

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

Matheus Willian Dias Civardi

**Construção de um *Dataset* de Imagens de Soja e Algodão
com Amostragem Limitada e Aumento de Dados via
Sistema Embarcado**

São Carlos

2025

Matheus Willian Dias Civardi

**Construção de um *Dataset* de Imagens de Soja e Algodão
com Amostragem Limitada e Aumento de Dados via
Sistema Embarcado**

Monografia apresentada ao Curso de Engenharia Mecatrônica, da Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro Mecatrônico.

Orientador: Prof. Dr. Marcelo Becker

**São Carlos
2025**

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Ficha catalográfica elaborada pela Biblioteca Prof. Sérgio Rodrigues Fontes
e pelo Serviço de Comunicação e Marketing da EESC-USP,
com dados inseridos pelo(a) autor(a).

C582c Civardi, Matheus Willian Dias

 Construção de um Dataset de Imagens de Soja e Algodão
com Amostragem Limitada e Aumento de Dados via Sistema
Embarcado / Matheus Willian Dias Civardi ; orientador
Marcelo Becker. -- São Carlos, 2025.

 71 p.

 Monografia - Graduação em Engenharia Mecatrônica --
Escola de Engenharia de São Carlos da Universidade de São
Paulo, 2025.

 1. Dataset. 2. Aumento de Dados. 3. Sistema
Embarcado. 4. Agricultura de Precisão. I. Becker,
Marcelo, orient. II. Título.

Responsáveis pela estrutura de catalogação da publicação segundo a AACR2: Bibliotecários da EESC/USP.

FOLHA DE AVALIAÇÃO

Candidato: Matheus Willian Dias Civardi

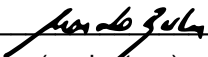
Título: Construção de um *Dataset* de Imagens de Soja e Algodão com Amostragem Limitada e Aumento de Dados via Sistema Embarcado

Trabalho de Conclusão de Curso apresentado à
Escola de Engenharia de São Carlos da
Universidade de São Paulo
Curso de Engenharia Mecatrônica.

BANCA EXAMINADORA

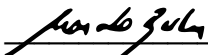
Professor Dr. Marcelo Becker
(Orientador)

Nota atribuída: 8,0 (Oito)


(assinatura)

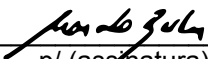
Professor Dr. Daniel Varela Magalhães

Nota atribuída: 8,0 (Oito)


p/ (assinatura)

Thiago Henrique Segreto Silva

Nota atribuída: 8,0 (Oito)


p/ (assinatura)

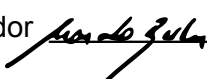
Média: 8,0 (Oito)

Resultado: APROVADO

Data: 09/02/2026.

Este trabalho tem condições de ser hospedado no Portal Digital da Biblioteca da
EESC

SIM X NÃO ☐ Visto do orientador



*Este trabalho é dedicado aos meus pais, pelo suporte ao longo de minha vida,
e aos funcionários da Universidade de São Paulo (USP), em especial à equipe IMAmt,
pela contribuição e assistência durante a elaboração deste TCC.*

AGRADECIMENTOS

Agradeço aos meus pais, Jacira Fabiana Dias Civardi e Henrique Civardi, pelo apoio ao longo da minha formação.

Aos meus irmãos, Paulo Henrique Dias Civardi e Lucas Thiago Dias Civardi, pelo incentivo.

Ao meu orientador, Marcelo Becker, pelo acompanhamento e orientação durante este trabalho.

À equipe IMAmt, em especial ao Paulo Polegato, que acompanhou as minhas atividades e disponibilizou materiais e orientações essenciais para o desenvolvimento deste trabalho.

Aos demais funcionários da USP, que contribuem para a manutenção de um ambiente acadêmico agradável e acolhedor.

*“Escolha sempre o caminho que pareça melhor,
mesmo que seja o mais difícil. O hábito
brevemente o tornará fácil e agradável.”*

Pitágoras de Samos

RESUMO

CIVARDI, M.W.D. **Construção de um *Dataset* de Imagens de Soja e Algodão com Amostragem Limitada e Aumento de Dados via Sistema Embarcado**. 2025. 69 p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2025.

Este trabalho tem como objetivo contribuir com a teoria e a prática na construção de *datasets* de imagens voltados à identificação de plantas, abrangendo as etapas de coleta, armazenamento, rotulagem e gerenciamento das imagens. Para isso, foi cultivada uma cultura controlada com um número reduzido de plantas de soja e algodão, para aquisição sistemática de imagens. As fotografias das folhas foram obtidas por meio de câmeras de telefone celular, e dois sistemas embarcados, baseados no microcomputador Raspberry Pi 3 Model B+, foram desenvolvidos para automatizar a coleta de dados: um utilizando a câmera modular V2.1 da Raspberry, e outro utilizando uma câmera webcam Logitech C920.

Foram implementados algoritmos para automatizar a captura, classificação e rotulagem das imagens, garantindo uma organização eficiente do *dataset* e facilitando a manipulação dos dados. Além disso, desenvolveram-se rotinas de aumento de dados (*data augmentation*), com o intuito de gerar versões das imagens originais contendo variações — como rotações, espelhamentos e ajustes de brilho — preservando, contudo, as características essenciais das plantas. Essa abordagem contribui para a ampliação da diversidade do conjunto de dados, fator essencial para reduzir o problema de *overfitting*, recorrente em conjuntos com amostragem limitada.

Como resultado, foi obtido um *dataset* estruturado e diversificado, adequado para aplicações de visão computacional e treinamento de modelos de aprendizado de máquina voltados à classificação ou detecção de espécies vegetais. O estudo reforça a viabilidade da utilização de sistemas embarcados de baixo custo para coleta e processamento de imagens no contexto da agricultura de precisão.

Palavras-chave: *Dataset*. Aumento de Dados. Sistema Embarcado. Agricultura de Precisão.

ABSTRACT

CIVARDI, M.W.D. **Construction of a Soybean and Cotton Image Dataset with Limited Sampling and Data Augmentation via Embedded System.** 2025. 69 p. Monograph (Conclusion Course Paper) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2025.

This work aims to contribute to the theoretical and practical aspects of building image datasets for plant identification, encompassing the stages of image collection, storage, labeling, and management. To this end, a controlled environment was established with a limited number of soybean and cotton plants for the systematic acquisition of images. Photographs of the leaves were captured using smartphone cameras, and two embedded systems, based on the Raspberry Pi 3 Model B+ microcomputer, were developed to automate data collection: one using the Raspberry Pi Camera Module V2.1 and the other using a Logitech C920 webcam.

Algorithms were implemented to automate the processes of image capture, classification, and labeling, ensuring efficient dataset organization and facilitating data handling. In addition, data augmentation routines were developed to generate versions of the original images containing variations—such as rotations, mirroring, and brightness adjustments—while preserving the essential characteristics of the plants. This approach contributes to increasing the diversity of the dataset, which is a key factor in mitigating the issue of overfitting, especially in datasets with limited sampling.

As a result, a structured and diversified dataset was obtained, suitable for computer vision applications and the training of machine learning models focused on plant species classification or detection. The study reinforces the feasibility of using low-cost embedded systems for image collection and processing within the context of precision agriculture.

Keywords: Dataset. Data Augmentation. Embedded System. Precision Agriculture.

LISTA DE FIGURAS

Figura 1 – Aplicações de <i>Machine Learning</i>	28
Figura 2 – Exemplo de uma estrutura de <i>deep learning</i>	29
Figura 3 – Aplicações de <i>Deep Learning</i>	29
Figura 4 – Diagrama dos campos de atuação	30
Figura 5 – Pipeline da arquitetura geral da CNN	31
Figura 6 – Modelos de Raspberry	34
Figura 7 – Aplicações do Raspberry Pi na agricultura	35
Figura 8 – Sistema com Raspberry Pi para monitoramento de culturas vegetais . .	35
Figura 9 – <i>Deep learning</i> implementado no Raspberry Pi	36
Figura 10 – Coleta de dados no ambiente industrial	37
Figura 11 – Panorama dos tópicos da coleta de dados	38
Figura 12 – Técnicas de <i>data augmentation</i>	42
Figura 13 – Estufa com ambiente controlado	45
Figura 14 – Mudanças transplantadas	46
Figura 15 – Plantas em etapa inicial de crescimento	47
Figura 16 – Plantas em etapa final de crescimento	47
Figura 17 – Exemplo da estrutura do código do nome	48
Figura 18 – Exemplos de imagens salvas	49
Figura 19 – Especificações do Raspberry Pi 3 Model B+	51
Figura 20 – Pinout do Raspberry Pi 3 Model B+	52
Figura 21 – Protótipo com webcam C920	53
Figura 22 – Visão explodida do sistema	54
Figura 23 – CAD do sistema montado	55
Figura 24 – Raspberry Pi 3 Modelo B+ com câmera modular v2.1	56
Figura 25 – Case adaptada para câmera modular	56
Figura 26 – Protótipo utilizando Webcam C920	57
Figura 27 – Protótipo utilizando câmera modular	58
Figura 28 – Drive com imagens do banco de dados	58
Figura 29 – Dicionário de transformações utilizadas no processo de <i>data augmentation</i> . 67	
Figura 30 – Trecho do código dedicado à seleção do projeto, versão, cultura e câmera 69	

LISTA DE TABELAS

Tabela 1 – Parâmetros de avaliação de performance de modelos <i>deep learning</i> de classificação	32
Tabela 2 – Exemplos de desempenho de CNNs relacionados à quantidade de imagens no <i>dataset</i>	33
Tabela 3 – Comparação entre <i>Small data</i> e <i>Big data</i>	39
Tabela 4 – Manipulações básicas de imagens	42

LISTA DE QUADROS

Quadro 1 – Questões analisadas por Kamilaris e Prenafeta-Boldú (2018)	32
Quadro 2 – Desafios na construção de dados anotados	40
Quadro 3 – Desafios na validação dos dados rotulados	41

LISTA DE ABREVIATURAS E SIGLAS

CNN	<i>Convolutional Neural Network</i>
DL	<i>Deep Learning</i>
FN	<i>False Negative</i>
FP	<i>False Positive</i>
FSL	<i>Few-Shot Learning</i>
GAN	<i>Generative Adversarial Networks</i>
IA	Inteligência Artificial
IMAmt	Instituto Mato-Grossense do Algodão
IoT	<i>Internet of Things</i>
ML	<i>Machine Learning</i>
SSL	<i>Small Sample Learning</i>
SBC	<i>Single-Board Computer</i>
TP	<i>True Positive</i>

SUMÁRIO

1	INTRODUÇÃO	25
1.1	Motivação	25
1.2	Objetivos	26
1.3	Organização	26
2	REVISÃO BIBLIOGRÁFICA	27
2.1	Conceitos de Machine Learning e Deep Learning	27
2.1.1	<i>Machine Learning</i>	27
2.1.2	<i>Deep Learning</i>	28
2.1.3	<i>Visão computacional</i>	30
2.1.4	<i>Deep learning</i> aplicado a visão computacional	30
2.1.4.1	Redes neurais convolucionais	30
2.1.4.2	Aplicações na agricultura	31
2.1.4.3	<i>Few Shot Learning e Transfer learning</i>	33
2.2	IoT e Sistemas Embarcados na agricultura	33
2.2.1	Raspberry Pi na agricultura	33
2.2.2	Raspberry Pi e Machine Learning	36
2.3	Construção de <i>datasets</i>, rotulagem e <i>data augmentation</i>	37
2.3.1	Dados para aprendizado de máquinas	38
2.3.1.1	<i>Small data</i>	39
2.3.2	<i>Data labeling</i>	40
2.3.2.1	Construção de <i>datasets</i> anotados	40
2.3.2.2	<i>Crowdsourcing</i>	41
2.3.3	<i>Data Augmentation</i>	41
3	DESENVOLVIMENTO	45
3.1	Cultivo das amostras	45
3.2	Software	48
3.2.1	Nomenclatura e <i>data augmentation</i>	48
3.2.2	Comunicação com a nuvem	50
3.3	Hardware	50
3.3.1	Raspberry Pi com Webcam C920	52
3.3.2	Raspberry Pi com câmera modular V2.1	55
3.4	Construção dos protótipos	57
4	CONCLUSÃO	59

4.1	Resultados	59
4.2	Trabalhos futuros	59
	REFERÊNCIAS	61
	APÊNDICES	65
	APÊNDICE A – DICIONÁRIO DE AUMENTO DE DADOS	67
	APÊNDICE B – CÓDIGOS DE NOMENCLATURA	69

1 INTRODUÇÃO

Com o avanço científico e tecnológico, o setor agrícola tem sido fortemente impactado por novas ferramentas e abordagens. Entre elas, destaca-se a aplicação da Inteligência Artificial (IA) no contexto da visão computacional em múltiplas etapas e tarefas da produção agrícola, com ênfase em técnicas de aprendizado de máquina (*Machine Learning* - ML). Dentre essas técnicas, o *deep learning* (DL) aplicado a conjuntos de imagens organizadas e classificadas, *datasets* de imagens, tem-se consolidado como uma abordagem muito promissora para identificação de doenças, classificação de culturas, contagem de frutas, reconhecimento de plantas e fenotipagem (Santos *et al.*, 2020).

Essas aplicações possuem uma necessidade em comum, que é a de coletar e utilizar grandes bases de dados de imagens, o que envolve a captura, rotulagem (*labeling*) e a manipulação de fotografias dos objetos em análise (Roh; Heo; Whang, 2021). Essa atividade de preparação e rotulagem de dados, frequentemente, representa até 80% das tarefas dos envolvidos no desenvolvimento de aplicações de ML (Fredriksson *et al.*, 2020).

Diante disso, otimizar e automatizar essas tarefas tem-se mostrado uma importante frente de pesquisa e desenvolvimento. Melhorias, essas, que englobam desde o aprimoramento dos algoritmos de DL, passando pelo aumento artificial da quantidade de dados (*data augmentation*), até a integração com sistemas de Internet das Coisas (*Internet of things* - IoT) e técnicas de gerenciamento de dados (Kamilaris; Prenafeta-Boldú, 2018).

1.1 Motivação

Tendo em vista a importância da aquisição e manipulação de dados na construção de *datasets*, foram formadas frentes de pesquisa relacionadas ao tema, uma delas em associação ao IMAmt, que trabalhava em um escopo maior, mas com aplicações e condições para o estudo da construção de *datasets* de folhas de soja e algodão. Então, foram cultivadas plantas de ambas as espécies em estufas e, posteriormente, ao ar livre, possibilitando a obtenção de imagens e o desenvolvimento de sistemas embarcados que facilitam os processos de construção de um *dataset* útil.

Como poucos exemplares de plantas foram utilizados para a construção do *dataset*, fez-se necessária a utilização de técnicas de aumento de dados para ampliar a quantidade de imagens e evitar *overfitting* (Kumar *et al.*, 2024). A rotulagem e o armazenamento de imagens também apresentam desafios que foram abordados e mitigados com a automação da aquisição das imagens.

1.2 Objetivos

O objetivo geral deste trabalho é estudar e implementar técnicas para a obtenção e preparação de dados de imagem voltados para a construção de *datasets* utilizando uma cultura de plantas reduzida, de modo que seja um *dataset* aplicável a sistemas de *deep learning* no contexto agrícola.

Para alcançar esse objetivo, as etapas podem ser fracionadas da seguinte maneira:

- Realizar uma revisão dos estudos relacionados à construção de banco de dados de imagens.
- Desenvolver sistemas automatizados para a aquisição e manipulação das imagens.
- Construir um dataset de folhas de soja e algodão para o uso em deep learning, através de fotografias de uma cultura limitada.
- Utilizar técnicas de *data augmentation* para ampliar a quantidade de imagens.

1.3 Organização

Após a apresentação da motivação e dos objetivos que fundamentam este trabalho, no capítulo 1, expõe-se, a seguir, uma descrição de como o trabalho está organizado.

No capítulo 2, discute-se com mais detalhes a partir de uma revisão bibliográfica, o impacto das tecnologias de *deep learning* no meio agrícola, e os desafios na construção de bancos de dados confiáveis e relevantes.

Em seguida, no capítulo 3, é apresentada a evolução e o desenvolvimento das abordagens na construção do *dataset* das folhas de soja e algodão, as atividades realizadas para obter as imagens, armazená-las, nomeá-las, manipulá-las, desde os algoritmos até a construção dos hardwares.

Por fim, no capítulo 4, entendendo as limitações descritas no trabalho, são discutidas as conclusões que podem ser inferidas a partir da praticidade colhida dos sistemas desenvolvidos para a construção de uma base de dados, apontando possíveis direções para trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

O objetivo deste capítulo é apresentar uma revisão da literatura relacionada a todas as etapas do trabalho desenvolvido. Serão contextualizadas as tecnologias no meio agropecuário e serão abordadas técnicas e estudos de: uso da Internet das Coisas (IoT) na construção de *datasets*, gestão e preparação de dados, algoritmos de *data augmentation* e conceitos fundamentais de *Machine Learning* (ML) e *Deep Learning* (DL).

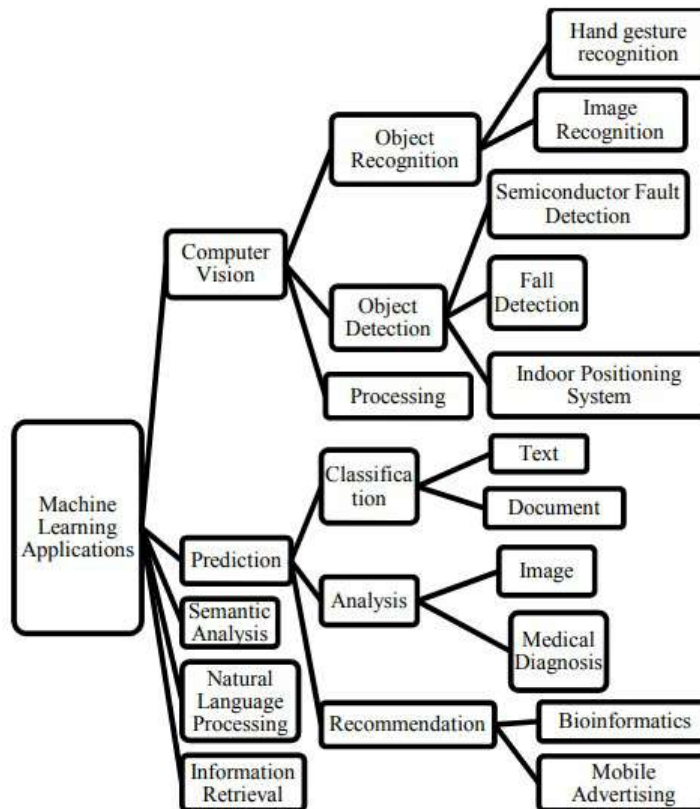
2.1 Conceitos de Machine Learning e Deep Learning

A Inteligência Artificial é um campo da computação que busca mimetizar o processo humano de reconhecer padrões e tomar decisões (Shinde; Shah, 2018), podendo ser baseada em padrões pré-programados, como um programa que joga um jogo da velha ou descreve uma rotina para robôs autônomos; ou pode basear-se em um aprendizado automático, caracterizando a técnica de *machine learning*.

O *deep learning*, por sua vez, é um subcampo do ML que busca realizar tarefas com maiores níveis de abstração de dados através do uso de arquiteturas hierárquicas de aprendizado (Guo *et al.*, 2016).

2.1.1 *Machine Learning*

Com maiores detalhes, em ML usam-se modelos matemáticos alimentados por dados de resultados anteriores para, através de parâmetros de performance, melhorar os resultados esperados. A escolha do modelo de ML e sua eficiência vão depender dos dados de entrada disponíveis e do que se quer adquirir como saída (*output*), ou seja, dependerá da aplicação, que é bastante ampla, como a imagem a seguir evidencia:

Figura 1 – Aplicações de *Machine Learning*

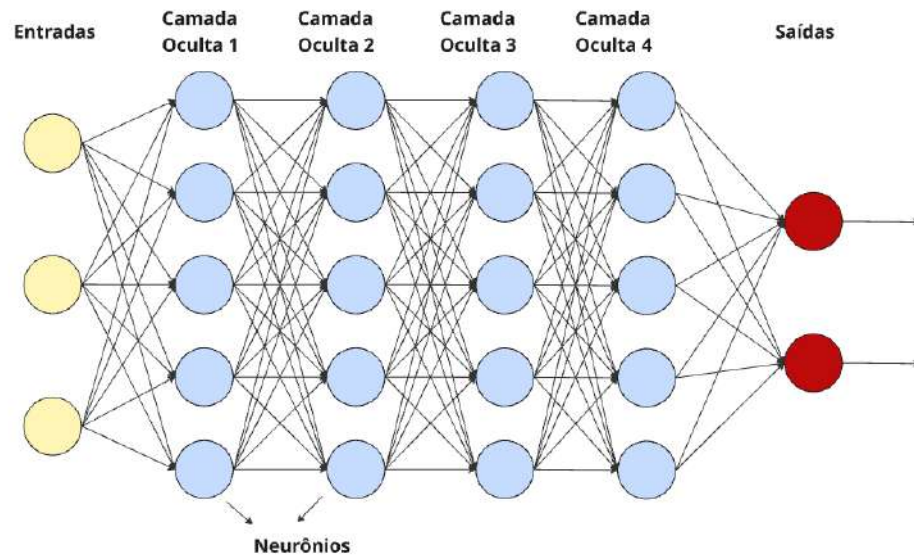
Fonte: Retirado de Shinde e Shah (2018)

Cada aplicação possibilita a utilização de diferentes modelos. Entre os mais utilizados, temos: Regressão Linear, Regressão Logística, Árvore de Decisão e Floresta Aleatória, SVM (*Support Vector Machines*), k-NN (*k-Nearest Neighbour*) e Rede Neural Artificial (RNA) (Shinde; Shah, 2018).

2.1.2 *Deep Learning*

Alguns modelos de ML, como o de Rede Neural Artificial, possibilitam um aprofundamento em sua aplicação, adicionando mais camadas e parâmetros, originando, assim, o que se entende como algoritmos de *deep learning*. O nome de Rede Neural Artificial surge pois o modelo matemático possui uma estrutura que se assemelha à rede neural animal, na qual várias unidades básicas ("neurônios") são conectadas formando as camadas que contribuem para determinar a saída de acordo com as respostas selecionadas nas suas conexões.

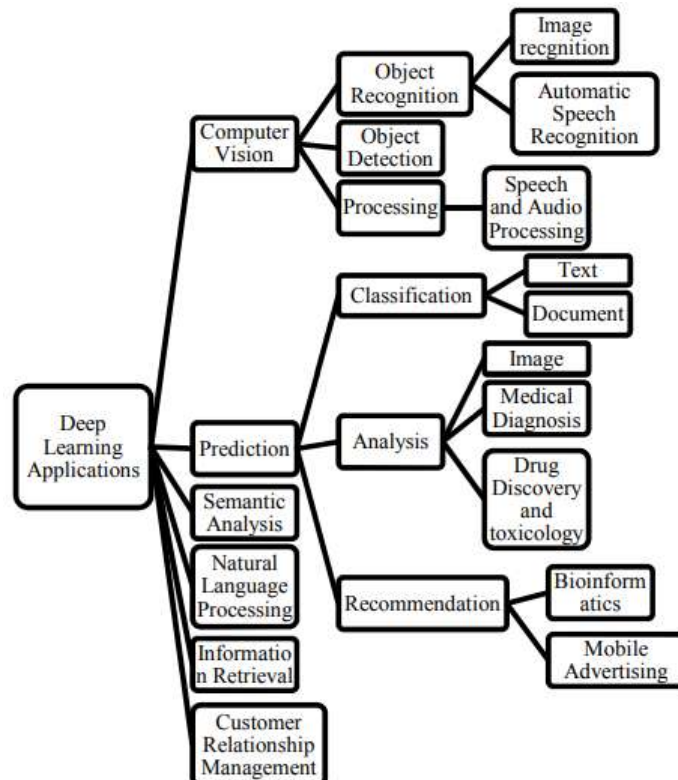
Figura 2 – Exemplo de uma estrutura de *deep learning*



Fonte: Elaborado pelo autor.

O DL abrange múltiplas aplicações em comum com o ML, mas com resultados e desenvolvimentos distintos devido à diferença entre os modelos matemáticos. Como pode ser observado na figura seguinte:

Figura 3 – Aplicações de *Deep Learning*

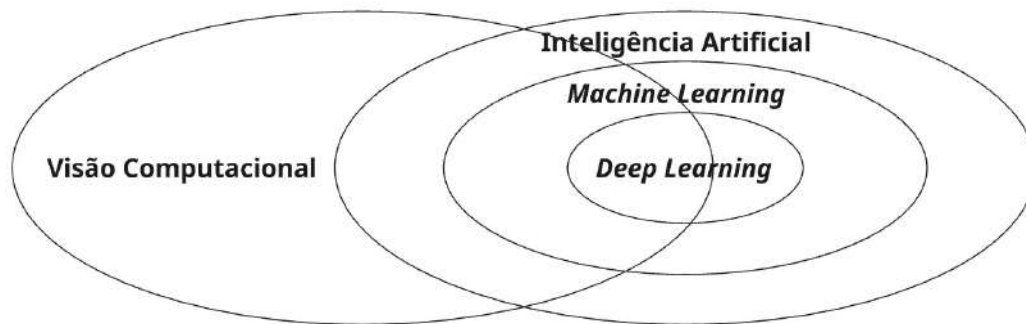


Fonte: Retirado de Shinde e Shah (2018)

2.1.3 Visão computacional

Uma área que aparece com destaque nas aplicações de ML e DL é a Visão Computacional, campo que possibilita à máquina interpretar informações visuais. Dentre suas aplicações, temos a classificação de imagens, detecção de objetos, recuperação de imagens e construção de espaços tridimensionais; que, embora mais frequentes, não estão restritas à utilização de IA.

Figura 4 – Diagrama dos campos de atuação



Fonte: Elaborado pelo autor

Segundo Guo *et al.* (2016), em competições organizadas pelo ImageNet Large Scale Visual Recognition Challenge (ILSVRC), métodos de DL foram amplamente adotados e alcançaram as maiores pontuações de acurácia, parâmetro que apresenta a porcentagem total de previsões corretas.

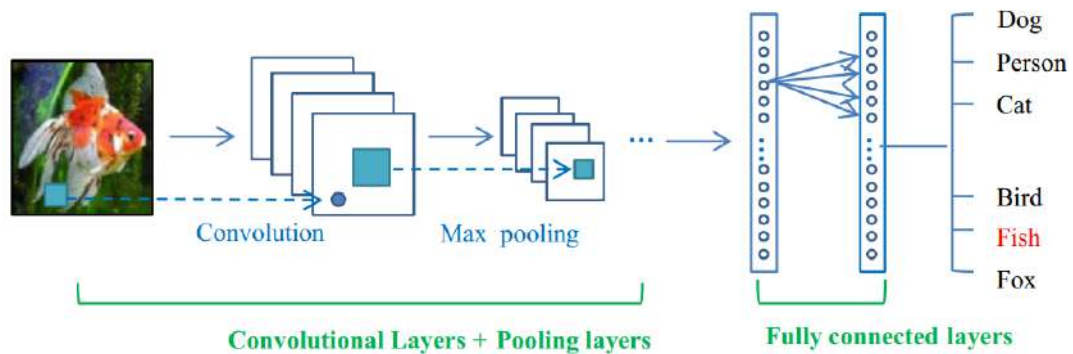
2.1.4 Deep learning aplicado a visão computacional

2.1.4.1 Redes neurais convolucionais

As redes neurais convolucionais (*Convolutional Neural Network* - CNN) são um dos modelos mais notáveis de rede neural no DL, elas têm se mostrado altamente eficazes e também são as mais comumente utilizadas em diversas aplicações de visão computacional (Guo *et al.*, 2016).

Todo o processo realizado pela arquitetura CNN - seu *pipeline* - pode ser compreendido pela aplicação de três camadas neurais principais, que são: camadas convolucionais (*Convolutional layer*), camadas de agregação (*pooling layer*) e camadas totalmente conectadas (*Fully connected layer*); cada tipo de camada desempenhando um papel diferente (Guo *et al.*, 2016).

Figura 5 – Pipeline da arquitetura geral da CNN



Fonte: Retirado de (Guo *et al.*, 2016)

Inicialmente, a CNN é alimentada com uma imagem, a qual passa pelo processo de convolução, uma combinação linear dos valores na vizinhança dos pixels de entrada. As camadas iniciais são capazes de encontrar linhas e bordas dos objetos, as camadas seguintes delimitam esses padrões em texturas e estruturas mais simples, sendo o *maxpooling* método responsável por transmitir para a camada seguinte apenas as regiões de maior valor de interesse. Finalmente, na camada totalmente conectada, esses elementos são sumarizados para classificar os objetos de interesse: uma planta, um peixe, uma pessoa (Santos *et al.*, 2020).

2.1.4.2 Aplicações na agricultura

A agricultura é um campo fértil para a utilização de visão computacional, pois trabalhos de inspeção e avaliação visual são frequentes. Como em OYAMA, JORGE e GOMES (2019), no qual foi desenvolvido um sistema que classificava exemplares de sementes de café. Todos os processos de classificação foram baseados em uma rede neural do tipo *Multilayer Perceptron* com apenas uma camada escondida, o que não caracteriza uma aplicação de *deep learning*.

Mas, como indica Santos *et al.* (2020), na pesquisa agropecuária há um número crescente de aplicações de visão computacional utilizando técnicas de DL. Fatores como - câmeras digitais acessíveis e facilmente integráveis, resultados satisfatórios e produção científica crescente - explicam essa popularidade.

Em Kamilaris e Prenafeta-Boldú (2018), foram avaliadas 40 pesquisas que utilizavam DL em aplicações de contexto agrícola. Cada pesquisa abordava uma aplicação ou desenvolvimento distinto: detecção de ervas daninhas, identificação de sementes, classificação de folhas de diferentes espécies, identificação de folhas doentes. O trabalho de Kamilaris e Prenafeta-Boldú (2018) chega à conclusão de que o DL supera as técnicas de processamento de imagem comumente utilizadas. A conclusão foi construída a partir das perguntas do Quadro 1:

Quadro 1 – Questões analisadas por Kamilaris e Prenafeta-Boldú (2018)

1. Qual foi o problema agrícola abordado?
2. Qual foi a abordagem geral e qual modelo de Deep Learning utilizado?
3. Quais foram as fontes e os tipos de dados utilizados?
4. Quais foram as classes e rótulos utilizados na modelagem pelos autores? Houve alguma variação entre eles, observada pelos autores?
5. Algum pré-processamento dos dados ou técnicas de data augmentation foram utilizadas?
6. Qual foi o parâmetro de avaliação de performance utilizado?
7. Os autores testaram o desempenho de seus modelos em diferentes <i>datasets</i> ?
8. Os autores compararam sua abordagem com outras técnicas e, em caso afirmativo, qual foi a diferença na performance?

Fonte: Adaptado de Kamilaris e Prenafeta-Boldú (2018)

E as questões levantadas acerca da avaliação da performance dos modelos de DL foram analisadas através de parâmetros, como:

Tabela 1 – Parâmetros de avaliação de performance de modelos *deep learning* de classificação

Métrica de performance	Descrição
<i>Accuracy</i>	Porcentagem de previsões corretas sobre o total.
<i>Precision</i>	Proporção de verdadeiros positivos (<i>True Positive</i> - TP) entre todas as previsões positivas (TP + (<i>False Positive</i> - FP)). Mede quantos dos positivos previstos realmente são verdadeiros.
<i>Recall</i>	Proporção de verdadeiros positivos identificados corretamente dentre todos os positivos reais (TP + (<i>False Negative</i> - FN)). Mede a capacidade de identificar todos os positivos reais.
F1 Score	Média harmônica entre Precisão e <i>recall</i> .

Fonte: Adaptado de Kamilaris e Prenafeta-Boldú (2018).

Por exemplo, no contexto da identificação de doenças em plantas, Barbedo (2018) analisa como a *accuracy* varia com o tamanho do *dataset*. No geral, na utilização de CNNs, alcançar bons valores para essas métricas de performance demanda uma grande quantidade e variedade de imagens, muito devido às doenças se apresentarem de diferentes formas e padrões, exigindo uma maior quantidade de exemplos. O que reflete nos resultados obtidos apresentados na Tabela 2 (Barbedo, 2018).

Além dessas dificuldades, no campo da agricultura, ainda não existem muitos *datasets* públicos disponíveis para a utilização em pesquisas e desenvolvimento. Em muitos casos, é necessário construir o próprio *dataset* do zero, o que pode levar dias de trabalho (Kamilaris; Prenafeta-Boldú, 2018).

Tabela 2 – Exemplos de desempenho de CNNs relacionados à quantidade de imagens no *dataset*

Cultura	Número de imagens	Acurácia (%)
Feijão	64	65
Café	142	82
Soja	377	86
Milho	165	71

Fonte: Adaptado de Barbedo (2018).

2.1.4.3 *Few Shot Learning e Transfer learning*

Few-Shot Learning (FSL) é um tipo de problema de ML no qual há um número limitado e pequeno de dados supervisionados para treinamento (Wang *et al.*, 2020). A FSL pode reduzir o esforço para atingir resultados satisfatórios, aliviando a carga da coleta de dados supervisionados (Ravi; Larochelle, 2017).

Ainda em Wang *et al.* (2020) é apresentada uma técnica popular de FSL que é a utilização de *transfer learning*. Na qual se aproveita de modelos treinados com outro *dataset*, por exemplo, para identificar uma folha de soja, é possível aproveitar um modelo que já foi treinado para identificar uma folha de café; exigindo menos ciclos e imagens para alcançar uma boa acurácia.

2.2 IoT e Sistemas Embarcados na agricultura

Além de desenvolver e aperfeiçoar algoritmos de *Machine Learning*, com a facilitação do uso da internet e de tecnologias em nuvem, possibilita-se o uso de *Internet of Things* (IoT), na qual as práticas agrícolas tradicionais podem ser combinadas com dispositivos inteligentes e sistemas embarcados para automatizar alguns ou todos os processos envolvidos. Essas tecnologias permitem que os agricultores aumentem o rendimento das colheitas, controlando vários aspectos da produção agrícola, como a manutenção de condições climáticas, utilização de fertilizantes e pesticidas no momento e na quantidade corretas, construção de *datasets* e refinamento dos dados (Mathe *et al.*, 2022). Automatizar esses processos é uma característica de fazendas inteligentes, na qual a agricultura de precisão permite uma produção agrícola otimizada por meio de decisões em tempo real, baseadas em dados e repetíveis.

2.2.1 Raspberry Pi na agricultura

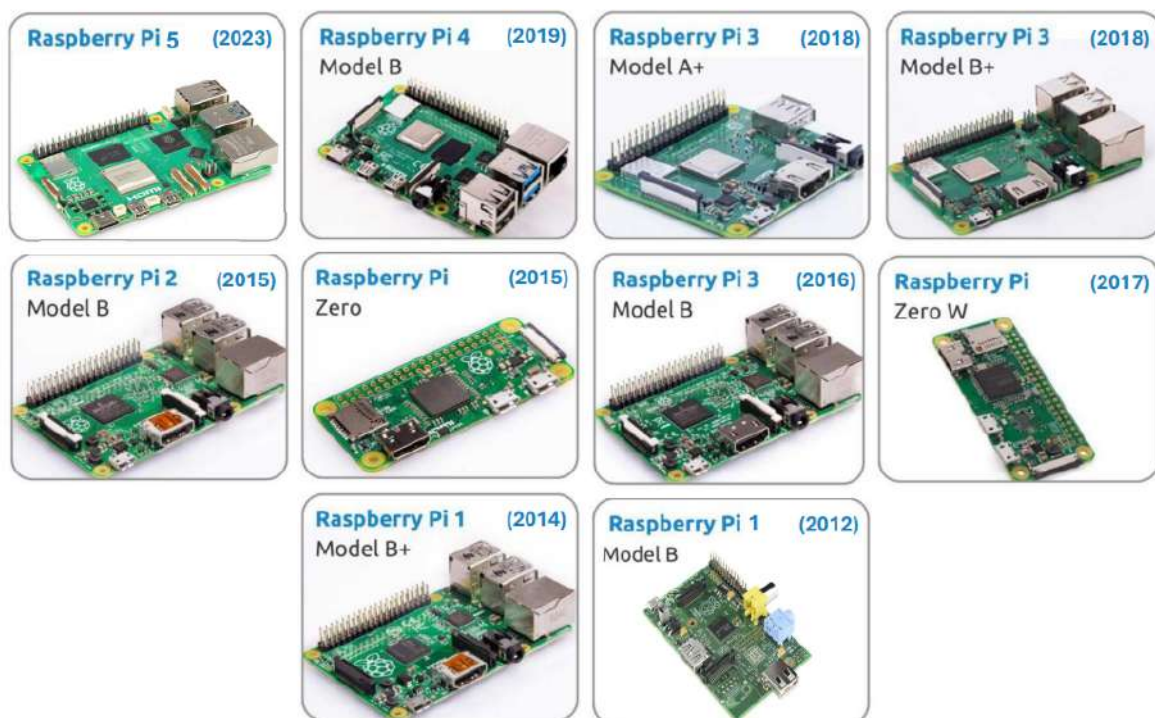
Em Joice *et al.* (2025) destaca-se que, para a análise de dados em tempo real e tomada de decisões no campo, são essenciais dispositivos/sensores que sejam fáceis de manusear, portáteis e com alta capacidade de processamento. *Single-Board Computers*

(SBCs), que oferecem mobilidade, versatilidade, alto poder computacional em um formato compacto e baixa latência, podem ser adotados para tais finalidades.

O Raspberry Pi é um SBC popular que tem sido amplamente utilizado em diversas aplicações, incluindo educação em ciência da computação, robótica, redes de sensores e IoT. Trata-se de uma plataforma de baixo custo que reduz drasticamente o custo do sistema de sensores com câmera, facilitando para os cientistas a implantação de unidades de forma econômica (Curtin; Matthews, 2019). Ainda em Curtin e Matthews (2019), a placa Raspberry Pi 3B+ com um sistema de câmera modular foi utilizada para identificação de vida silvestre por meio de um modelo já treinado de DL.

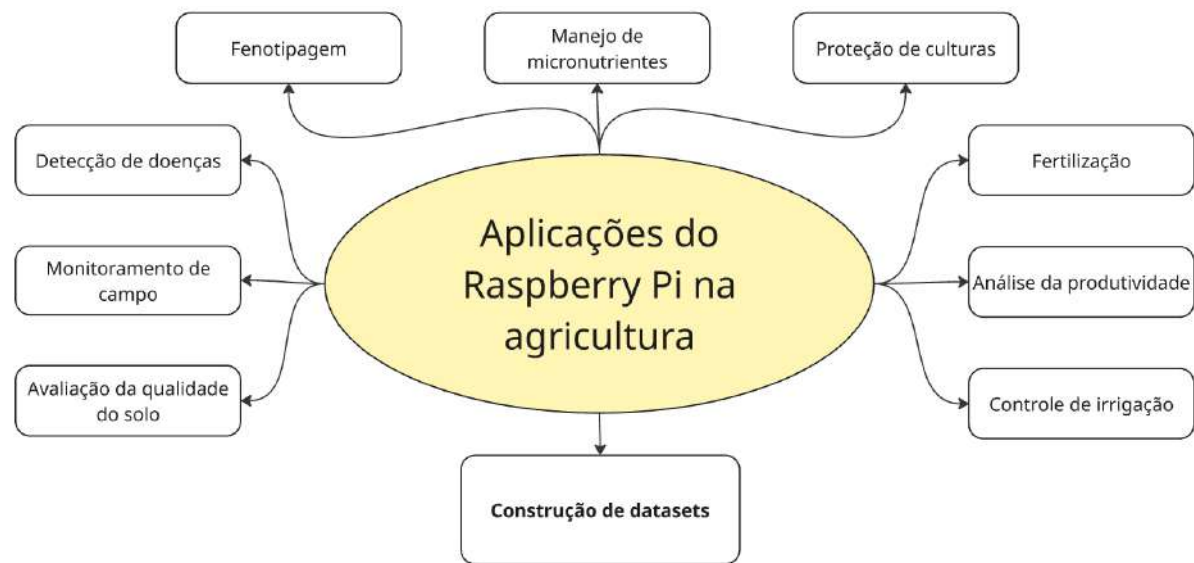
Já em (Mathe *et al.*, 2022), são abordadas algumas aplicações da Raspberry Pi no ambiente agrícola, as quais fazem parte do desenvolvimento de diversos sistemas automatizados de sensoriamento e monitoramento para o acompanhamento de culturas, detecção de doenças, fenotipagem, controle de irrigação, manejo do solo e proteção das plantas. Seu processador potente, custo reduzido, ampla interface de entradas e saídas e a sua compatibilidade permitiram que a maioria dos projetos fosse executada nele.

Figura 6 – Modelos de Raspberry



Fonte: Retirado de Joice *et al.* (2025)

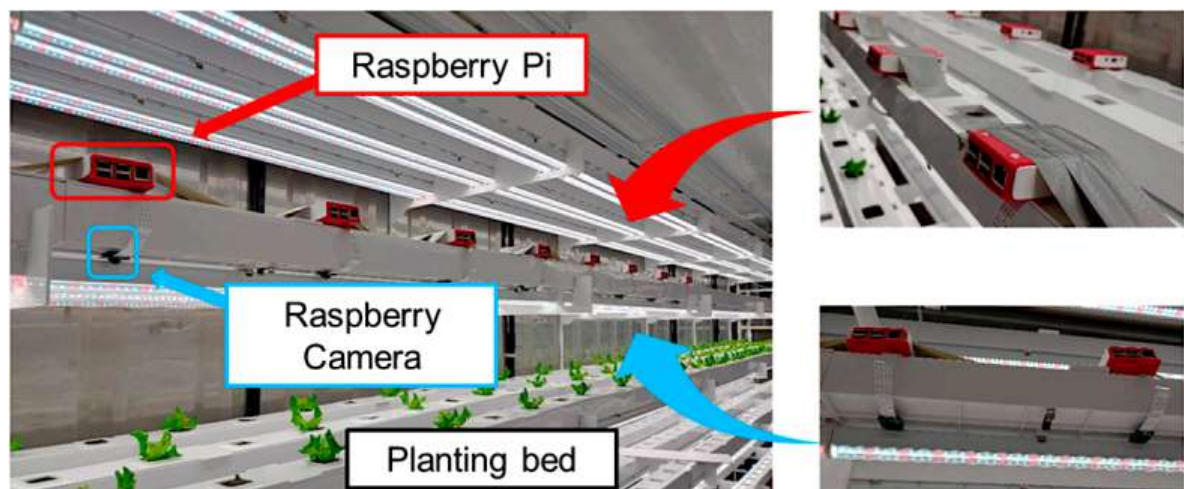
Figura 7 – Aplicações do Raspberry Pi na agricultura



Fonte: Adaptado de Mathe *et al.* (2022)

A construção de *datasets* permeia todas as aplicações, pois o armazenamento de dados é parte fundamental para análises mais detalhadas. Como em Cho e Yang (2023), no qual foi desenvolvido um sistema de baixo custo composto por um computador como servidor e múltiplos Raspberry Pi 3B+ com suas câmeras modulares para monitorar o crescimento controlado de alfaces, possibilitando o armazenamento de múltiplas imagens ao longo do tempo. Essas imagens foram utilizadas para realizar a fenotipagem da cultura, que seria a avaliação de estruturas físicas das plantas (quantidade de folhas, tamanho das folhas, cor, etc.).

Figura 8 – Sistema com Raspberry Pi para monitoramento de culturas vegetais



Fonte: Retirado de Cho e Yang (2023)

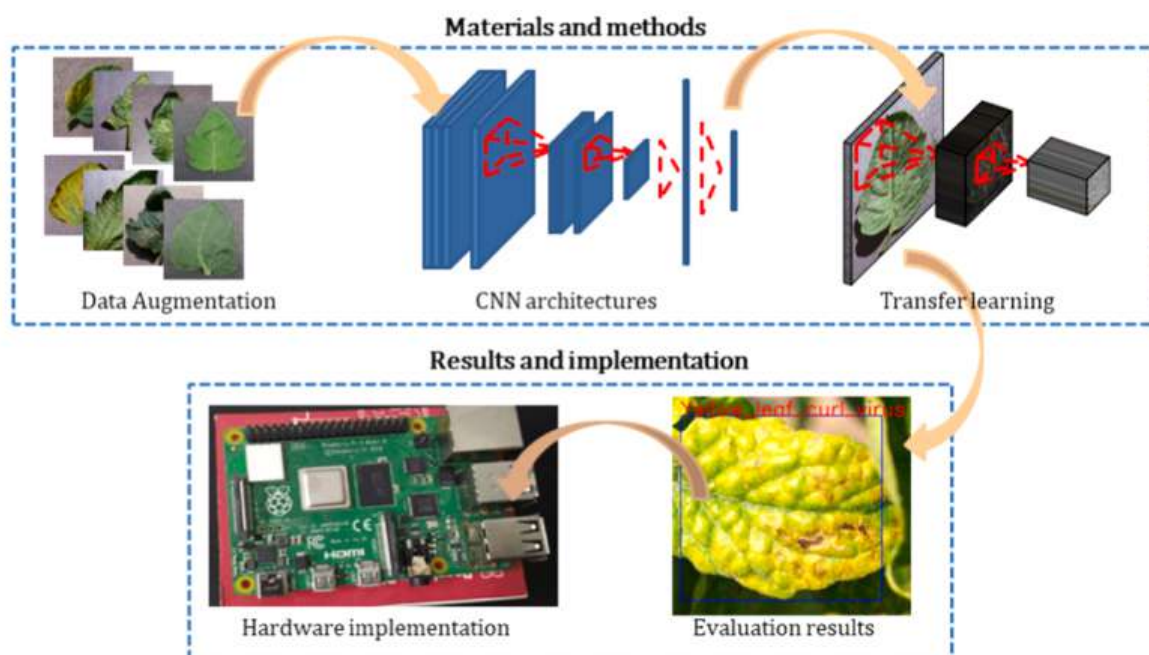
2.2.2 Raspberry Pi e Machine Learning

A aplicação de *deep learning* tem feito contribuições essenciais para tarefas de classificação e detecção aplicadas à agricultura de precisão; é importante avançar e facilitar a adoção dessas técnicas e algoritmos por meio de dispositivos de baixo custo e baixo consumo (Gonzalez-Huitron *et al.*, 2021).

O Raspberry Pi, por apresentar uma boa relação custo-benefício, portabilidade, eficiência energética e facilidade de manutenção, ajuda a enfrentar os desafios, caso modelos de ML e DL possam ser implantados nele. Entre as diferentes versões do Raspberry Pi, a versão 3 modelo B e superiores (V3 B+, V4B e V5) são comumente utilizadas para a implantação de modelos de ML devido ao seu poder de processamento adequado (processador quad-core), memória RAM (de 4 a 8 GB) e suporte a aceleradores de hardware externos, além de possibilitar a implementação de periféricos embarcados com facilidade e possuir acessibilidade à rede (Ethernet gigabit, Wi-Fi de banda dupla, Bluetooth) (Joice *et al.*, 2025).

Em Gonzalez-Huitron *et al.* (2021) utilizaram um sistema composto por um Raspberry Pi 4 e a câmera webcam C920 para capturar imagens de folhas de tomate e avaliá-las (classificar com a arquitetura de CNN). Essa CNN foi treinada em outro hardware mais potente, utilizando um dataset de livre acesso e com aplicação de *data augmentation*. A imagem a seguir ilustra o processo até a implementação no Raspberry.

Figura 9 – *Deep learning* implementado no Raspberry Pi

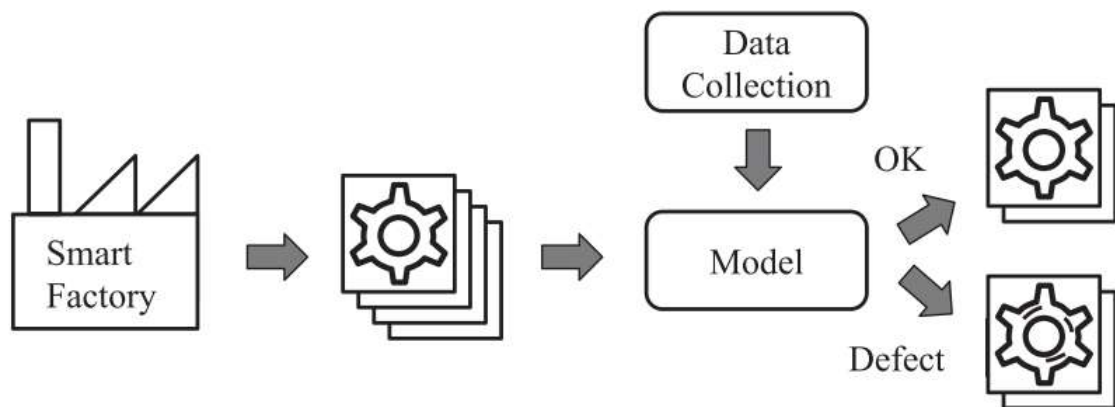


Fonte: Retirado de Gonzalez-Huitron *et al.* (2021)

2.3 Construção de *datasets*, rotulagem e *data augmentation*

A coleta de dados e a construção de *datasets* é um dos gargalos críticos na aplicação de projetos de *machine learning*. A maior parte do tempo é despendida na coleta, limpeza, análise e avaliação de características dos dados. Em múltiplas aplicações, já se têm dados de treinamento em grandes quantidades disponíveis. Porém, em fábricas inteligentes, por exemplo, em que se realiza o controle de qualidade dos produtos com inteligência artificial, sempre haverá um produto novo ou uma nova característica a ser detectada. A construção de um novo *dataset* de treinamento e o processo manual de rotulagem desses dados pode não ser viável, pois exige um investimento financeiro, de tempo e de conhecimento especializado para cada nova aplicação; processo que pode ser ainda mais difícil se a aplicação for de *deep learning*, a qual exige uma grande quantidade de dados para funcionar satisfatoriamente (Roh; Heo; Whang, 2021).

Figura 10 – Coleta de dados no ambiente industrial

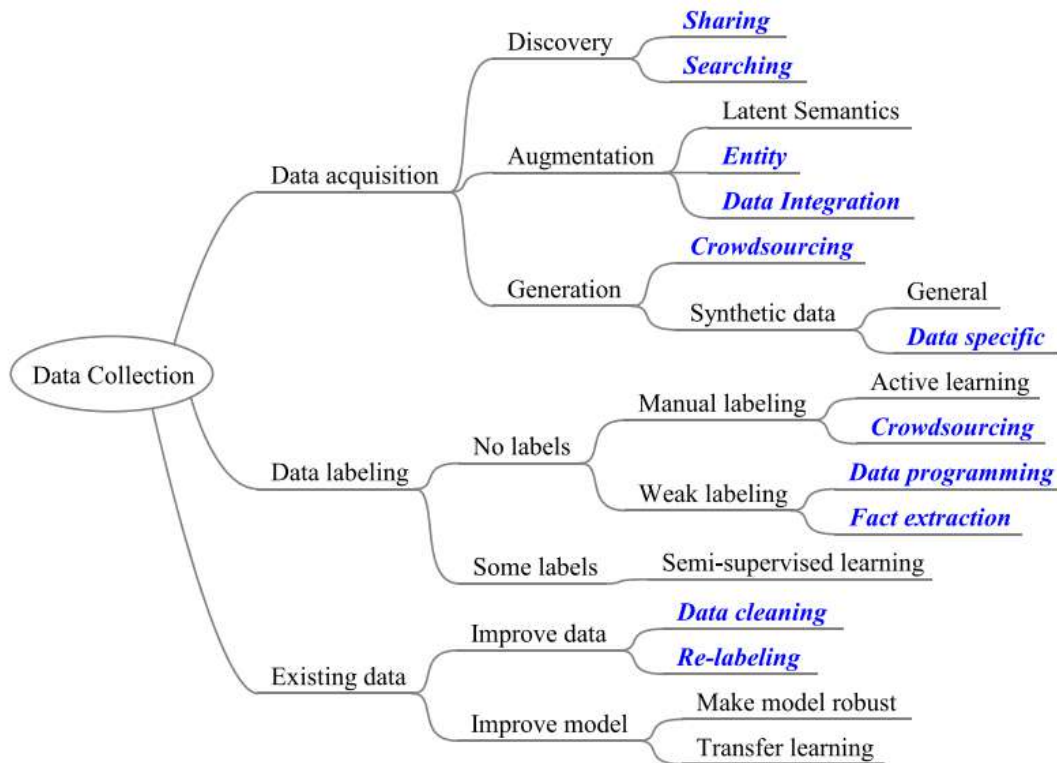


Fonte: Retirado de Roh, Heo e Whang (2021)

Ainda em Roh, Heo e Whang (2021), destaca-se que a coleta e gerenciamento de dados não é uma novidade que surgiu para as implementações em ML e DL, mas sim um campo que é estudado há décadas pela comunidade de gerenciamento de dados, principalmente sob as denominações de ciência de dados e análise de dados, que têm feito contribuições significativas nessa área. A exemplo, em Brown *et al.* (2005) um design de banco de dados flexível e escalável é implementado para dar suporte à pesquisa em análise quantitativa de imagens na medicina.

A Figura 11 apresenta uma visão geral do panorama de pesquisa, na qual os tópicos que receberam contribuições da comunidade de gerenciamento de dados estão destacados em texto azul e *itálico*.

Figura 11 – Panorama dos tópicos da coleta de dados



Fonte: Retirado de Roh, Heo e Whang (2021)

2.3.1 Dados para aprendizado de máquinas

O primeiro passo para iniciar a coleta dos dados é entender quais informações procura-se extrair desses dados. Tendo isso em mente, os dados a serem coletados são facilmente identificáveis, e seus requisitos de qualidade e valor ficam claramente definidos. Para o treinamento de modelos de ML, pode-se segmentar as etapas da preparação dos dados em:

- **Coleta:** Obter e armazenar os dados que podem levar ao *output* definido previamente.
- **Limpeza:** Remoção de ruídos ou inconsistências nos dados, incluindo a eliminação de dados faltantes ou contraditórios.
- **Transformação:** Adaptação dos dados brutos para formatos adequados ao processamento por algoritmos de ML.
- **Redução:** Eliminação de informações redundantes ou irrelevantes, mantendo apenas os dados essenciais para o desenvolvimento da aplicação.

Dados limpos, pré-processados e consistentes produzem resultados significativamente mais precisos quando comparados a dados inconsistentes, ruidosos e com erros (Ndung'u, 2022).

2.3.1.1 *Small data*

Grandes quantidades de dados são utilizadas para treinamentos e aplicações de ML e DL, muitas vezes essa base de dados pode ser vista como uma aplicação de *Big data*. Mas o conceito de *Small data*, caracterizado por um volume limitado, coleta descontínua, variedade restrita e geração voltada a atender questões específicas continua a ser popular e a apresentar utilidade em diferentes tarefas (Kitchin; Lauriault, 2015).

Tabela 3 – Comparação entre *Small data* e *Big data*

Característica	Small data	Big data
Volume	Limitado a grande	Muito Grande
Exaustividade	Amostras	Populações inteiras
Relacionalidade	Fraca a forte	Forte
Velocidade	Lenta	Rápida
Variedade	Limitada a ampla	Ampla
Flexibilidade e escalabilidade	Baixa a média	Alta

Fonte: Adaptado de Kitchin e Lauriault (2015).

Uma aplicação clara para *small data* envolve o conceito de *Few-Shot Learning* (Wang *et al.* (2020)) que já foi apresentado anteriormente no subseção 2.1.4.3. Trata-se de uma prática que está dentro de um conceito maior que é o de *Small Sample Learning* (SSL), cuja motivação é mimetizar a capacidade humana de generalizar conceitos a partir de poucos exemplos. Shu, Xu e Meng (2018) aponta alguns elementos que reforçam essa necessidade:

- **Escassez de rótulos por causa do custo elevado da anotação manual** - Alto custo e dificuldade de anotar dados com precisão (identificar qual doença em uma planta por exemplo). Muitas vezes, dados coletados por crowdsourcing contêm anotações imprecisas, prejudicando o desempenho.
- **As distribuições de cauda longa são amplamente encontradas em conjuntos de big data** - Em muitos conjuntos de dados, poucas classes possuem muitos exemplos, enquanto a maioria tem poucos. Isso faz com que os modelos priorizarem as classes mais frequentes e apresentem baixo desempenho nas classes com poucas amostras.
- **Quantidade insuficiente de dados para métodos tradicionais de aprendizagem de máquina** - Mesmo na era dos *Big Data*, existem áreas onde a quantidade de dados é limitada, como diagnósticos médicos de doenças raras ou comunicação inteligente com poucos sinais de referência.

- **Surgiu a partir de estudos da ciência cognitiva** - O SSL se inspira em estudos sobre como humanos aprendem a generalizar a partir de poucas experiências. Pesquisas interdisciplinares com ciência cognitiva e neurociência buscam criar sistemas de IA que imitem o aprendizado humano.

Os estudos com *big data* não substituem os estudos com *small data*; na verdade, eles se complementam. Pequenas bases de dados apresentam facilidades na alteração de padrões, formatos, metadados e documentação, permitindo um controle muito maior do desenho da pesquisa e respostas a perguntas específicas e direcionadas;

Ademais, bases de dados pequenas estão sendo reinterpretadas no âmbito de novas infraestruturas de armazenamento e compartilhamento de dados. Isso com a finalidade de garantir sua compatibilidade com sistemas mais complexos, maximizar sua descoberta e facilitar a vinculação entre conjuntos de dados (Kitchin; Lauriault, 2015).

2.3.2 *Data labeling*

2.3.2.1 Construção de datasets anotados

Para treinar modelos de ML, é necessário marcar, anotar, nomear as amostras de dados que serão utilizadas; esse processo é conhecido como rotulagem de dados (*data labelling*). Por exemplo, se um modelo deve distinguir entre imagens de soja e algodão, ele será inicialmente treinado com imagens de soja e algodão que estão marcadas como 'soja' ou 'algodão'. O processo de anotação é feito através de *crowdsourcing*, *active learning*, aprendizado semi-supervisionado ou manualmente (Fredriksson *et al.*, 2020). Esta parte do aprendizado supervisionado permite que o modelo forme uma base para aprendizados futuros (Ndung'u, 2022).

Fredriksson *et al.* (2020) destaca desafios tanto na exploração e construção de *datasets* devidamente anotados (Quadro 2), quanto na validação e utilização ao longo do tempo desses *datasets* (Quadro 3).

Quadro 2 – Desafios na construção de dados anotados

1. Falta de uma abordagem sistemática para rotular dados com base em características específicas
2. Ausência de definição clara de responsabilidades no processo de rotulagem
3. Rótulos ruidosos (inconsistentes)
4. Dificuldade de mapeamento rótulo-característica
5. Distribuição desbalanceada de labels
6. Dependência temporal
7. Dificuldade em perceber futuros usos do <i>dataset</i>

Fonte: Adaptado de Fredriksson *et al.* (2020)

Quadro 3 – Desafios na validação dos dados rotulados

1. Má distribuição de rótulos, possuir uma amostragem muito grande de um rótulo enquanto os outros possuem uma amostragem pequena
2. Retrabalho para adicionar ou modificar os rótulos quando os mesmos dados são utilizados para múltiplas tarefas
3. Conflito entre colaboradores ao rotular dados
4. Falha na reutilização de <i>datasets</i> para obter outros outputs desejados

Fonte: Adaptado de Fredriksson *et al.* (2020)

Mostra-se desafiador estruturar e implementar um processo de rotulagem. É necessário definir uma abordagem sistemática para a rotulagem, analisar os dados para selecionar o modelo ideal e, por fim, escolher um oráculo (fonte de referência) capaz de produzir rótulos de alta qualidade, além de planejar como lidar com distribuições desbalanceadas de rótulos.

Primeiro, com base em um estudo de caso envolvendo duas empresas, identificamos problemas que elas enfrentam na rotulagem de dados. Validamos esses problemas por meio de entrevistas nas duas empresas e consolidamos todos eles em desafios.

Segundo, apresentamos uma visão geral das estratégias de mitigação que as empresas utilizam (ou poderiam utilizar) para enfrentar esses desafios.

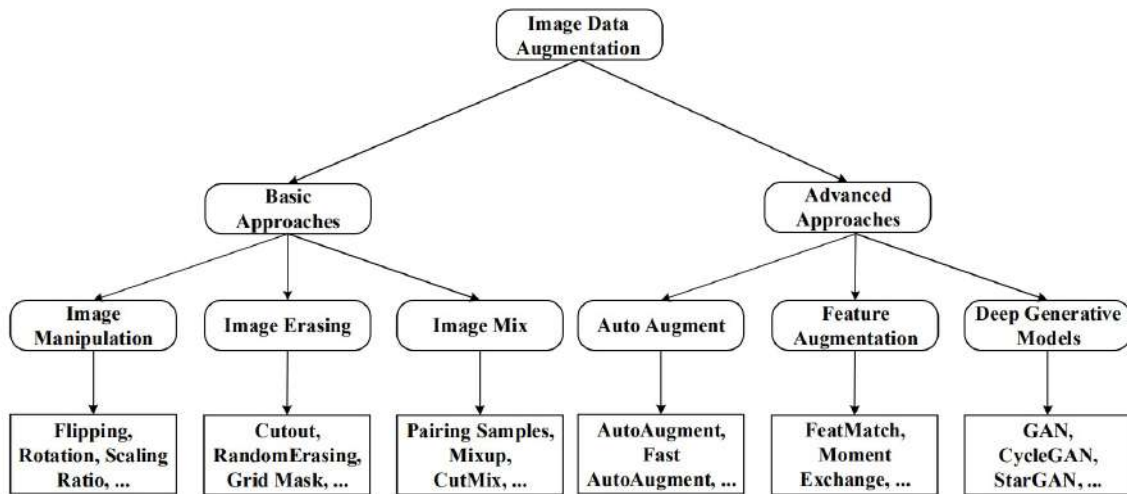
2.3.2.2 *Crowdsourcing*

A criação de *labels* para as imagens que compõem um *dataset* frequentemente é uma tarefa que demanda muito tempo, e, portanto, alocar engenheiros ou profissionais especializados pode não ser a melhor maneira de abordar a tarefa. Uma alternativa é a utilização de serviços de *Crowdsourcing* que é um método eficaz para externalizar tarefas massivas de rotulagem, utilizando mão de obra sob demanda. Esses serviços podem ser divulgados em plataformas populares como Amazon Mechanical Turk e Lionbridge AI, visando garantir uma maior segurança e confiabilidade (Lionbridge, 2024).

Mas a utilização de *crowdsourcing* apresenta alguns desafios significativos, como: gestão de colaboradores, decomposição de problemas em tarefas menores e, o maior deles, sendo o controle de qualidade das *labels* geradas (Roh; Heo; Whang, 2021).

2.3.3 *Data Augmentation*

O aumento de dados (*data augmentation*) é um método utilizado para ampliar a quantidade e diversidade de um conjunto de dados para o treinamento de modelos de *deep learning*. Consiste em gerar novos exemplos de dados a partir das amostras já existentes; a Tabela 4 apresenta alguns exemplos de manipulações básicas realizadas em imagens com essa finalidade (Ndung'u, 2022).

Figura 12 – Técnicas de *data augmentation*

Fonte: Retirado de Yang *et al.* (2022)

Tabela 4 – Manipulações básicas de imagens

Método	Descrição
Flip (inversão)	Inverte a imagem horizontalmente, verticalmente ou em ambos os eixos.
Rotação	Gira a imagem em um determinado ângulo.
Escala	Aumenta ou reduz o tamanho da imagem proporcionalmente.
Injeção de Ruído	Adiciona ruído artificial à imagem.
Modificação de Canais de Cor	Altera os canais de cores (ex.: RGB para tons de cinza).
Ajuste de Contraste	Modifica o contraste da imagem (mais claro/escuro).
Nitidez	Aumenta ou reduz a nitidez dos detalhes da imagem.
Translação	Desloca a imagem horizontalmente, verticalmente ou em ambas as direções.
Crop (Recorte)	Extraí uma sub-região (seção retangular) da imagem original.

Fonte: Adaptado de Yang *et al.* (2022).

Além das manipulações básicas de imagem apresentadas na Tabela 4, a Figura 12 apresenta diversas outras técnicas para o aumento de dados. Vale destacar a *Generative Adversarial Networks* (GAN), que tornou-se o padrão da indústria e da pesquisa no sentido da modelagem generativa, método em que se utiliza duas redes: uma sendo o gerador, que irá produzir imagens inéditas, e a outra sendo o discriminador, que irá verificar se

essas imagens são falsas. Durante esse processo, ambas as redes se aprimoram, tornando o discriminador cada vez melhor em distinguir imagens falsas das reais, e o gerador cada vez melhor em gerar imagens falsas indistinguíveis das verdadeiras (Karagiannakos, 2020).

Por fim, *data augmentation* é importante mesmo em aplicações mais comuns e com grande amostragem de dados, pois um modelo pode ser treinado com muitas imagens, mas pouca variedade entre elas, levando ao *overfitting*, em que o modelo aprende demais os detalhes e ruídos do conjunto de treinamento, em vez de aprender padrões gerais que realmente importam para generalizar em dados novos. Fica claro que o aumento de dados é uma técnica muito útil para a construção de *datasets* mais eficientes (Shorten; Khoshgoftaar, 2019).

3 DESENVOLVIMENTO

Este capítulo apresenta as etapas da elaboração de um sistema embarcado com a finalidade de facilitar a construção de *datasets*. Para isso, utilizaram-se algumas poucas amostras de plantas de soja e algodão para contextualizar o uso agrícola. Primeiramente, o cultivo das amostras de soja e algodão é contextualizado. Então, é abordado como a captura das imagens dessa cultura foi realizada e quais outros materiais e métodos foram desenvolvidos para facilitar e agilizar esse procedimento.

3.1 Cultivo das amostras

As sementes de soja e algodão foram inicialmente cultivadas em estufas controladas, nas quais a manipulação da umidade, da temperatura e da iluminação fazia parte de outra pesquisa. Com o crescimento das amostras, tornou-se necessário transplantar as mudas para vasos maiores e prosseguir com seu cultivo ao ar livre em outro ambiente.

Figura 13 – Estufa com ambiente controlado



Fonte: Elaborado pelo autor

No total, foram transplantadas 25 mudas: 13 de soja e 12 de algodão. Durante todo o processo de crescimento das plantas, foram tiradas fotos utilizando a câmera traseira 2201117TG de um telefone celular. A Figura 14, a Figura 15 e a Figura 16 apresentam, respectivamente, o momento da transplantação, a fase inicial e uma etapa avançada de crescimento.

As imagens obtidas integraram um *dataset* compartilhado por uma equipe maior, na qual mais colaboradores compartilhavam imagens de outras plantas de soja e algodão em diferentes ambientes e condições. Com esse sistema de dados compartilhados, surgiram algumas complicações para realizar a manutenção de informações relevantes das imagens capturadas, levando ao desenvolvimento de abordagens padronizadas através de sistemas embarcados discutidas nos capítulos seguintes.

Figura 14 – Mudas transplantadas



Fonte: Elaborado pelo autor

Figura 15 – Plantas em etapa inicial de crescimento



Fonte: Elaborado pelo autor

Figura 16 – Plantas em etapa final de crescimento



Fonte: Elaborado pelo autor

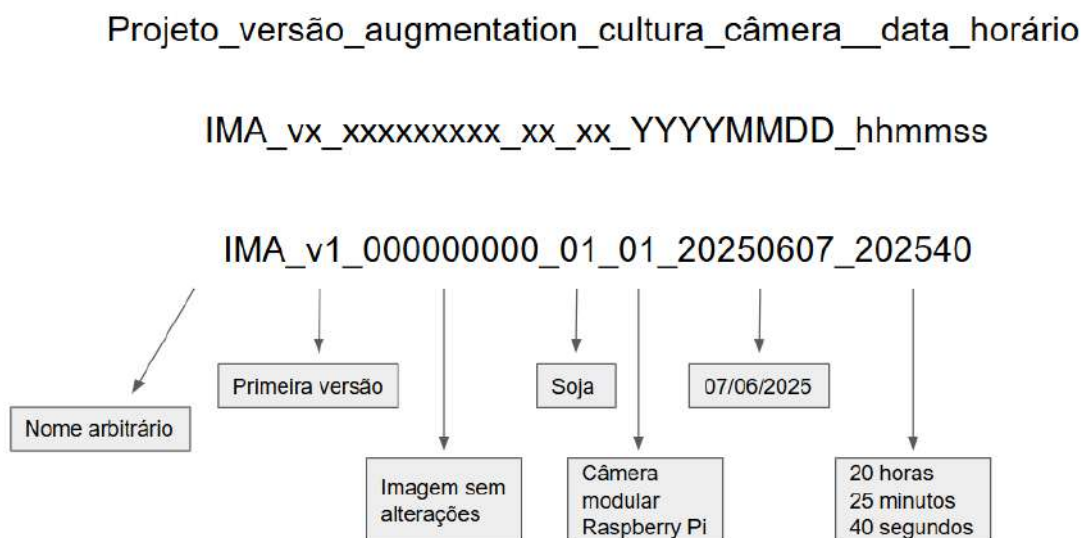
3.2 Software

Em Scharr *et al.* (2014), foi abordada a construção de *datasets* organizados de culturas de *Arabidopsis thaliana* e *Nicotiana tabacum*; estabelecendo um protocolo próprio para a anotação e rotulação das imagens obtidas. Inspirando-se em Scharr *et al.* (2014), buscou-se construir um sistema semelhante, com a anotação e a nomeação das imagens por meio de um processo automatizado e escalável, sem a utilização de especialistas, os quais podem analisar e reclassificar as imagens de outras maneiras posteriormente.

3.2.1 Nomenclatura e data augmentation

Todas as imagens obtidas pela câmera do celular foram inicialmente organizadas e separadas manualmente em pastas distintas. Em seguida, utilizando códigos em Python e informações dos metadados, as imagens foram renomeadas seguindo a seguinte estrutura:

Figura 17 – Exemplo da estrutura do código do nome



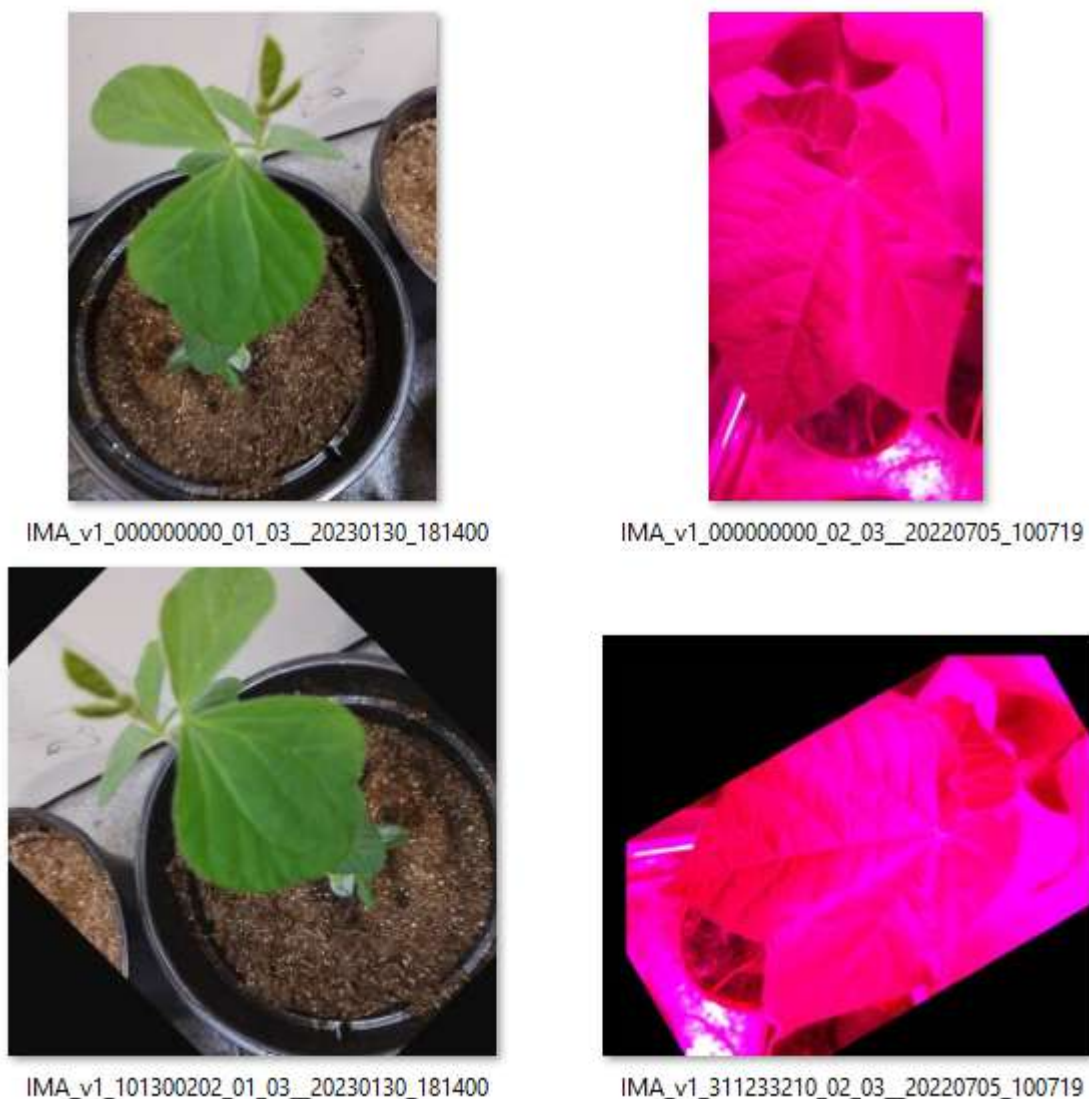
Fonte: Elaborado pelo autor

As fotos nomeadas como descrito na Figura 17 passaram, então, pelo processo de *data augmentation* para aumentar a quantidade de amostras. Essa atividade, e todas as seguintes, também foram realizadas utilizando a linguagem Python com o auxílio de trabalhos como (Chollet, 2023). O Apêndice A apresenta um dicionário de como o código numérico no nome se relaciona com as modificações aplicadas na imagem.

O código no nome da imagem dedicado ao aumento de dados contém nove dígitos, cada dígito indica qual modificação foi realizada e em qual nível essa mudança foi aplicada, como o Apêndice A apresenta, sendo as alterações lidas no código da direita para a esquerda. Portanto, 000000000 significa que se trata da imagem original sem modificações, e outros códigos, como os da Figura 18 indicam quais modificações a imagem carrega.

Para além do trecho de nove dígitos, o Apêndice B traduz o restante das informações das imagens.

Figura 18 – Exemplos de imagens salvas



Fonte: Elaborado pelo autor

Para cada imagem, é possível gerar mais de 100.000 imagens alternativas, com as transformações aplicadas sendo feitas de forma randomizada. Na Figura 18 têm-se dois exemplos de imagens após o processo de aumento de dados; ambas são do projeto IMA, primeira versão do código e foram obtidas com a câmera de celular 2201117TG. Além disso, IMA_v1_101300202_01_03_20230130_181400 é uma planta de soja, cuja foto foi tirada no dia 30/01/2023 às 18 horas, 14 minutos e 00 segundos (BRT), e que passou pelas seguintes alterações: flip vertical, rotação de 45 graus, *zoom out*, sem adição de ruído, sem alteração nos canais de cores, contraste altamente reforçado, nitidez levemente alterada, imagem sem ser transladada e 4% da imagem cortada.

Com a possibilidade de realizar o *tracking* dessas características das imagens, é possível fazer estudos mais detalhados sobre quais alterações são mais impactantes no treinamento e desempenho de inteligências artificiais.

3.2.2 Comunicação com a nuvem

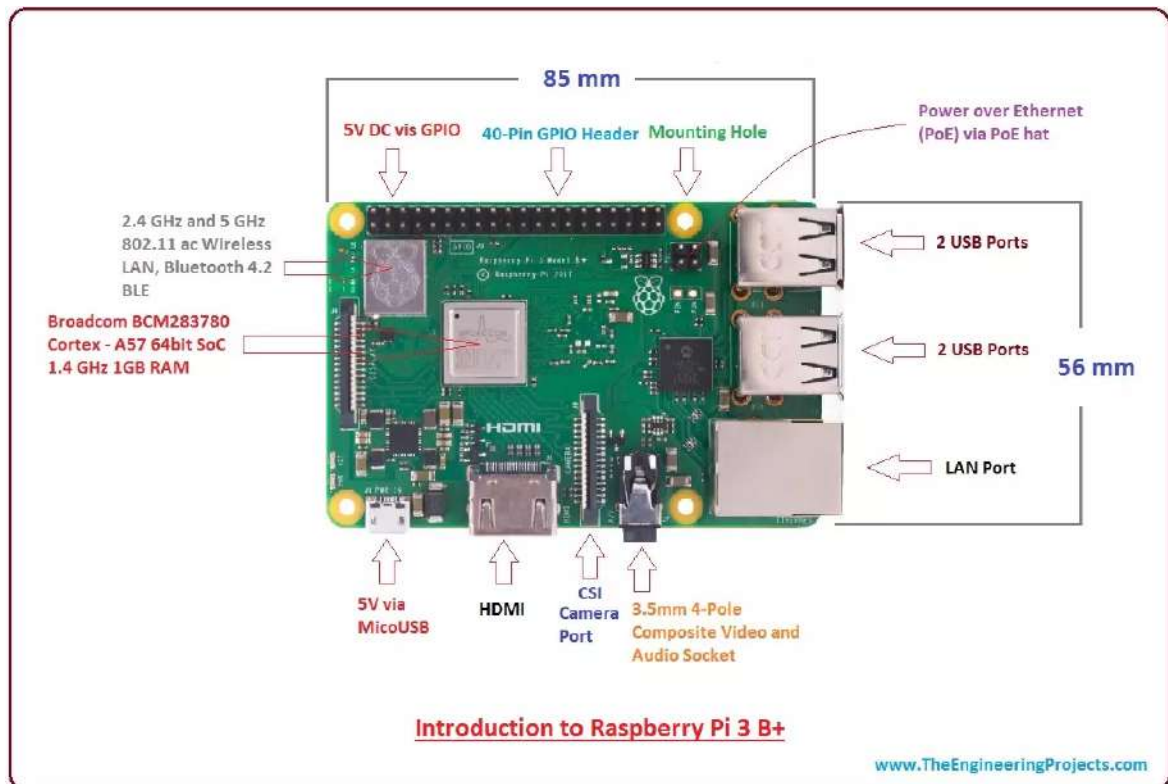
Com as imagens já adquiridas, renomeadas e aumentadas; foi elaborado um algoritmo para que essas imagens não ficassem apenas no armazenamento interno do hardware de captura. Utilizando a API do Google Drive (<https://console.cloud.google.com>) e credenciais de cliente OAuth 2.0, foi criada uma rotina de upload das imagens; em um determinado horário do dia, o algoritmo inicia o upload, pegando todas as imagens salvas em uma pasta interna e enviando-as para uma pasta na nuvem, no google drive, então, um arquivo de texto é atualizado para manter um log da data do último upload e as imagens que já tiveram seu upload realizado são movidas para outra pasta no armazenamento interno, evitando imagens duplicadas.

3.3 Hardware

Com a utilização da câmera do celular, ainda era necessário realizar algumas operações, como renomear, organizar em pastas e rotular as imagens obtidas. A fim de facilitar e agilizar esse processo, foi idealizada a utilização de um hardware dedicado. Duas possibilidades foram abordadas e tiveram seus protótipos elaborados, além de integrar a parte de software comentada na seção 3.2, originando uma aplicação embarcada simples e automatizada, mas que não chegou a ser utilizada na construção do *dataset* da versão 1 do projeto.

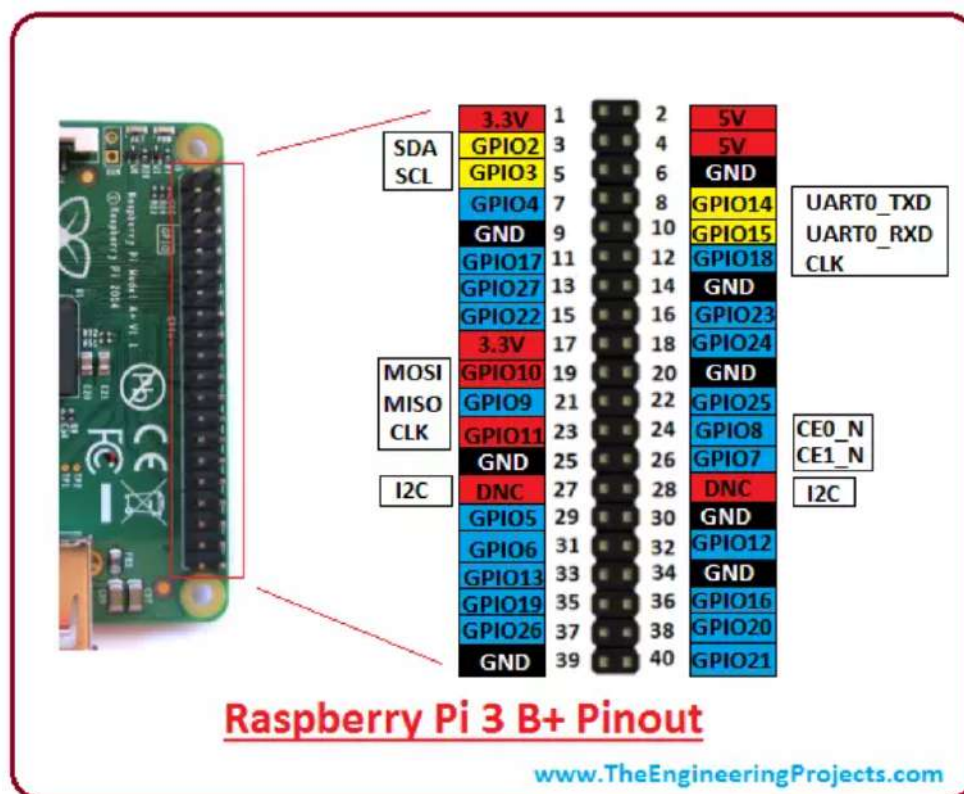
Nos dois sistemas desenvolvidos, utilizou-se o SBC Raspberry Pi 3 Model B+ como plataforma de processamento. A Figura 19 detalha as portas e outros detalhes técnicos da placa, como seu processador e as conexões Bluetooth e Wi-Fi. Já a Figura 20 apresenta o pinout do SBC com maiores detalhes, sendo as portas GPIO o que possibilita uma gama de automações utilizando outros componentes eletrônicos.

Figura 19 – Especificações do Raspberry Pi 3 Model B+



Fonte: Retirado de Nasir (2018)

Figura 20 – Pinout do Raspberry Pi 3 Model B+



Fonte: Retirado de Nasir (2018)

3.3.1 Raspberry Pi com Webcam C920

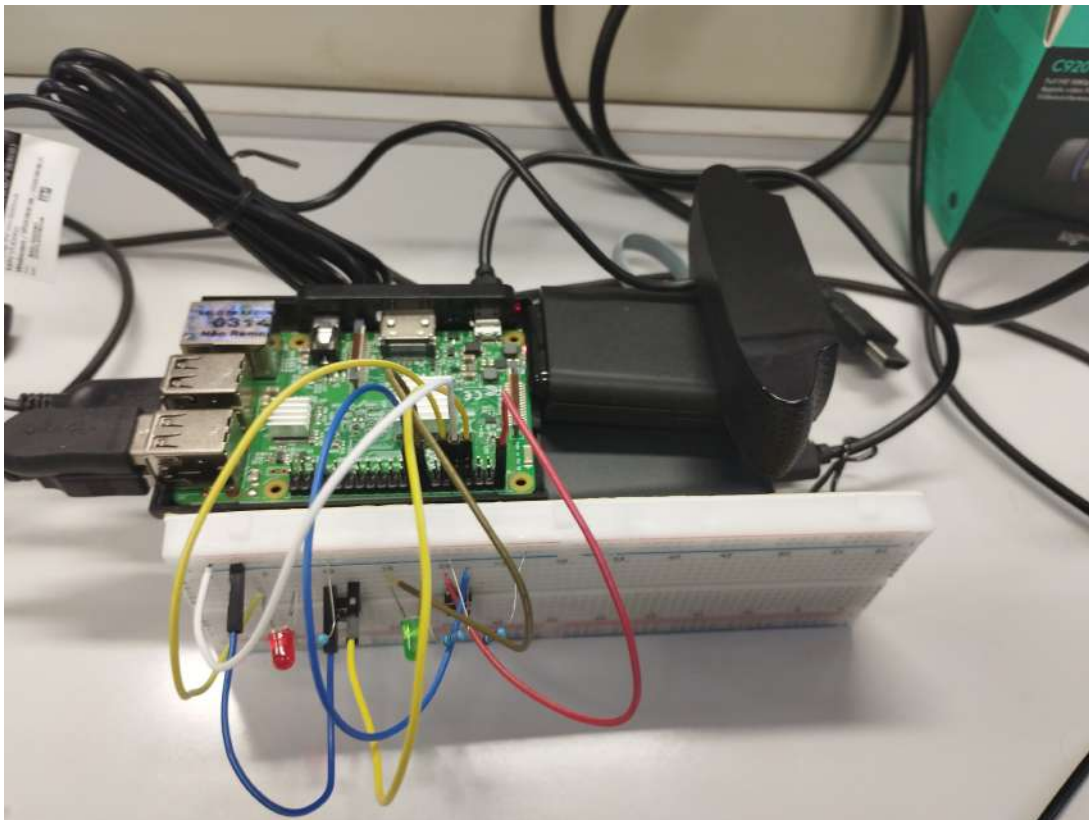
O primeiro protótipo foi desenvolvido a partir de uma câmera webcam Logitech C920, adquirindo imagens com uma resolução de 1920 x 1080. A webcam é conectada a uma das portas USB e sua instalação requer alguns passos simples no sistema operacional, como desativar o conector CSI e instalar alguns pacotes para a interação com esse modelo de câmera. A captura da imagem é realizada por uma sequência de instruções no prompt de comando (script de shell); e, através de um código em Python, quando o botão mais distante da câmera é pressionado, a imagem é capturada, acendendo o LED vermelho para confirmar a captura. As imagens podem passar pelo processo de nomenclatura e rotulagem de modo automático, como descrito na seção 3.2, e, estando conectadas à rede, são enviadas para pastas de armazenamento em nuvem conforme descrito em subseção 3.2.2.

Por padrão, o Raspberry liga automaticamente quando conectado a uma fonte de energia compatível e desliga por comandos no sistema ou quando a alimentação é interrompida. Como o módulo precisava funcionar de maneira remota, esse processo precisou ser modificado; um dos botões que aparece nas imagens, ao lado do LED verde, funciona para ligar e desligar a placa. Dessa maneira, o Raspberry só ligará após o botão ser pressionado, e o LED verde, que está mais próximo à câmera, enquanto aceso, indica

que o sistema está ligado. Nesse esquema, foi necessário editar alguns parâmetros de boot do SBC, incluindo as configurações para que os códigos em Python inicializassem junto ao boot do sistema operacional. É necessário esperar alguns segundos quando o Raspberry for ligado e também quando for desligado para que os programas sejam inicializados e as imagens salvas.

Para a alimentação de energia da placa, utilizou-se um Power Bank de 20.000 mAh, possibilitando autonomia suficiente para obter as imagens em campo, podendo ultrapassar 8 horas de funcionamento contínuo. Na Figura 21 apresenta-se o sistema já montado e funcionando.

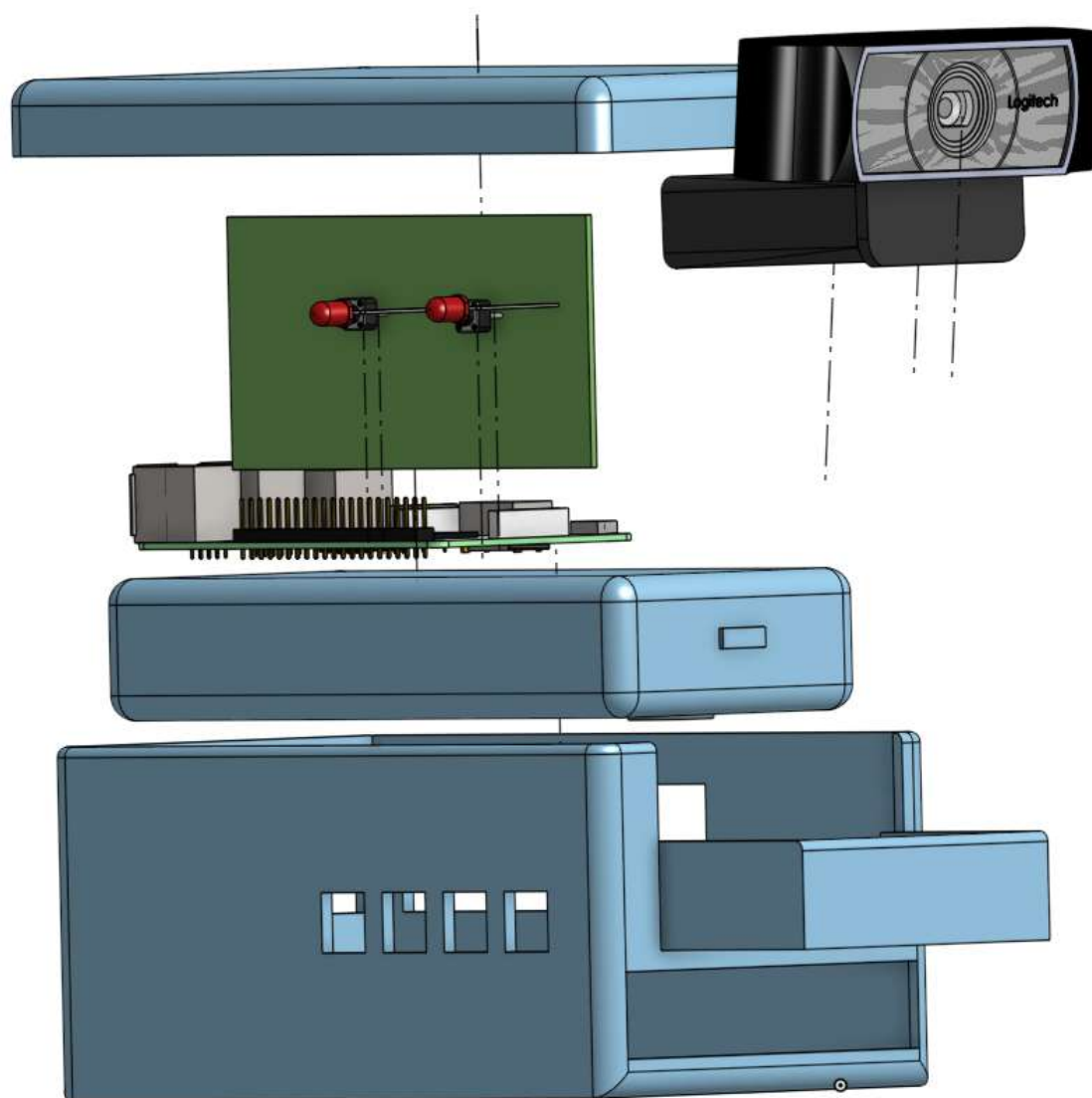
Figura 21 – Protótipo com webcam C920



Fonte: Elaborado pelo autor

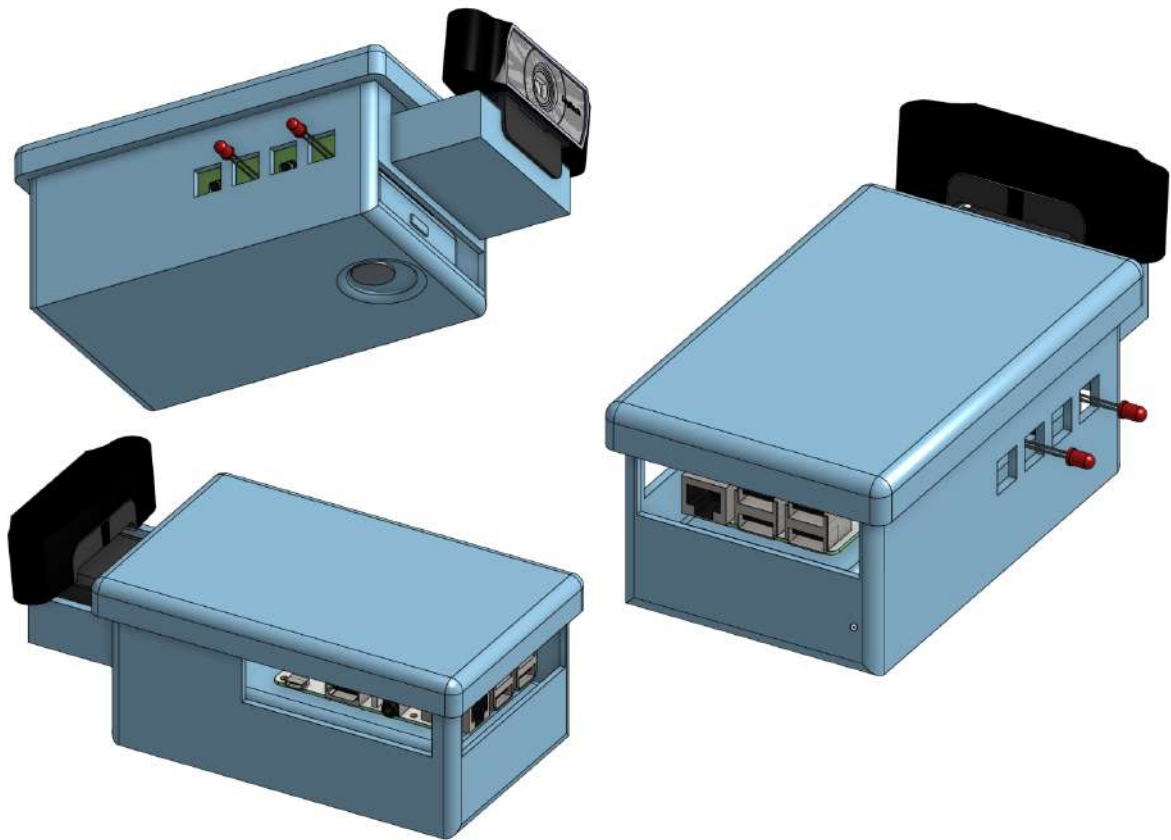
Para facilitar a utilização do sistema, foi desenvolvida uma case através da plataforma Onshape. O modelo da raspberry, (repository, 2023), foi retirado do domínio público do Onshape, e o CAD da câmera, (TASKIRAN, 2016), foi retirado da plataforma GrabCAD. Utilizando uma impressora 3D, a case foi impressa e o sistema integrado conforme as Figura 22 e Figura 23 apresentam. Utilizou-se uma placa de fenolite para fixar os componentes e foram deixadas algumas aberturas para a passagem de fios e conexões, como a de cabo de rede, necessário para a transmissão dos arquivos para o armazenamento em nuvem, conforme a subseção 3.2.2 descreve.

Figura 22 – Visão explodida do sistema



Fonte: Elaborado pelo autor

Figura 23 – CAD do sistema montado



Fonte: Elaborado pelo autor

3.3.2 Raspberry Pi com câmera modular V2.1

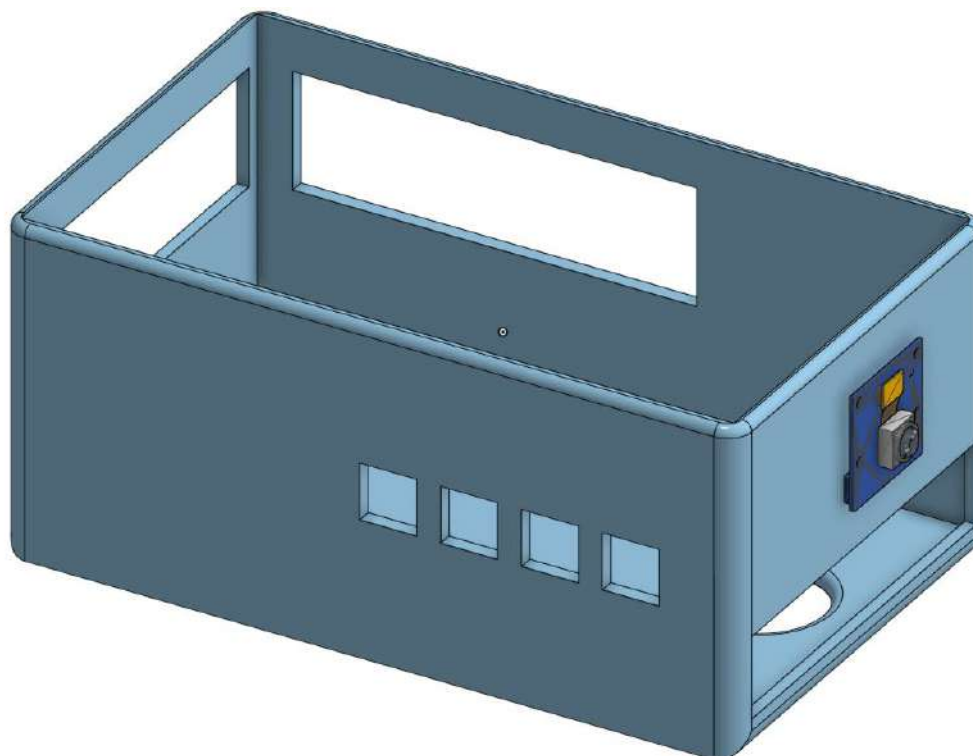
O segundo sistema abordado substitui a Webcam por uma câmera mais barata e menor, a câmera modular v2.1 dedicada à Raspberry, possuindo seu encaixe próprio na placa (CSI Camera Port na Figura 19). Esta câmera já foi utilizada em alguns estudos de agricultura de precisão, como em Kamath, Balachandra e Prabhu (2019), demonstrando sua viabilidade técnica. A instalação e utilização da câmera modular também são simplificadas em relação ao primeiro protótipo (subseção 3.3.1), não sendo mais necessário executar códigos .sh, apenas Python. O restante do hardware permaneceu o mesmo.

Figura 24 – Raspberry Pi 3 Modelo B+ com câmera modular v2.1



Fonte: Elaborado pelo autor

Figura 25 – Case adaptada para câmera modular



Fonte: Elaborado pelo autor

Por ser uma opção com maior portabilidade, menor custo e mais abordada no meio da agricultura de precisão, este segundo sistema de hardware apresenta múltiplas vantagens em relação ao primeiro sistema. Além das vantagens de hardware, as bibliotecas

desenvolvidas para a câmera própria do Raspberry facilitam novas implementações e testes de software, resultando em um sistema embarcado modular eficiente e escalável para aplicações agrícolas.

3.4 Construção dos protótipos

Por fim, os protótipos foram elaborados, como ilustram as imagens Figura 26 e Figura 27. Embora alguns detalhes, como a fixação dos componentes e a manipulação da *case*, tenham ficado aquém do ideal, trata-se de um protótipo funcional, capaz de contribuir para a construção e organização de *datasets* individuais e, principalmente, colaborativos, graças à integração do código que padroniza a nomenclatura dos arquivos e dos *labels*, exemplificado pela Figura 28.

Figura 26 – Protótipo utilizando Webcam C920



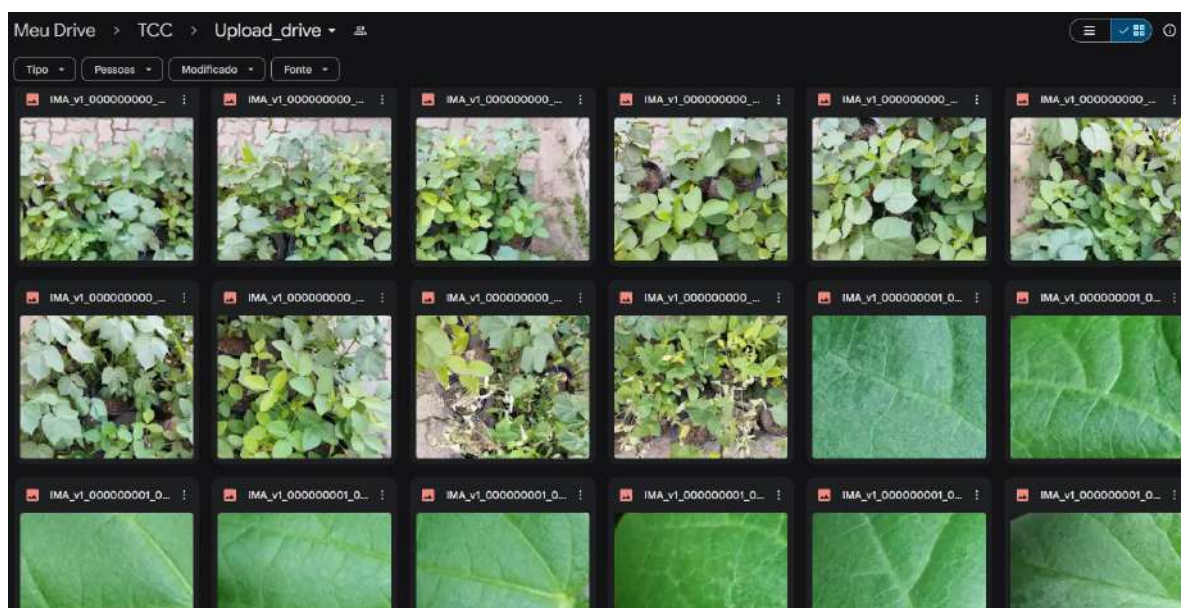
Fonte: Elaborado pelo autor

Figura 27 – Protótipo utilizando câmera modular



Fonte: Elaborado pelo autor

Figura 28 – Drive com imagens do banco de dados



Fonte: Elaborado pelo autor

4 CONCLUSÃO

4.1 Resultados

Este trabalho buscou soluções para padronizar a construção e a organização de *datasets* por meio de sistemas embarcados de baixo custo. Para isso, foram testadas duas estruturas de hardware com o Raspberry como unidade central de processamento: uma com a webcam C920 e outra com a câmera modular v2.1, na qual a câmera modular apresentou vantagens em relação ao custo e à implementação.

Também foram elaborados códigos para nomear as imagens de forma a apresentar informações relevantes e facilitar futuras manipulações e estudos. Além disso, todo o processo de obtenção, nomenclatura, *data augmentation* e armazenamento em nuvem foi automatizado por meio de códigos em python. Como resultado, tem-se um sistema embarcado simples, barato e reproduzível para elaborar *datasets* de maneira organizada e de fácil manipulação.

4.2 Trabalhos futuros

O estudo abre margem para múltiplos aprimoramentos, possibilitando, inclusive, sua utilização fora do meio de pesquisa, tanto em ambientes agrícolas quanto industriais. Pode-se, nesse contexto, substituir o sistema de alimentação por um mais adequado, permitindo a implementação de múltiplos sensores adicionais à câmera, sem que o Raspberry funcione em baixa potência; desenvolver uma nova case para o sistema; utilizar computação em nuvem para trabalhos em tempo real; e empregar sistemas de *data augmentation* mais complexos.

Além das melhorias técnicas do projeto, outras aplicações podem ser abordadas, como implementar o sistema em uma linha de produção, construindo bancos de dados organizados de novos produtos, permitindo o uso dessas imagens para o treinamento de inteligências artificiais, análises de dados e gestão da informação.

REFERÊNCIAS

- BARBEDO, J. G. A. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. **Computers and Electronics in Agriculture**, v. 153, p. 46–53, 2018. ISSN 0168-1699. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168169918304617>.
- BROWN, M. *et al.* Database design and implementation for quantitative image analysis research. **IEEE Transactions on Information Technology in Biomedicine**, v. 9, n. 1, p. 99–108, 2005.
- CHO, W.-J.; YANG, M. High-throughput plant phenotyping system using a low-cost camera network for plant factory. **Agriculture**, v. 13, n. 10, 2023. ISSN 2077-0472. Disponível em: <https://www.mdpi.com/2077-0472/13/10/1874>.
- CHOLLET, F. **Transfer learning fine-tuning**. 2023. Acesso em: 24 jul. 2024. Disponível em: https://keras.io/guides/transfer_learning/.
- CURTIN, B. H.; MATTHEWS, S. J. Deep learning for inexpensive image classification of wildlife on the raspberry pi. *In: 2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*. [S.l.: s.n.], 2019. p. 0082–0087.
- FREDRIKSSON, T. *et al.* Data labeling: An empirical investigation into industrial challenges and mitigation strategies. *In: MORISIO, M.; TORCHIANO, M.; JEDLITSCHKA, A. (ed.). Product-Focused Software Process Improvement*. Cham: Springer International Publishing, 2020. p. 202–216. ISBN 978-3-030-64148-1.
- GONZALEZ-HUITRON, V. *et al.* Disease detection in tomato leaves via cnn with lightweight architectures implemented in raspberry pi 4. **Computers and Electronics in Agriculture**, v. 181, p. 105951, 2021. ISSN 0168-1699. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168169920331562>.
- GUO, Y. *et al.* Deep learning for visual understanding: A review. **Neurocomputing**, Elsevier, v. 187, p. 27–48, 2016.
- JOICE, A. *et al.* Applications of raspberry pi for precision agriculture—a systematic review. **Agriculture**, MDPI, v. 15, n. 3, p. 227, 2025.
- KAMATH, R.; BALACHANDRA, M.; PRABHU, S. Raspberry pi as visual sensor nodes in precision agriculture: A study. **Ieee Access**, v. 7, n. 0, p. 45110–45122, 2019.
- KAMILARIS, A.; PRENAFETA-BOLDÚ, F. X. Deep learning in agriculture: A survey. **Computers and Electronics in Agriculture**, v. 147, p. 70–90, 2018. ISSN 0168-1699. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168169917308803>.
- KARAGIANNAKOS, S. **How to Use Data Augmentation to 10x Your Image Datasets**. 2020. Acesso em: 25 jun. 2024. Disponível em: <https://medium.com/data-science/how-to-use-data-augmentation-to-10x-your-image-datasets-dab42858be55>.

KITCHIN, R.; LAURIAULT, T. P. Small data in the era of big data. **GeoJournal**, Springer, v. 80, p. 463–475, 2015.

KUMAR, T. *et al.* Image data augmentation approaches: A comprehensive survey and future directions. **IEEE Access**, IEEE, 2024.

LIONBRIDGE. **Crowdsourcing Data Labeling for Machine Learning Projects [A How-To Guide]**. 2024. Acesso em: 12 abr. 2020. Disponível em: <https://hackernoon.com/crowdsourcing-data-labeling-for-machine-learning-projects-a-how-to-guide-cp6h32nd>.

MATHE, S. E. *et al.* A survey of agriculture applications utilizing raspberry pi. *In: 2022 International Conference on Innovative Trends in Information Technology (ICITIIT)*. [S.l.: s.n.], 2022. p. 1–7.

NASIR, S. Z. **Introduction to Raspberry Pi 3 B+**. 2018. Disponível em: <https://www.theengineeringprojects.com/2018/07/introduction-to-raspberry-pi-3-b-plus.html>.

NDUNG’U, R. Data preparation for machine learning modelling. **International Journal of Computer Applications Technology and Research**, v. 11, p. 231–235, 06 2022.

OYAMA, P. d. C.; JORGE, L. d. C.; GOMES, C. Sistema de visão computacional para classificação de grãos de café por cor e forma. **UFPR**, 2019.

RAVI, S.; LAROCHELLE, H. Optimization as a model for few-shot learning. *In: International conference on learning representations*. [S.l.: s.n.], 2017.

REPOSITORY, O. public. **raspberrypi3BSimplified**. 2023. Arquivo CAD. Acesso em: 07 mai. 2023. Disponível em: <https://cad.onshape.com/documents?nodeId=3&resourceType=filter>.

ROH, Y.; HEO, G.; WHANG, S. E. A survey on data collection for machine learning: A big data - ai integration perspective. **IEEE Transactions on Knowledge and Data Engineering**, v. 33, n. 4, p. 1328–1347, 2021.

SANTOS, T. *et al.* Visão computacional aplicada na agricultura. *In: _____*. [S.l.: s.n.]: Embrapa, 2020. p. 146–164. ISBN 978-65-86056-37-2.

SCHARR, H. *et al.* Annotated image datasets of rosette plants. *In: SUISSE ZÜRICH. European conference on computer vision*. [S.l.: s.n.], 2014. p. 6–12.

SHINDE, P. P.; SHAH, S. A review of machine learning and deep learning applications. *In: IEEE. 2018 Fourth international conference on computing communication control and automation (ICCUBEA)*. [S.l.: s.n.], 2018. p. 1–6.

SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. **Journal of Big Data**, v. 6, n. 1, p. 60, 2019. Disponível em: <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0>.

SHU, J.; XU, Z.; MENG, D. Small sample learning in big data era. **arXiv preprint arXiv:1808.04572**, 2018.

TASKIRAN, A. **Logitech C920 Webcam**. 2016. Arquivo CAD. Acesso em: 07 mai. 2023. Disponível em: <https://grabcad.com/library/logitech-c920-webcam-1>.

WANG, Y. *et al.* Generalizing from a few examples: A survey on few-shot learning. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 53, n. 3, jun. 2020. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/3386252>.

YANG, S. *et al.* Image data augmentation for deep learning: A survey. **arXiv preprint arXiv:2204.08610**, 2022.

APÊNDICES

APÊNDICE A – DICIONÁRIO DE AUMENTO DE DADOS

Figura 29 – Dicionário de transformações utilizadas no processo de *data augmentation*.

```
augmentation_map = {

    "flip":{0: "nenhum",1: "horizontal",2: "vertical",3: "ambos"},

    "rotacao":{0: "45°", 1: "120°", 2: "235°", 3: "330°"},

    "escala":{0: "nenhum", 1: "zoom_in", 2: "zoom_out"},

    "ruído":{0: "nenhum", 1: "leve", 2: "medio", 3: "forte"},

    "canais_cor":{0:"normal",1:"shuffle",2:"drop",3:"equalizacao_hist"},

    "contraste":{0: "original", 1: "baixo", 2: "medio", 3: "alto"},

    "nitidez":{0: "original", 1: "leve", 2: "medio", 3: "forte"},

    "translacao":{0: "nenhum", 1: "leve", 2: "medio", 3: "forte"},

    "crop":{0: "nenhum", 1: "4%", 2: "7%", 3: "10%"}

}
```


APÊNDICE B – CÓDIGOS DE NOMENCLATURA

Figura 30 – Trecho do código dedicado à seleção do projeto, versão, cultura e câmera

```
# =====
# CONFIGURAÇÕES INICIAIS
# =====

# Pasta a ser renomeada
pasta = r"D:\Matheus Civardi\Projeto IMAmt (TCC)\Imagens (dataset)"

# Nome do projeto
projeto = "IMA"

# Versão do código
versao = "v1"

# Versão da imagem (data augmentation)
data_augmentation = "000000000" # Sem modificações

# Código da cultura:
# 01 - Soja
# 02 - Algodão
# 03 - Soja e Algodão
# 04 - Outro
codigo_cultura = "02"

# Código da câmera:
# 01 - Raspberry Pi Camera
# 02 - Logitech
# 03 - 2201117TG (Celular)
codigo_camera = "03"

# =====
```