



UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DE COMPUTAÇÃO

FLÁVIO JOSÉ DE MELO SOUZA

**Fenotipagem de Plantas sob Estresse Hídrico Utilizando Características  
Derivadas de Imagens**

Recife

2025

FLÁVIO JOSÉ DE MELO SOUZA

**Fenotipagem de Plantas sob Estresse Hídrico Utilizando Características  
Derivadas de Imagens**

Dissertação apresentada ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

**Área de Concentração:** Sistemas Embarcados

**Orientador:** Prof. Dr. Stefan Michael Blawid

Recife

2025

.Catalogação de Publicação na Fonte. UFPE - Biblioteca Central

Souza, Flávio José de Melo.

Fenotipagem de plantas sob estresse hídrico utilizando características derivadas de imagens / Flávio José de Melo Souza. - Recife, 2025.

154f.: il.

Dissertação (Mestrado)- Universidade Federal de Pernambuco, Centro de Informática, Programa de Pós-Graduação em Ciência da Computação, 2025.

Orientação: Stefan Michael Blawid.

1. Visão de máquina na agricultura; 2. Monitoramento remoto de umidade do solo; 3. Imagens RGB; 4. Irrigação de precisão; 5. Detecção de estresse hídrico. I. Blawid, Stefan Michael. II. Título.

UFPE-Biblioteca Central

**Flávio José de Melo Souza**

**“Fenotipagem de Plantas sob Estresse Hídrico  
Utilizando Características Derivadas de Imagens”**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação. Área de Concentração: Engenharia da Computação.

Aprovado em: 18/07/2025.

**BANCA EXAMINADORA**

---

Prof. Dr. Francisco Paulo Magalhães Simões  
Centro de Informática / UFPE

---

Prof. Dr. Antonio Samuel Alves da Silva  
Departamento de Estatística e Informática/UFRPE

---

Prof. Dr. Stefan Michael Blawid  
Centro de Informática / UFPE  
(orientador)



Dedico esta obra à Jacqueline e ao nossos filhos, Nicolas e Thomas.

## **AGRADECIMENTOS**

Primeiramente, ao meu amado Pai, o grande Designer, de quem tenho o suporte necessário para continuar no Caminho.

À minha esposa e amiga, Jacqueline Souza, por sempre me encorajar e acompanhar em todos os desafios e aventuras que a vida impõe. Sou grato pelos nossos preciosos filhos Nicolás e Thomas Souza, excelentes companheiros.

À minha mãe Maria Luísa, por ter exercido sua tarefa com excelência, me dando exemplo com sua vida sobre moral e ética, e me estimulando aos estudos. Evidência concreta de que o nosso futuro depende de nossas próprias ações.

Quanto aos meus irmãos, os estimo pelas diversas formas de apoio. Dentre diversos aspectos, sou grato, em especial, ao Rober e Sandra Oliveira, João e Vânia Costa e Alex e Suelene Barros. Agradeço o apoio, mesmo que de longe, de amigos como Mateus Gonçalves e Laércio Leal. À parte da Igreja que se reúne em Teutônia (RS) e Olinda (PE), por todo o suporte quando mais precisamos.

Ao Professor Stefan Michael Blawid, por ser o orientador que esperava ter, excelente colaboração em todo andamento da pesquisa, contribuindo na construção de um tema de pesquisa fascinante. Agradeço, igualmente, à Professora Rosana Blawid pelo valioso apoio prestado em momentos decisivos desta pesquisa. Agradeço também ao Centro de Informática (CIn) por proporcionar oportunidades e infraestrutura que permitem formação de excelência.

*"Será como árvore plantada junto às águas, que estende as suas raízes para o ribeiro e não receia quando vem o calor, mas suas folhas ficam sempre verdes; e, no ano da seca, não se perturba, nem deixa de dar fruto."*  
– Yirmeyahu (Jeremias)

## RESUMO

A irrigação de precisão em ambientes controlados exige o monitoramento contínuo e remoto do solo, mas métodos tradicionais, baseados em medições pontuais, enfrentam limitações de abrangência e durabilidade dos sensores. Como alternativa, esta dissertação apresenta o desenvolvimento e validação de um sistema inteligente de fenotipagem, fundamentado em imagens RGB e aprendizado de máquina embarcado em microcontroladores de baixo custo. Os experimentos demonstraram que alterações na coloração foliar podem ser utilizadas como indicativos sensíveis do estado hídrico, permitindo antecipar intervenções de irrigação com maior precisão. Foram avaliadas diferentes estratégias de classificação de estresse hídrico: (i) análise da componente Saturação, destacando-se como a solução mais simples e eficiente para implementação em dispositivos com recursos limitados; (ii) modelos rasos baseados em textura, que apresentaram desempenho promissor; e (iii) arquiteturas profundas do tipo CNN, que exigem maior poder computacional. Os resultados confirmam a viabilidade de descentralizar o processamento em dispositivos embarcados (Edge Computing), reduzindo a dependência de infraestrutura de nuvem, e evidenciam o potencial da visão computacional como ferramenta para o monitoramento agrícola de baixo custo.

**Palavras-chaves:** Visão de máquina na agricultura; Monitoramento remoto de umidade do solo; Imagens RGB; Irrigação de precisão; Detecção de estresse hídrico

## ABSTRACT

Precision irrigation in controlled environments requires continuous and remote soil monitoring, but traditional methods, based on point measurements, face limitations in sensor coverage and durability. As an alternative, this dissertation presents the development and validation of an intelligent phenotyping system, based on RGB images and embedded machine learning on low-cost microcontrollers. The experiments demonstrated that changes in leaf color can be used as sensitive indicators of water status, allowing irrigation interventions to be anticipated with greater precision. Different water stress classification strategies were evaluated: (i) analysis of the H color component, which stood out as the simplest and most efficient solution for implementation on devices with limited resources; (ii) shallow texture-based models, which showed promising performance; and (iii) deep CNN architectures, which require greater computational power. The results confirm the feasibility of decentralizing processing on embedded devices (Edge Computing), reducing dependence on cloud infrastructure, and highlight the potential of computer vision as a tool for low-cost agricultural monitoring.

**Keywords:** Machine vision in agriculture; Remote soil moisture monitoring; RGB images; Precision irrigation; Water stress detection

## LISTA DE FIGURAS

Figura 1 – Fazenda de alface em Encinitas, CA, EUA . . . . .	21
Figura 2 – Fenotipagem no Donald Danforth Plant Science Center, MO, EUA. .	22
Figura 3 – Detecção de movimento capturado por quadros consecutivos . . . .	29
Figura 4 – Ilustração do cálculo da matriz GLCM . . . . .	32
Figura 5 – Medidas estatísticas derivadas da GLCM diferenciando regiões em uma imagem de exemplo . . . . .	33
Figura 6 – Tarefas relacionadas ao reconhecimento de objetos . . . . .	34
Figura 7 – Técnicas de classificação de aprendizagem supervisionada . . . . .	36
Figura 8 – Hierarquia das abordagens de inteligência artificial . . . . .	36
Figura 9 – Uma rede neural artificial . . . . .	37
Figura 10 – Uma operação de convolução . . . . .	41
Figura 11 – Funções de ativação comumente aplicadas a redes neurais . . . . .	42
Figura 12 – Uma operação de Max pooling . . . . .	43
Figura 13 – Treinamento de uma rede neural convolucional . . . . .	45
Figura 14 – Arquitetura geral da MobileNetV1 . . . . .	49
Figura 15 – Triagem e seleção de tomates baseada em visão computacional . .	56
Figura 16 – Inkbird ITC608T - Medições de temperatura e umidade . . . . .	57
Figura 17 – Inkbird IIC-800-WIFI - Irrigação programada . . . . .	57
Figura 18 – CloudlineT4 - Ventilação . . . . .	58
Figura 19 – Argus Titan - Sistema holístico . . . . .	58
Figura 20 – Bluelab OnePen - Monitoramento do solo . . . . .	59
Figura 21 – AS7341 Adafruit - Monitoramento de iluminação . . . . .	59
Figura 22 – Raspberry Pi e Módulo Câmera . . . . .	60
Figura 23 – FarmBot . . . . .	61
Figura 24 – Sistema AgOpenGPS . . . . .	61
Figura 25 – Experimento de referência para absorção de água: Imagens de folhas	65
Figura 26 – Experimento de referência para absorção de água: Valores RGB . .	65
Figura 27 – Fenotipagem NIR para detecção de água em plantas . . . . .	67
Figura 28 – O sistema de fenotipagem desenvolvido - SPheRe . . . . .	72
Figura 29 – Diagrama em Blocos do SPheRe . . . . .	74

Figura 30 – Seeed Studio XIAO ESP32S3 Sense . . . . .	75
Figura 31 – Sensor capacitivo de umidade de solo . . . . .	75
Figura 32 – Sistema de fenotipagem na tenda de crescimento . . . . .	76
Figura 33 – Calibração do sensor de solo . . . . .	77
Figura 34 – Planta sob estudo (alface estiolado) . . . . .	78
Figura 35 – VSCode com PlatformIO . . . . .	79
Figura 36 – Google Colab . . . . .	79
Figura 37 – Ambiente de desenvolvimento em nuvem do Google Apps Script . .	81
Figura 38 – Planilha exibindo dados, gráfico e interface de controle do SPheRe .	81
Figura 39 – Trecho do conjunto de imagens armazenado na nuvem . . . . .	82
Figura 40 – Imagem da planta com região de referência . . . . .	83
Figura 41 – Pipeline PDI . . . . .	85
Figura 42 – Pipeline para extração de dados . . . . .	86
Figura 43 – Pipeline para classificação via arquiteturas rasas . . . . .	87
Figura 44 – Pipeline para classificação via arquitetura profunda . . . . .	87
Figura 45 – Cenário Nuvem versus Edge-Computing . . . . .	88
Figura 46 – Ciclos de irrigação para aquisição de imagens . . . . .	89
Figura 47 – Amostra do dataset do experimento . . . . .	89
Figura 48 – Sensor ajustado . . . . .	90
Figura 49 – Tríade do SPheRe . . . . .	92
Figura 50 – Diagrama de estados do SPheRe . . . . .	93
Figura 51 – Painel de interação do usuário para irrigação e definição de limiares	94
Figura 52 – Esquema elétrico e PCI da placa auxiliar montada . . . . .	96
Figura 53 – Parâmetros ambientais medidos dentro do SPheRe . . . . .	96
Figura 54 – Módulo para medições de dados ambientais na tenda . . . . .	97
Figura 55 – Imagens capturadas e segmentadas antes e depois da irrigação . .	98
Figura 56 – Valores RGB médios durante dois ciclos de irrigação . . . . .	99
Figura 57 – Imagens capturadas em momentos distintos da série experimental	
(parte 1) . . . . .	100
Figura 58 – Imagens capturadas em momentos distintos da série experimental	
(parte 2) . . . . .	101
Figura 59 – Intensidade da luz durante a coleta de imagens . . . . .	103
Figura 60 – RGB médio versus umidade . . . . .	103

Figura 61 – Valores HSV médios durante dois ciclos de irrigação . . . . .	104
Figura 62 – HSV médio versus umidade . . . . .	104
Figura 63 – Média móvel do canal S durante dois ciclos de irrigação . . . . .	105
Figura 64 – Desvio padrão RGB durante dois ciclos de irrigação . . . . .	106
Figura 65 – Desvio padrão HSV durante dois ciclos de irrigação . . . . .	106
Figura 66 – Desvios Padrão RGB e HSV versus umidade . . . . .	106
Figura 67 – Movimentação da planta durante dois ciclos de irrigação . . . . .	107
Figura 68 – Subtração de imagens para detecção de movimento . . . . .	108
Figura 69 – Imagem segmentada em escala de cinza para análise de textura . .	109
Figura 70 – Matriz de correlação cruzada entre as características de Haralick . .	109
Figura 71 – Boxplot das características de Haralick . . . . .	110
Figura 72 – Contraste durante o primeiro ciclo de irrigação . . . . .	111
Figura 73 – Imagens de plantas evidenciando a mudança de contraste após a irrigação . . . . .	111
Figura 74 – Dissimilaridade durante o primeiro ciclo de irrigação . . . . .	112
Figura 75 – <i>contrast</i> e <i>dissimilarity</i> versus umidade do solo . . . . .	112
Figura 76 – Outras características de textura durante o primeiro ciclo de irrigação	113
Figura 77 – Matriz de confusão do classificador de textura da árvore de decisão	116
Figura 78 – Matriz de confusão do classificador de textura de floresta aleatória .	116
Figura 79 – Matriz de confusão do classificador de textura da KNN . . . . .	117
Figura 80 – Matriz de confusão do classificador de imagens baseado em Mobi- leNetV3Small . . . . .	119
Figura 81 – Sensor de imagem bidimensional . . . . .	133
Figura 82 – Processo de aquisição de imagem digital . . . . .	134
Figura 83 – Operação de filtragem diretamente na imagem . . . . .	135
Figura 84 – Aplicando um filtro de suavização a uma imagem . . . . .	138
Figura 85 – Kernel Gaussiano para filtragem . . . . .	138
Figura 86 – Aplicando um filtro Gaussiano a uma imagem . . . . .	139
Figura 87 – Redução do ruído de impulso por filtragem . . . . .	139
Figura 88 – Cubo de cores RGB . . . . .	143
Figura 89 – Outra perspectiva do cubo RGB . . . . .	144
Figura 90 – Espaço de cores HSV . . . . .	145
Figura 91 – Exemplo de um histograma de imagem com um único limiar . . . . .	147



Figura 92 – Uma imagem de exemplo real e seu histograma . . . . .	149
Figura 93 – Segmentação de imagens baseada em dois algoritmos diferentes .	149
Figura 94 – Segmentação aplicada neste trabalho . . . . .	149
Figura 95 – Erosão para remoção de componentes de imagem . . . . .	153
Figura 96 – Dilatação para fundir componentes de imagem quebrados . . . . .	153

## LISTA DE QUADROS

Quadro 1 – Distinção entre PDI e Visão Computacional . . . . .	27
Quadro 2 – Matriz de confusão para classificação binária. . . . .	50
Quadro 3 – Descrição das anomalias identificadas nas curvas RGB . . . . .	102

## LISTA DE TABELAS

Tabela 1 – Estatísticas descritivas das variáveis de textura e umidade . . . . .	110
Tabela 2 – Especificação do cliente para o desenvolvimento em nuvem de modelos de ML . . . . .	114
Tabela 3 – Correlação cruzada entre as características de Haralick e a variável <i>alert</i> . . . . .	115
Tabela 4 – Camadas e parâmetros meta do modelo MobileNetV3Small customizado. . . . .	118
Tabela 5 – Medidas de desempenho de diferentes classificadores de textura .	118
Tabela 6 – Principais recursos de hardware da ESP32-S3 . . . . .	119
Tabela 7 – Número de parâmetros treinados e não treinados do modelo MobileNetV3Small customizado . . . . .	120
Tabela 8 – Consumo de memória flash e tempo de execução do pipeline na ESP32-S3. . . . .	121

## LISTA DE ABREVIATURAS E SIGLAS

<b>CMY</b>	Cyan, Magenta, and Yellow
<b>CMYK</b>	Cyan, Magenta, Yellow, and Key (Black)
<b>CNC</b>	Computer Numerical Control
<b>CNN</b>	Convolutional Neural Network
<b>CPU</b>	Central Processing Unit
<b>DIY</b>	Do It Yourself
<b>DL</b>	Deep Learning
<b>DNN</b>	Deep Neural Network
<b>DSP</b>	Digital Signal Processor
<b>DSS</b>	Decision Support System
<b>EC</b>	Electrical Conductivity
<b>FDR</b>	Frequency Domain Reflectometry
<b>FSM</b>	Finite State Machine
<b>GLCM</b>	Gray-Level Co-Occurrence Matrix
<b>GPS</b>	Global Positioning System
<b>GPU</b>	Graphics-Processing Unit
<b>HSV</b>	Hue, Saturation, and Value
<b>HTTP</b>	Hypertext Transfer Protocol
<b>IA</b>	Inteligência Artificial
<b>IDE</b>	Integrated Development Environment
<b>IoT</b>	Internet of Things
<b>JPEG</b>	Joint Photographic Experts Group
<b>KNN</b>	K-Nearest Neighbors
<b>LED</b>	Light-Emitting Diode
<b>M2M</b>	Machine-to-Machine

<b>MAE</b>	Mean Absolute Error
<b>ML</b>	Machine Learning
<b>MLP</b>	Multi-Layer Perceptron
<b>NIR</b>	Near InfraRed
<b>PCI</b>	Placa de Circuito Impresso
<b>PDI</b>	Processamento Digital de Imagens
<b>pH</b>	Potential of Hydrogen
<b>RGB</b>	Red, Green, and Blue
<b>RNA</b>	Redes Neurais Artificiais
<b>ROI</b>	Region Of Interest
<b>SGD</b>	Stochastic Gradient Descent
<b>SoC</b>	System-on-Chip
<b>SPAD</b>	Soil Plant Analysis Development
<b>SPheRe</b>	Smart Phenotyping Research Lab
<b>SVM</b>	Support Vector Machine
<b>SWR</b>	Standing Wave Ratio
<b>TDR</b>	Time Domain Reflectometry
<b>TPU</b>	Tensor Processing Unit
<b>UIB</b>	Universal Inverted Bottleneck

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>21</b>
1.1	OBJETIVO	23
1.2	JUSTIFICATIVA	24
1.3	ORGANIZAÇÃO DO DOCUMENTO	25
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>26</b>
2.1	A VISÃO HUMANA FRENTE A VISÃO DE MÁQUINA	26
2.2	O PROCESSAMENTO DIGITAL DE IMAGENS E A VISÃO COMPUTACIONAL	27
<b>2.2.1</b>	<b>Deteccção de Movimento Entre Imagens</b>	<b>28</b>
<b>2.2.2</b>	<b>Análise de Textura</b>	<b>29</b>
2.2.2.1	Construção da GLCM	31
2.2.2.2	Aplicações	33
2.2.2.3	Técnicas modernas de classificação de texturas	33
2.3	APRENDIZAGEM DE MÁQUINA E CLASSIFICAÇÃO DE IMAGENS	34
<b>2.3.1</b>	<b>Redes Neurais e Aprendizagem Profunda</b>	<b>36</b>
<b>2.3.2</b>	<b>Redes Neurais Convolucionais</b>	<b>39</b>
2.3.2.1	Camada Convolucional	40
2.3.2.2	Função de Ativação	42
2.3.2.3	Camada de Subamostragem	42
2.3.2.4	Camada Densa	43
<b>2.3.3</b>	<b>Treinamento de uma CNN</b>	<b>44</b>
<b>2.3.4</b>	<b>CNNs e Classificação de Imagens</b>	<b>47</b>
2.3.4.1	Arquitetura MobileNet	48
<b>2.3.5</b>	<b>Métricas de Avaliação</b>	<b>48</b>
2.3.5.1	Matriz de Confusão	49
2.3.5.2	Acurácia	50
2.3.5.3	Precisão para Classe Positiva	50
2.3.5.4	Precisão para Classe Negativa	51
2.3.5.5	Recall	51
2.3.5.6	F1-Score	51

<b>3</b>	<b>REVISÃO DA LITERATURA: AGRICULTURA DE PRECISÃO . . . .</b>	<b>53</b>
3.1	A TELEMETRIA NA AGRICULTURA . . . . .	53
3.2	O PROTOCOLO HTTP APLICADO NA AGRICULTURA . . . . .	54
3.3	VISÃO COMPUTACIONAL APLICADA NA AGRICULTURA . . . . .	54
3.4	SOLUÇÕES DISPONÍVEIS NA AGRICULTURA DE PRECISÃO . . .	57
<b>3.4.1</b>	<b>Soluções Comerciais . . . . .</b>	<b>57</b>
<b>3.4.2</b>	<b>Soluções Open Source . . . . .</b>	<b>60</b>
3.4.2.1	Sistemas Baseados em Microcontroladores . . . . .	60
3.4.2.2	FarmBot . . . . .	60
3.4.2.3	AgOpenGPS . . . . .	61
3.4.2.4	Plataformas IoT Colaborativas . . . . .	61
<b>4</b>	<b>REVISÃO DA LITERATURA: TRABALHOS RELACIONADOS . . .</b>	<b>63</b>
4.1	AVANÇOS EM TECNOLOGIAS DE SENSORES DE UMIDADE DO SOLO . . . . .	63
4.2	ESTIMATIVA DA ABSORÇÃO DE ÁGUA DE IRRIGAÇÃO COM RGB	64
4.3	VISÃO COMPUTACIONAL PARA FENOTIPAGEM DE CARACTERÍSTICAS DE PLANTAS . . . . .	66
4.4	PREVISÃO FENOTÍPICA DO STATUS HÍDRICO DO SUBSTRATO .	67
4.5	MODELOS DE IMAGENS COM SMARTPHONES PARA ESTIMATIVA DE CLOROFILA . . . . .	68
4.6	ABORDAGENS DE APRENDIZADO PROFUNDO PARA AVALIAÇÃO DE CLOROFILA FOLIAR . . . . .	68
4.7	IMAGENS ESPECTRAIS E RGB COM IA PARA STATUS HÍDRICO DE PLANTAS . . . . .	69
4.8	RESUMO DO CAPÍTULO . . . . .	70
<b>5</b>	<b>MATERIAIS E MÉTODOS . . . . .</b>	<b>71</b>
5.1	VISÃO GERAL DA METODOLOGIA . . . . .	72
5.2	ARQUITETURA GERAL DO SPHERE . . . . .	73
5.3	COMPONENTES DE HARDWARE UTILIZADOS . . . . .	75
5.4	CONTEXTO EXPERIMENTAL . . . . .	76
5.5	PSEUDO-CALIBRAÇÃO DO SENSOR DE UMIDADE . . . . .	77
5.6	PLANTA ALVO E ILUMINAÇÃO . . . . .	78
5.7	A PLATAFORMA DE DESENVOLVIMENTO E O FIRMWARE . . . .	78

5.8	A PLATAFORMA DE PROCESSAMENTO DE IMAGENS . . . . .	79
5.9	ARMAZENAMENTO DOS DADOS EM SERVIDOR EM NUVEM . . .	80
<b>5.9.1</b>	<b>Comunicação entre SPheRe e Google Sheets . . . . .</b>	<b>80</b>
<b>5.9.2</b>	<b>Comunicação entre SPheRe e Drive . . . . .</b>	<b>80</b>
5.10	PROCESSAMENTO E ANÁLISE DAS IMAGENS RGB . . . . .	82
<b>5.10.1</b>	<b>Segmentação das Folhas . . . . .</b>	<b>82</b>
5.10.1.1	Remoção de Interferência Luminosa . . . . .	82
5.10.1.2	Suavização da Imagem . . . . .	83
5.10.1.3	Segmentação Baseada em Cor (Verde) . . . . .	83
5.10.1.4	Segmentação por Limiarização . . . . .	84
5.10.1.5	Recorte Final das Folhas . . . . .	84
<b>5.10.2</b>	<b>Extração de Dados Estatísticos . . . . .</b>	<b>84</b>
5.11	ANÁLISE DE ESPAÇOS DE COR EM FUNÇÃO DA UMIDADE . . .	85
5.12	CLASSIFICAÇÃO POR MEIO DE ARQUITETURAS RASAS . . . . .	85
5.13	VISÃO GERAL DOS EXPERIMENTOS . . . . .	86
5.14	PREPARAÇÃO DOS DADOS . . . . .	88
5.15	DESAFIOS DE OTIMIZAÇÃO DO SETUP . . . . .	90
<b>5.15.1</b>	<b>Sensor de umidade . . . . .</b>	<b>90</b>
<b>5.15.2</b>	<b>Iluminação artificial . . . . .</b>	<b>90</b>
<b>5.15.3</b>	<b>Reservatório de água . . . . .</b>	<b>90</b>
<b>5.15.4</b>	<b>Pintura do suporte . . . . .</b>	<b>91</b>
<b>5.15.5</b>	<b>Configuração da câmera . . . . .</b>	<b>91</b>
<b>5.15.6</b>	<b>Bomba de irrigação . . . . .</b>	<b>91</b>
5.16	CONSIDERAÇÕES METODOLÓGICAS FINAIS . . . . .	91
<b>6</b>	<b>RESULTADOS: IMPLEMENTAÇÃO . . . . .</b>	<b>92</b>
6.1	ESTRUTURAMENTO DO FIRMWARE . . . . .	92
6.2	LÓGICA DE FUNCIONAMENTO DO SISTEMA . . . . .	93
6.3	CONFIGURAÇÃO DA CAPTURA DE IMAGEM E COMPRESSÃO . .	95
6.4	GERENCIAMENTO DE ENERGIA . . . . .	95
6.5	HARDWARE PARA ACIONAMENTO DA MINI BOMBA D'ÁGUA . . .	95
6.6	MÓDULO PARA COLETA DE DADOS AMBIENTAIS . . . . .	96
6.7	REPOSITÓRIO DO PROJETO . . . . .	97
<b>7</b>	<b>RESULTADOS: PROVAS DE CONCEITO . . . . .</b>	<b>98</b>



7.1	ANÁLISE DE ESPAÇOS DE COR EM FUNÇÃO DA UMIDADE . . . .	98
7.1.1	<b>Médias dos Canais . . . . .</b>	<b>99</b>
7.1.2	<b>Desvios Padrão dos Canais . . . . .</b>	<b>105</b>
7.2	ÍNDICE DE TROPISMO . . . . .	107
7.3	ANÁLISE DE TEXTURA . . . . .	108
7.4	RUMO AOS FUTUROS SENSORES DE IRRIGAÇÃO INTELIGENTES	113
7.4.1	<b>Classificação por meio de Arquiteturas Rasas . . . . .</b>	<b>114</b>
7.4.2	<b>Classificação por meio de Arquiteturas Profundas . . . . .</b>	<b>117</b>
7.4.3	<b><i>Edge Computing</i> . . . . .</b>	<b>119</b>
7.5	DISCUSSÃO . . . . .	121
8	<b>CONCLUSÃO . . . . .</b>	<b>123</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>125</b>
	<b>ANEXO A – AQUISIÇÃO DE IMAGENS COM SENSOR 2D . . . .</b>	<b>133</b>
	<b>ANEXO B – FUNDAMENTOS DA FILTRAGEM ESPACIAL . . . .</b>	<b>135</b>
	<b>ANEXO C – COMPRESSÃO DE IMAGENS . . . . .</b>	<b>141</b>
	<b>ANEXO D – SISTEMAS DE COR . . . . .</b>	<b>143</b>
	<b>ANEXO E – SEGMENTAÇÃO . . . . .</b>	<b>147</b>
	<b>ANEXO F – OPERAÇÕES ARITMÉTICAS . . . . .</b>	<b>151</b>
	<b>ANEXO G – PROCESSAMENTO MORFOLÓGICO . . . . .</b>	<b>152</b>

## 1 INTRODUÇÃO

Estufas modernas evoluíram para fábricas de plantas, permitindo a agricultura urbana e aprimorando as capacidades de produção de alimentos (SHAMSHIRI et al., 2018). Sistemas de Suporte à Decisão baseados em dados (do inglês DSS) são ferramentas essenciais para atingir alta produtividade e sustentabilidade (AIELLO et al., 2018). Entre esses sistemas, a irrigação automatizada desempenha um papel crítico. Sistemas de irrigação automatizados em estufas promovem a saúde das plantas, otimizam o uso da água e maximizam os rendimentos das colheitas. Essa abordagem garante que a quantidade precisa de água seja fornecida às plantas conforme necessário, evitando assim o desperdício e contribuindo para a sustentabilidade ambiental. Além disso, a irrigação precisa é um requisito essencial para estudos fenotípicos, que envolvem a avaliação das características observáveis das plantas resultantes da interação entre seus fatores genéticos e ambientais, em ambientes controlados.

Nessa perspectiva, fazendas-fábricas similares à Go Green, Fig. 1, aplicam abordagens inovadoras na produção agrícola sustentável. A empresa utiliza um sistema de irrigação que consome 80% menos água para cultivar alface, e produz 25 vezes mais cabeças de alface por hectare do que os métodos tradicionais. Robótica e automação complementam as atividades da empresa (ARGUS, 2023).



Figura 1 – Fazenda de alface em Encinitas, CA, EUA. Fonte: Argus (2023).

A irrigação autônoma fornece suprimento de água adequado e personalizado para cada espécie cultivada, evitando tanto a irrigação insuficiente quanto a excessiva, o que pode levar ao desenvolvimento de fungos e outras doenças vegetais. O monitoramento adequado ajuda a manter as condições ideais dentro da estufa, reduzindo

esses riscos. O objetivo de uma estufa inteligente é garantir o controle preciso da irrigação e criar um microambiente favorável para as plantas, o que geralmente é impossível de garantir em ambientes externos.

Os dados coletados por sensores ambientais permitem manter a estabilidade climática dentro da estufa, o que é essencial para plantas sensíveis a mudanças repentinas de umidade e temperatura. A agricultura de precisão baseada em sensores minimiza a intervenção humana e promove o monitoramento em tempo real, permitindo intervenções imediatas com base nas condições predominantes do solo e do ambiente. A automação da irrigação e a telemetria da cultura beneficiam tanto os produtores quanto a pesquisa científica. A irrigação automatizada melhora a eficiência do uso da água (GOUMOPOULOS et al., 2014), economiza mão de obra (CHAPPELL et al., 2013; WHEELER et al., 2018) e permite estudos de fisiologia do estresse (NEMALI; IERSEL, 2006).



Figura 2 – Imagens do sistema de fenotipagem vegetal da *Donald Danforth Plant Science Center*, MO, EUA: (a) estufa de crescimento em laboratório de fenotipagem com, esteiras, sensores, irrigação e pesagem automatizada, (b) extração de características fenotípicas por imagem em câmara controlada. Fonte: Danforth (2025).

A fenotipagem vegetal é fundamental para compreender como as plantas reagem ao ambiente. O laboratório de pesquisa *Bellwether Foundation Phenotyping Facility*, Fig. 2, do *Donald Danforth Plant Science Center*, dispõe de uma instalação pioneira nos EUA que permite aos cientistas monitorar características como tamanho das folhas, estrutura das raízes e taxa de crescimento, coletando grandes volumes de dados (DANFORTH, 2025). Essas informações são utilizadas para desenvolver culturas mais sustentáveis e resistentes a condições adversas, como seca e calor extremo. O centro de pesquisa oferece fenotipagem em ambiente controlado para caracterizar plantas

em diferentes ambientes, tratamentos e genética. Os resultados incluem informações relacionadas ao crescimento e desenvolvimento das plantas e às respostas das plantas ao estresse.

Nesse contexto de pesquisa científica, propomos um sistema de monitoramento construído com componentes de baixo custo, alinhado à proposta de Ferrarezi et al. (2015). Esses autores desenvolveram um sistema automatizado de irrigação para cultivos em vasos, baseado em microcontrolador, sensores capacitivos de umidade do solo e uma bomba de água submersa, visando reduzir custos e facilitar a adoção em ambientes controlados e por pequenos produtores. O estudo demonstrou que é possível realizar o manejo hídrico de forma precisa e com baixo custo, sem a necessidade de infraestrutura complexa, representando um avanço importante para o uso de tecnologias acessíveis na agricultura de precisão. Inspirado nesse conceito, o sistema aqui proposto incorpora também uma câmera RGB para análise fenotípica, expandindo a abordagem original para além do monitoramento pontual da umidade, a fim de integrar informações visuais no processo de decisão da irrigação.

## 1.1 OBJETIVO

Desenvolver e validar um protótipo de sistema de fenotipagem inteligente, capaz de monitorar o estresse hídrico em plantas por meio de imagens e sensores de umidade, em ambiente controlado. Para atingir esse propósito, foi concebido o SPheRe (*Smart Phenotyping Research Lab* – Laboratório de Pesquisa em Fenotipagem Inteligente), uma plataforma que integra o controle da umidade do solo, telemetria e técnicas de visão computacional sobre imagens de plantas cultivadas em uma tenda de crescimento, destinada a estudos fenotípicos diante de variações hídricas do sistema solo-planta.

Com esse propósito, as etapas a seguir devem ser realizadas:

1. Instalar um sistema de irrigação automático em uma tenda de cultivo;
2. Projetar um sistema de captura de imagens;
3. Calibrar o sistema de sensoriamento de umidade do solo;

4. Estabelecer comunicação em nuvem para fim de telemetria e armazenamento de imagens;
5. Desenvolver software para medir índices RGB de imagens de plantas sob cultivo, e mensurar outros fatores pertinentes;
6. Conduzir experimentos de estresse hídrico;
7. Fornecer uma prova de princípio de que a descoloração das folhas, é, ou não, um indicador de estresse hídrico;
8. Investigar outros fatores correlacionados com o estresse hídrico;
9. Avaliar a possibilidade de executar análises de imagens diretamente em um microcontrolador.

## 1.2 JUSTIFICATIVA

A aplicação da Internet das Coisas (do inglês IoT) na agricultura tem sido reconhecida como um componente essencial para a integração de sensores e a coleta contínua de informações em tempo real, permitindo decisões mais precisas e rentáveis (Li et al. (2016); Singh et al. (2021)). Além de possibilitar o monitoramento climático, da fertilidade do solo e da demanda nutricional das culturas (MOHAMED et al., 2021), a IoT tem desempenhado um papel crescente na agricultura de precisão, com destaque para o controle de irrigação e a prevenção de perdas relacionadas ao manejo inadequado da água (ABDELMONEIM et al., 2025).

Entretanto, mesmo com os avanços em sistemas automatizados de irrigação, persiste a dificuldade em detectar precocemente o estresse hídrico, especialmente em ambientes controlados, como estufas. A maior parte das soluções atuais depende de sensores de solo, que, embora eficazes em medições pontuais, apresentam limitações de abrangência espacial e deterioração ao longo do tempo (MILLER et al., 2025). Nesse contexto, abordagens baseadas em visão computacional têm se mostrado promissoras, pois permitem capturar alterações fenotípicas sutis, como a variação da coloração foliar, que são indicadores indiretos, mas sensíveis, do estado hídrico das plantas (ATANASOV, 2021). Além disso, parte-se da **hipótese de que a visão computacional pode revelar nuances de cor e padrões espectrais imperceptíveis ao**

**olho humano**, ampliando a capacidade de diagnóstico precoce do estresse hídrico e de outras condições fisiológicas (AL-TAMIMI et al., 2022).

Diante desse cenário, esta dissertação justifica-se pela necessidade de superar a dependência de sensores intrusivos de solo, propondo um sistema de fenotipagem acessível e embarcado, instalado em uma tenda de cultivo com irrigação automatizada. A solução integra uma câmera RGB de baixo custo para captura de imagens, sensores de telemetria microambiental e algoritmos de aprendizado de máquina otimizados para microcontroladores. A principal contribuição está na demonstração de que é possível monitorar o estresse hídrico de forma não invasiva, em tempo real e com baixo custo computacional, promovendo a descentralização do processamento (Edge Computing) e ampliando as perspectivas de adoção prática em pequenas e médias propriedades agrícolas.

Assim, além de preencher uma lacuna identificada na literatura quanto à detecção precoce do estresse hídrico por meio de visão computacional, este trabalho contribui para a consolidação da agricultura digital ao oferecer uma solução prática, escalável e alinhada aos desafios globais de sustentabilidade e segurança alimentar previstos para 2030.

### 1.3 ORGANIZAÇÃO DO DOCUMENTO

Os próximos capítulos desta dissertação estão organizados da seguinte maneira: o Capítulo 2 apresenta a fundamentação teórica essencial e estudos que servem de alicerce para a compreensão do tema em questão, permitindo situar o estudo dentro do contexto acadêmico, evidenciando sua relevância e fundamentação. O Capítulo 3 apresenta trabalhos relacionados na literatura que empregam ideias e conceitos semelhantes aos utilizados neste trabalho. O Capítulo 4 descreve a metodologia adotada na presente pesquisa. O Capítulo 5 explica em detalhes o desenvolvimento dos experimentos implementados. O Capítulo 6 informa e discute os resultados obtidos. O Capítulo 7 conclui a dissertação e indica trabalhos futuros. O Capítulo 8, apresentado como apêndice, traz um aprofundamento adicional sobre a fundamentação teórica.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, introduzimos alguns dos conceitos cruciais para o desenvolvimento desta pesquisa, além de expor estudos e soluções contemporâneas focadas na agricultura de precisão. É importante salientar que os principais conceitos aplicados neste trabalho serão abordados neste capítulo. Enquanto, conceitos auxiliares podem ser consultados a partir do Anexo A. Dessa forma, a fundamentação primária e a revisão da literatura serão suficientes para conferir relevância ao presente estudo.

A Seção 2.2 introduz o uso do processamento de imagens a fim de adequar uma imagem para extração de dados ou ser entregue à próxima etapa, o Aprendizado de Máquina (do inglês ML). Então, a Seção 2.3 apresenta o ML como ferramenta para a classificação de imagens.

### 2.1 A VISÃO HUMANA FRENTE A VISÃO DE MÁQUINA

A visão é o mais desenvolvido dos sentidos. Por isso, não surpreende que as imagens desempenhem um papel central na percepção humana. Por outro lado, somos limitados à faixa visual do espectro eletromagnético, enquanto as máquinas podem cobrir quase todo o espectro eletromagnético, operando com imagens geradas por fontes não visíveis ao olho humano. Isso inclui ultrassom, ondas de rádio, microondas, do infravermelho até raios gama, e ainda imagens geradas por computador. Assim, o Processamento Digital de Imagens abrange um amplo e variado campo de aplicações. Porém, este trabalho está restrito ao processamento de imagens digitais geradas no âmbito da luz visível.

A partir de algumas observações sobre as limitações da visão humana, Jähne (1995) destaca que, embora o sistema visual humano seja extremamente eficaz no reconhecimento de objetos, ele apresenta deficiências em tarefas como a estimativa absoluta de valores de cinza, de distâncias e áreas. Além das barreiras inerentes à percepção humana, incluindo o fato de um dos integrantes deste estudo ser daltônico, Deutscher (2010) traz constatações controversas ao apontar que a percepção das cores pode variar entre diferentes culturas e até mesmo entre civilizações separadas por longos períodos históricos. Fatores como esses reforçam a ideia da inclusão da visão

de máquina como crivo em tarefas que envolvem análise de cor, promovendo maior objetividade, reprodutibilidade e independência de interpretações subjetivas. Jähne (1995) ainda afirma que avanços importantes na visão computacional foram alcançados graças ao progresso na compreensão do sistema visual humano.

A seção seguinte reúne e descreve os principais recursos de visão de máquina utilizados ao longo deste estudo.

## 2.2 O PROCESSAMENTO DIGITAL DE IMAGENS E A VISÃO COMPUTACIONAL

Szeliski (2022) situa o Processamento Digital de Imagens (PDI) como o estágio inicial da maioria dos algoritmos de Visão Computacional, definindo sua função como pré-processar a imagem e convertê-la para um formato adequado para obter resultados satisfatórios. Complementarmente, de Gonzalez e Woods (2008) depreende-se a Quadro 1, que apresenta uma breve distinção entre essas duas áreas relacionadas.

<b>Característica Principal</b>	<b>PDI</b>	<b>Visão Computacional</b>
<b>Objetivo Primário</b>	Manipular, aprimorar, restaurar e transformar imagens.	Fazer com que computadores interpretem o conteúdo de imagens e vídeos do mundo real.
<b>Entrada</b>	Imagem.	Imagem ou vídeo.
<b>Saída</b>	Imagem (melhorada, modificada, transformada).	Descrição, medição, reconhecimento, decisão, segmentação, reconstrução 3D.
<b>Foco</b>	Pixel-a-pixel, manipulação de baixo nível da imagem.	Extração de informações de alto nível, interpretação semântica do conteúdo da imagem.
<b>Exemplos de Tarefas</b>	Ajuste de brilho/contraste, filtros (desfoque, nitidez), remoção de ruído, equalização de histograma, compressão JPEG.	Reconhecimento facial, detecção de objetos, rastreamento de movimento, navegação de robôs, realidade aumentada, diagnósticos médicos por imagem.
<b>Pergunta Principal</b>	Como melhorar/modificar a imagem?	O que há na imagem? / O que a imagem representa?
<b>Entrada → Saída</b>	<b>Imagem → Imagem</b>	<b>Imagem → Descrição</b>

Quadro 1 – Distinção entre PDI e Visão Computacional. Fonte: Própria.

Concluimos que este trabalho envolve as duas áreas. No âmbito do PDI, a motivação é obter **imagem a partir da imagem**, enquanto no âmbito da Visão Compu-



tacional, focamos na extração de **informação a partir da imagem**. Nossa busca é quantificar o estresse hídrico da planta por meio de sua imagem, seja por **análise direta**, utilizando o índice RGB (do inglês Red, Green e Blue) da imagem da planta, quanto por **análise indireta**, via métodos estatísticos para reconhecimento de padrões e pela aplicação de técnicas de ML.

Este trabalho contempla as principais etapas de um *pipeline* típico de visão computacional, incluindo aquisição de imagens, filtragem, conversão de espaços de cor, segmentação, processamento morfológico, análise de padrões, compressão de imagens e, por fim, aprendizado de máquina.

A combinação de conversão de espaços de cor, filtros de suavização, segmentação e operações morfológicas constitui um *pipeline* robusto para extração de regiões de interesse — aqui, a planta — e serve de base para as subseções a seguir.

### 2.2.1 Detecção de Movimento Entre Imagens

A fim de se investigar outros fatores além da variação dos índices RGB com relação ao fluxo de água no interior da planta, este estudo propõe o uso de demais recursos no âmbito do PDI: a detecção de movimento (SINGLA, 2014) e o reconhecimento de padrões. A expectativa é de que o *SPheRe* seja capaz de coletar informações correlatas ao fluxo de água no interior da planta, inclusive de suas folhas, influenciando sua turgidez (rigidez que resulta da pressão interna das células vegetais, causada pela água em fluxo por osmose), conforme cita Guimarães e Stone (2008). Tais abordagens estão descritas a seguir.

Com foco na detecção de movimento entre frames, a metodologia proposta por Singla (2014) emprega a operação de subtração entre imagens. Essa técnica é eficaz para identificar o movimento resultante da recuperação do turgor nas células da planta, um efeito natural que ocorre durante a reidratação e que restaura sua postura original. Um exemplo da aplicação desse procedimento de subtração de imagens pode ser observado na Fig.3.

Além da detecção de movimento, as folhas das plantas também podem ser um forte indicativo do teor de água por meio de sua textura. A seguir, abordaremos este aspecto com maior profundidade.

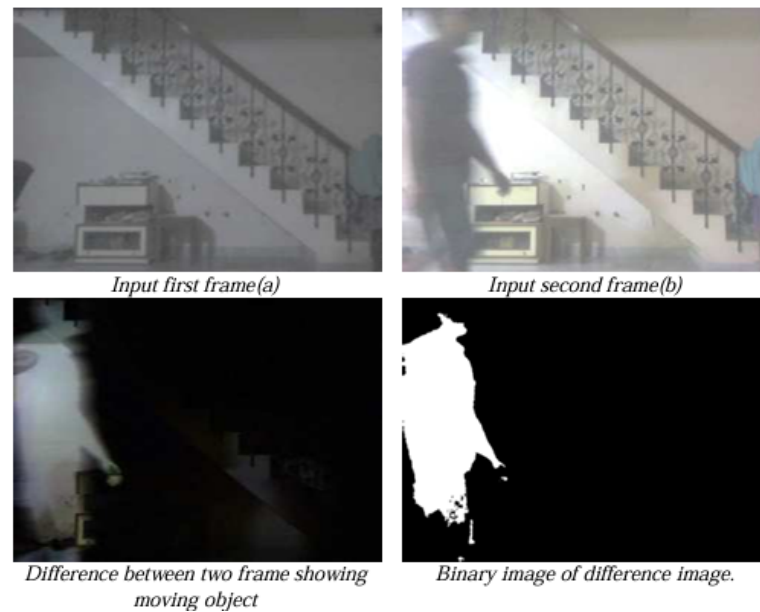


Figura 3 – Quando há movimento nas cenas, a imagem binária da diferença entre os dois quadros mostra o movimento na cor branca e, onde não há mudança, mostra a cor preta. Fonte: Singla (2014).

### 2.2.2 Análise de Textura

Após a segmentação de uma imagem em regiões, cada conjunto de pixels correspondente pode ser representado de duas maneiras principais para o processamento subsequente:

1. **Representação externa:** descreve a região por seu contorno, usando características como comprimento da borda, orientação da reta que une pontos extremos e número de concavidades.
2. **Representação interna:** baseia-se nos pixels que compõem a região, destacando propriedades como cor e textura.

Escolher o esquema de representação é apenas o primeiro passo. Em seguida, cada região deve ser descrita por descritores compatíveis com a abordagem escolhida. Seja externa ou interna (ou ambas), os descritores devem ser invariantes a variações de tamanho, translação e rotação.

No contexto deste presente estudo foi aplicada a representação interna baseada na textura das folhas, supondo que essa descrição regional seja relevante para indicar o nível de estresse hídrico da planta.

Dentre diversas definições para textura, Rosenfeld e Troy (1970) definem textura como um arranjo repetitivo de padrões sobre uma área, e propõem sua quantização por meio de fatores como dependência de níveis de cinza e autocorrelação. Para caracterizar a textura de uma região — medida intuitiva de atributos como suavidade, aspereza e regularidade — empregam-se três abordagens principais em processamento de imagens:

- **Estatística:** quantifica propriedades como granulação e uniformidade por meio de métricas derivadas de histogramas ou matrizes de coocorrência.
- **Estrutural ou Geométrica:** modela o arranjo de elementos básicos (primitivas), por exemplo, padrões de linhas paralelas e espaçadas regularmente.
- **Espectral:** analisa o espectro de Fourier para identificar periodicidades globais, detectando picos estreitos e de alta energia.

A abordagem adotada neste trabalho é de natureza estatística, fundamentando-se na caracterização da distribuição dos níveis de cinza de um conjunto de pixels por meio de medidas como média, variância e desvio padrão, conforme descrito por Pedrini e Schwartz (2008). Entre os métodos estatísticos disponíveis, empregou-se especificamente a Matriz de Co-ocorrência de Níveis de Cinza ou *Gray-Level Co-Occurrence Matrix* (GLCM). A GLCM conta quantas vezes um certo nível de cinza (em um pixel) aparece ao lado de outro certo nível de cinza (em pixel vizinho), em uma direção e distância específicas.

A análise de textura via GLCM baseia-se em extrair estatísticas (Haralick features) que quantificam propriedades como contraste, homogeneidade e correlação. A construção da GLCM será descrita na próxima subseção, e sua aplicação permite capturar texturas direcionais e, somando matrizes rotacionadas, obter invariância a rotações. A normalização da GLCM culmina numa distribuição de probabilidades, e a partir disto, os descritores são calculados, formando um vetor característico de textura. Essa abordagem é amplamente utilizada em aplicações que vão de classificação de imagens médicas a análise remota de padrões de superfície.

### 2.2.2.1 Construção da GLCM

Dada uma imagem  $I$  com  $L$  níveis de cinza, segue o procedimento para construir sua Matriz de Co-ocorrência de Níveis de Cinza (GLCM):

1. **Escolha da Direção e Distância (offset):** Definição do deslocamento  $(\Delta x, \Delta y)$  (ou ângulo  $\theta$  e distância  $d$ ) que especifique a relação espacial entre pares de pixels. Em outras palavras, primeiramente, se decide quem são os pixels vizinhos. Por exemplo: a partir do pixel de referência, o seu vizinho é o pixel imediatamente à direita (distância 1, ângulo 0 graus) ou o pixel na diagonal superior direita (distância 1, ângulo 45 graus)
2. **Inicialização com uma Tabela Vazia:** Criada uma matriz  $P \in \mathbb{N}^{L \times L}$  com todos os elementos zerados. Ou seja, uma matriz quadrada  $L \times L$  nula. Se a imagem tem níveis de cinza de 0 a 255 (de preto a branco), a tabela terá 256 linhas e 256 colunas. Cada célula dessa tabela começa com o valor zero.
3. **Contagem de pares.** Para cada pixel  $(x, y)$  na imagem  $I(x, y)$ :
  - Para cada pixel (**pixel atual**), se verifica o seu **pixel vizinho** na direção e distância escolhidas no passo 1.
  - Computar o nível de cinza do **pixel atual** (esse será o número da linha da tabela).
  - Computar o nível de cinza do **pixel vizinho** (esse será o número da coluna da tabela).
  - Na célula correspondente da tabela é **adicionado 1** a ela.
4. **Normalização:** A tabela  $P$  é convertida em distribuição de probabilidades:

$$p(i, j) = \frac{P(i, j)}{\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} P(i, j)}$$

de modo que

$$\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} p(i, j) = 1.$$

Essa matriz GLCM mostra padrões de textura. Se muitos números altos estiverem perto da diagonal principal da tabela, significa que os pixels vizinhos têm níveis de

cinza muito parecidos (a imagem é homogênea, com poucas mudanças, como uma área lisa). Se os números estiverem espalhados por toda a tabela, significa que há muitas mudanças nos níveis de cinza entre pixels vizinhos (a imagem tem mais textura ou é mais “áspera”). Uma ferramenta poderosa para descrever a textura de uma imagem de forma numérica.

Na Fig.4, um exemplo de obtenção da matriz GLCM normalizada a partir de uma imagem hipotética com base em 3 níveis de cinza:

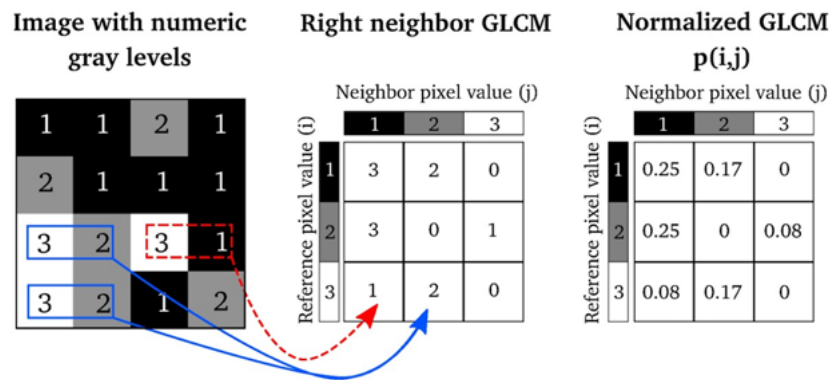


Figura 4 – Operações sobre uma imagem (esq.) para obtenção da matriz GLCM normalizada (dir.) com base no pixel vizinho à direita. Fonte: Tran et al. (2020).

A partir de  $p(i, j)$ , calculam-se estatísticas de textura (HARALICK et al., 1973), por exemplo:

- **Contraste:**  $\sum_{i,j} (i - j)^2 p(i, j)$ .
- **Dissimilaridade:**  $\sum_{i,j} |i - j| p(i, j)$ .
- **Homogeneidade (IDM):**  $\sum_{i,j} \frac{p(i,j)}{1 + (i-j)^2}$ .
- **Energia (ASM):**  $\sum_{i,j} p(i, j)^2$ .
- **Entropia:**  $-\sum_{i,j} p(i, j) \log p(i, j)$ .
- **Correlação:**  $\frac{\sum_{i,j} (i - \mu_i)(j - \mu_j) p(i, j)}{\sigma_i \sigma_j}$ .

Considerar múltiplos offsets e ângulos (por exemplo  $\theta = 0^\circ, 45^\circ, 90^\circ, 135^\circ$ ) resulta em GLCMs mais robustas, invariantes a rotações.

A seguir, a Fig. 5 retrata o cálculo de duas das características de *Haralick* sobre oito regiões da imagem original, quatro no céu, uma região bastante homogênea, e quatro regiões situadas na grama, notoriamente variando sua textura na direção vertical.

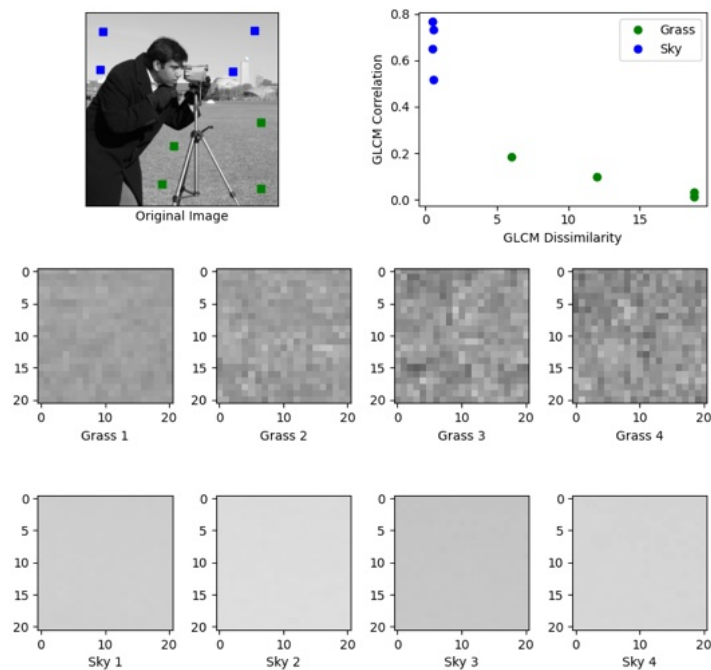


Figura 5 – Correlação e dissimilaridade a partir da GLCM de 256 níveis de cinza de 4 regiões no céu e 4 na grama. Fonte: Gomede (2024)

#### 2.2.2.2 Aplicações

A GLCM e suas feições de Haralick são amplamente usadas em análise de imagens médicas, sensoriamento remoto, classificação de terrenos e detecção de defeitos de fabricação, graças à sua capacidade de capturar padrões de textura de forma robusta ao ruído (NANNI et al., 2013).

#### 2.2.2.3 Técnicas modernas de classificação de texturas

Além das características de Haralick, tradicionalmente empregadas na análise de texturas, técnicas modernas têm sido exploradas para ampliar a capacidade de discriminação em aplicações de visão computacional. Entre elas, destacam-se os Local Binary Patterns (LBP), amplamente utilizados por sua simplicidade e robustez a variações de iluminação, e os filtros de Gabor, capazes de capturar informações multi-escalares e multiorientadas da textura. Mais recentemente, abordagens baseadas em aprendizado profundo, como redes neurais convolucionais (CNNs), têm demonstrado elevado desempenho na extração automática de descritores texturais, dispensando a necessidade de engenharia manual de atributos. Estudos de revisão e aplicações

práticas (Liu et al. (2019);Cimpoi et al. (2015)) indicam que a combinação entre descritores clássicos e métodos modernos pode potencializar análises em fenotipagem de plantas, configurando-se como um caminho promissor para futuras investigações.

A fim de complementar os conceitos até aqui discutidos, a seção seguinte expõe a última etapa do *pipeline* deste desenvolvimento, a IA para classificação de imagens.

## 2.3 APRENDIZAGEM DE MÁQUINA E CLASSIFICAÇÃO DE IMAGENS

De acordo com Sze et al. (2017), Inteligência Artificial (IA) é a ciência e a engenharia que desenvolvem máquinas capazes de atingir objetivos com autonomia e desempenho semelhantes aos humanos. No âmbito da IA, o ML estuda métodos que permitem aos computadores aprender a partir de dados, sem a necessidade de programação explícita. Theodoridis (2015) compara esse processo ao modo como o cérebro humano aprende e faz previsões.

Atualmente, o uso do Machine Learning está cada vez mais presente em nosso cotidiano, com aplicações que vão desde sistemas de recomendação até a análise de grandes volumes de informações. Um exemplo fundamental é o reconhecimento de objetos — um conjunto de tarefas da Visão Computacional voltadas à identificação automática de itens em imagens. A Fig. 6 ilustra algumas dessas tarefas de detecção e classificação de objetos.

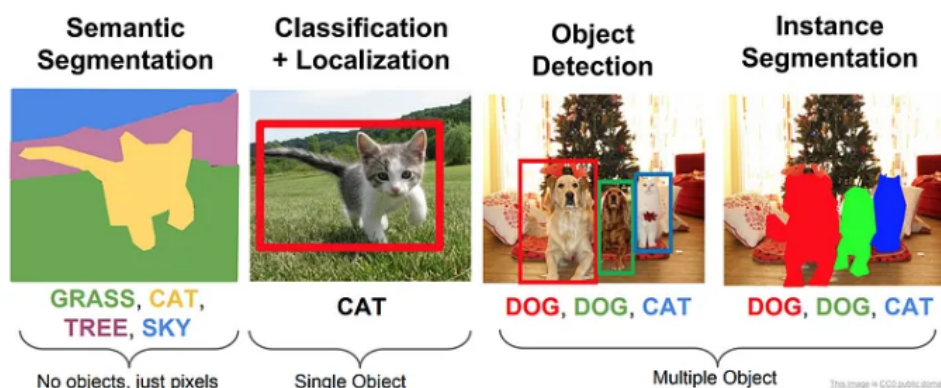


Figura 6 – Tarefas relacionadas ao reconhecimento de objetos. Fonte: Agarwal (2015).

Segundo Agarwal (2015), no campo da Visão Computacional é possível distinguir três tarefas principais:

1. **Classificação de Imagens:** Consiste em atribuir uma categoria semântica a

uma imagem que contém, em geral, um único objeto. O objetivo é prever a qual classe o objeto pertence.

2. **Deteção de Objetos:** Trata-se de uma generalização do problema de classificação. Envolve tanto a identificação quanto a localização de múltiplos objetos em uma imagem, utilizando caixas delimitadoras (*bounding boxes*) para circundar cada objeto detectado e atribuir-lhe uma classe.
3. **Segmentação de Objetos:** Busca atribuir a cada pixel da imagem um rótulo de categoria, permitindo distinguir precisamente os contornos dos objetos. A segmentação compreende tanto o particionamento quanto a associação semântica dos segmentos a classes conhecidas.

De modo geral, como destacado por Theodoridis (2015), o objetivo da classificação é associar um padrão desconhecido a uma dentre várias classes previamente estabelecidas. Abraham et al. (2025) definem a classificação de imagens como o processo de atribuir um rótulo a uma imagem de entrada com base em seu conteúdo.

Com o avanço das pesquisas, a classificação de imagens passou a ser amplamente aplicada em diversos domínios, como educação, segurança, saúde, comércio e agricultura. Alguns exemplos de aplicação incluem: reconhecimento de caligrafia, detecção facial, análise de cenas, visão computacional para veículos autônomos, reconhecimento de gestos manuais e identificação de doenças (RAWAT; WANG, 2017).

Pardede et al. (2020) afirmam que os métodos de classificação podem ser organizados segundo sua profundidade em duas categorias principais: arquiteturas rasas e arquiteturas profundas. As arquiteturas rasas abrangem técnicas tradicionais de ML, como:

- Máquinas de Vetores de Suporte (SVM) (BOSER et al., 1992);
- Classificador Bayes Naïve (FRIEDMAN et al., 1997);
- K-Vizinhos Mais Próximos (k-NN) (FIX, 1985).

Esses métodos são ilustrados por Soofi e Awan (2017) na Fig. 7. Já as arquiteturas profundas baseiam-se principalmente em redes neurais convolucionais (do inglês CNNs), que serão abordadas nas seções seguintes.



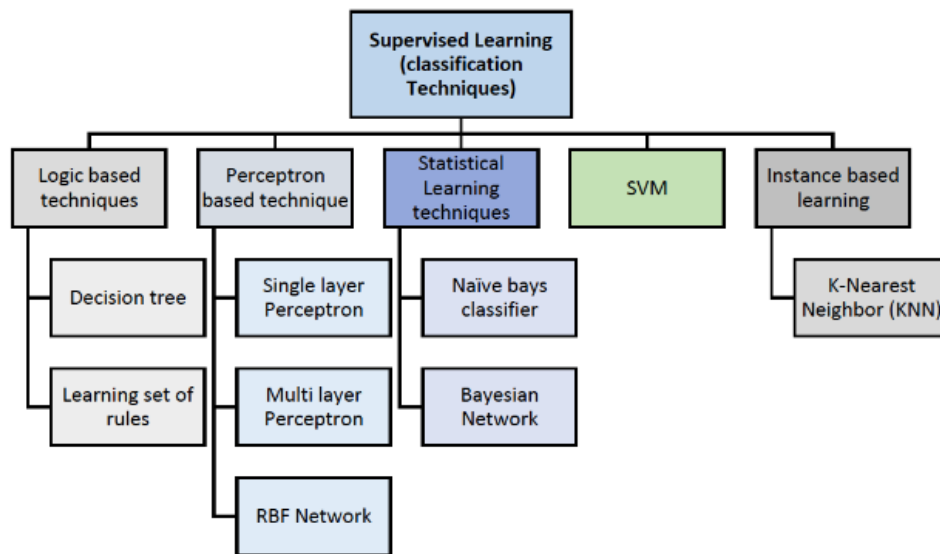


Figura 7 – Técnicas de classificação de aprendizagem supervisionada. Fonte: Soofi e Awan (2017).

### 2.3.1 Redes Neurais e Aprendizagem Profunda

Inicialmente apresentamos as definições de IA e ML. Sze et al. (2017) aprofundam essa discussão, identificando um subcampo inspirado no funcionamento do cérebro: em especial, as redes neurais artificiais (RNAs). A Fig. 8 ilustra a posição das RNAs e do Aprendizado Profundo (do inglês DL) dentro da IA.

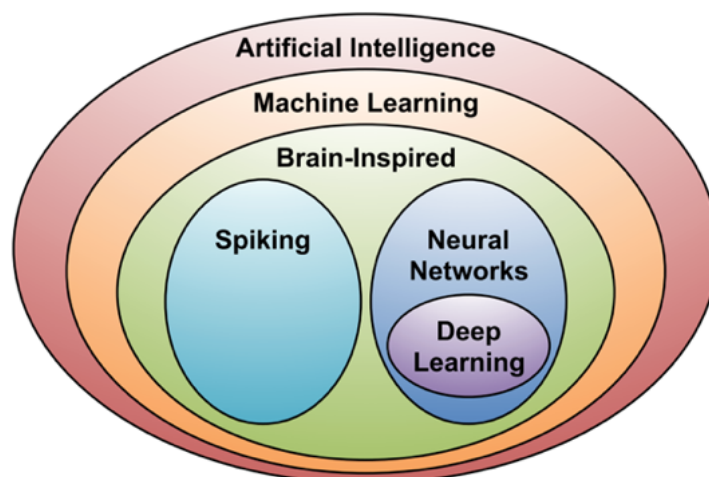


Figura 8 – Relação entre Deep Learning e Inteligência Artificial. Fonte: Sze et al. (2017).

Como o próprio nome indica, as redes neurais artificiais foram inspiradas no funcionamento do sistema nervoso biológico. Nos sistemas naturais, os neurônios conectam-se por meio de ligações funcionais chamadas sinapses, que podem ser ativadas ou inibidas. Essas conexões são responsáveis pela mediação das informações entre os

neurônios, organizados de maneira hierárquica, conforme descrito por Theodoridis (2015).

Foi o trabalho de McCulloch e Pitts (1943) que primeiro desenvolveu um modelo computacional para um neurônio. Posteriormente, Rosenblatt (1958) construiu uma máquina de aprendizado baseada no modelo de neurônio que aprende a partir de um conjunto de dados de treinamento. Essa máquina é chamada de perceptron e é o ponto de partida para redes neurais artificiais. De acordo com Theodoridis (2015), redes neurais são máquinas de aprendizado compostas por um grande número de neurônios conectados em camadas. A Fig. 9 exemplifica um sistema de rede neural.

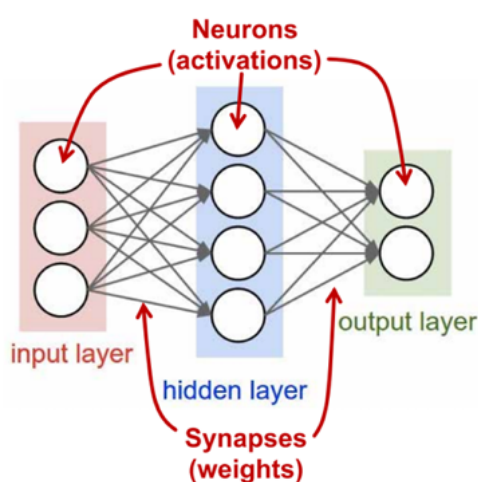


Figura 9 – Sistema de Rede Neural. Fonte: Sze et al. (2017).

Um Perceptron Multicamadas (do inglês MLP) é uma rede neural simples composta por mais de uma camada de perceptrons. Os neurônios na camada de entrada recebem os valores de entrada e os propagam para a camada intermediária, ou “camada oculta” da rede. Cada valor é associado a um “peso”, e a computação de cada neurônio envolve uma soma ponderada dos valores de entrada. A rede propaga esses valores ponderados pelas camadas ocultas até a camada de saída, e a camada de saída apresentará as saídas da rede ao usuário (SZE et al., 2017).

Por muitos anos, modelos de redes neurais rasas com poucos estágios foram a principal escolha para computação inspirada no cérebro. No entanto, no início da década de 1990, redes neurais profundas (compostas por muitas camadas) tornaram-se um objeto de pesquisa amplamente explorado (SCHMIDHUBER, 2015). Vários estudos definem DL como a área dentro das redes neurais onde há mais de três camadas (mais de uma camada oculta). O número de camadas em uma Rede Neural Profunda (do inglês DNN) é geralmente entre 5 e 1000, de acordo com Sze et al. (2017). O

processo de aprendizagem em uma DNN, também chamado de treinamento, envolve a determinação do valor dos pesos na rede. A inferência executa o modelo de rede com fatores de peso fornecidos.

Em uma rede neural multicamadas *feed-forward*, cada nó em uma camada se conecta a todos os outros nós na camada vizinha (SVOZIL et al., 1997). Os valores de entrada se propagam pelas camadas e os neurônios se conectam entre si por meio de pesos. Cada neurônio recebe informações dos neurônios precedentes e produz uma saída passando a soma ponderada desses sinais por meio de uma função de ativação (SAZLI, 2006). O nome rede feed-forward indica que a informação flui da camada de entrada para a camada de saída (THEODORIDIS, 2015).

O processo de otimização usado no treinamento de redes é chamado de “gradiente descendente”. Nesse processo, a derivada parcial da perda relacionada a cada peso determina o valor do peso atualizado, conforme explicado por Theodoridis (2015). O processo é repetido a cada iteração para reduzir a perda geral. O algoritmo usa retropropagação para calcular as derivadas parciais dos gradientes, ou seja, os valores de peso são passados de volta pela rede para calcular a função de perda dependente do peso. A diferença entre a saída de rede real e a desejada define a perda. Portanto, esta última deve estar disponível para o treinamento da rede, implicando em uma técnica de aprendizado supervisionado (SAZLI, 2006). Uma série de métodos como, por exemplo, lote, aprendizado supervisionado, aprendizado por reforço e ajuste fino, melhoram o desempenho e a eficiência do treinamento.

De acordo com Svozil et al. (1997), uma rede neural multicamadas feed-forward pode operar em dois modos: treinamento e predição. O processo de treinamento ajusta os fatores de peso a cada iteração para reduzir o erro (perda), partindo de valores arbitrários iniciais. Cada iteração é chamada de época e, geralmente, várias épocas são necessárias para concluir o treinamento. Os pesos convergirão para um conjunto de valores considerado o ótimo local à medida que o processo iterativo continua.

Durante a predição, o modelo recebe uma imagem como entrada e gera um vetor de pontuações, uma para cada classe. O conteúdo do vetor indica a probabilidade do objeto pertencer a essa classe. Normalmente, a pontuação mais alta representa a classe mais provável (SZE et al., 2017). Além disso, podemos descrever a perda como a lacuna entre as pontuações ideais corretas e as pontuações computadas pela

DNN com base em seus pesos atuais. Portanto, o objetivo do treinamento da DNN é determinar os fatores de peso que maximizam a pontuação correta da classe ou que minimizam a perda média em um conjunto de treinamento extenso. O erro resultante é uma estimativa da qualidade da predição da rede treinada (SVOZIL et al., 1997).

De acordo com Voulodimos et al. (2018), na última década, houve uma série de desenvolvimentos em arquiteturas profundas e algoritmos de DL. Entre os fatores que contribuíram para essas melhorias está o surgimento de grandes conjuntos de dados disponíveis publicamente, potencializados pela computação de Unidade de Processamento Gráfico (do inglês GPU). Além disso, novas técnicas de regularização e estruturas poderosas aceleraram a revolução do DL.

Atualmente, algumas das aplicações que utilizam DL incluem assistência médica, processamento de dados visuais, análise de redes sociais e processamento de áudio e fala (HASAN et al., 2020). Além disso, as técnicas de DL alcançam bom desempenho em vários problemas de visão computacional, como detecção de objetos, rastreamento de movimento, reconhecimento de ações, estimativa de pose humana e segmentação semântica, como afirmam Voulodimos et al. (2018). Um dos tipos mais relevantes de modelos de DL para aplicações de visão computacional e imagem são as CNNs, às quais nos voltamos na seção seguinte.

### **2.3.2 Redes Neurais Convolucionais**

As aplicações em classificação de imagens e detecção de objetos aumentaram com o desenvolvimento e o aprimoramento de algoritmos de DL. De acordo com Pathak et al. (2018), métodos de detecção de objetos usando técnicas de DL baseadas em CNN têm sido extensivamente aplicados. Esse tipo de rede neural tem bom desempenho ao processar dados que vêm na forma de múltiplas matrizes, como muitas modalidades de dados de mídia são, conforme descrito por LeCun et al. (2015).

Na década de 1980, estudos em Neurociência concluíram que o cérebro tem diferentes regiões para tarefas distintas. Portanto, o cérebro tem uma organização hierárquica e localizada. Fukushima (1988) publicou o primeiro modelo computacional baseado no cérebro humano e suas conectividades locais. No entanto, o termo CNN entrou em uso apenas na década de 1990, com a pesquisa de LeCun et al. (2015), que envolveu o uso de uma rede neural para reconhecer caracteres em imagens.

Uma CNN é um algoritmo de aprendizado profundo que recebe uma imagem como entrada, atribui pesos aprendíveis a vários objetos nela e pode diferenciar um do outro. Em uma CNN, apenas a última camada é totalmente conectada, enquanto em uma Rede Neural Artificial (RNA), cada neurônio é conectado a todos os outros neurônios (WAGH et al., 2019). As CNNs são compostas por múltiplas camadas convolucionais. A rede gera um mapa de características em cada camada, consistindo em uma abstração de nível cada vez mais alto dos dados de entrada, que preserva informações essenciais, porém únicas.

Por definição, uma CNN usa uma única rede para aprender diversas características de uma determinada imagem e realizar sua classificação, tarefas anteriormente realizadas separadamente. A estrutura do sistema visual inspirou essa ideia. Hoje, as CNNs alcançam resultados muito bons em reconhecimento de padrões, conforme declarado em Voulodimos et al. (2018).

Os principais componentes de uma CNN são a camada convolucional, a função de ativação, a camada de subamostragem e a camada totalmente conectada. Uma arquitetura típica de CNN consiste em repetições de sequências de diversas camadas convolucionais e uma camada de agrupamento, seguidas por uma ou mais camadas totalmente conectadas (YAMASHITA et al., 2018). Descrevemos esses componentes nas seções a seguir.

#### 2.3.2.1 Camada Convolucional

Em uma CNN, cada camada convolucional é responsável por realizar a extração de características e gerar uma abstração sucessivamente de nível superior dos dados de entrada, chamada de “mapa de características”. Essa abstração preserva informações essenciais, porém únicas, sobre a imagem de entrada (SZE et al., 2017). A extração de características consiste em uma combinação de operações lineares e não lineares, como, por exemplo, funções de convolução e ativação (YAMASHITA et al., 2018).

A camada convolucional em uma CNN, ao contrário das redes MLP, preserva a estrutura espacial da imagem de entrada. De acordo com Yamashita et al. (2018), ela recebe a entrada como uma matriz tridimensional e, em seguida, convolui um filtro tridimensional (também chamado de kernel) com a imagem (chamado de tensor). O filtro desliza sobre a imagem espacialmente, computando produtos por elemento. A

operação realizada pelo filtro em cada posição é uma convolução matricial entre a imagem de entrada e o filtro e, como saída, produz um mapa de ativação composto pelas características da imagem. A Fig. 10 mostra um exemplo ilustrativo de uma operação de convolução.

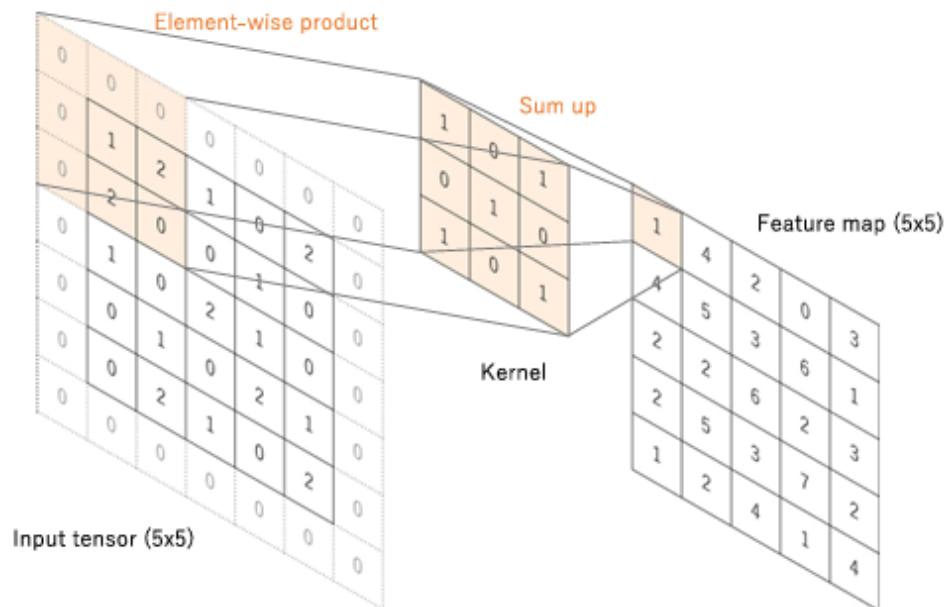


Figura 10 – Uma operação de convolução. Fonte: Yamashita et al. (2018).

Cada camada convolucional de uma CNN inclui vários filtros, e cada filtro gerará um mapa de características diferente, aprendendo um atributo específico da imagem (YAMASHITA et al., 2018). A profundidade do filtro deve ser equivalente à profundidade de entrada corrente. Como as CNNs são compostas por várias camadas convolucionais, e cada camada inclui vários filtros, muitos mapas de características se combinam para classificar uma imagem (RAWAT; WANG, 2017).

Devido ao seu comportamento, as características aprendidas pelo filtro são robustas à translação, como afirmam LeCun et al. (2015). Em outras palavras, o filtro, que atua como um detector de padrões, pode identificar esses padrões em qualquer local da imagem. Um filtro aprendido se aplica a qualquer local da rede, uma vez que os parâmetros são compartilhados.

Os parâmetros básicos de cada camada convolucional são o tamanho do filtro, o número de filtros, o preenchimento e o passo (YAMASHITA et al., 2018). O preenchimento está relacionado ao tamanho das bordas da imagem, e o passo é o deslocamento que o filtro realizará cada vez que deslizar pela imagem. A convolução na Fig. 6, por exemplo, tem preenchimento zero e passo igual a 1. Outro parâmetro possível é a taxa

de expansão. Uma taxa de expansão superior a um implica um kernel dilatado, o que faz com que o filtro perca sua característica de localização em algum grau.

### 2.3.2.2 Função de Ativação

Após a camada de convolução, uma função de ativação é adicionada à rede para modular não linearidades, de acordo com Yamashita et al. (2018). Dependendo da função, ela varrerá a rede e permitirá que alguns valores sejam replicados. Em outras palavras, os pixels que não são necessários serão desativados e apenas os pixels essenciais serão mantidos (WAGH et al., 2019). Entre as funções de ativação mais utilizadas, estão: passo binário, sigmoide, tangente hiperbólica (TanH), unidades lineares retificadas (ReLU) e Softmax. Cada uma gerará um mapa de ativação de saída diferente. Na Fig.11 a representação de algumas dessas funções de ativação.

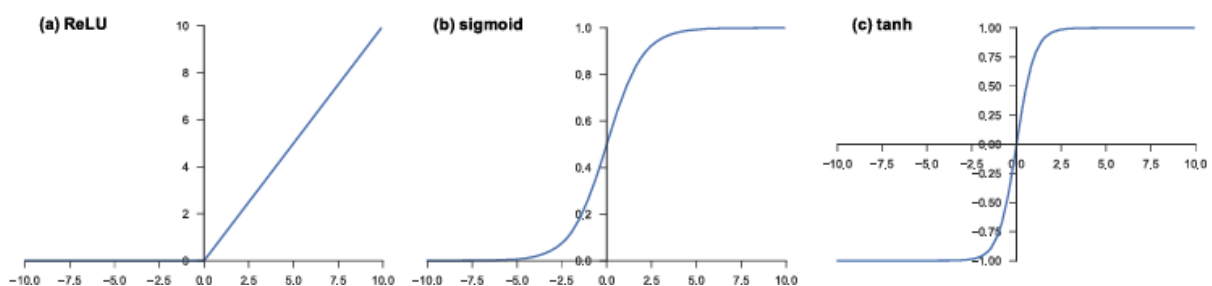


Figura 11 – Funções de ativação comumente aplicadas a redes neurais: (a) unidade linear retificada (ReLU), (b) sigmóide e (c) tangente hiperbólica (tanh). Fonte: Yamashita et al. (2018).

### 2.3.2.3 Camada de Subamostragem

LeCun et al. (2015) afirma que a camada de subamostragem, também chamada de camada de *pooling* ou *downsampling*, é responsável por gerar representações menores a partir dos mapas de características produzidos nas camadas anteriores, para criar uma representação com menor custo computacional e trabalhar contra o *overfitting*. De acordo com Voulodimos et al. (2018), a camada de *pooling* reduz as dimensões espaciais do volume de entrada para a próxima camada convolucional. Além disso, também diminui o número de parâmetros aprendíveis (YAMASHITA et al., 2018). Mesmo que não afete a dimensão de profundidade do volume, leva a uma certa perda de informação (VOULODIMOS et al., 2018). No entanto, outra característica desta ca-

mada é que ela reduz a sensibilidade a pequenas distorções na imagem. As principais estratégias utilizadas para subamostragem são:

- *Max Pooling*: Replica o valor máximo de um grupo;
- *Average Pooling*: Gera o valor médio de um grupo.

A Fig. 12 mostra um exemplo de uma operação de agrupamento máximo com um tamanho de filtro de  $2 \times 2$ , sem preenchimento de borda e um passo de 2.

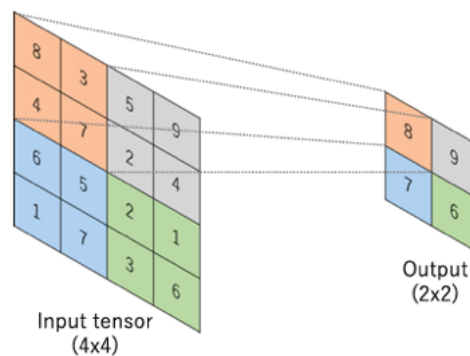


Figura 12 – Uma operação de Max pooling. Fonte: Yamashita et al. (2018).

#### 2.3.2.4 Camada Densa

A camada de saída de uma CNN é uma camada básica de rede neural totalmente conectada, onde todos os neurônios estão totalmente conectados a todos os neurônios da camada anterior (WAGH et al., 2019). Conforme declarado por Voulodimos et al. (2018), esse tipo de camada realiza o raciocínio de alto nível na rede neural. A camada de saída converte mapas de características bidimensionais em vetores unidimensionais. Esses vetores de características constituem o resultado direto da classificação ou permitem processamento posterior. A ideia principal dessa camada é a mesma usada em um MLP. Ela compreende um classificador e uma unidade de computação para calcular a função de perda, atuando como uma camada de saída (CUI, 2018). Em uma visão geral, uma CNN combina todos esses componentes para otimizar o processamento e a classificação de imagens (LECUN et al., 2015). Cada camada pode aparecer várias vezes e pode ser combinada de muitas maneiras diferentes, dependendo da aplicação. Portanto, existe uma arquitetura específica para cada tarefa, permitindo a extração e a classificação de características. O treinamento da CNN determina os



pesos das camadas totalmente conectadas, os vieses das funções de ativação e os filtros das camadas convolucionais.

### 2.3.3 Treinamento de uma CNN

Conforme mencionado em 2.3.2, o treinamento de uma CNN é o processo que determina alguns valores e parâmetros envolvidos na rede neural, a saber, os filtros das camadas convolucionais, os vieses (*bias*) da função de ativação e os pesos das camadas totalmente conectadas. O aprendizado desses valores ocorre por meio de um processo que envolve múltiplas repetições de toda a sequência CNN e o cálculo da perda após cada uma delas. A perda é propagada da última para a camada inicial (razão pela qual é chamada de retropropagação), e os pesos e parâmetros são ajustados dependendo do valor da perda a fim de minimizá-la (YAMASHITA et al., 2018).

Atualmente, existem vários algoritmos para otimizar esse treinamento. Um dos mais conhecidos e utilizados é o *Mini-Batch Gradient Descent*, descrito em Li et al. (2014). Os valores iniciais dos parâmetros a serem aprendidos são determinados aleatoriamente. Primeiro, ele seleciona N amostras de dados (neste caso, N imagens), onde N corresponde ao tamanho do lote associado. Em seguida, o minilote é propagado pela CNN para calcular a perda de treinamento. Os gradientes são calculados na próxima etapa por meio de retropropagação pela rede. Finalmente, os parâmetros são atualizados usando os gradientes. O processo se repete para todos os minilotes e, em seguida, todo o processo é repetido por um determinado número de vezes (chamados de épocas), ou até que a perda atinja um valor mínimo, indicando que a CNN atingiu um bom desempenho.

Diferentes algoritmos de otimização possíveis podem substituir o Gradiente Descendente Estocástico (do inglês SGD) frequentemente usado. Alguns exemplos são: SGD usando também o Momentum (outro hiperparâmetro da CNN), RMSProp ou Adam. A escolha do algoritmo depende da aplicação.

A Fig. 13 representa o fluxo completo de treinamento de uma CNN básica, conforme descrito em Yamashita et al. (2018). A imagem de entrada passa inicialmente por vários blocos, cada um composto por uma sequência de camadas convolucionais e funções de ativação (neste caso, representadas pela função ReLU), seguidos por uma camada de agrupamento máximo. Em seguida, os mapas de características ge-

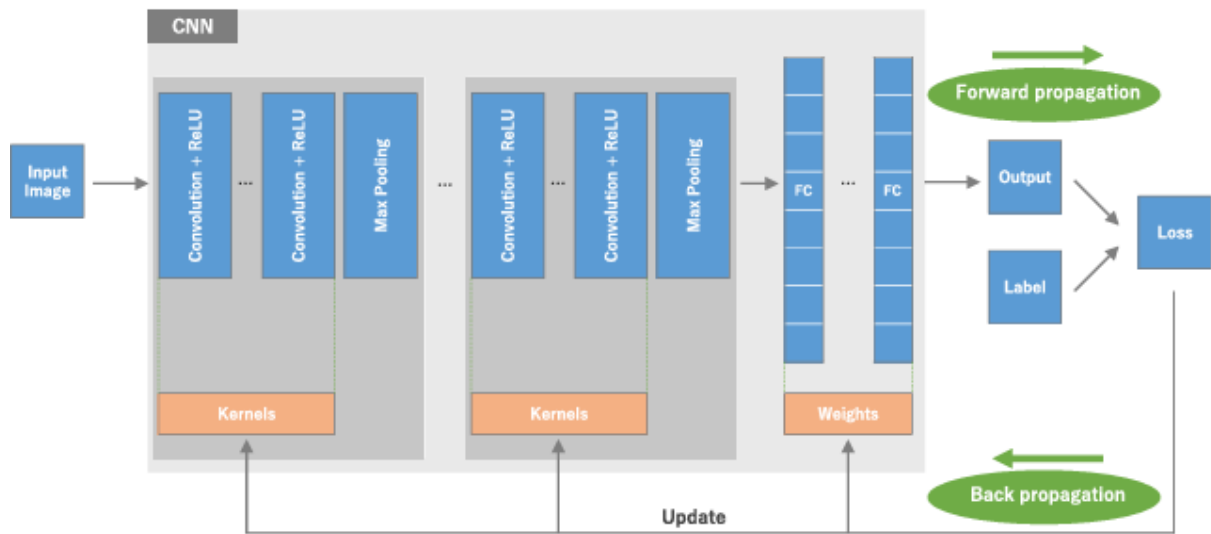


Figura 13 – Treinamento de uma CNN. Fonte: Yamashita et al. (2018).

dados passam por uma série de camadas totalmente conectadas, gerando a saída da rede, ou seja, a previsão. Essa previsão é, por sua vez, comparada com o valor verdadeiro para esse dado de entrada, e a perda calculada é retropropagada pela rede para atualizar os pesos e os *kernels* de convolução. A ordem e o número de camadas variam, dependendo da arquitetura da CNN escolhida.

O primeiro passo no processo de treinamento é o pré-processamento dos dados. Existem várias maneiras de pré-processar os dados de entrada antes do processo de treinamento. Uma estratégia para oferecer maior flexibilidade ao modelo e torná-lo menos sensível a mudanças nos pesos é normalizá-lo, conforme expresso por Patro e Sahu (2015). Essa etapa garante que cada parâmetro de entrada tenha uma distribuição semelhante, o que permite que o algoritmo trabalhe mais na região central dos dados do espaço N-dimensional. Além disso, os dados centralizados simplificam a representação dessas informações e minimizam o impacto na borda de separação.

A próxima etapa, de posse do conjunto de dados pré-processado, o divide em sub-conjuntos, cada um com um papel diferente no processo de treinamento da CNN. De acordo com Yamashita et al. (2018), normalmente, os dados disponíveis são divididos em três conjuntos:

- O conjunto de treinamento é necessário para treinar a rede. A propagação direta calcula a perda e a propagação reversa atualiza os parâmetros aprendíveis;
- O conjunto de validação é necessário para avaliar o modelo durante o processo de treinamento e ajustar os hiperparâmetros;

- O conjunto de teste permite avaliar o desempenho do modelo final após o treinamento.

A inicialização dos pesos (inclusive *bias*) associados a cada neurônio na rede, conforme descrito em 2.3.2, segue a preparação dos dados para treinamento. Conforme descrito por Narkhede et al. (2022), existem duas estratégias possíveis para esse processo: realizar a inicialização com novos pesos ou, por meio da técnica de aprendizado por transferência, usar os valores de peso de um modelo pré-treinado. O primeiro tipo de inicialização pode gerar valores de peso aleatórios, realizar uma inicialização orientada a dados ou uma híbrida, que combina os dois métodos. A inicialização usando pesos pré-treinados melhora a generalização ao aprender características de qualidade dos dados (ERHAN et al., 2010). A ideia é que, a partir de um modelo treinado em um banco de dados mais genérico, por exemplo, ImageNet (DENG et al., 2009), possamos usar seus pesos de diferentes maneiras, dependendo do tamanho do conjunto de dados. As possibilidades incluem treinar (i) o modelo inteiro (camadas convolucionais e totalmente conectadas), (ii) algumas camadas convolucionais (deixando outras congeladas) ou (iii) as camadas totalmente conectadas congelando a base convolucional.

Em seguida, o modelo de rede neural usa o algoritmo de retropropagação, conforme já mencionado. De acordo com Yamashita et al. (2018), o principal objetivo neste ponto é encontrar *kernels* nas camadas de convolução e pesos nas camadas totalmente conectadas, que minimizem as diferenças entre as previsões de saída e os rótulos reais fornecidos em um conjunto de dados de treinamento. Valores específicos dos *kernels*, pesos e *bias* determinam o desempenho do modelo. A propagação direta gera uma perda em um conjunto de dados de treinamento, e um algoritmo de otimização atualiza os parâmetros aprendíveis de acordo com o valor da perda. Para classificações binárias, uma das funções de perda mais utilizadas é a perda de entropia cruzada binária (RUBY et al., 2020), uma classe particular de perda de entropia cruzada onde os dois alvos de predição são 0 e 1. A função de perda mais comum em problemas de classificação multiclasse é a perda de entropia cruzada categórica (descrita em Koidl (2013)), que mede a dissimilaridade entre a distribuição de rótulos verdadeira e a prevista.

Conforme expresso por Rawat e Wang (2017), um problema comum ao realizar a

classificação de imagens é o *overfitting* (sobreajuste), o que significa que o modelo atinge um bom desempenho no conjunto de treinamento, enquanto, um mau desempenho diante de um conjunto de teste. O *overfitting* implica que o modelo não aprendeu a capacidade de generalizar em dados não vistos. Uma técnica para reconhecer o *overfitting* é monitorar a perda e uma métrica de avaliação nos conjuntos de treinamento e validação (YAMASHITA et al., 2018). Esse roteiro verificará se o modelo tem um desempenho muito bom no conjunto de treinamento em comparação com o conjunto de validação, indicando que ocorreu *overfitting*. Algumas estratégias ajudam a mitigar esse problema. Uma das mais comuns é realizar o aumento de dados, conforme afirmado por Krizhevsky et al. (2012). Essa abordagem amplia artificialmente o conjunto de dados usando transformações que preservam rótulos. Essas transformações incluem translações de imagens e reflexões horizontais ou alterações de intensidade dos canais RGB nas imagens de treinamento. Uma outra estratégia para prevenir *overfitting* é o *dropout*, proposta por Srivastava et al. (2014), uma técnica que em cada uma das épocas (e em cada lote de treinamento) seleciona aleatoriamente um percentual de neurônios e os desativa, para que não participem da propagação para frente ou para trás durante o processo de treinamento de uma rede neural. Isso ajuda a garantir que a rede seja robusta e redundante, genérica o suficiente para que não dependa de nenhuma área específica para obter respostas.

#### 2.3.4 CNNs e Classificação de Imagens

Desde o início dos anos 2000, pesquisadores têm aplicado CNNs com sucesso para a detecção, segmentação e reconhecimento de objetos e regiões em imagens (LECUN et al., 2015). Algumas das tarefas mais comuns incluem reconhecimento de sinais de trânsito, segmentação de imagens biológicas e detecção de rostos, pedestres e corpos humanos em imagens naturais. No entanto, foi a partir da competição ImageNet em 2012 (*ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012*) que o uso de CNNs cresceu exponencialmente. Conjunto de treinamento padrão composto por milhares de imagens que combinados com GPUs eficientes e com o desenvolvimento de novas abordagens causaram uma revolução na visão computacional. Atualmente, a maioria dos trabalhos e estudos que usam CNNs para classificação de imagens aplicam a técnica de aprendizado por transferência, usando mo-

delos pré-treinados sobre o conjunto de dados ImageNet. Entre as arquiteturas mais famosas, podemos citar AlexNet (KRIZHEVSKY et al., 2012), GoogLeNet (SZEGEDY et al., 2015), MobileNet (HOWARD et al., 2017), VGG-Net (SIMONYAN; ZISSERMAN, 2014) e Inception-V3 (SZEGEDY et al., 2015). No entanto, como este trabalho utiliza a arquitetura MobileNet, a descreveremos com mais detalhes na seção seguinte.

#### 2.3.4.1 Arquitetura MobileNet

A arquitetura MobileNet, proposta inicialmente por Howard et al. (2017), foi projetada para oferecer arquiteturas CNN leves e eficientes para aplicações de visão em dispositivos móveis, ou seja, considerando restrições e características de *hardware* (como latência, consumo de energia ou uso de memória). A MobileNetV1 introduziu as *depthwise separable convolutions* para reduzir drasticamente o número de parâmetros e operações em CNNs (HOWARD et al., 2017; MOBILENET, 2024), isso quando comparado com outros modelos de CNN obtidos até então. A partir dessa primeira versão, as demais versões trouxeram aprimoramentos com ênfase na redução ainda maior do número de parâmetros da CNN, e na elevação de sua acurácia. Atualmente, em sua quarta versão, a MobileNetV4 apresentada por Qin et al. (2024), introduz ainda mais inovações como o *Universal Inverted Bottleneck* (UIB), composto pelo *Inverted Bottleneck*, *ConvNeXt*, *Feed Forward Networks* e uma variante *Extra Depthwise*, além do bloco de atenção *Mobile MQA* otimizado para aceleradores móveis, e um *pipeline* de NAS consciente de *hardware* que produz modelos quase Pareto-ótimos em CPUs, DSPs, GPUs, dentre outros. Comparado ao MobileNetV1, que emprega apenas *depthwise separable convolutions* com hiperparâmetros  $\alpha$  e  $\rho$  para ajustar o trade-off entre precisão e eficiência, o MobileNetV4 logra ganhos significativos de desempenho e redução de latência, mantendo ou diminuindo o custo computacional e o número de parâmetros em dispositivos móveis.

A Fig. 14 exemplifica a arquitetura da MobileNet em sua primeira versão.

#### 2.3.5 Métricas de Avaliação

Experimentos de classificação incluem diversas métricas de avaliação para analisar o desempenho da predição do modelo. Segundo Hossin e Sulaiman (2015), a mé-

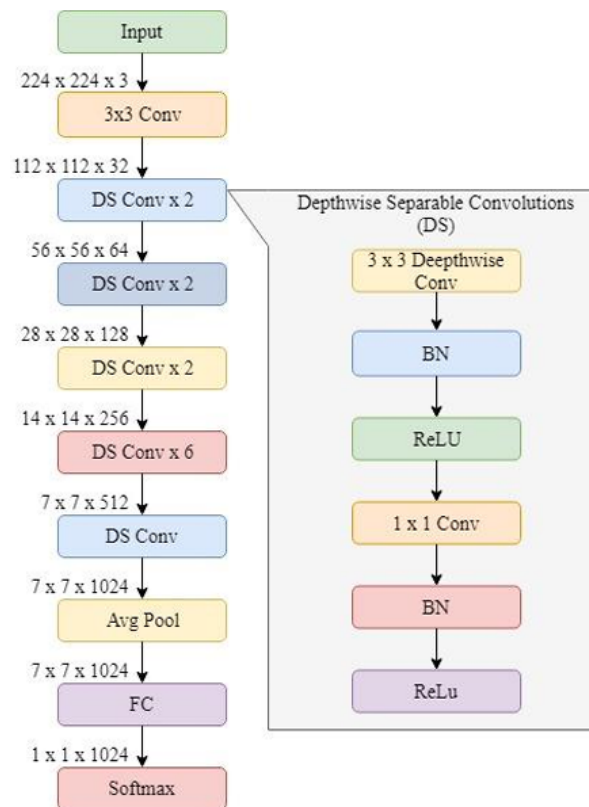


Figura 14 – Arquitetura geral da MobileNetV1 com destaque às convoluções separáveis em profundidade (DS). Fonte: Phiphatphaisit e Surinta (2020).

trica de avaliação pode ser descrita como a ferramenta de mensuração que mede o desempenho do classificador. Cada métrica diferente avalia uma característica distinta do classificador, mas são calculadas com base nos elementos da matriz de confusão. Esta seção descreve sua definição geral.

#### 2.3.5.1 Matriz de Confusão

A matriz de confusão exibe o número de amostras de teste previstas como certas e erradas. Em um exemplo de classificação binária (duas classes), ela pode ser representada pelo quadro 2 e é composta pelos seguintes elementos:

**Verdadeiros Positivos (TP):** Número de objetos classificados positivamente e que realmente pertencem à classe positiva.

**Verdadeiros Negativos (TN):** Número de objetos classificados negativamente e que realmente pertencem à classe negativa.

**Falsos Positivos (FP):** Número de objetos classificados positivamente, mas que na realidade pertencem à classe negativa.

**Falsos Negativos (FN):** Número de objetos classificados negativamente, mas que na realidade pertencem à classe positiva.

Quadro 2 – Matriz de confusão para classificação binária.

	<b>Actual Positive Class</b>	<b>Actual Negative Class</b>
<b>Predicted Positive Class</b>	True positive ( <i>tp</i> )	False negative ( <i>fn</i> )
<b>Predicted Negative Class</b>	False positive ( <i>fp</i> )	True negative ( <i>tn</i> )

Diversas métricas são derivadas das componentes da matriz de confusão para analisar o desempenho do modelo. Aqui, supondo uma hipotética feature do SPheRe, *regar apenas quando tiver certeza de que é necessário regar, e tolera deixar de regar em ocasiões em que poderia ter regado*, portanto, a ênfase seria na métrica Precisão de Classe Positiva.

#### 2.3.5.2 Acurácia

A *acurácia* é uma medida de desempenho que indica a proporção de classificações corretas em relação ao total de exemplos avaliados. Matematicamente, define-se pela equação (2.1):

$$Acurácia = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.1)$$

A acurácia varia entre 0 e 1 (ou 0 % a 100 %), sendo 1 correspondente a classificações perfeitamente corretas.

#### 2.3.5.3 Precisão para Classe Positiva

Esta métrica mede a proporção de predições positivas corretas em relação ao total de predições feitas como positivas. É definida pela equação (2.2):

$$Precisão_{pos} = \frac{TP}{TP + FP} \quad (2.2)$$

A precisão varia de 0 a 1 (ou 0 % a 100 %), com valor mais alto indicando menor taxa de falsos positivos. Ou seja, o termo  $FP$  no denominador de (2.2) penaliza os resultados falso positivos do teste.

#### 2.3.5.4 Precisão para Classe Negativa

Esta métrica mede a proporção de predições negativas corretas em relação ao total de predições feitas como negativas. É definida pela equação (2.3):

$$Precisão_{neg} = \frac{TN}{TN + FN} \quad (2.3)$$

A precisão varia de 0 a 1 (ou 0 % a 100 %), com valor mais alto indicando menor taxa de falsos negativos.

#### 2.3.5.5 Recall

A métrica também é chamada de "Taxa de Verdadeiros Positivos (TPR)". Ela mede qual proporção de verdadeiros positivos é classificada corretamente. Em outras palavras, é uma meta de otimização apropriada quando queremos identificar todos os positivos possíveis. É útil quando não queremos perder nenhum elemento positivo ou quando o custo de falsos negativos é alto. A Equação (2.4) calcula o recall:

$$Recall = \frac{TP}{TP + FN} \quad (2.4)$$

O valor de recall varia entre 0 e 1 (ou 0 % a 100 %), sendo 1 correspondente à identificação de todos os exemplos positivos sem falsos negativos.

#### 2.3.5.6 F1-Score

O  $F_1$ -score compila a média harmônica de Precisão e Recall, quantificando o trade-off mútuo. Um bom valor do F1-Score significa que o classificador identifica corretamente ameaças reais, mas não é perturbado por "alarmes falsos". Ele é calculado usando a equação (2.5):



$$F_1 = 2 \times \frac{Precisao \times Recall}{Precisao + Recall} \quad (2.5)$$

O  $F_1$ -score varia de 0 a 1 (ou 0 % a 100 %), atingindo 1 quando precisão e recall são ambos perfeitos.

Finalizando a fundamentação teórica, a seguir, uma visão geral sobre pesquisas na área da agricultura de precisão.

### 3 REVISÃO DA LITERATURA: AGRICULTURA DE PRECISÃO

Este capítulo descreve estudos recentes e abrangentes no campo da agricultura de precisão, enquanto a seção 3.4 exhibe soluções, comerciais e de código aberto, atualmente disponíveis.

#### 3.1 A TELEMETRIA NA AGRICULTURA

O estudo de Suciú et al. (2019) aponta que as mudanças climáticas e o crescimento populacional têm imposto desafios significativos à agricultura. Entre esses desafios, os autores destacam o racionamento hídrico na irrigação, tema que vem ganhando relevância nas pesquisas recentes devido à sua importância para a agricultura de precisão. O trabalho ressalta ainda que a qualidade da irrigação atua como um indicador de desempenho, variando conforme as necessidades específicas de cada cultura. Além disso, o estudo sublinha que a agricultura irrigada ocupa 20% das terras cultivadas globalmente, utilizando métodos como irrigação por gotejamento, vazamento superficial e aspersão.

Diante desse cenário, os autores defendem a adoção de sistemas de automação e telemetria, fornecendo dados contínuos relevantes que permitem otimizar o uso da água e dos insumos agrícolas, contribuindo diretamente para práticas mais sustentáveis. Nesse contexto, Alhasnawi et al. (2020) reforçam que a integração de sensores para coleta de dados e processamento inteligente de informações está criando uma conexão eficiente entre os ambientes cibernético e físico, promovendo decisões mais informadas e sustentáveis.

Além do impacto técnico e produtivo, essas tecnologias estão alinhadas aos Objetivos de Desenvolvimento Sustentável (ODS) da ONU. Por exemplo, o ODS 2 – “Fome Zero e Agricultura Sustentável” – é apoiado ao se aumentar a produtividade agrícola de forma eficiente e sustentável, garantindo segurança alimentar e nutrição adequada. O ODS 12 – “Consumo e Produção Sustentáveis” – é atendido pelo uso racional de recursos hídricos e insumos, promovendo padrões de produção mais responsáveis. Por fim, o ODS 11 – “Cidades e Comunidades Sustentáveis” – se beneficia indiretamente à tornar o fornecimento de alimentos mais resiliente e sustentável, contribuindo

para a estabilidade das comunidades rurais e urbanas.

Essa perspectiva é ampliada por Morchid et al. (2024)), que destacam como tais abordagens não só elevam a produtividade agrícola, mas também promovem a segurança hídrica e alimentar em escala global, reforçando a importância de tecnologias inteligentes para um desenvolvimento agrícola sustentável e alinhado às metas globais.

### 3.2 O PROTOCOLO HTTP APLICADO NA AGRICULTURA

Christensen e Fonseca (2023) destacam o ano de 2011 como marco na introdução e popularização do conceito de Indústria 4.0, inicialmente aplicado à manufatura. Com o tempo, esse conceito expandiu-se para outros setores, como a agricultura, resultando na Agricultura de Precisão. Tal abordagem enfatiza que o monitoramento e controle precisos otimizam o manejo agrícola.

O estudo apresenta a Agricultura 4.0 como uma evolução que substitui os métodos tradicionais por soluções digitais e tecnológicas — como computação de alto desempenho, rede de sensores, comunicação M2M (Machine-to-Machine), conectividade móvel, computação em nuvem e análise de grandes volumes de dados —, apoiando decisões de manejo e possibilitando a redução do consumo de água, fertilizantes e pesticidas por meio de aplicações mais precisas em áreas específicas.

Além disso, o trabalho integra IoT e Agricultura 4.0 para monitorar plantações de forma eficiente e sustentável, enviando dados relevantes para um banco de dados em nuvem via protocolo HTTP(Hypertext Transfer Protocol).

Bayılmış et al. (2022) apresentam o HTTP como o protocolo mais utilizado em sistemas IoT, inclusive na área da agricultura. Outros estudos, Percebes et al. (2023) e Anass et al. (2022), também utilizam o protocolo HTTP para integrar seus sistemas à nuvem, permitindo o controle remoto de atuadores, visualização, armazenamento e download das informações, abordagem bastante similar a este presente trabalho.

### 3.3 VISÃO COMPUTACIONAL APLICADA NA AGRICULTURA

A eficiência e a produtividade da agricultura dependem significativamente da supervisão humana contínua, o que exige tempo e recursos. O avanço das tecnologias

digitais e a crescente adoção de sistemas ciberfísicos na agricultura têm proporcionado novas oportunidades para a automação e otimização dos processos produtivos, conforme discutido no estudo de Dutta et al. (2025).

A aplicação da visão computacional na agricultura possibilita a automação de tarefas repetitivas, reduzindo o tempo necessário para execução e minimizando a dependência da supervisão humana. Dutta et al. (2025) enfatizam que a visão computacional tem se tornado uma ferramenta indispensável para o monitoramento agrícola, favorecendo a substituição da percepção visual humana por sistemas automatizados mais rápidos e precisos. Além disso, o uso de visão de máquina na agricultura tem crescido em resposta à necessidade de métodos ágeis e confiáveis para o rastreamento da colheita. A adoção de visão computacional viabiliza a análise eficaz de grandes volumes de dados, mesmo em cenários de alta complexidade, permitindo a tomada de decisões precisas e instantâneas. Por fim, a pesquisa discute detalhadamente a implementação da inteligência artificial (IA) na produção agrícola, especialmente no cultivo de frutas, abordando suas aplicações e perspectivas. A incorporação de sistemas baseados em IA nos processos produtivos representa um avanço significativo para a agricultura moderna, promovendo maior eficiência operacional, precisão e sustentabilidade no setor.

Islam et al. (2025) destacam que a precisão na identificação de variedades específicas de um determinado tipo de cereal sob seu estudo, é um processo fundamental tanto para prevenir fraudes quanto para assegurar a qualidade do produto. Nesse contexto, os autores também reforçam a importância da visão computacional na produção, com o objetivo de tornar o processo de identificação mais ágil e confiável.

No âmbito da detecção de doenças, Hossain et al. (2025) afirmam que as técnicas convencionais dependem significativamente da observação manual e da avaliação de especialistas, e que são frequentemente demoradas, trabalhosas e suscetíveis a discrepâncias. Os autores enfatizam que tais limitações tornam imprescindível empregar tecnologias de detecção de doenças que sejam automatizadas, escaláveis, confiáveis e instantâneas. Ainda afirmam que essa abordagem sugerida facilita o diagnóstico precoce, auxiliando agricultores e agrônomos a executar tratamentos oportunos e acurados, minimizando, assim, as perdas de safra.

Zamani e Baleghi (2025) também citam a aplicação de visão computacional para o controle de ervas daninhas, além de outras possíveis aplicações, como reconhe-

cimento de espécies de plantas, monitoramento contínuo do estado das culturas e colheita robótica de frutas.

Petrovic et al. (2023) expõem a respeito da aplicação da visão computacional na produção agrícola, afirmando que a indústria alimentícia é a que mais contribui para o setor agrícola, e a automação da triagem de vegetais é imperativo. O principal objetivo da classificação na agricultura é gerar cada vez mais renda. Portanto, a classificação tem um impacto significativo no agronegócio, gerando mais lucro. Sistemas inteligentes de classificação e triagem de culturas desempenham um papel crucial no aumento da eficiência e precisão na indústria agrícola. Esses sistemas utilizam tecnologias avançadas para automatizar o processo de triagem e classificação de culturas com base em vários parâmetros, como tamanho, cor, peso e qualidade. Para os agricultores, a triagem e a classificação de culturas permite que eles separem os produtos em categorias com mais precisão. A Fig.15 ilustra a seleção e triagem do tomate com aplicação de Visão Computacional.



Figura 15 – Exemplo de triagem e seleção de frutos de tomate por meio de Visão Computacional. Fonte: Petrovic et al. (2023)

Esses estudos destacaram a visão computacional sendo empregada para automatizar tarefas manuais, identificar doenças e pragas, classificar variedades agrícolas e otimizar a produção.

A seção seguinte apresenta uma breve descrição das soluções atualmente disponíveis no mercado para viabilizar, especificamente, a automação de estufas e tendas de crescimento.

### 3.4 SOLUÇÕES DISPONÍVEIS NA AGRICULTURA DE PRECISÃO

#### 3.4.1 Soluções Comerciais

O objetivo primordial da agricultura de precisão é criar um microclima ideal que maximize a produtividade e a qualidade dos cultivos. Nesse contexto, os principais parâmetros monitorados em ambientes fechados – como umidade do solo, umidade e temperatura do ar, luminosidade e níveis de CO<sub>2</sub> – devem ser controlados de forma precisa. Além do monitoramento por meio de sensores, é fundamental a intervenção automatizada, realizada por atuadores que ajustam essas variáveis ambientais conforme a necessidade.

Como exemplo de controlador de temperatura e umidade, destaca-se o Inkbird ITC-608T, Fig.16, da empresa Inkbird, que mede esses parâmetros e pode ser configurado para acionar equipamentos de aquecimento, resfriamento, umidificação ou desumidificação do ar.



Figura 16 – InkbirdITC608T. Fonte: Inkbird (2025b).

No âmbito da irrigação, uma das soluções é o Inkbird IIC-800-WIFI, Fig.17, da mesma empresa, permitindo o agendamento de horários específicos para a rega.



Figura 17 – Inkbird IIC-800-WIFI. Fonte: Inkbird (2025a).

Em relação à ventilação, o sistema Cloudline T4, Fig. 18, da AC Infinity, consiste em um ventilador de duto inteligente, projetado para manter a circulação de ar em ambientes fechados.



Figura 18 – CloudlineT4. Fonte: AC Infinity (2025).

Esta solução possibilita o controle dos níveis de temperatura e umidade, além de oferecer telemetria para acompanhamento remoto.

Para a gestão de instalações hortícolas, a empresa Argus Controls disponibiliza o Argus Titan System, Fig. 19, um sistema que abrange desde o controle ambiental até fertirrigação, permitindo o monitoramento e a automação dos processos por meio de um computador em rede ou dispositivos móveis.



(a) Controlador Argus.



(b) Monitoramento de estufa.

Figura 19 – Sistema Argus Titan. Fonte: Argus Controls (2025).

Para otimizar o manejo do microclima, torna-se fundamental a inclusão de parâmetros mais sofisticados, tais como a medição do pH e da condutividade elétrica (EC) do solo, o monitoramento do oxigênio dissolvido na água, a análise da qualidade espectral da iluminação artificial e a detecção de pragas e doenças. Essas variáveis complementam os controles convencionais e possibilitam ajustes precisos que podem elevar a produtividade e a qualidade dos cultivos.

No que se refere ao monitoramento do pH e da EC, o Bluelab OnePen, Fig.20, da Bluelab, permite a medição desses parâmetros, fornecendo informações essenciais para avaliar a disponibilidade de nutrientes no solo.



Figura 20 – Bluelab OnePen. Fonte: Bluelab (2025).

Em sistemas hidropônicos, a saúde radicular e a prevenção de estresses podem ser asseguradas com o uso do Dissolved Oxygen Kit, da empresa Atlas Scientific, que mede o oxigênio dissolvido em soluções.

Quanto à qualidade da iluminação artificial, a análise do espectro luminoso é crucial para ajustar a iluminação de acordo com as necessidades específicas das culturas em diferentes fases de crescimento. Para isso, o sensor AS7341, Fig.21, da Adafruit, realiza a avaliação da qualidade espectral da luz, contribuindo para a otimização da iluminação.

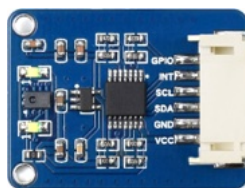


Figura 21 – AS7341 Adafruit. Fonte: Adafruit (2025).

Por fim, para a detecção precoce de pragas e doenças, embora ainda não exista um sensor comercial específico para essa finalidade, a utilização de tecnologias de visão computacional se mostra crucial para este fim. Estudos, como o artigo *Green Leaf Disease Detection System for Agriculture Using Raspberry Pi* de BABU et al. (2024), por exemplo, destacam o Raspberry Pi, que, aliado à inteligência artificial e câmera, é eficaz na análise de imagens para identificar sinais iniciais de infestação e sintomas de doenças nas plantas.

Cada uma dessas soluções comerciais ilustra a aplicação prática de tecnologias atuais na automação de ambientes controlados, contribuindo significativamente na implementação da agricultura de precisão em casas de vegetação.

Na próxima seção, condensaremos algumas das soluções open source (também conhecidas como código aberto) disponíveis.





Figura 22 – Raspberry Pi e Módulo Câmera. Fonte: IoT-Store (2025).

### 3.4.2 Soluções Open Source

A comunidade tem investido no desenvolvimento de sistemas open source e projetos DIY (faça-você-mesmo) mais acessíveis, preenchendo lacunas deixadas pelas soluções comerciais, permitindo uma maior personalização e a integração de tecnologias emergentes. Nesse contexto, destacam-se as seguintes iniciativas:

#### 3.4.2.1 Sistemas Baseados em Microcontroladores

Plataformas como Arduino, Raspberry Pi e ESP32 possibilitam a integração de diversos sensores capazes de medir variáveis essenciais, tais como umidade do solo, pH, temperatura, luminosidade e oxigênio dissolvido, além de atuadores para o controle da irrigação e ventilação. Essas soluções, amplamente compartilhadas em fóruns e repositórios open source, permitem que agricultores e pesquisadores customizem os sistemas de acordo com as necessidades específicas de cada cultivo.

#### 3.4.2.2 FarmBot

O FarmBot, Fig.23 é uma máquina CNC agrícola open source, desenvolvida para automatizar pequenas hortas e cultivos urbanos. Ele integra sensores, atuadores e um sistema de monitoramento remoto, possibilitando o cultivo automatizado, desde o plantio até a colheita.



Figura 23 – FarmBot. Fonte: FarmBot (2025).

#### 3.4.2.3 AgOpenGPS

O projeto AgOpenGPS, Fig.24 é uma solução open source que busca a orientação autônoma de máquinas agrícolas. Por meio do uso de dados de GPS e sensores, o sistema mapeia e direciona operações no campo, contribuindo para a precisão na aplicação de insumos e para a eficiência na colheita.

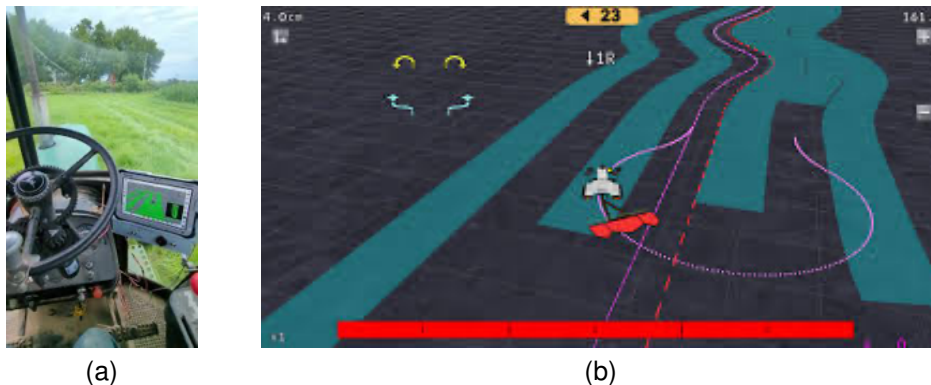


Figura 24 – Sistema AgOpenGPS: (a) máquina agrícola autônoma e (b) mapeamento de operações em campo. Fonte: AgOpen Shop (2025).

#### 3.4.2.4 Plataformas IoT Colaborativas

Diversas iniciativas desenvolvidas por universidades e comunidades online focam na integração de dados de sensores, imagens (através de visão computacional) e previsões meteorológicas utilizando algoritmos de ML. Essas plataformas possibilitam a criação de dashboards customizados e sistemas de alerta, que facilitam intervenções

rápidas e são adaptados às condições específicas de cada cultivo. Exemplos notáveis incluem o ThingsBoard, ThingSpeak e FarmOS, os quais viabilizam a coleta, o processamento e a visualização de dados essenciais para a tomada de decisão em projetos colaborativos e acadêmicos.

Em suma, as abordagens colaborativas demonstram o potencial das tecnologias open source para a agricultura de precisão, permitindo a implementação de sistemas adaptáveis, que atendem às demandas específicas dos produtores onde as soluções comerciais se mostram insuficientes.

A seguir, discute-se as pesquisas mais alinhadas com este trabalho.

## 4 REVISÃO DA LITERATURA: TRABALHOS RELACIONADOS

Neste capítulo, apresentamos os estudos mais relevantes para esta pesquisa, que foram analisados ao longo do desenvolvimento do SPheRe e desta dissertação. Um número crescente de pesquisas sobre sensoriamento na agricultura tem se apoiado em novas tecnologias e nos avanços da visão computacional. Por fim, o capítulo encerra com um resumo das principais oportunidades de contribuição para o avanço dessa área.

### 4.1 AVANÇOS EM TECNOLOGIAS DE SENSORES DE UMIDADE DO SOLO

Yu et al. (2021) em “*Review of research progress on soil moisture sensor technology*” apresentam uma análise abrangente dos avanços nos sensores de umidade do solo ao longo das últimas décadas, ressaltando a importância de realizar medições precisas do teor de água no solo, tanto para o manejo da irrigação quanto para a garantia da produtividade das culturas. O estudo, assim como esta dissertação, contextualiza a relevância da medição da umidade do solo, ressaltando os desafios decorrentes da disponibilidade limitada de água doce e a necessidade de um uso mais eficiente deste recurso.

Essa revisão detalha diversos métodos empregados na medição da umidade do solo. São abordadas técnicas convencionais, como o método gravimétrico (ou método de secagem), e métodos mais avançados, como os sensores baseados em tensiômetros, sondas de nêutrons, transmissão de raios gama, sensoriamento remoto por infravermelho e, principalmente, os métodos dielétricos. Dentro desta última categoria, o artigo descreve com ênfase os princípios do TDR (Time Domain Reflectometry), FDR (Frequency Domain Reflectometry) e SWR (Standing Wave Ratio), demonstrando as vantagens e limitações de cada técnica em termos de precisão, custo e aplicabilidade.

Além da análise dos métodos de medição, o estudo discute os fatores que influenciam o desempenho dos sensores de umidade. São considerados fatores intrínsecos, como o design do sensor, a instalação e a calibração, bem como fatores ambientais, como variações de temperatura e salinidade do solo. Essas variáveis impactam diretamente a exatidão e a estabilidade das medições, exigindo a adoção de técnicas de

calibração específicas para diferentes tipos de solo e condições ambientais, a fim de garantir resultados confiáveis.

Por fim, Yu et al. (2021) apontam tendências futuras para o desenvolvimento dos sensores de umidade do solo, enfatizando a necessidade de sistemas que sejam de alta precisão, baixo custo, não destrutivos e integrados com tecnologias modernas, como a IoT. A integração desses sensores em sistemas de irrigação inteligente pode revolucionar o manejo hídrico na agricultura, permitindo a tomada de decisões mais rápidas e fundamentadas. Em síntese, esse estudo estabelece uma base de conhecimento que pode auxiliar tanto pesquisadores quanto profissionais do setor agrícola na seleção e no aprimoramento de produtos para medição de umidade do solo.

#### 4.2 ESTIMATIVA DA ABSORÇÃO DE ÁGUA DE IRRIGAÇÃO COM RGB

Atanasov (2021) em “*Methodology for irrigation water uptake time estimation based on RGB colorimetric measurements of leaves (a visual-graphical observation)*” propõe uma metodologia peculiar para estimar o tempo de absorção de água de irrigação por plantas de tomate em estufa, utilizando medições colorimétricas em RGB nas folhas. A abordagem se baseia na observação e na análise gráfica das mudanças na cor das folhas após a irrigação, assumindo que o clareamento das folhas indica a chegada da umidade numa seção da planta. Essa hipótese, que sugere um tempo de 1 hora para cada 1 metro de altura, é testada experimentalmente sob condições controladas.

A pesquisa se apoia em uma série de experimentos realizados ao longo de cinco dias, nos quais a variedade de tomate híbrido “Amati” foi utilizada devido à sua alta resistência a doenças e características morfológicas favoráveis. Em cada experimento, as medições RGB foram coletadas imediatamente antes e em diversos intervalos após a irrigação, em diferentes alturas da planta, para mapear com precisão o tempo de transporte da água até a parte superior.

As Fig. 25 e 26 exemplificam imagens de folhas e medições com colorímetro durante o experimento dessa pesquisa.

A metodologia empregada combina observações imediatas e mediadas, utilizando um colorímetro para quantificar as mudanças na tonalidade das folhas. Atanasov (2021) realiza análises estatísticas, como testes T de amostras pareadas, para confirmar que as variações nos valores RGB são estatisticamente significativas, garantindo

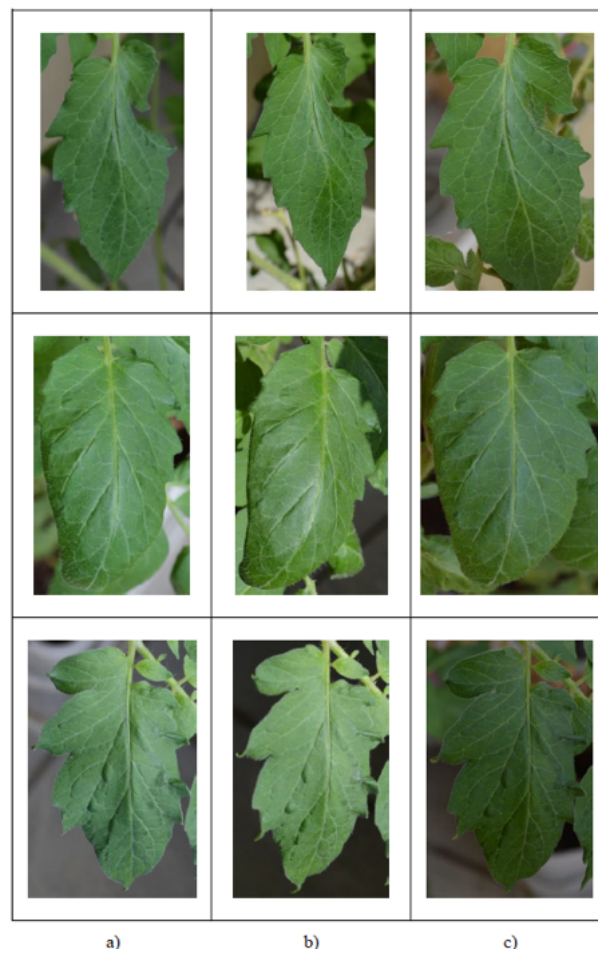


Figura 25 – Exemplos de fotos de folhas examinadas ao longo do tempo: a) Antes da rega, b) 120 min após a rega e c) 300 min após a rega. Cada linha representa a mesma folha. Fonte: (ATANASOV, 2021).

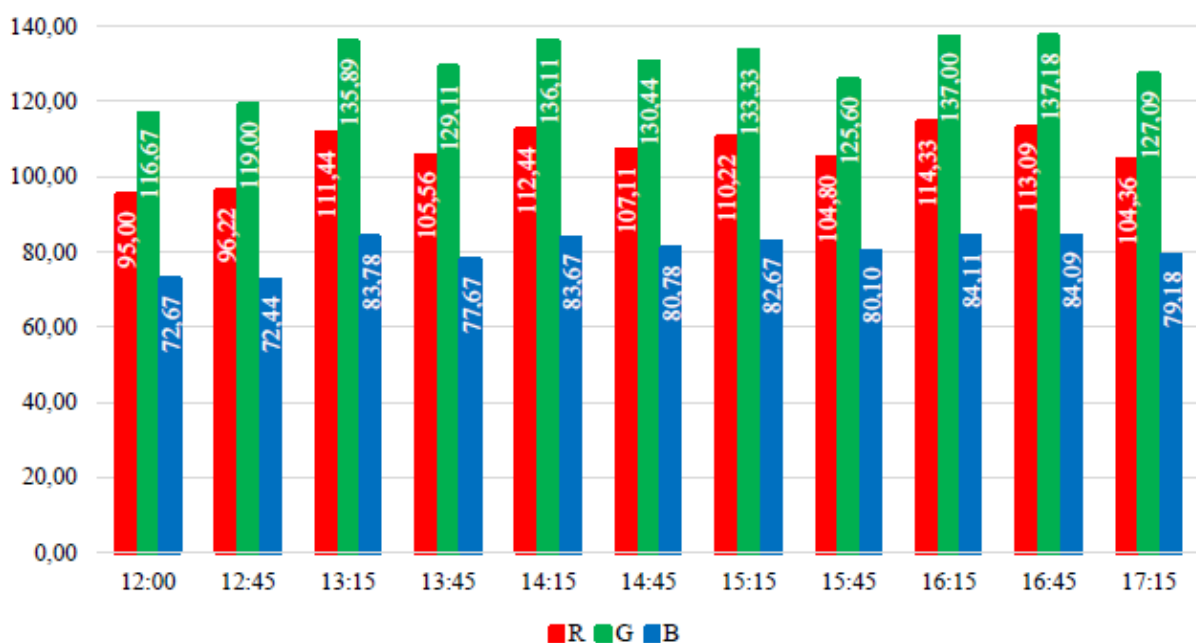


Figura 26 – Alteração dos valores RGB médios no tempo antes e depois da rega. Fonte: (ATANASOV, 2021).

que o clareamento observado não decorre do acaso, mas sim da absorção efetiva de água.

Os resultados indicam que, sob as condições experimentais descritas, a água da irrigação absorvida, atinge o seu primeiro metro de altura na planta em aproximadamente 60 minutos, e 80 minutos para atingir 1,3 metros e 90 minutos para 1,6 metros de altura.

Por fim, o estudo demonstra que a análise de mudanças na cor das folhas via medições RGB pode ser uma ferramenta prática, não invasiva e eficiente para monitorar a absorção de água em plantas dessa espécie de tomate, contribuindo para a otimização do manejo da irrigação.

#### 4.3 VISÃO COMPUTACIONAL PARA FENOTIPAGEM DE CARACTERÍSTICAS DE PLANTAS

Kolhar e Jagtap (2021) em *“Plant trait estimation and classification studies in plant phenotyping using machine vision – A review”* apresentam uma revisão abrangente dos métodos de visão de máquina aplicados à fenotipagem de plantas, enfatizando a estimativa e classificação de traços estruturais e fisiológicos. São abordadas diversas técnicas de imagem, incluindo a captura de imagens RGB, hiperespectrais, térmicas e 3D, que permitem a análise não invasiva de características como área foliar, contagem e forma de folhas, bem como indicadores fisiológicos, como conteúdo de água e eficiência fotossintética.

Além disso, Kolhar e Jagtap (2021) destacam a aplicação de algoritmos de ML e deep learning para aprimorar o processamento de imagens e a segmentação de traços das plantas. Modelos CNNs, algoritmos de *clustering* e técnicas supervisionadas, como máquinas de vetor de suporte (do inglês SVM), são empregados para extrair características estruturais detalhadas e lidar com desafios como folhas sobrepostas e variações de iluminação. Essa abordagem possibilita uma quantificação precisa e automática dos parâmetros morfológicos, contribuindo para análises em larga escala.

Por fim, a revisão enfatiza que técnicas de visão de máquina têm revolucionado a fenotipagem de plantas e que a integração de processamento de imagem com algoritmos de inteligência artificial não só aprimora a identificação de traços estruturais, mas também facilita a avaliação de respostas fisiológicas das plantas a fatores ambientais

e de manejo, abrindo caminho para avanços significativos em melhoramento genético e manejo agrícola.

#### 4.4 PREVISÃO FENOTÍPICA DO STATUS HÍDRICO DO SUBSTRATO

Chang et al. (2019) em “*A Phenotype-Based Approach for the Substrate Water Status Forecast of Greenhouse Netted Muskmelon*” apresentam uma abordagem baseada em fenotipagem para prever o estado hídrico do substrato em cultivos de *melão rendilhado* (*Cucumis melo L. var. reticulatus Naud.*) em estufa. Para isso, os autores utilizam um sistema fenotípico comercial que captura imagens em espectros visível e infravermelho, conforme ilustrado na Fig.27. A figura destaca imagens de plantas no âmbito da luz visível e no espectro do infra-vermelho próximo, permitindo a extração de informações de morfologia, de cor e quantidade de água das plantas em diferentes estágios de crescimento. Esses aspectos, que incluem desde a área de projeção até índices de cor e contagem de pixels com intensidade específica no NIR (*Near InfraRed* - Infravermelho Próximo), são computados por meio de etapas de pré-processamento e segmentação para, em seguida, alimentar um modelo de classificação baseado em algoritmo de random forest. Esse modelo, validado por análises estatísticas e cruzadas, demonstra alta acurácia na previsão do conteúdo de água do substrato, facilitando a tomada de decisão para irrigação de precisão.

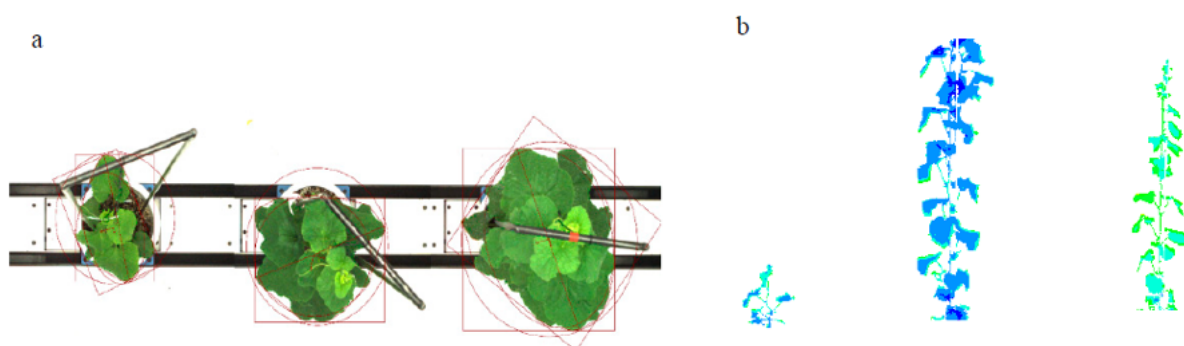


Figura 27 – Etapas de processamento de imagens fenotípicas. (a) Imagens dentro do espectro visível; (b) Imagens no espectro infravermelho-próximo. Cores diferentes representam diferentes conteúdos de água nas plantas. Fonte: (CHANG et al., 2019)

Embora o foco principal do estudo seja a previsão do estado hídrico do substrato, o artigo destaca a aplicação de técnicas de visão de máquina para a extração automatizada de traços fenotípicos. O uso do sistema Scanalyzer 3D e a subsequente



análise de imagens demonstram como métodos de segmentação e extração de características – típicos da visão computacional – podem ser empregados para determinar características estruturais e fisiológicas das plantas. Essa integração de imagem e ML exemplifica o potencial da fenômica baseada em visão de máquina para monitoramento em tempo real e para o desenvolvimento de sistemas de irrigação inteligentes que respondem dinamicamente às condições do cultivo.

#### 4.5 MODELOS DE IMAGENS COM SMARTPHONES PARA ESTIMATIVA DE CLO-ROFILA

Özreçberoğlu e Kahramanoğlu (2020) em “*Mathematical models for the estimation of leaf chlorophyll content based on RGB colours of contact imaging with smartphones: A pomegranate example*” desenvolveram um método não destrutivo e rápido para estimar o conteúdo de clorofila em folhas de romã utilizando imagens capturadas por smartphones. As imagens são obtidas em condições controladas dentro de caixas fechadas, usando fontes de luz LED (com destaque para a luz vermelha, que demonstrou melhores correlações) para iluminar as folhas. A partir dessas imagens, são extraídos os valores de cor RGB e calculados índices de cor (como os índices normalizados e diferenças entre canais), que servem como base para a estimativa da clorofila.

A análise dos dados envolve regressão linear para relacionar os valores de cor e os índices de clorofila. O modelo resultou em uma equação que combina valores dos canais verde (G) e azul (B). Os resultados demonstram que a metodologia proposta pode fornecer estimativas precisas de clorofila, contribuindo para o monitoramento da saúde das plantas e a gestão sustentável da nutrição vegetal.

#### 4.6 ABORDAGENS DE APRENDIZADO PROFUNDO PARA AVALIAÇÃO DE CLO-ROFILA FOLIAR

Barman e Saikia (2024) em “*Smartphone Contact Imaging and 1-D CNN for Leaf Chlorophyll Estimation in Agriculture*” apresentam uma metodologia de baixo custo para estimar o teor de clorofila em folhas utilizando apenas imagens capturadas por smartphones. O trabalho parte da premissa de que o teor de clorofila é um indica-

dor importante do estado nutricional e da saúde da planta, tradicionalmente medido por sensores específicos como o SPAD e espectrômetros, que são caros e pouco acessíveis para pequenos agricultores. Nesse contexto, os autores propõem o uso de dispositivos móveis como alternativa acessível e prática para agricultores e pesquisadores.

A pesquisa utilizou aproximadamente 15.000 imagens de folhas de chá capturadas em contato direto com a câmera de um smartphone, sob iluminação controlada e com o uso do flash. A partir das imagens foram extraídas 12 características de cor, considerando estatísticas dos canais RGB e HSV. Essas variáveis foram usadas como entrada em diferentes modelos de predição, incluindo regressão linear, redes neurais tradicionais e uma rede neural convolucional unidimensional (1-D CNN), escolhida por sua capacidade de aprender padrões complexos a partir de dados sequenciais.

Os resultados mostraram que a abordagem baseada em 1-D CNN foi a mais eficaz, superando os demais métodos testados. O modelo atingiu um coeficiente de regressão em torno de 0,82 e erro médio absoluto (MAE) de aproximadamente 2,96 na estimativa do teor de clorofila, quando comparado às leituras do sensor SPAD. Os autores concluem que o uso de smartphones aliado a técnicas de visão computacional e inteligência artificial representa uma alternativa promissora para o monitoramento nutricional de plantas, democratizando o acesso a tecnologias de agricultura de precisão.

#### 4.7 IMAGENS ESPECTRAIS E RGB COM IA PARA STATUS HÍDRICO DE PLANTAS

No estudo *“Water status and plant traits of dry bean assessment using integrated spectral reflectance and RGB image indices with artificial intelligence”*, El-baki et al. (2025) investigam o uso de aprendizado de máquina para monitoramento de plantas em condições de estresse hídrico. O estudo parte da necessidade de superar limitações de sensores de solo tradicionais, que apresentam cobertura espacial reduzida e deterioração ao longo do tempo, propondo a análise de imagens digitais como alternativa não invasiva para avaliar o estado hídrico das plantas. A pesquisa enfatiza o potencial da visão computacional para capturar alterações fenotípicas sutis e antecipar sinais de déficit de água.

A metodologia consistiu na aquisição de imagens de plantas submetidas a diferen-

tes níveis de irrigação, das quais foram extraídas características de cor, forma e textura. Esses atributos serviram de entrada para modelos de aprendizado de máquina, incluindo algoritmos de classificação e regressão, que foram avaliados em termos de acurácia e robustez na predição do estresse hídrico. Os resultados mostraram que os modelos conseguiram diferenciar de forma consistente plantas sob condições normais e sob déficit hídrico, com desempenho superior às técnicas convencionais de monitoramento baseadas apenas em sensores de solo.

Na conclusão, os autores destacam que o emprego de visão computacional e aprendizado de máquina representa um avanço significativo para a agricultura de precisão, possibilitando um monitoramento mais abrangente, de baixo custo e com menor necessidade de manutenção. O estudo sugere que, com a evolução de algoritmos e dispositivos embarcados, será viável a implementação de sistemas inteligentes de irrigação em tempo real, capazes de reduzir perdas e aumentar a eficiência no uso da água.

#### 4.8 RESUMO DO CAPÍTULO

A cor das folhas é a característica mais simples de ser extraída de imagens de plantas e pode refletir a umidade do solo, como demonstrado para o tomateiro por Atanasov (2023). Se essa relação for confirmada em outras culturas, imagens RGB poderão atuar como sensores de umidade de baixo custo e não invasivos, permitindo à visão de máquina identificar as necessidades hídricas das plantas pela coloração foliar.

## 5 MATERIAIS E MÉTODOS

Este capítulo descreve os materiais e métodos empregados na composição do SPheRe, laboratório de fenotipagem inteligente, um sistema concebido para monitorar periodicamente a umidade do solo, capturar imagens de plantas e enviar esses dados para a nuvem, além de possibilitar a irrigação automática de acordo com pré-configuração definida pelo usuário-pesquisador. O sistema foi desenvolvido como uma solução de baixo custo e automatizada para monitoramento fenotípico, correlacionando dados visuais com condições de estresse hídrico.

A abordagem metodológica adotada foi prática, iterativa e baseada em prototipagem de hardware com integração à nuvem. As etapas seguiram uma sequência lógica que incluiu a definição da arquitetura, testes e regulação de sensores (pseudo calibração), montagem da estrutura física de cultivo e validação funcional dos sensores e atuadores. Parte desses procedimentos, inicialmente apresentados como resultados — em especial aqueles relacionados à otimização de parâmetros do sistema, como iluminação, distância da câmera e uso de referências visuais — são aqui tratados como elementos metodológicos, pois dizem respeito ao processo de concepção e consolidação da abordagem.

Deve-se enfatizar que este trabalho não teve como objetivo a realização de experimentos que permitissem análises estatísticas aprofundadas, como repetições em larga escala com múltiplos cultivares ao longo de períodos extensivos. Embora essa fosse a intenção inicial, diversas dificuldades surgiram durante a fase de prototipagem, incluindo falhas recorrentes nos sensores de umidade e problemas na configuração das imagens capturadas. Após etapa de otimização do setup — brevemente comentada ao final deste capítulo na Seção 5.15 —, o caráter experimental limitou-se à execução de uma prova de conceito, com análises predominantemente qualitativas destinadas a demonstrar a viabilidade técnica da proposta. Assim, a metodologia concentrou-se na descrição da concepção e do funcionamento do sistema, abrangendo desde a integração dos componentes até a implementação de algoritmos de aprendizado de máquina em microcontroladores de baixo custo.

No âmbito metodológico, duas hipóteses distintas foram consideradas. A primeira buscou avaliar a capacidade da visão computacional em subsidiar decisões de irriga-

ção em cenários ideais de processamento, como o uso de computação em nuvem e imagens com resolução satisfatória. A segunda concentrou-se em cenários restritos, nos quais a aquisição de imagens ocorre em baixa resolução e o processamento é realizado em dispositivos embarcados de baixo custo, com recursos computacionais limitados. Essas duas perspectivas — uma voltada ao potencial teórico e outra à aplicação prática — foram tratadas de forma complementar, permitindo avaliar tanto a eficácia conceitual da abordagem quanto sua viabilidade em contextos reais de uso agrícola.

### 5.1 VISÃO GERAL DA METODOLOGIA

A Fig. 28, a seguir, apresenta uma visão geral do sistema desenvolvido, com seus principais módulos físicos e suas interações.

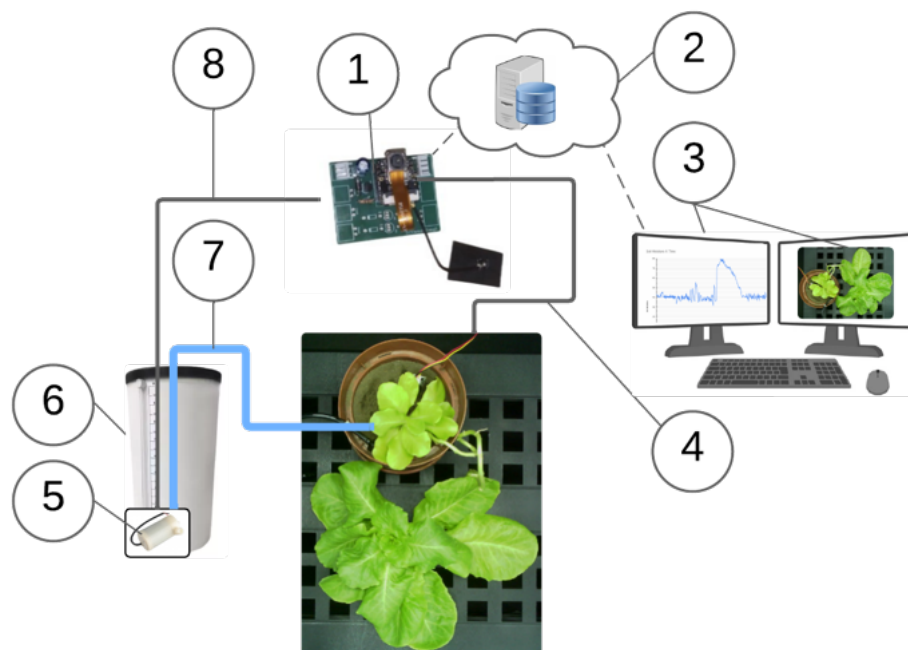


Figura 28 – O sistema de fenotipagem desenvolvido, *SPheRe*. Fonte: o autor.

A arquitetura *SPheRe* contempla os principais módulos responsáveis pela aquisição, transmissão e processamento de dados, além do controle do sistema de irrigação.

Esses componentes estão descritos a seguir:

1. **Módulo de Controle:** Com o microcontrolador ESP32-S3 (Espressif Systems) e câmera em sua composição, é responsável pelo periódico monitoramento e controle da irrigação, e pela captura de imagens da planta sob cultivo. Enviando esses dados à internet via Wi-Fi. É Alimentado por uma fonte DC externa de 5 V, e possui conversor de tensão que fornece 3.3 V<sub>DD</sub> ao SoC (System-on-Chip);
2. **Servidor na Nuvem:** armazena imagem e umidade do solo enviadas pelo **Módulo de Controle**, possibilitando posterior análise remota;
3. **Computador Remoto:** para visualização e processamento dos dados disponibilizados pelo **Servidor na Nuvem**, a fim de analisar os dados recebidos e configurar o sistema de irrigação;
4. **Sensor de Umidade Capacitivo:** inserido no solo do vaso de cultivo, realiza a medição contínua da umidade, dispondo o dado para o **Módulo de Controle** por meio de barramento analógico;
5. **Mini Bomba d'Água:** modelo JT100, instalada no interior do **Reservatório de Água**;
6. **Reservatório de Água:** recipiente em PVC com capacidade de armazenamento de até 5 L de água para a irrigação;
7. **Sistema de Irrigação por Gotejamento:** duto para condução da água a partir do **Reservatório de Água** até o vaso de cultivo;
8. **Barramento de Ativação:** canal de controle digital entre o **Módulo de Controle** e a **Mini Bomba d'Água**, responsável por acionar a bomba para a irrigação da planta.

## 5.2 ARQUITETURA GERAL DO SPHERE

A arquitetura do SPheRe é composta por módulos interconectados que integram sensores, unidade de controle, elemento atuador e infraestrutura de comunicação. O

fluxo de dados é contínuo e estruturado, permitindo o monitoramento da planta e da umidade de seu solo, com a rega configurada remotamente pelo usuário-pesquisador.

A Fig. 29 apresenta o diagrama de blocos da solução, em seguida, os passos de seu funcionamento são descritos.

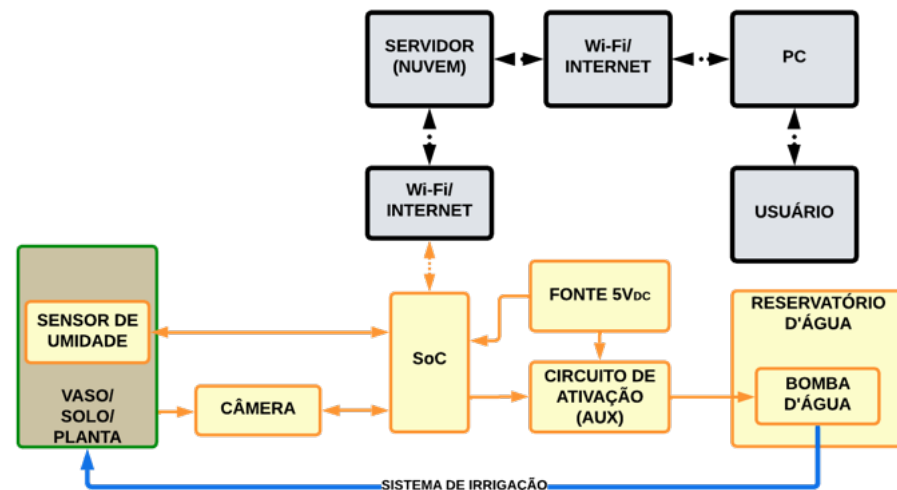


Figura 29 – Diagrama em Blocos *SPheRe*. Fonte: o autor.

1. **Aquisição de dados no ambiente de cultivo:** coleta de dado de umidade através do sensor de umidade capacitivo e captura de imagem por meio da câmera OV5640;
2. **Leitura e controle pelo SoC:** os dados são processados e preparados para envio;
3. **Envio à nuvem:** os dados são enviados via Wi-Fi para Google Sheets (dados) e Google Drive (imagens);
4. **Análise remota:** o usuário-pesquisador analisa as imagens com scripts em Python, lê dados de umidade do solo e configura a irrigação;
5. **Acionamento da bomba:** Dependendo dos limiares de histerese pré-configurados ou do comando de rega aplicado pelo usuário, o SoC ativa a bomba d'água com o auxílio de um circuito de ativação, liberando aproximadamente 100 mL de água por rega.

### 5.3 COMPONENTES DE HARDWARE UTILIZADOS

O núcleo do sistema é a placa de desenvolvimento Seeed Studio XIAO ESP32S3 Sense, representada na Fig. 30, (1).

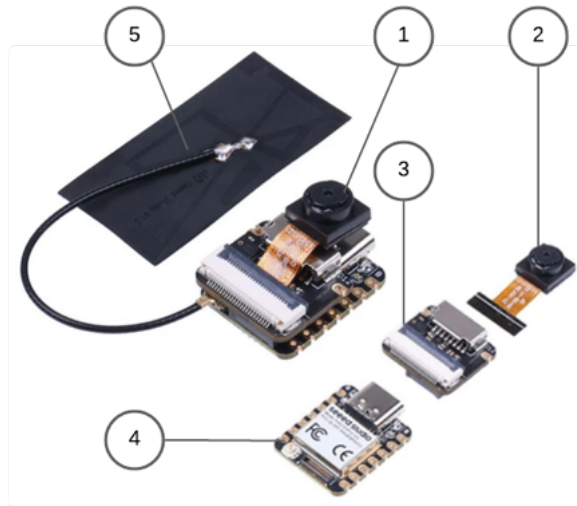


Figura 30 – Seeed Studio XIAO ESP32S3 Sense. Fonte: Seeed Studio (2025).

Essa placa é alimentada por fonte externa de 5 V e equipada com câmera OV5640 de 5 megapixels, Fig. 30 (2), conectada à placa de expansão (3). O SoC ESP32-S3R8, (4), é dual-core, com conectividade Wi-Fi, e dispõe de antena flexível, (5), um sistema adequado para aplicações embarcadas de monitoramento remoto e viável para implantação de IA.

Para a medição da umidade do solo, foi adotado o sensor do tipo capacitivo, Fig. 31.



Figura 31 – Sensor capacitivo de umidade de solo. Fonte: o autor.



Sensor de baixo custo, escolhido por sua durabilidade superior em comparação aos sensores resistivos.

#### 5.4 CONTEXTO EXPERIMENTAL

O SPheRe consiste em um sistema de fenotipagem instalado em uma tenda de cultivo localizada no laboratório “Estufa” do Centro de Informática da UFPE. Seu desenvolvimento inicial foi apresentado no *XIV Symposium on Computing Systems Engineering* (SOUZA; BLAWID, 2024). O SPheRe se trata de um ambiente sob condições controladas de iluminação e irrigação, conforme ilustrado na Fig. 32(a), contendo prateleiras para acomodação das plantas envasadas, iluminação artificial programável, exaustor e o módulo embarcado de sensoriamento.



(a) Tenda de cultivo



(b) Irrigação por gotejamento



(c) SoC com câmera



(d) Sensor de umidade

Figura 32 – Sistema de fenotipagem na tenda de crescimento (a). Sistema de irrigação por gotejamento (b). O módulo posicionado acima da planta em estudo a uma distância de 50 cm (c). Degradação do sensor capacitivo por infiltração (d). Fonte: Souza e Blawid (2024).

A Fig. 32(b) destaca o sistema de irrigação por gotejamento, acionado por uma bomba d'água instalada externamente à tenda, enquanto o módulo (c), posicionado a aproximadamente 50 cm acima da planta, realiza a captura de imagens e leituras de umidade. A Fig. 32(d) mostra um sensor capacitivo danificado por infiltração durante testes em solo extremamente úmido, justificando sua posterior impermeabilização com verniz para circuitos eletrônicos, proporcionando maior resistência à umidade e durabilidade.

## 5.5 PSEUDO-CALIBRAÇÃO DO SENSOR DE UMIDADE

Para assegurar a coerência nas leituras do sensor capacitivo adotado, foi realizado um ajuste comparativo, utilizando como referência um sensor de umidade comercial (modelo ST-03), conforme ilustrado na Fig. 33(a). A regulação foi feita com dois vasos plásticos contendo, aproximadamente, 1 kg de terra para jardinagem cada, ambos com furos de drenagem. Um dos vasos permaneceu com solo seco, enquanto o outro foi saturado até escorrer água, resultando em solo altamente úmido, como mostra a Fig. 33(b).



Figura 33 – Itens usados na regulação do sensor sob estudo. Fonte: o autor.

A fim de garantir uniformidade, o solo de ambos os vasos foi homogeneizado com auxílio de espátula. O sensor ST-03 foi inserido no vaso com solo seco, realizando-se 14 medições em diferentes pontos, resultando numa média de 4% de umidade relativa. No vaso úmido, o mesmo procedimento indicou 59% de umidade relativa média.

Em seguida, repetimos o mesmo processo aplicando o sensor de umidade capacitivo sobre os dois vasos. Com SoC configurado para uma leitura analógica de 9 bits (0 a 511) de resolução, correspondentes a 0 e 3.3 V, respectivamente, iniciamos a leitura analógica no solo seco, medindo o valor 330, enquanto no solo úmido, 214. Considerando esses dois pontos, deduziu-se uma função linear de primeiro grau para estimar a umidade relativa do solo: se concluiu, portanto, que leituras de 338 correspondem a 0% e leituras de 128 a 100%. Feito isso, essa função linear passou a mapear as leituras seguintes, limitadas entre 0% e 100%. Valores intermediários foram observados durante os testes.

## 5.6 PLANTA ALVO E ILUMINAÇÃO

Como planta sob estudo, foi utilizada a espécie *Lactuca sativa* (alface), conhecida por sua sensibilidade à umidade do solo e necessidade de irrigação constante. As plantas foram envasadas em recipientes semelhantes aos usados na regulação e posicionadas sob o campo de visão do Módulo de Controle, conforme Fig.34.



Figura 34 – Planta sob estudo (alface estiolado). Fonte: o autor.

Apesar de dispormos de controle automático da iluminação da tenda por meio do temporizador programável (modelo TE-30, Elcon), decidimos pelo ciclo contínuo de 24 horas de luz ativada com auxílio de lâmpada LED branca (18 W, 6500 K). Essa configuração é similar à utilizada por Wang et al. (2024) em seus testes com coentro. Esse setup foi mantido durante toda a fase de experimento a fim gerar mais imagens para o dataset em menor período.

## 5.7 A PLATAFORMA DE DESENVOLVIMENTO E O FIRMWARE

Todo o firmware foi desenvolvido com base na linguagem C++, utilizando a plataforma PlatformIO integrada à IDE Visual Studio Code, Fig. 35.

O PlatformIO é um ecossistema de desenvolvimento open-source para sistemas embarcados, como placas Arduino, ESP32, STM32, entre outras. Essa integração facilita bastante o desenvolvimento em comparação ao Arduino IDE tradicional, especialmente para projetos maiores e mais complexos. A estrutura de programação adota o paradigma orientado a objetos, o que favorece a modularização das principais funcionalidades.

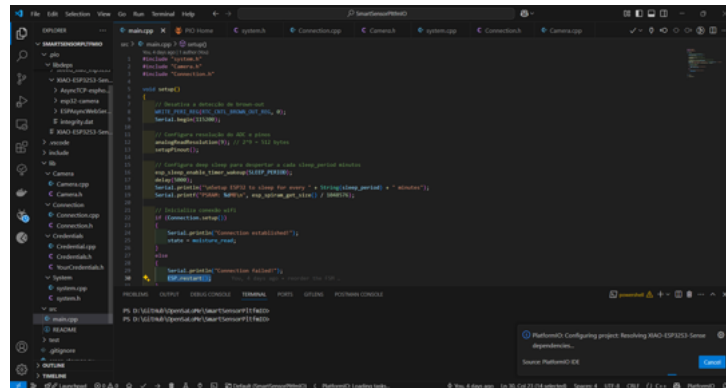


Figura 35 – VSCode com PlatformIO. Fonte: o autor.

## 5.8 A PLATAFORMA DE PROCESSAMENTO DE IMAGENS

Para processamento das imagens geradas pelo sistema foi utilizado o *Google Colaboratory*, Fig.36, uma plataforma gratuita baseada em nuvem que permite o desenvolvimento e a execução de notebooks Jupyter diretamente no navegador, eliminando a necessidade de configurações locais complexas.

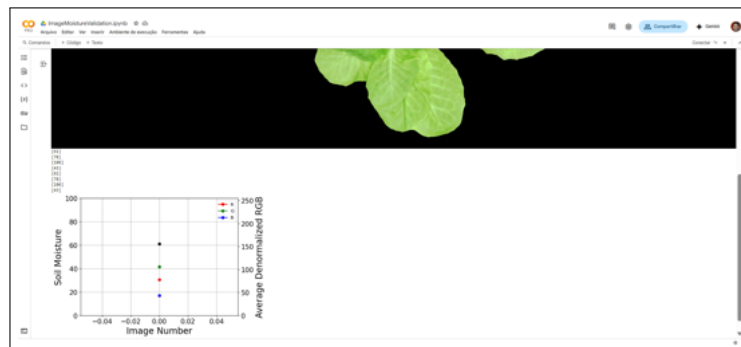


Figura 36 – Google Colab. Fonte: o autor.

No contexto de **processamento e análise de imagens**, o Colab oferece uma série de vantagens relevantes:

- **Acesso a aceleradores de hardware**, como GPUs e TPUs, o que o torna uma ferramenta valiosa para o treinamento de modelos DL com bibliotecas como TensorFlow, PyTorch e OpenCV;
- **Ambiente pré-configurado**, com diversas bibliotecas amplamente utilizadas em visão computacional já instaladas;
- **Integração nativa com o Google Drive**, permitindo o armazenamento, organização e compartilhamento eficientes de imagens, conjuntos de dados e modelos;

- **Execução em células de código**, facilitando a experimentação e a depuração de forma interativa e incremental;
- **Suporte a visualizações embutidas** no próprio notebook, por meio de ferramentas como *matplotlib*, *seaborn* e OpenCV, o que favorece a análise e interpretação dos resultados obtidos.

Na etapa do processamento de imagens, especialmente na definição das máscaras para segmentação de cor, utilizou-se o GIMP como ferramenta de apoio. Trata-se de um software gratuito e de código aberto amplamente utilizado para edição e manipulação de imagens.

## 5.9 ARMAZENAMENTO DOS DADOS EM SERVIDOR EM NUVEM

A recepção e armazenamento tanto dos dados de umidade quanto das imagens são gerenciados no Google Apps Script, que é uma plataforma em nuvem baseada em JavaScript desenvolvida pelo Google. Ela permite automatizar tarefas, integrar e estender a funcionalidade de produtos Google como Google Sheets, Google Docs, Google Forms, Google Drive, etc.

### 5.9.1 Comunicação entre SPheRe e Google Sheets

Desenvolvemos um script, Fig.37, que permite ao SPheRe enviar dados de umidade para uma planilha online, Fig. 38, e solicitar parâmetros de controle (como limiares de irrigação e comando de rega baseado em checkbox) oriundos da planilha. Ou seja, o fluxo de dados diz respeito ao valor da umidade, atualização de limiares de histerese da umidade e comando de rega forçada.

### 5.9.2 Comunicação entre SPheRe e Drive

Em síntese, desenvolvido esse segundo script que permite que o SPheRe envie imagens para serem automaticamente armazenadas em uma pasta específica do Google Drive, Fig.39, com um nome de arquivo que inclui um dado contextual (umidade) e um carimbo de data/hora.

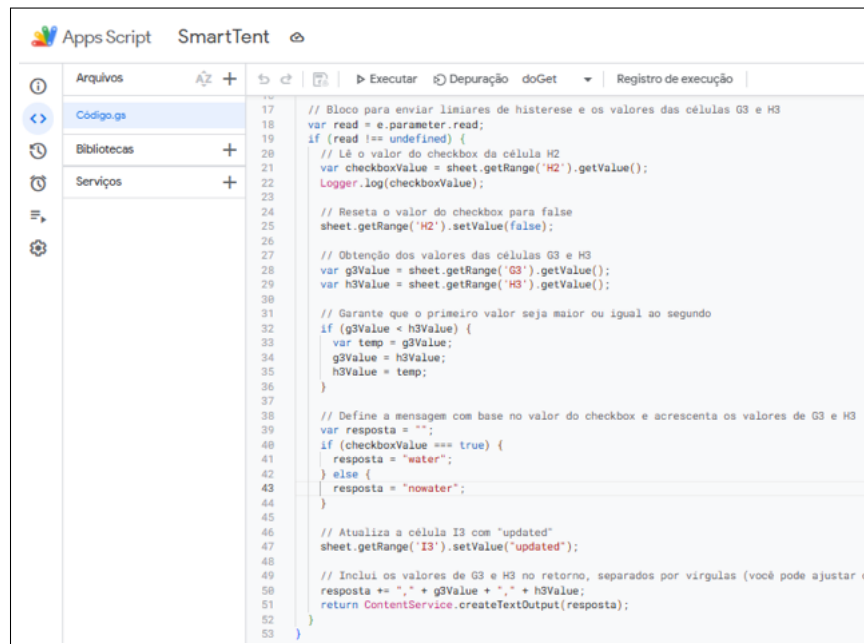


Figura 37 – Ambiente de desenvolvimento em nuvem do Google Apps Script. Fonte: o autor.

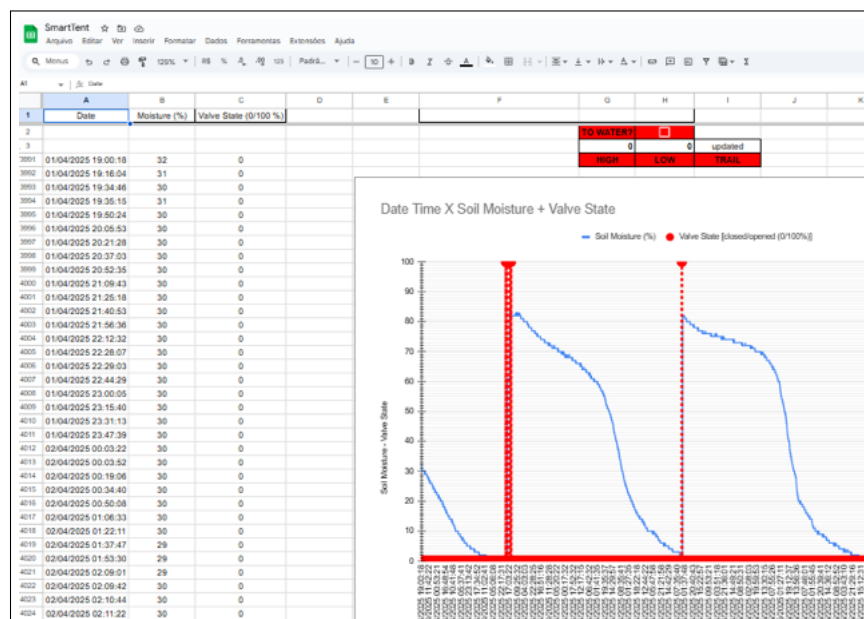


Figura 38 – Planilha exibindo dados, gráfico e interface de controle do SPheRe. Fonte: o autor.

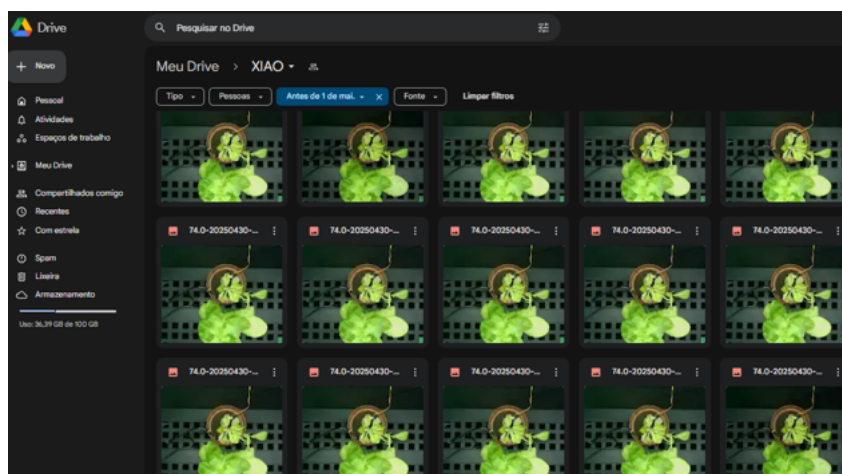


Figura 39 – Dataset de imagens em nuvem - Google Drive. Fonte: o autor.

## 5.10 PROCESSAMENTO E ANÁLISE DAS IMAGENS RGB

Para investigar a relação entre a coloração das folhas e a umidade do solo, foi desenvolvido um conjunto de scripts em Python na plataforma de desenvolvimento Google Colab, em nuvem, para análise das imagens armazenadas em dataset no Google Drive. O processamento envolveu as seguintes etapas:

### 5.10.1 Segmentação das Folhas

A função `cut_leaves_without_spark()` foi projetada para realizar o recorte das folhas presentes em uma imagem de entrada, ao mesmo tempo em que busca mitigar variações luminosas indesejadas. O processo é estruturado em etapas sequenciais, conforme detalhado a seguir.

#### 5.10.1.1 Remoção de Interferência Luminosa

A metodologia inicia-se com a definição de uma **região de referência** dentro da imagem de entrada — localizada na prateleira e caracterizada pela cor preta (Fig. 40). Nessa região, são calculadas as médias dos valores dos componentes de cor (RGB). Em seguida, tais médias são subtraídas de todos os pixels da imagem, com o objetivo de atenuar ou eliminar possíveis pulsos luminosos espúrios.

Dessa forma, a região de referência atua como um padrão de calibração, garantindo que os valores de cor analisados não sejam interpretados de forma absoluta,



mas relativa ao ponto de referência. Essa abordagem melhora a robustez da análise diante de variações indesejadas de iluminação.

É importante destacar que, diferentemente de softwares de edição de imagem convencionais — e mesmo da convenção usual do sistema HSV —, o Python utiliza faixas específicas de valores para cada canal de cor. Assim, após a normalização baseada na região de referência, os novos valores atribuídos aos pixels não correspondem necessariamente às suas cores reais, mas a valores corrigidos para fins comparativos dentro do processamento.



Figura 40 – Imagem com região de referência em destaque. Fonte: o autor.

#### 5.10.1.2 Suavização da Imagem

Para preparar a imagem para as etapas subsequentes de segmentação e minimizar o ruído, um **filtro Gaussiano de desfoque** (`cv2.GaussianBlur`) é aplicado. Este filtro contribui para a homogeneização das regiões.

#### 5.10.1.3 Segmentação Baseada em Cor (Verde)

A imagem é convertida do espaço de cor RGB para o espaço HSV, já abordado na Seção D.0.1.2. Esta faixa de valores para a matiz verde foi obtida de forma observacional, com base em experimentos em diversas imagens, demonstrando excelente robustez em variados testes. Uma **máscara binária** é então gerada, delimitando uma faixa específica de tonalidades de verde. Essa máscara é utilizada para isolar e reter



apenas as regiões da imagem que correspondem à coloração das folhas, eliminando as demais cores presentes no fundo. A segmentação de cor relativa a regiões de folha seca também foi utilizada.

#### 5.10.1.4 Segmentação por Limiarização

A porção da imagem resultante da etapa de segmentação por cor, contendo predominantemente as áreas verdes, é convertida para **tons de cinza**. Após uma nova aplicação de desfoque, é empregado o método de **limiarização de Otsu** (`cv2.threshold`). Este algoritmo automático determina um valor de limiar ideal para binarizar a imagem, distinguindo efetivamente as folhas do fundo. Posteriormente, um **filtro de mediana** (`cv2.medianBlur`) é aplicado à máscara binária obtida, contribuindo para a suavização de bordas e a remoção de pequenos ruídos.

#### 5.10.1.5 Recorte Final das Folhas

Na etapa final, a **máscara binária refinada** (gerada pelas etapas de segmentação) é aplicada à imagem oriunda da etapa 5.10.1.1 (já sem interferência luminosa). O resultado é uma nova imagem que contém exclusivamente as folhas segmentadas, com o fundo e quaisquer outros elementos não pertinentes removidos, ou seja, preenchidos com pixels nulos, facilitando análises posteriores focadas na folhagem.

Esse processo garante que os pixels considerados na média RGB pertençam predominantemente à área foliar da planta monitorada.

É importante ressaltar que a segmentação apresentou um artefato indesejado, uma pequena parte do módulo de monitoramento ambiental afastado da planta, apesar de aparentemente insignificante, utilizamos artifício de anulação de pixels apenas nessa região.

A Fig. 41 ilustra estágios envolvidos no recorte de folhas de cada imagem.

### 5.10.2 Extração de Dados Estatísticos

A função `image_statistics` processa cada imagem de folha recortada resultante da função `cut_leaves_without_spark()` a fim de extrair os dados de cor “relativa”. Va-



Figura 41 – Pipeline de Processamento de Imagens Digitais. Fonte: o autor.

riáveis do tipo `cropped_leaves_mean_blue_list` e `cropped_leaves_sd_blue_list`, correspondem a valores médios e de desvio padrão do canal de cor azul “relativo” de cada imagem, formando séries temporais.

### 5.11 ANÁLISE DE ESPAÇOS DE COR EM FUNÇÃO DA UMIDADE

Para aprofundar a análise entre cor e umidade, cada imagem do dataset foi segmentada utilizando a função implementada `cut_leaves_without_spark()`, que realiza a remoção de variações indesejadas de luminosidade, aplica suavização e segmentação por cor e limiarização. Como resultado, a função retorna a imagem segmentada, no espaço de cor RGB, que em seguida é convertida em HSV. A partir dessas imagens, foram extraídas estatísticas como média e desvio padrão de cada canal (R, G, B, e o H, S e V), medidas relativas à região de referência, juntamente com o valor de umidade.

### 5.12 CLASSIFICAÇÃO POR MEIO DE ARQUITETURAS RASAS

Para a etapa de classificação, foram adotados três algoritmos de aprendizado supervisionado: *Decision Tree*, *Random Forest* e *KNN* (*K-Nearest Neighbors*).

O *Decision Tree* é um método baseado em uma estrutura hierárquica em forma de árvore, semelhante a um fluxograma, no qual as decisões são tomadas por meio de regras extraídas dos atributos das amostras. O *Random Forest*, por sua vez, consiste em um conjunto de múltiplas árvores de decisão combinadas, o que permite aumentar a precisão da classificação e reduzir a ocorrência de erros decorrentes do overfitting.

Já o *KNN* é um algoritmo baseado em medidas de similaridade. Para classificar uma nova amostra, calcula-se a distância — geralmente euclidiana — entre ela e todas as amostras do conjunto de treinamento, selecionando-se os *K* vizinhos mais próximos. A classe atribuída é a mais frequente entre esses vizinhos. Embora simples e de fácil implementação, o KNN apresenta como desvantagens o maior custo computacional durante a inferência e a necessidade de manter todo o conjunto de dados armazenado em memória.

### 5.13 VISÃO GERAL DOS EXPERIMENTOS

Todos os experimentos apresentados neste trabalho referem-se a análises realizadas sobre um conjunto de arquivos de imagens RGB, capturadas pelo SPheRe, formando um dataset com mais de três mil imagens coletadas ao longo de um período de 42 dias.

As análises foram organizadas em quatro etapas principais: primeiramente, uma análise dos dados extraídos das imagens em conjunto com a percepção visual de amostras do dataset. Na sequência, foi conduzida uma análise baseada em ML, utilizando **arquiteturas rasas**. E em seguida, uma análise por meio de arquiteturas **profundas**. Por fim, a quarta e última etapa, a avaliação da execução de modelos obtidos até então diretamente no SoC.

Na primeira etapa do experimento, de caráter estatístico e observacional, foram empregadas ferramentas de PDI — conforme detalhado na Seção 2.2 e no Anexo — incluindo: estatísticas de cor, indicadores de movimento e caracterização de textura. O fluxo desse processo é ilustrado na Fig. 42.

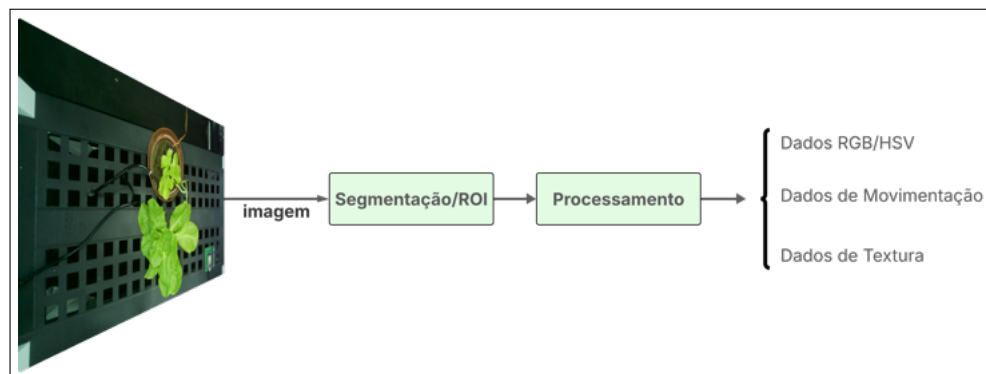


Figura 42 – Pipeline para extração de dados de cor, textura e tropismo. Fonte: o autor.

A segunda etapa corresponde a uma **classificação rasa**, onde as imagens passam por pré-processamento, incluindo a segmentação da região de interesse (ROI) e a extração de texturas, resultando em vetores de atributos utilizados como entrada para algoritmos de aprendizado supervisionado. O fluxo metodológico encontra-se ilustrado na Fig. 43.

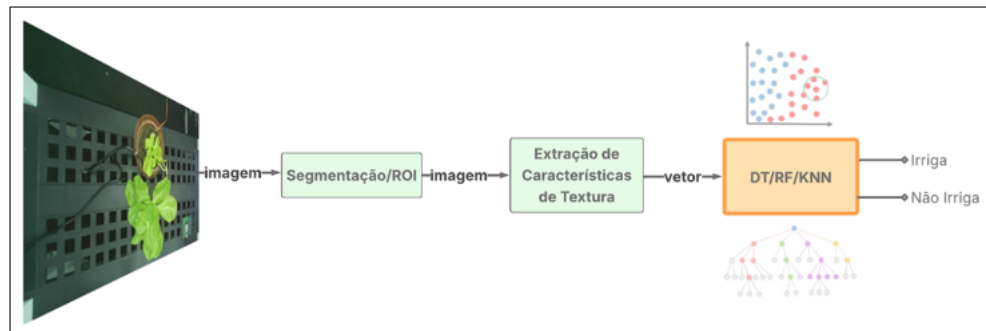


Figura 43 – Pipeline para classificação indireta via arquiteturas rasas e por meio de dados de textura. Fonte: o autor.

A terceira etapa, a aplicação de **classificação profunda**, onde as imagens são processadas diretamente por uma rede neural convolucional do tipo MobileNetV3. Nesse caso, a rede aprende automaticamente os descritores relevantes para diferenciar plantas com e sem estresse hídrico, dispensando a etapa de extração de atributos. O fluxo dessa classificação é apresentado na Figura 44.

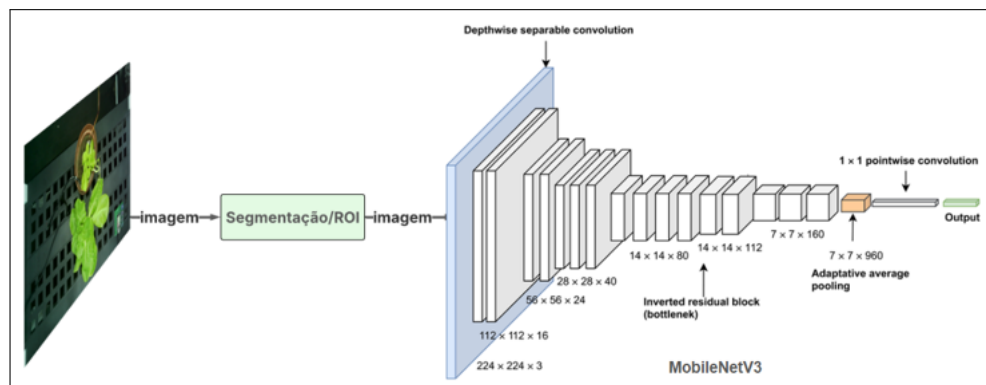


Figura 44 – Pipeline para classificação direta de imagem via arquitetura profunda. Fonte: o autor.

Por fim, a última etapa consistiu no experimento de implementação de PDI e do emprego dos modelos avaliados — outrora, em nuvem — diretamente no SoC ESP32S3. A Fig 45, ilustra esse cenário desafiador, cujo foco está no potencial dos modelos obtidos fornecerem resultados em cenários práticos, contornando as limitações do dispositivo, seja por meio de redução de modelos, quanto na refatoração de código para processar imagens.

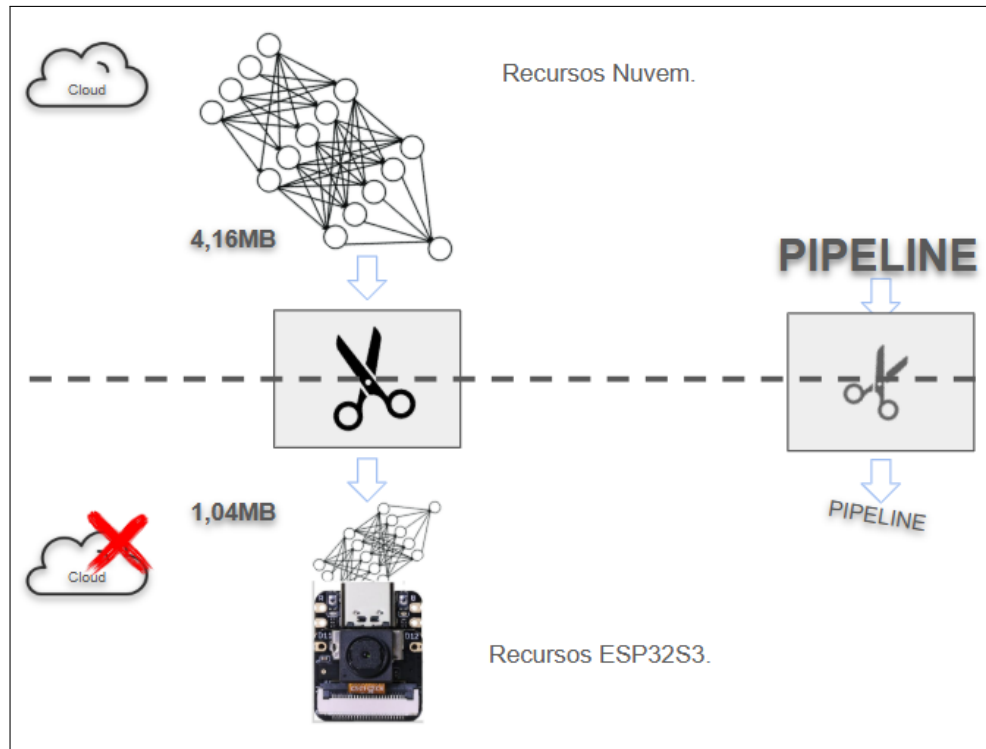


Figura 45 – Cenário Nuvem versus Edge-Computing. Fonte: o autor.

#### 5.14 PREPARAÇÃO DOS DADOS

A partir das imagens armazenadas em uma pasta no Google Drive, nomeadas pelo padrão umidade-AAAAMDD-HHMMSS.jpg, foi construído o dataset, cronologicamente ordenado, com todas as imagens capturadas entre 1 de abril de 2025 e 12 de maio de 2025, totalizando 3966 imagens. O gráfico da Fig. 46 representa os valores de umidade do solo em função do tempo, compreendendo ciclos de estresse hídrico, com regas realizadas a cada 15 dias, aproximadamente.

A Fig. 47 representa uma amostra do dataset nesse período.

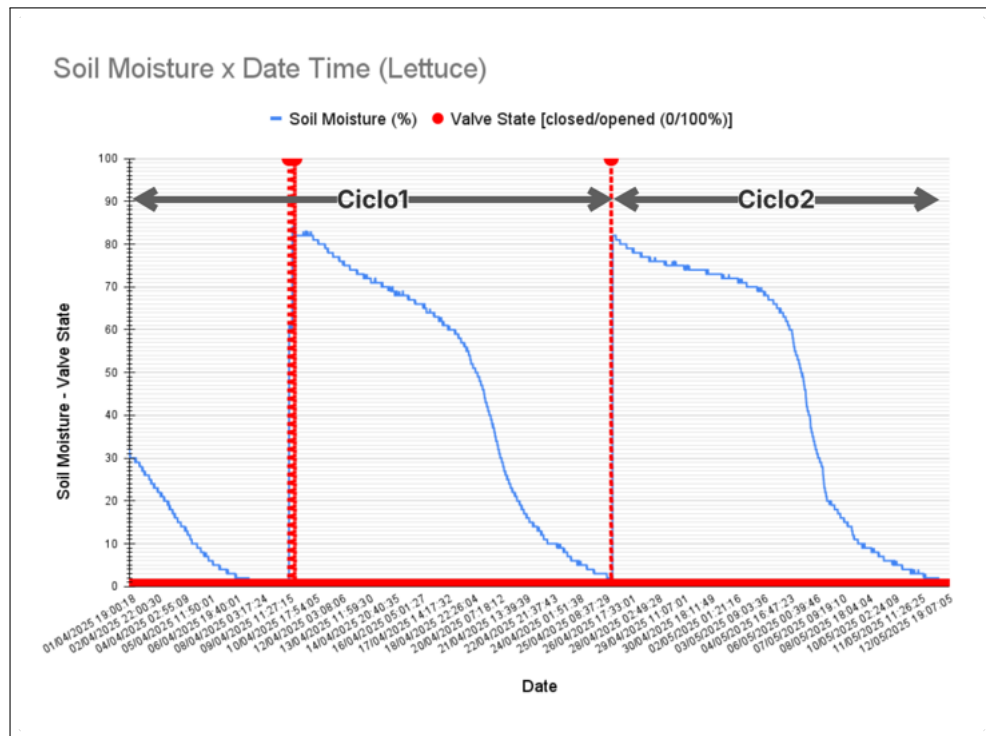


Figura 46 – Gráfico da umidade do solo e do comando de irrigação, em função do tempo. A segunda rega separa o experimento em dois ciclos. Fonte: o autor.

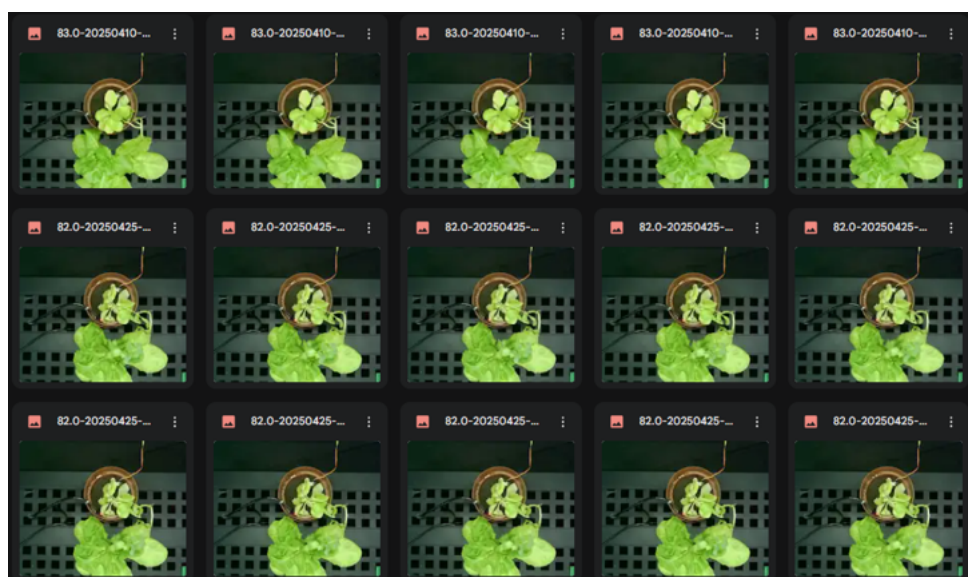


Figura 47 – Amostra do dataset do experimento. Fonte: o autor.

## 5.15 DESAFIOS DE OTIMIZAÇÃO DO SETUP

### 5.15.1 Sensor de umidade

Para assegurar o desempenho adequado do sensor capacitivo adotado, foi necessária uma intervenção corretiva decorrente de falha de fabricação: a conexão de pino de resistor ao terminal de terra do circuito. Adicionalmente, realizou-se a impermeabilização da borda da PCI com verniz Isotec Incolor (Implastec), de modo a reduzir o risco de infiltrações. Esses ajustes estão evidenciados na Figura 48.

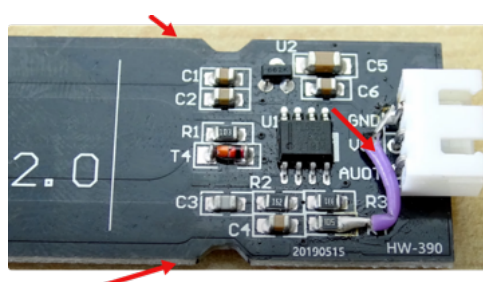


Figura 48 – Sensor ajustado: conexão do terra e impermeabilização das bordas. Fonte: o autor.

### 5.15.2 Iluminação artificial

Durante as primeiras etapas, observou-se significativa interferência nas imagens capturadas, além da má formação de diversas mudas de alface devido ao estiolamento. Esse desafio foi superado mediante a substituição da iluminação por lâmpadas LED mais adequadas e pela adoção de um ciclo contínuo de 24 horas de iluminação.

### 5.15.3 Reservatório de água

O primeiro reservatório adotado apresentou intensa formação de lodo, comprometendo a qualidade da água utilizada na irrigação. O problema foi solucionado com a substituição do recipiente por um modelo em PVC, que apresentou maior resistência a esse tipo de ocorrência.

#### **5.15.4 Pintura do suporte**

A superfície da prateleira de apoio às mudas foi pintada na cor preta com o intuito de facilitar os processos de segmentação das imagens, reduzindo interferências visuais no pré-processamento.

#### **5.15.5 Configuração da câmera**

A configuração da câmera OV5640 constituiu um dos maiores desafios do estudo, devido à escassa documentação técnica disponível. A calibração foi realizada a partir de uma combinação de pesquisa em diferentes fontes e de extensivos testes empíricos baseados em tentativa e erro.

#### **5.15.6 Bomba de irrigação**

Após longos períodos sem acionamento do sistema, foram identificadas falhas no funcionamento da bomba de irrigação. A solução encontrada restringiu-se a ajustes manuais e leves impactos físicos no componente, o que evidencia uma limitação do modelo adotado, mesmo após a substituição por bombas de diferentes fabricantes.

### **5.16 CONSIDERAÇÕES METODOLÓGICAS FINAIS**

A câmera OV5640 foi configurada para capturar imagens em resolução de  $2560 \times 1600$  pixels com fator de compressão JPEG igual a 10, oferecendo qualidade satisfatória com economia de espaço na nuvem. Inclusive, por razão desconhecida, testes de transmissão de imagens com resolução superior não foram bem sucedidos. As imagens são capturadas e enviadas a cada 15 min, e associadas aos dados de umidade do solo, para posteriormente serem analisadas com o intuito de verificar a relação entre a imagem e o nível de umidade, objetivo central da fenotipagem conduzida neste estudo.

Concluídas as etapas de regulação e definidas as condições experimentais do cultivo, procedeu-se ao desenvolvimento e à implementação do sistema embarcado, descritos no capítulo seguinte.



## 6 RESULTADOS: IMPLEMENTAÇÃO

Este capítulo apresenta a implementação do sistema embarcado de fenotipagem proposto, SPheRe, detalhando sua arquitetura de software, lógica de funcionamento, desenvolvimento do firmware e hardware. Com base nos requisitos definidos e no ambiente experimental descrito no capítulo anterior, o sistema foi desenvolvido com foco em modularidade, robustez e baixo consumo energético, utilizando tecnologias acessíveis e amplamente documentadas. O desenvolvimento bem-sucedido do SPheRe representa um resultado importante da presente dissertação.

### 6.1 ESTRUTURAMENTO DO FIRMWARE

A arquitetura do código foi baseada na seguinte tríade da Fig.49:



Figura 49 – Tríade do *SPheRe*. Fonte: o autor.

Podemos traduzir essa tríade nos passos a seguir, os quais compõem o núcleo do SPheRe:

- Configuração e leitura da umidade através do sensor capacitivo de umidade do solo;
- Configuração do módulo câmera OV5640 e captura de imagens;
- Configuração da conectividade Wi-Fi e comunicação com a nuvem.

O programa principal realiza a inicialização de registradores e periféricos, configurando o conversor analógico-digital, definindo os pinos de entrada e saída, e ativando

a comunicação serial para depuração. Após essa etapa, a comunicação Wi-Fi é estabelecida, e o sistema entra no ciclo principal, baseado em Máquina de Estados Finitos (do inglês FSM).

## 6.2 LÓGICA DE FUNCIONAMENTO DO SISTEMA

O comportamento do SoC é regido por meio de uma FSM composta por seis estados, conforme ilustrado na Fig.50. Esses estados regem a coleta de dados, a comunicação com a nuvem, o acionamento da irrigação e o controle energético.

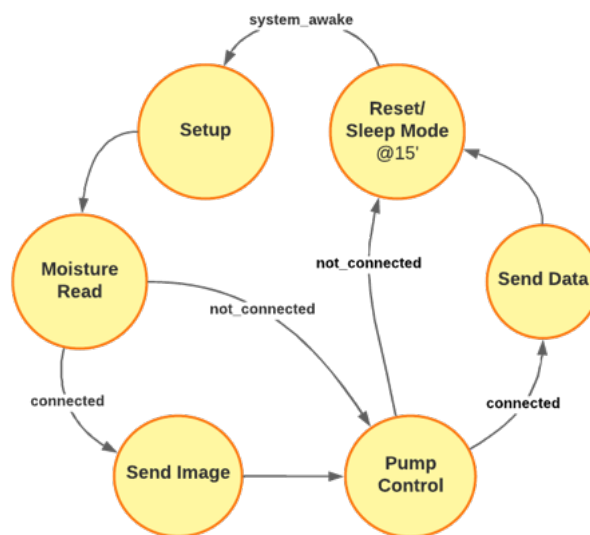


Figura 50 – Diagrama de estados do *SPheRe*. Fonte: o autor.

1. **Setup**: estado de inicialização do microcontrolador, responsável por configurar pinos, sensores, conexão Wi-Fi e periféricos.
2. **Leitura da Umidade** (*moisture\_read*): amostragem de 258 valores analógicos do sensor capacitivo, com descarte de outliers (mínimo e máximo), seguida do cálculo da média. O valor médio é mapeado para uma escala percentual entre 0% e 100%.
3. **Captura e Envio de Imagem** (*send\_image*): após verificar se há iluminação suficiente, uma imagem com resolução de  $2560 \times 1600$  pixels é capturada e associada ao valor de umidade do solo. Em seguida, é enviada via requisição HTTP

para um script no Google Drive, onde é rotulada com as informações de umidade, data e horário de recebimento, e por fim, salva em pasta.

4. **Controle da Rega** (`pump_control`): A rega pode ser acionada mediante duas circunstâncias excludentes: com maior prioridade, o sistema consulta *checkbox* da planilha no Google Sheets via HTTP. Caso o *checkbox* esteja habilitado, a bomba d'água é acionada por 1 min e, em seguida, o comando é desabilitado automaticamente. Caso não haja esse comando de acionamento por parte do usuário, o acionamento da irrigação pode se dar por meio da comparação do valor de umidade atual, lido em `moisture_read`, com os limiares (da histerese) salvos em memória anteriormente - durante o estado `send_data` do ciclo anterior. Em caso de acionamento, a bomba é ativada por 5 s. A Fig. 51 ilustra parte da planilha responsável pela transmissão de dados a partir da nuvem para o *SPheRe*.

TO WATER?	<input type="checkbox"/>
0	0
HIGH	LOW

Figura 51 – Pequeno painel de interação com o usuário, para rega forçada e definição de limiares. Fonte: o autor.

5. **Envio de Dados** (`send_data`): os dados de umidade e estado da bomba são enviados via HTTP para o Google Sheets, atualizando a planilha de monitoramento. Em seguida os valores de limiar da histerese são lidos da planilha e atualizados em memória não volátil.
6. **Modo de Suspensão** (`deep_sleep`): após o ciclo completo, o sistema entra em modo de hibernação por 15 min, reduzindo o consumo energético.

É importante frisar que na ocasião de falha de conexão com o servidor a FSM garantirá a irrigação da planta, por meio de sua passagem pelos estados `moisture_data` e `pump_control`, com base nos valores de limiares previamente salvos, seja de “fábrica” ou oriundos da última comunicação com o servidor.

Outro recurso disponível no código é a verificação de falhas, tanto de comunicação quanto do módulo de imagem, com a capacidade de reinicialização do sistema.

A seguir, um melhor detalhamento de algumas das funcionalidades do *SPheRe*.

### 6.3 CONFIGURAÇÃO DA CAPTURA DE IMAGEM E COMPRESSÃO

O módulo de imagem, composto pela câmera OV5640, é primeiramente configurado com uma baixa resolução de  $96 \times 96$  pixels em escala de cinza. Uma imagem inicial é capturada para verificar as condições de iluminação do ambiente. Se o ambiente estiver iluminado — condição que só ocorre com a ativação da iluminação artificial —, a câmera é reconfigurada para a captura da imagem final. Esta é obtida com uma resolução superior ( $2560 \times 1600$  pixels), no espaço de cor RGB, e um fator de compactação JPEG de 10. Tal fator, situado entre 0 (maior qualidade) e 63 (menor qualidade), foi escolhido para assegurar uma qualidade visual adequada sem sobrecarregar o armazenamento em nuvem.

### 6.4 GERENCIAMENTO DE ENERGIA

Visando prevenir o sobreaquecimento e otimizar o consumo energético, o SPheRe é configurado no modo de hibernação, `deep_sleep`, ao final de cada ciclo da FSM. Esse modo desliga os periféricos não essenciais e reduz drasticamente o consumo de corrente, permitindo que o sistema opere de forma prolongada mesmo em condições de alimentação limitada, a exemplo de baterias.

Adicionalmente, o módulo câmera, responsável pela maior parte do consumo com relação a todo o SPheRe, é apenas ativado durante o estado `send_image`, promovendo ainda maior economia de energia e proteção térmica. Ou seja, na eventualidade de não haver comunicação com o servidor, esse módulo permanece desativado.

### 6.5 HARDWARE PARA ACIONAMENTO DA MINI BOMBA D'ÁGUA

Devido à limitação de corrente elétrica da placa de desenvolvimento, XIAO ESP32S3 Sense, projetamos uma placa auxiliar contendo um circuito de acionamento baseado em transistor bipolar e resistor, conforme exhibe a Fig. 52. Esse circuito permite o controle da bomba d'água, operando com a corrente adequada.

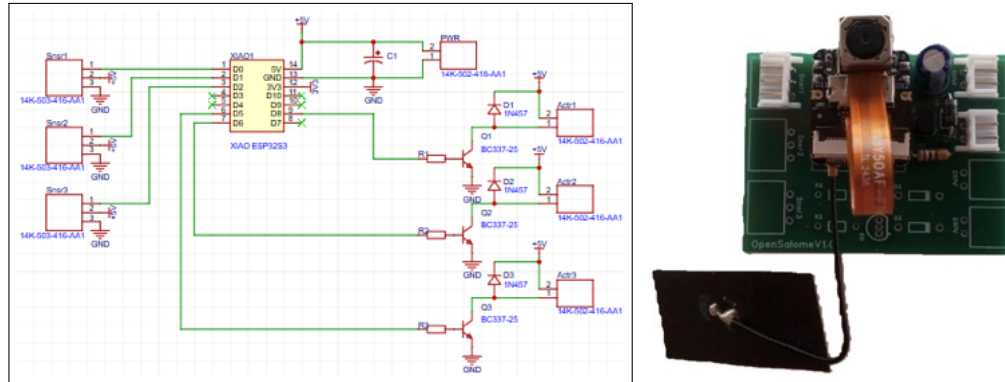


Figura 52 – Esquema elétrico e PCI da placa auxiliar montada. Fonte: o autor.

## 6.6 MÓDULO PARA COLETA DE DADOS AMBIENTAIS

O módulo originário do trabalho de Cardoso e Blawid (2022) foi utilizado nesta pesquisa para coleta e monitoramento de dados como: temperatura e umidade do ar e intensidade luminosa, Fig. 53, dentre outros fatores. A Fig. 54(a) evidencia este módulo equipado com ESP32 e demais módulos sensores, enquanto, a Fig. 54(b) ilustra seu uso na tenda de crescimento.

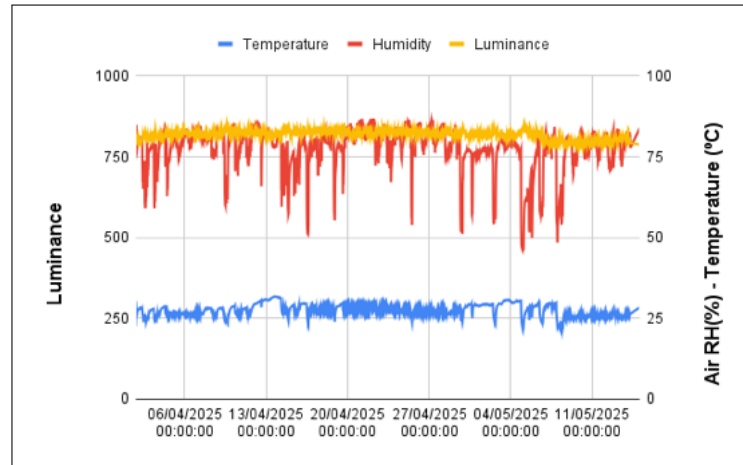


Figura 53 – Alguns dos fatores monitorados no interior do SPheRe. Fonte: o autor.

O firmware disponível para uso nesse módulo necessitou de pequena adequação para uso no contexto do SPheRe.

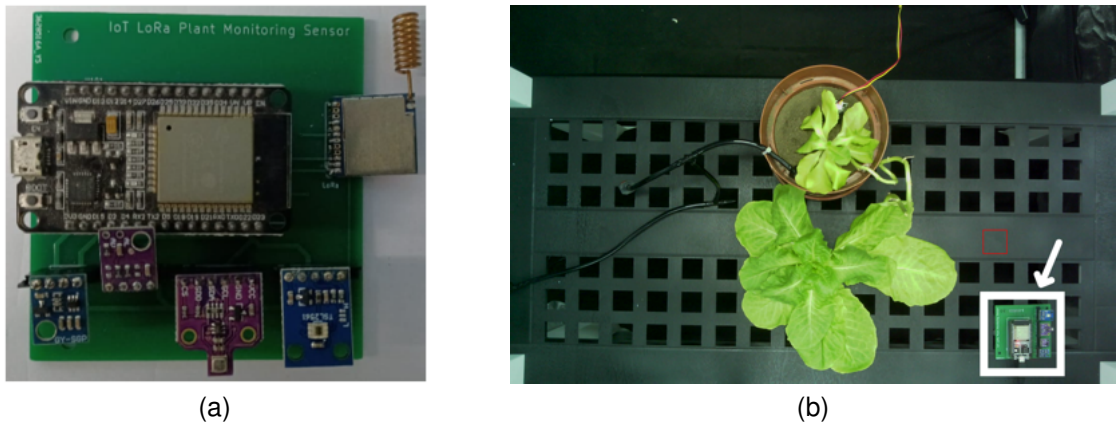


Figura 54 – (a) Módulo para medições de dados ambientais na tenda. Fonte: Cardoso e Blawid (2022); (b) Posicionamento do sensor ambiental. Fonte: o autor.

## 6.7 REPOSITÓRIO DO PROJETO

O repositório Github do projeto pode ser encontrado aqui.

## 7 RESULTADOS: PROVAS DE CONCEITO

Neste capítulo, apresentamos os resultados obtidos na fenotipagem automática de plantas submetidas a diferentes níveis de umidade do solo.

### 7.1 ANÁLISE DE ESPAÇOS DE COR EM FUNÇÃO DA UMIDADE

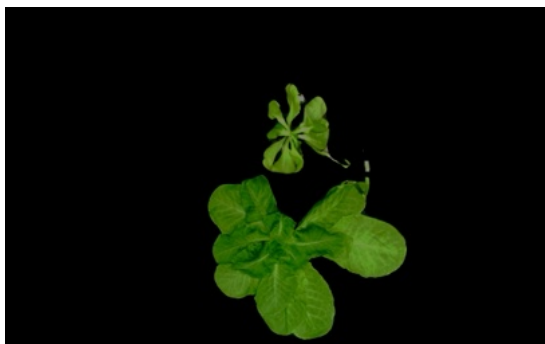
A Fig. 55 ilustra dois exemplos das imagens em RGB segmentadas, e destaca poucas mudanças perceptíveis na planta após a irrigação.



(a) Imagem original. Capturada imediatamente antes da rega.



(b) Imagem original. Capturada dez horas após a rega. Movimento das folhas em destaque.



(c) Imagem (a) segmentada.



(d) Imagem (b) segmentada.

Figura 55 – Imagens capturadas e segmentadas antes e depois da irrigação. (Esq.) solo a 0% de umidade relativa. (Dir.) a 82%.

Ao observar ambas as imagens originais na Fig.55, antes e após a irrigação, não se nota alteração perceptível na cor, exceto por pequenas movimentações em algumas partes das folhas.

### 7.1.1 Médias dos Canais

A Fig. 56 mostra a variação da média dos canais RGB das imagens segmentadas (eixo direito) em função da umidade do solo (eixo esquerdo) ao longo do tempo (número da imagem no eixo x).

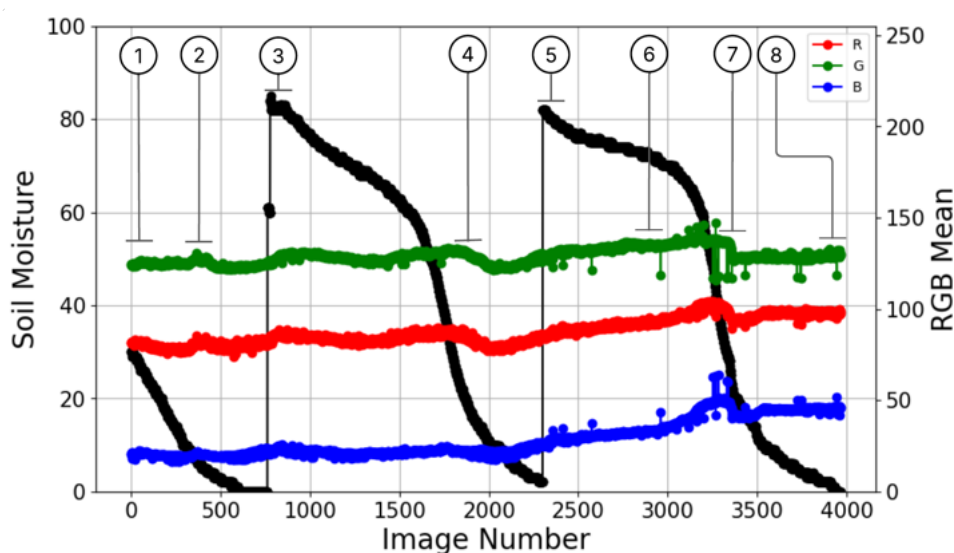


Figura 56 – Valores RGB médios durante dois ciclos de irrigação. Média R, G e B de cada uma das 3966 imagens capturadas a cada 15 minutos durante 42 dias. Com destaque de 8 instantes relevantes durante o experimento. Fonte: o autor.

Para a análise dos valores médios dos canais de cor R, G e B das folhas ao longo do tempo, destacamos períodos específicos no gráfico da Fig. 56, enumerados em sequência. Durante todo o período de testes, o gráfico sugere que imagens sofreram alteração de cor e de intensidade luminosa. Principalmente, em torno dos instantes **3**, **4** e **5** percebe-se alterações da taxa de variação do canal verde (G), o que sugere alguma relação com a umidade. Porém, outros trechos após o evento **5** apresentam certas anomalias, como outliers, ou mudanças bruscas.

A análise com foco apenas no gráfico se mostra desafiadora, portanto, outra abordagem como a observação direta de amostras foi aplicada. As figuras a seguir (57 e 58) trazem algumas destas amostras de cada um dos trechos apontados no gráfico.



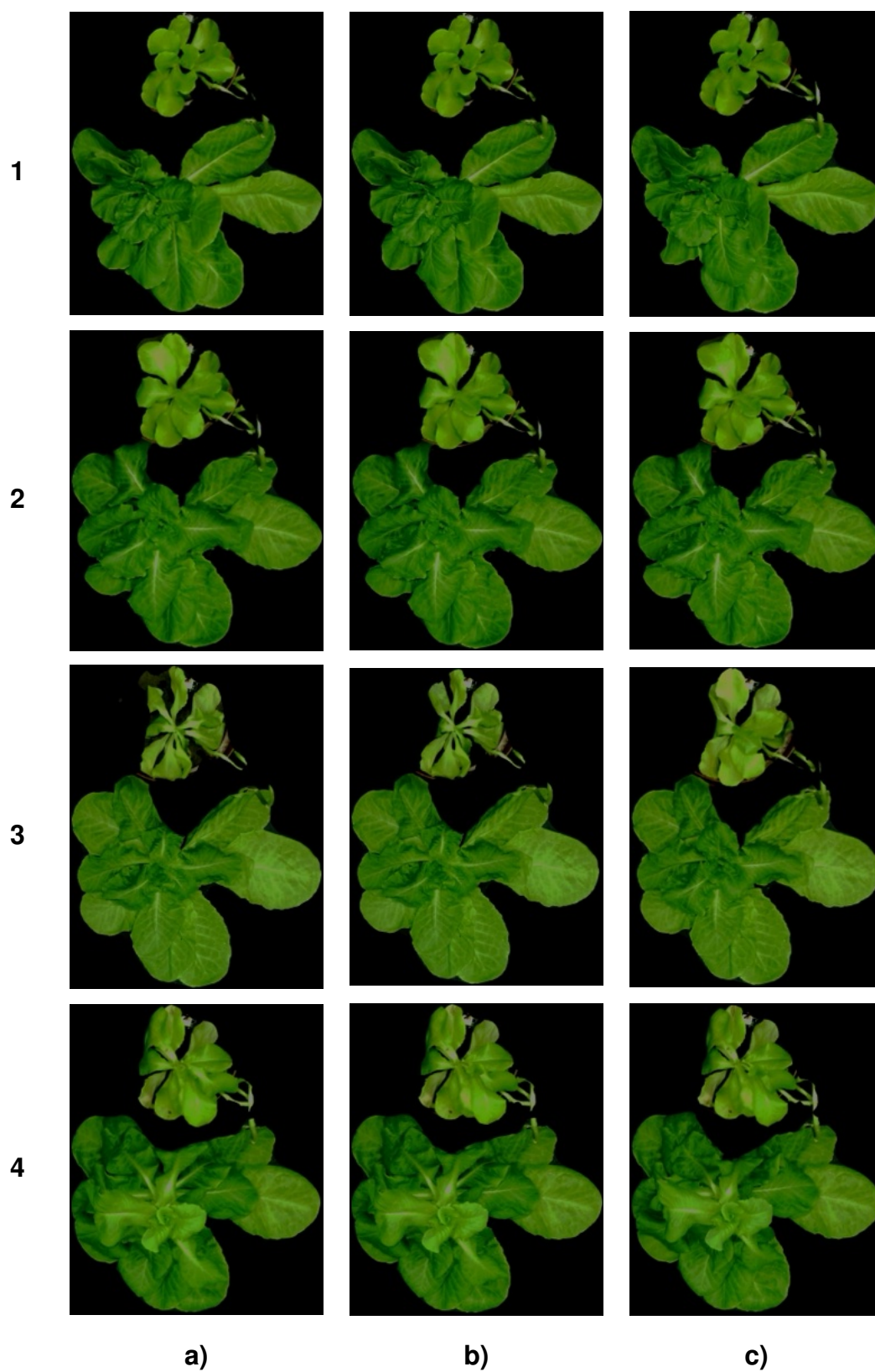


Figura 57 – Amostras de imagens processadas durante os eventos 1,2,3 e 4: **a)** Início do evento, **b)** 2 horas após e **c)** 10 horas após **a)**. Cada linha representa o mesmo "trecho".

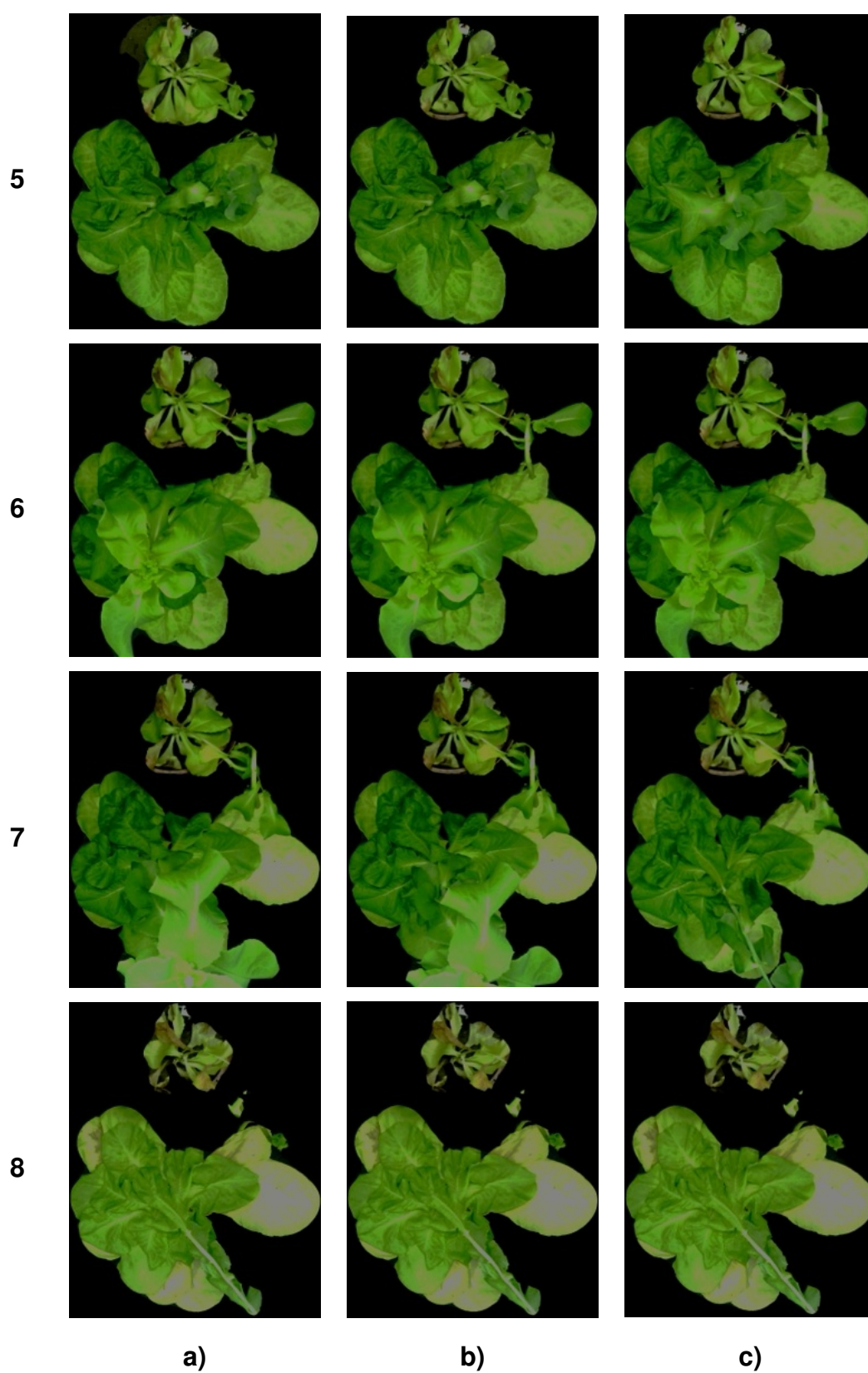


Figura 58 – Da mesma forma, amostras de imagens processadas durante os intervalos 5,6,7 e 8: a) Início do evento, b) 2 horas após, e c) 10 horas após a).

Com o objetivo de reunir as informações referentes a cada trecho analisado, apresenta-se o Quadro 3, a qual deve ser consultado em conjunto com a Fig. 56 e com as amostras ilustradas nas Figuras 57 e 58.

<b>Trecho</b>	<b>Descrição</b>
<b>1</b>	Amostras de imagens processadas nas primeiras 10 horas do experimento.
<b>2</b>	Gráfico sugere alteração de cor, mas mudança não perceptível nas amostras.
<b>3</b>	Imagens antes e após a primeira rega. Gráfico indica aumento dos índices antes da rega, sugerindo provável clareamento das folhas. Algum tempo após a rega os índices param de aumentar e seguem com leve inflexão dos mesmos índices.
<b>4</b>	Amostras evidenciam o início de estiolamento. Índices R e G decaem com maior evidência se comparados com B.
<b>5</b>	Imagens antes e após a segunda rega. Antes da rega, os índices R, G e B se elevam igualmente, provável clareamento das imagens subsequentes. Novamente, após a rega, há uma leve mudança da taxa de crescimento destes índices. Imagens com estiolamento mais evidente. Início de secamento de algumas folhas.
<b>6</b>	Ao contrário do que ocorre entre <b>3</b> e <b>4</b> , Índices não estacionam, continuam se elevando. O que traduz o início de imagens com regiões de brilho intenso.
<b>7</b>	Observa-se uma variação acentuada nas médias dos canais R, G e B, acompanhada por indícios visuais de uma movimentação abrupta do estiolamento. Além disso, nota-se um discreto aumento nas áreas com folhas secas.
<b>8</b>	As imagens registradas nas últimas 10 horas do experimento evidenciam a queda do estiolamento, conforme observado nas amostras analisadas.

Quadro 3 – Descrição de cada trecho em destaque na Fig. 56. Fonte: o autor.

A partir do trecho **5**, correspondente ao segundo estágio de estresse hídrico, o experimento passou a apresentar imagens com regiões excessivamente claras. No entanto, a análise do gráfico de luminância (Fig. 59) indica que não houve variação significativa na iluminação ambiente durante essa fase, o que sugere que as alterações observadas, tanto no gráfico RGB quanto nas amostras, apontam para uma limitação

de configuração da câmera.

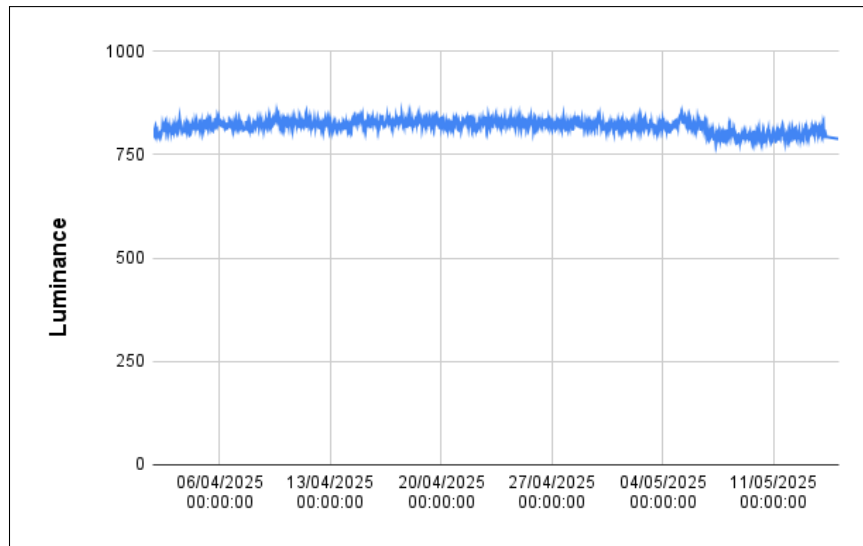


Figura 59 – Gráfico da luminância durante todo o experimento. Fonte: o autor.

A seguir, os gráficos da Fig. 60 representam os índices RGB médios em função da umidade do solo para cada um dos ciclos, separadamente:

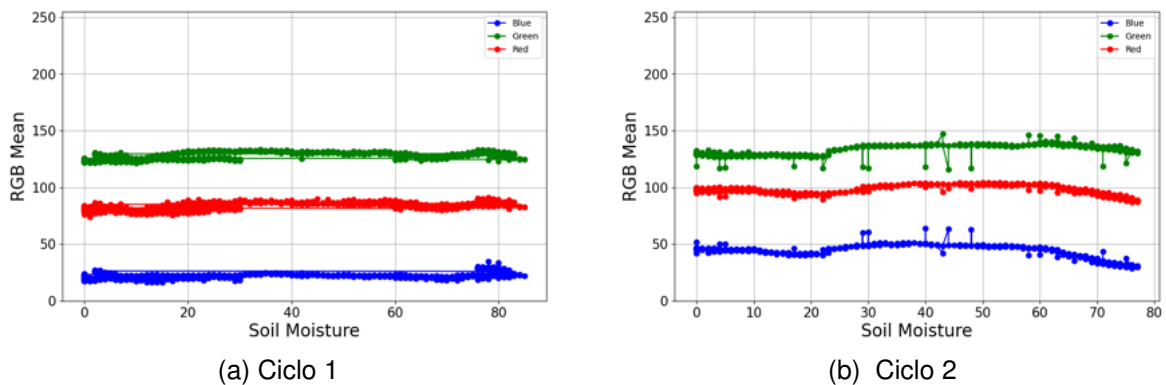


Figura 60 – RGB médio versus umidade do solo em cada um dos ciclos. Fonte: o autor.

A busca por um “bom” indicador de estresse hídrico não se mostra trivial por meio da análise do espaço de cor RGB, não sob a atual condição do experimento. Uma outra alternativa é analisar os índices médios do espaço de cor HSV.

De forma análoga à abordagem anterior, a Fig. 61, apresenta a variação média relativa dos canais H,S e V de cada imagem do dataset.

A partir do início de coleta dos dados até o momento em torno da segunda irrigação, tanto a matiz (H) quanto a saturação (S) e o valor de intensidade (V) sugerem relação com a umidade. Conforme mencionado na Tab. 3, o estiolamento mais evidente da alface, observado a partir do trecho 5, é provavelmente a principal causa das

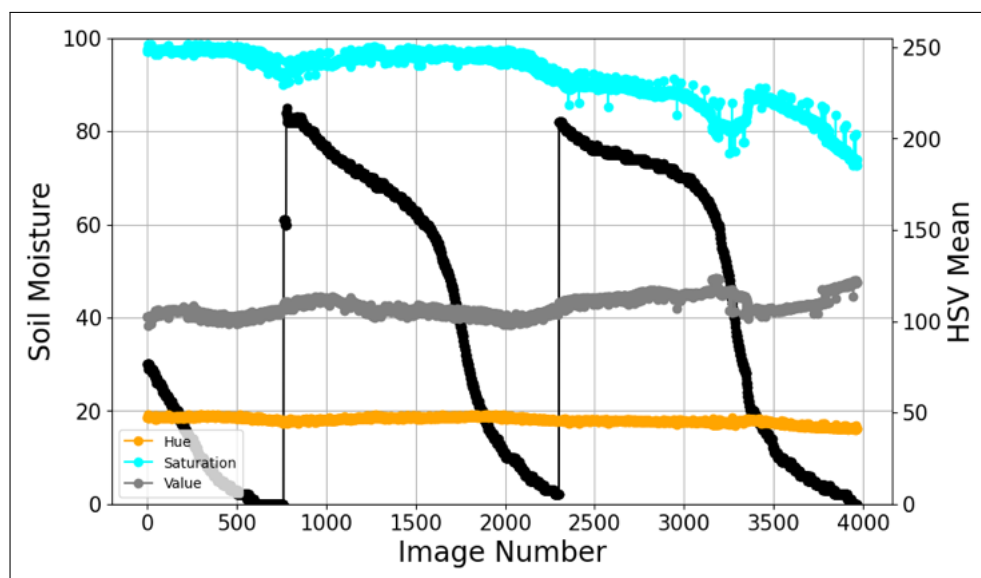


Figura 61 – Média H, S e V de cada imagem. Fonte: o autor.

discrepâncias nas medições em relação aos trechos anteriores — tanto nos espaços de cor RGB quanto HSV.

A seguir, os gráficos da Fig. 62 representam os índices HSV médios em função da umidade do solo para cada um dos ciclos, separadamente:

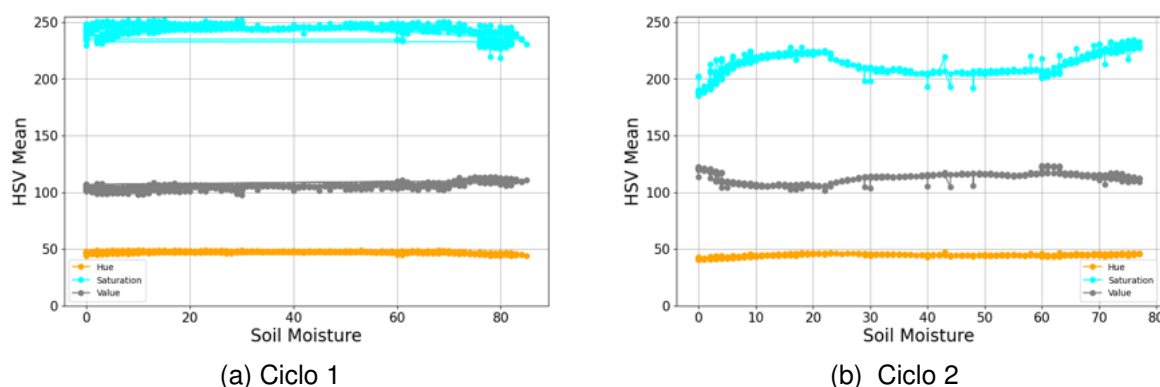


Figura 62 – HSV médio versus umidade do solo em cada um dos ciclos. Fonte: o autor.

Com o objetivo de avaliar com maior precisão o comportamento do canal de saturação nas imagens do primeiro ciclo (trechos de 1 a 5), a Fig. 63 apresenta a média móvel desse canal, utilizando uma janela de dez elementos, juntamente com a umidade relativa do solo — ambos plotados em função do tempo de amostragem.

Nesse contexto, os dados indicam que, cerca de 105 minutos (1 hora e 45 minutos) após a irrigação, ocorre uma inflexão no canal de saturação. No entanto, essa alteração não é perceptível visualmente. Considerando a hipótese de evitar que o solo atinja 0% de umidade relativa, uma abordagem possível seria impedir que a derivada

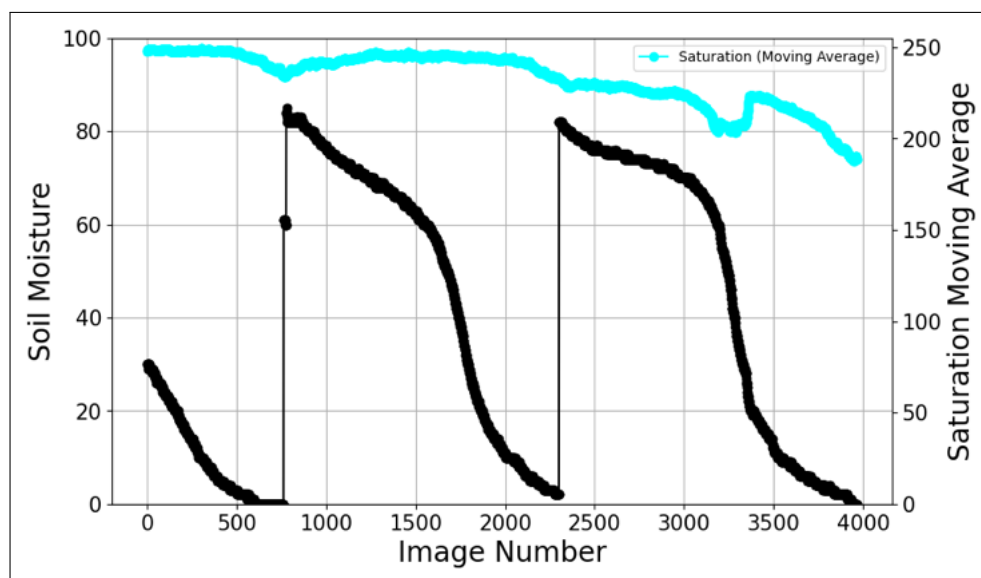


Figura 63 – Média móvel do canal S com janela = 10. Fonte: o autor.

da média móvel da saturação se torne negativa após a irrigação, ou ainda, definir um valor mínimo para esse dado como critério de controle.

Diante disso, surge o seguinte questionamento: a cor poderia auxiliar na antecipação de informações relacionadas à umidade? Para explorar essa possibilidade, foi calculada a média móvel para uma janela de dez elementos sobre os dados H. A detecção de reidratação foliar com base na cor ocorreu 6 horas e 30 minutos após a irrigação. Portanto, o canal de cor não é o melhor “detector” de umidade na planta, mas sim, o canal de saturação. Este é capaz de antecipar essa informação em quase 5 horas, quando comparado à cor.

### 7.1.2 Desvios Padrão dos Canais

Os desvios padrão de cada imagem segmentada em RGB e HSV estão registrados nos gráficos das Fig.64 e Fig. 65.

A seguir, os gráficos da Fig. 66 representam os desvios padrão RGB e HSV em função da umidade do solo para o primeiro ciclo:

Mais uma vez, o sistema de cor HSV — especialmente o canal de saturação — demonstra ser um forte indicativo de estresse hídrico da planta, pelo menos durante o primeiro ciclo do experimento.



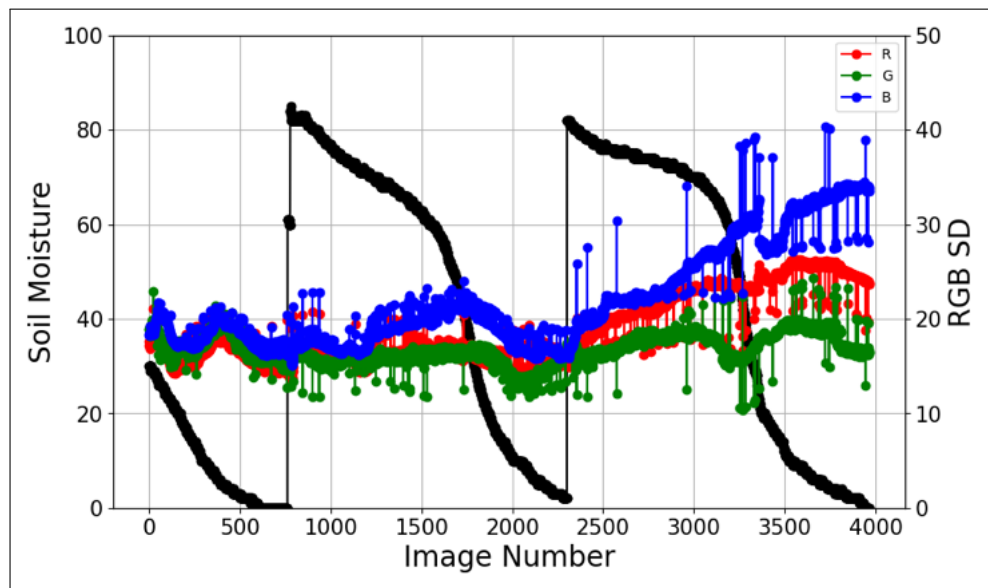


Figura 64 – Desvio padrão R, G e B de cada imagem. Fonte: o autor.

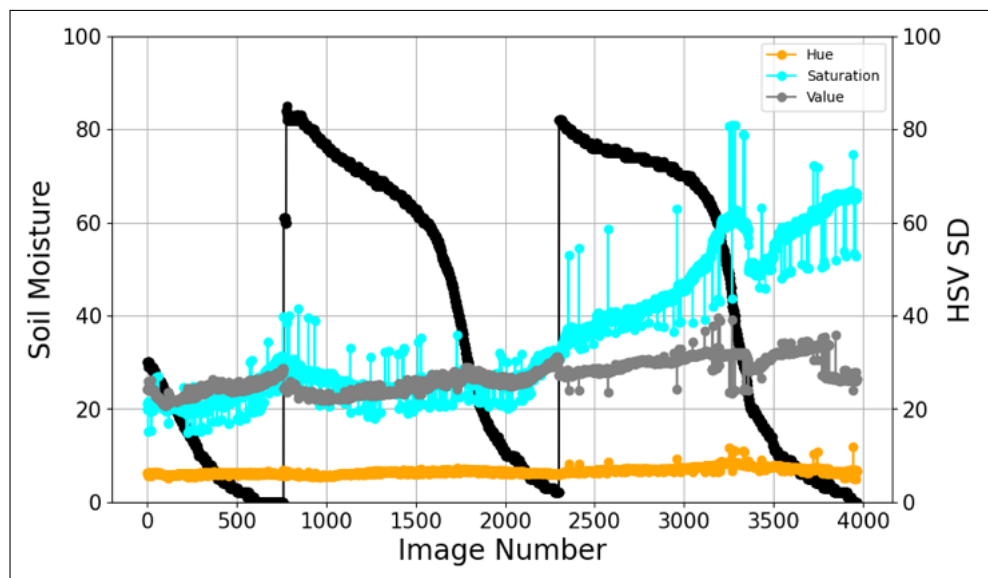
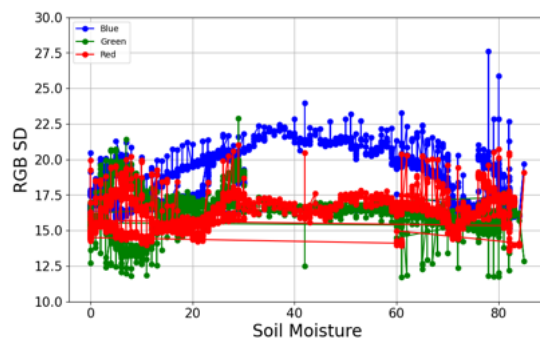
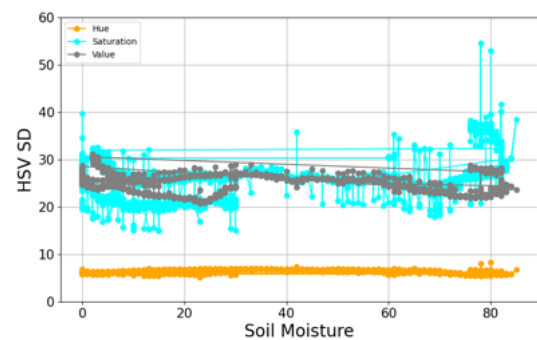


Figura 65 – Desvio padrão H, S e V de cada imagem. Fonte: o autor.



(a) Desvio Padrão RGB



(b) Desvio Padrão HSV

Figura 66 – Desvios Padrão RGB e HSV versus umidade do solo para o primeiro ciclo do experimento. Fonte: o autor.

## 7.2 ÍNDICE DE TROPISMO

Outra abordagem de fenotipagem através do SPheRe consiste na quantificação da mobilidade natural da planta, ou seja, o percentual de movimento relativo entre imagens consecutivas, indicando mudanças físicas observadas na planta, como deslocamento de folhas, inclinação ou recuperação pós-rega. O índice de tropismo proposto é estimado a partir da imagem “diferença” entre os pares, sendo então calculada a soma dos pixels dessa imagem resultante e a soma dos pixels da imagem anterior, e por fim, a razão entre estas somas.

Conforme gráfico da Fig. 67, períodos de baixo movimento (valores em torno de 3%) indicam estagnação visual (planta pouco responsiva). Picos de movimento (valores mais altos) indicam momentos em que a planta mudou significativamente de uma imagem para a próxima.

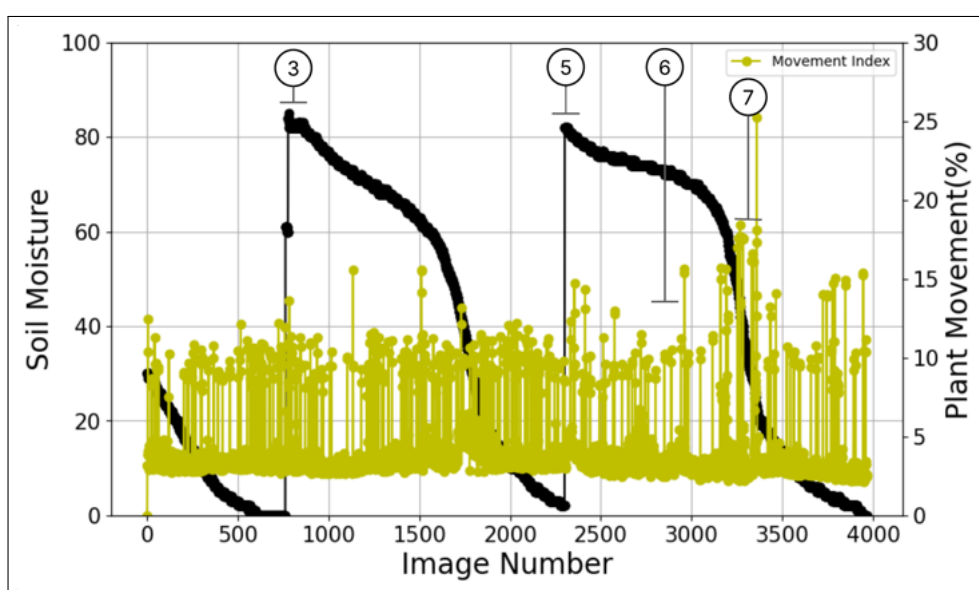


Figura 67 – Gráfico da diferença percentual entre imagens consecutivas, indicativo de movimentação da planta ao longo do tempo. Fonte: o autor.

A maioria das medições indicam pouca movimentação, com valores entre 2% e 3%, enquanto uma outra razoável parcela, em torno de 13%. No entanto, a partir do trecho **6**, observou-se um aumento gradual nesse índice, o que sugere um tropismo induzido pelo aumento do estiolamento. No trecho **7** esse índice ultrapassa 25%, e a partir dele diminui gradualmente, voltando a valores acima de 2%, o que assinalaria o efeito da gravidade sobre o estiolamento identificado nas imagens.

Destaca-se ainda a presença de tênues elevações desse índice imediatamente



após a irrigação, nos trechos **3** e **5**, o que poderia ser uma evidência do tropismo por fluxo de água no interior da planta.

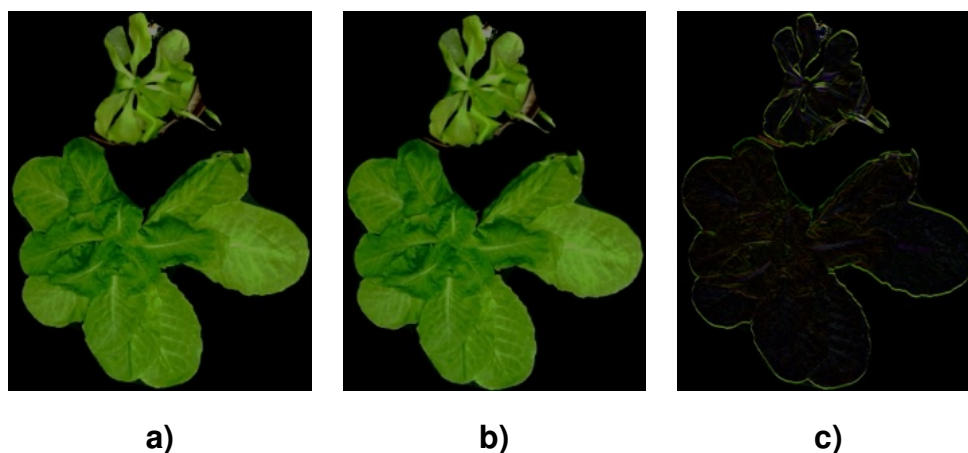


Figura 68 – Par de imagens que deu origem ao pico de 13% no trecho **3**: **a)** Imagem 5 horas e 15 minutos após a irrigação, **b)** imagem consecutiva, e **c)** subtração entre **a)** e **b)**. Fonte: o autor.

### 7.3 ANÁLISE DE TEXTURA

Com foco na primeira parte do experimento, denominada *Ciclo 1* – incluindo também amostras iniciais do *Ciclo 2* –, foram analisadas 2500 imagens para a extração das seguintes características de textura: *Contraste*, *Dissimilaridade*, *Homogeneidade*, *ASM* (Angular Second Moment), *Energia*, *Correlação*, *Média*, *Variância*, *Desvio Padrão* e *Entropia*, conhecidas como *características de Haralick*.

Conforme descrito na Seção 2.2.2, as imagens RGB segmentadas foram convertidas para escala de cinza e, em seguida, analisou-se a organização dos pixels de cada imagem. Para uma análise mais precisa, com foco nas folhas, utilizou-se o recorte automatizado das regiões segmentadas, conforme ilustrado na Fig.69, resultado da remoção de pixels nulos excedentes. Enquanto, a Fig 70 destaca as correlações entre as estatísticas de textura.

Após a extração das características de Haralick para cada uma das 2500 amostras, procedeu-se à normalização dos dados. Com os dados normalizados, foi possível verificar a existência de correlação entre características derivadas umas das outras — como, por exemplo, entre *ASM* e *energia*, ou entre *variância* e *desvio padrão* —, o que é evidenciado na matriz de correlação cruzada apresentada na Fig. 70.



Figura 69 – Amostra de imagem segmenta recortada. Fonte: o autor.

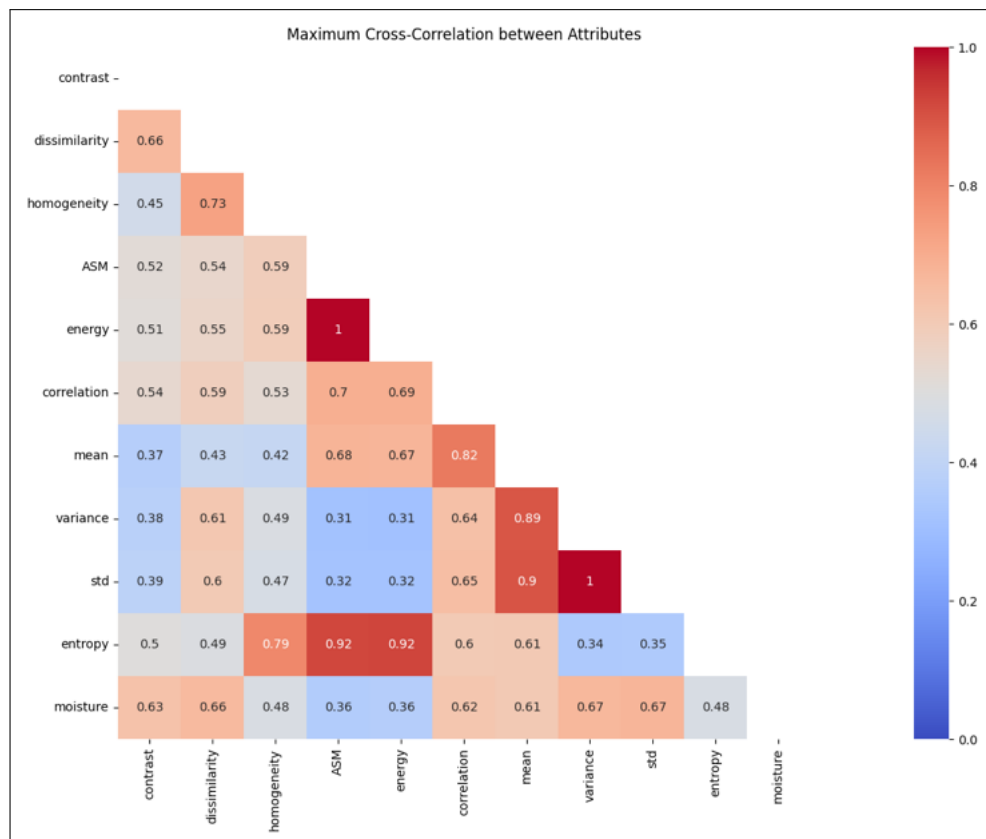


Figura 70 – Matriz de correlação cruzada entre as características de Haralick, incluindo o dado umidade do solo. Fonte: o autor.

Com base na análise dessa matriz, removemos as seguintes características do conjunto de dados: *ASM*, *energy*, *mean* e *variance*. A Tabela 1 apresenta estatísticas descritivas gerais do conjunto resultante.

Tabela 1 – Estatísticas descritivas das variáveis de textura e umidade

	contrast	dissimilarity	homogeneity	correlation	std	entropy	moisture
count	2500.000000	2500.000000	2500.000000	2500.000000	2500.000000	2500.000000	2500.000000
mean	0.196907	0.467374	0.277164	0.497662	0.286666	0.634657	0.456725
std	0.086752	0.143800	0.081843	0.138450	0.113677	0.090444	0.364338
min	0.000000	0.206978	0.000000	0.000000	0.000000	0.370929	0.000000
25%	0.127480	0.373631	0.247732	0.401268	0.195094	0.569082	0.094118
50%	0.189421	0.432648	0.291766	0.521743	0.271354	0.614747	0.341176
75%	0.249025	0.534993	0.320115	0.601810	0.357737	0.711267	0.835294
max	0.517463	0.967907	0.611053	0.779285	0.581058	0.848450	1.000000

E o *Boxplot* da Fig. 71, resume a distribuição dos atributos do conjunto de dados, destacando a tendência central, a dispersão e a presença de *outliers*.

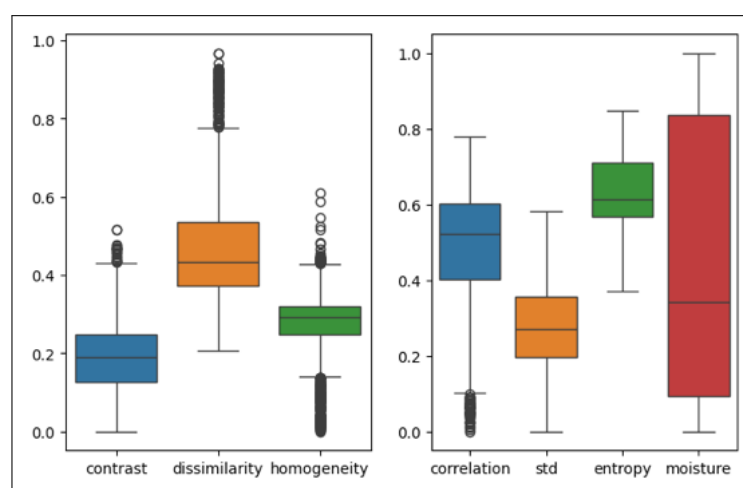


Figura 71 – *Boxplot* do conjunto de dados. Fonte: o autor.

Da análise da correlação cruzada destacamos duas das características mais associadas à umidade (*moisture*): *contraste* e *dissimilaridade*. Na sequência, são apresentados os gráficos dessas características em conjunto com os valores de umidade, ao longo do Ciclo 1.

Na Fig. 72, a métrica *contraste* representa a intensidade da variação local de tons de cinza entre um pixel e seus vizinhos. Valores elevados indicam uma textura com transições bruscas, enquanto valores baixos sugerem uma textura mais uniforme. O gráfico revela uma tendência de o contraste ultrapassar 0,4 antes da irrigação, quando o solo e a planta já se encontram praticamente desidratados. O oposto também é

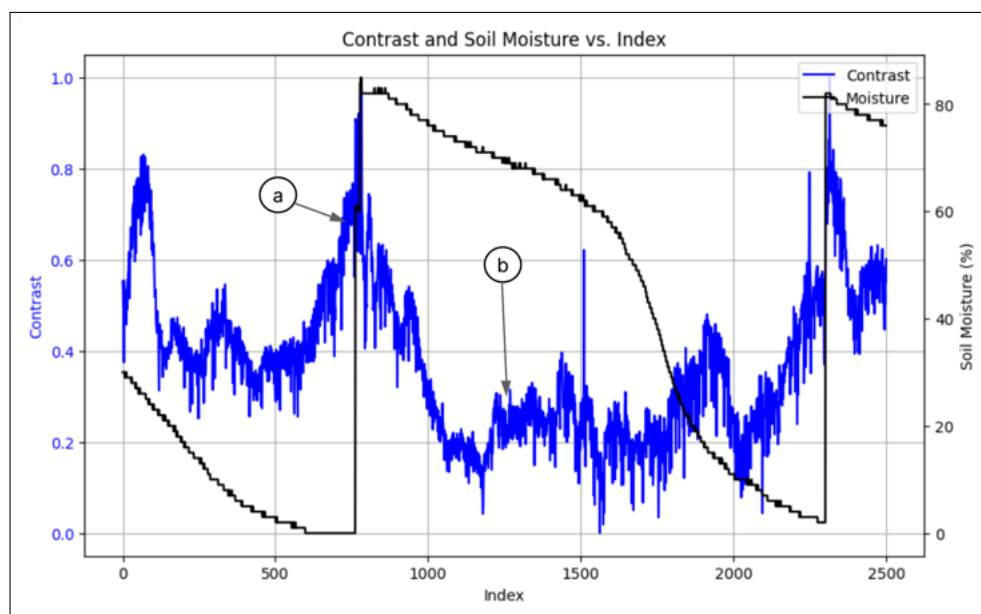


Figura 72 – Contraste e Umidade em função do período entre os dias 2 e 27 de abril. Fonte: o autor.

observado: após a hidratação, o contraste tende a diminuir, atingindo valores inferiores a 0,2.

Ainda na Fig. 72, os instantes **a** e **b** correspondem às amostras ilustradas na Fig. 73, ambas exibidas em escala de cinza e com distintos níveis da característica *contraste*, conforme indicado no gráfico.

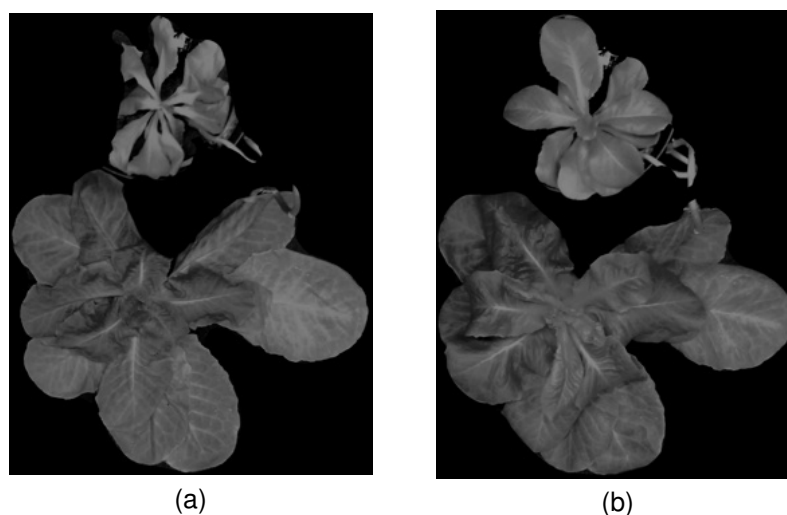


Figura 73 – Imagens de plantas evidenciando a mudança de contraste após a irrigação. (a) imagem da planta momentos antes da rega, com solo a 0% de umidade relativa, e medida contraste superior a de (b), outra amostra 5 dias após a rega, com solo a 70%. Fonte: o autor.

Ao analisar a média móvel do *contraste* com uma janela de dez leituras, foi possível identificar indícios de reidratação foliar cerca de 3 horas e 15 minutos após a irrigação.

Outra métrica relevante é a *dissimilaridade* (Fig. 74), que também mede a diferença entre tons de cinza vizinhos. Porém, diferentemente do *contraste*, essa medida não eleva as diferenças ao quadrado, como discutido na Seção 2.2.2, sendo mais sensível a variações leves do que o *contraste*.

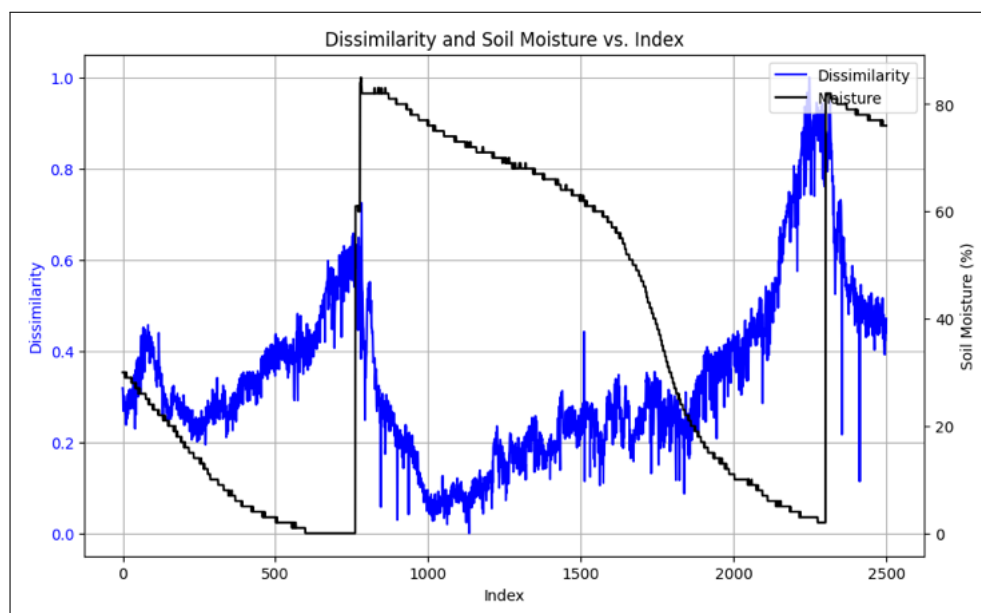


Figura 74 – Dissimilaridade e Umidade em função do período entre os dias 2 e 27 de abril. Fonte: o autor.

Uma provável discrepância entre os picos da *dissimilaridade* nos dois momentos da irrigação deve estar no desenvolvimento natural da planta, com sua morfologia dinâmica e pela formação de novas folhas.

A seguir, a Fig. 75 representa os dados *contrast* e *dissimilarity* em função da umidade do solo.

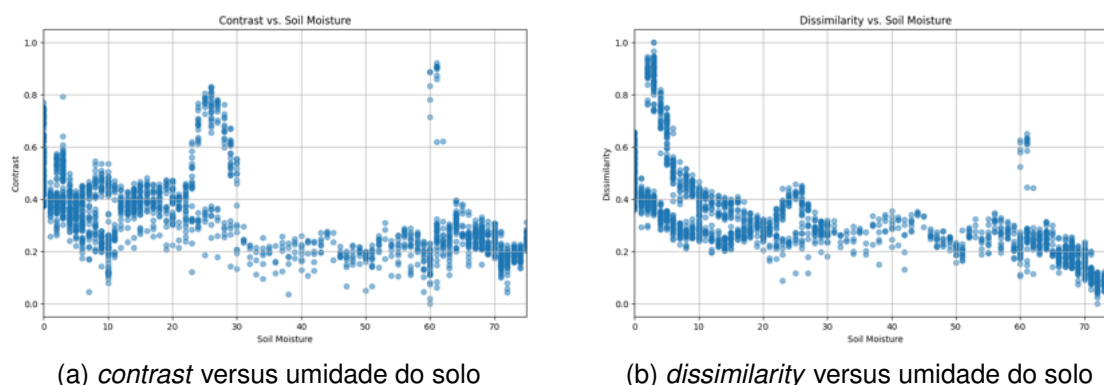


Figura 75 – Descritores de textura *contrast* e *dissimilarity* versus umidade do solo. Fonte: o autor.

Por fim, na Fig. 76, apresentamos gráficos das demais características extraídas.

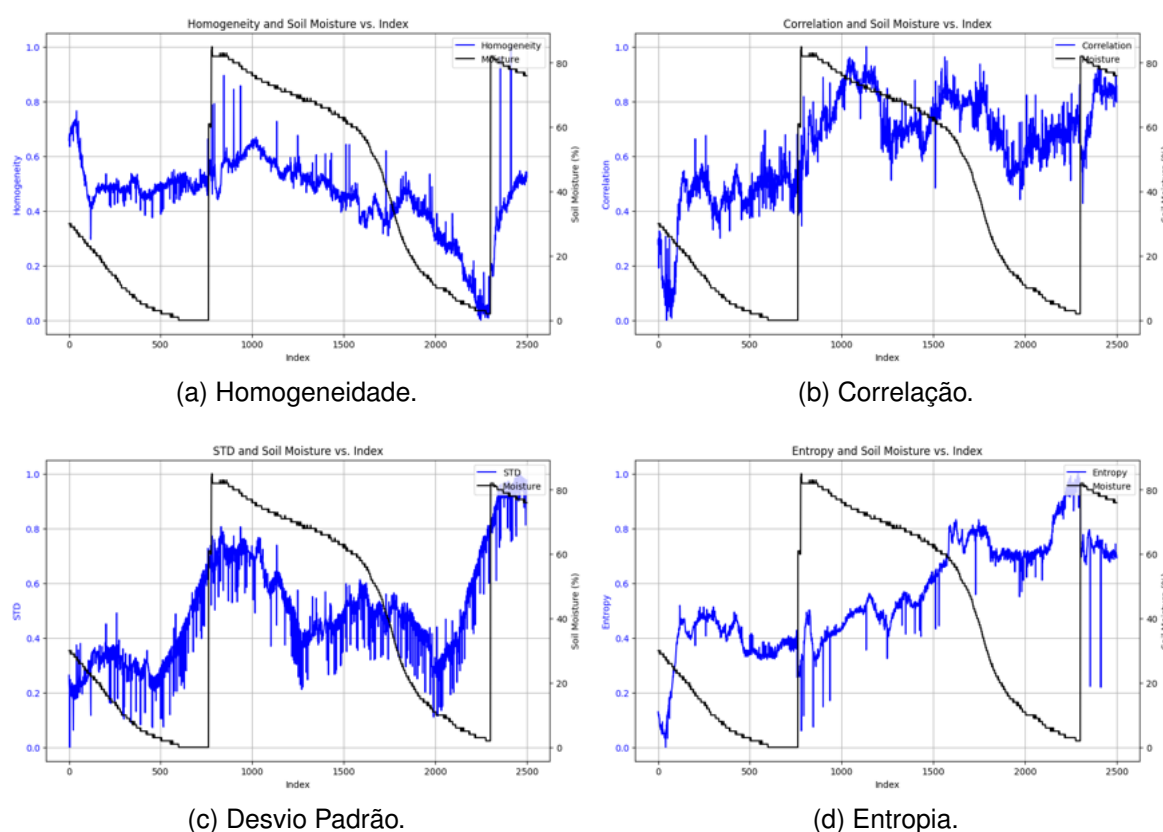


Figura 76 – Algumas das demais características de textura.

Quanto à *homogeneidade*, seus valores reduzidos indicam uma textura menos uniforme, com maior variação entre os tons de cinza. A *correlação* mede a relação linear entre tons de cinza em uma determinada direção — valores altos indicam repetições consistentes. Já o *desvio padrão* quantifica a dispersão dos tons. Por fim, a *entropia* é maior em texturas visualmente complexas, o que é o caso de experimentos com plantas, indicando maior desordem na distribuição tonal.

Até o momento, o foco do SPheRe tem sido realizar um diagnóstico detalhado da planta em cultivo, com o objetivo de identificar quais dados são mais relevantes para traduzir, de forma precisa, informações relacionadas ao estresse hídrico. Com vistas à implantação de sensores de irrigação inteligentes, apresentam-se a seguir alguns resultados obtidos a partir do emprego de técnicas de ML.

## 7.4 RUMO AOS FUTUROS SENSORES DE IRRIGAÇÃO INTELIGENTES

Visando expandir as funcionalidades do SPheRe, foram investigadas técnicas de ML, com o objetivo de aplicar inteligência na borda (*Edge Computing*). Essas inves-

tigações foram realizadas em ambiente em nuvem, especificamente Google Colab, dispondo dos seguintes recursos descritos na Tabela 2:

Tabela 2 – Resumo das características do sistema utilizado para os experimentos.

Recurso	Descrição
Processador	x86_64
Arquitetura	(64bit, ELF)
Sistema Operacional	Linux 6.1.123+
Núcleos físicos	4
Núcleos lógicos	8
Frequência da CPU	2,2 GHz
RAM	54,8 GB

#### 7.4.1 Classificação por meio de Arquiteturas Rasas

Dedicada à aplicação de aprendizado de máquina, esta etapa utilizou um dataset formado pelas características de textura extraídas na fase anterior e pelo atributo binário *Alert*. Esse atributo foi definido em duas classes: *Alert 1* (umidade entre 0% e 39%) e *Alert 0* (umidade entre 40% e 100%). O objetivo foi investigar a correlação entre os descritores de Haralick e essas classes, de modo a selecionar os mais relevantes para a classificação em modelos de arquitetura rasa por meio de algoritmos de aprendizado supervisionado, tais como Decision Tree, Random Forest e K-Nearest Neighbors (KNN).

Por fim, com as características GLCM já extraídas, o dataset foi categorizado nas duas classes de alerta com base nos valores de umidade, substituindo o valor contínuo pelo rótulo binário. Essa transformação permitiu a construção de uma nova matriz de correlação cruzada entre os descritores de Haralick e as classes *Alert*, cujos resultados são apresentados na Tab. 3. Cabe destacar, entretanto, as limitações do conjunto de dados: restrição a uma única espécie, número reduzido de amostras (um único vaso e módulo) e foco exclusivo no Ciclo 1, conforme discutido anteriormente.

Por meio de avaliação da matriz, foram eliminadas as mesmas características da seção anterior, apresentando alta correlação entre si, priorizando-se a permanência daquelas com maior correlação com o atributo *alert*. Dessa forma, as características *ASM*, *energy*, *mean* e *variance* foram excluídas do conjunto de dados, permanecendo

Tabela 3 – Correlação cruzada entre as características de Haralick e a variável *alert*

Atributo	Correlação com <i>alert</i>
contrast	0.57
dissimilarity	0.61
homogeneity	0.58
ASM	0.46
energy	0.47
correlation	0.68
mean	0.60
variance	0.60
std	0.61
entropy	0.56

os seguintes atributos selecionados: *contrast*, *dissimilarity*, *homogeneity*, *correlation*, *std* e *entropy*. Nenhum *outlier* foi removido do dataset.

Com o dataset reduzido em atributos, efetuamos a divisão do conjunto em dados de treinamento e dados de testes, 70% e 30%, correspondendo a 1750 e 750 amostras, respectivamente.

A proporção de 70% para treinamento e 30% para teste foi adotada visando equilibrar a necessidade de dados suficientes para o aprendizado dos modelos e a obtenção de um conjunto de teste estatisticamente representativo. Embora a divisão 80/20 seja comumente empregada, optou-se por 70/30 devido ao tamanho limitado do conjunto de imagens e à necessidade de assegurar uma avaliação mais robusta do desempenho, evitando vieses decorrentes de um conjunto de teste reduzido (KUHN; JOHNSON, 2013; RASCHKA; MIRJALILI, 2019).

Realizamos o treinamento com base em três tipos de modelos. A seguir traremos a matriz de confusão correspondente e teceremos algumas discussões.

O primeiro modelo treinado foi baseado em *Árvore de Decisão* (*Decision Tree*). A partir da aplicação do modelo sobre os dados de teste, foi obtida a matriz de confusão apresentada na Fig. 77. Essa matriz resultou em uma acurácia global de 96,53%, com uma precisão de 96,46% para a classe positiva.

Na sequência, foi utilizado um modelo baseado em *Floresta Aleatória* (*Random Forest*). Após o treinamento, o modelo foi avaliado sobre o mesmo conjunto de dados de teste, resultando em uma nova matriz de confusão, apresentada na Fig.78. Os resultados indicaram uma acurácia de 97,87% e uma precisão de 98,71% para a classe



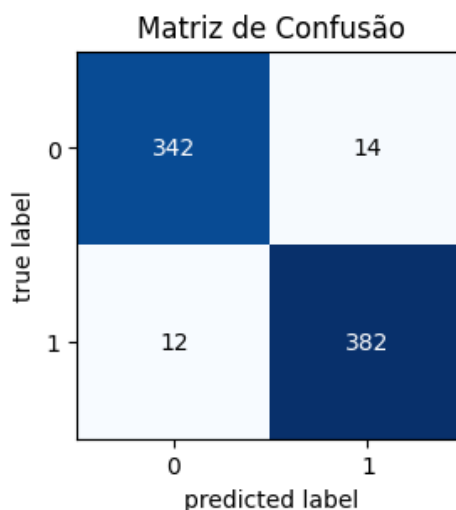


Figura 77 – Matriz de Confusão da *Decision Tree*. Fonte: o autor.

positiva.

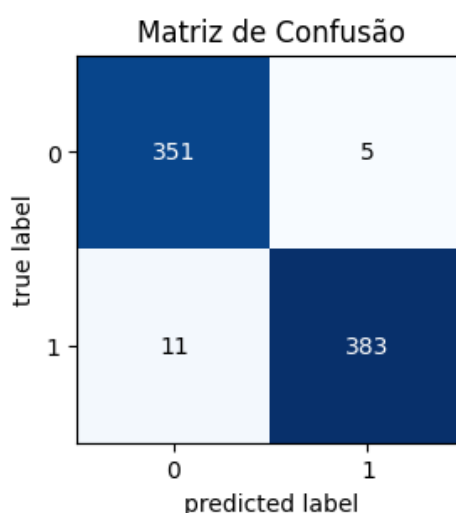


Figura 78 – Matriz de Confusão do *Random Forest*. Fonte: o autor.

Por fim, foi treinado um modelo do tipo *KNN*. No contexto deste trabalho, considerando um conjunto de 2500 amostras, com 6 atributos representados em ponto flutuante (4 bytes por valor), o consumo de memória apenas por parte do dataset fica estimado em aproximadamente 60KB.

A aplicação do modelo sobre os dados de teste resultou na matriz de confusão apresentada na Fig.79, a partir da qual foi obtida uma acurácia de 97,73% e uma precisão de 98,96% para a classe positiva.

O algoritmo baseado em *KNN* demonstra ser o mais robusto a falsos positivos. Porém, mais oneroso frente ao *Decision Tree* quanto à demanda de memória RAM,

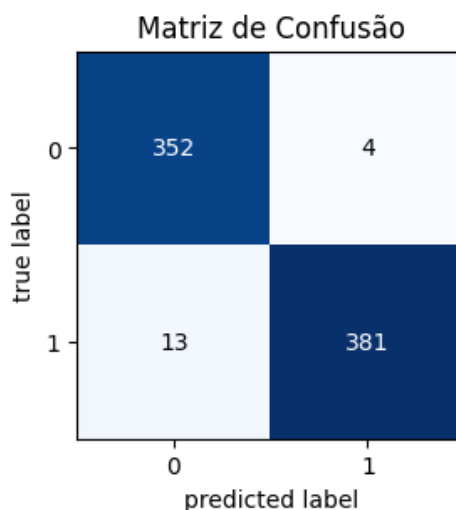


Figura 79 – Matriz de Confusão da *KNN*. Fonte: o autor.

pois exige que todo, ou parte do dataset, transite entre a memória secundária e a principal.

Além dos métodos tradicionais de classificação, abordagens baseadas em arquiteturas profundas têm ganhado destaque devido à sua capacidade de capturar padrões complexos em imagens. Nesse sentido, a próxima seção apresenta a classificação por meio de redes neurais convolucionais (CNNs), utilizando estratégias de aprendizado por transferência na tentativa de potencializar os resultados obtidos.

#### 7.4.2 Classificação por meio de Arquiteturas Profundas

Com base no mesmo dataset da subseção anterior, utilizou-se o conjunto de imagens que envolve as 2500 amostras pertencentes ao Ciclo 1 e ao início do Ciclo 2. O processo de aprendizado por transferência foi conduzido a partir da organização e pré-processamento do dataset de imagens, separadas em duas classes conforme o valor de umidade do solo, seguindo o mesmo critério explicado anteriormente. Como modelo, adotou-se a arquitetura MobileNetV3Small, pré-treinada no ImageNet. Modelo que une excelente acurácia e ideal para implante em dispositivos com recursos computacionais limitados. Se trata de um modelo configurado para receber imagens no formato  $(224 \times 224 \times 3)$ . Portanto, para inferência, as imagens foram submetidas a recorte sobre a área de interesse e redimensionamento para  $224 \times 224$  pixels e normalização dos valores dos pixels. O treinamento foi realizado em lotes (batch size) de 32 imagens, com divisão do conjunto de dados em 20% para validação e 10% para

teste (250 imagens), mantendo o valor da seed em 42 para reprodutibilidade. O encoder da MobileNet foi mantido congelado, sendo acrescentadas camadas densas (com 128 unidades e ativação ReLU), camadas de normalização em lote (BatchNormalization) e camadas de Dropout (com taxas de 0.3 e 0.5) para mitigar overfitting. A Tab.4 representa o modelo final.

Tabela 4 – Resumo das camadas e parâmetros do modelo MobileNetV3Small customizado.

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 224, 224, 3)	0
sequential (Sequential)	(None, 224, 224, 3)	0
rescaling_1 (Rescaling)	(None, 224, 224, 3)	0
MobileNetV3Small (Functional)	(None, 7, 7, 576)	939,120
global_average_pooling2d	(None, 576)	0
batch_normalization	(None, 576)	2,304
dropout	(None, 576)	0
dense	(None, 128)	73,856
batch_normalization_1	(None, 128)	512
dropout_1	(None, 128)	0
dense_1	(None, 2)	258

O otimizador utilizado foi o Adam, com função de perda *sparse categorical cross-entropy*. O modelo foi treinado por 50 épocas, aplicando técnicas de *data augmentation* (flip horizontal, rotação, zoom e ajuste de contraste) para aumentar a robustez do aprendizado.

Após o treinamento, a avaliação do modelo foi conduzida utilizando os dados de teste. Da matriz de confusão resultante, Fig. 80, extraímos métricas como acurácia, em torno de 80,40%, e precisão de 73% para classe positiva.

A Tabela 5 reúne a acurácia e precisão de classe positiva de cada modelo avaliado até então, o tempo do pipeline executado em nuvem, envolvendo captura, processamento e inferência também está incluso na tabela.

Tabela 5 – Acurácia, Precisão e Tempo do pipeline no Colab.

Modelo	Acurácia(%)	Precisão(%)	Tempo(s)
Decision Tree	96,53	96,46	0,29
Random Forest	97,87	98,71	0,30
KNN	97,73	98,96	0,29
CNN	80,40	73	0,39

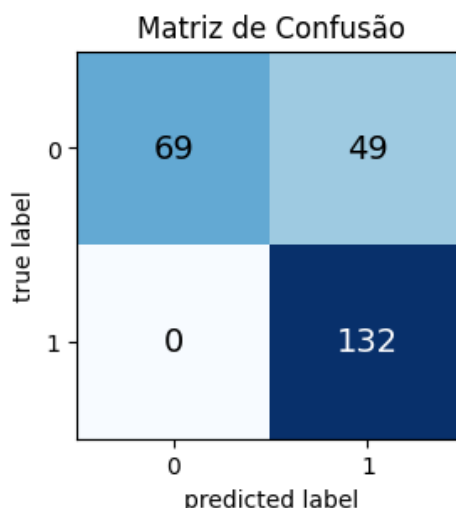


Figura 80 – Matriz de Confusão após testes com a *MobileNetV3Small*. Fonte: o autor.

Com os resultados do modelo validados em ambiente de nuvem, buscou-se a implementação do processamento e inferência diretamente no dispositivo embarcado. A seguir, uma descrição dos resultados desta implementação.

### 7.4.3 Edge Computing

Até o momento, todos os testes dos modelos, bem como a avaliação de suas métricas de desempenho, foram realizados em ambiente de nuvem utilizando a plataforma Google Colab. A partir desta etapa, no contexto de computação de borda, o SPheRe passa a ser explorado visando executar as etapas do pipeline de inferência no próprio módulo ESP32-S3, sem a dependência de infraestrutura em nuvem. Antes de avançarmos, a Tab. 6 apresenta um resumo das principais características desse SoC.

Tabela 6 – Principais recursos de hardware da ESP32-S3

SRAM(KB)	PSRAM (MB)	Flash (MB)	Clock (MHz)
413,21	8	8	160

Nesse âmbito, SRAM (Static RAM) é a memória interna dos microcontroladores ESP32-S3, rápida e estável, mas de capacidade limitada. E a PSRAM (Pseudo SRAM), uma memória externa baseada em DRAM, acessada de forma semelhante à SRAM pelo processador, oferecendo mais espaço a custo e desempenho intermediários. Enquanto a SRAM é usada para tarefas críticas e buffers pequenos, a PSRAM é ideal para processar grandes volumes de dados, como imagens e modelos de IA. A PSRAM

permite expandir a capacidade do sistema, apesar de ser mais lenta que a SRAM.

Com base nesses recursos disponíveis, prosseguimos com o desafio de classificação de imagens.

O foco do experimento passa a ser a implementação dos modelos diretamente no microcontrolador ESP32-S3, possibilitando a análise de sua performance em termos de tempo de execução e demais restrições inerentes ao hardware embarcado.

Ressalta-se que, a ênfase não está na qualidade (ou coerência) da classificação, mas sim em verificar a factibilidade de execução dessas abordagens no dispositivo.

Para viabilizar a execução das arquiteturas tradicionais no dispositivo embarcado, foi necessário realizar ajustes, como a redução da resolução das imagens capturadas, de  $2560 \times 1600$  para  $1024 \times 768$ , a utilização de filtros menores nas convoluções, entre outras simplificações. Esses ajustes permitiram a execução total do pipeline, embora sem avaliação da qualidade dos resultados.

Após a exploração das arquiteturas tradicionais e das adaptações necessárias para sua execução em ambiente embarcado, destaca-se a adoção de modelo baseado em redes neurais convolucionais (CNNs).

A Tabela 7 indica que o modelo treinado resultou em um total de 1.091.574 parâmetros do tipo *ponto-flutuante*, de 32 bits cada. Visando a implementação em hardware embarcado, esse modelo passou pelo processo de quantização e conversão para o formato TensorFlow Lite (TFLite), reduzindo cada parâmetro real para um valor inteiro de 8 bits. A quantização possibilitou a redução do tamanho do modelo em 4 vezes, de 4,16MB para 1,04MB, tornando-o adequado para as restrições de memória e processamento do microcontrolador ESP32-S3, permitindo assim a execução da inferência local, sinalizando viabilidade do monitoramento inteligente através do dispositivo de borda.

Tabela 7 – Resumo dos parâmetros do modelo MobileNetV3Small customizado.

<b>Total params</b>	1,091,574 (4.16 MB)
<b>Trainable params</b>	75,522 (295.01 KB)
<b>Non-trainable params</b>	940,528 (3.59 MB)

A seguir, a Tab. 8 apresenta um resumo do consumo de memória Flash para armazenamento do firmware, contemplando as etapas de captura e processamento de imagens — incluindo a extração de características de textura apenas no caso das ar-

quitaturas rasas —, bem como a inferência dos modelos implantados na ESP32-S3. O tempo total de execução do pipeline compõe a tabela.

Tabela 8 – Consumo de memória flash e tempo de execução do pipeline na ESP32-S3.

<b>Modelo</b>	<b>Flash (MB)</b>	<b>Tempo execução (s)</b>
Decision Tree	0,303	10,06
Random Forest	0,437	10,06
KNN	0,370	9,57
CNN	1,64	17,75

Com base nos testes preliminares, o *KNN*, ao contrário do esperado, demonstrou desempenho levemente superior aos demais modelos no que diz respeito ao tempo de execução do fluxo.

Devemos ressaltar que, por causas ainda desconhecidas não foi possível obter métricas para modelo CNN reduzido por quantização completa de 8 *bits*. Para o modelo CNN com quantização híbrida (*int8* e *float*) foi registrada uma menor acurácia de 53%, aproximadamente, em contrapartida, seu *deploy* não foi possível, também por razões desconhecidas.

## 7.5 DISCUSSÃO

Mesmo com um conjunto de dados limitado, os resultados deste trabalho demonstram a viabilidade do uso de imagens RGB para a detecção precoce de estresse hídrico em plantas, apoiando a hipótese de que alterações de cor nas folhas refletem, de forma mensurável, a condição hídrica do solo. Os experimentos conduzidos confirmaram que, mesmo empregando dispositivos de baixo custo e capacidade computacional limitada, é possível realizar a aquisição, o processamento e a análise automática de imagens, assim como a integração de dados ambientais em tempo real.

A comparação entre diferentes modelos ML — desde arquiteturas rasas, como Decision Tree, Random Forest e KNN, até arquiteturas profundas, como a MobileNetV3Small — evidencia que modelos de arquitetura rasa apresentam maior acurácia e precisão, desde que a análise de textura seja realizada com imagens de resolução mais alta. Somado a isto, para a adequação aos modelos de arquiteturas profundas, usualmente, a resolução da imagem de entrada é significativamente menor, o que

pode indicar baixa capacidade de capturar nuances de textura, convergindo para uma classificação baseada em cor.

Observou-se, ainda, que uma melhor configuração de câmera deve ser investigada. Apesar dessa limitação, a abordagem proposta mostrou-se promissora para aplicações reais, indicando caminhos para ajustes futuros, como o uso de técnicas de normalização de iluminação e o aumento do dataset. Por fim, a integração do sistema de sensoriamento inteligente ao ciclo automático de irrigação representa um avanço para soluções autônomas e escaláveis na agricultura de precisão, promovendo sustentabilidade e eficiência no uso de recursos hídricos.

O estudo demonstra viabilidade de fenotipagem automática via métricas simples e CNN, com potencial aplicação real em sistemas de irrigação inteligente.

## 8 CONCLUSÃO

Cientes do alcance dos objetivos expostos na seção 1.1, esta dissertação apresentou o desenvolvimento e a validação de um sistema inteligente de fenotipagem para monitoramento do estresse hídrico em plantas, com ênfase no uso de imagens RGB e aprendizado de máquina embarcado em microcontroladores de baixo custo. Os resultados experimentais evidenciaram que a coloração das folhas pode, de fato, ser empregada como um indicativo sensível do estado hídrico do solo, permitindo a antecipação de intervenções de irrigação de forma precisa e eficiente.

Os testes realizados mostraram que é possível implementar soluções de análise de imagens e classificação de estresse hídrico em dispositivos com recursos limitados, promovendo a descentralização do processamento (*Edge Computing*) e reduzindo a dependência de infraestrutura de nuvem. Os modelos de aprendizado devem ser melhor avaliados diretamente nos dispositivos embarcados, a fim de obtenção de métricas de desempenho em campo, exigindo ajustes para equilibrar precisão, uso de memória e tempo de resposta.

O sistema proposto apresentou diferentes alternativas de *deploy*. Dentre elas, destaca-se como a de mais fácil implementação a análise da componente de cor H. Uma segunda alternativa, que se mostrou promissora, consiste no uso de modelos de arquitetura rasa baseados em dados de textura. Por fim, também foi considerada a utilização de modelos de arquitetura profunda do tipo CNN, embora tal abordagem exija experimentos ainda mais rigorosos.

Entre as limitações do estudo, destacam-se a variabilidade inerente das plantas, a oscilação na qualidade das imagens — possivelmente decorrente de limitações de configuração da câmera —, a necessidade de ampliação do banco de dados de imagens, bem como a adaptação dos modelos para diferentes culturas e ambientes.

Como perspectivas para trabalhos futuros, destacam-se as seguintes propostas:

- Explorar modelos baseados em CNN customizados a imagens com dimensões de entrada maiores. Isso pode contribuir com o aumento da acurácia e robustez do modelo, ponderando frente às restrições de memória, tempo de execução e consumo energético dos microcontroladores utilizados.
- Realizar experimentos com ciclos de estresse hídrico menores.



- Incluir o armazenamento local de dados e imagens em cartão de memória, com envio posterior quando a comunicação for restabelecida, garantindo maior robustez frente a falhas de conectividade.
- Investigar o impacto da variação da altura da câmera sobre a qualidade das análises.
- Expandir o sistema com a inclusão de múltiplos sensores de umidade distribuídos pelo cultivo.
- Mapear o consumo energético do sistema, visando a otimização da eficiência operacional.
- Analisar informações adicionais coletadas pelo módulo ambiental.
- Instalar câmeras em diferentes posições para ampliar a captação de movimentos, identificar novas características fenotípicas e aumentar o volume de dados coletados.
- Extrair atributos de textura diretamente de imagens em escala de cinza, dispensando a etapa de recorte das folhas, o que pode simplificar e agilizar o processamento.
- Investigar abordagens de fusão sensorial e integrar sensores multiespectrais, incluindo câmeras no infravermelho próximo.
- Aplicar técnicas de normalização para mitigar interferências externas.

Em suma, o SPheRe pode contribuir para um avanço em soluções inteligentes e acessíveis para a agricultura de precisão, promovendo o uso sustentável da água e possibilitando o desenvolvimento de sensores autônomos e não invasivos baseados em visão computacional.

## REFERÊNCIAS

- ABDELMONEIM, A. A.; KIMAITA, H. N.; KALAANY, C. M. A.; DERARDJA, B.; DRAGONETTI, G.; KHADRA, R. IoT sensing for advanced irrigation management: A systematic review of trends, challenges, and future prospects. *Sensors (Basel, Switzerland)*, v. 25, n. 7, p. 2291, 2025.
- ABRAHAM, T.; TODD, A.; ORRINGER, D. A.; LEVENSON, R. Applications of artificial intelligence for image enhancement in pathology. In: *Artificial Intelligence in Pathology*. [S.l.]: Elsevier, 2025. p. 183–210.
- AC Infinity. *Cloudline Pro T4 Quiet Inline Duct Fan System with Temperature, Humidity, VPD Controller, 4-inch*. 2025. <[https://acinfinity.com/hydroponics-growers/cloudline-pro-t4-quiet-inline-duct-fan-system-with-temperature-humidity-vpd-controller-4-inch/?searchid=2803676&search\\_query=cloudline+t4](https://acinfinity.com/hydroponics-growers/cloudline-pro-t4-quiet-inline-duct-fan-system-with-temperature-humidity-vpd-controller-4-inch/?searchid=2803676&search_query=cloudline+t4)>. Acesso em: 25 maio 2025.
- Adafruit. *Adafruit STEMMA Soil Sensor - I2C Capacitive Moisture Sensor*. 2025. <<https://www.adafruit.com/product/4698>>. Acesso em: 25 maio 2025.
- AGARWAL, R. *Object Detection: An End-to-End Theoretical Perspective*. 2015. Medium, Toward Data Science, 22 de setembro de 2018. Acesso em: 19 de maio de 2025. Disponível em: <<https://medium.com/data-science/object-detection-using-deep-learning-approaches-an-end-to-end-theoretical-perspective-4ca27eee8a9a>>.
- AgOpen Shop. *AgOpenGps*. 2025. <<https://agopen.shop/>>. Acesso em: 26 maio 2025.
- AIELLO, G.; GIOVINO, I.; VALLONE, M.; CATANIA, P.; ARGENTO, A. A decision support system based on multisensor data fusion for sustainable greenhouse management. *Journal of Cleaner Production*, v. 172, p. 4057–4065, 2018. ISSN 0959-6526. <<https://doi.org/10.1016/j.jclepro.2017.02.197>>.
- AL-TAMIMI, N.; LANGAN, P.; BERNÁD, V.; WALSH, J.; MANGINA, E.; NEGRÃO, S. Capturing crop adaptation to abiotic stress using image-based technologies. *Open Biology*, The Royal Society, v. 12, n. 6, p. 210353, 2022.
- ALHASNAWI, B. N.; JASIM, B. H.; ISSA, B. A. Internet of things (IoT) for smart precision agriculture. *Iraqi Journal for Electrical and Electronic Engineering*, v. 16, n. 1, p. 28–38, 2020.
- ANASS, D.; MADI, A. A.; ALIHAMIDI, I.; CHARAF, L. A.; SABER, M. A novel autonomous remote system applied in agriculture using transmission control protocol. *Int J Reconfigurable & Embedded Syst*, v. 11, n. 1, p. 1–12, 2022.
- ARGUS. *Go Green - Client Stories*. 2023. <https://arguscontrols.com/client-stories/go-green>. Accessed May 23, 2025.
- Argus Controls. *Titan System*. 2025. <<https://arguscontrols.com/products-and-solutions/control-systems/titan-system>>. Acesso em: 25 maio 2025.
- ATANASOV, S. Methodology for irrigation water uptake time estimation based on rgb colorimetric measurements of leaves (a visual-graphical observation). In: IOP PUBLISHING. *IOP Conference Series: Materials Science and Engineering*. [S.l.], 2021. v. 1031, n. 1, p. 012016.

- ATANASOV, S. Automated remote sensing system for crops monitoring and irrigation management, based on leaf color change and piecewise linear regression models for soil moisture content predicting. *Scientific Horizons*, v. 27, n. 1, p. 127–139, 2023.
- BABU, E. V.; SYAMALA, Y.; BALARAMAKRISHNA, K.; RAMAKRISHNAIAH, T.; TALASILA, S. Green leaf disease detection system for agriculture using raspberry pi. *Nonlinear Optics, Quantum Optics: Concepts in Modern Optics*, v. 59, 2024.
- BARMAN, U.; SAIKIA, M. J. Smartphone contact imaging and 1-d cnn for leaf chlorophyll estimation in agriculture. *Agriculture*, MDPI, v. 14, n. 8, p. 1262, 2024.
- BAYILMIŞ, C.; EBLEME, M. A.; ÇAVUŞOĞLU, Ü.; KÜÇÜK, K.; SEVIN, A. A survey on communication protocols and performance evaluations for internet of things. *Digital Communications and Networks*, Elsevier, v. 8, n. 6, p. 1094–1104, 2022.
- Bluelab. *Bluelab One Pen*. 2025. <<https://bluelab.com/usa/bluelab-one-pen?srsId=AfmBOoqoSqJ4UbtGHWvQh9qqmpioFyNPZXI3d11CP-ZxphUiPy7BKe8h>>. Acesso em: 25 maio 2025.
- 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*. [S.l.: s.n.], 1992. p. 144–152.
- CARDOSO, F.; BLAWID, S. Monitoring the Daily Rhythm of Total Green Leaf Volatiles with a Low-Cost Multi-Sensor Node. In: . [S.l.: s.n.], 2022. (SloT - Symposium on Internet of Things), p. 1–4.
- CHANG, L.; YIN, Y.; XIANG, J.; LIU, Q.; LI, D.; HUANG, D. A Phenotype-Based Approach for the Substrate Water Status Forecast of Greenhouse Netted Muskmelon. *Sensors (Basel, Switzerland)*, v. 19, n. 12, p. 2673, 2019.
- CHAPPELL, M.; DOVE, S. K.; IERSEL, M. W. v.; THOMAS, P. A.; RUTER, J. Implementation of Wireless Sensor Networks for Irrigation Control in Three Container Nurseries. *HortTechnology*, v. 23, n. 6, p. 747–753, 2013. ISSN 1943-7714. <<https://doi.org/10.21273/horttech.23.6.747>>.
- CHRISTENSEN, D. C.; FONSECA, Z. P. da. Sistema de monitoramento agrícola utilizando lora 433-mhz, iot e aplicativo móvel. In: SBC. *Anais do XIV Congresso Brasileiro de Agroinformática*. [S.l.], 2023. p. 80–87.
- CIMPOI, M.; MAJI, S.; VEDALDI, A. Deep filter banks for texture recognition and segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2015. p. 3828–3836.
- CUI, N. Applying gradient descent in convolutional neural networks. In: IOP PUBLISHING. *Journal of Physics: Conference Series*. [S.l.], 2018. v. 1004, p. 012027.
- DANFORTH. *Phenotyping Core Facility*. 2025. <https://www.danforthcenter.org/our-work/core-facilities/phenotyping/>. Accessed May 23, 2025.
- DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. Imagenet: A large-scale hierarchical image database. In: IEEE. *2009 IEEE conference on computer vision and pattern recognition*. [S.l.], 2009. p. 248–255.

DEUTSCHER, G. *Through the language glass: Why the world looks different in other languages*. [S.l.]: Metropolitan books, 2010.

DUTTA, S. K.; BHUTIA, B.; MISRA, T.; MISHRA, V.; SINGH, S.; PATEL, V. Application and prospects of artificial intelligence (ai)-based technologies in fruit production systems. *Applied Fruit Science*, Springer, v. 67, n. 1, p. 1–18, 2025.

EL-BAKI, M. S. A.; IBRAHIM, M.; ELSAYED, S.; YASEEN, Z. M.; EL-FATTAH, N. G. A. Water status and plant traits of dry bean assessment using integrated spectral reflectance and rgb image indices with artificial intelligence. *Scientific Reports*, Nature Publishing Group UK London, v. 15, n. 1, p. 16808, 2025.

ERHAN, D.; COURVILLE, A.; BENGIO, Y.; VINCENT, P. Why does unsupervised pre-training help deep learning? In: JMLR WORKSHOP AND CONFERENCE PROCEEDINGS. *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. [S.l.], 2010. p. 201–208.

FarmBot. *FarmBot*. 2025. <[https://farm.bot/?srsltid=AfmBOopiQ\\_-Pv92UYyYZs-6fL7uiNNKCIKFFyL3Hul\\_2vIIosvyHFSsp](https://farm.bot/?srsltid=AfmBOopiQ_-Pv92UYyYZs-6fL7uiNNKCIKFFyL3Hul_2vIIosvyHFSsp)>. Acesso em: 26 maio 2025.

FERRAREZI, R. S.; DOVE, S. K.; IERSEL, M. W. v. An Automated System for Monitoring Soil Moisture and Controlling Irrigation Using Low-cost Open-source Micro-controllers. *HortTechnology*, v. 25, n. 1, p. 110–118, 2015. ISSN 1943-7714. <<https://doi.org/10.21273/horttech.25.1.110>>.

FIX, E. *Discriminatory analysis: nonparametric discrimination, consistency properties*. [S.l.]: USAF school of Aviation Medicine, 1985. v. 1.

FRIEDMAN, N.; GEIGER, D.; GOLDSZMIDT, M. Bayesian network classifiers. *Machine learning*, Springer, v. 29, p. 131–163, 1997.

FUKUSHIMA, K. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks*, Elsevier, v. 1, n. 2, p. 119–130, 1988.

GOMEDE, E. *Gray Level Co-occurrence Matrix (GLCM): A Comprehensive Analysis*. 2024. <https://medium.com/aimonks/gray-level-co-occurrence-matrix-glcma-comprehensive-analysis-a58bbc49c0f6>. Accessed May 22, 2025.

GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 3rd. ed. Upper Saddle River, NJ, USA: Prentice Hall, 2008. ISBN 0-13-168728-X, 978-0-13-168728-8.

GOUMOPOULOS, C.; O'FLYNN, B.; KAMEAS, A. Automated zone-specific irrigation with wireless sensor/actuator network and adaptable decision support. *Computers and Electronics in Agriculture*, v. 105, p. 20–33, 2014. ISSN 0168-1699. <<https://doi.org/10.1016/j.compag.2014.03.012>>.

GUIMARÃES, C.; STONE, L. Métodos de avaliação das condições hídricas das plantas. 2008.

HAAS, A.; MATHERON, G.; SERRA, J. Morphologie mathématique et granulométries en place. In: *Annales des mines*. [S.l.: s.n.], 1967. v. 11, n. 736-753, p. 7–3.

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.

HASAN, R. I.; YUSUF, S. M.; ALZUBAIDI, L. Review of the state of the art of deep learning for plant diseases: A broad analysis and discussion. *Plants*, MDPI, v. 9, n. 10, p. 1302, 2020.

HOSSAIN, S.; SEYAM, T. A.; CHOWDHURY, A.; GHOSE, R.; RAHAMAN, A.; HADIKA, Z.; PATHAK, A. Enhancing agricultural diagnostics: Advanced training of pre-trained cnn models for paddy leaf disease detection. *Machine Learning*, v. 10, n. 1, p. 1–13, 2025.

HOSSIN, M.; SULAIMAN, M. N. A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, Academy & Industry Research Collaboration Center (AIRCC), v. 5, n. 2, p. 1, 2015.

HOWARD, A. G.; ZHU, M.; CHEN, B.; KALENICHENKO, D.; WANG, W.; WEYAND, T.; ANDREETTO, M.; ADAM, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Inkbird. *Smart Sprinkler Controller IIC-800-WIFI*. 2025. <[https://inkbird.com/products/smart-sprinkler-controller-iic-800-wifi?\\_pos=1&\\_psq=IIC-800-WIFI&\\_ss=e&\\_v=1.0](https://inkbird.com/products/smart-sprinkler-controller-iic-800-wifi?_pos=1&_psq=IIC-800-WIFI&_ss=e&_v=1.0)>. Acesso em: 25 maio 2025.

Inkbird. *Temperature and Humidity Controller ITC-608T*. 2025. <[https://inkbird.com/products/temperature-and-humidity-controller-itc-608t?\\_pos=1&\\_psq=608t&\\_ss=e&\\_v=1.0](https://inkbird.com/products/temperature-and-humidity-controller-itc-608t?_pos=1&_psq=608t&_ss=e&_v=1.0)>. Acesso em: 25 maio 2025.

IoT-Store. *Arducam 8 MP Sony IMX219 Camera Module for Raspberry Pi*. 2025. <<https://iot-store.com.au/products/arducam-8-mp-sony-imx219-camera-module-for-raspberry-pi>>. Acesso em: 25 maio 2025.

ISLAM, M. M.; HIMEL, G. M. S.; MOAZZAM, M. G.; UDDIN, M. S. Artificial intelligence-based rice variety classification: A state-of-the-art review and future directions. *Smart Agricultural Technology*, Elsevier, p. 100788, 2025.

JÄHNE, B. Concepts, algorithms, and scientific applications. *Digital Image Processing*, Springer, p. 321–3, 1995.

KOIDL, K. Loss functions in classification tasks. *School of Computer Science and Statistic Trinity College, Dublin*, p. 1–5, 2013.

KOLHAR, S.; JAGTAP, J. Plant Trait Estimation and Classification Studies in Plant Phenotyping Using Machine Vision - A Review. *Information Processing in Agriculture*, 2021. ISSN 2214-3173.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, v. 25, 2012.

KUHN, M.; JOHNSON, K. *Applied Predictive Modeling*. New York: Springer, 2013.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *nature*, Nature Publishing Group UK London, v. 521, n. 7553, p. 436–444, 2015.

LI, J.; GU, W.; YUAN, H. Research on IOT Technology Applied to Intelligent Agriculture. In: *Proceedings of the 5th International Conference on Electrical Engineering and Automatic Control*. Weihai, China: Springer Berlin Heidelberg, 2016. (Lecture Notes in Electrical Engineering, v. 367), p. 1217–1224. <[https://doi.org/10.1007/978-3-662-48768-6\\_136](https://doi.org/10.1007/978-3-662-48768-6_136)>.

LI, M.; ZHANG, T.; CHEN, Y.; SMOLA, A. J. Efficient mini-batch training for stochastic optimization. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.: s.n.], 2014. p. 661–670.

LIU, L.; OUYANG, W.; WANG, X.; FIEGUTH, P.; CHEN, J.; LIU, X.; PIETIKÄINEN, M. Deep learning for generic object detection: A survey. *International Journal of Computer Vision*, Springer, v. 128, p. 261–318, 2019.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, Springer, v. 5, p. 115–133, 1943.

MILLER, T.; MIKICIUK, G.; DURLIK, I.; MIKICIUK, M.; ŁOBODZIŃSKA, A.; ŚNIEG, M. The iot and ai in agriculture: The time is now—a systematic review of smart sensing technologies. *Sensors (Basel, Switzerland)*, v. 25, n. 12, p. 3583, 2025.

MOBILENET. 2024. <<https://en.wikipedia.org/wiki/MobileNet>>. Acesso em: 20 de maio de 2025.

MOHAMED, E. S.; BELAL, A.; ABD-ELMABOD, S. K.; EL-SHIRBENY, M. A.; GAD, A.; ZAHRAN, M. B. Smart farming for improving agricultural management. *The Egyptian Journal of Remote Sensing and Space Science*, Elsevier, v. 24, n. 3, p. 971–981, 2021.

MORCHID, A.; JEBABRA, R.; KHALID, H. M.; ALAMI, R. E.; QJIDAA, H.; JAMIL, M. O. Iot-based smart irrigation management system to enhance agricultural water security using embedded systems, telemetry data, and cloud computing. *Results in Engineering*, Elsevier, v. 23, p. 102829, 2024.

NANNI, L.; BRAHNAM, S.; GHIDONI, S.; MENEGATTI, E.; BARRIER, T. Different approaches for extracting information from the co-occurrence matrix. *PLOS ONE*, v. 8, p. e67874, 2013.

NARKHEDE, M. V.; BARTAKKE, P. P.; SUTAONE, M. S. A review on weight initialization strategies for neural networks. *Artificial intelligence review*, Springer, v. 55, n. 1, p. 291–322, 2022.

NEMALI, K. S.; IERSEL, M. W. v. An automated system for controlling drought stress and irrigation in potted plants. *Scientia Horticulturae*, v. 110, n. 3, p. 292–297, 2006. ISSN 0304-4238. <<https://doi.org/10.1016/j.scienta.2006.07.009>>.

OTSU, N. A threshold selection method from gray-level histograms, *IEEE Trans. Systems, Man and Cybernetics*, vol. smc-9, n. 1, p. 62–66, 1979.

ÖZREÇBEROĞLU, N.; KAHRAMANOĞLU, İ. Mathematical models for the estimation of leaf chlorophyll content based on RGB colours of contact imaging with smartphones: A pomegranate example. *Folia Horticulturae*, v. 32, n. 1, p. 57–67, 2020.

- PARDEDE, H. F.; SURYAWATI, E.; KRISNANDI, D.; YUWANA, R. S.; ZILVAN, V. Machine learning based plant diseases detection: A review. In: IEEE. *2020 International conference on radar, antenna, microwave, electronics, and telecommunications (ICRAMET)*. [S.l.], 2020. p. 212–217.
- PATHAK, A. R.; PANDEY, M.; RAUTARAY, S. Application of deep learning for object detection. *Procedia computer science*, Elsevier, v. 132, p. 1706–1717, 2018.
- PATRO, S.; SAHU, K. K. Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*, 2015.
- PEDRINI, H.; SCHWARTZ, W. R. *Análise de imagens digitais: princípios, algoritmos e aplicações*. [S.l.]: Cengage Learning, 2008.
- PERCEBES, B.; REIS, G.; ANGULO, A. *Monitoramento da Erosão do Solo através de um Sistema IoT que Integra o Método da Parcela de Erosão e uma Estação Meteorológica*. 2023. <https://www.ic.unicamp.br/reltech/2023/abstracts.html>. Accessed in 2025-05-24.
- PETROVIC, B.; TUNGUZ, V.; BARTOS, P. Application of computer vision in livestock and crop production—a review. *Computing and Artificial Intelligence*, v. 1, n. 1, p. 360–360, 2023.
- PHIPHIPHATPHAISIT, S.; SURINTA, O. Food image classification with improved mobilenet architecture and data augmentation. In: *Proceedings of the 3rd International Conference on Information Science and Systems*. [S.l.: s.n.], 2020. p. 51–56.
- QIN, D.; LEICHNER, C.; DELAKIS, M.; FORNONI, M.; LUO, S.; YANG, F.; WANG, W.; BANBURY, C.; YE, C.; AKIN, B. et al. Mobilenetv4: universal models for the mobile ecosystem. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 2024. p. 78–96.
- RASCHKA, S.; MIRJALILI, V. *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2*. 3. ed. Birmingham, UK: Packt Publishing, 2019. ISBN 978-1789955750.
- RAWAT, W.; WANG, Z. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, MIT Press, v. 29, n. 9, p. 2352–2449, 2017.
- RIGUEIRA, X.; MARTINEZ, J.; ARAUJO, M.; RECAMAN, A. Computer vision application for improved product traceability in the granite manufacturing industry. *arXiv preprint arXiv:2207.01323*, 2022.
- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958.
- ROSENFELD, A.; TROY, E. *Visual Texture Analysis, No.* [S.l.], 1970.
- RUBY, U.; YENDAPALLI, V. et al. Binary cross entropy with deep learning technique for image classification. *Int. J. Adv. Trends Comput. Sci. Eng*, v. 9, n. 10, 2020.

SAZLI, M. H. A brief review of feed-forward neural networks. *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering*, Ankara University, v. 50, n. 01, 2006.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. *Neural networks*, Elsevier, v. 61, p. 85–117, 2015.

Seeed Studio. *Seeed Studio XIAO ESP32S3 Getting Started*. 2025. <[https://wiki.seeedstudio.com/xiao\\_esp32s3\\_getting\\_started/](https://wiki.seeedstudio.com/xiao_esp32s3_getting_started/)>. Acesso em: 26 maio 2025.

SHAMSHIRI, R. R.; KALANTARI, F.; TING, K. C.; THORP, K. R.; HAMEED, I. A.; WELTZIEN, C.; AHMAD, D.; SHAD, Z. M. Advances in greenhouse automation and controlled environment agriculture: A transition to plant factories and urban agriculture. *International Journal of Agricultural and Biological Engineering*, v. 11, n. 1, p. 1–22, 2018. <<https://doi.org/10.25165/j.ijabe.20181101.3210>>.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

SINGH, P. K.; NARESH, R.; KUMAR, L.; CHANDRA, M.; KUMAR, A. Role of iot technology in agriculture for reshaping the future of farming in india: a review. *Int. J. Curr. Microbiol. App. Sci*, v. 10, n. 02, p. 439–451, 2021.

SINGLA, N. Motion detection based on frame difference method. *International Journal of Information & Computation Technology*, v. 4, n. 15, p. 1559–1565, 2014.

SOOFI, A. A.; AWAN, A. Classification techniques in machine learning: applications and issues. *Journal of Basic & Applied Sciences*, v. 13, p. 459–465, 2017.

SOUZA, F.; BLAWID, S. Remote monitoring of plant water stress with rgb imaging. In: SBC. *Simpósio Brasileiro de Engenharia de Sistemas Computacionais (SBESC)*. [S.l.], 2024. p. 25–28.

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014.

SUCIU, G.; MARCU, I.; BALACEANU, C.; DOBREA, M.; BOTEZAT, E. Efficient iot system for precision agriculture. In: IEEE. *2019 15th International Conference on Engineering of Modern Electric Systems (EMES)*. [S.l.], 2019. p. 173–176.

SVOZIL, D.; KVASNICKA, V.; POSPICHAL, J. Introduction to multi-layer feed-forward neural networks. *Chemometrics and intelligent laboratory systems*, Elsevier, v. 39, n. 1, p. 43–62, 1997.

SZE, V.; CHEN, Y.-H.; YANG, T.-J.; EMER, J. S. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, IEEE, v. 105, n. 12, p. 2295–2329, 2017.

SZEGEDY, C.; LIU, W.; JIA, Y.; Sermanet, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCHE, V.; RABINOVICH, A. Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2015. p. 1–9.



SZELISKI, R. *Computer vision: algorithms and applications*. [S.l.]: Springer Nature, 2022.

THEODORIDIS, S. *Machine learning: a Bayesian and optimization perspective*. [S.l.]: Academic press, 2015.

TRAN, S.; ZAYTSEV, S.; CHARYKOVA, V.; YUSUPOVA, M.; BASHKATOV, A.; GENINA, E.; TUCHIN, V.; BLONDEL, W.; AMOUROUX, M. Analysis of image features for the characterization of skin optical clearing kinetics performed on in vivo and ex vivo human skin using linefield-confocal optical coherence tomography (lc-oct). In: *SPIE. Optics in Health Care and Biomedical Optics X*. [S.l.], 2020. v. 11553, p. 335–345.

VOULODIMOS, A.; DOULAMIS, N.; DOULAMIS, A.; PROTOPAPADAKIS, E. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, Wiley Online Library, v. 2018, n. 1, p. 7068349, 2018.

WAGH, T. A.; SAMANT, R.; GUJARATHI, S. V.; GAIKWAD, S. B. Grapes leaf disease detection using convolutional neural network. *Int. J. Comput. Appl*, v. 178, n. 20, p. 7–11, 2019.

WANG, F.; GAO, Q.; JI, G.; WANG, J.; DING, Y.; WANG, S. Effects of light intensity and photoperiod on morphological development and photosynthetic characteristics of coriander. *Horticulturae*, MDPI, v. 10, n. 3, p. 215, 2024.

WHEELER, W. D.; THOMAS, P.; IERSEL, M. v.; CHAPPELL, M. Implementation of Sensor-based Automated Irrigation in Commercial Floriculture Production: A Case Study. *HortTechnology*, v. 28, n. 6, p. 719–727, 2018. ISSN 1943-7714. <<https://doi.org/10.21273/horttech04114-18>>.

YAMASHITA, R.; NISHIO, M.; DO, R. K. G.; TOGASHI, K. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, Springer, v. 9, p. 611–629, 2018.

YU, L.; GAO, W.; SHAMSHIRI, R. R.; TAO, S.; REN, Y.; ZHANG, Y.; SU, G. Review of research progress on soil moisture sensor technology. *International Journal of Agricultural and Biological Engineering*, v. 14, n. 3, p. 32–42, 2021. ISSN 1934-6344.

ZAMANI, S. A.; BALEGHI, Y. A novel plant arrangement-based image augmentation method for crop, weed, and background segmentation in agricultural field images. *Computers and Electronics in Agriculture*, Elsevier, v. 233, p. 110151, 2025.

## ANEXO A – AQUISIÇÃO DE IMAGENS COM SENSOR 2D

O marco na tecnologia de detecção de imagem foi a invenção dos arranjos de fotodetectores semicondutores. dentre esses sensores o do tipo CCD (*Charge-Coupled Device*), bem como do tipo CMOS (*Complementary Metal-Oxide Semiconductor*). Tal sensor consiste em um grande número de elementos fotossensíveis. Durante a fase de acumulação, cada elemento coleta cargas elétricas, que são geradas pelos fótons absorvidos.

A Figura 81 (a) ilustra um elemento sensor sendo usado para transformar energia luminosa em imagem digital. A forma de onda da tensão de saída do sensor é análoga à energia luminosa incidente na entrada (Jähne (1995)).

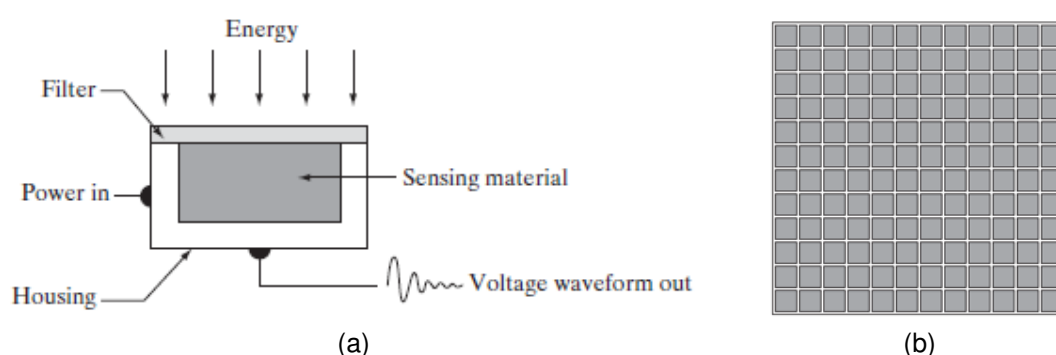


Figura 81 – Composição de sensor matricial.

A Figura 81 (b) mostra vários desses sensores dispostos na forma de um arranjo bidimensional. Este arranjo é predominantemente encontrado em câmeras digitais.

Uma imagem que incide na superfície do sensor bidimensional, conforme a Figura 82 ilustra, é contínua no espaço e em amplitude. Porém, para converter tal imagem em formato digital, é necessário amostrar a função tanto em coordenadas quanto em amplitude. A digitalização dos valores das coordenadas é chamada de amostragem, enquanto a digitalização dos valores de amplitude, quantização.

Portanto uma imagem resultante é definida como imagem digital e pode ser definida como uma função bidimensional,  $f(x,y)$ , onde  $x$  e  $y$  são coordenadas espaciais (planares) finitas, e a amplitude de  $f$ , também finita, em qualquer par de coordenadas  $(x, y)$ , é chamada de intensidade ou nível de cinza da imagem naquele ponto Gonzalez e Woods (2008). Ou seja, uma imagem digital é composta por um número finito de elementos, cada um com uma localização e valor específicos. Esses elementos são

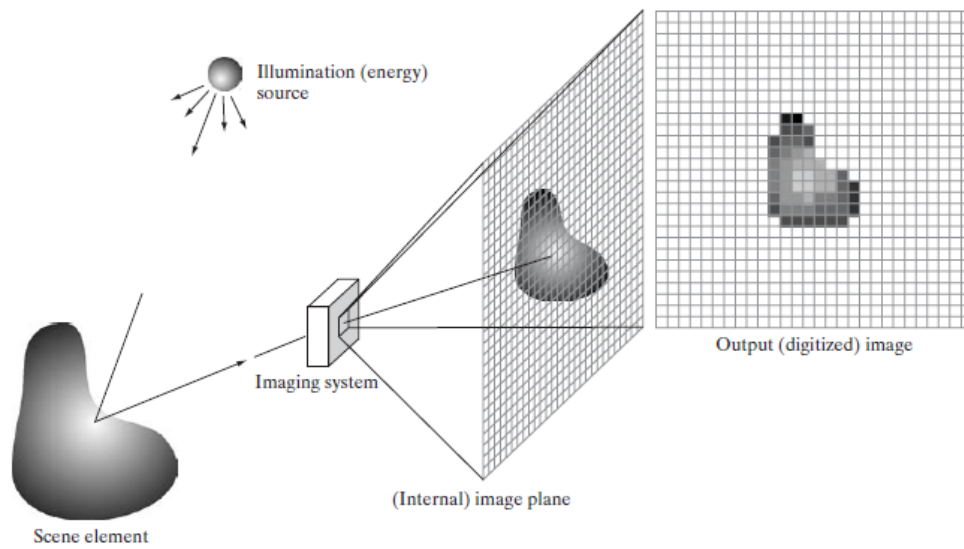


Figura 82 – Processo de Aquisição de imagem digital

chamados de elementos de imagem, pels ou pixels, o termo amplamente usado.

## ANEXO B – FUNDAMENTOS DA FILTRAGEM ESPACIAL

O domínio espacial é o plano que contém os pixels de uma imagem. Operações de domínio espacial computam diretamente nos pixels de uma imagem, entretanto, operações no domínio da frequência são realizadas na transformada de Fourier dessa imagem. Algumas tarefas de processamento de imagens são mais viáveis de implementar no domínio espacial, enquanto outras são mais adequadas para outras abordagens. Geralmente, as técnicas de domínio espacial são mais eficientes computacionalmente requerendo menos recursos de processamento para serem implementadas.

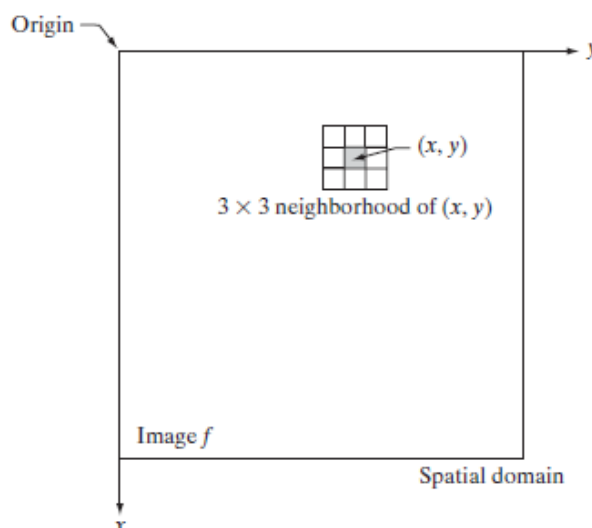


Figura 83 – Operação de filtragem diretamente na imagem

Os processos do domínio espacial podem ser denotados pela expressão

$$g(x, y) = f(x, y) * h(x, y) \quad (\text{B.1})$$

Onde  $f(x, y)$  é a imagem de entrada,  $h(x, y)$  é o filtro operando sobre uma vizinhança do ponto  $(x, y)$  na imagem  $f(x, y)$ , enquanto  $g(x, y)$  é a imagem resultante de saída.  $h(x, y)$ , geralmente muito menor do que a imagem  $f(x, y)$ , opera sobre a mesma pixel a pixel para um fim específico, como redução de ruído, por exemplo, ou mesmo erro de transmissão, conforme pontua Jähne (1995). A Figura 83 mostra a implementação básica da equação (B.1) sobre uma imagem. O ponto  $(x, y)$  corresponde tanto a um ponto arbitrário da imagem quanto ao centro da filtro, e a pequena região em destaque é a região da imagem sob o filtro. Normalmente,  $h(x, y)$  é quadrado, muito menor em tamanho do que a imagem, e de ordem ímpar – a fim de garantir kernel com centro a

ordem  $M \times N$  deve ser ímpar, ou seja, com dimensões  $M$  e  $N$  ímpares, o que simplifica a indexação e as operações.

O processo ilustrado pela Figura 83 consiste em mover o filtro  $h(x,y)$  sobre a imagem  $f(x,y)$ , de pixel a pixel computando seus coeficientes sobre os pixels da imagem de entrada para gerar um único pixel de saída. Resultando numa imagem de saída menor do que a imagem original de entrada. Toda a operação de convolução em (B.1), consiste num “deslizar” incremental do centro de  $h(x,y)$  sobre toda a imagem  $f(x,y)$ . Normalmente, o processo começa no canto superior esquerdo da imagem de entrada e prossegue pixel por pixel em uma varredura horizontal, uma linha de cada vez. Quando o centro de  $h(x,y)$  atinge a borda da imagem, parte da vizinhança residirá fora da imagem. Um artifício é preencher a imagem com uma borda de 0s ou de outros valores de intensidade específicos. A espessura da borda preenchida depende do tamanho de  $h(x,y)$ .

A respeito dessas operações entre o kernel sobre toda a imagem, vale distinguir o que a literatura designa como **correlação** e **convolução**.

Na **correlação** a máscara “desliza” pixel a pixel sobre a imagem, e em cada posição é computada a soma ponderada dos pixels, utilizando os valores da máscara como pesos.

A fórmula da correlação 2D é:

$$g(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k f(x + i, y + j) \cdot h(i, j) \quad (\text{B.2})$$

onde:

- $f(x, y)$  é o pixel da imagem original,
- $h(i, j)$  é a máscara (ou kernel),
- $g(x, y)$  é a imagem resultante.

*Nota: a máscara é aplicada como está, sem alterações.*

Na **convolução**, a operação é semelhante à correlação, mas com uma diferença essencial: a máscara é **rotacionada em 180°** (espelhada horizontal e verticalmente) antes da aplicação.

A fórmula da convolução 2D é:

$$g(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k f(x+i, y+j) \cdot h(-i, -j) \quad (\text{B.3})$$

Essa rotação influencia a orientação da aplicação do filtro, o que é importante em operações como detecção de bordas. Quando a máscara é simétrica, correlação e convolução produzem o mesmo resultado.

O termo *filtro* é emprestado do processamento no domínio da frequência, onde “filtragem” se refere à aceitação (passagem) ou rejeição de certos componentes de frequência. Por exemplo, um filtro que passa baixas frequências é chamado de filtro passa-baixa. O efeito líquido produzido por um filtro passa-baixa é borrar (suavizar) uma imagem atenuando variações bruscas de intensidade entre pixels vizinhos. Podemos realizar uma suavização semelhante diretamente na própria imagem usando filtros espaciais (também chamados de máscaras, núcleos ou *kernels*). Há uma correspondência biunívoca entre filtros espaciais lineares e filtros no domínio da frequência. No entanto, os filtros espaciais são mais versáteis pois podem ser usados também para filtragem não linear, algo que não é possível fazer no domínio da frequência.

## B.1 FILTROS ESPACIAIS PARA SUAVIZAÇÃO

Filtros de suavização são usados para desfoque e redução de ruído. O desfoque é usado em tarefas de pré-processamento, como a remoção de pequenos detalhes de uma imagem antes da extração de objetos (grandes). A redução de ruído pode ser obtida por desfoque com um filtro linear e também por filtragem não linear.

### B.1.1 Filtros Lineares

A saída (resposta) de um filtro espacial linear de suavização é simplesmente a média ponderada dos pixels abrangidos pela máscara do filtro, denominado como *filtro de média*, *filtro passa-baixa*, ou mesmo, *filtro box* (Jähne (1995)). Esse processo resulta em uma imagem cujas transições, outrora “nítidas” na imagem original, passam a ser atenuadas em intensidade. Como o ruído aleatório normalmente consiste em transições nítidas em níveis de intensidade, a aplicação mais óbvia da suavização é a redução desse ruído. No entanto, as bordas (que quase sempre são características

desejáveis de uma imagem) também são caracterizadas por transições de intensidade nítidas, portanto, os filtros de média têm o efeito colateral de borrar tais bordas. Outra aplicação desse tipo de processo inclui a suavização de falsos contornos, resultantes de número insuficiente de cores ou de níveis de intensidade. Portanto filtros de média reduzem detalhes “irrelevantes” em uma imagem, ou seja, regiões de pixels que são pequenas em relação ao tamanho da máscara do filtro. A Figura 84 mostra um filtro de suavização e seu resultado em média padrão da imagem - etapa que antecede uma limiarização, por exemplo.

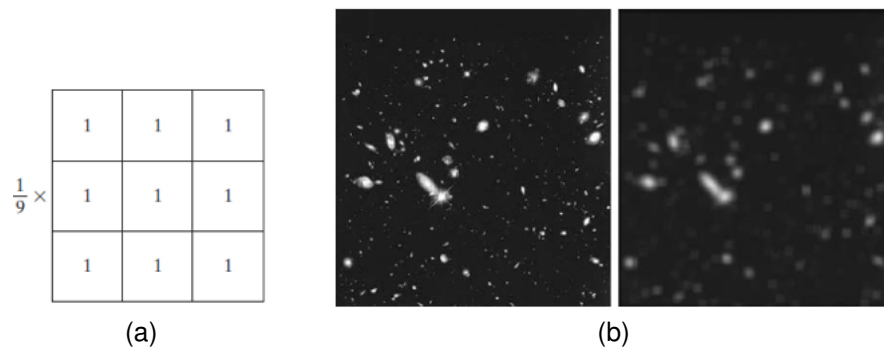


Figura 84 – (a) kernel  $3 \times 3$ . (b) Imagem original (esq.) e imagem borrada (dir.).

Além do *filtro média*, também conhecido como do *filtro box*, representado na Figura 84 (a), existem diversos outros filtros espaciais lineares aplicados à suavização, dentre eles, vale salientar um bastante sofisticado também utilizado neste trabalho, o *filtro gaussiano*, obtido com base na seguinte expressão gaussiana(B.4).

$$G(x, y; \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (\text{B.4})$$

A seguir, um exemplo de *kernel gaussiano* derivado da expressão anterior.

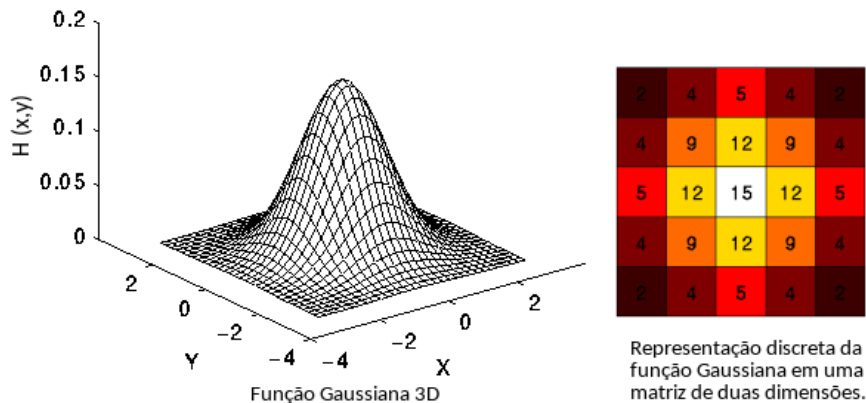


Figura 85 – Kernel Gaussiano para filtragem

O *filtro gaussiano* preserva contornos melhor do que o filtro de média simples.



Figura 86 – Imagem original half-tone e Imagem filtrada por filtro gaussiano, respectivamente.

#### B.1.1.1 Filtros de Ordem Estatística (Não Lineares)

Filtros lineares suprimiram efetivamente o ruído gaussiano, mas tiveram um desempenho muito ruim no caso de ruído binário, afirma Jähne (1995), neste caso, recomenda o uso de filtros não lineares. Filtros de ordem estatística são operações espaciais não lineares que, em cada posição, ordenam (rankeam) os valores dos pixels na vizinhança e substituem o valor do pixel central pelo elemento correspondente a uma determinada posição no ranking — por exemplo, a mediana, o máximo ou o mínimo.

O filtro de mediana, talvez o mais usado dessa classe, atribui ao pixel central o valor da mediana das intensidades na vizinhança (incluindo o próprio pixel). Essa estratégia preserva bordas melhor do que filtros lineares de mesma dimensão, sendo especialmente eficaz contra ruído do tipo *sal e pimenta* (impulse noise).

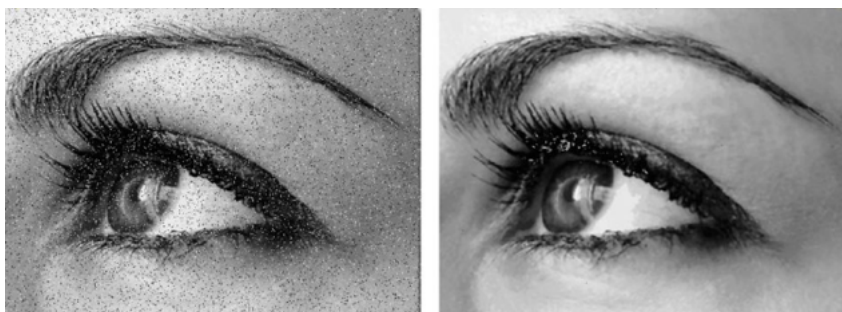


Figura 87 – Imagem original corrompida com "sal e pimenta" e imagem filtrada com mediana  $3 \times 3$ , respectivamente.



## Definição de Mediana

A mediana é uma medida de tendência central usada em estatística representando o valor central de um conjunto de dados organizados em ordem crescente (ou decrescente). Seja um conjunto de  $N$  valores ordenados:

$$z_{(1)} \leq z_{(2)} \leq \cdots \leq z_{(N)}.$$

$$\tilde{z} = \begin{cases} z_{(\lceil N/2 \rceil)}, & \text{se } N \text{ for ímpar,} \\ \frac{z_{(N/2)} + z_{(N/2+1)}}{2}, & \text{se } N \text{ for par.} \end{cases}$$

A mediana divide o conjunto de dados em duas partes iguais — metade dos valores fica abaixo dela e metade acima.

## Propriedades Principais

- **Redução de ruído impulsivo:** elimina agrupamentos de pixels isolados cuja área seja menor que metade da janela, forçando-os ao valor mediano dos vizinhos .
- **Preservação de bordas:** evita o borramento excessivo comum em filtros lineares.
- **Generalização:** a mediana é apenas o 50º percentil; o mesmo mecanismo gera outros filtros de ordem estatística, como o *max filter* (100º percentil) e o *min filter* (0º percentil).

Jähne (1995) acrescenta que, se apenas pixels individuais estiverem distorcidos numa imagem, um filtro mediano 3x3 é suficiente para eliminá-los. No caso de aglomerados de pixels distorcidos, filtros medianos maiores devem ser usados.

## ANEXO C – COMPRESSÃO DE IMAGENS

Imagens e vídeos gerados por dispositivos de hardware precisam ser armazenados ou transmitidos, mas o grande volume de bytes torna esse processo oneroso. Por isso, emprega-se a compressão de imagens — técnica que reduz o tamanho dos arquivos ao diminuir o número de bits necessários para representá-los, sem comprometer (ou com mínima perda de) qualidade visual, mantendo-a dentro de limites aceitáveis. Dessa forma, garante-se tanto o armazenamento eficiente quanto a rápida transmissão dos dados em redes.

A compressão de dados consiste em reduzir o número de bits necessário para representar uma mesma informação, eliminando redundâncias sem perda (*loss/less*) ou descartando informação menos perceptível (*lossy*) para obter taxas maiores de redução.

Sejam duas representações da mesma informação com  $b$  e  $b'$  bits, sendo  $b$  a informação original e  $b'$  a informação comprimida. Define-se a *razão de compressão*

$$C = \frac{b}{b'}$$

e a *redundância relativa*

$$R = 1 - \frac{1}{C}$$

Assumindo,  $C = 10$  ("10:1") implica  $R = 0,9$ , ou seja, 90% dos bits originais eram redundantes.

Em geral, as imagens apresentam três tipos de informação redundante:

1. **Codificação:** uso ineficiente de bits por pixel (ex. valores de 8 bits usam mais símbolos do que o necessário);
2. **Espacial/Temporal:** correlação entre pixels vizinhos em imagens estáticas ou entre quadros em vídeo;
3. **Irrelevante:** detalhes que o sistema visual humano não percebe, removíveis sem degradar a qualidade aparente.

Shannon, em sua teoria da informação, estabeleceu os princípios que contribuem na quantificação da informação presente em uma imagem, possibilitando, por exem-

plo, a determinação do número mínimo de bits por pixel, necessário numa imagem, sem degradá-la.

A remoção de “informação visual irrelevante” implica perda de dados quantitativos da imagem. Para quantificar essa perda, utilizam-se dois tipos de critérios de fidelidade na compressão de imagens:

- **Fidelidade objetiva:** quando a degradação é descrita por uma expressão matemática em função da imagem de entrada e de saída do processo de compressão. Um exemplo comum é o *erro quadrático médio* ( $e_{rms}$ ) entre as imagens.
- **Fidelidade subjetiva:** baseada na percepção humana, avalia-se a qualidade visual por meio de ranqueamento que reflita o nível de aceitabilidade da imagem descomprimida.

Esses critérios permitem controlar o trade-off entre taxa de compressão e qualidade visual, garantindo medições objetivas e avaliações alinhadas à experiência do observador.

Dentre os padrões de compressão de imagem mais populares, está o JPEG – Joint Photographic Experts Group. Um sistema de codificação do tipo Lossy que adota os seguintes métodos de compressão: Transformada Discreta do Cosseno (DCT) em blocos de imagem  $8 \times 8$ , Huffman e Run-Length.

No processo inverso, descompressão de imagens – para exibição, por exemplo –, métodos complementares aos utilizados durante a codificação, são aplicados a fim de decodificar os arquivos outrora comprimidos. Ambos os processos, compressão e descompressão, podem ser implementados tanto em software quanto em hardware, o que eleva o desempenho.

## ANEXO D – SISTEMAS DE COR

### D.0.1 Sistemas de Cor

Um sistema, modelo ou espaço de cor define um sistema de coordenadas onde cada cor é representada por um ponto no espaço, permitindo especificar cores de forma padronizada e consistente. Na prática de PDI, adotam-se modelos orientados a *hardware* - RGB, por exemplo, para monitores - e modelos orientados à percepção humana - como o HSV (do inglês Hue, Saturation e Value), que desacopla a informação de cor da intensidade. Embora existam muitos outros sistemas de cores, esses são, de fato, utilizados durante o desenvolvimento.

#### D.0.1.1 O Sistema RGB

No espaço RGB, cada cor é formada pela combinação de três cores básicas - vermelho, verde e azul — num sistema de coordenadas cartesiano. O espaço de cor, Fig. 88, corresponde a um cubo unitário (valores de R, G e B normalizados em  $[0,1]$ ): seus vértices abrigam as cores primárias (vermelho, verde e azul), as secundárias (ciano, magenta e amarelo), o preto (origem) e o branco (vértice oposto). A escala de cinza aparece ao longo da diagonal que liga o preto ao branco, e qualquer cor é representada por um ponto dentro desse cubo, definido pelo vetor que o une à origem.

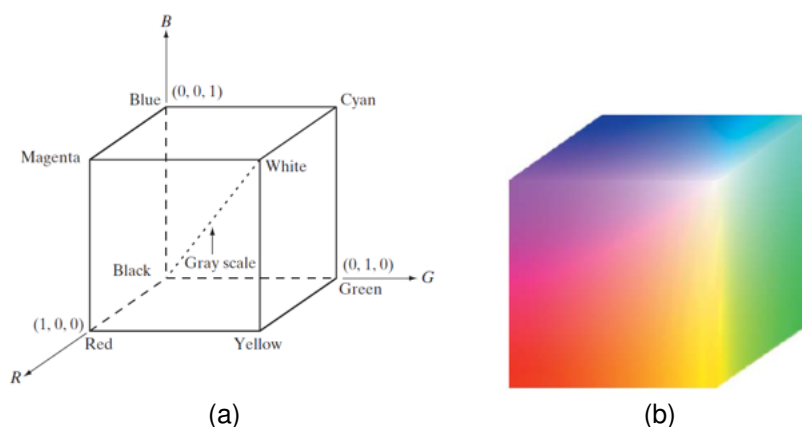


Figura 88 – (a) Cubo de cores RGB normalizado (b) Cubo de cores RGB 24 bits. Fonte: Gonzalez e Woods (2008).

Quando uma imagem é exibida em um monitor RGB, os seus três componentes, R, G e B, são combinados na tela para produzir uma imagem colorida.

O número de bits usados para representar cada pixel no espaço RGB é chamado de *profundidade de pixel*. Em geral, cada um dos componentes R, G e B é codificado em 8 bits. Nessa situação, cada pixel RGB, ou seja, cada tripla  $(R, G, B)$ , possui profundidade de:

$$\text{Profundidade} = 3 \times 8 = 24 \text{ bits}$$

Imagens com essa profundidade são frequentemente denominadas *full-color*. O número total de cores distintas em uma imagem RGB de 24 bits é:

$$2^{24} = 16\,777\,216$$

A Fig. 88 (b) ilustra o cubo de cores RGB de 24 bits, no qual cada eixo varia de 0 a 255 e cada ponto no interior do cubo representa uma cor possível.

Diante das limitações do modelo RGB em representar a percepção humana das cores, a próxima seção apresenta o sistema HSV, que organiza as cores de forma mais alinhada à forma como as pessoas as identificam e descrevem.

#### D.0.1.2 O Sistema HSV

Sistemas de cor como *RGB* e *CMY* (*CMYK*) são eficientes para *hardware* — câmeras, monitores e impressoras — e permitem conversão direta entre si. Contudo, eles não refletem a forma como o ser humano lida com as cores no dia a dia: não se refere às cores em percentuais de cores primárias, mas em *matiz*, *saturação* e *brilho*.

Observando a mesma Fig. 88 (a), centralizando o vértice “branco” do cubo RGB, é possível redesenhá-lo como sugere a Fig. 89 (a), e sua variante(b).

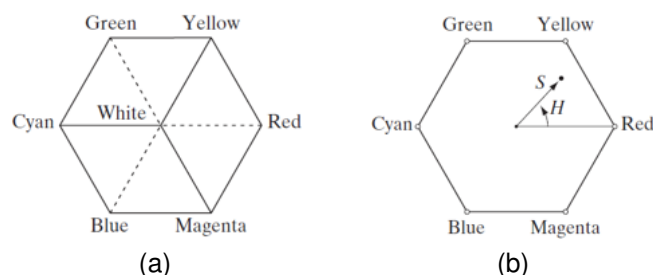


Figura 89 – (a) Cubo de cores RGB noutra perspectiva. (b) Cubo de cores RGB com perfil hexagonal. Fonte: Gonzalez e Woods (2008).

A partir dessa perspectiva o modelo HSV foi proposto para o desacoplamento das três informações: cor, saturação e brilho.

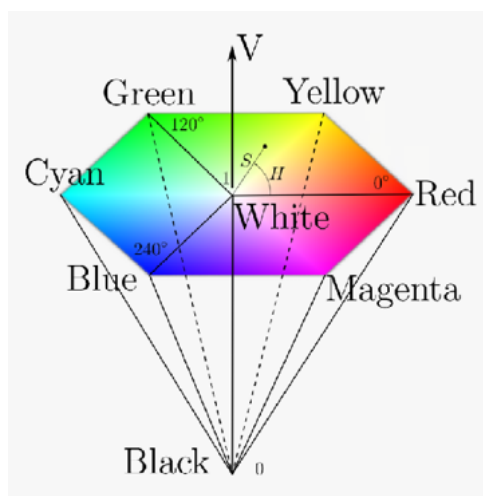


Figura 90 – Espaço de cores HSV normalizado. Fonte: Rigueira et al. (2022).

- **Matiz (H):** O ângulo corresponde à cor e varia entre  $0^\circ$  e  $360^\circ$ ;
- **Saturação (S):** O comprimento do vetor é a saturação e varia de 0 a 1, indicando o percentual de pureza da cor;
- **Luminância (V):** A intensidade de todas as cores no eixo vertical, variando de 0 (preto) a 1 (branco).

Na Fig. 90, qualquer ponto nesse espaço, a posição angular determina  $H$ , o comprimento radial determina  $S$ , e a altura determina  $V$ .

Esse desacoplamento do sistema HSV o torna particularmente adequado ao contexto deste trabalho, sendo especialmente útil para a segmentação baseada na matiz verde (relacionada à planta) e apresentando independência em relação à iluminação. Como já mencionado, as imagens são originalmente obtidas no sistema de cor RGB; entretanto, a fim de auxiliar a etapa de segmentação, é comum a conversão entre diferentes sistemas de cor. A seguir, apresentam-se as equações que regem tais transformações.

- **Conversão de RGB para HSV:**

$$R' = \frac{R}{255}, \quad G' = \frac{G}{255}, \quad B' = \frac{B}{255} \quad (\text{normalização dos canais})$$

$$M = \max(R', G', B')$$

$$m = \min(R', G', B')$$

$$\Delta = M - m$$

$$H = \begin{cases} 0^\circ & , \text{ se } \Delta = 0 \\ 60^\circ \times \left( \frac{G' - B'}{\Delta} \right) & , \text{ se } M = R' \\ 60^\circ \times \left( \frac{B' - R'}{\Delta} + 2 \right) & , \text{ se } M = G' \\ 60^\circ \times \left( \frac{R' - G'}{\Delta} + 4 \right) & , \text{ se } M = B' \end{cases} \quad S = \begin{cases} 0 & , \text{ se } M = 0 \\ \frac{\Delta}{M} & , \text{ caso contrário.} \end{cases} \quad V = M$$

• **Conversão de HSV para RGB:**

$$C = V \cdot S, \quad X = C \cdot \left( 1 - \left| \left( \frac{H}{60^\circ} \bmod 2 \right) - 1 \right| \right), \quad m = V - C$$

Seja  $(R', G', B')$  definido por:

$$(R', G', B') = \begin{cases} (C, X, 0) & \text{se } 0^\circ \leq H < 60^\circ \\ (X, C, 0) & \text{se } 60^\circ \leq H < 120^\circ \\ (0, C, X) & \text{se } 120^\circ \leq H < 180^\circ \\ (0, X, C) & \text{se } 180^\circ \leq H < 240^\circ \\ (X, 0, C) & \text{se } 240^\circ \leq H < 300^\circ \\ (C, 0, X) & \text{se } 300^\circ \leq H < 360^\circ \end{cases}$$

E, finalmente:

$$R = (R' + m) \cdot 255$$

$$G = (G' + m) \cdot 255$$

$$B = (B' + m) \cdot 255$$

Após a conversão para o sistema HSV, a segmentação torna-se mais eficiente, pois isola regiões de interesse de maneira mais robusta frente a variações de iluminação. A seguir, são apresentados os principais conceitos e métodos empregados no processo de segmentação de imagens, com destaque para a limiarização e a segmentação por cor.

## ANEXO E – SEGMENTAÇÃO

Segmentar uma imagem significa particioná-la em regiões homogêneas, de modo a isolar um objeto ou área de interesse do fundo e de outras estruturas, facilitando sua análise e interpretação. Dentre diversas técnicas de segmentação de imagens, se destaca a limiarização. Devido à sua intuição simples, facilidade de implementação e baixo custo computacional, a limiarização ocupa posição central em segmentação de imagens (Gonzalez e Woods (2008)). Uma de suas aplicações é a segmentação de imagens em níveis de cinza, resultando em imagens binárias - preto e branco.

Suponha o histograma da Fig. 91, dispondo de duas regiões, ou modos, se deduz que um dos modos represente o objeto de interesse, mais claro, e o outro modo, o fundo escuro, e um limiar  $T$ , em um “vale”, separando essas duas regiões.

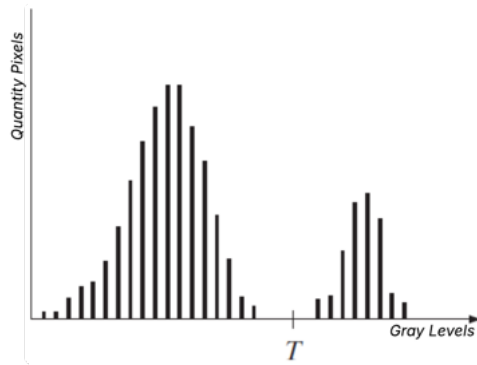


Figura 91 – Histograma de imagem com um único limiar em  $T$ . Fonte: Gonzalez e Woods (2008).

Quando um histograma, idêntico ao da Fig. 91, possui dois “modos” separados por um vale bem definido entre os mesmos, é dito haver um valor de intensidade,  $T$ , único para toda a imagem, e o método para sua determinação é chamado de *limiarização global*. Definido assim, a segmentação binária  $g(x, y)$  é, então, expressa pela seguinte equação (E.1):

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

Em outros casos, onde imagens apresentem mais de um limiar para a extração de objetos, é dito *limiarização múltipla*, enquanto, imagens com iluminação não uniforme ou mesmo presença de ruído, sua técnica se denomina como *limiarização variável*.



No contexto da limiarização global, quando as distribuições de intensidade dos pixels do objeto de interesse e do fundo são bem distintas, pode-se empregar um algoritmo simples e intuitivo: iterar sobre o histograma da imagem para estimar a intensidade média entre os dois picos, que atua como limiar de separação entre as duas regiões.

Entretanto, é possível desenvolver um procedimento mais eficiente expressando todos os cálculos diretamente em termos do histograma da imagem, o qual precisa ser computado apenas uma única vez.

O método de Otsu (Otsu (1979)) define o limiar ótimo como aquele que maximiza a variância entre classes,  $\sigma_B^2$ , no histograma da imagem. Em outras palavras, ele busca o valor de  $T$  que torna as distribuições de intensidade de fundo e objeto o mais distintas possível. Tal método consiste em assumir duas funções de densidade de probabilidade, iterando em busca de valor de limiar  $T$  entre 0 e 255 - no caso de tons de cinza - tal que  $T$  maximize a variância entre classes.

Esse método oferece uma formulação baseada unicamente em estatísticas do histograma (variância de cada classe), o que o torna um algoritmo eficiente, não requerendo processamento direto dos pixels após o cálculo do histograma.

$$\sigma_B^2(T) = P_1(T)(m_1(T) - m_G)^2 + P_2(T)(m_2(T) - m_G)^2$$

onde

- $m(T)$  é a média acumulada até o nível  $T$ ,
- $m_G$  é a média geral dos pixels da imagem,
- para uma quantidade  $n_i$  de pixels para um determinado nível  $i$  de cinza,  $P_1(k)$  é dado pela soma cumulativa

$$P_1(k) = \sum_{i=0}^k p_i = \sum_{i=0}^k n_i / MN \quad (\text{E.2})$$

A maximização de  $\sigma_B^2(T)$  sobre todos os possíveis  $T$  fornece o limiar de Otsu. As Figs. 92 e 93 ilustram um exemplo.

A imagem a ser segmentada nem sempre está livre de artefatos que prejudicam o limiar. Quando o ruído não pode ser eliminado na fonte, uma prática comum é suavizar a imagem com um filtro passa-baixa antes de aplicar o limiar, reduzindo variações

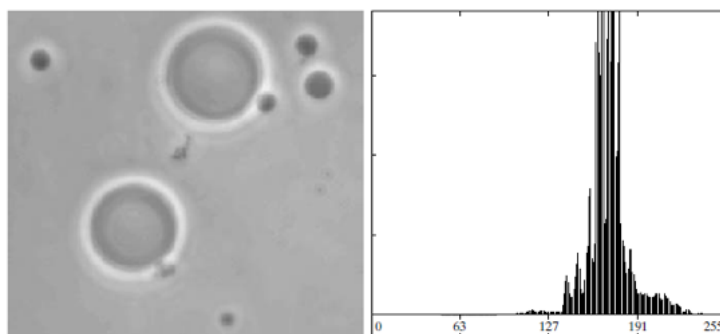


Figura 92 – Uma imagem de exemplo real e seu histograma. (Esq.) Imagem original e (dir.) seu histograma. Fonte: Gonzalez e Woods (2008).



Figura 93 – Segmentação de imagens baseada em dois algoritmos diferentes. (Esq.) Segmentação usando algoritmo global básico e (dir.) segmentação usando método de Otsu. Fonte: Gonzalez e Woods (2008).

indesejadas. Como alternativa, pode-se realçar o contorno da região de interesse filtrando a imagem com um filtro do tipo passa-alta para evidenciar os níveis de cinza das bordas, e então usar esses valores para definir um limiar mais preciso.

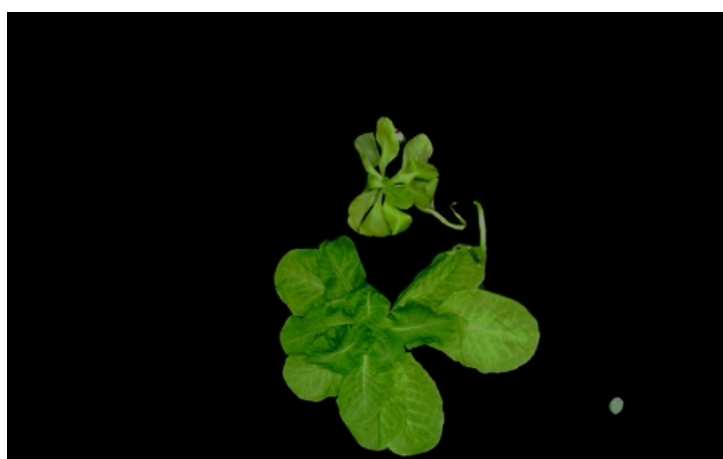


Figura 94 – Exemplo de Segmentação em cascata: por Limiarização e Cor. Fonte: O autor.

Uma outra abordagem na segmentação de imagens é a segmentação por cor, conforme podemos verificar na Fig. 94, onde o plano de fundo foi retirado da imagem. Para segmentar por cor, é mais viável trabalhar com a imagem em espaço de cor

desacoplada da intensidade, como o HSV, por exemplo, onde a matiz (H) oferece uma representação direta das cores. A partir disso é possível isolar o intervalo de cores desejado. As demais componentes S e V não carregam informação de cor, portanto, dispensáveis na segmentação de cor.

## ANEXO F – OPERAÇÕES ARITMÉTICAS

As operações aritméticas entre imagens são operações de matriz, o que significa que as operações aritméticas são realizadas entre pares de pixels correspondentes (GONZALEZ; WOODS, 2008). As quatro operações são denotadas em (F.1):

$$\begin{aligned}
 s(x, y) &= f(x, y) + g(x, y) \\
 d(x, y) &= f(x, y) - g(x, y) \\
 p(x, y) &= f(x, y) \times g(x, y) \\
 v(x, y) &= f(x, y) \div g(x, y)
 \end{aligned}
 \tag{F.1}$$

Entende-se que as operações são realizadas entre pares de pixels correspondentes em  $f$  and  $g$  for  $x = 0, 1, 2, \dots, M - 1$  and  $y = 0, 1, 2, \dots, N - 1$  onde  $M$  e  $N$  são os tamanhos das linhas e colunas das imagens. Claramente,  $s$ ,  $d$ ,  $p$  e  $t$  também são imagens de tamanho  $M \times N$ .

## ANEXO G – PROCESSAMENTO MORFOLÓGICO

De acordo com Pedrini e Schwartz (2008), a introdução ao estudo da morfologia matemática se deu a partir dos estudos de Haas et al. (1967), uma área originalmente desenvolvida para manipular imagens binárias, posteriormente estendida para tratar imagens em níveis de cinza. Suas aplicações são variadas e consistem em extração de componentes na imagem, busca por padrões específicos, afinamento ou dilatação de bordas e muitas outras. Jähne (1995) a reforça como ferramenta útil na análise de formas de objetos numa imagem.

A morfologia matemática se baseia na teoria dos conjuntos para representar a forma dos objetos em uma imagem. Em geral, compreende operações lógicas da álgebra Booleana entre dois conjuntos, neste caso, a imagem binária  $A$  como uma matriz de conjunto de pixels e  $B$ , o *elemento estruturante*, ou seja, uma *máscara* de pixels não-nula, comumente,  $B$  é um conjunto muito menor do que  $A$ , sendo  $A$  e  $B \subset \mathbb{Z}^2$ . A fim de manter a coerência com a abordagem aplicada neste presente trabalho, será destacada a operação morfológica aqui empregada, a Erosão, e em seguida sua operação complementar, a Dilatação, descritas a seguir:

- **Erosão:**

$$A \ominus B = \{p \in \mathbb{Z}^2 \mid B_p \subseteq A\},$$

Com  $B_p$  denotando a máscara deslocada com seu ponto de referência sobre o pixel  $p$ . A erosão do conjunto de pixels  $A$  pelo conjunto de pixels  $B$  é o conjunto de todos os pixels  $p$  para os quais  $B$  está completamente contido em  $A$  (Jähne (1995)). A erosão pode ser vista como uma filtragem morfológica, onde detalhes na imagem menores do que o elemento estruturante são filtrados, removidos da imagem. Em resumo, se trata de uma operação que reduz objetos e remove pixels isolados em  $A$ .

- **Dilatação:**

$$A \oplus B = \{z \in \mathbb{Z}^2 \mid B_p \cap A \neq \emptyset\},$$

A dilatação de  $A$  por  $B$  é o conjunto de todos os pixels para os quais a interseção entre  $A$  e  $B_p$  não é um conjunto vazio. Em síntese, uma operação que aumenta objetos e preenche pequenas regiões nulas.

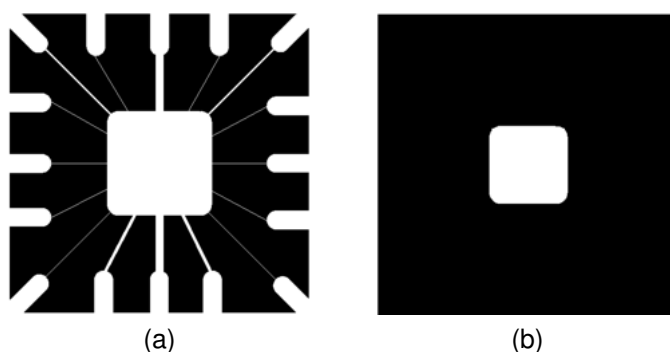


Figura 95 – Erosão para remoção de componentes. (a) Imagem original  $486 \times 486$ . (b) Imagem erodida por elemento estruturante  $45 \times 45$ , com todos seus elementos 1s.

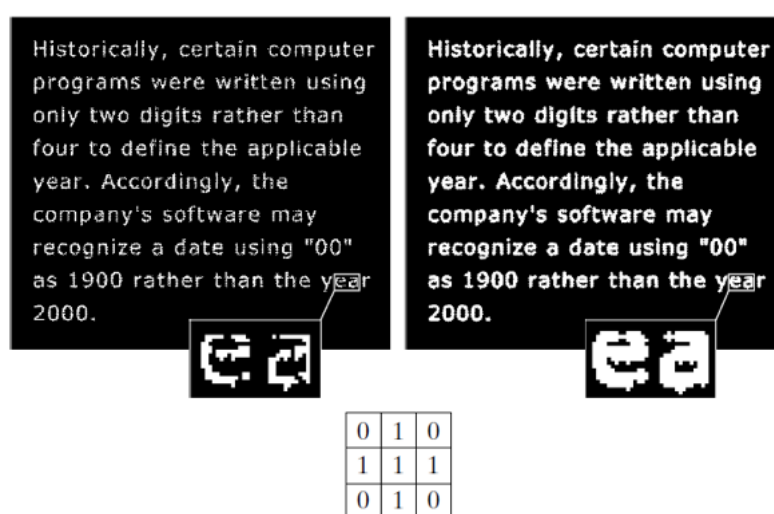


Figura 96 – (Esq.) Texto de exemplo com caracteres quebrados. (Dir.) Imagem resultante da dilatação. (Abaixo) Elemento estruturante.

Muitos outros algoritmos morfológicos são derivados dessas duas operações primitivas.

## PRINCIPAIS APLICAÇÕES

### 1. Extração de contornos (Boundary Extraction)

Obtém as bordas de um objeto pela diferença entre o conjunto original e sua erosão.

### 2. Preenchimento de regiões (Region Filling)

Preenche áreas conectadas a partir de um ponto-semente usando dilatações condicionais.

### 3. Extração de componentes conectados (Connected-Component Extraction)

Identifica cada objeto isolado em uma imagem binária por dilatação iterativa a partir de pixels-semente.

**4. Envoltória convexa (Convex Hull)**

Gera o menor conjunto convexo que contém um objeto, aplicando sucessivas erosões e dilatações.

**5. Afinamento (Thinning)**

Remove recortes de borda sem quebrar a conectividade, reduzindo objetos à sua “linha central”.

**6. Espessamento (Thickening)**

Operação dual ao afinamento, que amplia estruturas finas “engrossando” objetos.

**7. Esqueletonização (Skeletonization)**

Extrai a representação medial de uma forma, produzindo seu esqueleto topológico.

**8. Poda (Pruning)**

Remove ramificações indesejadas do esqueleto, refinando a estrutura extraída.