido por A. G. F. Costa e P. L. C. A. Alves, campus UNESP Jaboticabal.

Essas bases foram escolhidas e desenvolvidas devido ao alto interesse na agricultura que ambos os produtos de cultivo têm. O milho é usado na alimentação humana e animal, na produção de espessantes e óleo e na produção de bioetanol (ECKERT et al., 2018). Já o amendoim tem uso mais focado na alimentação e na produção de óleo (PRASAD et al., 2010).

Todos os *datasets* gerados são de imagens RGB, no espectro do visível. Para a análise do algoritmo de segmentação, 11 imagens (um terço do total das imagens) de valor-verdade da segmentação do *Dataset 1* foram geradas no âmbito desta pesquisa, demarcando todas as plantas presentes na imagem com a cor branca e o resto com a cor preta. Para a análise do algoritmo de classificação, ambos os *datasets* foram rotulados manualmente a partir da morfologia das plantas invasoras para análise.

Figura 25 – Campo experimental Jaboticabal: vista lateral.



### 3.7 Visão estéreo computacional

Sistemas de visão estéreo são sistemas em que se pode criar informação de profundidade a partir do uso de duas câmeras, baseado no próprio sistema humano. Assim, é possível encontrar a distância dos objetos retratados pelas câmeras em relação ao plano de referência em que as câmeras estejam posicionadas, a partir das diferenças entre as duas imagens registradas (NEVATIA, 1976). Pode-se caracterizar o sistema estéreo como passivo, caso as câmeras estejam estacionárias; ou ativo, quando as câmeras se movem em relação ao objeto de interesse, gerando várias imagens de ângulos diferentes (OKURA, 2022; CHIANG; LIN, 2022). Nesta pesquisa é considerado um sistema com arranjo passivo que viabiliza adquirir imagens e informações de profundidade de objetos presentes em ambiente imageado, sem entretanto necessariamente considerar a cena de forma estática.

#### 3.7.1 Geometria e princípios do sistema estéreo

Para o sistema estéreo, é considerado o uso das câmeras coplanares e alinhadas entre si, como apresentado na Figura 26. As variáveis definidas pela câmera são a linha de base  $b$  e a distância focal  $f$ . Já  $P(X, Y, Z)$  representa um ponto que será registrado pelas câmeras e  $u_L = (X_L, Y_L)$  e  $u_R = (X_R, Y_R)$  as projeções deste ponto em cada imagem. A partir dos

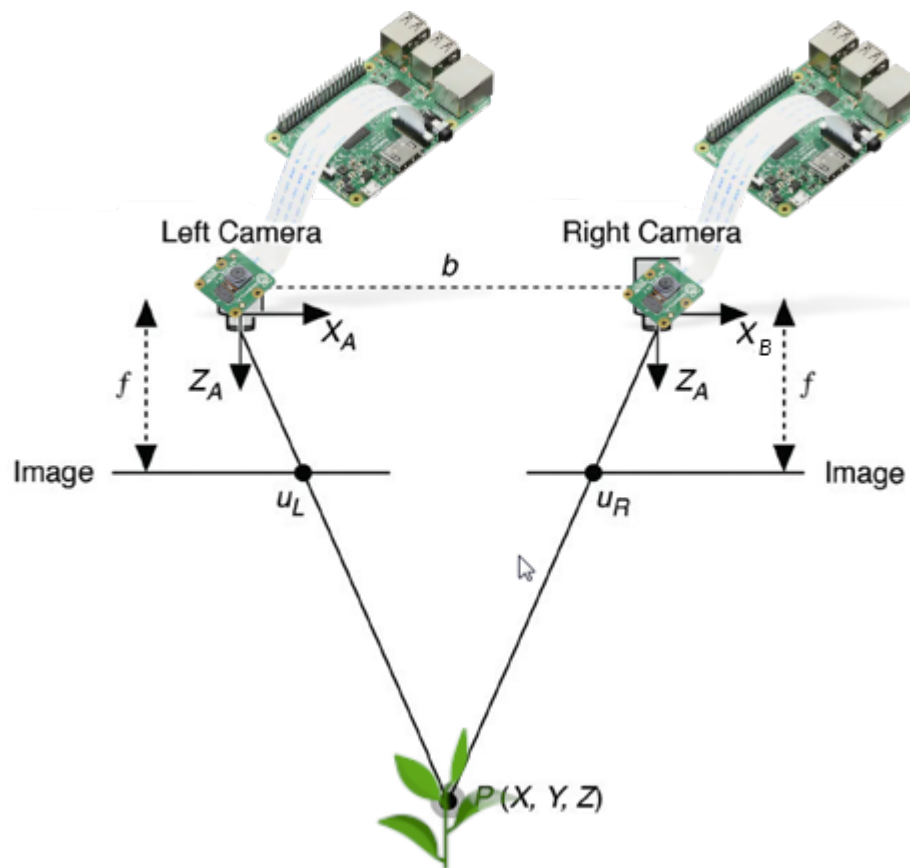
conceitos de geometria e semelhança de triângulos, é possível se obter (SCHARSTEIN, 2003):

$$\frac{X_L}{f} = \frac{X}{Z} \Rightarrow X_L = \frac{Xf}{Z} \quad (3.5)$$

$$\frac{-X_R}{f} = \frac{b - X}{Z} \Rightarrow X_R = \frac{(X - b)f}{Z} \quad (3.6)$$

$$Y_L = Y_R = \frac{fY}{Z} \quad (3.7)$$

Figura 26 – Modelo de Visão Estéreo.



Considerando que a diferença entre as coordenadas  $x$  do ponto em cada imagem é igual a  $d = X_L - X_R$ , em que  $d$  é chamado de disparidade, tem-se que as Equações (3.5) e (3.6) podem ser reescritas como:

$$X = \frac{bX_L}{d} \quad (3.8)$$

$$Y = \frac{bY_L}{d} \quad (3.9)$$



$$Z = \frac{bf}{d} \quad (3.10)$$

Percebe-se então que a profundidade do ponto ao plano das câmeras  $Z$  pode ser encontrada a partir de três parâmetros. A distância focal  $f$  e a linha de base  $b$  são definidas previamente e são mantidas constantes. Assim, para encontrar a profundidade de objetos na imagem basta encontrar a disparidade de todos os *pixels* presentes, chamado de mapa de disparidades.

Também é importante observar a distorção que variações no mapa de disparidades pode causar na estimação de profundidade, isto é, verificar a acurácia da medida obtida. Assim, para uma variação na profundidade tem que:

$$\Delta Z = Z - \frac{bf}{d + \Delta d} = Z - \frac{bf}{\frac{bf}{Z} + \Delta d} = \frac{Z\Delta d}{\frac{bf}{Z} + \Delta d} = \frac{Z^2\Delta d}{bf + Z\Delta d} \approx \frac{Z^2\Delta d}{bf} \quad (3.11)$$

Percebe-se então que para diminuir as distorções,  $Z$  não pode ser muito grande enquanto que  $b$  e  $f$  não podem ser muito pequenos. Como  $f$  é uma característica própria da câmera, planeja-se então o sistema de modo que  $b$  possua uma distância que permite que no  $Z$  desejado haja poucas distorções.

Logo, o problema final se torna em encontrar o mapa de disparidade desejado. Para se encontrar o mapa, deve-se associar corretamente cada *pixel* de uma imagem com um *pixel* na outra, caso ele exista. Como a configuração do sistema garante que a coordenada  $Y$  entre as imagens é a mesma (garantido pela sincronização de ambas as câmeras de modo que a Equação (3.7) seja válida), dado um *pixel*  $(x_i, y_i)$  na imagem 1, deve-se procurar o par correspondente  $(x_j, y_i)$  na imagem 2 para obter a disparidade local, dada por  $|x_j - x_i|$ . Nesta pesquisa, foi adotado como imagem de referência a imagem registrada pela câmera direita, enquanto que a imagem de trabalho foi a da esquerda.

Deseja-se então analisar a imagem da direita e encontrar os objetos presentes nela na imagem da esquerda, formando pares de *pixels* correspondentes pois eles são representações do mesmo ponto do objeto físico. As dificuldades em se resolver o problema de correspondência podem apresentar por de forma de ambiguidades (mais de um par correspondente) geradas a partir de regiões com características semelhantes como textura e intensidade. Além disso, o método considera que a intensidade luminosa recebida por ambas as câmeras de um determinado objeto será igual apesar da variação de ângulo de observação (isto é, as superfícies analisadas seguem a lei de emissão de Lambert) e que as câmeras não apresentam diferenças intrínsecas entre si (como o ganho e viés presente em cada receptor). Outro problema é a presença de pontos ocultos entre as câmeras, o que faz com que um ponto em determinada imagem não terá correspondência na outra pois ele é visível apenas pelo ângulo da primeira. Este efeito é chamado de oclusão.

Os primeiros algoritmos que trabalharam com visão estéreo eram baseados na extração de características da imagem. Tais métodos eram bons especificamente para objetos com bordas bem definidas, já que se filtrava as imagens para obtê-las e desta forma identificar a borda correspondente, diminuindo significativamente o número de ambiguidades. Porém, eles não eram capazes de gerar um bom mapa de disparidade, já que o mapa só tinha valores em certos pontos (nas bordas) e o resto do mapa era definido apenas pelos pontos próximos, utilizando então aproximações e suposições para completá-lo. Apesar disso, em casos em que se tenha grandes variações na iluminação da região de interesse, em que o contorno se torna a única característica estável, estes procedimentos apresentam melhores resultados (SZELISKI, 2011).

Nesta pesquisa, é utilizada uma técnica baseada em área, de forma a evitar ambiguidades, conforme técnica apresentado por Scharstein (2003). Neste contexto, foram considerados os seguintes passos:

1. Pré-processamento da imagem;
2. Para cada disparidade em consideração, calcula-se um custo de correspondência para cada pixel;
3. Considera-se a agregação de suporte espacial (apenas no método não-global);
4. Varre-se todos os valores de disparidade possíveis e se encontra o melhor par correspondente de *pixels*;
5. Estima-se a disparidade sub-pixel.

### 3.7.2 Métodos de correspondência estéreo

Os métodos de correspondência tradicionais utilizam geralmente características de baixo nível da vizinhança do *pixel* para calcular a dissimilaridade. Entre os descritores locais mais comuns, se tem a *absolute difference* (AD), CENSUS, BRIEF, *normalized cross-correlation* (NCC), ou as suas combinações, como o AD-CENSUS (ZHOU et al., 2020). A partir destes métodos, são geradas funções de custo de pareamento (LAZAROS et al., 2008), em que as mais utilizadas são apresentadas nas Equações 3.12, 3.13 e 3.14, em que  $I_l$  e  $I_r$  são a imagem esquerda e direita, respectivamente,  $d$  é a disparidade,  $x$  e  $y$  são as coordenadas da imagem e  $W$  é a região de suporte de vizinhança do *pixel*, que pode ser quadrada ou retangular, com tamanho fixo ou adaptável.

$$SAD(x, y, d) = \sum_{x,y \in W} |I_l(x, y) - I_r(x, y - d)| \quad (3.12)$$

$$SSD(x, y, d) = \sum_{x,y \in W} (I_l(x, y) - I_r(x, y - d))^2 \quad (3.13)$$

$$NNC(x, y, d) = \frac{\sum_{x,y \in W} I_l(x, y) I_r(x, y - d)}{\sqrt{\sum_{x,y \in W} I_l^2(x, y) \sum_{x,y \in W} I_r^2(x, y - d)}} \quad (3.14)$$

O descritor CENSUS também é muito utilizado, em que é gerado um vetor  $C(p)$  para cada *pixel*  $p$  comparando com a sua intensidade  $I_p$  com a dos *pixels* vizinhos  $I_q$ , em que os *pixels*  $q$  são limitados pela janela  $W$ , como pode ser visto na Equação 3.15 e 3.16. Assim, a disparidade de cada ponto é obtida a partir da disparidade  $d$  em que a distância Hamming entre os vetores  $C_l$  e  $C_r$  é a menor ao comparar com *pixels* na mesma linha na imagem esquerda e direita, como descrito na Equação 3.17. É possível realizar alterações no CENSUS, alterando o formato da janela  $W$  ou modificando como a função  $\xi(I_p, I_q)$  é construída, podendo por exemplo definir uma ordem entre os *pixels* dentro de  $W$  e criar o vetor comparando as suas intensidades sucessivamente (LEE et al., 2016)

$$C(p) = \otimes_{q \in W} \xi(I_p, I_q) \quad (3.15)$$

$$\xi(I_p, I_q) = \begin{cases} 1, & \text{se } I_p < I_q, \\ 0, & \text{c.c.} \end{cases} \quad (3.16)$$

$$d(p) = \operatorname{argmin} \sum \operatorname{Hamming}[C_l(p), C_r(p + d)] \quad (3.17)$$

O mapa de disparidade pode ser esparso ou denso. Métodos esparsos consideram apenas as bordas dos objetos e calculam a disparidade apenas destes pontos. Já os métodos densos geram um mapa de disparidade para toda a imagem, sendo separados entre métodos locais, baseados em áreas e métodos globais, baseado em funções de energia (LAZAROS et al., 2008).

Os algoritmos de pareamento estéreo podem ser separados entre métodos locais, globais e semi-globais. Nos métodos locais, que restringem a área de análise com o intuito de diminuir o custo computacional, a disparidade com menor custo de pareamento é selecionada. Esta estratégia é conhecida como *winner takes all* (WTA), descrita na Equação 3.18, porém apresenta baixa qualidade. É possível também realizar tal pareamento de forma iterativa, pois é comum a disparidade de um *pixel* depender da disparidades dos *pixels* da sua vizinhança (LAZAROS et al., 2008). Já os métodos globais definem uma função global de energia de pareamento que deve ser minimizada, analisando a imagem como um todo, apresentando então tanto eficácia como tempo de processamento maior. Alguns métodos semi-globais tentam compensar as falhas de ambos os outros a partir de suas junções (ZHOU et al., 2020).

$$D(x, y) = \operatorname{argmin} SAD(x, y, d) \quad (3.18)$$

De qualquer modo, será realizado uma medida de similaridade para comparar os *pixels* e determinar um grau de correspondência. O custo de pareamento mais comum inclui o uso da soma quadrada da diferença das intensidade (SSD - *sums of squared intensity differences*) e da diferença absoluta das intensidades (SAD - *absolute intensity differences*). Outros métodos mais robustos podem ser utilizados, como quadráticas truncadas ou Gaussianas contaminadas. Há também métodos que são invariantes ao ganho e ao viés das câmeras, à escala e à iluminação, como por exemplo a Transformada Census (LEE et al., 2016).

Em métodos globais de pareamento, o objetivo é encontrar uma solução  $d$  que minimiza a energia global  $E(d)$  dada pelas Equações (3.19) e (3.20), em que  $E_d(d)$  mede como a função de disparidade  $d$  se relaciona com o par de imagens de entrada, com  $L$  sendo o valor inicial de custo de pareamento e o  $E_s(d)$  as suavizações feitas pelo algoritmo.

$$E(d) = E_d(d) + \lambda * E_s(d) \quad (3.19)$$

$$E_d(d) = \sum_{(x,y)} L(x, y, d(x, y)) \quad (3.20)$$

Enquanto isso, os métodos locais utilizam as medidas de custo apenas em uma região de suporte no espaço  $L(x, y, d)$ , descrito na Equação (3.21). A agregação com uma região de suporte é performada através de uma convolução e é importante que o tamanho dela seja grande suficiente para conter uma variação de textura que evite descontinuidades no mapa de disparidade.

$$L(x, y, d) = w(x, y, d) \otimes L_0(x, y, d) \quad (3.21)$$

Adicionalmente, o cálculo da disparidade sub-pixel serve para a estimação de disparidade nos pontos em que ela não foi encontrada a partir dos passos anteriores, como nos pontos de oclusão. Ela pode ser computada por variados métodos como gradiente iterativo ou interpolação em uma curva realizada em cima de valores discretos de disparidade. Assim, há um aumento na resolução da visão estéreo, ainda mais se for garantida uma variação gradual de valores de disparidade. Outro tipo de pós-processamento que pode ser aplicado ao mapa de disparidades é a associação de níveis de confiança para cada estimação de profundidade por pixel.

A avaliação da eficiência dos algoritmos de visão estéreo pode ser obtida a partir do cálculo do erro relativo absoluto, do erro relativo quadrático, do *root mean square error* (RMSE), RMSE logaritmo ou do erro  $\delta < 1,25$ , apresentados nas equações abaixo, em que  $N_d$  é o número total de *pixels* com valores calculados de disparidade,  $Disp_{pred}$  é a disparidade prevista e  $Disp_{gd}$  a disparidade real. O erro  $\delta < 1,25$  representa a porcentagem de *pixels* que satisfazem a Equação 3.26 com valor menor que 1,25 (ZHOU et al., 2020). Para todas elas se é necessário

ter o valor-verdade do mapa.

$$\text{RMSE} = \sqrt{\frac{1}{N_d} \sum_{i \in N_d} |Disp_{pred} - Disp_{gd}|^2} \quad (3.22)$$

$$\text{RMSE logaritmo} = \sqrt{\frac{1}{N_d} \sum_{i \in N_d} |\log Disp_{pred} - \log Disp_{gd}|^2} \quad (3.23)$$

$$\text{Erro relativo absoluto} = \frac{1}{N_d} \sum_{i \in N_d} \frac{|Disp_{pred} - Disp_{gd}|}{Disp_{gd}} \quad (3.24)$$

$$\text{Erro relativo quadrático} = \frac{1}{N_d} \sum_{i \in N_d} \frac{|Disp_{pred} - Disp_{gd}|^2}{Disp_{gd}} \quad (3.25)$$

$$\delta = \max \left( \frac{Disp_{pred}}{Disp_{gd}}, \frac{Disp_{gd}}{Disp_{pred}} \right) \quad (3.26)$$

### 3.7.3 Cálculo de volume a partir do mapa de disparidade

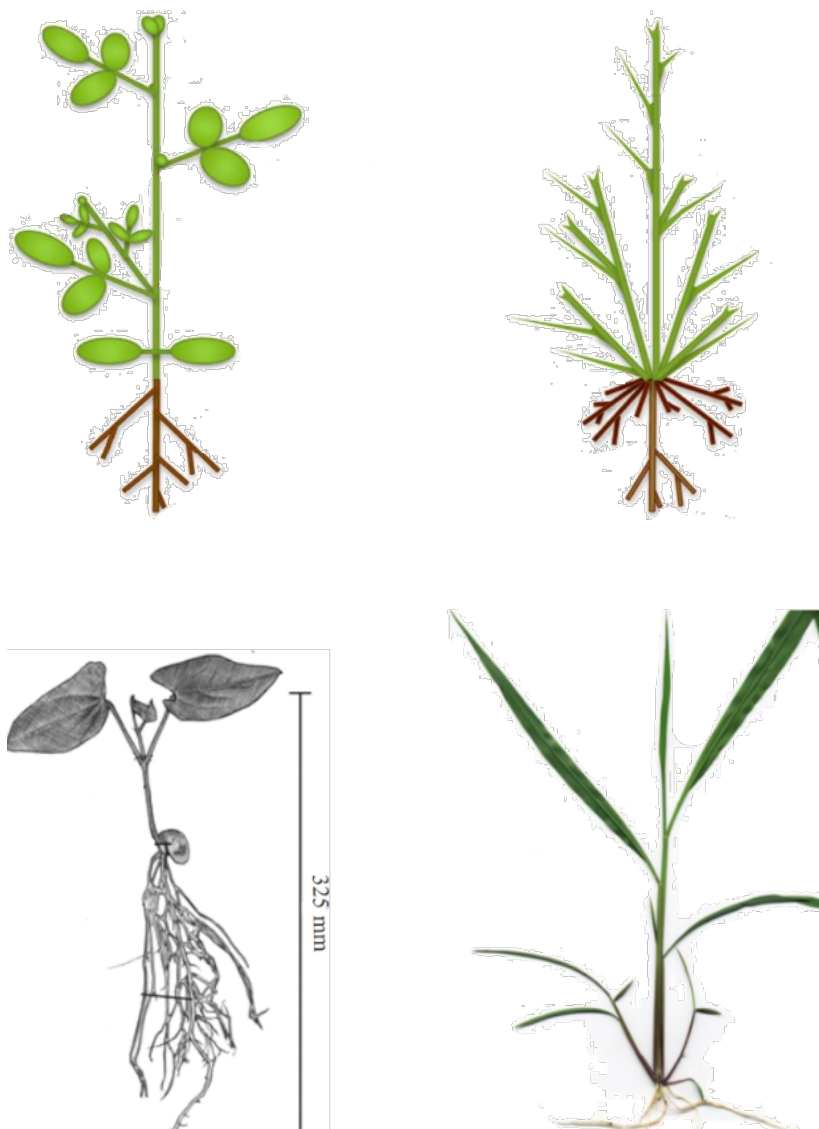
O mapa de disparidades gerado pelo algoritmo de visão estéreo informa uma medida inteira  $d$  em *pixels* da distância do objeto a câmera, definidos *pixel a pixel* pela imagem. Com o valor fixo conhecido da distância de base  $b$  e a distância focal  $f$  da lente, estes valores podem então ser convertidos para metros a partir da Equação 3.10, obtendo-se  $Z$ . Conforme apresentado por Lin et al. (2011), é possível então realizar o cálculo do volume dos objetos presentes na imagem tendo-se os valores reais das dimensões dos *pixels* da imagem. Como os sensores do sistema desenvolvido se encontram paralelo ao chão e acima das plantas (tanto da cultura quanto das invasoras), a altura de cada *pixel* então pode ser obtida a partir da subtração da altura fixa do aparelho por  $Z$ .

As plantas invasoras analisadas possuem estruturas de plantas dicotiledôneas e monocotiledôneas, conforme apresentadas na Figura 27, em que cada folha e elementos conectados crescem de forma radial em torno de um eixo. Durante o desenvolvimento inicial da plântula de sentido vertical, as folhas tendem a possuir maior área a medida que ela esteja mais longe do solo. Quanto maior a planta, maior a densidade de folhas. Assim, tendo em consideração tal arquitetura, define-se o volume de uma planta como uma medida da sua densidade foliar a partir de um cone, em que o pico se encontra no solo e a base no topo da planta. O volume então pode ser calculado como o terço do produto da área da base versus a altura.

A informação de volume da planta é importante para o controle de plantas invasoras, na tomada de decisão do volume de calda de agrotóxicos que deve ser utilizada para o manejo da região afetada. Volume de calda, também chamado de volume de aplicação ou taxa de aplicação, é baseado na deposição visual sobre o alvo em campo, em que na prática é comum fixar em um valor único para aplicação em toda a área de cultivo. Um parâmetro utilizado na definição do

volume de calda é o índice de área foliar, que é a área somada de todas as folhas de um espaço por unidade de solo ocupada. A partir deste índice, pode-se por exemplo definir o volume de calda a ser pulverizado, a concentração de agrotóxico presente na calda e a composição dele (CHECHI et al., 2018; LIAO et al., 2020). Para cada espécie de planta invasora e para a quantidade de presença dela na área de cultivo, há um protocolo distinto a ser seguido. Outros parâmetros que podem ser utilizados na definição do volume de calda é a densidade de folhas e a biomassa, que necessita de pesagem de material para ser obtida (TACKENBERG et al., 2016).

Figura 27 – Estrutura típica de plantas dicotiledôneas (como a mucuna) e monocotiledôneas (como o capim).



Fonte: Adaptado de Abud et al. (2009) e Pereira et al. (2016).

A sobreposição de folhas também é um fator importante na definição de tal volume de aplicação, já que as folhas mais altas acabam recebendo mais produto enquanto as plantas mais

baixas recebem uma quantidade menor de insumo. Tal variação segue uma proporção não-linear, já que uma pulverização com maior volume de calda acaba depositando maior percentual de ingrediente ativo nas folhas mais altas e em menor percentual nas folhas mais baixas (COSTA et al., 2019). Assim, a informação da altura da planta também é essencial na identificação de tais cenários. O conjunto de técnicas que consideram vários parâmetros para a tomada de decisão do volume de calda e o uso de sistemas de pulverização inteligentes, capazes de alterar o volume a partir de sensores em tempo real, concretizam uma otimização do controle da aplicação de insumos no campo.

### 3.7.4 Pseudocódigo do algoritmo estéreo

Para validação do método de visão estéreo, foram considerados e testados dois algoritmos de correspondência, ambos apresentados no pseudocódigo 6. O primeiro, mais simples, é um algoritmo de pareamento local de blocos, em que se tem como parâmetros o tamanho da região de suporte e o intervalo máximo de disparidade de busca (que depende da proximidade dos objetos a câmera). Um bloco maior implica em um mapa de disparidade mais suave, porém mais impreciso. Para esse algoritmo, foi necessário a aplicação de uma curva para que o mapa de disparidade fosse contínuo. O segundo método é um algoritmo de pareamento semi-global do tipo CENSUS, desenvolvido por Hirschmuller (2008). De forma resumida, o algoritmo lê as imagens, aplica um filtro de suavização Gaussiana, computa o CENSUS nas duas direções da imagem, realiza a etapa de agregação de custo em quatro direções e por fim selecionada a melhor disparidade, pós-processando o mapa com um filtro de mediana.

---

#### Algorithm 6 Pseudocódigo dos algoritmos estéreo

---

**Entrada:** Imagem direita  $I_{right}$  e esquerda  $I_{left}$

70 **início**

```

71     // Algoritmo 1:
72     Ste = criar_objeto_estéreo(numero_disparidade,tamanho_bloco)
73     Mapad1 = calcular_mapa(Ste, $I_{right}$ , $I_{left}$ )
74     Mapad1 = preencher_mapa(Mapad1,curva)
75     // Algoritmo 2:
76      $I_{w1}$  = aplicar_filtro( $I_{right}$ ,Gauss)
77      $I_{w2}$  = aplicar_filtro( $I_{left}$ ,Gauss)
78     Cs[] = calcular_census( $I_{w1}$ ,  $I_{w2}$ )
79     Mapad2 = menor_custo_agregação(Cs)
80     Mapad2 = aplicar_filtro(Mapad2,mediana)

```

79 **fim**

**Saída:** Mapas de disparidade  $Mapa_{d1}$  e  $Mapa_{d2}$

---

### 3.8 Calibração das câmeras

Para calibração da câmera é considerada o conjunto de parâmetros internos para validação do método. Toda câmera pode ser descrita a partir de parâmetros intrínsecos e extrínsecos, que contribuem em como a imagem é formada a partir da cena no mundo real. Os parâmetros intrínsecos são os relativos aos vieses internos, devido ao sensor e seu formato, à lente e suas distorções e a outras características envolvidas na fabricação da câmera, enquanto os parâmetros extrínsecos se referem a posição da câmera no espaço em relação ao mundo (HARTLEY; ZISSERMAN, 2004). Já os parâmetros extrínsecos podem ser simplificados por uma matriz de rotação  $\mathbf{R}_m$  e uma de translação  $\mathbf{T}_m$ .

A distância focal  $f$  é um parâmetro intrínseco, pois é a distância entre o centro da câmera para o plano da imagem, i.e., das lentes para o sensor. Muitas câmeras utilizam um sensor *Charge-Coupled Device* (CCD), sensor semiconductor formado por um circuito integrado que contém uma matriz de capacitores acoplados, capaz de gerar estímulos elétricos a partir da luz recebida. Como o *pixel* em um sensor deste tipo pode não ser perfeitamente quadrado, há a possibilidade de haver uma pequena distorção no número de *pixels* por unidade de comprimento. Desta forma, a distância focal das lentes das câmeras será diferente em cada sentido, resultando então nas variáveis  $f_u$  e  $f_v$ , sendo que a razão de aspecto é definida por  $f_v/f_u$ .

Outro parâmetro da câmera é o centro óptico, definido pelas coordenadas  $(u_0, v_0)$ , que representa um fator de translação da origem da imagem em relação ao centro do sensor, de tal forma que a origem da imagem esteja corretamente na borda superior esquerda dela. Há também o coeficiente *skew* ( $\tau$ ) que corrige a imagem nos casos em que o sensor CCD não possua orientação perpendicular entre os eixos de comprimento e largura. Como para a maioria dos sensores essa situação é rara, é comum assumir que  $\tau = 0$ .

Por fim, devido a natureza curva das lentes, os últimos parâmetros intrínsecos a serem considerados ao modelar uma câmera são os coeficientes de distorção (WANG et al., 2018). Os coeficientes de distorção tangencial são definidos por duas variáveis,  $k_{p1}$  e  $k_{p2}$ , enquanto os coeficientes de distorção radial de segunda, quarta e sexta ordem são representados respectivamente por  $k_{q1}$ ,  $k_{q2}$  e  $k_{q3}$ .

Portanto, o processo de captura de uma imagem digital por um sensor pode ser descrito de forma simplificada a partir da Equação 3.27, baseado na projeção do espaço no sensor, em que  $u_d$  e  $v_d$  representa as coordenadas de um ponto na imagem sem correção de distorção,  $s$  o fator de escala ou resolução e  $X_w$ ,  $Y_w$  e  $Z_w$  as coordenadas de um ponto no mundo (YANG et



al., 2018).

$$s \begin{bmatrix} u_d \\ v_d \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & \tau & u_0 & 0 \\ 0 & f_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}_m & \mathbf{T}_m \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (3.27)$$

Para se encontrar a coordenadas sem distorções  $(u, v)$  da imagem, corrigindo a projeção, deve-se solucionar o sistema de equações 3.28.

$$\begin{cases} x_d = \frac{u_d - u_0}{f_u} \\ y_d = \frac{v_d - v_0}{f_v} \\ r^2 = x_n^2 + y_n^2 \\ x_d = x_n(1 + k_{q1}r^2 + k_{q2}r^4 + k_{q3}r^6) + 2k_{p1}x_ny_n + k_{p2}(r^2 + 2x_n^2) \\ y_d = y_n(1 + k_{q1}r^2 + k_{q2}r^4 + k_{q3}r^6) + 2k_{p2}x_ny_n + k_{p1}(r^2 + 2y_n^2) \\ u = f_u x_n + u_0 \\ v = f_v y_n + v_0 \end{cases} \quad (3.28)$$

Portanto, ao se caracterizar os parâmetros intrínsecos de qualquer câmera, pode-se sumarizar a informação partir de duas matrizes, a matriz da câmera  $\mathbf{M}_{\text{cam}}$  e a matriz de coeficientes de distorção  $\mathbf{K}_{\text{cam}}$ , como pode ser visto nas Equações 3.29 e 3.30.

$$\mathbf{M}_{\text{cam}} = \begin{bmatrix} f_u & \tau & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.29)$$

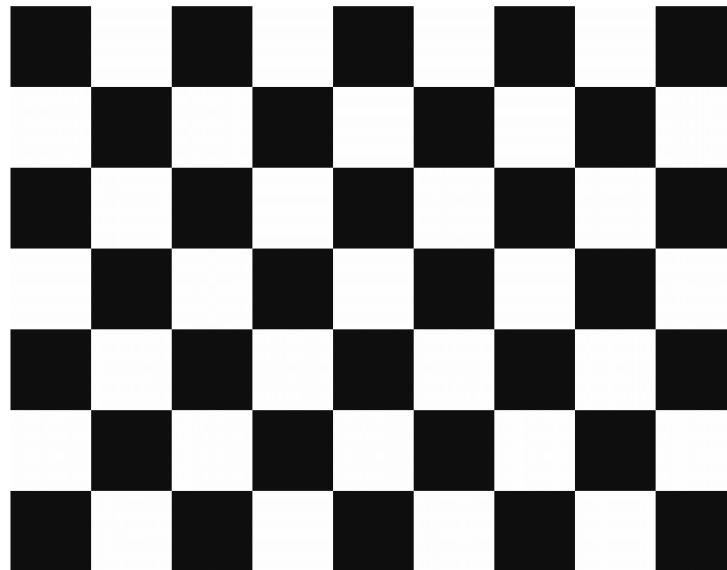
$$\mathbf{K}_{\text{cam}} = \begin{bmatrix} k_{q1} & k_{q2} & k_{p1} & k_{p2} & k_{q3} \end{bmatrix} \quad (3.30)$$

Denomina-se o processo de obter tais parâmetros como calibração da câmera. Os métodos de calibração dependem do modelo utilizado para aproximar o comportamento real da câmera. Os modelos mais utilizados são os lineares de Hall e Faugeras–Toscani, desenvolvidos respectivamente em 1982 e 1986 e os não lineares de Tsai e Weng, implementados em 1987 e 1992, que apresentam no geral menos erros (SALVI et al., 2002).

Da Equação 3.27, tem-se que a matriz de projeção entre o mundo real e o universo da imagem terá dimensão  $3 \times 4$ , o que resulta em 11 parâmetros que devem ser obtidos. Comumente, para a calibração de sistemas estéreo e obtenção dos valores da matriz, utiliza-se imagens de tabuleiros de xadrez (Figura 28) com dimensões conhecidas (número de quadrados e tamanho do lado deles no mundo real), em que os pontos de calibração são os vértices internos dos

quadrados do tabuleiro. Como cada ponto corresponde a duas equações (uma na coordenada  $x$  e outra em  $y$ ), são necessários 5 pontos e meio para a calibração do sistema, porém experimentos demonstraram que 5 vezes mais pontos que o necessário apresentaram resultados melhores (HARTLEY; ZISSERMAN, 2004; WANG et al., 2010). Um número maior de imagens para a calibração também diminuem o erro total de localização em mm, sendo que 13 imagens em que pelo menos 30% delas eram compostas pelo tabuleiro, em orientações aleatórias, já apresentaram bons resultados (YANG et al., 2018). Outros métodos de calibração além do tabuleiro também podem ser utilizados, como a calibração por laser (YANG et al., 2020) ou a partir de objetos esféricos (SUN et al., 2015).

Figura 28 – Tabuleiro de xadrez para calibração.



Também se deve realizar a calibração de todo o sistema de visão estéreo, onde são obtidas o fator de rotação  $\mathbf{R}_{\text{stereo}}$  e o de translação  $\mathbf{T}_{\text{stereo}}$  entre a imagem esquerda e direita (WANG et al., 2018; YANG et al., 2020). Para tal cálculo, utiliza-se os parâmetros da câmera calculados anteriormente e as imagens de tabuleiro capturadas simultaneamente. Ao processo de corrigir a orientação das imagens estéreo é denominado de retificação. Observa-se que, ao contrário da matriz da câmera e dos coeficientes de distorção que dependem apenas da câmera, as matrizes  $\mathbf{R}_{\text{stereo}}$  e  $\mathbf{T}_{\text{stereo}}$  devem ser recalculadas se quaisquer configurações do sistema estéreo mudarem, como, por exemplo, a distância da linha de base.

### 3.9 Espaço de cor

Uma imagem digital pode ser representada como uma matriz de intensidades  $F(x, y, z)$ , em que  $x$  e  $y$  indicam a posição do *pixel* e o  $z$  o canal de registro. Em imagens na escala de cinza,  $z$  só assume o valor unitário, já que há apenas um canal. A câmera digital que foi utilizada

registra imagens no espectro do visível em três canais, sendo que cada um é formado a partir dos estímulos que os fótons que chegam até o sensor geram em cada receptor. Cada *pixel* então é composto de três receptores, sendo que cada um tem sensibilidade calibrada para três diferentes frequências do espectro visível da luz, correspondentes a cor vermelho, a verde e a azul. Este espaço de cor é denominado como o espaço de cor RGB. O valor de  $F(x, y, z)$  então é obtido a partir da intensidade do estímulo em cada receptor da câmera, em  $F(x, y, z) = 0$  quando o estímulo é mínimo e  $F(x, y, z) = 1$  quando máximo (BUSIN et al., 2008).

O espaço RGB é classificado como um espaço de cor primária, que se baseia da teoria tricromática, que assume que é possível obter qualquer cor pela mistura apropriada de quantidades definidas das três cores primárias. Os métodos de Visão Computacional então podem ser aplicados em cada canal, mas é desejado lidar com toda a informação de forma conjunta, aproveitando então de forma integral todo o sensor, preferencialmente sem ponderar de forma excessiva um canal em detrimento de outro. Comumente, operações com os três canais RGB são realizadas com o intuito de gerar outros espaços de cor que integrem melhor a informação apresentada, como o já citado espaço de cor de escala de cinza.

Há também espaços de cores de eixos independentes, que são resultados de métodos estatísticos com o objeto de gerar canais com a mínima correlação possível entre eles; e há espaços de luminância-crominância, em que um dos componentes a luminância, i.e., a densidade da intensidade de uma luz refletida no sensor e os outros dois a crominância da cor. A última classificação de espaços de cor é a de percepção, que tem como objetivo quantificar a percepção subjetiva de cor da visão humana a partir de três elementos, a intensidade, a matiz e a saturação. Analisando a literatura na área, foi escolhida trabalhar no espaço de cor HSV, que é um espaço de cor de percepção, pois é um dos mais usados e que apresentam respostas melhor do que outros espaços.

O espaço de cor HSV (*Hue, Saturation, Value*) possui três componentes, a intensidade  $H$ , a saturação  $S$  e o valor  $V$ .  $H$  representa a cor verdadeira, relacionada a frequência de onda da luz.  $S$  representa a pureza da cor, o quanto a cor está diluída pelo branco, em que valores baixos produz uma cor mais acinzentada e valores altos uma cor mais profunda. Por fim,  $V$  denota o brilho, o quão longe do preto a cor está, em que valores próximos de zero correspondente a uma cor mais escurecida (CHERNOV et al., 2015). Esse espaço de cor é possível ser descrito como um cilindro, em que  $H$  é o ângulo,  $S$  o raio e  $V$  a altura, como pode ser visto na Figura 29, porém há representações como um cone hexagonal.

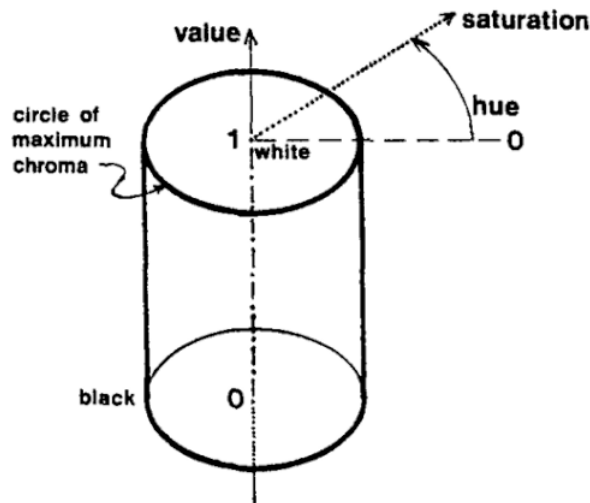
A conversão da cor RGB para o HSV, considerando uma variação de todos os parâmetros entre 0 e 1, pode ser realizada, *pixel a pixel*, a partir dos seguintes passos:

1. Encontrar o valor máximo  $M_x = \max(R, G, B)$  e mínimo  $m_n = \min(R, G, B)$ ;
2. Definir o valor  $V$  como igual a  $M_x$ ;

3. Calcular a diferença  $d_{Mm} = M_x - m_n$ ;
4. Se  $d_{Mn} = 0$ ,  $S$  é igual a 0 e  $H$  é indeterminado;
5. Defina o valor de  $S$  com  $d_{Mn}/M_x$ ;
6. Defina o valor de  $H$  a partir das equações:

$$H = \frac{1}{6} \times \begin{cases} \frac{G-B}{d_{Mm}}, & \text{se } M_x = R \text{ e } m_n = B \\ 1 + \frac{R-B}{d_{Mm}}, & \text{se } M_x = G \text{ e } m_n = B \\ 2 + \frac{B-R}{d_{Mm}}, & \text{se } M_x = G \text{ e } m_n = R \\ 3 + \frac{G-R}{d_{Mm}}, & \text{se } M_x = B \text{ e } m_n = R \\ 4 + \frac{R-G}{d_{Mm}}, & \text{se } M_x = B \text{ e } m_n = G \\ 5 + \frac{B-G}{d_{Mm}}, & \text{se } M_x = R \text{ e } m_n = G \end{cases} \quad (3.31)$$

Figura 29 – Espaço de cor HSV.



Fonte: [Joblove e Greenberg \(1978\)](#).

Para realizar a conversão de volta do espaço HSV para o RGB, basta seguir as seguintes etapas:

1. Se  $S$  for igual a 0, atribua o valor  $V$  para R, G e B;
2. Multiplique  $H$  por 6, obtendo  $H'$ ;
3. Determine o índice de setor  $Index$  da intensidade arredondando para baixo  $H'$  de tal forma que se tenha um inteiro;

4. Determine a fração  $F_{hue}$  pela subtração de  $H'$  por  $Index$ ;
5. Calcule o valores  $M$ ,  $N$  e  $K$  por:

$$\begin{cases} P = V \times (1 - S) \\ Q = V \times (1 - S \times F_{hue}) \\ K = V \times (1 - S \times (1 - F_{hue})) \end{cases} \quad (3.32)$$

6. Defina o valor do RGB pela equação:

$$[R \quad G \quad B] = \begin{cases} [V \quad K \quad P], & \text{se } Index = 0 \\ [Q \quad V \quad P], & \text{se } Index = 1 \\ [P \quad V \quad K], & \text{se } Index = 2 \\ [P \quad Q \quad V], & \text{se } Index = 3 \\ [K \quad P \quad V], & \text{se } Index = 4 \\ [V \quad P \quad Q], & \text{se } Index = 5 \end{cases} \quad (3.33)$$

Uma das principais vantagens para a conversão do espaço de cores para o HSV é que a segmentação de plantas utiliza no geral a cor como um dos parâmetros mais significativos (PEREIRA et al., 2017), assim as variações da saturação e do brilho (iluminação) não se tornam muito prejudiciais. A conversão de volta para o RGB foi mais utilizada para recuperar, visualizar e salvar as imagens em formato parecido com a imagem de entrada.

## 3.10 Segmentação

O processo de segmentação da imagem digital visa separar o objeto de interesse, neste caso as plantas e suas folhas, do fundo. A maioria dos métodos desenvolvidos na literatura são baseados na segmentação de imagens em escala de cinza, com diferentes representações de cor. Os métodos então possuem as diferentes abordagens: limiarização de histograma, agrupamento de características espaciais, baseado em região, detecção de bordas, *Fuzzy*, redes neurais, baseado em física, ou uma combinação das abordagens anteriormente citadas. Os métodos então podem ser aplicados diretamente em um componente do espaço de cor, ou em cada componente e o resultado final ser uma combinação entre eles (CHENG et al., 2001).

### 3.10.1 Limiarização

A limiarização de histograma é uma das técnicas mais utilizadas na segmentação monocromática, assumindo que a imagem é composta de regiões com diferentes tons de intensidade e o histograma de uma imagem pode ser separado por um número de picos, cada um correspondendo a uma região. Os valores de limiar, que separa as regiões, seriam o valor correspondente ao

vale entre dois picos adjacentes. Apesar de ser um método que requer pouco processamento computacional, não considera os detalhes espaciais, não garantindo uma segmentação contínua, não funcionando bem para imagens com histograma sem picos e vales bem definidos.

Como as folhas de todas as famílias de plantas em análise possuem cores similares e o fundo apresenta elementos distintos (palha seca e terra), a limiarização simples foi escolhido como método de segmentação. Assim, a imagem de entrada é convertida para o espaço de cor HSV e apenas os elementos presentes em uma faixa de tonalidade são selecionados. Portanto, dado uma imagem  $I(x, y)$  (um canal de  $F(x, y, z)$ ), a imagem pode ser então limiarizada a partir da Equação 3.34, gerando uma nova imagem binária  $I_{bin}$ .

$$I_{bin}(x, y) = \begin{cases} 1, & \text{se } I(x, y) \in [limiar_1, limiar_2] \\ 0, & \text{c.c.} \end{cases} \quad (3.34)$$

Como utilizou apenas as informações de cor da imagem, apenas o canal H foi considerado e os valores de limiares foram escolhidos de tal forma que apenas o intervalo correspondente a cor verde fosse mantido. Erros de origem espaciais, como pequenos objetos e furos, ainda se mantêm na imagem binária. Uma forma de remove-los ou atenuá-los é a partir de operações morfológicas.

### 3.10.2 Operações morfológicas

Os operadores de morfologia permitem abordar de maneira diferenciada o processamento de imagens baseado em suas formas. Se usados de forma apropriada, tendem a simplificar informações enquanto mantêm as características essenciais de forma, eliminando irrelevâncias (HARALICK et al., 1987). Para tal processamento, é necessário duas entradas, sendo uma uma imagem binária e outro o elemento estruturante, em que o operador transformar a entrada a partir do formato do elemento. O resultado, portanto, tem correlação direta com a estrutura escolhida. Há dois operadores morfológicos básicos: erosão e dilatação (LE et al., 2020).

A erosão é definida como uma transformação de redução da imagem, diminuindo a região próxima das bordas e aumentando os buracos internos ao objeto. Já a dilatação é a operação inversa, em que as bordas são dilatadas e os buracos internos reduzidos. A combinação dos dois operadores criam outros dois: o de abertura e o de fechamento. No de abertura, uma erosão é realizada seguida por uma dilatação, suavizando o contorno do objeto e removendo os pequenos demais; já no de fechamento, uma dilatação seguida por uma erosão é performada, removendo pequenos buracos internos. Tal ferramenta é muito útil para lidar com o ruído presente na imagem binária.

Assim, dado  $I_{bin}(x, y)$  e um elemento estruturante  $E(a, b)$ , a operação de erosão é

definida pela Equação 3.35 e a de dilatação pela Equação 3.36.

$$I_{bin} \oplus E = \min[I(x + a, y + b) - S(a, b)] \quad (3.35)$$

$$I_{bin} \ominus E = \max[I(x + a, y + b) + S(a, b)] \quad (3.36)$$

Nesta pesquisa, foi analisado o uso de elementos estruturantes que possuem apenas o formato quadrado e aplicado abertura para remover pequenos objetos seguindo por uma operação de fechamento para eliminar pequenos buracos, refinando assim o resultado da segmentação. As dimensões do quadrado foram variados até apresentarem uma resposta satisfatória, tendo o objetivo de seu uso cumprido.

### 3.10.3 Métricas de qualidade

O resultado da segmentação pode ser avaliado tanto visualmente quanto a partir de diferentes métricas, entre elas a acurácia e área de sobreposição (também conhecido como coeficiente de similaridade de Jaccard), definidas respectivamente pelas Equações 3.37 e 3.38, em que  $A$  é a área dos *pixels* da imagem binária e  $B$  a do valor de referência do objeto real (ZHANG, 2001). Diferentes formas de avaliação são importantes pois enquanto a acurácia dá mais peso para os valores positivos da segmentação, a área de sobreposição considera de forma maior casos em que a área de  $I_{bin}$  é muito maior do que a da região de interesse, i.e., o objeto em análise.

$$\text{Acurácia} = \frac{\text{pixels corretamente segmentados do objeto}}{\text{número total de pixels do objeto}} \quad (3.37)$$

$$\text{Área de sobreposição} = \frac{A \cap B}{A \cup B} \quad (3.38)$$

### 3.10.4 Pseudocódigo da segmentação

O pseudocódigo 7 apresenta o processo de segmentação sumarizada, como todas as etapas anteriormente descritas inclusas. Os valores de limiar foram otimizados a partir de um conjunto pequeno de imagens do *Dataset 1* com o valor-verdade da segmentação (anotados manualmente), a partir da visualização da e das métricas de qualidade. O resultado dessa etapa foi também aplicado para o *Dataset 2*.

## 3.11 Extração de Características

Com a região de interesse selecionada, é então necessário extrair características das imagens para permitir a classificação em grupos distintos. Como a maioria das características

analisadas na literatura são baseadas em forma e textura, foram criados descritores para esses fatores. As variáveis utilizadas para os descritores podem então ser visualizadas na Tabela 6.

---

**Algorithm 7** Pseudocódigo para segmentação das imagens
 

---

**Entrada:** Imagem RGB  $F$

```

80 início
81    $hsv\_image = mudar\_espaco\_cor(F, RGB\_para\_HSV)$ 
82    $I = selecionar\_canal(hsv\_image, H)$ 
83    $dim = obter\_dimensões(I)$ 
84    $I_{bin} = criar\_matriz(dim)$ 
85   para cada pixel  $(x,y)$  em  $I$  faça
86     se  $limiar_1 < I(x,y) \leq limiar_2$  então
87        $I_{bin}(x,y) = 1$ 
88     senão
89        $I_{bin}(x,y) = 0$ 
90     fim
91   fim
92    $I_{bin} = remover\_pequenos\_objetos(I_{bin})$ 
93    $I_{bin} = remover\_buracos(I_{bin})$ 
94 fim

```

**Saída:** Imagem binária  $I_{bin}$

---

Como a maioria dos métodos desenvolvidos na literatura analisam apenas folhas individuais e se deseja trabalhar com a planta inteira, i.e., um conjunto de folhas, decidi trabalhar com janelas quadradas de dimensões fixas para a extração de características. Assim, em vez de extrair informações de folhas individuais, as características que servem para o propósito de identificação valem para a região.

Tabela 6 – Variáveis utilizadas nos descritores.

Variável	Significado
$M$	Comprimento dimensional da janela de interesse
$N$	Largura dimensional da janela de interesse
$G_n$	O número de níveis de cinza
$P_z(i, j)$	Probabilidade de um elemento cinza $i$ e $j$ aparecerem respectivamente, $i, j = 0, 1, 2, \dots, G_n - 1$
$k$	tamanho do filtro
$\xi$	desvio padrão do filtro

### 3.11.1 Descritores de forma

Os descritores de forma foram definidos a partir das bordas presentes na imagem binária  $I_{bin}$ . Decidiu-se então utilizar o LBP apenas nas curvas presentes na janela de análise (LE et al., 2020). As curvas são obtidas primeiramente por um algoritmo detector de bordas de Canny (CANNY, 1986).



### 3.11.1.1 Detecção de bordas de Canny

O processo de detecção de bordas do algoritmo de Canny pode ser separado em quatro passos (SONG et al., 2017):

1. Aplicação de filtro Gaussiano  $H_{fg}$  para remoção de ruído, definido por:

$$\begin{cases} H_{fg}(a, b) = \frac{1}{2\pi\xi^2} \exp\left(-\frac{(a - (k + 1))^2 + (b - (k + 1))^2}{2\xi^2}\right) \\ 1 \leq a, b \leq (2k + 1) \end{cases} \quad (3.39)$$

2. Cálculo da magnitude dos gradientes  $G_m$  e orientação  $G_\theta$  da imagem a partir de:

$$\begin{cases} G_m(u, v) = \sqrt{G_x^2(u, v) + G_y^2(u, v)} \\ G_\theta = \arctan\left(\frac{G_y(u, v)}{G_x(u, v)}\right) \\ G_x(u, v) = \frac{1}{2}(I(u + 1, v) - I(u, v) + I(u + 1, v + 1) - I(u, v + 1)) \\ G_y(u, v) = \frac{1}{2}(I(u, v + 1) + I(u, v) - I(u + 1, v + 1) - I(u + 1, v)) \end{cases} \quad (3.40)$$

3. Aplicação de limiar de magnitude no gradiente para eliminação de bordas falsas, etapa denominada de supressão de não-máximos. O algoritmo utiliza uma área de vizinhança 3 x 3 que consiste na interpolação de  $G_m$  ao longo da direção do gradiente  $G_\theta$  da vizinhança do pixel. Se  $G_m$  de cada ponto ao longo da direção  $G_\theta$  for maior que o valores da vizinhança, ele será marcado como possível borda;
4. Aplicação de limiar duplo para determinar possíveis bordas e definição das bordas por histerese, i.e., suprimindo as curvas que possuem valores menores de magnitude. O limiar duplo tem um limite alto e outro baixo, em que os *pixels* em que  $G_m$  estiver acima do limiar alto já são considerados pontos de borda, os abaixo do limiar baixo são descartados e os *pixels* entre os dois valores só são bordas caso estejam conectados com os *pixels* já identificados como pertencentes a borda.

Os parâmetros do algoritmo são então os do filtro  $H_{fd}$ ,  $\xi$  e  $k$ , além dos limiares de magnitude inferior e superior.

### 3.11.1.2 Padrões Binários Locais das bordas

O LBP pode ser descrito como uma relação dos *pixels* com os outros *pixels* da vizinhança (LE et al., 2020). Geralmente, o código LBP de uma janela com *pixel* central  $(x_c, y_c)$  é dado pela Equação 3.41, em que  $g_c$  indica o valor de nível do *pixel* central e  $g_p$  os valores em cada *pixel*

$p$ , em que todos os  $P_w$  pixels formam uma vizinhança circular e simétrica ao centro  $R_w$ .  $s(x)$  representa uma função de limiar, ajudando com a invariância a iluminação.

$$LBP_{P_w, R_w} = \sum_{i=0}^{P_w-1} s(g_p - g_c), \text{ em que } s(h) = \begin{cases} 1, & \text{se } h \geq 0 \\ 0, & \text{se } h < 0 \end{cases} \quad (3.41)$$

O LBP então é constituído por um vetor de 1 e 0, que pode ser transformado a invariante a rotações caso seja rotacionado até que o código binário seja o menor possível (máximo número de 0 na frente). Ele será aplicado então na imagem resultante do algoritmo de Canny, logo  $g_p$  e  $g_c$  só possuem dois valores possíveis. Apesar do LBP ser comumente utilizado como um descritor de textura, aqui ele é denominado como de forma apenas pelo fato de que além de ser aplicado na imagem original, no canal em análise, ele é aplicado em cima de uma imagem binária obtida pelo formato dos objetos na janela de análise. O descritor então foi obtido a partir do histograma do LBP.

### 3.11.2 Descritores de textura

Para os descritores de textura, foi necessário retornar os valores de intensidade originais, i.e., trabalhar com o canal H da imagem  $I(x, y)$ , utilizando como máscara a imagem binária  $I_{bin}(x, y)$ . Para obter esta nova imagem de trabalho, foi multiplicado *pixel a pixel*  $I$  e  $I_{bin}$ , obtendo-se  $T(x, y)$ . A partir de  $T$ , foi calculado a matriz de probabilidade  $P_z(i, j)$ .

Dos momentos de Haralick (HARALICK et al., 1973), foram obtidos a energia, entropia, contraste, homogeneidade e a correlação, com base nas equações abaixo:

1. Energia:

$$\sum_{i=0}^{G_n-1} \sum_{j=0}^{G_n-1} P_z^2(i, j) \quad (3.42)$$

2. Entropia:

$$- \sum_{i=0}^{G_n-1} \sum_{j=0}^{G_n-1} P_z(i, j) \log_2 P_z(i, j) \quad (3.43)$$

3. Contraste:

$$\sum_{i=1}^{G_n} \sum_{j=1}^{G_n} (i - j)^2 P_z(i, j) \quad (3.44)$$

4. Homogeneidade:

$$\sum_{i=0}^{G_n-1} \sum_{j=0}^{G_n-1} \frac{P_z(i, j)}{1 + (i - j)^2} \quad (3.45)$$

5. Média:

$$u_1 = \sum_{i=0}^{G_n-1} iP_z(i, j) \text{ e } u_2 = \sum_{j=0}^{G_n-1} jP_z(i, j) \quad (3.46)$$

6. Variância:

$$\sigma_1^2 = \sum_{i=0}^{G_n-1} (i - u_1)^2 \sum_{j=0}^{G_n-1} P_z(i, j) \text{ e } \sigma_2^2 = \sum_{j=0}^{G_n-1} (j - u_2)^2 \sum_{i=0}^{G_n-1} P_z(i, j) \quad (3.47)$$

7. Correlação:

$$\frac{\sum_{i=0}^{G_n-1} \sum_{j=0}^{G_n-1} ij P_z(i, j) - u_1 u_2}{\sigma_1^2 \sigma_2^2} \quad (3.48)$$

### 3.11.3 Pseudocódigo da extração de características

O pseudocódigo 8 representa a etapa final da extração de características. A entrada é a o canal H da imagem original e a máscara binária resultante da segmentação e a saída final é o vetor de características que descreve a região de interesse.

---

**Algorithm 8** Pseudocódigo da extração de características

---

**Entrada:** Imagem convertida HSV  $I(x, y)$ , imagem binária da segmentação  $I_{bin}$  e janela  $W_j$

95 **início**

96     **para cada**  $W_j$  **faça**

97          $I_b = \text{recortar}(I_{bin}, W_j)$

98          $\text{dim} = \text{obter\_dimensões}(I_c)$

99          $B = \text{canny\_edge}(I_b)$

100          $\text{lbp} = \text{obter\_LBP}(B)$

101          $\text{ft}_b = \text{vetor\_histograma}(\text{lbp})$

102          $I_c = \text{recortar}(I, W_j)$

103          $T = \text{criar\_matriz}(\text{dim})$

104         **para cada pixel em**  $I_c$  **faça**

105              $T(x, y) = I_c(x, y) \times I_b(x, y)$

106         **fim**

107          $H_k = \text{obter\_momentos\_Haralick}(T)$

108          $\text{ft}_1 = H_k(\text{energia})$

109          $\text{ft}_2 = H_k(\text{entropia})$

110          $\text{ft}_3 = H_k(\text{contraste})$

111          $\text{ft}_4 = H_k(\text{homogeneidade})$

112          $\text{ft}_5 = H_k(\text{correlação})$

113          $V_{ft} = \text{concatenar}(\text{ft}_b, [\text{ft}_1, \text{ft}_2, \text{ft}_3, \text{ft}_4, \text{ft}_5])$

114         **fim**

115 **fim**

**Saída:** Vetor de características  $V_{ft}$  para cada janela

---

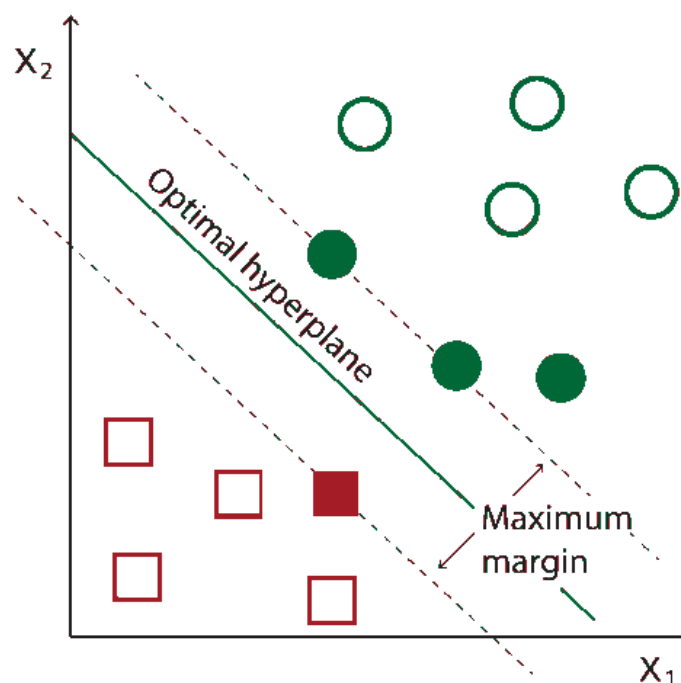
## 3.12 Classificação

A partir da revisão da literatura, observou-se a oportunidade de se avaliar o classificador SVM, entretanto considerando um aprofundamento na análise quanto ao tipo de *kernel* a ser utilizado.

O SVM é baseado no conceito de planos de decisão, que definem fronteiras entre os dados do conjunto em análise. Um plano de decisão, ou hiperplano, é um que separa os objetos em diferentes classes de pertencimento. O princípio do classificador vem a partir do caso simplificado em que duas classes de dados são linearmente separáveis e o hiperplano é capaz de diferenciar então todas as amostras em duas classes existentes. Caso os dados não sejam linearmente separáveis, o espaço de dados é então transformado em um de maior dimensão, no qual é capaz de se delimitar o hiperplano de forma linear. Funções de *kernel* então são usadas implicitamente para aplicar a transformação necessária (SABU; SREEKUMAR, 2017; REHMAN et al., 2019).

O algoritmo SVM então constrói um modelo em que novos dados são designados em uma categoria ou outra, baseado em que lado do plano ele se encontra. O SVM então requer uma fase de treinamento, com os dados de amostra providenciados para o cálculo dos parâmetros internos requisitados (os vetores de suporte), o que faz que o hiperplano seja muito dependente da amostra ofertada para o treinamento. O treinamento encontra o plano de decisão que tenha a maior distância em relação aos pontos de treinamento próximos de qualquer classe (denominado de margem funcional), de modo que quanto maior a margem menor o erro de generalização do classificador, como pode ser visto na Figura 30.

Figura 30 – Definição geométrica do SVM.



Fonte: Toledo-Pérez et al. (2019).

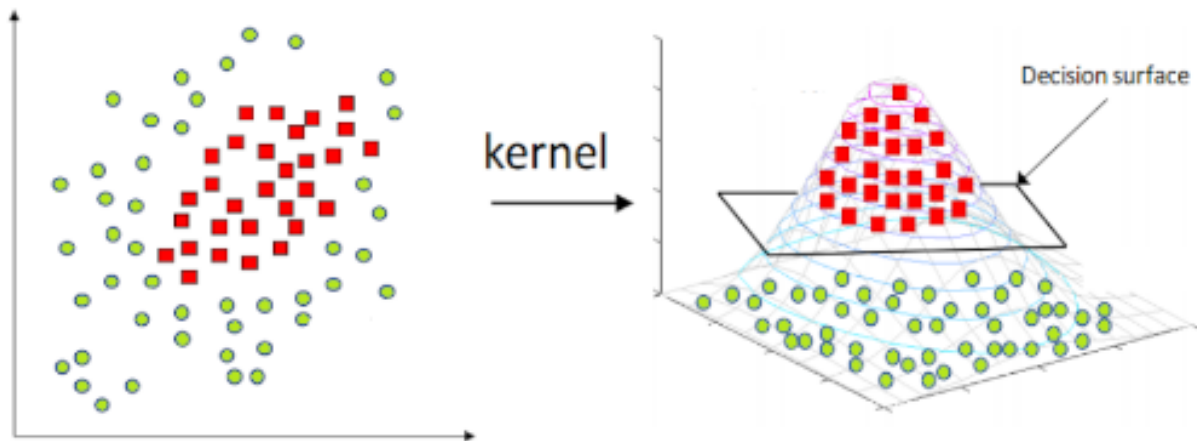
Os *kernel* mais comuns são o de função linear, polinomial, de base radial (RBF) e a sigmoide (PATLE; CHOUHAN, 2013). A Figura 31 apresenta como exemplo a transformação

do espaço por um *kernel* RBF, possibilitando que a superfície de decisão seja capaz de separar os dados de forma linear, o que não era possível antes.

O algoritmo base do treinamento de margem funcional é composto de etapas de desenvolvimento de função de decisão e de maximização da margem como método de treinamento (BOSER et al., 1992). Primeiro, considera-se um conjunto de dados  $\mathbf{v}_i$  de tamanho  $n$  e com rótulo  $lb_i$ , que se deseja agrupar em duas classe A e B:

$$(\mathbf{v}_1, lb_1), (\mathbf{v}_2, lb_2), \dots, (\mathbf{v}_n, lb_n), \text{ em que } \begin{cases} lb_k = 1, & \text{se } \mathbf{v}_k \in \text{classe A} \\ lb_k = -1, & \text{se } \mathbf{v}_k \in \text{classe B} \end{cases} \quad (3.49)$$

Figura 31 – Espaço de características original e depois de transformado por um *kernel*.



Fonte: Rizwan et al. (2021).

A partir das amostras de treino, o algoritmo retorna uma função de decisão (hiperplano)  $D(\mathbf{v})$ , que funciona da seguinte forma:

$$\begin{cases} \mathbf{v} \in A, & \text{se } D(\mathbf{v}) > 0 \\ \mathbf{v} \in B, & \text{c.c.} \end{cases} \quad (3.50)$$

A função  $D(\mathbf{v})$  é apresentada em suas duas formas na Equação 3.51. Na primeira, chamada de notação direta,  $\omega_i$  e  $b$  (o viés do classificador) são os parâmetros ajustáveis da função de decisão e  $\phi_i$  é uma função predefinida. Na segunda, denominada de notação de espaço dual,  $\alpha_k$  e  $b$  que são os coeficientes ajustáveis e  $\mathbf{v}_k$  são os padrões de treinamento utilizados.  $K$  é a função predefinida, que no caso é o *kernel*.

$$D(\mathbf{v}) = \sum_{i=1}^N \omega_i \phi_i(\mathbf{v}) + b = \sum_{k=1}^n \alpha_k K(\mathbf{v}_k, \mathbf{v}) + b \quad (3.51)$$

O objetivo do algoritmo de treinamento do SVM é encontrar o vetor  $\mathbf{w}$ , que é uma representação vetorial dos elementos  $\omega_i$  da Equação 3.51, que maximize a margem funcional  $M_g$ . A margem é dada pela Equação 3.52, sendo que a distância do hiperplano para uma amostra  $\mathbf{v}$  é dada pelo primeiro elemento da multiplicação.

$$\frac{D(\mathbf{v}_k)}{\|\mathbf{w}\|} \times lb_k \geq M_g \quad (3.52)$$

Após algumas manipulações, o treinamento do classificador pode ser resumido como um problema de minimizar  $\|\mathbf{w}\|^2$ , com a condição de que o produto  $lb_k D(\mathbf{v}_k) \geq 1$ . Os vetores em que este produto resulta em 1 são os vetores de suporte que dá nome ao classificador. A Equação 3.53 apresenta uma versão deste problema em que a margem pode ser suavizada, apresentando um erro dado por  $\zeta$  controlado pelo parâmetro de regularização  $C$ . Uma margem suave permite que o modelo apresente uma tolerância maior, permitindo a classificação incorreta de pontos a fim de obter um erro menor de generalização. Quanto maior o valor de  $C$ , mais rígido é o modelo e mais penalizado ele é por classificações incorretas.

$$\underset{\mathbf{w}, b, C}{\text{minimizar}} \left( \|\mathbf{w}\|^2 + C \sum_{i=1}^n \zeta_i \right), \text{ em que } lb_k D(\mathbf{v}_k) \geq 1 - \zeta_i \text{ e } \zeta_i \geq 0 \quad (3.53)$$

Os *kernel* analisados nesta pesquisa foram o linear (Equação 3.54), o RBF (Equação 3.55) e o Sigmoide (Equação 3.56). No kernel RBF, um  $\gamma$  maior torna a função com pico mais acentuado e conseqüentemente a função de decisão é mais complexa, melhor se adaptando para os dados de treinamento, mas menos generalizada para novos dados. Para o kernel Sigmoide, enquanto o  $\gamma$  possui mesmo impacto, há um outro parâmetro, o  $c_{f0}$ , que adiciona um grau de não-linearidade na função para valores mais longes de zero.

$$K(\mathbf{v}_k, \mathbf{v}) = \mathbf{v}_k^T \mathbf{v} \quad (3.54)$$

$$K(\mathbf{v}_k, \mathbf{v}) = \exp(-\gamma \|\mathbf{v} - \mathbf{v}_k\|^2), \text{ em que } \gamma > 0 \quad (3.55)$$

$$K(\mathbf{v}_k, \mathbf{v}) = \tanh(c_{f0} + \gamma \mathbf{v}_k^T \mathbf{v}), \text{ em que } \gamma > 0 \quad (3.56)$$

### 3.12.1 Métricas de qualidade

Para a avaliação da qualidade, são utilizadas duas métricas principais para análise de classificador: a matriz de confusão e a curva Característica de Operação do Receptor (curva ROC). Para torná-las mais robustas, foi adotado o modelo de validação cruzada de subamostragem aleatória *k-fold*.

### 3.12.1.1 Matriz de confusão

A matriz de confusão é feita a partir do resultado da classificação dos dados separados para teste ao passar pelo classificador já treinado, indicando os erros e acertos do modelo tendo o conhecimento prévio do rótulo esperado (POWERS, 2008). Ela é representada como uma matriz 2x2 dada pela Equação 3.57, em que  $TP$  é o número de verdadeiros positivos,  $FP$  o de falsos positivos,  $TN$  o de verdadeiros negativos o  $FN$  a de falsos negativos.

$$\text{Matriz de confusão} = \begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix} \quad (3.57)$$

As métricas de qualidade do classificador são definidas então como:

1. Acurácia (percentual de acertos), indica a performance geral do modelo, quantas amostras foram classificadas corretamente:

$$\frac{TP + TN}{TP + TN + FP + FN} \quad (3.58)$$

2. Precisão (confiança), quantas amostras classificadas como positivas estão corretas:

$$Prec = \frac{TP}{TP + FP} \quad (3.59)$$

3. Sensibilidade (*recall*), dentre todas as situações em que o valor esperado seria positivo, quantas estão corretas:

$$Sens = \frac{TP}{TP + FN} \quad (3.60)$$

4. F-score, média harmônica entre a precisão e a sensibilidade:

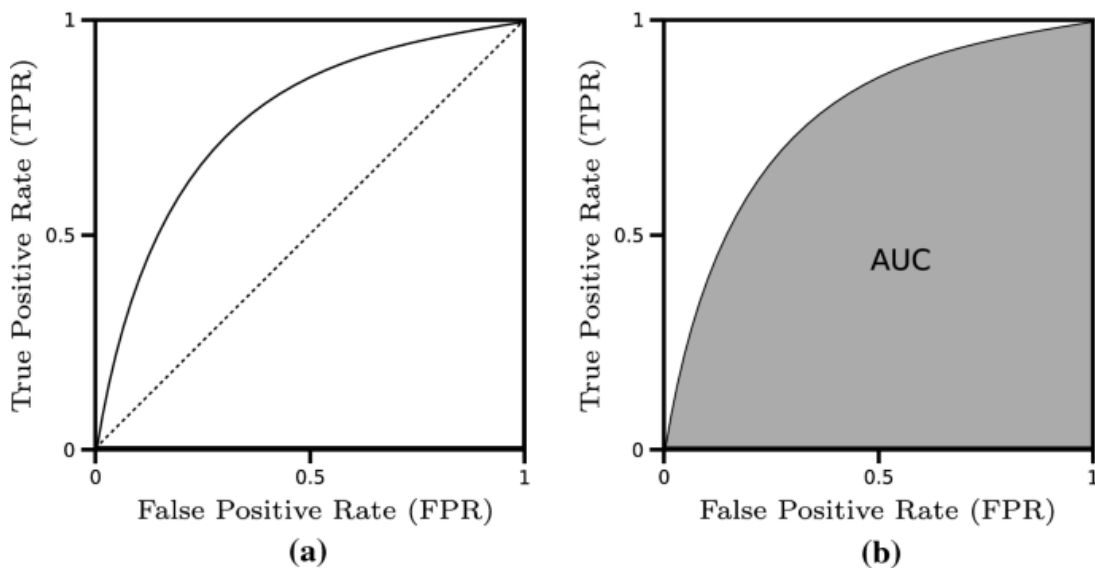
$$\frac{2(Prec - Sens)}{Prec + Sens} \quad (3.61)$$

A acurácia é uma medida geral de performance do classificador, porém ela pode não representar bem os dados caso a proporção esperada entre verdadeiros e falsos seja muito discrepante. A precisão já é mais significativa para casos em que  $FP$  são mais prejudiciais do que  $FN$ , enquanto a sensibilidade mede a situação inversa. Caso a aplicação de herbicida não seja prejudicial no momento de operação do sistema (podendo variar devido a cultura em análise e ao estágio de crescimento dela), a sensibilidade se torna mais relevante; já na situação que deseja evitar ao máximo a aplicação do produto, a precisão se torna essencial. Por fim, o F-score apenas combina essas duas informações, sendo que valores baixos indica que ou a precisão ou a sensibilidade estão baixos.

### 3.12.1.2 Curva ROC

A curva ROC (Figura 32) é uma representação gráfica do desempenho de um classificador binário, em que se analisa a relação normalizada entre a taxa de  $FP$  e a taxa de  $TP$ . A métrica principal que será usada a partir dela é a análise da área debaixo da curva ( $AUC$ ), que quanto maior melhor (JASKOWIAK et al., 2022). A diagonal do gráfico indica um classificador aleatório, em que todas as amostras foram classificadas sem critério com mesma probabilidade de pertencem a qualquer classe. Quanto mais a curva se afasta dessa diagonal, aproximando do canto superior esquerdo, melhor é o classificador e maior é a  $AUC$ . Quanto mais se afasta para o canto inferior direito, menor a  $AUC$  e pior é o classificador (apesar de quem em casos extremos basta invertê-lo, espelhando a curva pela diagonal).

Figura 32 – Exemplo de curvas ROC, com anotação da linha diagonal e da  $AUC$ .



Fonte: Jaskowiak et al. (2022).

### 3.12.1.3 Validação cruzada

A validação cruzada é uma técnica de amostragem de dados para avaliar a capacidade de generalização de modelos preditivos e evitar o sobre-ajuste do modelo de classificação com uma amostra específica utilizada no treinamento, garantindo a generalização do classificador supervisionado (BERRAR, 2018).

O conjunto de amostras que está disponível para construir e avaliar um modelo preditivo é referido como o conjunto de aprendizagem  $D_{apr}$ , que é considerado uma representação de uma população de interesse. Os métodos de subamostragem aleatória são usados então para gerar conjuntos de treinamento  $D_{tr}$  e conjuntos de teste  $D_{tst}$ , a partir de  $D_{apr}$ , para configuração do classificador.



A técnica de validação cruzada de subamostragem aleatória *k-fold* foi utilizado, em que a amostragem é feita de forma que não haja sobreposição entre os  $D_{tst}$  gerados.  $D_{apr}$  é particionado em  $k$  subconjuntos distintos de tamanho aproximadamente igual, de modo que enquanto um é usado para teste, os outros são utilizados no treinamento. Essa ação é repetida até que todas as partições tenham sido utilizadas para teste, com as métricas finais dadas pela média dos resultados obtidos para os  $k$  casos. Atenta-se que para a criação dos subconjuntos da amostra, ela deve também ser estratificada, garantindo que a proporção das classes neles reflita as proporções do conjunto de aprendizagem.

### 3.12.2 Pseudocódigo do classificador

O pseudocódigo 9 apresenta o processo de treinamento do classificador SVM a partir dos dados obtidos pela extração de características.

---

**Algorithm 9** Pseudocódigo para treinamento e teste do classificador

---

**Entrada:** Vetores de características  $V_{ft}$  de todas as  $n$  amostras, parâmetros do classificador  $kernel$  e  $param$ , número de dobras de validação cruzada  $k$

```

116 início
117   Dados = criar_matriz([n,número de características])
118   para cada amostra faça
119     |  $Dados[n] = V_{ft}$ 
120   fim
121   Dados = normalizar(Dados)
122   Dados_treino, Dados_teste = particionar_k_fold(Dados, k)
123   Mt = criar_matriz(k)
124   para cada dobra faça
125     |  $Cls = treinar\_SVM(Dados\_treino[dobra], kernel, param)$ 
126     |  $Mt[dobra] = testar\_SVM(Cls, Dados\_teste[dobra])$ 
127   fim
128   Métricas = média(Mt)
129 fim

```

**Saída:** Métricas de avaliação do classificador

---

Todas as amostras (de ambos os *datasets*) foram rotuladas manualmente, sendo que no *Dataset 1* há três classes, duas referentes às espécies de plantas invasoras e uma para o caso de não ser nenhuma delas; e no *Dataset 2* há quatro classes, duas para as invasoras, uma para o amendoim e uma para o caso nulo. As amostras foram divididas aleatoriamente em dois grupos, sendo que 80% delas foram usadas para treinamento e as outras 20% para teste, realizando assim uma validação cruzada de 5 dobras. Variou-se os tipos de *kernel* e os parâmetros deles a fim de se obter o melhor, utilizando para essa avaliação as métricas discutidas anteriormente.

Para as métricas de curva ROC e matriz de confusão, utilizou um classificador com os parâmetros validados em etapas anteriores. As amostras em análise são novamente sorteadas e cria-se um objeto de classificador com os parâmetros estabelecidos. Ele é treinado então

novamente e testado. A partir dos resultados do texto, é obtido tanto a matriz de confusão quanto a curva ROC para cada classe de interesse, com indicação da AUC dela.

Também é obtido uma curva ROC a partir da micro-média das classes, em que é operado uma média ponto a ponto das curvas, gerando então uma função final representante do classificador em sua totalidade.

# Capítulo 4

## RESULTADOS E DISCUSSÕES

---

---

Neste Capítulo são apresentados resultados e discussões, incluindo a caracterização e avaliação das etapas do método desenvolvido (*hardware*, *software* e *firmware*), a validação com base em experimentos de laboratório e experimentos reais realizados em ambiente de campo.

### 4.1 Qualificação do método e do sistema

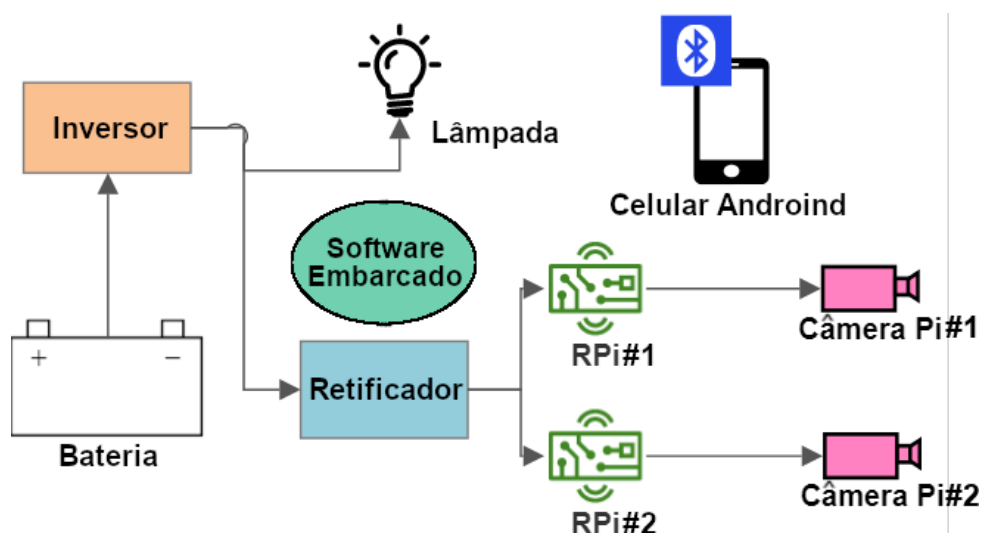
O sistema desenvolvido envolve algoritmos computacionais do processamento de imagens e sinais, bem como inclui algoritmos para classificação dos padrões de famílias de plantas invasoras, enquanto que para sua operação foi considerado o arranjo físico envolvendo partes mecânicas, partes eletrônicas e *hardware* computacional (Figura 33). Todos os componentes foram conectados com o auxílio de cabos e conectores apropriados. O sistema foi controlado por um aplicativo via comunicação serial Bluetooth, em um celular Galaxy de modelo SM-A715F com processador Snapdragon 730 2,2 GHz e sistema operacional Android versão 13. Todos os códigos foram desenvolvidos em Python, com o auxílio das bibliotecas necessárias para o controle da câmera, habilitação de comunicação serial e bluetooth, processamento de imagens e manipulação de dados. Apenas os algoritmos de captura de imagem foram realizados no sistema operacional da RPi; os processamentos e avaliações subsequentes foram feitas em sistema operacional Windows 10, em uma máquina com processador AMD Ryzen 5 3400G 3,70 GHz e em outra máquina com processador Intel Core i7-9700 3 GHz.

Todos os componentes foram fixados em uma estrutura metálica (Figura 34), que possui um puxador para auxílio do transporte e pés paralelos de lados opostos que permitem uma melhor estabilização do dispositivo no chão. Na parte frontal se encontra apenas a caixa para proteção das placas e os elementos acoplados a ela, que são o *stereo rig*, as câmeras e a lâmpada LED. Já na parte traseira (Figura 35) se encontram a bateria e o retificador, sendo que a bateria pode ser facilmente desprendida da estrutura caso necessário, para a troca por outra carregada ou para facilitar na locomoção do arranjo.

No interior da caixa de proteção (Figura 36) se tem as RPi empilhadas e o retificador. No centro dela, na parte inferior, se encontra o bocal da lâmpada, permitindo que ela seja trocada

caso necessário e na frente dela o *stereo rig* com as câmeras dispostas de modo coplanar; ambos os elementos podem ser visualizados na Figura 37. A lâmpada LED foi configurada de tal forma que o sistema pode operar com ela acessa ou não, porém para os experimentos a primeira opção foi mantida.

Figura 33 – Diagrama da conexão entre os componentes do sistema.



O sistema construído tem três elementos com possibilidade de ajuste. O primeiro é a altura da caixa protetora em relação ao solo, que também define a distância das câmeras ao chão. A partir de um trilho, ela pode ajustada para qualquer intervalo entre 10 a 100 cm. O segundo elemento é a distância entre as câmeras, i.e., a distância de base, que também podem ser ajustadas a partir de um trilho entre um limite de 6 até 24 cm. O último elemento é a inclinação das câmeras, que podem girar a partir do eixo central do *stereo rig* (giro de um grau de liberdade), assumindo quatro posições. Definindo a posição da câmera quando paralela ao chão como o ângulo referencial  $0^\circ$ , ela pode ser alternada para  $90^\circ$  (apontada para a frente),  $180^\circ$  ou  $270^\circ$ .

### 4.1.1 Gerenciamento de memória

Após a instalação do sistema operacional, das ferramentas e programas essenciais e do desenvolvimento dos códigos, os 32 GB do cartão de memória SD foi reduzido para 23,1 GB. Essa memória foi considerada então como o limite de memória disponível para uso geral do sistema. Para primeira avaliação da memória, foram analisadas as taxas de escrita e leitura de dados, resultando então na Tabela 7, como o valor mínimo requisitado pela RPi e os valores obtidos em cada placa (IOPS é uma abreviação de *Input/Output per Second*). Como se considerou como fator limitante o tempo em que o sistema embarcado leva para salvar os arquivos de imagem, descobriu-se que a taxa de gravação da memória sequencial do cartão SD é aproximadamente de 14,8 MBytes/s.

A segunda parte da análise do gerenciamento de memória foi a definição da resolução das imagens capturada. Seguindo a dimensão do sensor da câmera, a resolução aproveitou todo o tamanho dele, ou seja, a proporção 4:3 das imagens foi preservada. Para referência, o tamanho máximo que um arquivo pode ter foi calculado multiplicando a resolução pela profundidade de *pixel* (referente ao formato RAW). Foram testadas cinco resoluções, 640 x 480, 800 x 600, 1024 x 768, 1280 x 960 e a máxima 2592 x 1944, analisando a média do tamanho dos arquivos PNG gerados. O resultado então pode ser visto na Figura 38.

Figura 34 – Visão frontal do sistema.

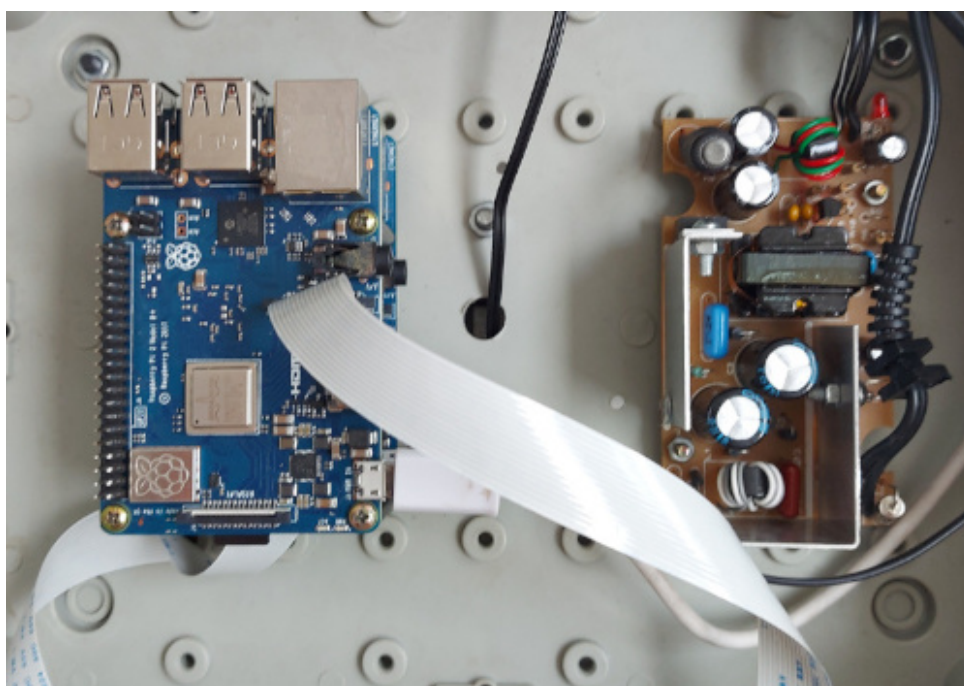


Considerando a aplicação em processamento de imagens, em que o custo computacional das operações tende a crescer exponencialmente em função do número de *pixels* presentes e da memória SD disponível, optou-se então pela resolução de 1280 x 960, em que as imagem tem aproximadamente 1,8 MBytes de tamanho. Com essa resolução, podem ser salvas na memória pelo menos 6.000 imagens, embora seja possível armazená-las posteriormente na nuvem, a partir da comunicação do sistema com uma rede externa, liberando espaço na placa. Com essa resolução, o tempo de escrita permite o registro de até 8 imagens por segundo, mas para minimizar possíveis erros de operação, decidiu adicionar um *delay* de 1 seg após cada operação

Figura 35 – Visão traseira do sistema.



Figura 36 – Interior da caixa protetora, com RPis e retificador.



de escrita do código desenvolvido. Observa-se que após a realização do experimento de campo,

Figura 37 – Detalhes da câmera, equipamento estéreo e lâmpada.



foram geradas 445 imagens e a memória armazenada foi de aproximadamente 850 MBytes em cada RPi.

Tabela 7 – Teste de velocidade do cartão SD.

	Escrita sequencial (kbytes/s)	Escrita aleatória (IOPS)	Leitura aleatória (IOPS)
Valor mínimo	10000	500	1500
RPi 1	14833	726	1881
RPi 2	14969	699	1892

#### 4.1.2 Gerenciamento de consumo de energia

A RPi pode ter consumo de energia de até 12,5 W, mas em testes de laboratório o valor usual durante a aplicação do *software* de captura de imagem era de apenas 3 W. Como o sistema foi projetado com inversor, a potência consumida por este equipamento também foi considerada para a avaliação. Neste caso, o inversor em questão apresentou uma potência gasta em torno de 8,4 W, significativamente superior à soma das RPis.

Para medir o gasto energético do sistema, foram obtidas a corrente e a potência em diferentes situações, conforme Tabela 8, com tensão da bateria fixada em 12,0 V. Para avaliar a capacidade da bateria, foi realizado um teste nas condições mais extremas situação, com o sistema em funcionamento contínuo com a lâmpada LED de 18 W acesa, o que resultou na

manutenção do funcionamento por aproximadamente 15 horas. Após descarregar para 11,7 V, por segurança, o inversor parou de funcionar. Observou-se que, neste modo de operação, o pico de corrente na inicialização do sistema foi próximo a 3,2 A, enquanto com a mesma configuração, porém com a lâmpada LED menos potente, o pico foi de 2,0 A.

Figura 38 – Experimentos de tamanho de arquivo de imagem.

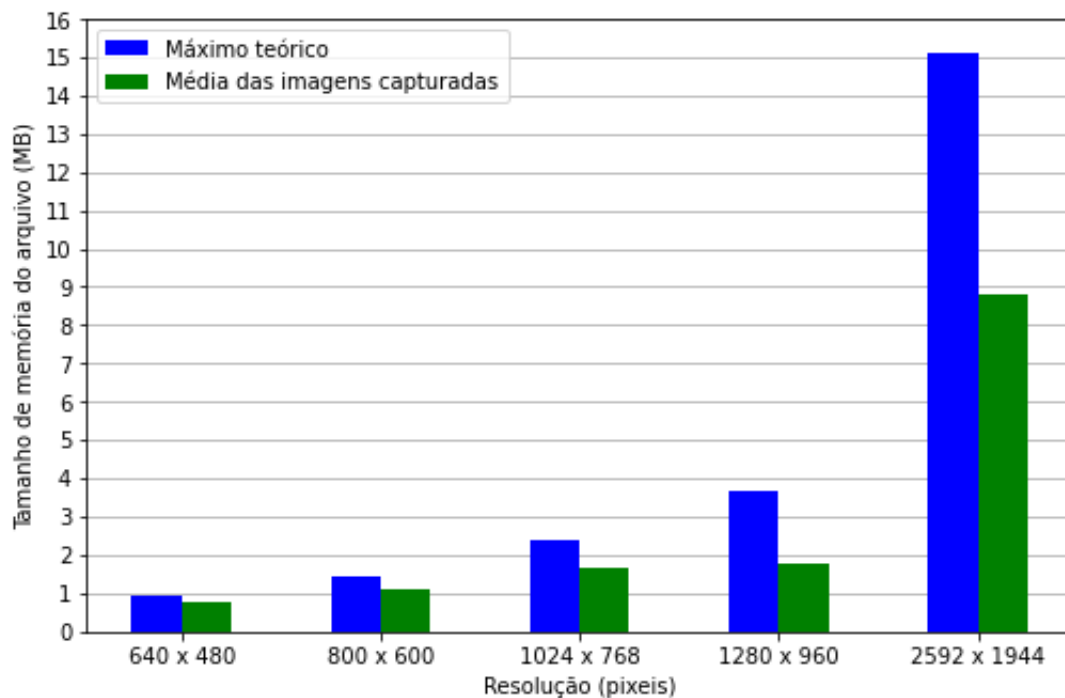


Tabela 8 – Energia do sistema em diferentes configurações.

Modo de operação	Corrente (A)	Potência (W)
Padrão	1,3	15,6
Com as câmeras ativas	1,4	16,8
Com as câmeras ativas e lâmpada LED de 4,5 W	1,9	22,8
Com as câmeras ativas e lâmpada LED de 18 W	3,0	36,0

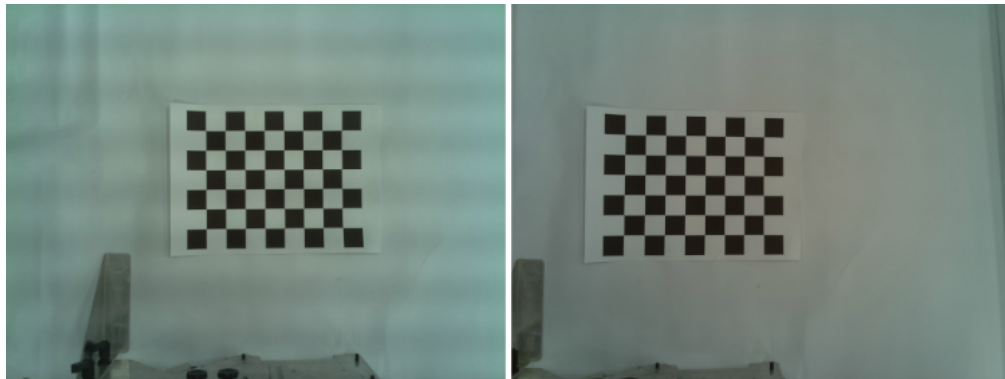
## 4.2 Validação das câmeras

Nesta seção é apresentada a validação das câmeras utilizadas, a partir da avaliação da MTF e do cálculo dos parâmetros delas, além de obter a MTF e a distância de base do sistema. Para ambos os métodos, foi necessário um conjunto de imagens de um tabuleiro de xadrez, capturadas simultaneamente pelas duas câmeras. O resultado pode ser exemplificado na Figura 39. O tabuleiro escolhido é formado por quadrados de 40 cm de lado, distribuídos de tal forma que tenha 5 x 8 pontos de intersecção internos (são 6 por 9 quadrados, mas como os métodos de



calibração utilizam para o rastreamento do objeto apenas os pontos internos, é convencional definir o tabuleiro por eles). Foram obtidas no total 24 imagens, com variação da posição e rotação do tabuleiro entre elas.

Figura 39 – Imagens do tabuleiro de xadrez, capturadas de forma síncrona e não processadas pelas câmeras esquerda e direita.



#### 4.2.1 Definição da linha de base

O protótipo desenvolvido possui linha de base mínima possível de 6 cm e máxima de 24 cm, que o torna capaz de simular a visão humana, que possui este valor na faixa de 5,4 a 7,4 cm, além de permitir a exploração de outros cenários. Para definir a distância foram consideradas as Equações (3.7) e (3.11), calculando tanto a disparidade esperada para um objeto a até 1 m de distância da câmera quanto o erro de distorção esperado a tal distância, para quatro valores de linha de base, 6 cm, 12 cm, 18 cm e 24 cm, como pode ser visto na Figura 40, considerando a resolução escolhida de 1280 x 960.

Ao definir a distância da linha de base, é sempre preferível usar os valores mais baixos para garantir uma maior interpolação entre as duas imagens geradas, o que permite que objetos mais próximos tenham sua distância calculada. Por exemplo, de acordo com o gráfico apresentado, para  $b = 24$  cm, objetos a até 23,8 cm de distância da câmera não estariam presentes em ambas as imagens, impossibilitando o cálculo da disparidade, enquanto para  $b = 6$  cm tal situação é válido apenas para objetos a menos de 5,9 cm de distância. Já para objetos de até 1 m longe das câmeras, o erro de distorção se mostrou pequeno para todos os casos, inclusive para o cenário com a menor linha de base. Portanto, foi definido a distância de base igual a 6 cm para a aquisição das imagens no experimento de campo.

#### 4.2.2 Obtenção da função de transferência modular

A partir do conjunto de imagens obtidos, foram selecionadas cinco regiões aleatórias (no mesmo local para ambas as câmeras) onde há transição de intensidade do branco para o

preto, para uso do método do gume de faca. É calculado a ESF e a LSF então para cada amostra, como pode ser exemplificado na Figura 41. A partir de todas as amostras, a MTF normalizada é calculada pela média das amostras. A MTF do sistema estéreo foi então calculado a partir da multiplicação de ambos as MTFs, como pode ser visto na Figura 42. Para a câmera esquerda, o contraste reduziu 50% na frequência especial normalizada de 0,327 ciclos/pixel e reduziu 90% em 0,551 ciclos/pixel; para a câmera direita, os valores foram respectivamente na frequência 0,286 e 0,673 ciclos/pixel; e para o sistema todo, o contraste de 50% foi em 0,224 ciclos/pixel e o de 10% em 0,367 ciclos/pixel. O valor da MTF na frequência de Nyquist foi então 14,31% para a câmera esquerda, 8,97% para a direita e 1,28% para todo o sistema. Como o valor da MTF foi maior que 5% (redução de contraste que ainda permite a recuperação das bordas dos objetos em imagens não tão ruidosas) em todos os sensores, assim como maior que 0% para o sistema como um todo, logo o sensor atende as necessidades do projeto.

As folhas das famílias de plantas invasoras apresentaram as seguintes médias de dimensões de comprimento versus largura em *pixels*: 36 x 19 para a leiteira; 45 x 10 para o capim-carrapicho; 220 x 15 para o capim braquiária; e 191 x 134 para a mucuna-preta. Com a MTF, é possível saber se a imagem possui contraste suficiente para diferenciar as folhas quando muito próximas, analisando então a frequência delas. Assim, como a menor dimensão das folhas é mais relevante, se tem que elas apresentam as frequências, em ciclos/pixel, de: 0,053 para a leiteira; 0,100 para o capim-carrapicho; 0,067 para o capim braquiária; e 0,007 para a mucuna. Considerando então a maior frequência das folhas como ponto crítico, a MTF em tal ponto apresentou valor de 97,23% para a câmera esquerda, 91,74% para a direita e 89,21% para todo o sistema. Tais valores de perda de contraste de aproximadamente 10% são bem pequenos, validando então o sensor como adequado para a aplicação.

### 4.2.3 Calibração das câmeras

A partir do conjunto de imagens de tabuleiros de xadrez e das equações de calibração da câmera, foram usados todos os pares de imagens para encontrar os parâmetros delas. As matrizes das câmeras e os coeficientes de distorção calculados foram então:

- Câmera esquerda:

$$\mathbf{M}_{\text{cam}} = \begin{bmatrix} 736 & 0 & 582 \\ 0 & 735 & 464 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

$$\mathbf{K}_{\text{cam}} = [0,0589 \quad -0,169 \quad 0,00139 \quad 0,00198 \quad 0,142] \quad (4.2)$$

Figura 40 – Disparidade da distância da linha de base e avaliação do erro de distorção.

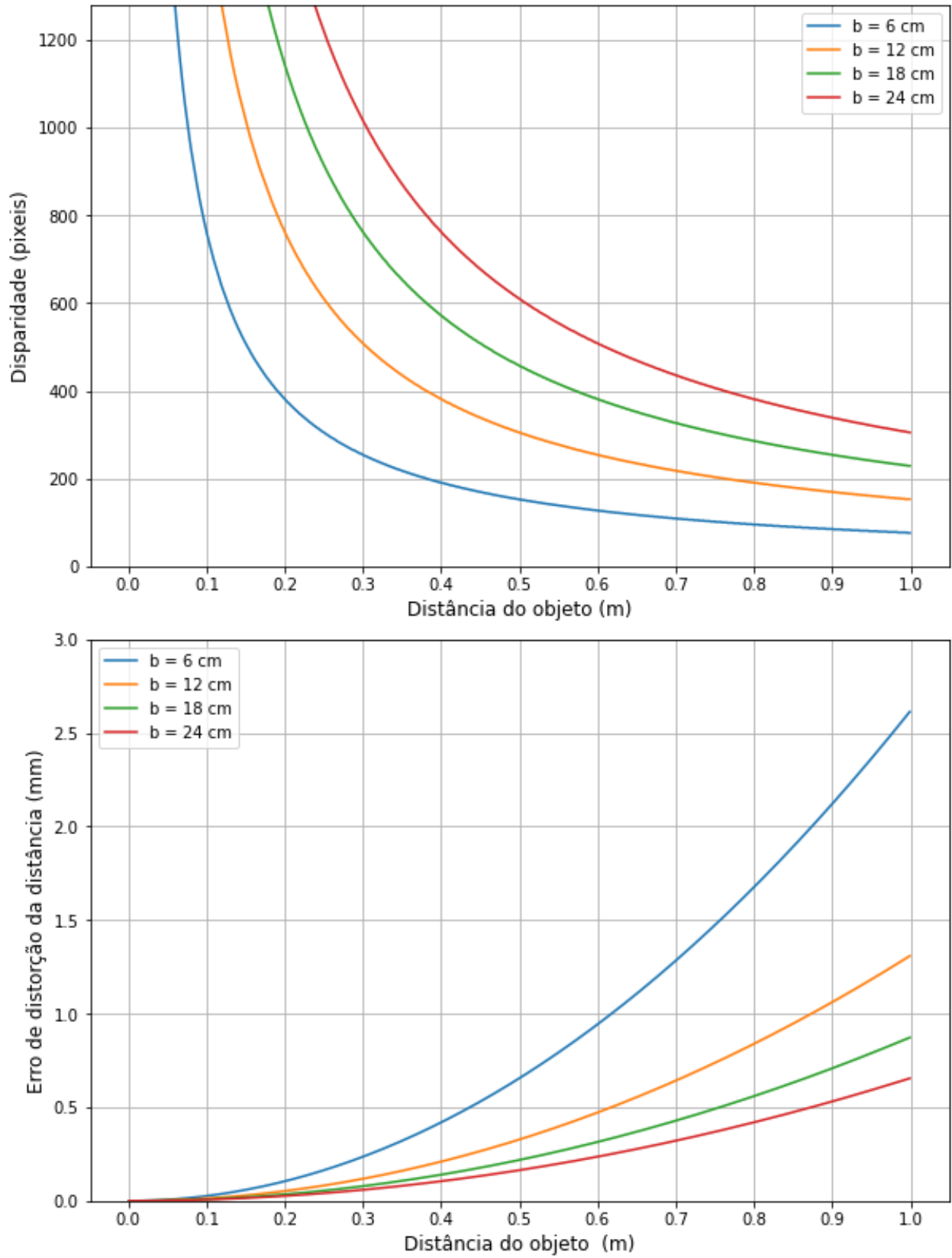


Figura 41 – ESF e LSF de uma amostra da câmera esquerda e direita.

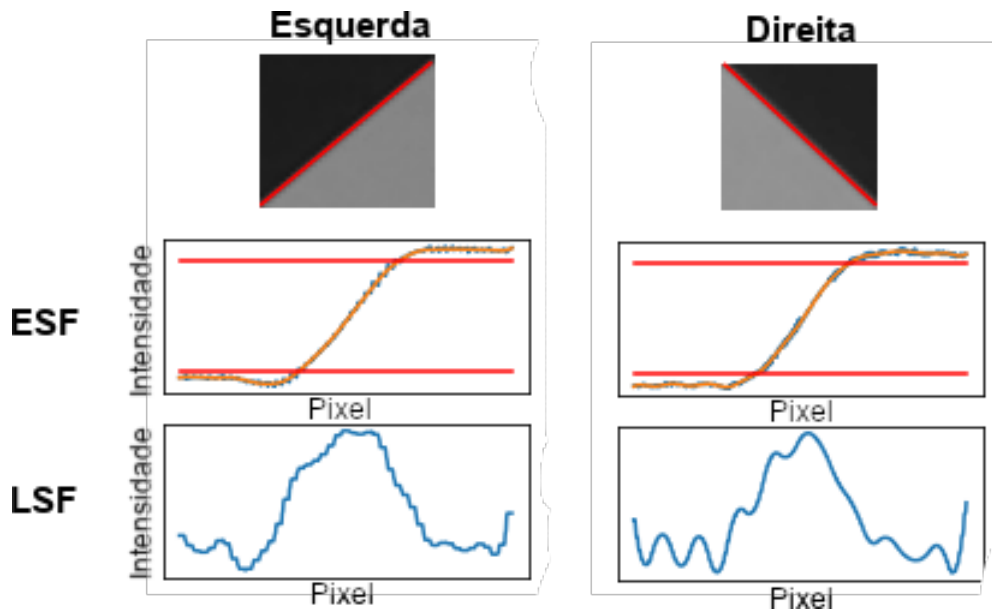
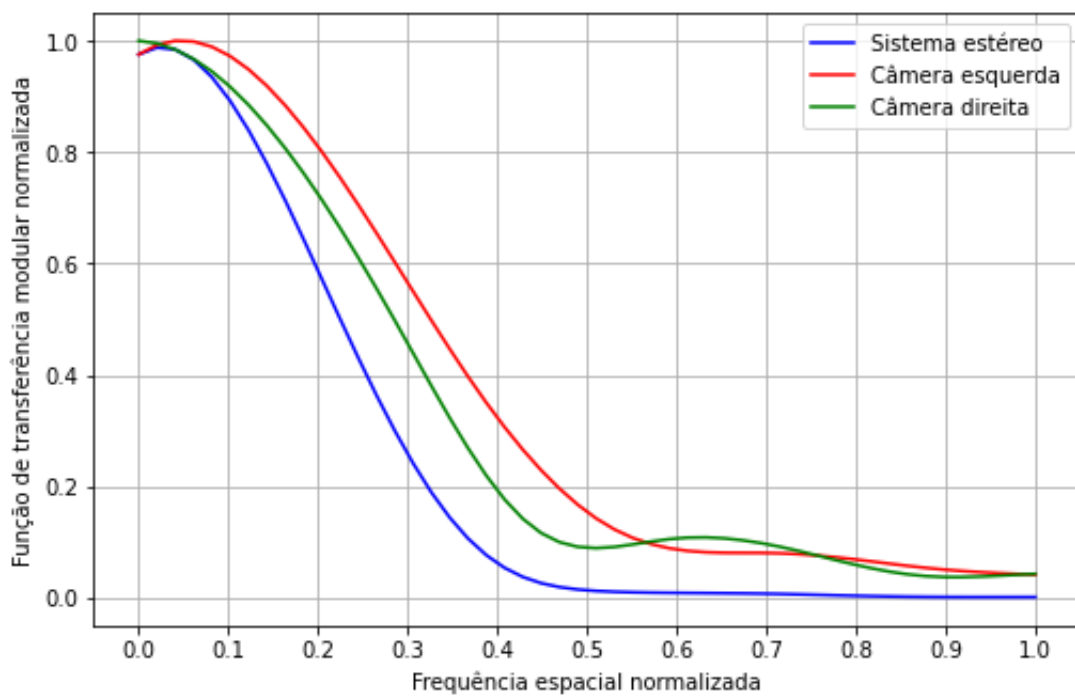


Figura 42 – MTF de cada sensor de câmera e do sistema.



- Câmera direita:

$$\mathbf{M}_{\text{cam}} = \begin{bmatrix} 1480 & 0 & 681 \\ 0 & 1480 & 480 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

$$\mathbf{K}_{\text{cam}} = \begin{bmatrix} -0,0728 & 3,98 & 0,00117 & 0,00630 & -22,6 \end{bmatrix} \quad (4.4)$$

Já a calibração do sistema estéreo foi dada pelas seguintes matrizes:

$$\mathbf{R}_{\text{stereo}} = \begin{bmatrix} 0.960 & -0.0133 & -0.281 \\ 0.0159 & 1.00 & 0.00721 \\ 0.280 & -0.0114 & 0.960 \end{bmatrix} \quad (4.5)$$

$$\mathbf{T}_{\text{stereo}} = \begin{bmatrix} -0.787 \\ -0.0670 \\ 5.65 \end{bmatrix} \quad (4.6)$$

A partir de todas as matrizes obtidas foi realizado a retificação das imagens capturadas.

### 4.3 Resultados em campo experimental

O foco desta seção foi a aplicação do algoritmo de aquisição de imagens para se obter o *Dataset 2*, realizado no campo experimental da cultura do amendoim. Como a área de análise era  $72 \text{ m}^2$  ( $5 \times 14,4 \text{ m}$ ) e na semana em que o experimento foi realizada as plantas tinham altura entre 20 e 30 cm, decidiu-se configurar o sistema de tal modo que a altura das câmeras ao solo era de 1 m, respeitando o foco da lente recomendado pelo fabricante.

Com tal altura e segundo a Equação 3.1, tem-se que:

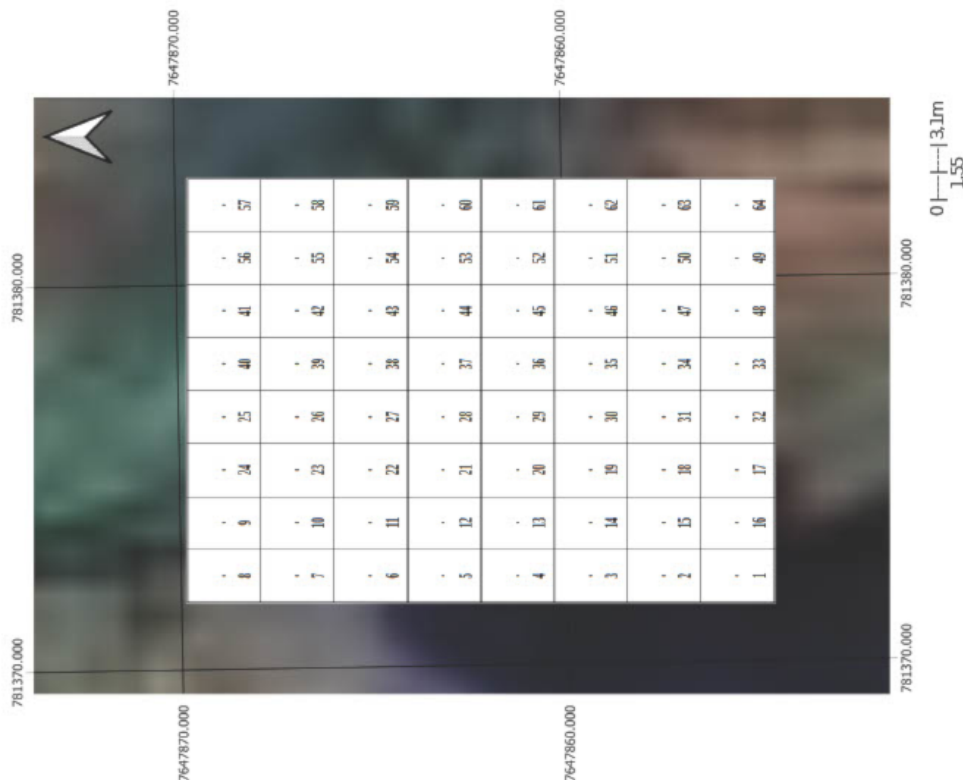
$$A_{\text{cap}} = 4(1)^2 \tan\left(\frac{53,50^\circ}{2}\right) \tan\left(\frac{41,41^\circ}{2}\right) = 0,76 \text{ m}^2 \quad (4.7)$$

A área experimental então foi particionada de modo  $8 \times 8$ , em que se tem uma imagem por bloco e cada imagem registra  $0,76 \text{ m}^2$  da área experimental. A localização dos dados georreferenciados das imagens e a divisão da área de estudo pode ser vistas na Figura 43.

O banco de imagens então foi formado da seguinte maneira: como para cada bloco foram capturadas em média 5 imagens, avaliou-se primeiro a qualidade delas e eliminou as que apresentaram problemas; selecionou de forma aleatória um exemplar entre as 5 para representar o bloco; por fim, as imagens foram renomeadas e armazenadas em um repositório específico. Assim, o *Dataset 2* é formado por 64 pares de imagens de  $1280 \times 960$ , um par por bloco e cada par com uma imagem referente à câmera esquerda e outra à câmera direita. Essas imagens

foram nomeadas de modo que as informações do bloco a qual se referem e da câmera de origem estejam corretamente indicadas. No momento de captura, o sistema estava orientado no sentido leste para os blocos 1 ao 8, 17 ao 24, 33 ao 40 e 49 ao 56. Para os blocos restantes, o sistema estava orientado com a frente para o oeste.

Figura 43 – Localização geográfica e divisão da área de estudo.



A Figura 44 apresenta o sistema no campo experimental, realizando a captura das imagens, com exemplo de dupla de imagens capturadas de forma simultânea e a composição 3D ciano-vermelho das imagens retificadas. Já a Figura 45 apresenta exemplos das imagens capturadas. Como o experimento foi realizado durante o período matutino de um dia nublado, em algumas imagens se tem a iluminação direta e em outras a difusa proveniente do sol, o que faz que se tenha a presença de sombras em parte da amostra. Devido a altura das câmeras, parte do sistema também aparece nas imagens, que será removido depois na segmentação. Percebe-se que, diferentemente das imagens do *Dataset 1* (Figura 23), boa parte da imagem é preenchida pela vegetação, com pouco solo aparente.

Para a qualificação dos algoritmos, as imagens de ambos os *datasets* tiveram valores verdades gerados manualmente para fim de validação da segmentação e da classificação. A Figura 46 apresenta exemplos do valor verdade da segmentação do *Dataset 1*, em que os *pixels* branco se referem ao objeto de interesse, enquanto que a Figura 47 apresenta exemplos do valor verdade para o *Dataset 2*. Na Figura 48 é possível visualizar exemplos da anotação manual das

classificação das plantas presentes nas imagens do *Dataset 1*, em que as regiões com plantas de folhas largas (leiteira) estavam presentes foram anotadas com a pseudo-cor verde, as plantas de folhas estreitas (capim-carrapicho) com a pseudo-cor azul, outras plantas presentes que não são as duas plantas invasoras em análise anotadas com a pseudo-cor vermelha e o fundo (terra, palha e suporte de madeira) com a pseudo-cor preta. Já no *Dataset 2*, as regiões em que plantas de folhas largas (mucuna-preta) estavam presentes foram anotadas com a pseudo-cor verde, as plantas de folhas estreitas (capim-braquiária) com a pseudo-cor azul, o amendoim com a pseudo-cor vermelha e outras plantas, terra e estrutura com a pseudo-cor preta, como pode ser visto na Figura 49.

Figura 44 – Tomada em tempo real do *Dataset 2* (imagens em escala ilustrativa).

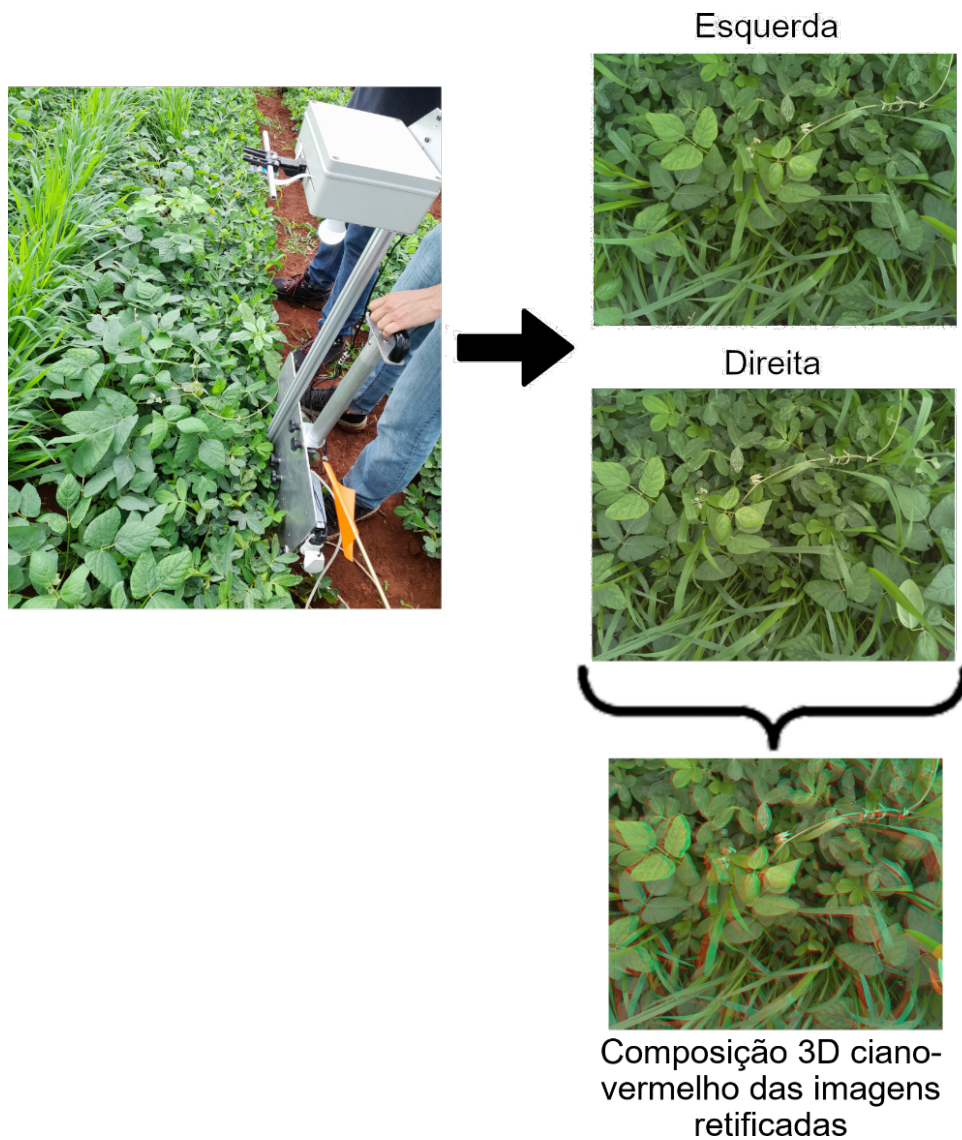


Figura 45 – Exemplo de imagens adquiridas para o *Dataset 2*, com tomadas respectivamente da câmera esquerda e da câmera direita.

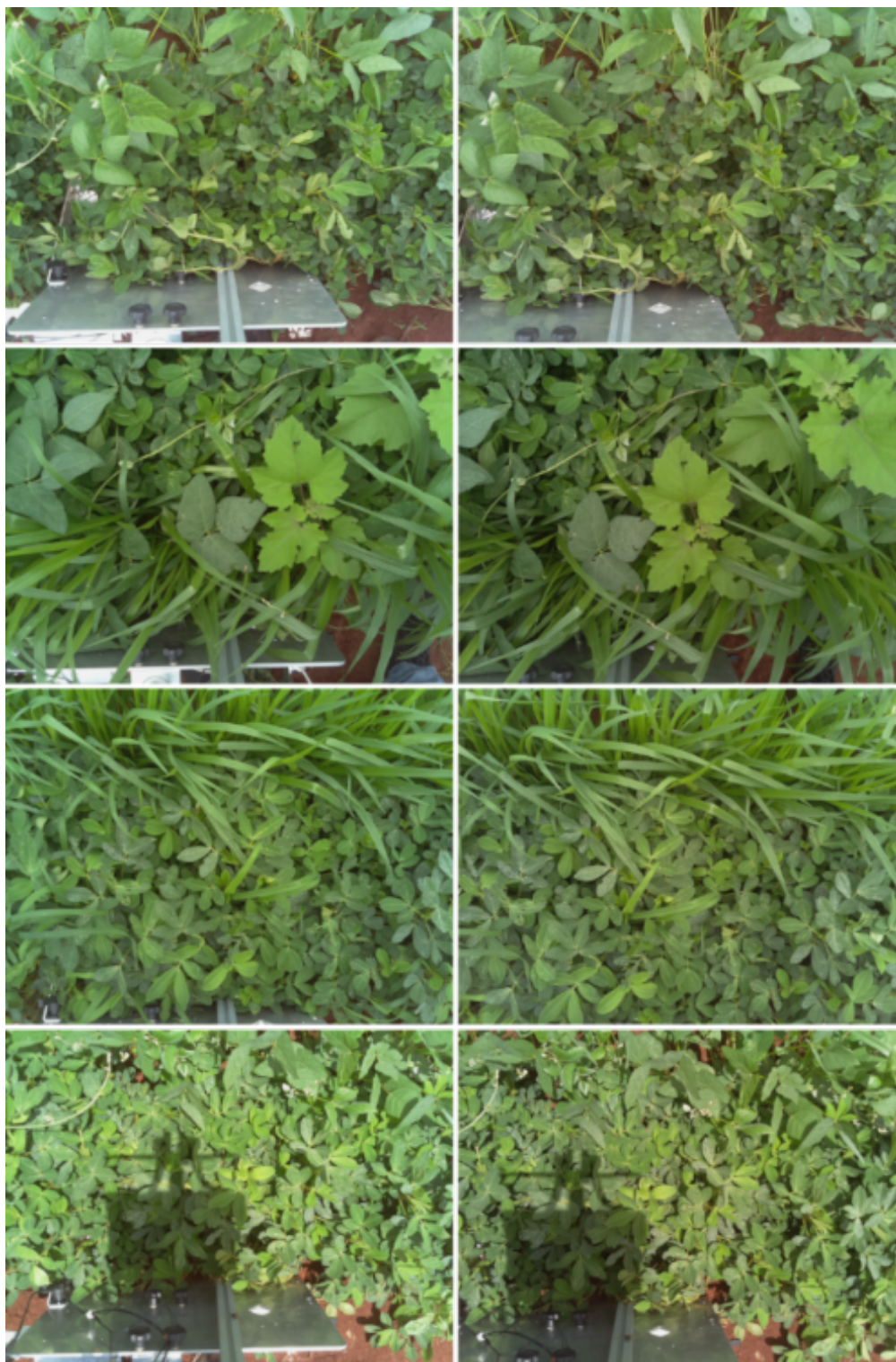




Figura 46 – Exemplo de imagens de valor verdade da segmentação do *Dataset 1*.

Figura 47 – Exemplo de imagens de valor verdade da segmentação do *Dataset 2*.

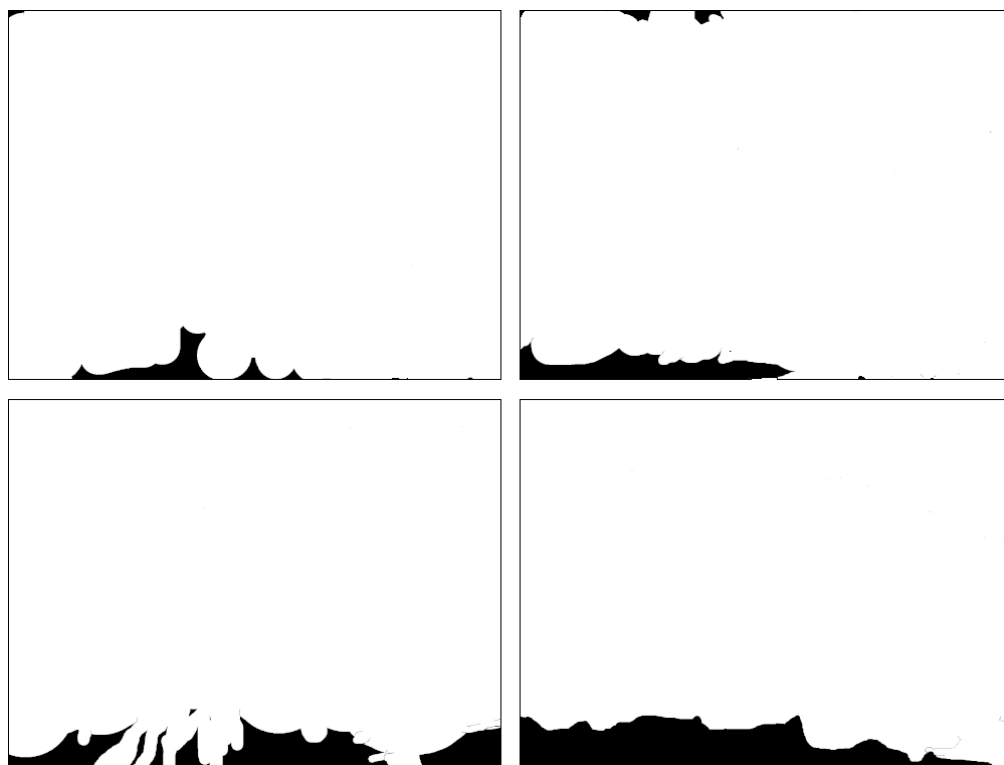


Figura 48 – Exemplo de imagens com anotação manual de classes do *Dataset 1*, com a leiteira em verde, o capim em azul e outras plantas em vermelha.

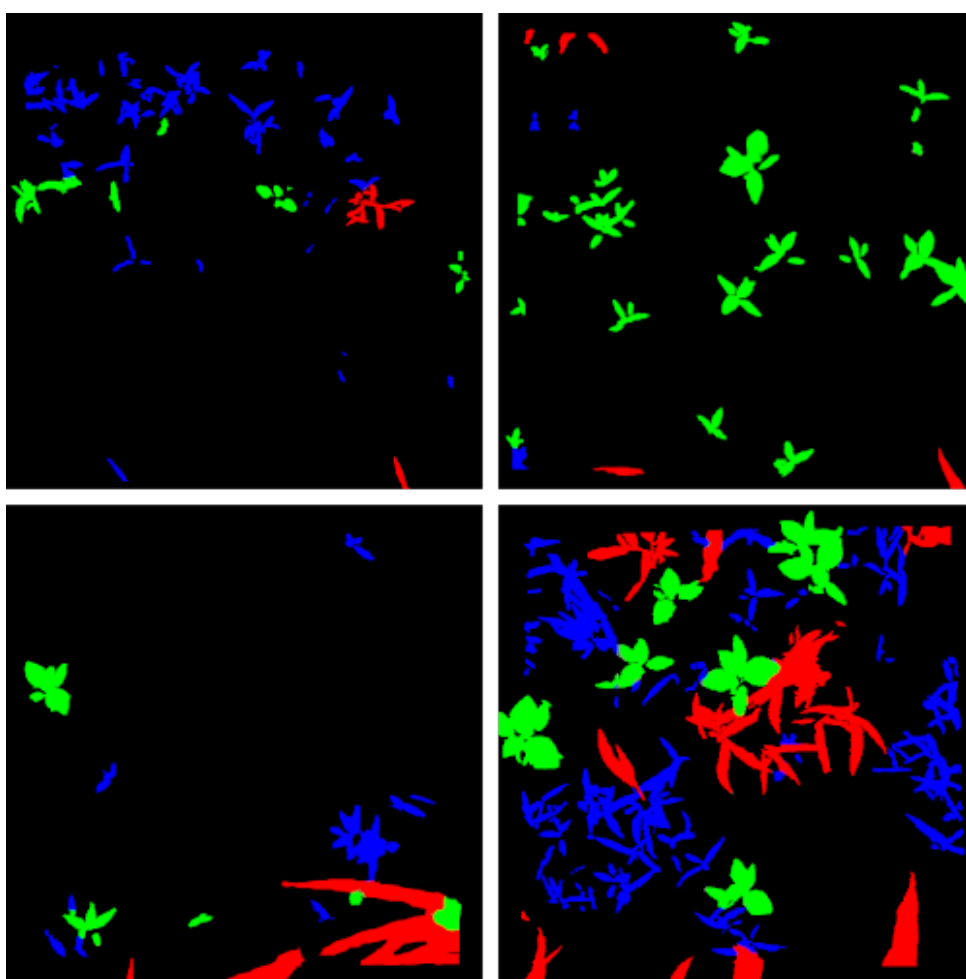
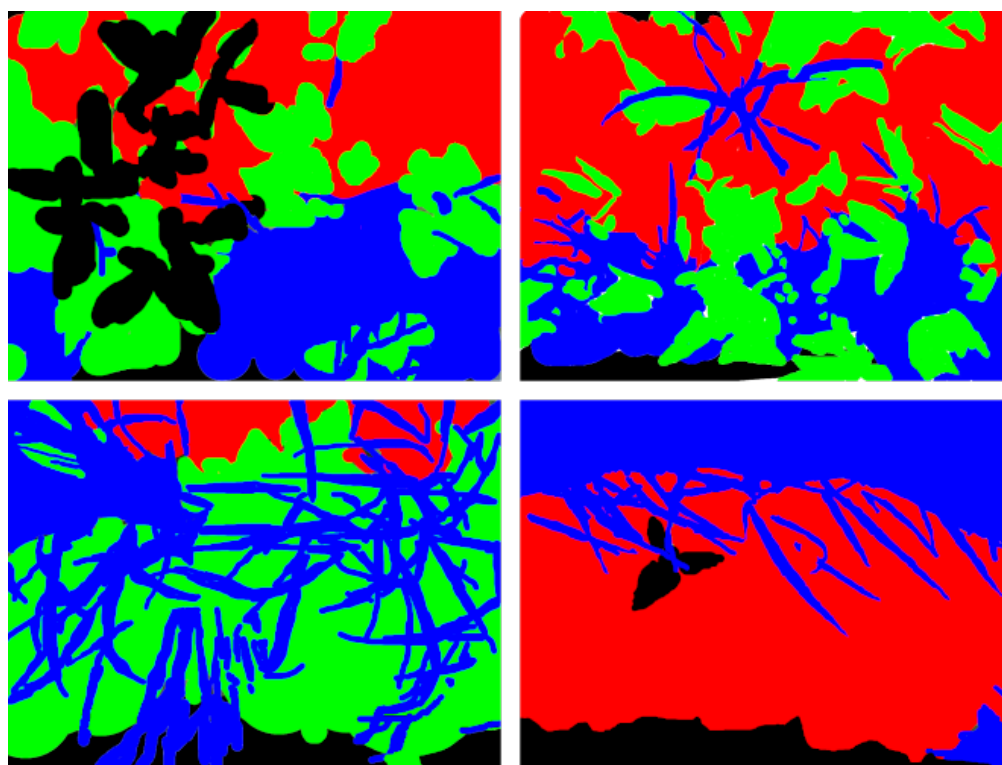


Figura 49 – Exemplo de imagens com anotação manual de classes do *Dataset 2*, com a mucuna em verde, o capim em azul e o amendoim em vermelha.



## 4.4 Resultados do algoritmo de visão estéreo

Depois de todas as imagens pareadas e retificadas, foi aplicado os dois algoritmos de visão estéreo anteriormente descritos. Para análise deles, como não se tem o valor-verdade das imagens capturadas do *Dataset 2*, foi avaliado a uniformidade e suavização do mapa de disparidade de forma visual, além da consideração do tempo de processamento. Para isso, será utilizado como referência o par de imagens retificadas da Figura 50. Cada imagem, após a retificação teve sua resolução alterada para 1118 x 909.

O primeiro algoritmo analisado, o de pareamento local de blocos, possui dois parâmetros que foram modificados de forma iterativa até que o resultado correspondesse ao observado nas imagens de entrada. Assim, foi definido um número máximo de 16 disparidades e que a janela de decisão tivesse tamanho 15 x 15. O resultado do método pode ser visto na Figura 51, em que se é apresentado o mapa de disparidade (valores mais próximos de zero indicam elementos mais distantes da câmera) e um preenchimento feito a partir de um filtro de suavização Gaussiano 11 x 11. Atenta-se que alterações no número de disparidades possíveis não influenciou significativamente no custo computacional, distintamente de quando os blocos têm tamanho maior. Na configuração final, o mapa demorou aproximadamente 6 segundos para ser processado.

O segundo algoritmo analisado foi o de pareamento semi-global, em que os parâmetros internos não foram alterados. Aplicando as mesmas imagens de entrada, o tempo de processamento foi disparatadamente maior, alcançando a marca de 905 segundos. Por outro lado, como se é possível analisar na esquerda da Figura 52, ele apresenta uma resposta bem mais contínua. Com o intuito de diminuir o custo computacional, as imagens de entrada tiveram sua resolução diminuída para 220 x 180 (mantendo a proporção dos dados), resultando no mapa de disparidade também apresentado na Figura 52, após voltar para a escala correta. Ele apresentou tempo menor de processamento, de 36 segundos, mas foi bem inferior aos mapas anteriores.

Logo, decidiu utilizar o primeiro algoritmo para as próximas etapas. É possível utilizar o valor do mapa como mais um dos vetores de características para a classificação, dado que as plantas das mesmas espécies tendem a terem ciclo de crescimento semelhante. Como o mapa é uma medida *pixel a pixel*, decidiu calcular a média dele como o valor definitivo do descritor na região de interesse.

## 4.5 Resultados do algoritmo de segmentação

O algoritmo de segmentação desenvolvido tem seu funcionamento descrito visualmente na Figura 53 e 54, com uma única imagem do *Dataset 1* e uma do *Dataset 2* utilizadas como exemplo. Escolheu este banco de imagens para iniciar a avaliação do método devido ao fato dele ter uma distinção maior entre planta e fundo, já que no *Dataset 2* a região de interesse cobre quase toda a imagem.

Figura 50 – Pares de imagens retificadas escolhidas para exemplificar a avaliação dos algoritmos de visão estéreo.

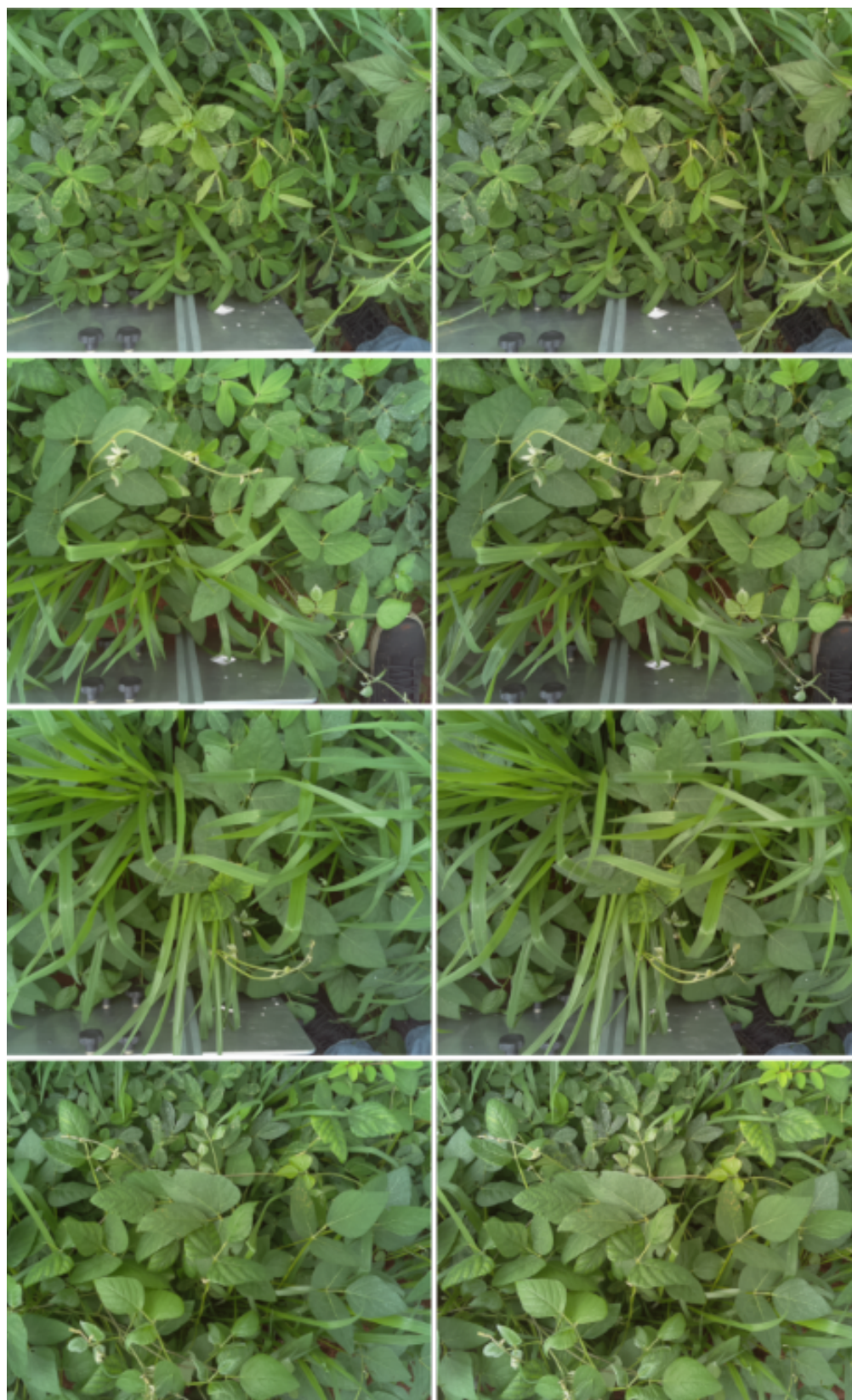


Figura 51 – Mapa de disparidade do primeiro algoritmo estéreo, pré e pós processado.

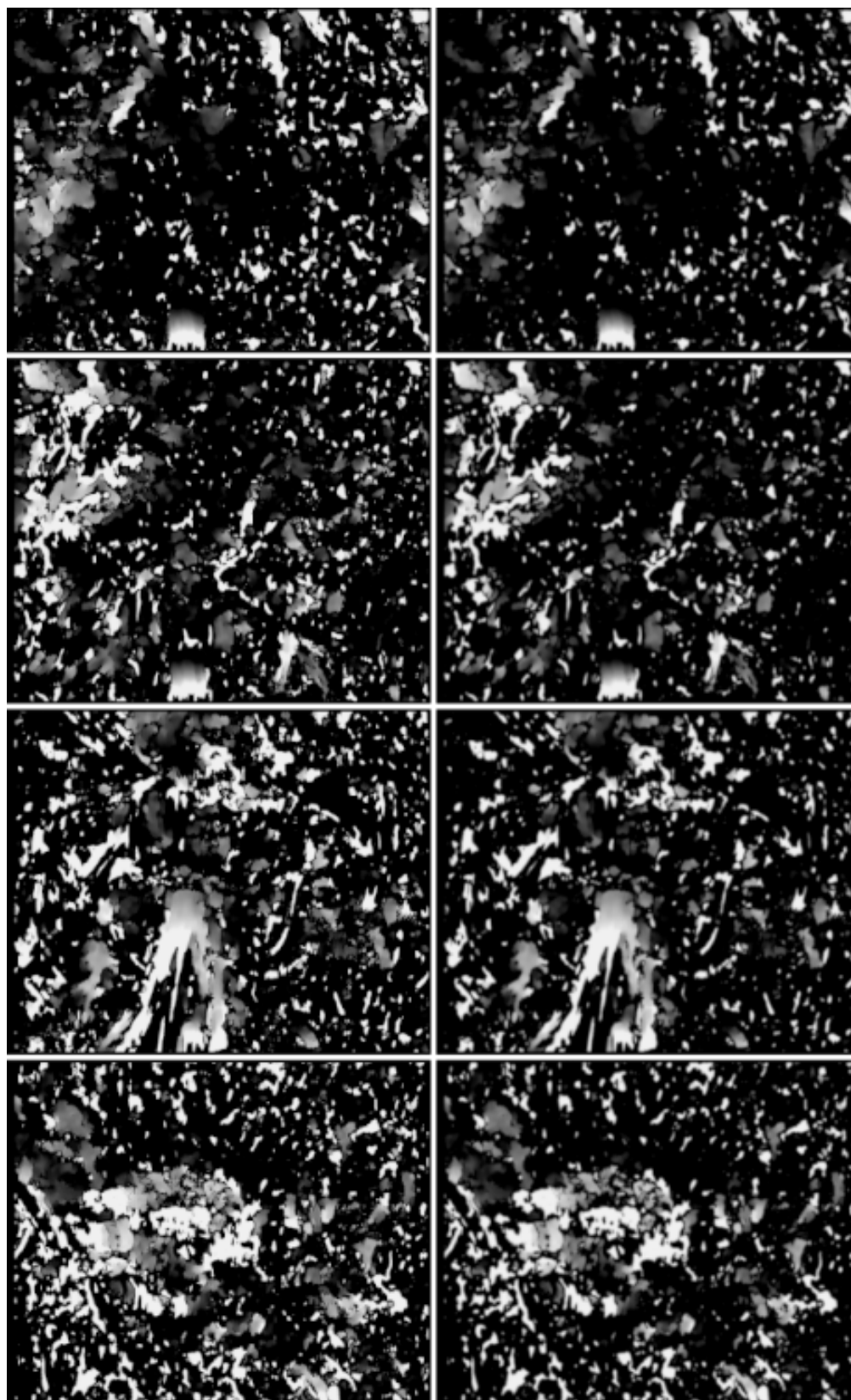


Figura 52 – Mapa de disparidade do segundo algoritmo estéreo, com diferentes resoluções de entrada.

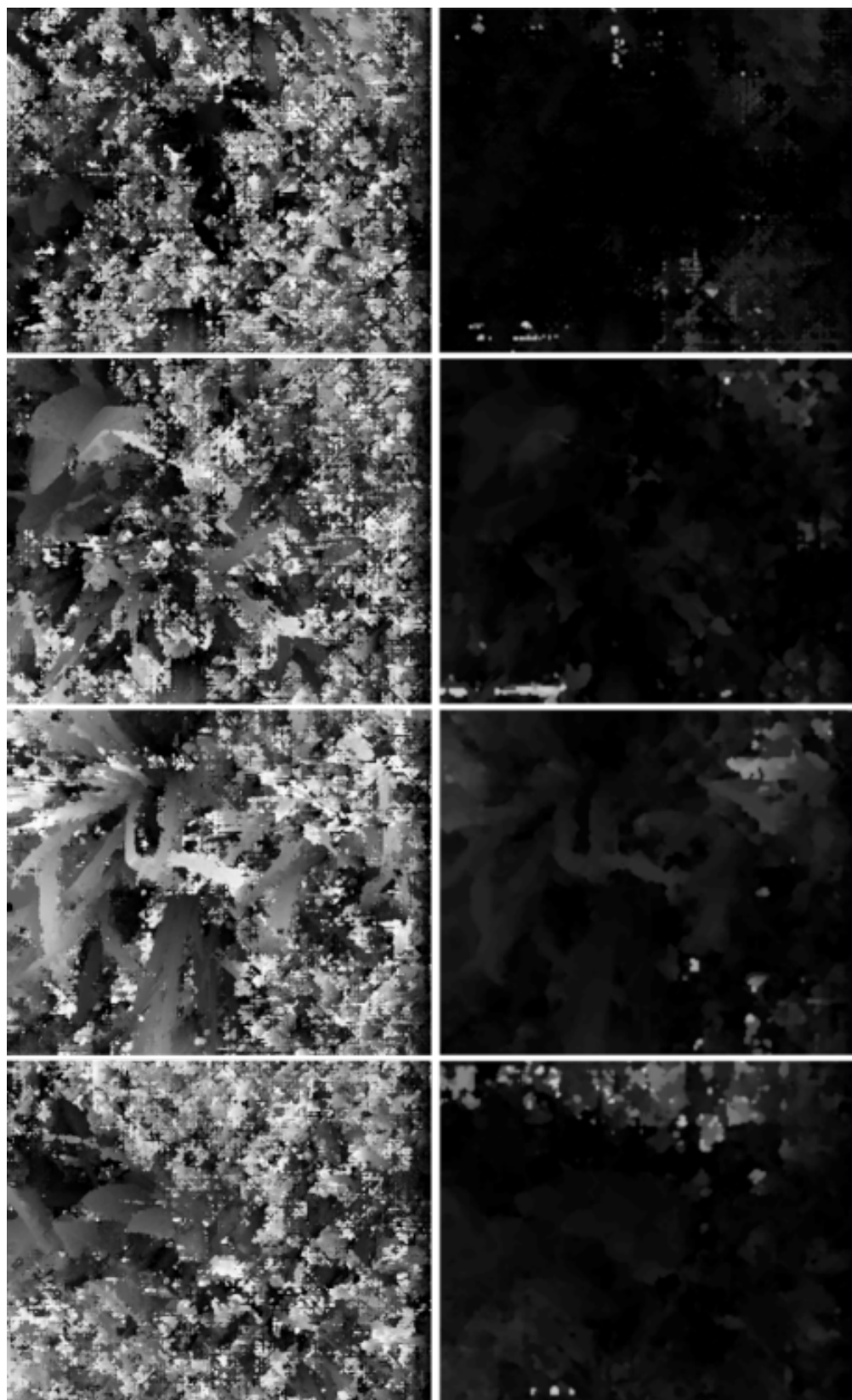




Figura 53 – Exemplar da segmentação do *Dataset 1*.

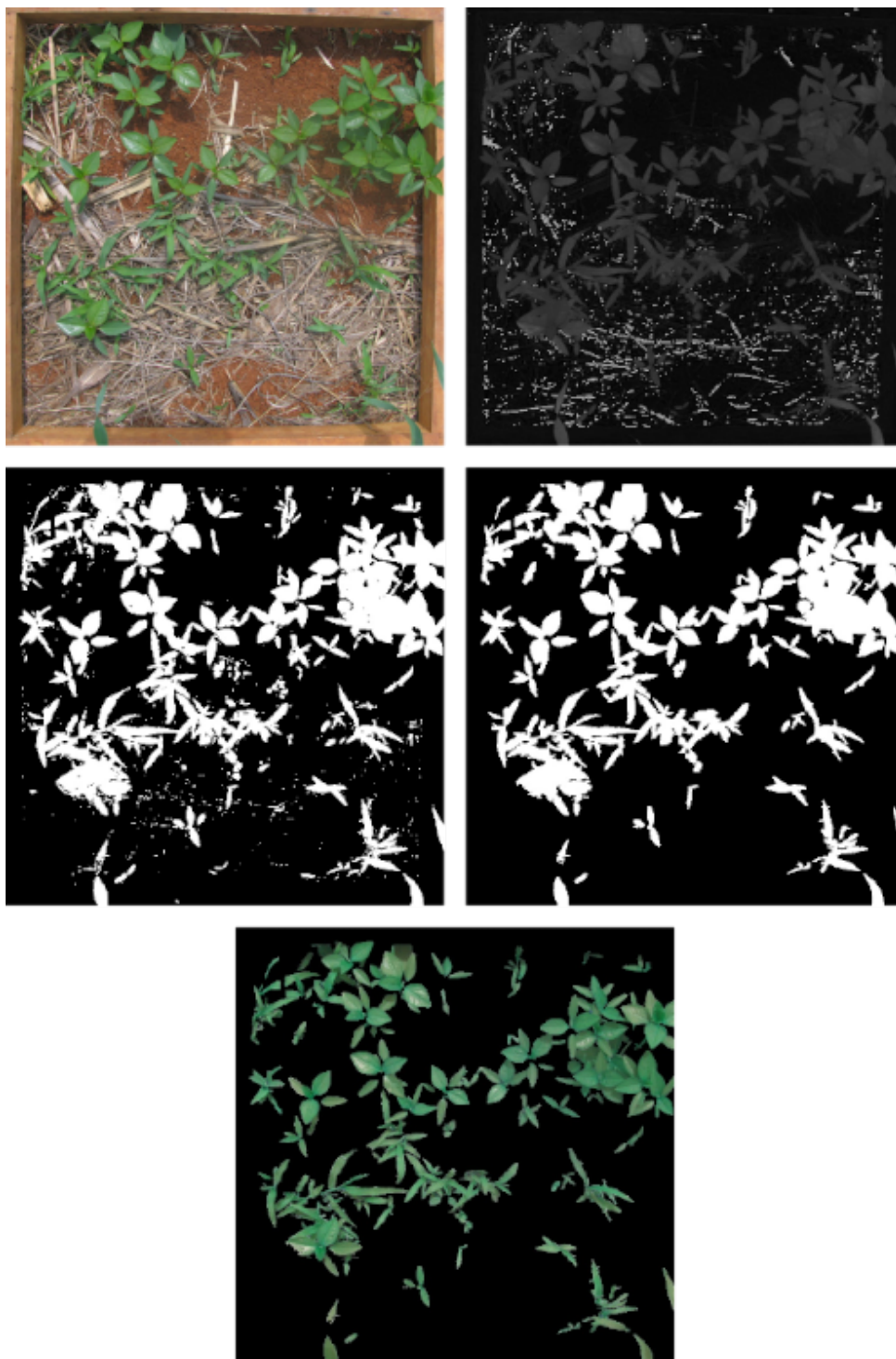
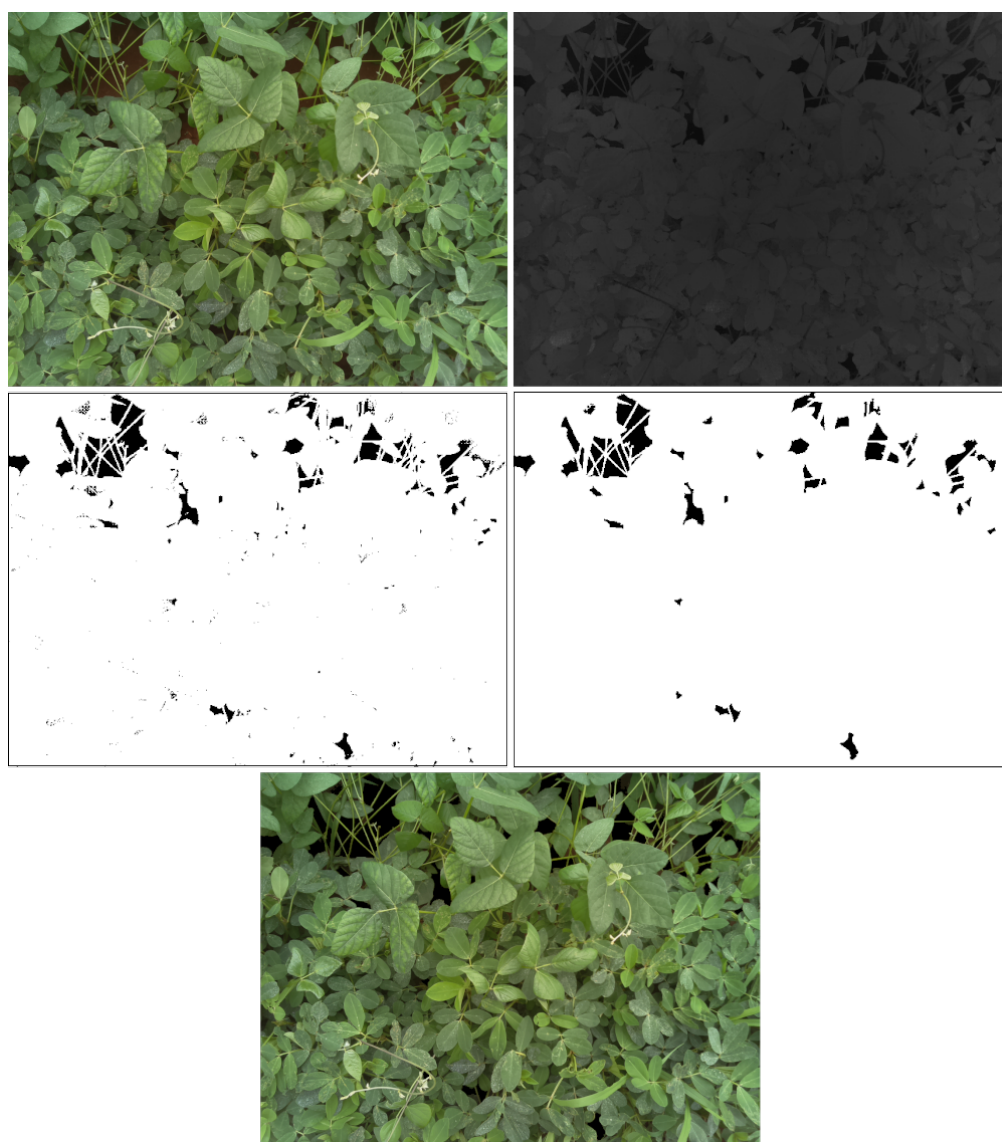


Figura 54 – Exemplar da segmentação do *Dataset 2*.



Tem-se então na figura a imagem original, o canal H do espaço de cores HSV, a limiarização e a remoção de buracos e objetos pequenos (a partir de operações morfológicas). A última imagem representa a aplicação da segmentação como uma máscara na imagem original. Para validar o algoritmo, 11 imagens, que correspondem a aproximadamente um terço do *dataset*, teve a segmentação esperada manualmente rotulada, gerando um conjunto de imagens de valor-verdade.

Quatro parâmetros foram então manipulados, a fim de encontrar a combinação que apresentasse melhor média das métricas de acurácia e de área de sobreposição. Esses parâmetros foram os dois valores que limitam o intervalo de valores de intensidade de H que são considerados pertencentes ao objeto (pode variar entre 0 e 180), o número de *pixels* mínimo que um objeto tem que ter e a área mínima que um buraco possa ter. Parte da iteração da variação dos parâmetros podem ser vistos na Tabela 9.

Tabela 9 – Variação dos parâmetros da segmentação.

Limiar inferior	Limiar superior	Área mínima do buraco	<i>pixels</i> mínimos do objeto	Acurácia (%)	Área de sobreposição (%)
10	80	30	100	99,71 ± 0,01	0,06 ± 0,07
20	80	30	100	94,79 ± 0,04	61,20 ± 11,62
30	80	30	100	82,86 ± 12,17	71,64 ± 11,33
<b>25</b>	80	30	100	82,77 ± 7,93	71,00 ± 10,51
25	<b>70</b>	30	100	88,75 ± 7,33	70,25 ± 10,22
25	60	30	100	82,94 ± 9,95	66,02 ± 10,96
25	70	100	100	88,96 ± 7,30	70,13 ± 10,22
25	70	20	100	88,69 ± 7,38	70,23 ± 10,23
25	70	<b>150</b>	100	89,06 ± 7,32	70,09 ± 10,20
25	70	200	100	89,06 ± 7,33	70,02 ± 10,26
25	70	150	50	91,89 ± 6,45	69,66 ± 11,99
25	70	150	200	84,33 ± 8,27	68,81 ± 8,66
25	70	150	<b>75</b>	90,56 ± 6,99	70,24 ± 11,31

Após os testes experimentais, definiu-se então o melhor valor de faixa de limiar, correspondente à cor verde, como entre 25 e 70, que selecionava mais plantas e folhas sem a maior parte do fundo. Apesar de que outros intervalos apresentarem métricas melhores, atenta-se que na base de dados não há imagens com paletas de cor muito variadas entre si, então se preferiu manter um intervalo de limiar curto.

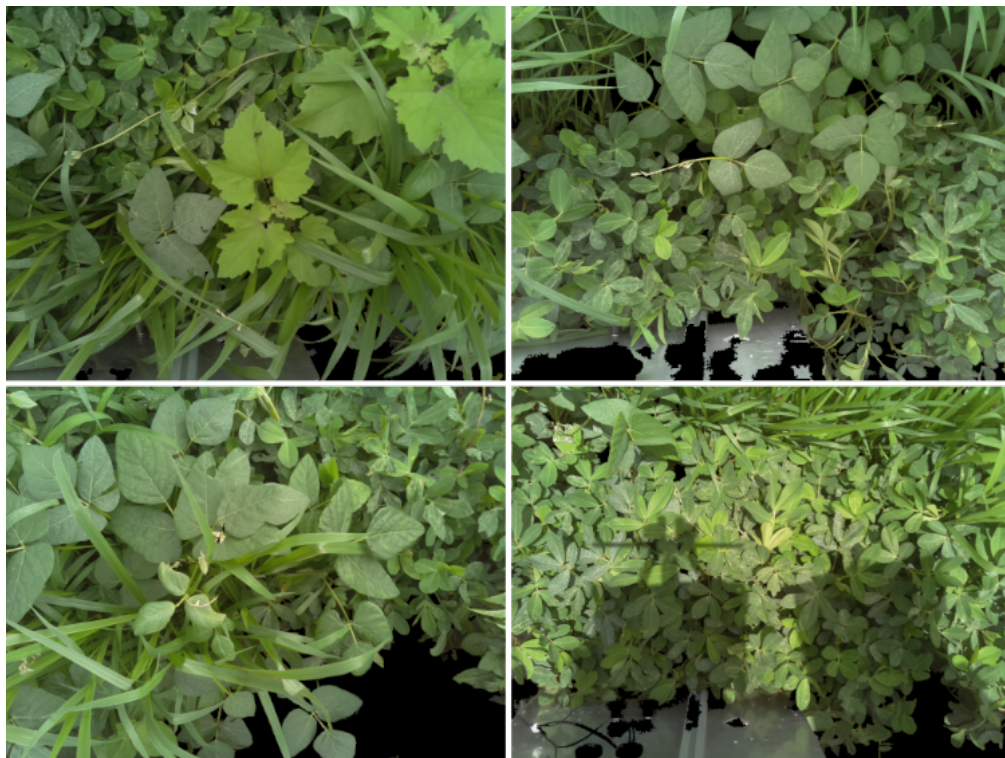
Como a maioria das plantas tem suas folhas próximas entre si, as regiões formadas pela segmentação apresentam poucos buracos internos, assim, percebeu-se que variando o tamanho máximo que um buraco poderia ter não influenciou tanto nas métricas, escolhendo então o valor de 150 *pixels* de área, momento em que variações do parâmetro não alteraram o resultado mais. Já para o tamanho mínimo de *pixels* que um objeto pode ter, foi definido que o valor igual a

75 apresentou melhores métricas, com acurácia acima de 90% e área de sobreposição acima de 70%.

Após esta etapa, foi aplicada a remoção de pequenos objetos e pequenos orifícios para reduzir o ruído. O resultado final da segmentação como um todo mostra que, embora tenhamos uma imagem limpa com a maioria das plantas selecionadas corretamente e com a forma inalterada, algumas folhas menores foram removidas como ruído enquanto outras foram desconectadas em diferentes partes ou reduzidas de tamanho. Neste ponto, podemos classificar a maioria dos *pixels* presentes na imagem corretamente como planta ou não.

Com estes parâmetros ajustados, utilizou o mesmo algoritmo na segmentação das imagens do *Dataset 2*, com alguns exemplares visíveis na Figura 55. Percebe-se que o algoritmo manteve o bom resultados na segmentação, mantendo corretamente as plantas na imagem e eliminando o solo, porém a estrutura cinza do sistema apresenta características reflexivas, o que a faz possuir alguns dos tons de verdes dentro do intervalo da limiarização. Como a estrutura é constante, mantendo-se sempre na mesma posição nas imagens, bastaria realizar o corte da imagem para eliminar ela, mas preferiu mantê-la com a confiança de que o classificador saberá distinguir o material das folhas.

Figura 55 – Exemplar da segmentação do *Dataset 2*.

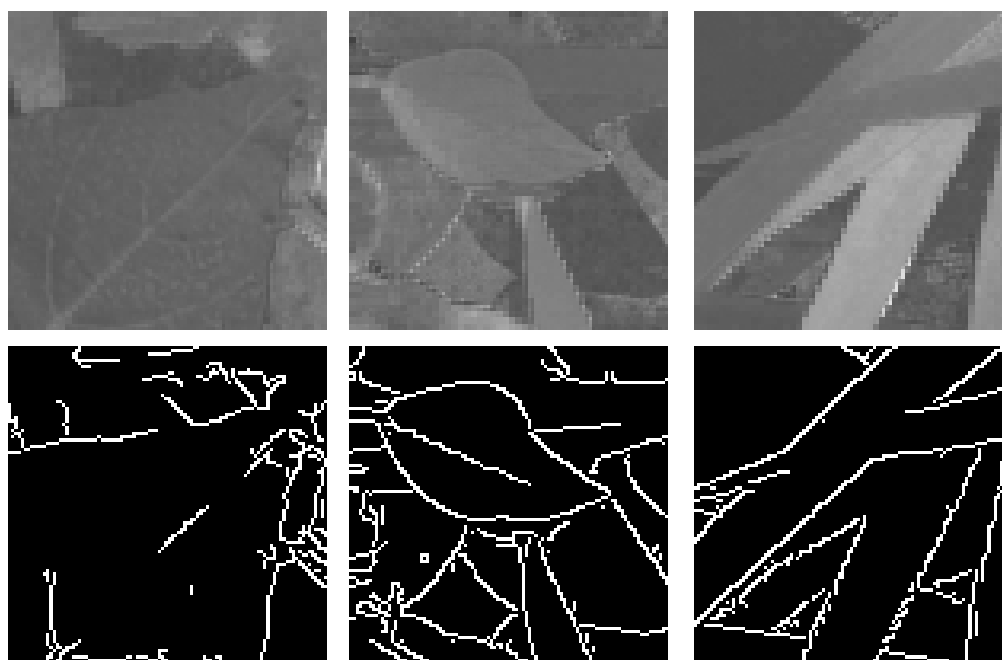


## 4.6 Resultado dos descritores

A imagem referencial do sistema estéreo foi convencionada como a imagem da direita, logo, apenas elas foram segmentadas e passaram pelo processo de extração de características.

Devido a resolução das imagens, definiu que elas seriam particionadas por uma janela quadrada de lado igual a 100 *pixels*, definido de forma experimental de tal modo que ao menos uma folha esteja totalmente contida na imagem. Para o *Dataset 1*, isso implica que cada uma das 34 imagens gera 36 amostras, totalizando em 1224 amostras para o treinamento do classificador SVM. Já para o *Dataset 2*, as dimensões das imagens não são múltiplas de 100. Assim, foi considerado para o treinamento do SVM um recorte de 1200 x 900 delas, não se valendo dos *pixels* mais à esquerda e ao fundo da imagem, região da imagem onde se encontra mais *pixels* referentes à estrutura do sistema. Para cada uma das 64 imagem então se terá 108 amostras, totalizando 6912 amostras para o treino do classificador. A Figura 56 apresenta exemplo de três amostras, com a janela superior indicando a imagem que foi usada para cálculo dos momentos de Haralick e as inferiores o resultado do algoritmo de Canny, com limiar inferior igual a 100 e superior igual a 200.

Figura 56 – Exemplar das janelas de decisão para cálculo dos descritores, com o canal H e o resultado do algoritmo de Canny.



Todas as amostras possuem vetores de características de tamanho 14 (os valores do histograma do LBP da borda e os momentos de Haralick) e foram rotulados manualmente, sendo que as amostras do *Dataset 1* foram rotulados em 3 classes e as do *Dataset 2* em quatro (uma para cada planta em análise e uma para o caso de não ser nenhuma delas).

O resultado dos descritores podem ser exemplificados na Tabela 10, em que há o retorno dos descritores para 10 amostras do *Dataset 1* (1-10) e mais 10 do *Dataset 2* (11-20), com os valores ainda não normalizados. Os oito primeiros valores no vetor do descritor se referem ao LBP e os outros cinco aos momentos de Haralick.

Tabela 10 – Exemplos do retorno dos descritores para 20 amostras.

Amostra	h0	h1	h2	h3	h4	h5	h6	h7	h8	Energia	Contraste	Correlação	Homogeneidade	Entropia
1	0	0	0	0	0	0	0	0	100	8,01E-1	3,37E2	8,72E-1	8,98E-1	1,70
2	0	0	0	0	0	0	0	0	100	7,37E-1	6,72E2	8,46E-1	8,63E-1	2,25
3	0	0	0	0	0	0	0	0	100	6,13E-1	7,15E2	8,74E-1	7,91E-1	3,43
4	1	1	0	0	0	0	0	0	98	6,00E-1	8,54E2	8,39E-1	7,84E-1	3,54
5	0	0	0	0	0	0	0	0	100	8,32E-1	4,17E2	8,25E-1	9,15E-1	1,43
6	0	0	0	0	0	0	0	0	100	1,00	0,00	1,00	1,00	0,00
7	0	2	0	0	0	0	0	0	98	9,57E-1	1,51E2	8,18E-1	9,79E-1	3,60E-1
8	1	4	0	0	0	0	0	0	93	8,84E-1	3,19E2	8,29E-1	9,42E-1	9,71E-1
9	0	8	0	0	0	0	0	0	84	5,82E-1	1,03e3	8,48E-1	7,70E-1	3,72
10	0	1	0	0	0	0	0	0	98	4,71E-4	1,09e3	5,34E-1	1,05E-1	1,20E1
11	2	4	0	0	0	0	0	0	94	1,08E-3	9,50E2	3,17E-1	1,19E-1	1,09E1
12	0	6	0	0	0	0	0	0	91	2,80E-4	9,58E2	8,40E-1	1,00E-1	1,28E1
13	2	4	0	0	0	0	0	0	92	3,03E-4	1,07E3	5,61E-1	7,20E-2	1,26E1
14	0	6	0	0	0	0	0	0	89	3,45E-4	1,50E3	5,98E-1	9,01E-2	1,26E1
15	1	2	0	0	0	0	0	0	96	2,72E-4	1,91E3	4,20E-1	9,13E-2	1,28E1
16	0	4	0	0	0	0	0	0	96	5,36E-4	1,58E3	5,75E-1	1,03E-1	1,27E1
17	0	2	0	0	0	0	0	0	98	4,90E-4	1,55E3	4,26E-1	1,11E-1	1,21E1
18	0	4	0	0	0	0	0	0	93	3,96E-4	1,03E3	5,17E-1	8,61E-2	1,22E1
19	1	3	0	0	0	0	0	0	95	3,71E-4	1,57E3	3,72E-1	9,83E-2	1,24E1
20	0	6	0	0	0	0	0	0	90	3,71E-4	2,16E3	2,79E-1	1,01E-1	1,24E1

## 4.7 Resultados dos classificadores

Com os descritores normalizados, os usou para treinar e testar os classificadores em cada *dataset*. Devido a natureza do problema, em que as plantas invasoras estão em ambiente natural e real de campo, há mais de uma planta por imagem e há sobreposição de folhas, é esperado que os classificadores apresentem resultados inferiores para o caso simples de uma folha escaneada por imagem. Foi avaliado e treinado os classificadores SVM para cada *dataset*, em dois cenários: um em que um classificador apenas (denominado classificador geral) identifica todas as espécies presentes de plantas invasoras; e outro em que há um classificador específico para cada espécie de interesse. Classificadores específicos apresentam melhores resultados ao comparar com o geral devido ao fato de especializarem e individualizarem o problema de agrupamento da amostra.

### 4.7.1 Dataset 1

#### 4.7.1.1 Classificador geral

Com os descritores normalizados, os usou para treinar e testar os classificadores. Para o *Dataset 1*, foram treinados por validação cruzada classificadores de *kernel* linear com  $C \in [1; 10; 100; 1000]$ , *kernel* RBF com  $C \in [1; 10; 100; 1000]$  e  $\gamma \in [1E-4; 1E-3; 0,01; 0,1; 0,2; 0,5]$  e *kernel* sigmoide com  $C \in [1; 10; 100; 1000]$  e  $\gamma \in [1E-4; 1E-3; 0,01; 0,1; 0,2; 0,5]$  e  $c_{f0} \in [-1, 0, 1]$ . Os resultados da validação cruzada podem ser conferidos na Tabela 7411 e 12. O melhor classificador então foi o linear com  $C=1000$ , resultando em 65,7% de acurácia.

Para analisar melhor essa configuração do SVM, foi então calculado a matriz de confusão, as métricas associadas e a curva ROC. As amostras foram rotuladas em três classes, em que a classe 0 refere ao caso de não ter plantas invasoras na região, a 1 de ser capim-carrapicho e 2 de ser a leiteira. Uma amostra aleatória de 80% foi sorteada para o treino e outra de 20% para teste, obtendo-se uma acurácia de 67,8%. A Tabela 13 apresenta o as métricas para as classes individuais, enquanto que a matriz de confusão resultante pode ser visualizada na Figura 57.

As curvas ROC, com a medida de AUC, para cada classe podem ser visualizadas nas Figuras 58 e 59. A Figura 60 apresenta a curva ROC do classificador geral, a partir da micro-média das curvas obtidas de cada classe (incluindo a classe da hipótese nula).

#### 4.7.1.2 Classificador específico: capim-carrapicho

Pelo método de validação cruzada, o classificador com maior acurácia foi o de *kernel* RBF ( $C = 1$  e  $\gamma = 0,5$ ), com acurácia de 78,8%. Os resultados de precisão e sensibilidade do classificador podem ser vistos na Tabela 14 e a matriz de confusão na Figura 61, em que a classe 1 se refere ao capim-carrapicho e a classe 0 ao caso em que não é. A Figura 62 apresenta a curva ROC deste classificador, com AUC igual a 0,85. Todos os valores obtidos pelo classificador específico foram superiores ao comparar com o classificador geral.

Tabela 11 – Resultado da validação cruzada no *Dataset 1* (anotado em negrito o escolhido) do *kernel* RBF e Linear.

Acurácia	Kernel	Parâmetros do classificador
0,650 ( $\pm 0,027$ )	RBF	C=1, $\gamma=0,0001$
0,645 ( $\pm 0,016$ )	RBF	C=1, $\gamma=0,001$
0,629 ( $\pm 0,032$ )	RBF	C=1, $\gamma=0,01$
0,649 ( $\pm 0,059$ )	RBF	C=1, $\gamma=0,1$
0,632 ( $\pm 0,046$ )	RBF	C=1, $\gamma=0,2$
0,634 ( $\pm 0,060$ )	RBF	C=1, $\gamma=0,5$
0,647 ( $\pm 0,022$ )	RBF	C=10, $\gamma=0,0001$
0,625 ( $\pm 0,015$ )	RBF	C=10, $\gamma=0,001$
0,614 ( $\pm 0,057$ )	RBF	C=10, $\gamma=0,01$
0,639 ( $\pm 0,047$ )	RBF	C=10, $\gamma=0,1$
0,638 ( $\pm 0,052$ )	RBF	C=10, $\gamma=0,2$
0,629 ( $\pm 0,051$ )	RBF	C=10, $\gamma=0,5$
0,640 ( $\pm 0,032$ )	RBF	C=100, $\gamma=0,0001$
0,603 ( $\pm 0,039$ )	RBF	C=100, $\gamma=0,001$
0,597 ( $\pm 0,059$ )	RBF	C=100, $\gamma=0,01$
0,641 ( $\pm 0,047$ )	RBF	C=100, $\gamma=0,1$
0,638 ( $\pm 0,054$ )	RBF	C=100, $\gamma=0,2$
0,629 ( $\pm 0,054$ )	RBF	C=100, $\gamma=0,5$
0,641 ( $\pm 0,020$ )	RBF	C=1000, $\gamma=0,0001$
0,618 ( $\pm 0,033$ )	RBF	C=1000, $\gamma=0,001$
0,605 ( $\pm 0,050$ )	RBF	C=1000, $\gamma=0,01$
0,639 ( $\pm 0,047$ )	RBF	C=1000, $\gamma=0,1$
0,643 ( $\pm 0,060$ )	RBF	C=1000, $\gamma=0,2$
0,629 ( $\pm 0,054$ )	RBF	C=1000, $\gamma=0,5$
0,633 ( $\pm 0,026$ )	Linear	C=1
0,656 ( $\pm 0,012$ )	Linear	C=10
0,656 ( $\pm 0,038$ )	Linear	C=100
<b>0,657 (<math>\pm 0,029</math>)</b>	Linear	C=1000

#### 4.7.1.3 Classificador específico: leiteira

Repetindo o mesmo procedimento para a planta invasora de folhas largas, pelo método de validação cruzada, o classificador com maior acurácia foi o com *kernel* sigmoide ( $C = 1$ ,  $c_{f0} = -1$  e  $\gamma = 0,001$ ), com acurácia de 85,6%. Os resultados de precisão e sensibilidade do classificador podem ser vistos na Tabela 15 e a matriz de confusão na Figura 63, em que a classe 1 se refere à leiteira e a classe 0 ao caso em que não é. Percebe-se que apesar de uma alta acurácia, a precisão e a sensibilidade do classificador são muito baixas, devido ao fato de que a distribuição entre as classes é muito desigual, já que o número de amostras pertencentes à classe 0 é aproximadamente 8 vezes maior do que o número de amostras na classe 1. A Figura 64 apresenta a curva ROC deste classificador, com AUC igual a 0,71.

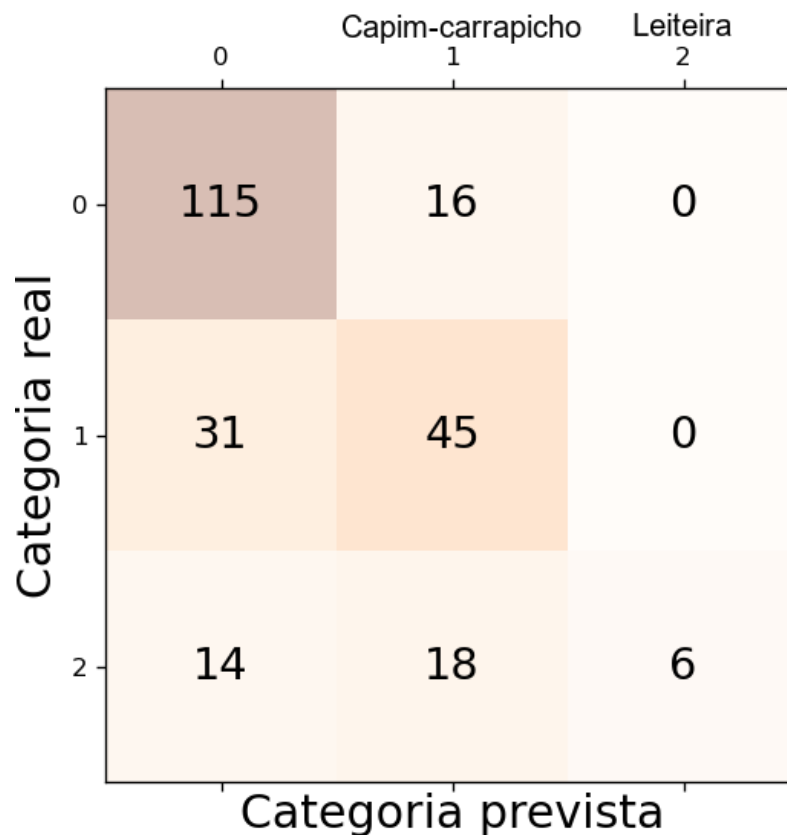


Tabela 12 – Resultado da validação cruzada no *Dataset 1* do kernel Sigmoide.

Acurácia	C	$c_{f0}$	$\gamma$	Acurácia	C	$c_{f0}$	$\gamma$
0,567 ( $\pm 0,045$ )	1	-1	1e-05	0,569 ( $\pm 0,043$ )	100	-1	1e-05
0,559 ( $\pm 0,078$ )	1	-1	0,0001	0,548 ( $\pm 0,078$ )	100	-1	0,0001
0,537 ( $\pm 0,004$ )	1	-1	0,001	0,537 ( $\pm 0,004$ )	100	-1	0,001
0,537 ( $\pm 0,004$ )	1	-1	0,01	0,537 ( $\pm 0,004$ )	100	-1	0,01
0,537 ( $\pm 0,004$ )	1	-1	0,1	0,537 ( $\pm 0,004$ )	100	-1	0,1
0,537 ( $\pm 0,004$ )	1	-1	0,2	0,537 ( $\pm 0,004$ )	100	-1	0,2
0,537 ( $\pm 0,004$ )	1	-1	0,5	0,537 ( $\pm 0,004$ )	100	-1	0,5
0,562 ( $\pm 0,057$ )	1	0	1e-05	0,559 ( $\pm 0,056$ )	100	0	1e-05
0,559 ( $\pm 0,100$ )	1	0	0,0001	0,552 ( $\pm 0,080$ )	100	0	0,0001
0,537 ( $\pm 0,004$ )	1	0	0,001	0,537 ( $\pm 0,004$ )	100	0	0,001
0,537 ( $\pm 0,004$ )	1	0	0,01	0,537 ( $\pm 0,004$ )	100	0	0,01
0,537 ( $\pm 0,004$ )	1	0	0,1	0,537 ( $\pm 0,004$ )	100	0	0,1
0,537 ( $\pm 0,004$ )	1	0	0,2	0,537 ( $\pm 0,004$ )	100	0	0,2
0,537 ( $\pm 0,004$ )	1	0	0,5	0,537 ( $\pm 0,004$ )	100	0	0,5
0,561 ( $\pm 0,067$ )	1	1	1e-05	0,556 ( $\pm 0,054$ )	100	1	1e-05
0,634 ( $\pm 0,032$ )	1	1	0,0001	0,553 ( $\pm 0,080$ )	100	1	0,0001
0,537 ( $\pm 0,004$ )	1	1	0,001	0,537 ( $\pm 0,004$ )	100	1	0,001
0,537 ( $\pm 0,004$ )	1	1	0,01	0,537 ( $\pm 0,004$ )	100	1	0,01
0,537 ( $\pm 0,004$ )	1	1	0,1	0,537 ( $\pm 0,004$ )	100	1	0,1
0,537 ( $\pm 0,004$ )	1	1	0,2	0,537 ( $\pm 0,004$ )	100	1	0,2
0,537 ( $\pm 0,004$ )	1	1	0,5	0,537 ( $\pm 0,004$ )	100	1	0,5
0,568 ( $\pm 0,041$ )	10	-1	1e-05	0,569 ( $\pm 0,043$ )	1000	-1	1e-05
0,550 ( $\pm 0,080$ )	10	-1	0,0001	0,548 ( $\pm 0,078$ )	1000	-1	0,0001
0,537 ( $\pm 0,004$ )	10	-1	0,001	0,539 ( $\pm 0,004$ )	1000	-1	0,001
0,537 ( $\pm 0,004$ )	10	-1	0,01	0,537 ( $\pm 0,004$ )	1000	-1	0,01
0,537 ( $\pm 0,004$ )	10	-1	0,1	0,537 ( $\pm 0,004$ )	1000	-1	0,1
0,537 ( $\pm 0,004$ )	10	-1	0,2	0,537 ( $\pm 0,004$ )	1000	-1	0,2
0,537 ( $\pm 0,004$ )	10	-1	0,5	0,537 ( $\pm 0,004$ )	1000	-1	0,5
0,559 ( $\pm 0,056$ )	10	0	1e-05	0,559 ( $\pm 0,056$ )	1000	0	1e-05
0,558 ( $\pm 0,074$ )	10	0	0,0001	0,549 ( $\pm 0,080$ )	1000	0	0,0001
0,537 ( $\pm 0,004$ )	10	0	0,001	0,537 ( $\pm 0,004$ )	1000	0	0,001
0,537 ( $\pm 0,004$ )	10	0	0,01	0,537 ( $\pm 0,004$ )	1000	0	0,01
0,537 ( $\pm 0,004$ )	10	0	0,1	0,537 ( $\pm 0,004$ )	1000	0	0,1
0,537 ( $\pm 0,004$ )	10	0	0,2	0,537 ( $\pm 0,004$ )	1000	0	0,2
0,537 ( $\pm 0,004$ )	10	0	0,5	0,537 ( $\pm 0,004$ )	1000	0	0,5
0,556 ( $\pm 0,053$ )	10	1	1e-05	0,556 ( $\pm 0,054$ )	1000	1	1e-05
0,559 ( $\pm 0,099$ )	10	1	0,0001	0,549 ( $\pm 0,080$ )	1000	1	0,0001
0,537 ( $\pm 0,004$ )	10	1	0,001	0,537 ( $\pm 0,004$ )	1000	1	0,001
0,537 ( $\pm 0,004$ )	10	1	0,01	0,537 ( $\pm 0,004$ )	1000	1	0,01
0,537 ( $\pm 0,004$ )	10	1	0,1	0,537 ( $\pm 0,004$ )	1000	1	0,1
0,537 ( $\pm 0,004$ )	10	1	0,2	0,537 ( $\pm 0,004$ )	1000	1	0,2
0,537 ( $\pm 0,004$ )	10	1	0,5	0,537 ( $\pm 0,004$ )	1000	1	0,5

Tabela 13 – Resultado do classificador SVM Linear C=1000: *Dataset 1*.

	Precisão	Sensibilidade	F-score	Amostras
Classe 0	0,72	0,88	0,79	131
Classe 1	0,57	0,59	0,58	76
Classe 2	1,00	0,16	0,27	38

Figura 57 – Matriz de confusão do SVM Linear C=1000: *Dataset 1*.

## 4.7.2 Dataset 2

### 4.7.2.1 Classificador geral

Para o *Dataset 2*, que possui muito mais amostras que o anterior, se utilizou o classificador linear, que apresentou melhor resultado. Observou-se que o tempo de treinamento do classificador linear, comparado com os outros *kernels*, foi muito alto. As amostras foram rotuladas em quatro classes, em que a classe 0 refere ao caso de não ter plantas invasoras na região, a 1 de ser braquiária, 2 de ser a mucuna-preta e a 3 a de ser amendoim. Calculando então um classificador SVM linear com C=1000, com 80% dos dados para treinamento e 20% para teste, obteve-se uma acurácia de 64,7%. A Tabela 16 apresenta o as métricas para as classes individuais, enquanto que a matriz de confusão resultante pode ser visualizada na Figura 65.

As curvas ROC, com as medidas de AUC, para cada classe podem ser visualizadas nas Figuras 66, 67 e 68. A Figura 69 apresenta a curva ROC do classificador geral, a partir da

Figura 58 – Curva ROC: capim-carrapicho.

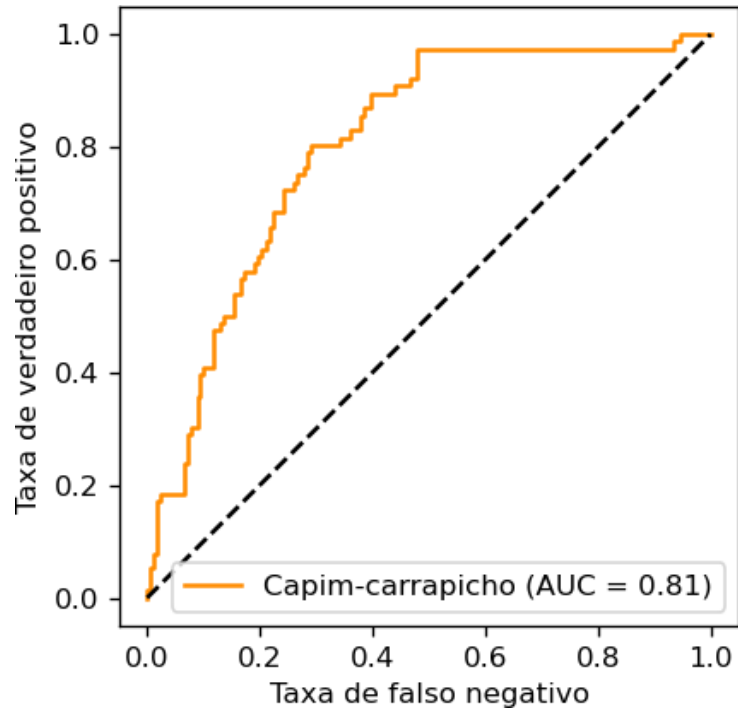
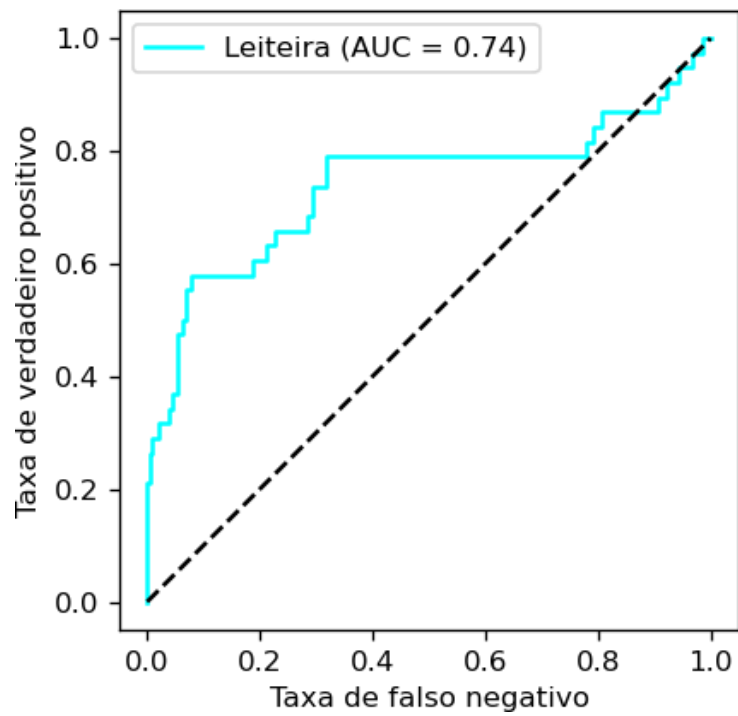


Figura 59 – Curva ROC: leiteira.



micro-média das curvas obtidas de cada classe (incluindo a classe da hipótese nula).

Analisando as curvas ROC obtidas pelos classificadores gerais quando aplicados nos dois *datasets*, é possível perceber que o classificador obteve um bom resultado, pois a AUC em todos os casos foi maior do que 0,5. No geral, considerando todas as classes, para o *Dataset 1* apresentou AUC de 0,84, enquanto o *Dataset 2* apresentou AUC de 0,88. É possível também perceber, a partir das matrizes de confusão, que os classificadores para ambos os casos acabaram classificando de forma discrepante as classes que possuíam menor exemplares na amostra, indicando um leve viés do método.

#### 4.7.2.2 Classificador específico: capim braquiária

Analisando todos os *kernels* e considerando o tempo de processamento e a acurácia dos classificadores, o que apresentou melhor resultado foi o classificador SVM de *kernel* RBF ( $C = 1000$  e  $\gamma = 0,01$ ), com acurácia de 79,2%. Os resultados de precisão e sensibilidade do classificador podem ser vistos na Tabela 17 e a matriz de confusão na Figura 70, em que a classe 1 se refere ao capim braquiária e a classe 0 ao caso em que não é. A Figura 71 apresenta a curva ROC deste classificador, com AUC igual a 0,76. Como esperado, o classificador específico foi superior ao classificador geral.

#### 4.7.2.3 Classificador específico: mucuna-preta

Para o caso de um classificador específico apenas para a mucuna-preta, o classificador que apresentou melhor resultado foi igualmente o classificador SVM de *kernel* RBF ( $C = 1000$  e  $\gamma = 0,01$ ), com acurácia de 81,1%. Os resultados de precisão e sensibilidade do classificador podem ser vistos na Tabela 18 e a matriz de confusão na Figura 72, em que a classe 1 se refere à mucuna-preta e a classe 0 ao caso em que não é. A Figura 73 apresenta a curva ROC deste classificador, com AUC igual a 0,85.

## 4.8 Cálculo de volume

Para o cálculo de volume, foi usado exclusivamente o *Dataset 2*, tendo em vista que o mesmo tem imagens estereoscópicas, apresentando todas as informações necessárias. A partir da Equação 3.10, do mapa de disparidade, da resolução de captura das imagens de 1280 x 960 *pixels*, do valor de distância de base de 6 cm, da altura do sistema fixa de 1 m do solo e dos parâmetros intrínsecos das câmeras, foi possível obter a altura dos objetos presentes na imagem. Ao calcular, as plantas invasoras não tiveram altura maior do que 50 cm, o que corresponde com o que foi observado durante o experimento de campo. A Figura 74 apresenta como exemplo a altura para dois blocos da plantação.

Como cada *pixel* teve área igual a  $6,2 \times 10^{-7} \text{ m}^2$  (a partir da área do bloco obtida pela Equação 3.1 e pela resolução da imagem), a área de cada janela de 100 x 100 definida para a

extração de características e classificação foi de  $6,2 \times 10^{-3} \text{ m}^2$ . Utilizando o modelo de cone para cálculo do volume da planta, a altura da janela de decisão foi definida como a média da altura dos objetos presentes e o volume então de cada janela foi dado por como o terço do produto desta média pela área da janela.

Analisando a área de cultivo do *Dataset 2* (Figura 43) e utilizando os classificadores específicos da mucuna-preta e do capim braquiária descritos anteriormente, foi possível então gerar um mapa com a taxa de ocupação das plantas invasoras. Cada imagem do *dataset* representa um bloco da área de cultivo e após todas passarem pelos classificadores, elas são corretamente agrupadas e orientadas em uma imagem só, que se refere ao campo georreferenciado total. O mapa de taxa de ocupação da mucuna-preta é apresentado na Figura 75, com uma escala em pseudo-cor verde de altura, em que quanto mais claro mais alta é a planta. O cálculo do volume total para a área de  $72 \text{ m}^2$  resultou em  $0,494 \text{ m}^3$ . Entretanto, para o controle de calda em sítio específico, pode-se separar a área de cultivo na dimensão da grade de amostragem, ou ainda em fatias de profundidade. Considerando o volume total para a área de análise, os mapas de ocupação podem ser visualizados, por exemplo, considerando três intervalos de altura: plantas de até 15 cm (Figura 76); plantas entre 15 cm e 30 cm (Figura 77); e plantas acima dos 30 cm (Figura 78).

Respectivamente e considerando toda a área de  $72 \text{ m}^2$ , para o capim-braquiária o mapa de taxa de ocupação pode ser visto Figura 79, com uma escala em pseudo-cor azul de altura, em que quanto mais claro mais alta é a planta. O cálculo do volume total resultou em  $0,144 \text{ m}^3$ . De modo similar, os mapas de ocupação para intervalos de altura podem ser visualizados nas Figuras 80, 81 e 82, que apresentam o mapa de plantas de até 15 cm, entre 15 cm e 30 cm e acima dos 30 cm, respectivamente. Percebe-se que o volume da mucuna foi maior do que a da braquiária na área selecionada.

Para o controle de calda em sítio específico na dimensão da grade de amostragem, é possível calcular o volume das plantas invasoras por bloco, e analisar fileira por fileira. Como exemplo, a Tabela 19 apresenta o volume das duas plantas invasoras em todos os blocos pertencentes à primeira fileira (blocos 1 ao 8).

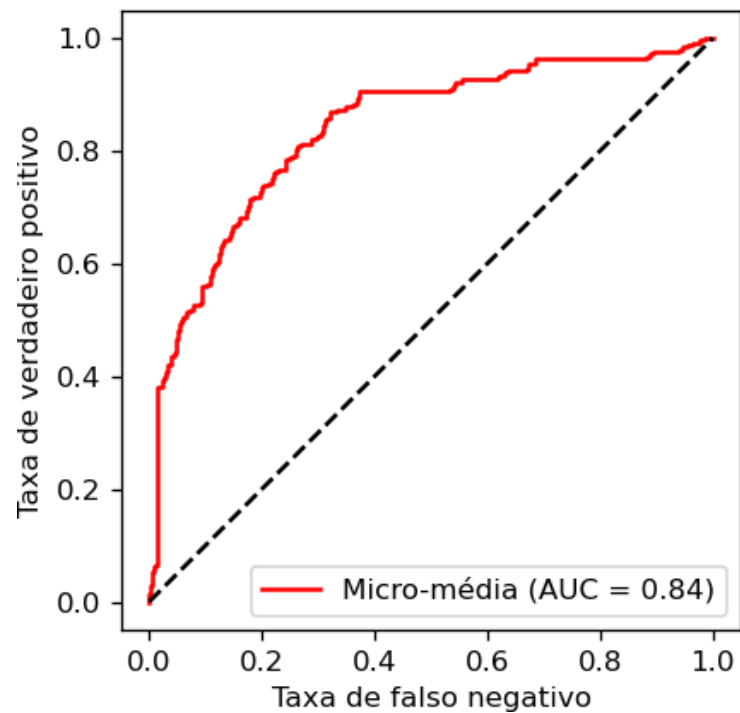
Figura 60 – Curva ROC: micro-média das curvas de cada classe do *Dataset 1*.

Tabela 14 – Resultado do classificador de capim-carrapicho.

	Precisão	Sensibilidade	F-score	Amostras
Classe 0	0,83	0,75	0,79	135
Classe 1	0,72	0,81	0,76	110

Tabela 15 – Resultado do classificador de leiteira.

	Precisão	Sensibilidade	F-score	Amostras
Classe 0	0,89	1,00	0,94	218
Classe 1	0,00	0,00	0,00	27

Tabela 16 – Resultado do classificador SVM Linear C=1000: *Dataset 2*.

	Precisão	Sensibilidade	F-score	Número de Amostras
Classe 0	0,56	0,61	0,58	38
Classe 1	0,54	0,25	0,34	313
Classe 2	0,66	0,56	0,60	349
Classe 3	0,66	0,88	0,76	683

Tabela 17 – Resultado do classificador de braquiária.

	Precisão	Sensibilidade	F-score	Amostras
Classe 0	0,80	0,98	0,88	1070
Classe 1	0,72	0,14	0,23	313

Figura 61 – Matriz de confusão do classificador de capim-carrapicho.

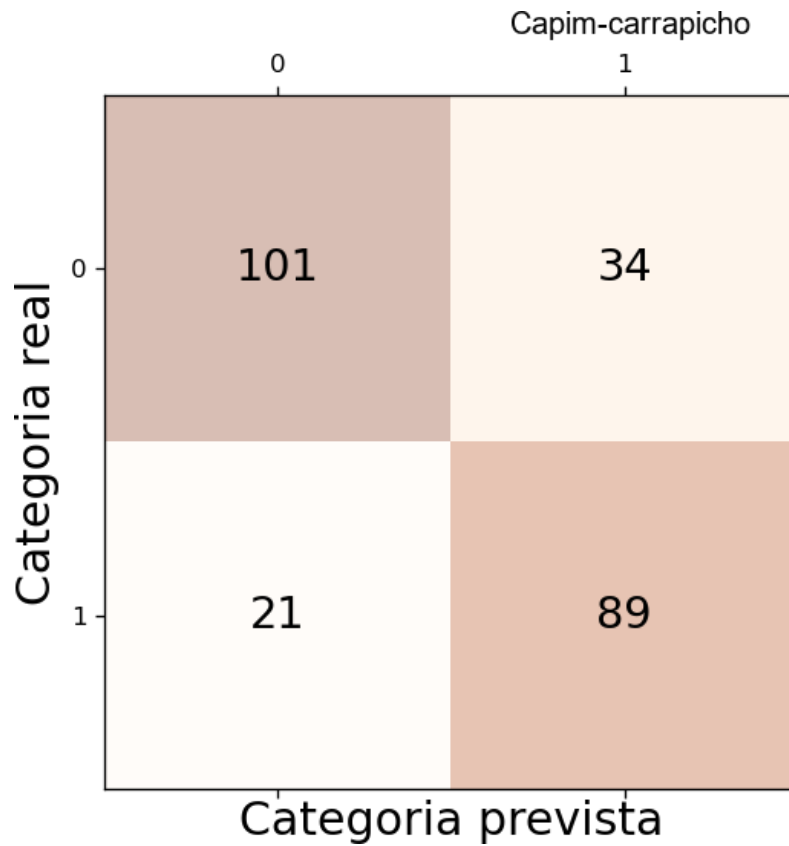


Figura 62 – Curva ROC classificador específico: capim-carrapicho.

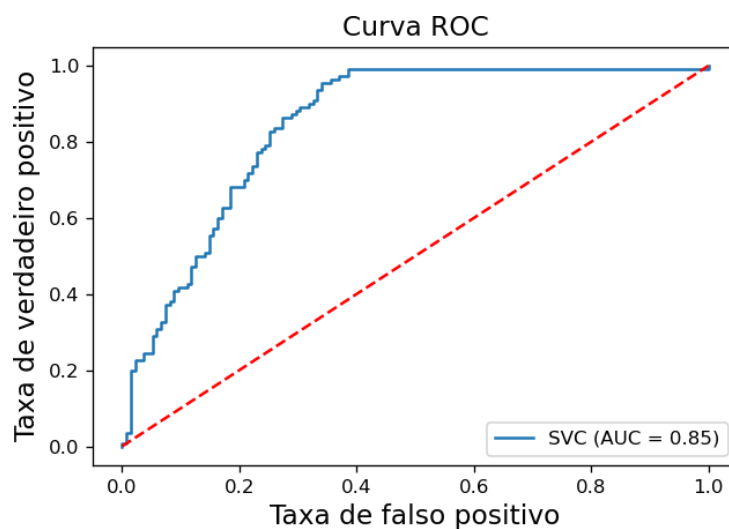


Tabela 18 – Resultado do classificador de mucuna-preta.

	Precisão	Sensibilidade	F-score	Amostras
Classe 0	0,83	0,97	0,89	1034
Classe 1	0,80	0,41	0,54	349

Figura 63 – Matriz de confusão do classificador de leiteira.

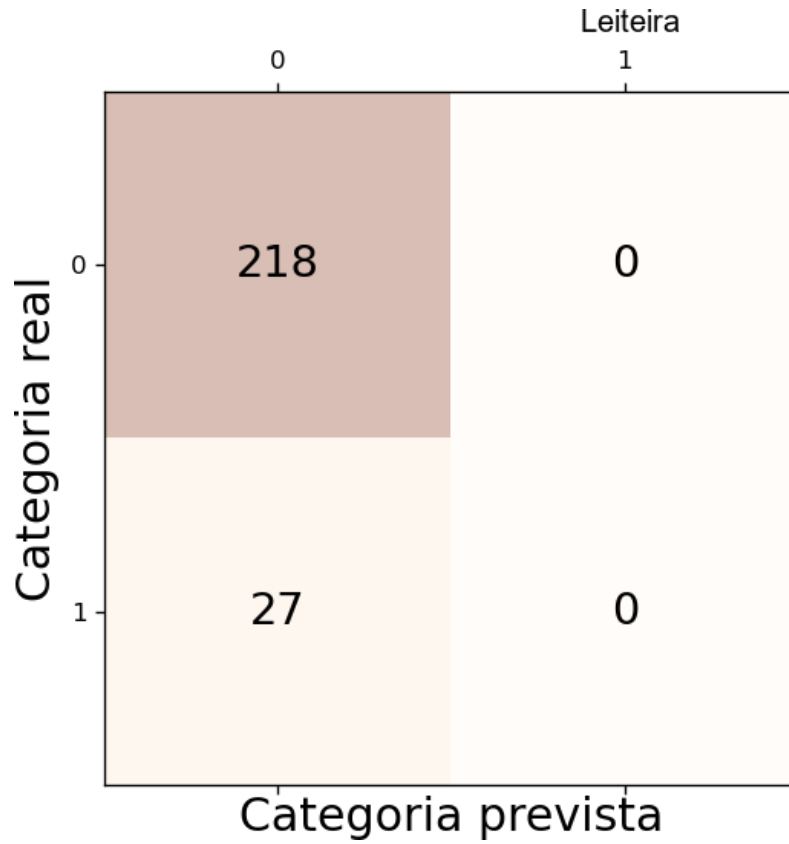


Figura 64 – Curva ROC classificador específico: leiteira.

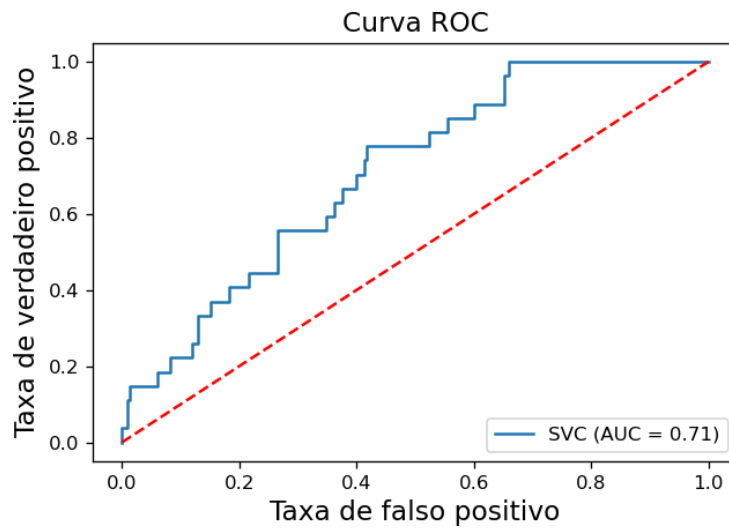


Tabela 19 – Volume das plantas invasoras na 1ª fileira do campo experimental.

Bloco	1	2	3	4	5	6	7	8
Volume mucuna (m <sup>3</sup> )	0,012	0,013	0,016	0,014	0,007	0,016	0,010	0,015
Volume braquiária (m <sup>3</sup> )	0,003	0,010	0,006	0,007	0,009	0,002	0,001	0,001



Figura 65 – Matriz de confusão do SVM Linear C=1000: Dataset 2.

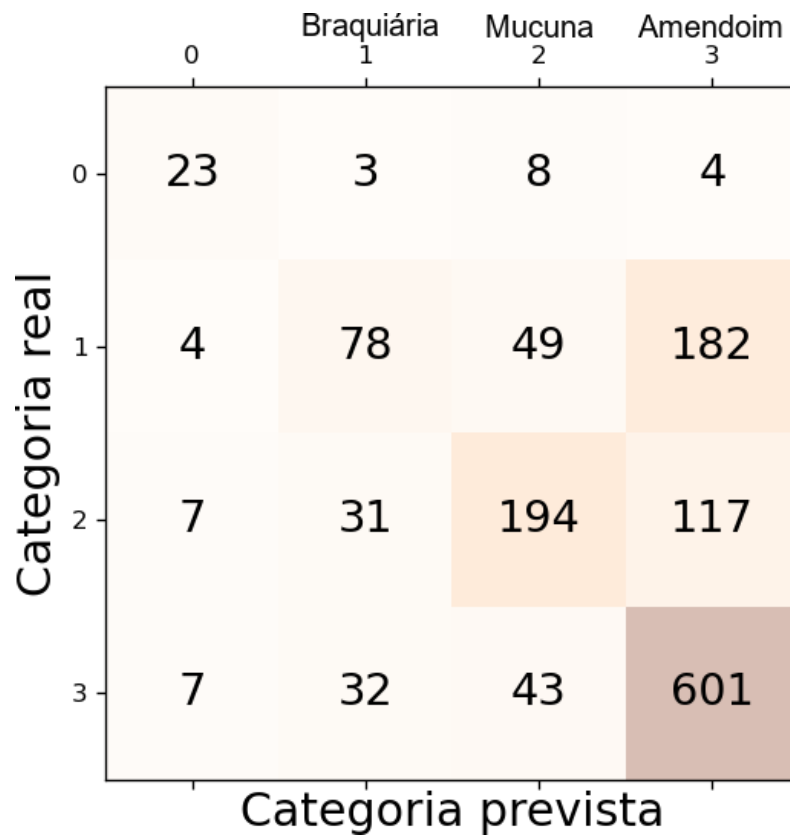


Figura 66 – Curva ROC: braquiária.

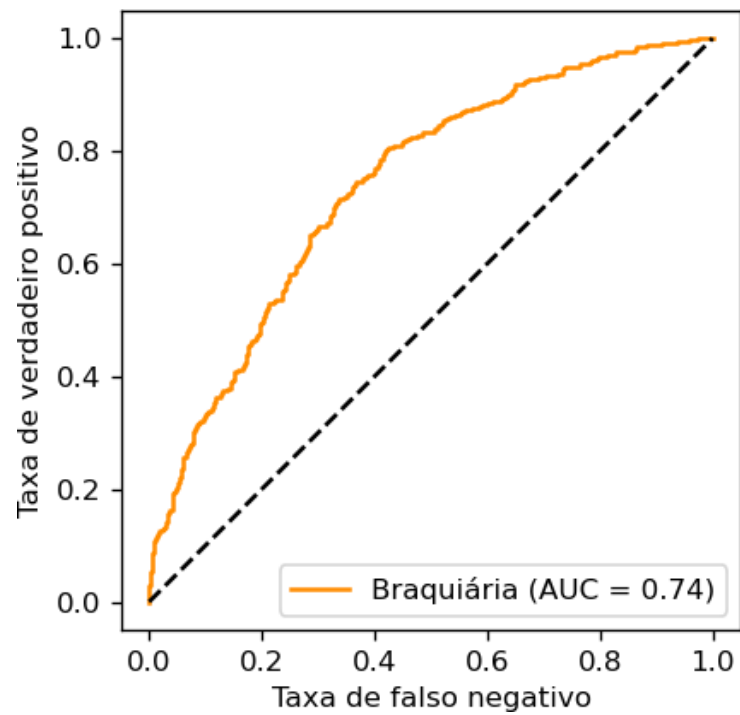


Figura 67 – Curva ROC: mucuna-preta.

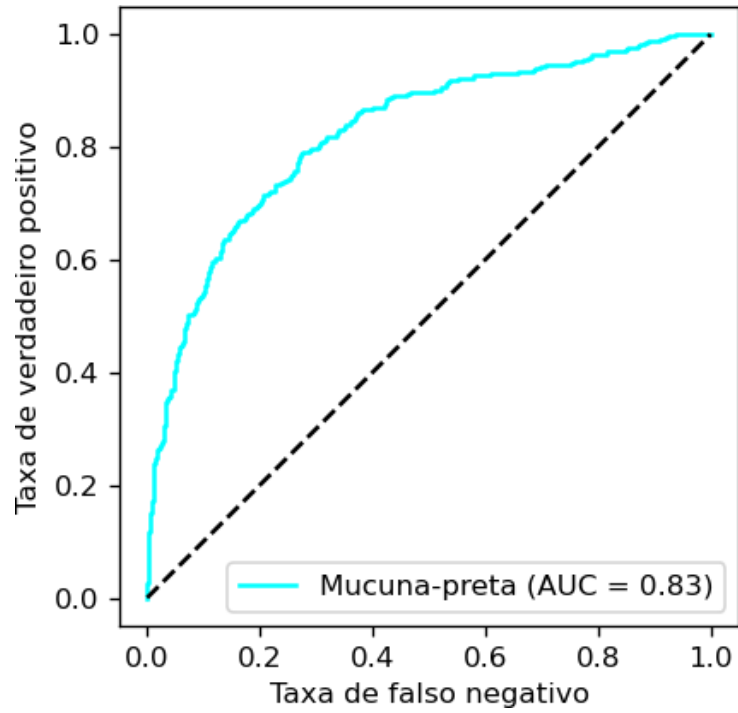


Figura 68 – Curva ROC: amendoim.

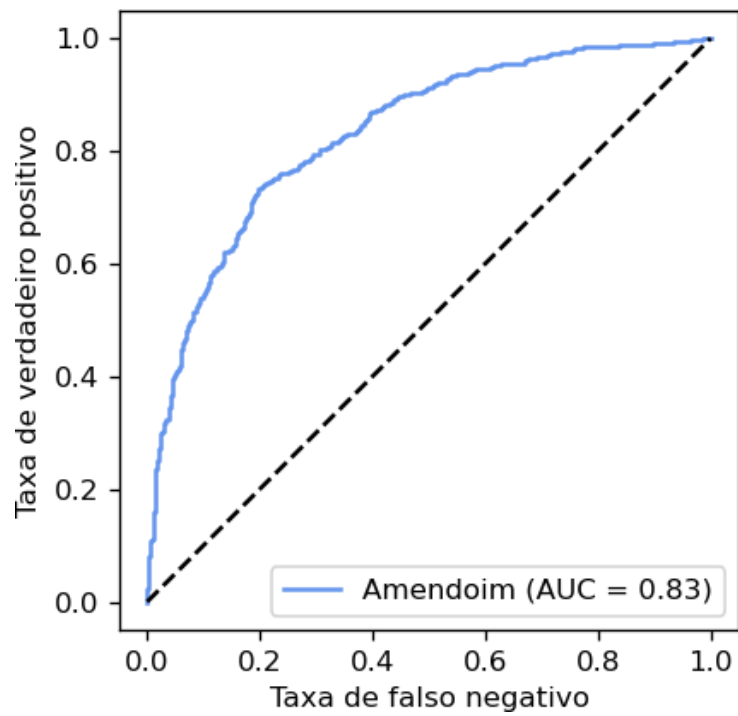


Figura 69 – Curva ROC: micro-média das curvas de cada classe do *Dataset 2*.

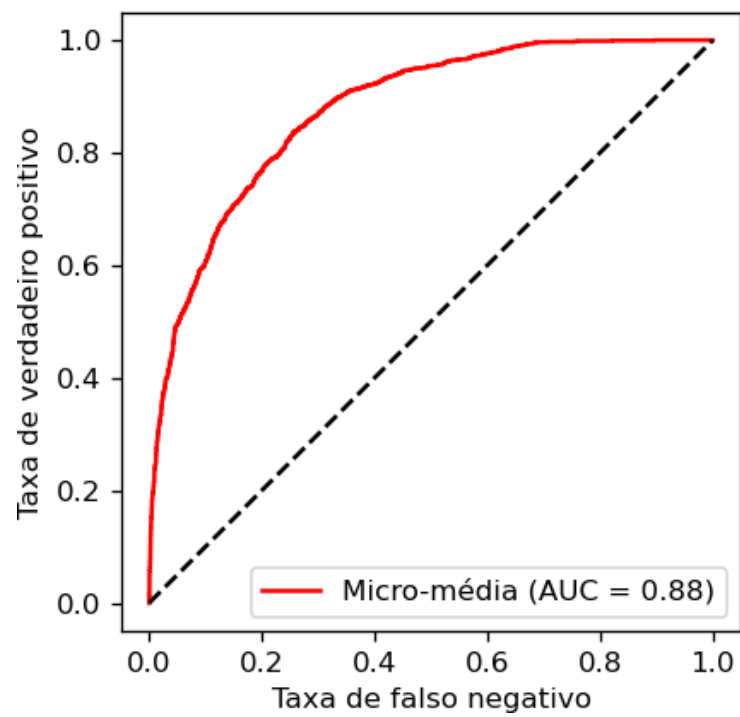


Figura 70 – Matriz de confusão do classificador de braquiária.

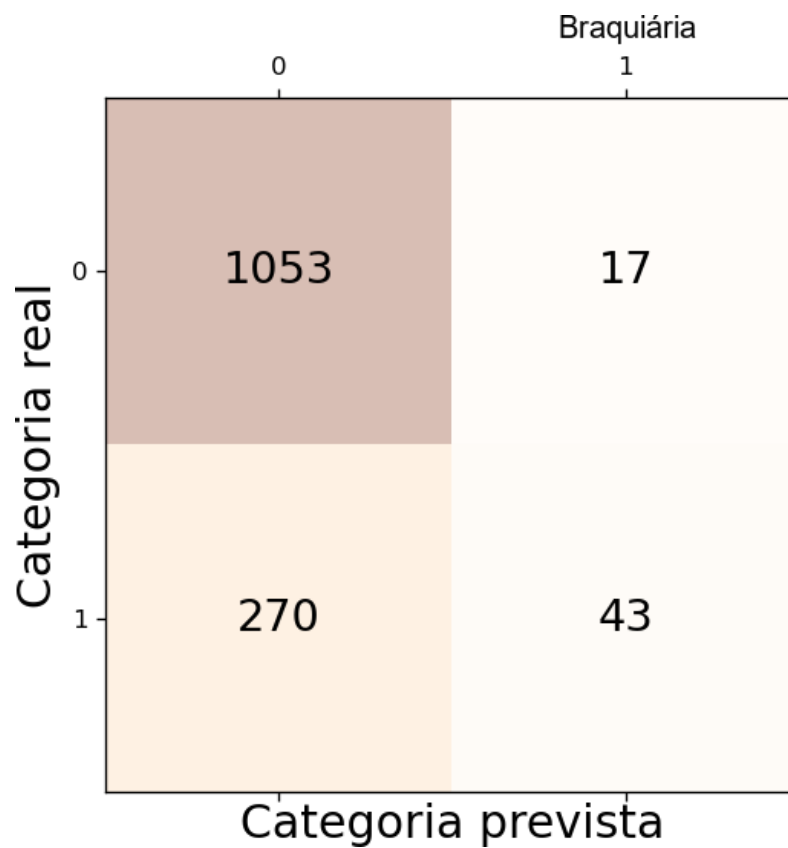


Figura 71 – Curva ROC classificador específico: braquiária.

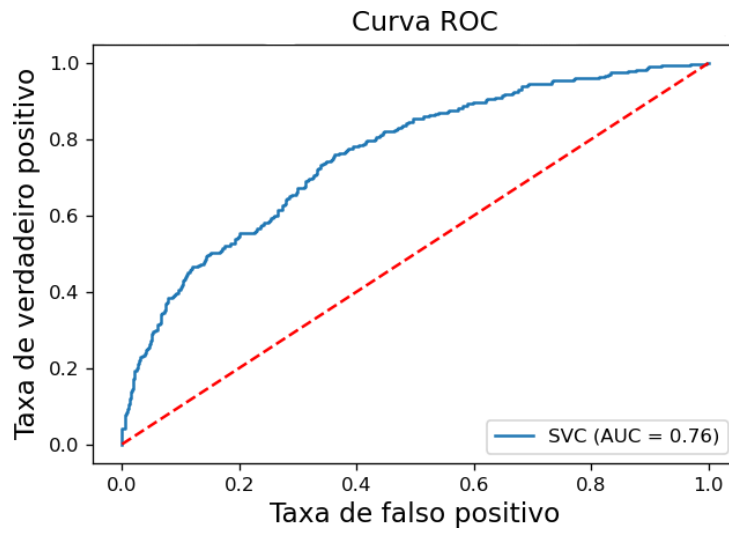


Figura 72 – Matriz de confusão do classificador de mucuna-preta.

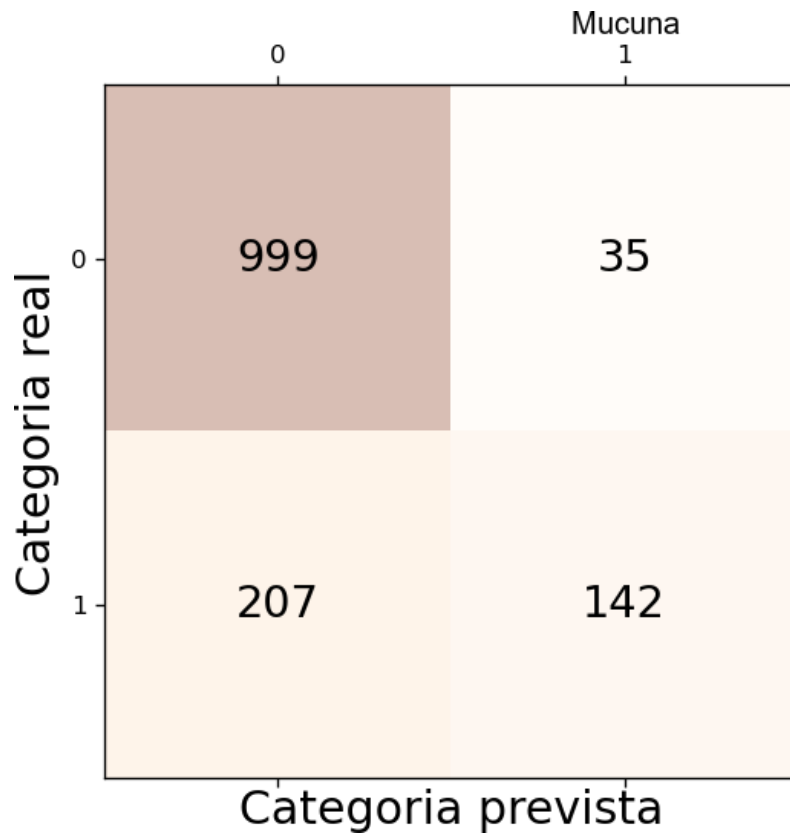


Figura 73 – Curva ROC classificador específico: mucuna-preta.

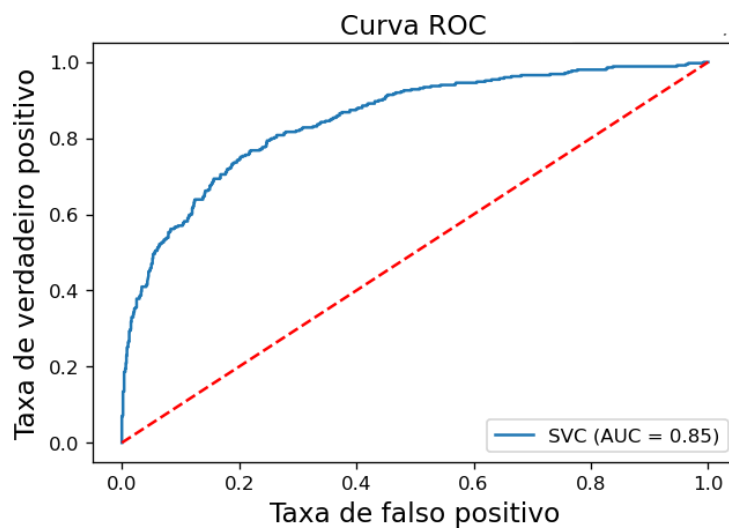


Figura 74 – Exemplos de obtenção da altura das plantas.

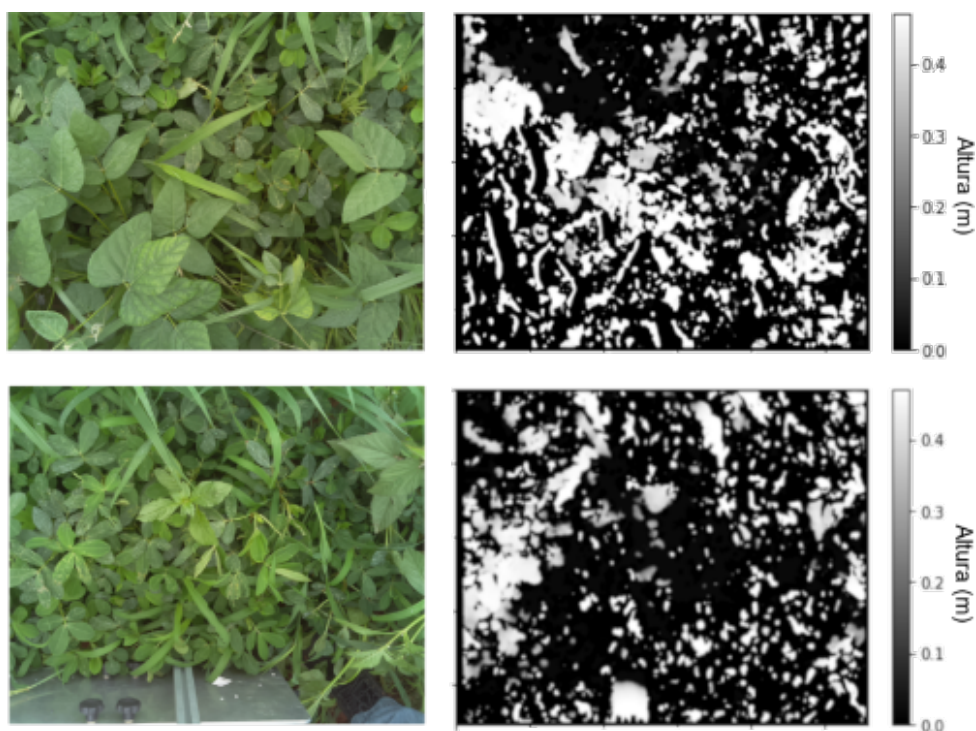


Figura 75 – Mapa taxa de ocupação mucuna-preta.

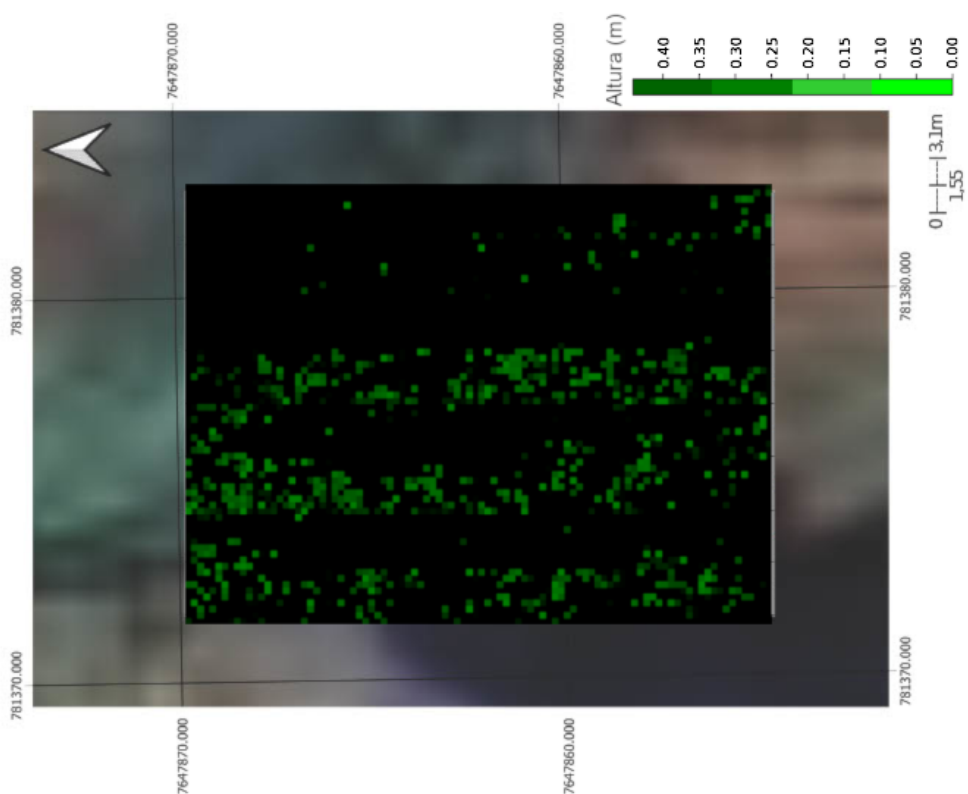


Figura 76 – Mapa taxa de ocupação mucuna-preta: fatia de profundidade de até 15 cm.

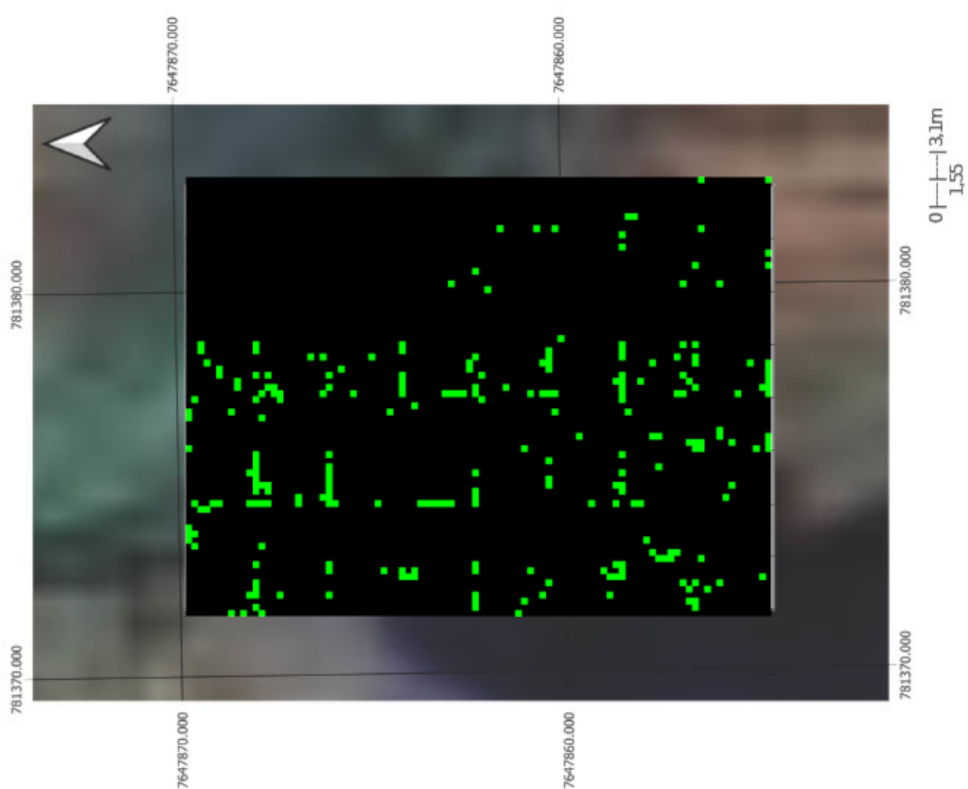


Figura 77 – Mapa taxa de ocupação mucuna-preta: fatia de profundidade entre 15 e 30 cm.

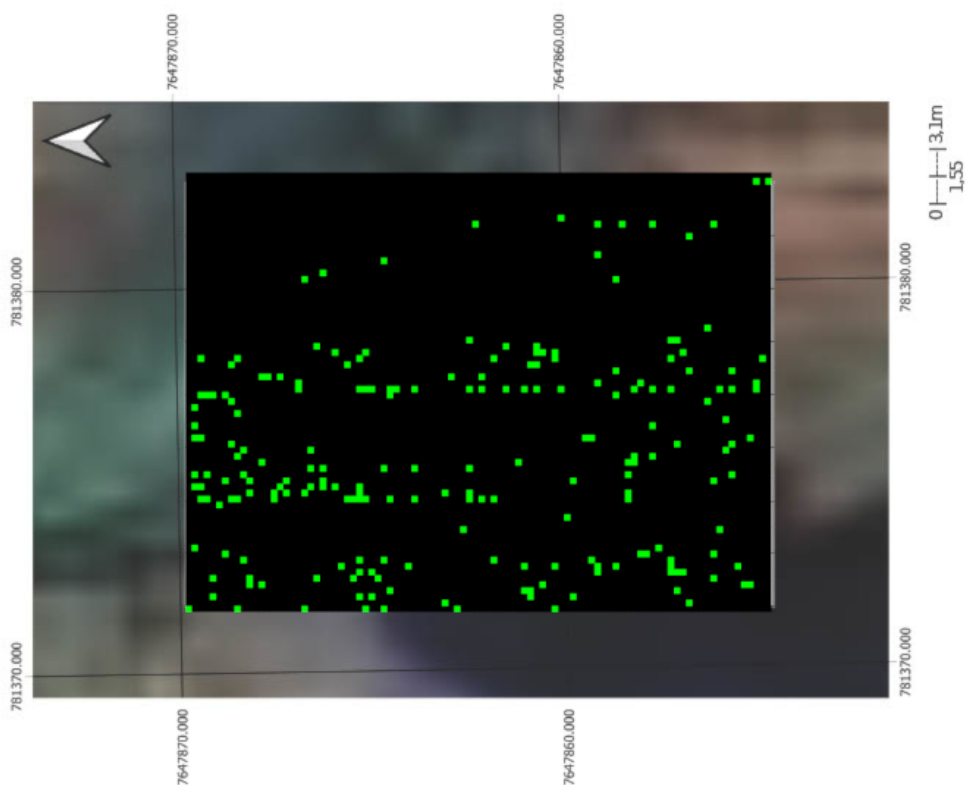


Figura 78 – Mapa taxa de ocupação mucuna-preta: fatia de profundidade acima de 30 cm.

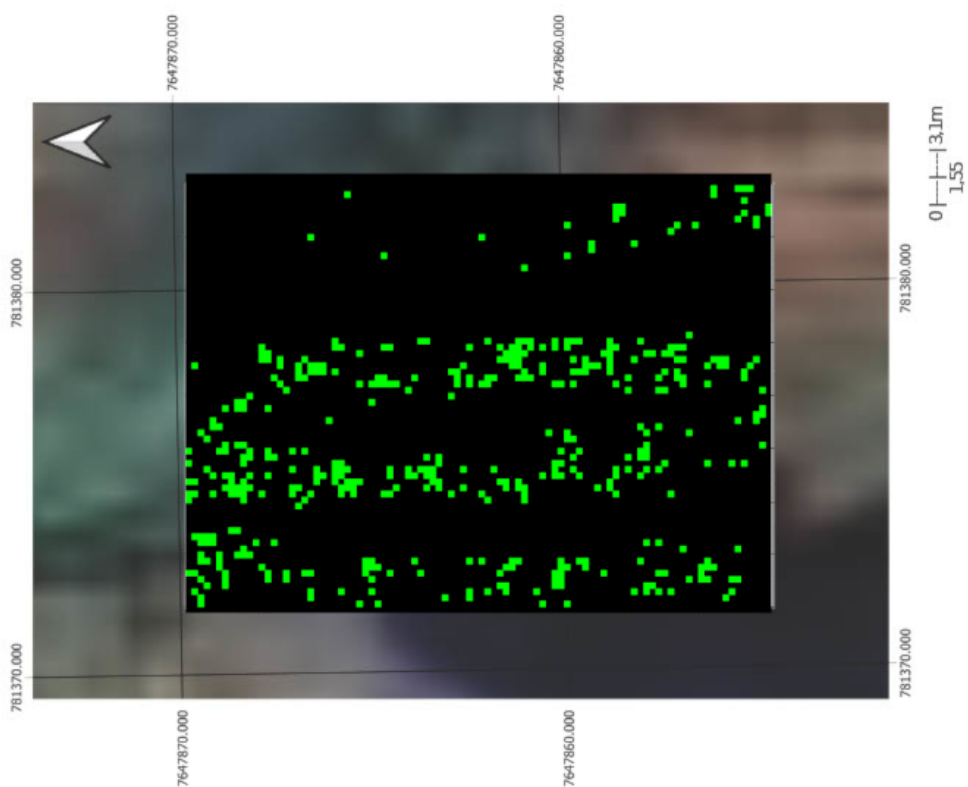


Figura 79 – Mapa taxa de ocupação capim braquiária.

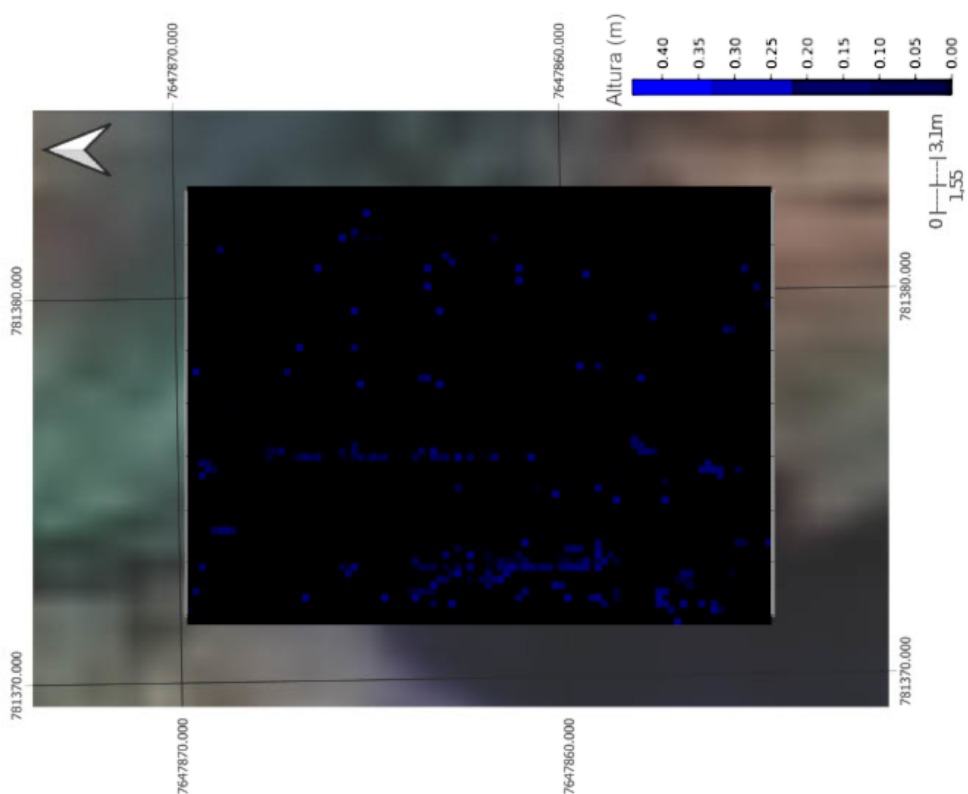


Figura 80 – Mapa taxa de ocupação capim braquiária: fatia de profundidade de até 15 cm.

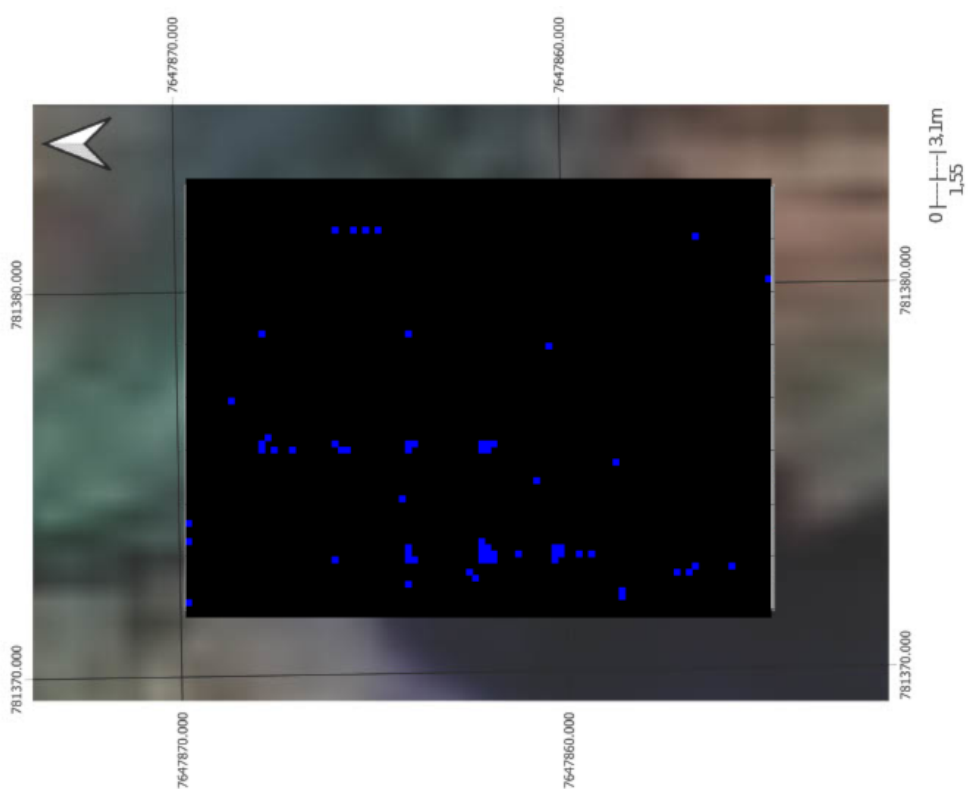




Figura 81 – Mapa taxa de ocupação capim braquiária: fatia de profundidade entre 15 e 30 cm.

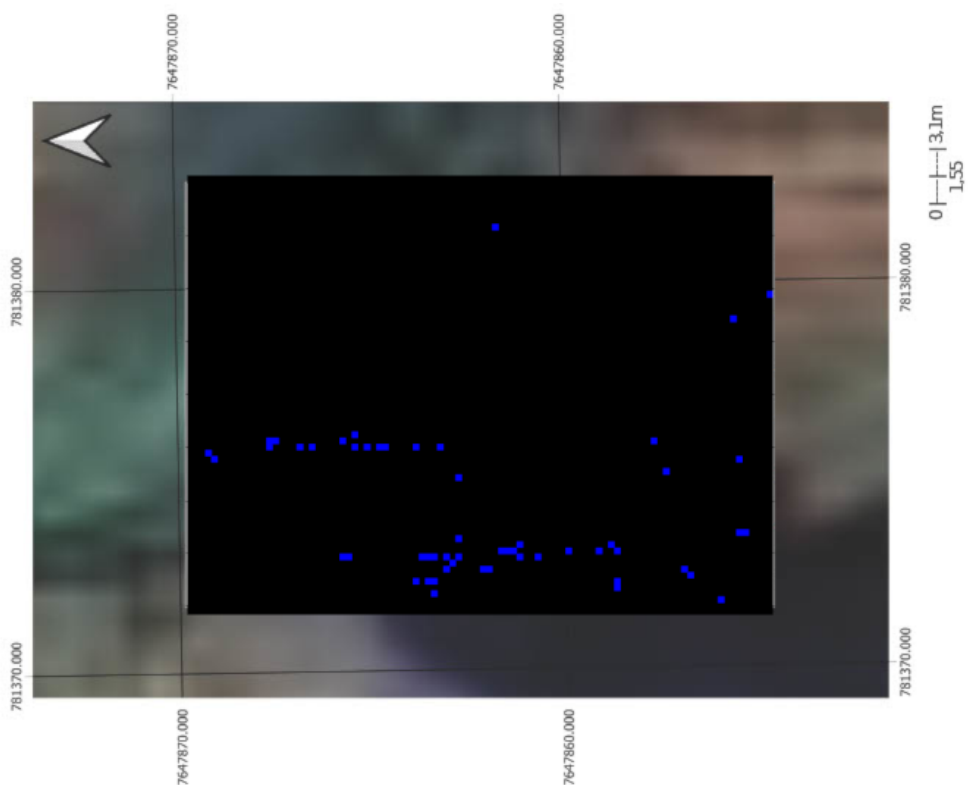
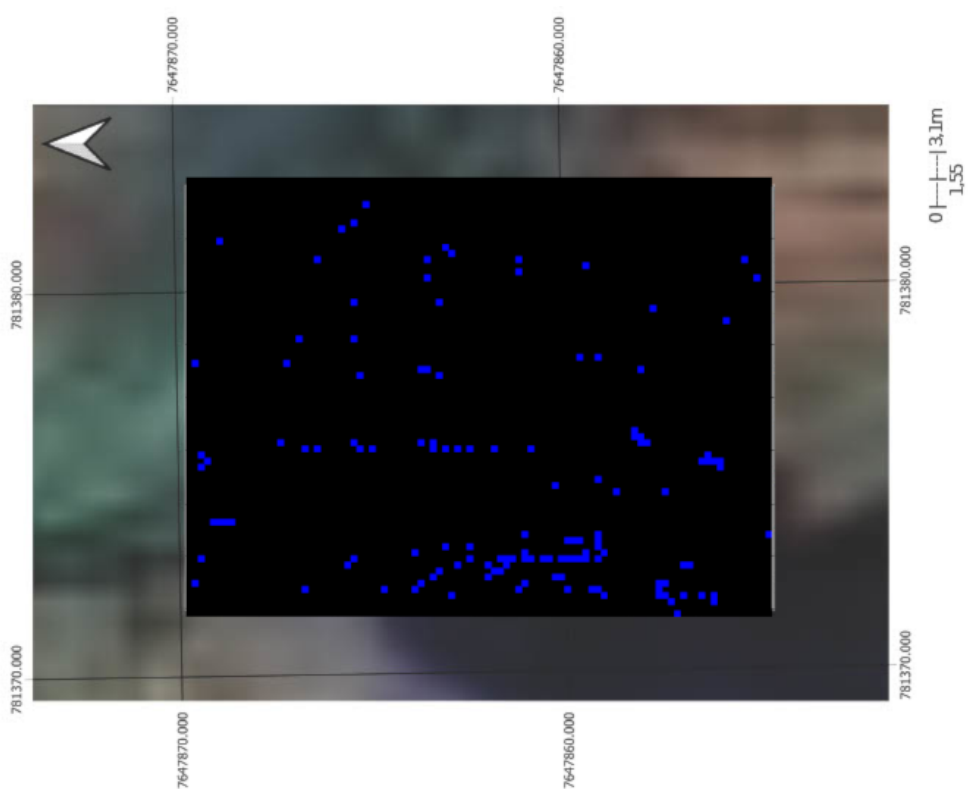


Figura 82 – Mapa taxa de ocupação capim braquiária: fatia de profundidade acima de 30 cm.



O sistema então como um todo foi capaz de gerar uma informação detalhada e localizada da presença, localização e densidade de plantas invasoras em uma área de cultivo. Tais tipos de dados permitem então que se possa ser realizado a pulverização automatizada de agrotóxicos de maneira muito mais controlada e precisa em campo.

# Capítulo 5

## CONCLUSÕES

---

---

Neste Capítulo são apresentadas as conclusões para a pesquisa desenvolvida em resposta ao alcance do objetivo geral e dos objetivos específicos programados. Adicionalmente, são sugeridas propostas para continuidade da pesquisa.

### 5.1 Principais conclusões

A classificação de famílias de plantas invasoras é de grande importância para o gerenciamento de plantações, pois permite a aplicação de insumos agrícolas de controle apenas na região necessária. Esta pesquisa apresentou um sistema fundamentado nos paradigmas da Visão Computacional capaz de promover o reconhecimento e classificação de padrões de diferentes famílias de plantas invasoras, tanto as plantas monocotiledôneas (folhas estreitas) como das dicotiledôneas (folhas largas).

O sistema foi desenvolvido para ser capaz de adquirir imagens a partir de comandos enviados por um aplicativo Android via Bluetooth, em tempo real, capaz de registrar vários pares de imagens de forma síncrona por minuto, viabilizando a obtenção de imagens estereoscópicas e garantindo a qualificação de plantas inteiras. Sua operação é independente de redes de energia, possuindo alimentação própria e portabilidade, com autonomia de até 15h. Resultados comprovaram que o protótipo desenvolvido pode ser usado em ambientes externos e em campos agrícolas experimentais em áreas reais de cultivo. Possui capacidade de armazenar até 6 mil imagens estereoscópicas com resolução de 1280 x 960 x 16 e capacidade de ajuste da altura da aquisição, linha de base e ângulo das câmeras.

As câmeras usadas foram validadas, tendo suas MTFs calculadas, analisando o desempenho das suas lentes e obtendo a resposta do sistema para as frequências espaciais. Além disso, os parâmetros intrínsecos a elas foram calculados, encontrando tanto a matriz da câmera de cada sensor quanto os coeficientes de distorção das suas respectivas lentes. Adicionalmente, foi considerado a avaliação da MTF total do sistema, cujos resultados auxiliaram na validação da qualidade esperada para a obtenção das imagens estereoscópicas.

Os resultados do experimento de campo que foram realizados validaram o desenvolvimento que possibilitaram a geração de bancos de imagens de famílias de plantas invasoras em ambiente real, ampliando a possibilidade de aplicação além de ambientes controlados de laboratório.

Dos dois algoritmos de pareamento de visão estéreo que foram analisados, foi possível concluir que deve-se balancear o tempo de processamento com a taxa de acerto, fatores críticos para aplicações.

Por fim, quanto aos classificadores SVM utilizados, resultados mostraram que a melhor resposta por validação cruzada para um classificador único geral, que agrupa todas as plantas invasoras presentes, ocorreu quando foi utilizado um *kernel* linear (acurácia de 65,7%), o que foi comprovado a partir das análises da matrizes de confusão e suas respectivas curvas ROC. Para classificadores específicos, isto é, em cada classificador é especializado em classificar apenas uma família de planta invasora, os melhores resultados de acurácia foram um *kernel* RBF (acurácia de 78,8%) para o capim-carrapicho, um *kernel* sigmoide (acurácia de 85,6%) para a leiteira, um *kernel* RBF (acurácia de 79,2%) para o capim-braquiária e um *kernel* RBF (acurácia de 81,1%) para a mucuna-preta. Todos os classificadores específicos apresentaram resultados e indicadores maiores do que quando se usa um classificador só para agrupar diferentes classes.

Agregando todos os resultados, foi possível então obter informações referentes ao volume das plantas invasoras e também gerar mapas de taxa de ocupação de diferentes famílias e em diferentes camadas de profundidade da área onde foi realizado o experimento de campo. Com esses dados, sistemas de pulverização inteligentes podem aplicar de forma mais precisa insumos agrícolas em campo;

## 5.2 Sugestões trabalhos futuros

Como contribuição para trabalhos futuros são considerados:

- Estudar e avaliar algoritmos de pareamento que possam melhorar a união de redução de tempo de processamento e de ampliação da taxa de acerto na integração das imagens primitivas;
- Estudar e avaliar o uso de modelos não-supervisionados e métodos CNN para reconhecimento de padrões de plantas invasoras.

## REFERÊNCIAS

---

---

- ABDESSELAM, A. Texture image retrieval using fourier transform. In: . [S.l.: s.n.], 2009. Citado na página 35.
- ABOUZIENA, H.; HAGGAG, W. Weed control in clean agriculture: a review. *Planta daninha*, SciELO Brasil, v. 34, p. 377–392, 2016. Citado na página 17.
- ABUD, H.; REIS, R.; TEÓFILO, E. Caracterização morfológica de frutos, sementes, plântulas e germinação de mucuna aterrima piper & tracy. *Revista Ciencia Agronomica*, v. 40, 01 2009. Citado na página 77.
- AGUILERA, P. Comparison of different image compression formats. *Wisconsin College of Engineering, ECE*, v. 533, 2006. Citado na página 62.
- AL-SHAKARJI, N. M.; KASSIM, Y. M.; PALANIAPPAN, K. Unsupervised learning method for plant and leaf segmentation. In: *2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*. [S.l.: s.n.], 2017. p. 1–4. ISSN 2332-5615. Citado 3 vezes nas páginas 22, 26 e 28.
- ALI, Z. H.; ALI, H. A.; BADAWEY, M. M. Internet of things (iot): definitions, challenges and recent research directions. *International Journal of Computer Applications*, Foundation of Computer Science, v. 128, n. 1, p. 37–47, 2015. Citado na página 61.
- ALIÓ, J. L.; SCHIMCHAK, P.; MONTÉS-MICÓ, R.; GALAL, A. Retinal image quality after microincision intraocular lens implantation. *Journal of Cataract & Refractive Surgery*, v. 31, n. 8, p. 1557–1560, 2005. ISSN 0886-3350. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0886335005002853>>. Citado na página 66.
- ALMAZAYDEH, L.; ALSALAMEN, R.; ELLEITHY, K. Herbal leaf recognition using mask-region convolutional neural network (mask r-cnn). *Journal of Theoretical and Applied Information Technology*, v. 100, n. 11, 2022. Citado 2 vezes nas páginas 29 e 43.
- ALTMAN, N. S. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, Taylor & Francis, v. 46, n. 3, p. 175–185, 1992. Disponível em: <<https://www.tandfonline.com/doi/abs/10.1080/00031305.1992.10475879>>. Citado na página 38.
- ANDREA, C.; DANIEL, B. B. M.; MISAEEL, J. B. J. Precise weed and maize classification through convolutional neuronal networks. In: *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*. [S.l.: s.n.], 2017. p. 1–6. Citado 5 vezes nas páginas 26, 27, 30, 43 e 45.
- ANTUNIASSI, U. R.; CAMARGO, T.; VELINE, E.; CAVENAGHI, A.; FIGUEIREDO, Z.; BONELLI, A. Controle de ferrugem da soja através de aplicações aéreas e terrestres. *Simpósio Internacional de Tecnologia de Aplicação de Agrotóxicos*, n. 3, 2004. Citado na página 17.

ARUNACHALAM, A.; ANDREASSON, H. Real-time plant phenomics under robotic farming setup: A vision-based platform for complex plant phenotyping tasks. *Computers and Electrical Engineering*, v. 92, 2021. Cited By 0. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85103247266&doi=10.1016%2fj.compeleceng.2021.107098&partnerID=40&md5=d389aa6a878fb865084a552e1169891c>>. Citado 5 vezes nas páginas 26, 28, 29, 30 e 36.

BACKES, A.; BRUNO, O. Plant leaf identification using color and multi-scale fractal dimension. In: . [S.l.: s.n.], 2010. p. 463–470. ISBN 978-3-642-13680-1. Citado na página 35.

BACKES, A. R.; CASANOVA, D.; BRUNO, O. M. Plant leaf identification based on volumetric fractal dimension. *International Journal of Pattern Recognition and Artificial Intelligence*, v. 23, n. 06, p. 1145–1160, 2009. Disponível em: <<https://doi.org/10.1142/S0218001409007508>>. Citado na página 35.

BADDOUR, N. Multidimensional wave field signal theory: Mathematical foundations. *AIP Advances*, v. 1, n. 2, p. 022120, 2011. Disponível em: <<https://aip.scitation.org/doi/abs/10.1063/1.3596359>>. Citado na página 64.

BAKHSIPOUR, A.; JAFARI, A. Evaluation of support vector machine and artificial neural networks in weed detection using shape features. *Computers and Electronics in Agriculture*, v. 145, p. 153–160, 2018. Cited By 61. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85039770602&doi=10.1016%2fj.compag.2017.12.032&partnerID=40&md5=d1f4c7cab2d995008f0260d5730cb256>>. Citado 6 vezes nas páginas 26, 27, 32, 33, 38 e 40.

BALAMURUGAN, C.; SATHEESH, R. Development of raspberry pi and iot based monitoring and controlling devices for agriculture. v. 6, p. 207–215, 08 2017. Citado na página 61.

BÁNDI, N.; TUNYOGI, R.-B.; SZABÓ, Z.; FARKAS, E.; SULYOK, C. Image-based volume estimation using stereo vision. In: *2020 IEEE 18th International Symposium on Intelligent Systems and Informatics (SISY)*. [S.l.: s.n.], 2020. p. 000055–000060. Citado na página 16.

BELFIELD, S.; BROWN, C. *Field crop manual: Maize: A guide to upland production in Cambodia*. [S.l.]: NSW Department of Primary Industries, 2008. Citado na página 56.

BELL, J.; DEE, H. Aberystwyth leaf evaluation dataset: A plant growth visible light image dataset of arabis thaliana. *Dynamic modelling of plant growth with computer vision (Nov. 2016)*. doi, v. 10, p. 5281, 2016. Citado 2 vezes nas páginas 22 e 24.

BERRAR, D. Cross-validation. In: \_\_\_\_\_. [S.l.: s.n.], 2018. ISBN 9780128096338. Citado na página 95.

BHATLA, S. C.; LAL, M. A. *Plant physiology, development and metabolism*. [S.l.]: Springer, 2018. Citado na página 17.

BISDIKIAN, C. An overview of the bluetooth wireless technology. *IEEE Communications Magazine*, v. 39, n. 12, p. 86–94, 2001. Citado na página 62.

BOJAMMA, A.; SHASTRY, C. A study on the machine learning techniques for automated plant species identification: current trends and challenges. *International Journal of Information Technology (Singapore)*, 2019. Cited By 0. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85091739824&doi=10.1007%2f>>

[2fs41870-019-00379-7&partnerID=40&md5=350a64ccf0f030cda96feb30dabb857a](#)>. Citado 3 vezes nas páginas 20, 21 e 31.

BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. New York, NY, USA: Association for Computing Machinery, 1992. (COLT '92), p. 144–152. ISBN 089791497X. Disponível em: <<https://doi.org/10.1145/130385.130401>>. Citado 2 vezes nas páginas 37 e 92.

BRAGA, A. F.; CHICONI, L. A.; BACHA, A. L.; TEIXEIRA, G. H. d. A.; JUNIOR, L. C. C.; ALVES, P. L. d. C. A. Discrimination of morningglory species (*ipomoea* spp.) using near-infrared spectroscopy and multivariate analysis. *Weed Science*, Cambridge University Press, v. 71, n. 2, p. 104–111, 2023. Citado na página 21.

BUSIN, L.; VANDENBROUCKE, N.; MACAIRE, L. Color spaces and image segmentation. *Advances in imaging and electron physics*, Academic Press, v. 151, n. 1, p. 1, 2008. Citado na página 82.

CANNY, J. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, Ieee, n. 6, p. 679–698, 1986. Citado 2 vezes nas páginas 28 e 87.

CARMONA, R. Problemática e manejo de bancos de sementes de invasoras em solos agrícolas. *Planta Daninha*, Sociedade Brasileira da Ciência das Plantas Daninhas, v. 10, n. 1-2, p. 05–16, 1992. ISSN 0100-8358. Disponível em: <<https://doi.org/10.1590/S0100-83581992000100007>>. Citado na página 68.

CARNEIRO, F. M.; FURLANI, C. E. A.; ZERBATO, C.; MENEZES, P. C. d.; GÍRIO, L. A. d. S. Correlations among vegetation indices and peanut traits during different crop development stages. *Engenharia Agrícola*, Associação Brasileira de Engenharia Agrícola, v. 39, n. spe, p. 33–40, Sep 2019. ISSN 0100-6916. Disponível em: <<https://doi.org/10.1590/1809-4430-Eng.Agric.v39nep33-40/2019>>. Citado na página 59.

CHECHI, A.; BOLLER, W.; FORCELINI, C. A.; ROEHRIG, R.; ZUCHELLI, E. Leaf area, spray volumes and applicable quantities of fungicide during the protection period for control of asian soybean rust. *Summa Phytopathologica*, Grupo Paulista de Fitopatologia, v. 44, n. 4, p. 326–331, Oct 2018. ISSN 0100-5405. Disponível em: <<https://doi.org/10.1590/0100-5405/174517>>. Citado na página 77.

CHEN, C. C.; TSAI, C. M.; FANG, Y. C. Optical design of lcos optical engine and optimization with genetic algorithm. *Journal of Display Technology*, v. 5, n. 8, p. 293–305, Aug 2009. ISSN 1551-319X. Citado na página 63.

CHEN, K. F.; YAN, F. L. On the integration schemes of retrieving impulse response functions from transfer functions. *Mathematical Problems in Engineering*, 2010. Disponível em: <<https://search.proquest.com/docview/859915370?accountid=26558>>. Citado na página 64.

CHEN, X.; WANG, B.; GAO, Y. Symmetric binary tree based co-occurrence texture pattern mining for fine-grained plant leaf image retrieval. *Pattern Recognition*, v. 129, p. 108769, 2022. ISSN 0031-3203. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0031320322002503>>. Citado 2 vezes nas páginas 34 e 39.

CHEN, X.; YOU, J.; TANG, H.; WANG, B.; GAO, Y. Fine-grained plant leaf image retrieval using local angle co-occurrence histograms. In: *2021 IEEE International Conference on Image Processing (ICIP)*. [S.l.: s.n.], 2021. p. 1599–1603. Citado 2 vezes nas páginas 34 e 43.

CHEN, Y.; BAIREDDY, S.; CAI, E.; YANG, C.; DELP, E. J. Leaf segmentation by functional modeling. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2019. p. 2685–2694. ISSN 2160-7516. Citado 4 vezes nas páginas 26, 28, 30 e 34.

CHENG, H.; JIANG, X.; SUN, Y.; WANG, J. Color image segmentation: advances and prospects. *Pattern Recognition*, v. 34, n. 12, p. 2259–2281, 2001. ISSN 0031-3203. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0031320300001497>>. Citado na página 84.

CHERNOV, V.; ALANDER, J.; BOCHKO, V. Integer-based accurate conversion between rgb and hsv color spaces. *Computers & Electrical Engineering*, v. 46, p. 328–337, 2015. ISSN 0045-7906. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0045790615002827>>. Citado na página 82.

CHIANG, P.-J.; LIN, C.-H. Active stereo vision system with rotated structured light patterns and two-step denoising process for improved spatial resolution. *Optics and Lasers in Engineering*, v. 152, p. 106958, 2022. ISSN 0143-8166. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0143816622000136>>. Citado na página 70.

CHOUHAN, S. S.; KAUL, A.; SINZLR, U. P. Plants leaf segmentation using bacterial foraging optimization algorithm. In: *2019 International Conference on Communication and Electronics Systems (ICCES)*. [S.l.: s.n.], 2019. p. 1500–1505. Citado na página 29.

CHRISTENHUSZ, M. J.; BYNG, J. W. The number of known plants species in the world and its annual increase. *Phytotaxa*, v. 261, n. 3, p. 201–217, 2016. Citado na página 17.

CHINA AGRICULTURAL UNIVERSITY. Zhang Chunlong, Han Xiaowu, Wang Han, Liao Qianfeng, Yuan Ting e Li Wei. *Precise pesticide spraying device based on visual deep learning and Internet of Things*. 2022. CN114946807A. China Patent Office. Citado na página 45.

COSTA, A.; MACIEL, C.; ROGOSKI, T.; LIRA, A.; HELVIG, E. Spraying deposits using different nozzles and application volumes for pest management of cotton at reproductive stage. *Australian Journal of Crop Science*, p. 1770–1776, 11 2019. Citado na página 78.

CUNHA, J. P.; CARVALHO, W. P. A. Tecnología de aplicación de agroquímicos por vía aérea. *S Tecnología de aplicación de agroquímicos*, p. 147–157, 2010. Citado na página 17.

DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, Wiley Online Library, v. 39, n. 1, p. 1–22, 1977. Citado na página 28.

DOBRESCU, A.; GIUFFRIDA, M. V.; TSAFTARIS, S. A. Understanding deep neural networks for regression in leaf counting. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2019. p. 2600–2608. ISSN 2160-7516. Citado 2 vezes nas páginas 22 e 43.

DOKANIA, N. K.; YADAV, S. S. Feature extraction techniques in agriculture with stressed vegetation: A review. In: *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*. [S.l.: s.n.], 2022. p. 1156–1162. Citado 3 vezes nas páginas 20, 21 e 26.



ECKERT, C.; FRIGO, E.; ALBRECHT, L.; ALBRECHT, A.; CHRIST, D.; SANTOS, W.; BERKEMBROCK, E.; EGEWARTH, V. Maize ethanol production in brazil: Characteristics and perspectives. *Renewable and Sustainable Energy Reviews*, v. 82, p. 3907–3912, 2018. ISSN 1364-0321. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S136403211731465X>>. Citado na página 69.

MILAR AGRO TECH SRL. Leonardo Elgart. *Device to detect and exercise control over weeds applied on agricultural machinery*. 2020. US20200375172. United States Patent and Trademark Office. Citado na página 45.

ELSTS, A.; MCCONVILLE, R. Are microcontrollers ready for deep learning-based human activity recognition? *Electronics*, v. 10, n. 21, 2021. ISSN 2079-9292. Disponível em: <<https://www.mdpi.com/2079-9292/10/21/2640>>. Citado na página 45.

EMBRAPA - Empresa Brasileira de Pesquisa Agropecuária. *Capim Carrapixo, Timbête, Capim-amoroso (Cenchrus echinatus L.)*. 2014. Disponível em: <<http://panorama.cnpms.embrapa.br/plantas-daninhas/identificacao/folhas-estreitas/capim-carrapixo-timbete-capim-amoroso-cenchrus-echinatus-l>>. Acesso em: 25 de abr. de 2023. Citado na página 57.

EMBRAPA - Empresa Brasileira de Pesquisa Agropecuária. *Leiteira (Amendoim-bravo) - Euphorbia heterophylla L.* 2014. Disponível em: <<http://panorama.cnpms.embrapa.br/plantas-daninhas/identificacao/folhas-largas/leiteira-amendoim-bravo-euphorbia-heterophylla-l>>. Acesso em: 25 de abr. de 2023. Citado na página 57.

ENVIRONMENTAL WEEDS OF AUSTRALIA. *Urochloa decumbens*. 2016. Disponível em: <[https://keyserver.lucidcentral.org/weeds/data/media/Html/urochloa\\_decumbens.htm](https://keyserver.lucidcentral.org/weeds/data/media/Html/urochloa_decumbens.htm)>. Acesso em: 25 de abr. de 2023. Citado na página 59.

FANG, Y.-C.; TSAI, C.-M. Miniature lens design and optimization with liquid lens element via genetic algorithm. *Journal of Optics A: Pure and Applied Optics*, v. 10, n. 7, p. 075304, 2008. Disponível em: <<http://stacks.iop.org/1464-4258/10/i=7/a=075304>>. Citado na página 63.

FANG, Y.-C.; TSAI, C.-M.; MACDONALD, J.; PAI, Y.-C. Eliminating chromatic aberration in gauss-type lens design using a novel genetic algorithm. *Applied Optics, OSA*, v. 46, n. 13, p. 2401–2410, May 2007. Disponível em: <<http://ao.osa.org/abstract.cfm?URI=ao-46-13-2401>>. Citado na página 63.

FAO. *FAOSTAT*. 2020. Disponível em: <<http://www.fao.org/faostat/en/#home>>. Acesso em: 12 de abr. de 2023. Citado na página 14.

FENG, X.; JIANG, Y.; YANG, X.; DU, M.; LI, X. Computer vision algorithms and hardware implementations: A survey. *Integration*, v. 69, p. 309–320, 2019. ISSN 0167-9260. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0167926019301762>>. Citado na página 15.

FLORINDO, J.; BRUNO, O. Discrete schrodinger transform for texture recognition. *Information Sciences*, v. 415-416, p. 142–155, 2017. Cited By 9. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85021196227&doi=10.1016%2fj.ins.2017.06.022&partnerID=40&md5=c5f98cfd7120caba203c84834f548b3c>>. Citado 3 vezes nas páginas 36, 37 e 41.

- FLORINDO, J. B.; METZE, K. A cellular automata approach to local patterns for texture recognition. *Expert Systems with Applications*, v. 179, p. 115027, 2021. ISSN 0957-4174. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0957417421004681>>. Citado 3 vezes nas páginas 35, 38 e 41.
- GARBAY, T.; DOBIAS, P.; DRON, W.; LUSICH, P.; KHALIS, I.; PINNA, A.; HACHICHA, K.; GRANADO, B. Cnn inference costs estimation on microcontrollers: the est primitive-based model. In: *2021 28th IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*. [S.l.: s.n.], 2021. p. 1–5. Citado na página 45.
- GAYATRI, K.; KANTI, R.; RAYAVARAPU, V. S. R.; SRIDHAR, B.; BOBBILI, V. R. G. Image processing and pattern recognition based plant leaf diseases identification and classification. In: . [s.n.], 2021. v. 1804, n. 1. Cited By 0. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85102350859&doi=10.1088%2f1742-6596%2f1804%2f1%2f012160&partnerID=40&md5=3c3167e1fc44c665ab4d39e37c75db24>>. Citado 4 vezes nas páginas 26, 28, 34 e 38.
- GE, L.; YANG, Z.; SUN, Z.; ZHANG, G.; ZHANG, M.; ZHANG, K.; ZHANG, C.; TAN, Y.; LI, W. A method for broccoli seedling recognition in natural environment based on binocular stereo vision and gaussian mixture model. *Sensors*, v. 19, n. 5, 2019. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/19/5/1132>>. Citado na página 46.
- GEBBERS, R.; ADAMCHUK, V. I. Precision agriculture and food security. *Science*, American Association for the Advancement of Science, v. 327, n. 5967, p. 828–831, 2010. Citado na página 15.
- GIUFFRIDA, M. V.; DOBRESCU, A.; DOERNER, P.; TSAFTARIS, S. A. Leaf counting without annotations using adversarial unsupervised domain adaptation. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2019. p. 2590–2599. ISSN 2160-7516. Citado 2 vezes nas páginas 22 e 43.
- GOËAU, H.; BONNET, P.; JOLY, A.; BOUJEMAA, N.; BARTHELEMY, D.; MOLINO, J.-F.; BIRNBAUM, P.; MOUYSSSET, E.; PICARD, M. The imageclef 2011 plant images classification task. In: *ImageCLEF 2011*. [S.l.: s.n.], 2011. Citado na página 25.
- GOËAU, H.; JOLY, A.; BONNET, P.; BAKIC, V.; BARTHÉLÉMY, D.; BOUJEMAA, N.; MOLINO, J.-F. The imageclef 2012 plant identification task. In: . [S.l.: s.n.], 2013. v. 1178, p. 23–28. Citado na página 25.
- GOMES, J. F. S.; LETA, F. R. Applications of computer vision techniques in the agriculture and food industry: a review. *European Food Research and Technology*, Springer, v. 235, p. 989–1000, 2012. Citado na página 16.
- GOSTA, M.; GRGIC, M. Accomplishments and challenges of computer stereo vision. In: *Proceedings ELMAR-2010*. [S.l.: s.n.], 2010. p. 57–64. Citado na página 15.
- GUAN, Y.; HAN, Y.; LIU, S. Deep learning approaches for image classification techniques. In: *2022 IEEE International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA)*. [S.l.: s.n.], 2022. p. 1132–1136. Citado na página 42.
- HAMEED, J.; ÜSTÜNDAĞ, B. Hybrid intelligent technology for plant health using the fusion of evolutionary optimization and deep neural networks. *Expert Systems*, v. 40, 06 2021. Citado 4 vezes nas páginas 26, 28, 32 e 43.

- HAN, L.; GAO, K.; GONG, C.; ZHU, Z.; GUO, Y. An optimized knife-edge method for on-orbit MTF estimation of optical sensors using powell parameter fitting. In: IFTEKHARUDDIN, K. M.; AWWAL, A. A. S.; VÁZQUEZ, M. G.; MÁRQUEZ, A.; DIAZ-RAMIREZ, V. H. (Ed.). *Optics and Photonics for Information Processing XI*. SPIE, 2017. v. 10395, p. 1039506. Disponível em: <<https://doi.org/10.1117/12.2271800>>. Citado na página 66.
- HARALICK, R. M.; SHANMUGAM, K.; DINSTEN, I. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3, n. 6, p. 610–621, 1973. Citado na página 89.
- HARALICK, R. M.; STERNBERG, S. R.; ZHUANG, X. Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9, n. 4, p. 532–550, 1987. Citado na página 85.
- HARTLEY, R.; ZISSERMAN, A. *Multiple View Geometry in Computer Vision*. 2. ed. [S.l.]: Cambridge University Press, 2004. Citado 2 vezes nas páginas 79 e 81.
- HIGGS, N.; LEYEZA, B.; UBBENS, J.; KOCUR, J.; KAMP, W. van der; CORY, T.; EYNCK, C.; VAIL, S.; ERAMIAN, M.; STAVNESS, I. Protractor: A lightweight ground imaging and analysis system for early-season field phenotyping. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2019. p. 2629–2638. ISSN 2160-7516. Citado 4 vezes nas páginas 25, 26, 27 e 29.
- HIRASEN, D.; VIRIRI, S. Plant species recognition using local binary and local directional patterns. In: *2020 2nd International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*. [S.l.: s.n.], 2020. p. 1–9. Citado 4 vezes nas páginas 21, 35, 38 e 39.
- HIRSCHMULLER, H. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 30, n. 2, p. 328–341, 2008. Citado na página 78.
- HOLM, L. G.; PLUCKNETT, D. L.; PANCHO, J. V.; HERBERGER, J. P. et al. The world's worst weeds. *The world's worst weeds.*, University Press., 1977. Citado na página 17.
- HU, R.; JIA, W.; LING, H.; HUANG, D. Multiscale distance matrix for fast plant leaf recognition. *IEEE Transactions on Image Processing*, v. 21, n. 11, p. 4667–4672, 2012. Citado na página 23.
- HUANG, J.; RATHOD, V.; SUN, C.; ZHU, M.; KORATTIKARA, A.; FATHI, A.; FISCHER, I.; WOJNA, Z.; SONG, Y.; GUADARRAMA, S.; MURPHY, K. Speed/accuracy trade-offs for modern convolutional object detectors. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2017. p. 3296–3297. ISSN 1063-6919. Citado 2 vezes nas páginas 25 e 44.
- IBRD. *World Development Indicators*. 2021. Disponível em: <<https://data.worldbank.org/indicator/>>. Acesso em: 12 de abr. de 2023. Citado na página 14.
- IMANOV, E.; ALZOUHBI, A. Machine learning comparative analysis for plant classification. *Advances in Intelligent Systems and Computing*, v. 896, p. 586–593, 2019. Cited By 1. Disponível em: <[https://www.scopus.com/inward/record.uri?eid=2-s2.0-85059739577&doi=10.1007%2f978-3-030-04164-9\\_77&partnerID=40&md5=16c042b8b61c7b959a036b13b922621e](https://www.scopus.com/inward/record.uri?eid=2-s2.0-85059739577&doi=10.1007%2f978-3-030-04164-9_77&partnerID=40&md5=16c042b8b61c7b959a036b13b922621e)>. Citado 3 vezes nas páginas 36, 38 e 39.

JASKOWIAK, P. A.; COSTA, I. G.; CAMPELLO, R. J. The area under the roc curve as a measure of clustering quality. *Data Mining and Knowledge Discovery*, Springer, v. 36, n. 3, p. 1219–1245, 2022. Citado na página 95.

JAT, R.; MEENA, H.; SINGH, A.; SURYA, J. N.; MISRA, J. Weed management in groundnut (*arachis hypogaea* l.) in india-a review. *Agricultural Reviews*, Agricultural Research Communication Centre, v. 32, n. 3, p. 155–171, 2011. Citado na página 57.

JIANGSU UNIVERSITY. Gao Jianmin e Jin Zhipeng. *Bionic four-foot walking intelligent rotary tillage weeding device and implementation method*. 2022. CN114794067A. China Patent Office. Citado na página 44.

JIMENEZ, A. P.; ROJAS, C.; GUZMAN, L. Procesamiento de imágenes para la identificación de plagas en los cultivos de espinaca. In: . [s.n.], 2017. Cited By 0. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85012004436&doi=10.1109%2fCCRA.2016.7811412&partnerID=40&md5=401db18fdf5c56368e077807ec1049a8>>. Citado 2 vezes nas páginas 27 e 28.

JOBLOVE, G. H.; GREENBERG, D. Color spaces for computer graphics. *SIGGRAPH Comput. Graph.*, Association for Computing Machinery, New York, NY, USA, v. 12, n. 3, p. 20–25, aug 1978. ISSN 0097-8930. Disponível em: <<https://doi.org/10.1145/965139.807362>>. Citado na página 83.

JYE, K. S.; MANICKAM, S.; MALEK, S.; MOSLEH, M.; DHILLON, S. K. Automated plant identification using artificial neural network and support vector machine. *FRONTIERS IN LIFE SCIENCE*, 10, n. 1, p. 98–107, 2017. ISSN 2155-3769. Citado 10 vezes nas páginas 26, 27, 29, 30, 31, 32, 33, 34, 38 e 41.

KAMATH, R.; BALACHANDRA, M.; PRABHU, S. Crop and weed discrimination using Laws' texture masks. *INTERNATIONAL JOURNAL OF AGRICULTURAL AND BIOLOGICAL ENGINEERING*, 13, n. 1, p. 191–197, JAN 2020. ISSN 1934-6344. Citado 5 vezes nas páginas 26, 30, 35, 37 e 41.

KAWAGISHI, N.; KAKINUMA, R.; YAMAMOTO, H. Aerial image resolution measurement based on the slanted knife edge method. *Opt. Express*, Optica Publishing Group, v. 28, n. 24, p. 35518–35527, Nov 2020. Disponível em: <<https://opg.optica.org/oe/abstract.cfm?URI=oe-28-24-35518>>. Citado na página 65.

KHAN, N. A.; LYON, O. A. S.; ERAMIAN, M.; MCQUILLAN, I. A novel technique combining image processing, plant development properties, and the hungarian algorithm, to improve leaf detection in maize. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2020. p. 330–339. ISSN 2160-7516. Citado 4 vezes nas páginas 26, 27, 32 e 34.

KLETTE, R. *Concise computer vision*. [S.l.]: Springer, 2014. v. 233. Citado na página 15.

KONSTANTAKOPOULOS, F.; GEORGA, E. I.; FOTIADIS, D. I. 3d reconstruction and volume estimation of food using stereo vision techniques. In: *2021 IEEE 21st International Conference on Bioinformatics and Bioengineering (BIBE)*. [S.l.: s.n.], 2021. p. 1–4. Citado na página 16.

KRAUSE, J.; BAEK, K.; LIM, L. A guided multi-scale categorization of plant species in natural images. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2019. p. 2639–2647. ISSN 2160-7516. Citado 2 vezes nas páginas 29 e 43.

KRITSIS, K.; KIOURT, C.; STAMOULI, S.; SEVETLIDIS, V.; SOLOMOU, A.; KARETSOS, G.; KATSOUROS, V.; PAVLIDIS, G. Grasp-125: A dataset for greek vascular plant recognition in natural environment. *Sustainability*, v. 13, n. 21, 2021. ISSN 2071-1050. Disponível em: <<https://www.mdpi.com/2071-1050/13/21/11865>>. Citado na página 43.

KUMAR, N.; BELHUMEUR, P. N.; BISWAS, A.; JACOBS, D. W.; KRESS, W. J.; LOPEZ, I. C.; SOARES, J. V. Leafsnap: A computer vision system for automatic plant species identification. In: SPRINGER. *European conference on computer vision*. [S.l.], 2012. p. 502–516. Citado na página 25.

KUZNICHOV, D.; ZVIRIN, A.; HONEN, Y.; KIMMEL, R. Data augmentation for leaf segmentation and counting tasks in rosette plants. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2019. p. 2580–2589. ISSN 2160-7516. Citado 2 vezes nas páginas 22 e 25.

LAZAROS, N.; SIRAKOULIS, G. C.; GASTERATOS, A. Review of stereo vision algorithms: From software to hardware. *International Journal of Optomechatronics*, Taylor & Francis, v. 2, n. 4, p. 435–462, 2008. Citado 2 vezes nas páginas 73 e 74.

LE, V.; AHDEROM, S.; APOPEI, B.; ALAMEH, K. A novel method for detecting morphologically similar crops and weeds based on the combination of contour masks and filtered local binary pattern operators. *GigaScience*, v. 9, n. 3, 2020. Cited By 5. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85081041719&doi=10.1093%2fgigascience%2fgiaa017&partnerID=40&md5=2a4843db0fee098b804769ae8b798491>>. Citado 7 vezes nas páginas 26, 29, 35, 38, 85, 87 e 88.

LEE, J.; JUN, D.; EEM, C.; HONG, H. Improved census transform for noise robust stereo matching. *Optical Engineering*, SPIE, v. 55, n. 6, p. 063107, 2016. Disponível em: <<https://doi.org/10.1117/1.OE.55.6.063107>>. Citado 2 vezes nas páginas 74 e 75.

LI, D.; SHI, G.; KONG, W.; WANG, S.; CHEN, Y. A leaf segmentation and phenotypic feature extraction framework for multiview stereo plant point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, v. 13, p. 2321–2336, 2020. Cited By 3. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85086233933&doi=10.1109%2fJSTARS.2020.2989918&partnerID=40&md5=e3173b82dbe098a60f9472d2701cfd3e>>. Citado 3 vezes nas páginas 21, 31 e 33.

LI, D.; XU, L.; TANG, X.-s.; SUN, S.; CAI, X.; ZHANG, P. 3d imaging of greenhouse plants with an inexpensive binocular stereo vision system. *Remote Sensing*, v. 9, n. 5, 2017. ISSN 2072-4292. Disponível em: <<https://www.mdpi.com/2072-4292/9/5/508>>. Citado na página 46.

LI, M.; LIM, S. C.; CHEN, S. Exact solution of impulse response to a class of fractional oscillators and its stability. *Mathematical Problems in Engineering*, 2011. Disponível em: <<https://search.proquest.com/docview/856225961?accountid=26558>>. Citado na página 64.

LI, X.; FAN, W.; WANG, Y.; ZHANG, L.; LIU, Z.; XIA, C. Detecting plant leaves based on vision transformer enhanced yolov5. In: *2022 3rd International Conference on Pattern Recognition and Machine Learning (PRML)*. [S.l.: s.n.], 2022. p. 32–37. Citado 2 vezes nas páginas 29 e 43.

LIAO, J.; ZANG, Y.; LUO, X.; ZHOU, Z.; ZANG, Y.; WANG, P.; HEWITT, A. J. The relations of leaf area index with the spray quality and efficacy of cotton defoliant spraying using

unmanned aerial systems (uass). *Computers and Electronics in Agriculture*, v. 169, p. 105228, 2020. ISSN 0168-1699. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S016816991931364X>>. Citado na página 77.

LIEBMAN, M.; DYCK, E. Weed management: a need to develop ecological approaches. *Ecological applications*, JSTOR, v. 3, n. 1, p. 39–41, 1993. Citado na página 17.

LIN, T.-T.; LAI, T.; LIU, C.; CHENG, Y. A three-dimensional imaging approach for plant feature measurement using stereo vision. *Journal of Agricultural Machinery Science*, v. 7, p. 153–158, 01 2011. Citado 2 vezes nas páginas 46 e 76.

LIN, T.-Y.; MAIRE, M.; BELONGIE, S.; HAYS, J.; PERONA, P.; RAMANAN, D.; DOLLÁR, P.; ZITNICK, C. L. Microsoft coco: Common objects in context. In: SPRINGER. *European conference on computer vision*. [S.l.], 2014. p. 740–755. Citado na página 22.

LIN, X.; LI, C.-T.; ADAMS, S.; KOUZANI, A. Z.; JIANG, R.; HE, L.; HU, Y.; VERNON, M.; DOEVEN, E.; WEBB, L.; MCCLELLAN, T.; GUSKICH, A. Self-supervised leaf segmentation under complex lighting conditions. *Pattern Recognition*, v. 135, p. 109021, 2023. ISSN 0031-3203. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0031320322005015>>. Citado 3 vezes nas páginas 22, 27 e 29.

LYU, B.; SMITH, S. D.; CHERKAUER, K. A. Fine-grained recognition in high-throughput phenotyping. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2020. p. 320–329. ISSN 2160-7516. Citado 3 vezes nas páginas 22, 27 e 33.

MA, Y.; DAI, R.; LI, L.; WEI, L. Image segmentation of embryonic plant cell using pulse-coupled neural networks. *Chinese Science Bulletin*, v. 47, p. 169–173, 01 2002. Citado na página 35.

MACDONALD, J. D. Etymology and cultural significance of the term agriculture. 2020. Citado na página 14.

MACQUEEN, J. et al. Some methods for classification and analysis of multivariate observations. In: OAKLAND, CA, USA. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. [S.l.], 1967. v. 1, n. 14, p. 281–297. Citado na página 28.

MALACH, T.; POMENKOVA, J. Comparing classifier's performance based on confidence interval of the roc. *Radioengineering*, v. 27, p. 827–834, 09 2018. Citado na página 44.

MALIK, O.; ISMAIL, N.; HUSSEIN, B.; YAHYA, U. Automated real-time identification of medicinal plants species in natural environment using deep learning models—a case study from borneo region. *Plants*, v. 11, p. 1952, 07 2022. Citado na página 43.

MAPA. *Agricultura de Precisão - Boletim Técnico*. 1. ed. Brasília, 2009. 31 p. Disponível em: <[https://www.gov.br/agricultura/pt-br/assuntos/insumos-agropecuarios/arquivos-publicacoes-insumos/agric\\_precisao.pdf](https://www.gov.br/agricultura/pt-br/assuntos/insumos-agropecuarios/arquivos-publicacoes-insumos/agric_precisao.pdf)>. Citado na página 15.

MAPA. *Relatório Valor Bruto da Produção Agropecuária (VBP) - 04/2023*. 2023. Disponível em: <<https://www.gov.br/agricultura/pt-br/assuntos/politica-agricola/valor-bruto-da-producao-agropecuaria-vbp>>. Acesso em: 15 de mai. de 2023. Citado na página 14.

- MARQUARDT, D. W. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, v. 11, n. 2, p. 431–441, 1963. Citado na página 40.
- MATHE, S. E.; BANDARU, M.; KONDAVEETI, H. K.; VAPPANGI, S.; RAO, G. S. A survey of agriculture applications utilizing raspberry pi. In: *2022 International Conference on Innovative Trends in Information Technology (ICITIIT)*. [S.l.: s.n.], 2022. p. 1–7. Citado na página 49.
- MCLOUGHLIN D. Daniel McLoughlin. *Image processing*. 2000. EP1000540A1. European Patent Office. Citado na página 44.
- MEER, P. Robust techniques for computer vision. *Emerging topics in computer vision*, Prentice Hall Upper Saddle River, p. 107–190, 2004. Citado na página 16.
- MIFLIN, B. Crop improvement in the 21st century. *Journal of experimental botany*, Oxford University Press, v. 51, n. 342, p. 1–8, 2000. Citado na página 15.
- MORENO, A.; OKAMOTO, E. Bluesnarf revisited: Obex ftp service directory traversal. In: SPRINGER. *NETWORKING 2011 Workshops: International IFIP TC 6 Workshops, PE-CRN, NC-Pro, WCNS, and SUNSET 2011, Held at NETWORKING 2011, Valencia, Spain, May 13, 2011, Revised Selected Papers 10*. [S.l.], 2011. p. 155–166. Citado na página 62.
- MORENO, B. M.; CRUVINEL, P. E. Sensors-based stereo image system for precision control of weed in the agricultural industry. p. 69–76, 2018. Citado na página 46.
- MORENO, B. M.; CRUVINEL, P. E. Computer vision system for identifying on farming weed species. In: *2022 IEEE 16th International Conference on Semantic Computing (ICSC)*. [S.l.: s.n.], 2022. p. 287–292. Citado na página 28.
- MORENO, B. M.; CRUVINEL, P. E. Characterization of an iot stereo image sensor system for weed control. p. 1–7, 2023. Citado na página 46.
- NAUMOV, A. A.; GOREVOY, A. V.; MACHIKHIN, A. S.; BATSHEV, V. I.; POZHA, V. E. Estimating the quality of stereoscopic endoscopic systems. *Journal of Physics: Conference Series*, IOP Publishing, v. 1421, n. 1, p. 012044, dec 2019. Disponível em: <<https://dx.doi.org/10.1088/1742-6596/1421/1/012044>>. Citado na página 66.
- NEVATIA, R. Depth measurement by motion stereo. *Computer Graphics and Image Processing*, v. 5, n. 2, p. 203–214, 1976. ISSN 0146-664X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0146664X76900289>>. Citado na página 70.
- NEVES, R.; MATOS, A. C. Raspberry pi based stereo vision for small size asvs. In: *2013 OCEANS - San Diego*. [S.l.: s.n.], 2013. p. 1–6. Citado na página 49.
- NGUYEN, T. T.; SLAUGHTER, D. C.; MALOOF, J. N.; SINHA, N. Plant phenotyping using multi-view stereo vision with structured lights. In: VALASEK, J.; THOMASSON, J. A. (Ed.). *Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping*. SPIE, 2016. v. 9866, p. 986608. Disponível em: <<https://doi.org/10.1117/12.2229513>>. Citado na página 46.
- NORTH CAROLINA STATE UNIVERSITY EXTENSION. *Arachis hypogaea*. 2023. Disponível em: <<https://plants.ces.ncsu.edu/plants/arachis-hypogaea/>>. Acesso em: 25 de abr. de 2023. Citado na página 59.

NOVOTNY, P.; SUK, T. Leaf recognition of woody species in central europe. *Biosystems Engineering*, v. 115, n. 4, p. 444–452, 2013. ISSN 1537-5110. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1537511013000731>>. Citado 2 vezes nas páginas 22 e 25.

NPARKS FLORA & FAUNA WEB. *Plants of Singapore - Zea mays L.* 2023. Disponível em: <<https://www.nparks.gov.sg/florafaunaweb/flora/2/5/2568>>. Acesso em: 25 de abr. de 2023. Citado na página 56.

OH, E.; CHOI, J.-K. Goci image enhancement using an mtf compensation technique for coastal water applications. *Opt. Express*, Optica Publishing Group, v. 22, n. 22, p. 26908–26918, Nov 2014. Disponível em: <<https://opg.optica.org/oe/abstract.cfm?URI=oe-22-22-26908>>. Citado na página 66.

OKURA, F. 3d modeling and reconstruction of plants and trees: A cross-cutting review across computer graphics, vision, and plant phenotyping. *Breeding Science*, v. 72, n. 1, p. 31–47, 2022. Citado na página 70.

OTSU, N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 9, n. 1, p. 62–66, 1979. Citado na página 27.

O'MAHONY, N.; CAMPBELL, S.; CARVALHO, A.; HARAPANAHALLI, S.; HERNANDEZ, G. V.; KRPALKOVA, L.; RIORDAN, D.; WALSH, J. Deep learning vs. traditional computer vision. In: SPRINGER. *Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC), Volume 1 I.* [S.l.], 2020. p. 128–144. Citado na página 44.

PARASHAR, J.; BHANDARKAR, S. M.; SIMON, J.; HOPKINSON, B. M.; PENNING, S. C. Estimation of abundance and distribution of salt marsh plants from images using deep learning. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. [S.l.: s.n.], 2021. p. 2635–2642. Citado 2 vezes nas páginas 29 e 43.

PATIL, K. A.; KALE, N. R. A model for smart agriculture using iot. In: *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPIC)*. [S.l.: s.n.], 2016. p. 543–545. Citado na página 61.

PATLE, A.; CHOUHAN, D. S. Svm kernel functions for classification. In: *2013 International Conference on Advances in Technology and Engineering (ICATE)*. [S.l.: s.n.], 2013. p. 1–9. Citado na página 91.

PATRÓN, J.; SENDOYA-LOSADA, D.; ROBAYO-BETANCOURT, F. Design and implementation of an algorithm for plants identification and classification based on physical characteristics of their leaves using computer vision. *ARNP Journal of Engineering and Applied Sciences*, v. 13, n. 5, p. 1563–1569, 2018. Cited By 0. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85044136598&partnerID=40&md5=e10abddaafa3f6e09255defalca3692a>>. Citado 3 vezes nas páginas 26, 34 e 40.

PAWARA, P.; OKAFOR, E.; SURINTA, O.; SCHOMAKER, L.; WIERING, M. Comparing local descriptors and bags of visualwords to deep convolutional neural networks for plant recognition. In: . [s.n.], 2017. v. 2017-January, p. 479–486. Cited By 38. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85046347005&doi=10.5220%2f0006196204790486&partnerID=40&md5=288a3e4634c21b4fed01a938eb7e909b>>. Citado 5 vezes nas páginas 33, 38, 39, 41 e 43.



- PEARLSTEIN, L.; KIM, M.; SETO, W. Convolutional neural network application to plant detection, based on synthetic imagery. In: . [s.n.], 2017. Cited By 1. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85028758422&doi=10.1109%2fAIPR.2016.8010596&partnerID=40&md5=21c7f53eac9add9b6c8a5b74bfa52874>>. Citado na página 43.
- PENGCHENG, P.; YUE, S.; CHENGBO, W.; XINMING, M.; WA, G.; RONGBO, Q. The uav image classification method based on the grey-sigmoid kernel function support vector machine. In: *2018 10th IAPR Workshop on Pattern Recognition in Remote Sensing (PRRS)*. [S.l.: s.n.], 2018. p. 1–6. ISSN 2377-0198. Citado na página 29.
- PEREIRA, C. S.; MORAIS, R.; REIS, M. J. C. S. Recent advances in image processing techniques for automated harvesting purposes: A review. In: *2017 Intelligent Systems Conference (IntelliSys)*. [S.l.: s.n.], 2017. p. 566–575. Citado 4 vezes nas páginas 20, 26, 45 e 84.
- PEREIRA, L. E. T.; BUENO, I. C. d. S.; HERLING, V. R.; ALMEIDA, O. J. I. d. A dinâmica do crescimento de plantas forrageiras e o manejo das pastagens. 2016. Citado na página 77.
- POWERS, D. Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation. *Mach. Learn. Technol.*, v. 2, 01 2008. Citado 2 vezes nas páginas 44 e 94.
- PRASAD, P. V.; KAKANI, V. G.; UPADHYAYA, H. D. Growth and production of groundnut. *UNESCO Encyclopedia, Encyclopedia of Life Support Systems (EOLSS)*, Developed under the Auspices . . . , p. 1–26, 2010. Citado na página 69.
- PREETHI, M.; RANI, S. J.; PRADHIKSHA, K. S.; KUMAR, J. R.; VISHAL, T. Medicinal herbs identification. In: *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*. [S.l.: s.n.], 2023. v. 1, p. 1050–1053. Citado 6 vezes nas páginas 21, 27, 29, 31, 32 e 43.
- QU, Z.; WEI, X.-M.; CHEN, S.-Q. An algorithm of image mosaic based on binary tree and eliminating distortion error. *PLoS ONE*, v. 14, n. 1, 2019. Cited By 5. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85059607623&doi=10.1371%2fjournal.pone.0210354&partnerID=40&md5=ff1ab5e097ec7ff4d66d943c9069daf1>>. Citado na página 32.
- RAJCAN, I.; SWANTON, C. J. Understanding maize–weed competition: resource competition, light quality and the whole plant. *Field Crops Research*, v. 71, n. 2, p. 139–150, 2001. ISSN 0378-4290. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0378429001001599>>. Citado na página 17.
- RAJORIYA, M.; TIWARI, U. Pattern recognition in agricultural areas. *Journal of Critical Reviews*, v. 7, n. 10, p. 1123–1127, 2020. Cited By 0. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85088024443&doi=10.31838%2fjcr.07.10.220&partnerID=40&md5=1f7a13b54c39a9780bc91c78ea2a2bd1>>. Citado na página 37.
- RAJORIYA, U. T. M. Pattern recognition in agricultural areas. *Journal of Critical Reviews*, n. 10, p. 1123–1127, 2020. Citado na página 16.
- RASMUSSEN, P. E.; GOULDING, K. W.; BROWN, J. R.; GRACE, P. R.; JANZEN, H. H.; KÖRSCHENS, M. Long-term agroecosystem experiments: assessing agricultural sustainability

and global change. *Science*, American Association for the Advancement of Science, v. 282, n. 5390, p. 893–896, 1998. Citado na página 15.

RAUT, S. P.; BHALCHANDRA, A. S. Plant recognition system based on leaf image. In: *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*. [S.l.: s.n.], 2018. p. 1579–1581. Citado 3 vezes nas páginas 26, 31 e 32.

REDDY, S. R.; VARMA, G. S.; DAVULURI, R. L. Optimized convolutional neural network model for plant species identification from leaf images using computer vision. *International Journal of Speech Technology*, Springer, p. 1–28, 2021. Citado 2 vezes nas páginas 21 e 43.

REHMAN, T. U.; MAHMUD, M. S.; CHANG, Y. K.; JIN, J.; SHIN, J. Current and future applications of statistical machine learning algorithms for agricultural machine vision systems. *COMPUTERS AND ELECTRONICS IN AGRICULTURE*, 156, p. 585–605, JAN 2019. ISSN 0168-1699. Citado 3 vezes nas páginas 20, 42 e 91.

REN, C.; DULAY, J.; ROLWES, G.; PAULI, D.; SHAKOOR, N.; STYLIANOU, A. Multi-resolution outlier pooling for sorghum classification. *CoRR*, abs/2106.05748, 2021. Disponível em: <<https://arxiv.org/abs/2106.05748>>. Citado na página 43.

RIZWAN, A.; IQBAL, N.; AHMAD, R.; KIM, D.-H. Wr-svm model based on the margin radius approach for solving the minimum enclosing ball problem in support vector machine classification. *Applied Sciences*, v. 11, n. 10, 2021. ISSN 2076-3417. Disponível em: <<https://www.mdpi.com/2076-3417/11/10/4657>>. Citado na página 92.

ROTHER, C.; KOLMOGOROV, V.; BLAKE, A. "grabcut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, ACM New York, NY, USA, v. 23, n. 3, p. 309–314, 2004. Citado na página 28.

SABU, A.; SREEKUMAR, K. Literature review of image features and classifiers used in leaf based plant recognition through image analysis approach. In: *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*. [S.l.: s.n.], 2017. p. 145–149. Citado 9 vezes nas páginas 20, 26, 31, 37, 39, 40, 41, 42 e 91.

SABU, A.; SREEKUMAR, K.; NAIR, R. R. Recognition of ayurvedic medicinal plants from leaves: A computer vision approach. In: *2017 Fourth International Conference on Image Information Processing (ICIIP)*. [S.l.: s.n.], 2017. p. 1–5. Citado 4 vezes nas páginas 29, 32, 33 e 39.

SADDIK, A.; LATIF, R.; OUARDI, A. E.; ELHOSENY, M.; KHELIFI, A. Computer development based embedded systems in precision agriculture: tools and application. *Acta Agriculturae Scandinavica, Section B — Soil & Plant Science*, Taylor & Francis, v. 72, n. 1, p. 589–611, 2022. Disponível em: <<https://doi.org/10.1080/09064710.2021.2024874>>. Citado na página 49.

SADIQ, M. I.; RAHMAN, S. M. P.; KAYES, S.; SUMAITA, A. H.; CHISTY, N. A. A review on the imaging approaches in agriculture with crop and soil sensing methodologies. In: *2021 Fifth International Conference On Intelligent Computing in Data Sciences (ICDS)*. [S.l.: s.n.], 2021. p. 1–7. Citado 2 vezes nas páginas 16 e 20.

SAEED, U.; WAQAS, M.; MIRBAHAR, N.; KHUHRO, M. Comparative analysis of different operating systems for raspberry pi in terms of scheduling, synchronization, and memory management. *Mehran University Research Journal of Engineering and Technology*, v. 41, p. 113–119, 07 2022. Citado na página 49.

SALVI, J.; ARMANGUÉ, X.; BATLLE, J. A comparative review of camera calibrating methods with accuracy evaluation. *Pattern Recognition*, v. 35, n. 7, p. 1617–1635, 2002. ISSN 0031-3203. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0031320301001261>>. Citado na página 80.

SANTANA, H. A.; REZENDE, B. R.; SANTOS, W. V. d.; SILVA, A. R. d. Models for prediction of individual leaf area of forage legumes. *Revista Ceres*, Universidade Federal de Viçosa, v. 65, n. 2, p. 204–209, Mar 2018. ISSN 0034-737X. Disponível em: <<https://doi.org/10.1590/0034-737X201865020013>>. Citado na página 59.

SANTIAGO, W.; LEITE, N.; TERUEL, B.; KARKEE, M.; AZANIA, C. Evaluation of bag-of-features (bof) technique for weed management in sugarcane production. *Australian Journal of Crop Science*, v. 13, n. 11, p. 1819–1825, 2019. Cited By 0. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85081699129&doi=10.21475%2Fajcs.19.13.11.p1838&partnerID=40&md5=628e8d8212a7845290bc324aac0bf8b2>>. Citado 4 vezes nas páginas 26, 27, 32 e 38.

SCHARSTEIN, D. *View synthesis using stereo vision*. [S.l.]: Springer, 2003. Citado 2 vezes nas páginas 71 e 73.

SHENG, H.; CHEN, X.; SU, J.; RAJAGOPAL, R.; NG, A. Effective data fusion with generalized vegetation index: Evidence from land cover segmentation in agriculture. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2020. p. 267–276. ISSN 2160-7516. Citado na página 20.

SHENG, H.; WEI, S.; YU, X. Image segmentation and object measurement based on stereo vision. In: *2020 Chinese Automation Congress (CAC)*. [S.l.: s.n.], 2020. p. 3637–3641. Citado na página 16.

SILVA, G.; AZANIA, C.; NOVO, M.; WUTKE, E.; ZERA, F.; AZANIA, A. Superação da profundidade de semeadura e densidades de palha para mucuna aterrima, mucuna deeringiana e mucuna cinerea. *Planta Daninha*, Sociedade Brasileira da Ciência das Plantas Daninhas, v. 31, n. 2, p. 313–317, Apr 2013. ISSN 0100-8358. Disponível em: <<https://doi.org/10.1590/S0100-83582013000200008>>. Citado na página 59.

SÖDERKVIST, O. Computer vision classification of leaves from swedish trees. *Linköping: Linköping University*, 2001. Citado 2 vezes nas páginas 21 e 23.

SONG, R.; ZHANG, Z.; LIU, H. Edge connection based canny edge detection algorithm. *Pattern Recognition and Image Analysis*, Springer, v. 27, p. 740–747, 2017. Citado na página 88.

SORRI, K.; MUSTAFEE, N.; SEPPÄNEN, M. Revisiting iot definitions: A framework towards comprehensive use. *Technological Forecasting and Social Change*, v. 179, p. 121623, 2022. ISSN 0040-1625. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S004016252200155X>>. Citado na página 60.

SUN, J.; CHEN, X.; GONG, Z.; LIU, Z.; ZHAO, Y. Accurate camera calibration with distortion models using sphere images. *Optics & Laser Technology*, v. 65, p. 83–87, 2015. ISSN 0030-3992. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0030399214001790>>. Citado na página 81.

SZELISKI, R. *Computer vision algorithms and applications*. Springer, London; New York, 2011. Disponível em: <<http://dx.doi.org/10.1007/978-1-84882-935-0>>. Citado na página 73.

TACKENBERG, M.; VOLKMAR, C.; DAMMER, K.-H. Sensor-based variable-rate fungicide application in winter wheat. *Pest Management Science*, v. 72, n. 10, p. 1888–1896, 2016. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/ps.4225>>. Citado na página 77.

TAUGER, M. *Agriculture in World History*. London: Routledge, 2010. 1-192 p. ISBN 9781136941610. Citado na página 14.

TAVARES, A. C. F.; SILVA, F. H. dos S.; VERÇOSA, J. P. dos S.; BRAGA, M. de B. *Algoritmo baseado em Rede Neural Convolutacional para distinguir cana de açúcar de erva daninha em imagens do satélite PlanetScope*. 2022. BR 51 2022 001287 1. Instituto Nacional da Propriedade Industrial. Citado na página 45.

TOLEDO-PÉREZ, D. C.; RODRÍGUEZ-RESÉNDIZ, J.; GÓMEZ-LOENZO, R. A.; JAUREGUI-CORREA, J. C. Support vector machine-based emg signal classification techniques: A review. *Applied Sciences*, v. 9, n. 20, 2019. ISSN 2076-3417. Disponível em: <<https://www.mdpi.com/2076-3417/9/20/4402>>. Citado na página 91.

TROMBINI, M.; SOLARNA, D.; MOSER, G.; DELLEPIANE, S. A goal-driven unsupervised image segmentation method combining graph-based processing and markov random fields. *Pattern Recognition*, v. 134, p. 109082, 2023. ISSN 0031-3203. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0031320322005623>>. Citado na página 29.

TURHAL, U. C.; DAGDELEN, C. Tensor based statistical segmentation of green vegetation canopy images. *Acta Scientiarum. Technology*, v. 44, p. e55708–e55708, 2022. Citado 2 vezes nas páginas 26 e 29.

UCHIYAMA, H.; SAKURAI, S.; MISHIMA, M.; ARITA, D.; OKAYASU, T.; SHIMADA, A.; TANIGUCHI, R.-i. An easy-to-setup 3d phenotyping platform for komatsuna dataset. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*. [S.l.: s.n.], 2017. Citado 2 vezes nas páginas 22 e 24.

VINCENT, L.; SOILLE, P. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, IEEE Computer Society, v. 13, n. 06, p. 583–598, 1991. Citado na página 28.

VIT, A.; SHANI, G.; BAR-HILLEL, A. Length phenotyping with interest point detection. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2019. p. 2609–2618. ISSN 2160-7516. Citado na página 43.

WÄLDCHEN, J.; MÄDER, P. Plant species identification using computer vision techniques: A systematic literature review. *Archives of Computational Methods in Engineering*, v. 25, n. 2, p. 507–543, 2018. Cited By 108. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85008498761&doi=10.1007%2fs11831-016-9206-z&partnerID=40&md5=154d64147f701729f955b3ac19237942>>. Citado 5 vezes nas páginas 20, 21, 25, 30 e 44.

WÄLDCHEN, J.; RZANNY, M.; SEELAND, M.; MÄDER, P. Automated plant species identification—trends and future directions. *PLoS Computational Biology*, v. 14, n. 4, 2018. Cited By 69. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85046372208&doi=10.1371%2fjournal.pcbi.1005993&partnerID=40&md5=5fa80dbabd11db5fbcf987979c252f51>>. Citado 5 vezes nas páginas 16, 20, 21, 31 e 43.

- WANG, H.; TIAN, D.; LI, C.; TIAN, Y.; ZHOU, H. Plant leaf tooth feature extraction. *PLoS ONE*, v. 14, n. 2, 2019. Cited By 3. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85061497406&doi=10.1371%2fjournal.pone.0204714&partnerID=40&md5=20cdb440542db0e1688145c0ebfccbe9>>. Citado 5 vezes nas páginas 21, 22, 27, 29 e 33.
- WANG, L.; ZHENG, L.; WANG, M. 3d point cloud instance segmentation of lettuce based on partnet. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2022. p. 1646–1654. Citado na página 28.
- WANG, Y.; WANG, X.; WAN, Z.; ZHANG, J. A method for extrinsic parameter calibration of rotating binocular stereo vision using a single feature point. *Sensors*, v. 18, n. 11, 2018. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/18/11/3666>>. Citado 2 vezes nas páginas 79 e 81.
- WANG, Y. M.; LI, Y.; ZHENG, J. B. A camera calibration technique based on opencv. In: *The 3rd International Conference on Information Sciences and Interaction Sciences*. [S.l.: s.n.], 2010. p. 403–406. Citado na página 81.
- WANG, Z.; LI, H.; ZHU, Y.; XU, T. Review of plant identification based on image processing. *Archives of Computational Methods in Engineering*, v. 24, n. 3, p. 637–654, 2017. Cited By 31. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84979231126&doi=10.1007%2fs11831-016-9181-4&partnerID=40&md5=4245861ef1e5ccb4b15a915bab276c99>>. Citado 3 vezes nas páginas 20, 31 e 42.
- WU, S. G.; BAO, F. S.; XU, E. Y.; WANG, Y.-X.; CHANG, Y.-F.; XIANG, Q.-L. A leaf recognition algorithm for plant classification using probabilistic neural network. In: *IEEE. 2007 IEEE international symposium on signal processing and information technology*. [S.l.], 2007. p. 11–16. Citado 2 vezes nas páginas 21 e 22.
- NANJING FORESTRY UNIVERSITY. Jin Xiaojun, Chen Yong e Yu Jialin. *Precise lawn and pasture weeding method based on cloud weed control spectrum*. 2021. CN113349188A. China Patent Office. Citado na página 44.
- XU, M.; YOON, S.; FUENTES, A.; PARK, D. S. A comprehensive survey of image augmentation techniques for deep learning. *Pattern Recognition*, v. 137, p. 109347, 2023. ISSN 0031-3203. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0031320323000481>>. Citado na página 26.
- YANG, A. Y.; WRIGHT, J.; MA, Y.; SASTRY, S. S. Feature selection in face recognition: A sparse representation perspective. *submitted to IEEE Transactions Pattern Analysis and Machine Intelligence*, v. 2, 2007. Citado na página 39.
- YANG, C. Plant leaf recognition by integrating shape and texture features. *Pattern Recognition*, v. 112, 2021. Cited By 0. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85099478543&doi=10.1016%2fj.patcog.2020.107809&partnerID=40&md5=b0e8efbb8fe5fada80171a23a489b260>>. Citado 4 vezes nas páginas 21, 33, 35 e 39.
- YANG, L.; WANG, B.; ZHANG, R.; ZHOU, H.; WANG, R. Analysis on location accuracy for the binocular stereo vision system. *IEEE Photonics Journal*, v. 10, n. 1, p. 1–16, 2018. Citado 2 vezes nas páginas 80 e 81.

YANG, S.; GAO, Y.; LIU, Z.; ZHANG, G. A calibration method for binocular stereo vision sensor with short-baseline based on 3d flexible control field. *Optics and Lasers in Engineering*, v. 124, p. 105817, 2020. ISSN 0143-8166. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0143816619301691>>. Citado na página 81.

ZHANG, D.; LU, G. Shape-based image retrieval using generic fourier descriptor. *Signal Processing: Image Communication*, v. 17, n. 10, p. 825–848, 2002. ISSN 0923-5965. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S092359650200084X>>. Citado na página 33.

ZHANG, Q.; ZENG, S.; ZHANG, B. Initial investigation of different classifiers for plant leaf classification using multiple features. In: . [s.n.], 2019. v. 11179. Cited By 1. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85072607246&doi=10.1117%2f12.2539654&partnerID=40&md5=e663961d21c8dc39cf703ce56e63729f>>. Citado 4 vezes nas páginas 38, 39, 41 e 42.

ZHANG, S.; HUANG, W.; WANG, Z. Combing modified grabcut, k-means clustering and sparse representation classification for weed recognition in wheat field. *Neurocomputing*, 2021. Cited By 0. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85100405110&doi=10.1016%2fj.neucom.2020.06.140&partnerID=40&md5=1bea60a193abfc5e574ec82ac3a7117a>>. Citado 4 vezes nas páginas 26, 28, 36 e 39.

ZHANG, S.; WU, X.; YOU, Z. Jaccard distance based weighted sparse representation for coarse-to-fine plant species recognition. *PLoS ONE*, v. 12, n. 6, 2017. Cited By 7. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85020304208&doi=10.1371%2fjournal.pone.0178317&partnerID=40&md5=f553c0a348038130bafaba476d47551a>>. Citado 4 vezes nas páginas 22, 26, 30 e 40.

ZHANG, S.; ZHANG, C.; ZHU, Y.; YOU, Z. Discriminant wsrc for large-scale plant species recognition. *Computational Intelligence and Neuroscience*, v. 2017, 2017. Cited By 5. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85039899492&doi=10.1155%2f2017%2f9581292&partnerID=40&md5=4660df619200e2f2866a15b53f982e36>>. Citado 5 vezes nas páginas 22, 26, 28, 30 e 40.

ZHANG, Y.; CUI, J.; WANG, Z.; KANG, J.; MIN, Y. Leaf image recognition based on bag of features. *Applied Sciences (Switzerland)*, v. 10, n. 15, 2020. Cited By 2. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85088800136&doi=10.3390%2fapp10155177&partnerID=40&md5=85e74db2f7bb1a476eced1724457713b>>. Citado 8 vezes nas páginas 21, 22, 25, 26, 30, 34, 35 e 38.

ZHANG, Y. J. A review of recent evaluation methods for image segmentation. In: *Proceedings of the Sixth International Symposium on Signal Processing and its Applications (Cat.No.01EX467)*. [S.l.: s.n.], 2001. v. 1, p. 148–151 vol.1. Citado na página 86.

JIANGSU HUAYUAN WATER SAVING CO LTD. Qiu Zhipeng, Zhang Jinxiang, Wang Houqun e Zhang Jiangbo. *Field weeding machine*. 2022. CN114424704A. China Patent Office. Citado na página 44.

ZHOU, K.; MENG, X.; CHENG, B. Review of stereo matching algorithms based on deep learning. *Computational intelligence and neuroscience*, Hindawi, v. 2020, 2020. Citado 3 vezes nas páginas 73, 74 e 75.

ZHU, N.; LIU, X.; LIU, Z.; HU, K.; WANG, Y.; TAN, J.; HUANG, M.; ZHU, Q.; JI, X.; JIANG, Y.; GUO, Y. Deep learning for smart agriculture: Concepts, tools, applications, and opportunities. *International Journal of Agricultural and Biological Engineering*, v. 11, n. 4, p. 32–44, 2018. Cited By 30. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85051454829&doi=10.25165%2Fijabe.v11i4.4475&partnerID=40&md5=f08dc039304f438873913484de15d25f>>. Citado na página 43.

ZIELINSKI, K. M. C.; RIBAS, L. C.; SCABINI, L. F. S.; BRUNO, O. M. Complex texture features learned by applying randomized neural network on graphs. In: *2022 Eleventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*. [S.l.: s.n.], 2022. p. 1–6. Citado na página 43.

ZWANENBERG, O. van; TRIANTAPHILLIDOU, S.; JENKIN, R.; PSARROU, A. Edge detection techniques for quantifying spatial imaging system performance and image quality. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2019. p. 1871–1879. Citado na página 63.