INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DE MINAS GERAIS (IFMG)
CAMPUS BAMBUÍ
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

André Dias Neto

DESENVOLVIMENTO DE UM SISTEMA PARA O POSICIONAMENTO AUTOMÁTICO DE BULBILHOS DE ALHO UTILIZANDO VISÃO COMPUTACIONAL

Bambuí – MG
2025

ANDRÉ DIAS NETO

DESENVOLVIMENTO DE UM SISTEMA PARA O POSICIONAMENTO AUTOMÁTICO DE BULBILHOS DE ALHO UTILIZANDO VISÃO COMPUTACIONAL

Trabalho de conclusão de curso apresentado ao Curso de Bacharelado em Engenharia de Computação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais (IFMG) – *Campus* Bambuí para obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Prof. Me. Calebe Giaculi Júnior

Coorientador: Prof. Me. Francisco Heider Willy dos Santos

Bambuí – MG
2025

Catalogação na Fonte Biblioteca IFMG - *Campus* Bambuí

D541d Dias Neto, André.

Desenvolvimento de um sistema para o posicionamento automático de bulbilhos de alho utilizando visão computacional [manuscrito] / André Dias Neto. – 2025.

51 f. : il.

Orientador: Calebe Giaculi Júnior.

Coorientador: Francisco Heider Willy dos Santos.

Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação) – Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais. *Campus* Bambuí, 2025.

1. Visão computacional. 2. Redes neurais convolucionais. 3. Plantio de alho. 4. Posicionamento automático. I. Giaculi Júnior, Calebe. II. Santos, Francisco Heider Willy dos. III. Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – *Campus* Bambuí. IV. Título.

CDD 006.37

Catalogação: João Batista Rodrigues - CRB-6/2022

André Dias Neto

DESENVOLVIMENTO DE UM SISTEMA PARA O POSICIONAMENTO AUTOMÁTICO DE BULBILHOS DE ALHO UTILIZANDO VISÃO COMPUTACIONAL

Trabalho de conclusão de curso apresentado ao Curso de Bacharelado em Engenharia de Computação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais (IFMG) – *Campus* Bambuí para obtenção do grau de Bacharel em Engenharia de Computação.

Aprovado em 05 de Fevereiro de 2025 pela banca examinadora:

Prof. Me. Calebe Giaculi Júnior – IFMG – *Campus* Bambuí – (Orientador)
Prof. Me. Francisco Heider Willy dos Santos – IFMG – Campus Bambuí – (Coorientador)
Prof. Carlos Renato Nolli – IFMG – Campus Bambuí
Prof. Dr. Marcos Roberto Ribeiro – IFMG – Campus Bambuí

Documento assinado eletronicamente por **Calebe Giaculi Junior**, **Professor**, em 05/02/2025, às 16:38, conforme Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Francisco Heider Willy dos Santos**, **Professor**, em 05/02/2025, às 16:39, conforme Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Carlos Renato Nolli**, **Professor**, em 05/02/2025, às 16:40, conforme Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Marcos Roberto Ribeiro**, **Professor**, em 05/02/2025, às 16:40, conforme Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **André Dias Neto**, **Usuário Externo**, em 05/02/2025, às 17:30, conforme Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade do documento pode ser conferida no site https://sei.ifmg.edu.br/consultadocs informando o código verificador **2188414** e o código CRC **FB8E8026**.

Aos meus amados pais, que me incentivaram a seguir meus sonhos, e aos meus irmãos, em especial, à Camila, por me apoiarem e trilharem esta jornada comigo, dedico este trabalho com muito amor e gratidão.

AGRADECIMENTOS

Agradeço primeiramente aos meus amados pais e irmãos, que, nos momentos de alegria ou dificuldade, sempre estiveram ao meu lado, me incentivando e me dando suporte; sem eles, nada disso seria possível.

Agradeço aos meus amigos, que sempre estiveram comigo em cada etapa dessa jornada, me apoiando, encorajando e me fazendo rir. Sou muito grato por ter vocês em minha vida. Sua companhia foi essencial nessa jornada.

Agradeço a cada professor que me inspirou, cedeu seu tempo e conhecimento e que me guiou ao longo desta jornada acadêmica.

Agradeço aos meus orientadores, Francisco Heider e Calebe Giaculi, pela paciência, apoio, sabedoria, confiança, motivação e orientação, que foram fundamentais em meu trabalho.

Agradeço também a todos que contribuíram, de alguma forma, para que esse momento fosse possível.

“Nunca desista de um sonho só por causa do tempo que você vai levar para realizá-lo. O tempo vai passar de qualquer forma.”

Autor Desconhecido

RESUMO

O alho está entre os condimentos mais populares do mundo, e, no Brasil, a maior parte do plantio é realizada de forma manual, exigindo uma grande demanda de mão de obra, elevando os custos de produção. Uma solução para reduzir os custos e aumentar a competitividade com o mercado internacional é utilizar a mecanização no plantio. Para atingir uma maior produtividade, o plantio da semente deve ser feito na posição correta, porém a maior parte das máquinas para o plantio de alho não atende ao requisito de posicionar a semente no solo. Nesse contexto, este trabalho apresenta o desenvolvimento de um sistema para o posicionamento automático de bulbilhos de alho, utilizando visão computacional. O objetivo é criar um protótipo que consiga identificar a posição em que a semente está e, através de imagens de vídeo em tempo real, fazer a correção da posição da semente antes do plantio. O trabalho aborda a importância de um plantio correto, com a raiz para baixo, e, se devidamente mecanizado, pode ajudar a reduzir os custos do plantio e maximizar a produção. O trabalho apresenta o desenvolvimento de um protótipo para o posicionamento das sementes de alho, que consiste na implementação de um mecanismo físico, eletrônico e de um software que utiliza processamento digital de imagens e redes neurais convolucionais para realizar a classificação dos bulbilhos de alho. Um conjunto de dados foi criado com imagens reais dos bulbilhos, sendo utilizado, para treinar, um modelo de rede neural convolucional (*MobileNetV2*) para identificar a região radicular e o ápice das sementes. O trabalho apresenta resultados com acurácia média de 90% na classificação das posições e demonstra a capacidade de realizar a detecção em tempo real no dispositivo, validando o projeto como uma solução promissora.

Palavras-chave: Visão computacional. Redes neurais convolucionais. Plantio de alho. Posicionamento automático.

ABSTRACT

Garlic is among the most popular condiments in the world, and in Brazil, most of its cultivation is carried out manually, requiring a large labor force and increasing production costs. One solution to reduce costs and enhance competitiveness in the international market is to adopt mechanized planting. To achieve higher productivity, the seed must be planted in the correct position; however, most garlic planting machines do not meet the requirement of properly positioning the seed in the soil. In this context, this work presents the development of a system for the automatic positioning of garlic cloves using computer vision. The objective is to create a prototype capable of identifying the seed's position and, through real-time video analysis, correcting its orientation before planting. The study highlights the importance of proper planting, with the root facing downward, which, if properly mechanized, can help reduce planting costs and maximize production. This work presents the development of a prototype for positioning garlic seeds, consisting of the implementation of a physical and electronic mechanism, along with software that utilizes digital image processing and convolutional neural networks to classify garlic cloves. A dataset was created with real images of garlic cloves, and a convolutional neural network model (*MobileNetV2*) was trained to identify the root region and the seed apex. The study presents results with an average classification accuracy of 90% and demonstrates the system's capability to perform real-time detection on the device, validating the project as a promising solution.

Keywords: Computer vision. Convolutional neural networks. Garlic planting. Automatic positioning.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE SIGLAS

ML – Machine Learning
RNA – Redes neurais artificiais
CNN – Convolutional Neural network
IA – Inteligência Artificial
PDI – Processamento digital de imagens
MLP – Multi-Layer Perceptron
FPS – Frames por Segundo
GPU – Graphics Processing Unit
3D – Três dimensões (Tridimensional)
ton/ha – Toneladas por hectare
ReLU – Unidade Linear Retificada
FC – Fully Connected
DA – Data augmentation
RGB – Red, Green, Blue

SUMÁRIO

1 INTRODUÇÃO

Este capítulo apresenta uma introdução do tema de estudo da presente monografia, expondo os principais pontos a serem abordados ao longo do documento. Serão discutidos a contextualização do tema, os objetivos geral e específicos, os resultados esperados, a justificativa para a realização deste estudo e a estrutura do trabalho.

1.1 Contextualização

O alho está entre os condimentos mais populares do mundo. No Brasil, em média, é consumido cerca de 1,5 kg/pessoa/ano. Grande parte do alho consumido é importada, sendo o Brasil o segundo maior importador de alho do mundo, mas o décimo terceiro maior produtor mundial. A maior parte do plantio de alho no Brasil é feita de forma manual, exigindo alta demanda de mão de obra. Cerca de 30% dos custos são com mão de obra, e outro fator que contribui para esse valor elevado é a escassez de mão de obra especializada. Uma solução para reduzir os custos e incrementar a competitividade com o mercado internacional é utilizar a mecanização no plantio (GRÜNDLING; GAZZOLA; ARAGÃO, 2021)

Para se atingir um maior potencial de produtividade e qualidade, o plantio da semente de alho deve ser feito com a raiz posicionada para baixo. Segundo Meneguzzo (2020), os erros causados no posicionamento da semente podem gerar perdas de até 40%. Um dos principais desafios ao utilizar a mecanização no plantio de alho se deve à grande dificuldade de padronização na classificação das sementes, devido à variedade de suas características, como tamanho, peso e formato.

A maioria das máquinas desenvolvidas para o plantio de alho não atende ao requisito de posicionamento das sementes no solo, o que pode gerar prejuízos quando utilizada a mecanização no plantio, sendo necessário o desenvolvimento de um sistema que garanta este posicionamento correto. (SCHMIDT, 1997).

Neste contexto, este trabalho propõe o desenvolvimento de um protótipo para o posicionamento da semente de alho. O sistema integra componentes mecânicos, eletrônicos e um software que utilizará imagens capturadas por uma câmera, para analisar a posição do bulbilho de alho. Com base nessa análise, o protótipo deve realizar o posicionamento dos bulbilhos quando necessário, garantindo que sua raiz fique orientada para baixo. O protótipo desenvolvido neste trabalho foi avaliado quanto à sua eficiência e velocidade e poderá ser integrado, futuramente, a um mecanismo automático de plantio de alho.

1.2 Objetivos

Essa seção tem como objetivo definir os objetivos desta pesquisa, estabelecendo as metas a serem alcançadas para o estudo em questão. O objetivo geral desta pesquisa expressa a finalidade principal do estudo, enquanto os objetivos específicos definem as etapas a serem realizadas para se alcançar o objetivo geral e os resultados esperados. Esses objetivos guiaram o desenvolvimento e as análises conduzidas ao longo deste trabalho.

1.2.1 Objetivos geral

O objetivo geral deste trabalho foi desenvolver um sistema para o posicionamento automático de bulbilhos de alho utilizando visão computacional.

1.2.2 Objetivos específicos

- desenvolver um mecanismo que permita girar o bulbilho de alho na posição correta para o plantio;
- implementar e avaliar um algoritmo utilizando processamento digital de imagens e inteligência artificial para classificação da posição do bulbilho de alho;
- implementar um algoritmo para corrigir a posição do bulbilho de alho, controlando atuadores para o posicionamento correto.

1.3 Resultados esperados

Com a realização deste trabalho, espera-se que seja desenvolvido um protótipo do sistema proposto, capaz de identificar a posição do bulbilho de alho e realizar o posicionamento correto para o plantio, reduzindo as perdas de produtividade devido ao plantio na posição incorreta das sementes.

O desenvolvimento deste sistema para o posicionamento automático de bulbilhos de alho busca contribuir para um avanço na mecanização agrícola. A utilização de visão computacional e redes neurais convolucionais permite reduzir erros no plantio, podendo contribuir para uma maior produtividade e qualidade na colheita.

Além disso, a automatização do plantio pode diminuir os custos, reduzindo a dependência de mão de obra, incrementando a competitividade com o mercado internacional. Essa inovação, além de beneficiar a cultura do alho, pode também abrir caminho para futuras aplicações em outras culturas agrícolas. Assim, este trabalho busca contribuir para o aprimoramento das técnicas de plantio, otimizando recursos e

impulsionando a mecanização agrícola.

1.4 Justificativa

Este trabalho busca avaliar uma possível solução baseada em processamento de imagem para o problema da variação do formato das sementes que geram erros no posicionamento mecânico do bulbilho de alho. Dada a capacidade de interpretação e generalização que as redes neurais convolucionais têm demonstrado em avaliação de sementes, espera-se que essa abordagem permita o posicionamento eficiente do bulbilho.

Do ponto de vista tecnológico, o presente trabalho propõe o desenvolvimento de um sistema para o posicionamento automático de bulbilhos de alho, que analisará a posição da semente por meio de técnicas de visão computacional e, quando necessário, corrigirá a posição das sementes por meio de atuadores eletrônicos.

Os resultados da avaliação do mecanismo proposto poderão contribuir para o desenvolvimento de novos mecanismos de plantio de alho, o que poderá levar a um aumento na produtividade, qualidade e redução dos custos de produção.

Este trabalho aborda vários conteúdos do curso de engenharia de computação, e a aplicação prática desses conteúdos será de grande importância para a consolidação dos conhecimentos do aluno.

1.5 Estrutura do documento

Este documento foi organizado em cinco capítulos, cada um abordando aspectos específicos relacionados ao tema de estudo.

No primeiro capítulo, apresentam-se a introdução, a contextualização do tema, os objetivos geral e específicos, os resultados esperados e a justificativa para a realização deste trabalho.

No segundo capítulo, abordam-se a fundamentação teórica, o referencial teórico e o estado da arte, relacionados ao tema de estudo.

No capítulo seguinte, é detalhada a metodologia utilizada, descrevendo a classificação da pesquisa, a solução proposta e os materiais e métodos empregados na realização do trabalho.

O quarto capítulo aborda os resultados e discussão, apresentando os resultados obtidos a partir da aplicação da metodologia, materiais e métodos descritos no capítulo três. Os resultados obtidos são analisados e discutidos, sendo relacionados aos objetivos propostos anteriormente.

O quinto capítulo é o último da monografia, no qual se exibem as conclusões do trabalho. São discutidos os resultados e as possíveis contribuições relevantes

para a área de conhecimento do estudo e também possíveis trabalhos futuros.

Além dos capítulos citados, esta monografia conta com uma seção de referências bibliográficas, em que todas as obras utilizadas como base para o seu desenvolvimento podem ser consultadas.

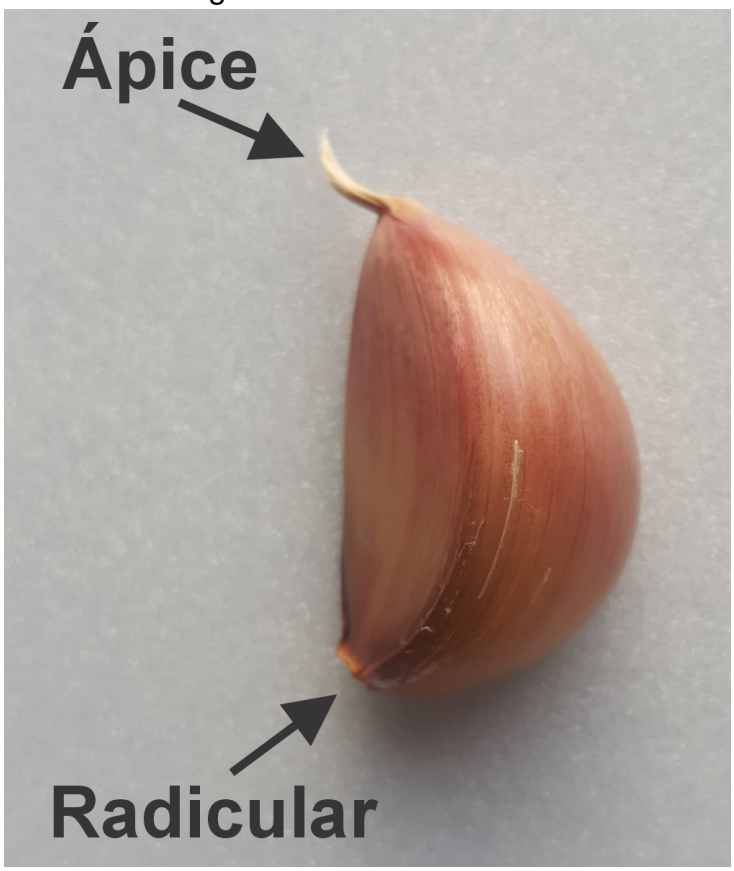
2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados o referencial teórico e o estado da arte relacionados ao tema de estudo. São detalhados todos os conceitos, fundamentos e trabalhos correlatos que contribuem para a base teórica necessária à compreensão do trabalho. Tem como objetivo apresentar os principais estudos e referências relevantes para embasar e contextualizar a pesquisa.

2.1 Plantio de alho

Os bulbilhos de alho (dentes) são compridos, de formato oval e arqueado, envolto por folhas protetoras, chamadas brácteas, que podem ter coloração branca, vermelha, violeta, roxa e marrom (RESENDE; HABER; PINHEIRO, 2015). A orientação da posição do bulbilho se dá pelas extremidades da semente, denominadas ápice e radicular, como mostrado na Figura 1

Figura 1 – Bulbilho de alho



Fonte: Elaborado pelo autor, 2025.

2.1.1 Plantio manual

Tradicionalmente, no plantio do alho, é preferível que os bulbilhos sejam plantados de forma que o ápice fique para cima, facilitando a germinação e contribuindo para uma maior produtividade (MENEGUZZO, 2020). Segundo Menezes Sobrinho *et al.* (1993), o plantio manual consiste na abertura do solo, incorporação do adubo, inserção e cobertura dos bulbilhos de alho no solo. Para este sistema de plantio, são gastas, em média, 370 horas de mão de obra por hectare com o espaçamento normal, de 25 a 30 cm entre linhas e de 8 a 10 cm entre plantas. O plantio manual de alho exige que o funcionário trabalhe em uma posição que compromete sua ergonomia, como pode ser visto na Figura 2. A repetição de movimentos com a postura inadequada durante o plantio pode causar danos à saúde do trabalhador, podendo resultar em lesões e problemas relacionados à ergonomia.

Figura 2 – Plantio manual de alho

Fonte: RESENDE; HABER; PINHEIRO, 2015.

2.1.2 Plantio mecanizado

A maior parte do plantio no Brasil é feito de forma manual, elevando os custos de produção, sendo que cerca de 30% são com mão de obra. Muitos produtores avaliam migrar para o plantio mecanizado, como uma maneira de reduzir os custos de produção, devido à escassez de mão de obra especializada necessária para o plantio do alho (RESENDE; HABER; PINHEIRO, 2015). Uma solução para reduzir os custos e incrementar a competitividade com o mercado internacional é utilizar a mecanização no plantio (GRÜNDLING; GAZZOLA; ARAGÃO, 2021). O plantio mecanizado de alho é comumente realizado por meio de implementos agrícolas que, geralmente, não atendem ao requisito de posicionamento correto das sementes no solo. Alguns desses equipamentos não possuem um sistema de posicionamento, resultando no depósito

das sementes em posição aleatória no solo.

2.2 Processamento digital de imagens

O interesse pelos métodos de processamento digital de imagens (PDI) tem como base duas áreas de aplicação, a melhora das informações visuais, facilitando a interpretação humana, e o processamento de dados de imagens, seja para armazenamento, transmissão ou representação, considerando a percepção automática por computador (GONZALEZ; WOODS, 2010). Segundo Gonzalez e Woods (2010), uma das primeiras aplicações das imagens digitais foi realizada pela indústria dos jornais, em 1920, que utilizava um cabo submarino para transmitir imagens entre Londres e Nova York (sistema Bartlane). O tempo necessário para transportar uma imagem foi reduzido de uma semana para menos de 3 horas.

Na década de 1960, houve um grande impulso na área de processamento de imagens, com o surgimento dos primeiros computadores digitais de grande porte e o início do programa espacial norte-americano. Teve início, em 1964, no Jet Propulsion Laboratory (Laboratório de propulsão a jato), o uso de técnicas de aprimoramento de imagens para corrigir vários tipos de distorção nas imagens da lua transmitidas pela câmera acoplada à sonda Ranger. Estas técnicas serviram de base para outros métodos de processamento de imagens utilizados em outros programas espaciais, como a missão Apollo (MARQUES FILHO; VIEIRA NETO, 1999).

Segundo Marques Filho e Vieira Neto (1999), um sistema de processamento de imagens pode ser representado pelo diagrama da Figura 3, que abrange as principais operações que se pode efetuar em uma imagem. Esse diagrama permite representar um sistema computadorizado capaz de adquirir, processar e interpretar imagens. Suas etapas são explicadas a seguir.

Primeiramente, devem ser definidos o domínio do problema e quais requisitos devem ser atendidos.

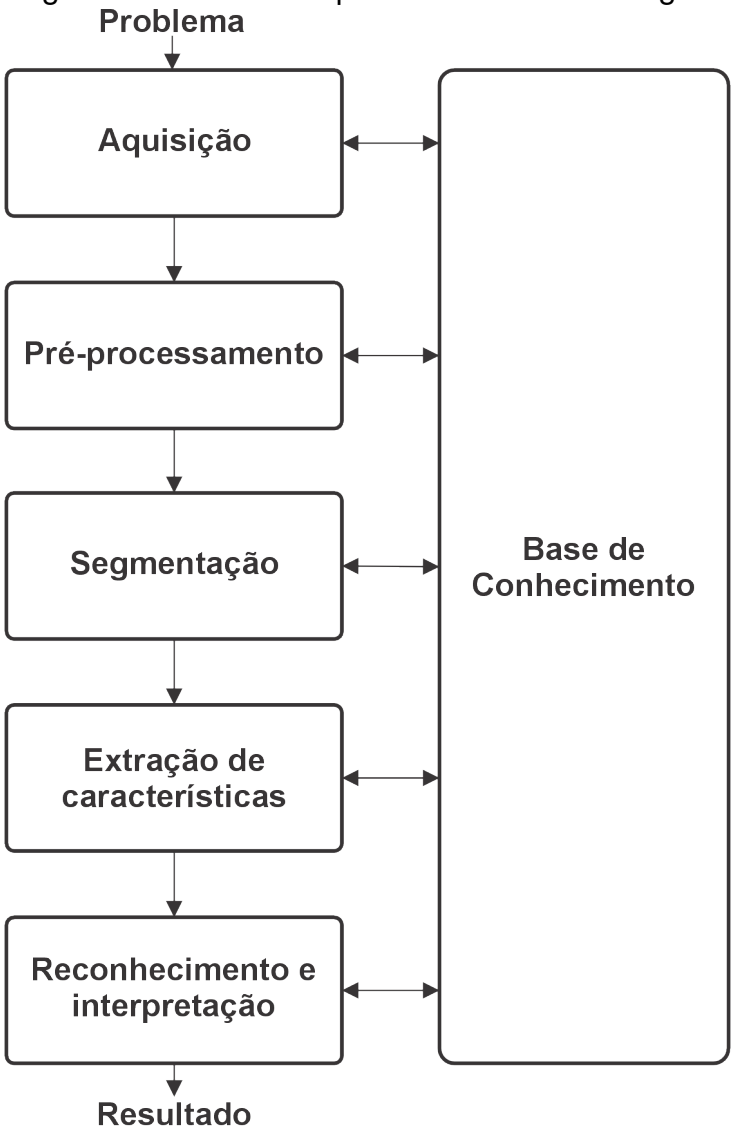
A primeira etapa é a de aquisição, que tem como função converter uma imagem em uma representação numérica adequada para o processamento, por exemplo, uma matriz de pixels.

A etapa de pré-processamento recebe a imagem, que pode apresentar imperfeições, e faz um aprimoramento da sua qualidade, resultando em uma imagem digitalizada de melhor qualidade que a original.

A etapa de segmentação tem, basicamente, a tarefa de dividir a imagem nos objetos de interesse que a compõem, sendo uma das mais importantes e difíceis de implementar.

A etapa de extração de características recebe as imagens resultantes da segmentação e busca fazer a extração de características-alvo por meio de descritores

Figura 3 – Sistema de processamento de imagens



Fonte: Adaptado de Marques Filho e Vieira Neto (1999).

que devem ser representados por uma estrutura de dados adequada ao algoritmo de reconhecimento. A entrada desta etapa é uma imagem, mas a saída é um conjunto de dados correspondente a ela.

Na etapa de reconhecimento, é realizado o processo de atribuição de um rótulo a um objeto, de acordo com suas características, e a interpretação atribui significado ao conjunto de objetos reconhecidos.

Todas as etapas estão conectadas a uma base de conhecimento sobre o problema a ser resolvido, a qual deve guiar o funcionamento de cada etapa e, idealmente, deveria permitir a integração entre elas (MARQUES FILHO; VIEIRA NETO, 1999).

2.3 Aprendizado de máquina

O aprendizado de máquina (*Machine Learning, ML*) é uma área da inteligência artificial que utiliza algoritmos e modelos estatísticos para que, a partir de dados, sistemas de computador possam tomar decisões e aprender por meio de exemplos, sem que haja a necessidade de serem programados para cada tarefa. Segundo Ludermir (2021), as técnicas de *ML* são orientadas a dados, possibilitando que os sistemas aprendam de forma automática a partir de grandes volumes de dados.

No aprendizado de máquina, se destacam três tipos de aprendizado: supervisionado, não supervisionado e por reforço. De acordo com Ludermir (2021), podemos definir esses tipos da seguinte maneira:

Aprendizado supervisionado: nesta abordagem, é utilizado um conjunto de dados rotulados para realizar o treinamento, ou seja, o modelo recebe os dados de entrada para treinamento e suas respectivas saídas, isso permite que o modelo aprenda a mapear as entradas para as saídas. O objetivo do algoritmo é ser capaz de classificar e determinar corretamente a classe de novos dados ainda não rotulados.

Aprendizado não supervisionado: nesta abordagem, é fornecido ao modelo um conjunto de dados não rotulados. O modelo analisa os dados recebidos e os agrupa pelas suas similaridades. Após os dados serem agrupados, pode ser necessária realizar uma análise para determinar o que cada agrupamento representa no contexto do problema analisado.

Aprendizado por reforço: nesta abordagem, o modelo não recebe a saída esperada, mas um reforço baseado em recompensas e punições. O modelo realiza uma hipótese baseada nos exemplos e determina se foi correta ou errada. Essa abordagem é bastante utilizada em jogos e controle de robôs.

Nem sempre é fácil solucionar problemas com o uso do Aprendizado de Máquina, é preciso seguir alguns pré-requisitos e utilizar um bom conjunto de dados, sendo que a qualidade dos dados pode influenciar diretamente no resultado que o modelo entregará. Se a base não tiver dados com boa qualidade, é necessário utilizar técnicas para aumentar a qualidade deles. Para cada problema analisado, devem ser selecionados os modelos mais adequados para resolver, de forma mais efetiva, o problema em questão. É preciso definir os parâmetros necessários para que o modelo possa atingir os resultados esperados. Após o treinamento, é preciso fazer uma validação para verificar se o modelo é capaz de resolver o problema com a precisão esperada.

2.4 Redes neurais artificiais

As redes neurais artificiais (*RNAs*) têm suas origens em pesquisas realizadas desde a década de 1940, com a construção do primeiro modelo de neurônio artificial proposto por McCulloch e Pitts, inspirado nos neurônios biológicos e no sistema nervoso. Porém, atualmente, esses dois sistemas não possuem grandes semelhanças (BARRETO, 2002). Conforme mencionado por Barreto (2002), pode-se entender uma rede neural artificial como uma abordagem para resolver problemas de inteligência artificial (IA).

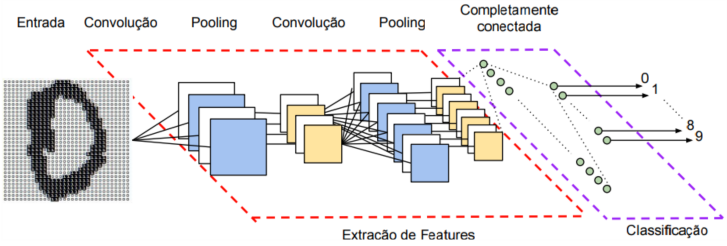
Segundo Barreto (2002), em uma *RNA*, o objetivo é construir circuitos neurais artificiais que possam se auto-organizar, e, diante de ambientes diversos, terem a capacidade de aprender novas tarefas, cometer erros, fazer generalizações e descobertas, tornando-se um sistema com comportamento inteligente.

Os sistemas de *RNA* são considerados paralelos e distribuídos. Essa característica se deve ao fato da estrutura das *RNAs* serem constituídas por um conjunto de neurônios interconectados trabalhando simultaneamente para processar dados (ROSSINI JUNIOR; CAMOLESI, 2018).

2.5 Redes neurais convolucionais

As redes neurais convolucionais (*CNNs*) têm aumentado sua popularidade cada vez mais. Uma das razões pelas quais a popularidade das *CNNs* tem aumentado mundialmente se deve à sua eficiência no processo de reconhecimento de imagens. Grandes avanços têm sido impulsionados por diversos centros de pesquisa no campo da visão computacional, que possui diversas aplicações, como veículos autônomos, robótica, drones, segurança, saúde e aplicações industriais (OLIVEIRA *et al.*, 2019). A Figura 4 mostra um exemplo de *CNN*. Esse tipo de rede vem sendo utilizado principalmente nas aplicações de classificação, detecção e reconhecimento de diversas classes em imagens e vídeos.

Figura 4 – Exemplo de uma rede neural convolucional e suas camadas



Fonte: Adaptado de Vargas, Carvalho e Vasconcelos (2016).

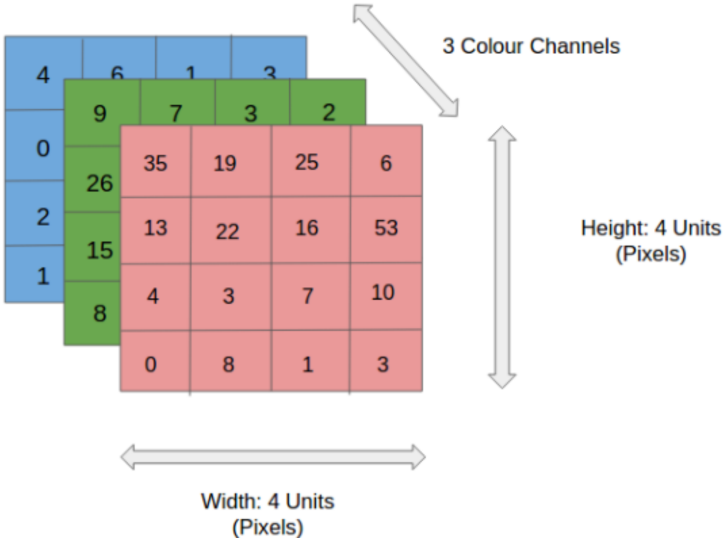
Uma rede neural convolucional (*CNN*) é um algoritmo de *deep learning* (aprendizado profundo) que pode captar uma imagem, atribuir importância a vários

objetos e aspectos da imagem, sendo capaz de diferenciá-los. Em comparação com outros algoritmos de classificação, uma *CNN* exige muito menos pré-processamento. Com treinamento suficiente, as *CNNs* têm capacidade de aprender filtros e características, não sendo necessária a criação manual como em métodos primitivos. Sua arquitetura é similar à do padrão de conectividade de neurônios no cérebro humano, inspirada na organização visual do córtex cerebral. Individualmente, os neurônios respondem a estímulos apenas em uma região do campo visual, conhecida como Campo Receptivo. Para cobrir totalmente a área visual, um conjunto desses campos se sobrepõe (DATA SCIENCE ACADEMY, 2022).

Segundo Data Science Academy (2022), uma *CNN* é capaz de capturar, através da aplicação de filtros, as dependências espaciais e temporais em uma imagem. A arquitetura realiza um melhor ajuste ao conjunto de dados da imagem devido à quantidade reduzida de parâmetros e à reutilização dos pesos. A rede pode ser treinada para ter um melhor entendimento da imagem.

Na Figura 5, podemos observar uma imagem *Red, Green, Blue* (*RGB*) que foi separada em três planos coloridos: vermelho, verde e azul. As imagens podem ser representadas em diversos espaços de cores, como escala de cinza, RGB, HSV e CMYK, entre outros.

Figura 5 – Imagem RGB



Fonte: Adaptado de Data Science Academy (2022).

A função da *CNN* é reduzir as imagens de uma forma que seja mais fácil de processar, sem perder recursos que são importantes para se obter uma boa previsão. Isso é importante quando se pretende projetar uma arquitetura que seja suficientemente boa em recursos de aprendizado e também escalável para grandes conjuntos de dados (DATA SCIENCE ACADEMY, 2022).

Antes das *CNNs*, para identificar objetos em imagens, eram utilizados métodos manuais e demorados para a extração de características. Atualmente, as redes

neurais convolucionais oferecem uma abordagem mais escalável para estas tarefas de classificação de imagens e reconhecimento de objetos, utilizando operações de convolução e *pooling* para aprender a representar dados de forma automática e eficiente. As *CNNs* utilizam princípios da álgebra linear, especialmente a multiplicação de matrizes, para identificar padrões e características relevantes dentro das imagens. Apesar da eficiência, o treinamento das *CNNs* pode ter um alto custo computacional, exigindo recursos de hardware, como unidades de processamento gráfico (*GPUs*), para acelerar o processo (IBM, 2024).

Segundo IBM (2024), as redes neurais convolucionais podem se distinguir de outras redes neurais por terem um desempenho superior ao processar imagens, fala ou sinal de áudio. As *CNNs* possuem três tipos principais de camadas: a camada convolucional (*Convolutional Layer*), a camada de agrupamento (*Pooling Layer*) e a camada totalmente conectada (*Fully Connected Layer – FC Layer*). Em uma rede convolucional, a camada convolucional é a primeira, podendo ser seguida por outras camadas convolucionais, ou camadas de agrupamento, e a camada totalmente conectada é a final.

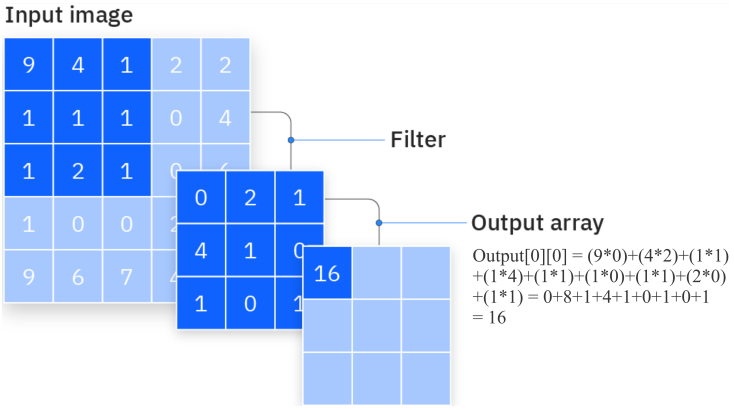
2.5.1 Camada convolucional

O bloco de construção central de uma *CNN* é a camada convolucional; é nela onde ocorre a maioria dos cálculos. Essa camada exige alguns componentes, como os dados de entrada, um filtro e um mapa de feições. Para compreender o funcionamento das camadas convolucionais, podemos considerar uma imagem colorida como entrada, representada por uma matriz 3D de *pixels*. Essa matriz possui três dimensões: altura, largura e profundidade, correspondendo à *RGB* em uma imagem. O processo de convolução envolve a aplicação de um detector de feições, conhecido também como *kernel* ou filtro, que se move pela imagem para identificar padrões visuais. Este processo é repetido até que o *kernel* tenha varrido toda a imagem. O resultado final da passagem do filtro é conhecido como mapa de feição (IBM, 2024).

De acordo com IBM (2024), a cada operação de convolução, a rede neural convolucional aplica uma transformação de Unidade Linear Retificada (*ReLU*) no mapa de feição, o que introduz uma não linearidade ao modelo. Na Figura 6, podemos ver um exemplo de aplicação de filtro.

De acordo com Vargas, Carvalho e Vasconcelos (2016), logo após a camada de convolução, é comum aplicar uma função de ativação, a qual está presente em cada neurônio e é responsável por aplicar uma transformação nos dados recebidos. Costumam-se utilizar funções com algum grau de não linearidade, o que permite que as aplicações dessas distorções tornem as categorias de saída linearmente separáveis.

Figura 6 – Aplicação de um filtro a uma imagem



Fonte: Adaptado de IBM (2024).

2.5.2 Camada de agrupamento

A camada de agrupamento (*pooling*) é muito importante e é utilizada, geralmente, após as camadas de convolução e ativação. Tem como função a redução da dimensionalidade dos dados da rede (VARGAS; CARVALHO; VASCONCELOS, 2016).

As camadas de *pooling* conduzem a redução de dimensionalidade, diminuindo o número de parâmetros na entrada. Semelhantemente à camada convolucional, a operação de *pooling* passa um filtro por toda a entrada, porém ele não tem peso. O *kernel* é um tipo de filtro que aplica uma função aos valores dentro do campo receptivo, preenchendo a matriz de saída. Dentro da camada de *pooling*, muitas informações são perdidas, mas essa camada tem vários benefícios para a *CNN*, como ajudar a reduzir a complexidade, melhorar a eficiência e diminuir o risco de superajuste (IBM, 2024).

2.5.3 Camada totalmente conectada

De acordo com IBM (2024), na camada *fully connected* (*FC*), cada nó da camada de saída se conecta a um nó da camada anterior diretamente. Ela é responsável por realizar a tarefa de classificação baseada nas feições extraídas através das camadas anteriores. As camadas convolucionais e de pooling costumam usar funções *ReLU*, já as camadas *FC* tendem a utilizar uma função de ativação (*SoftMax*), para classificar as entradas de forma adequada, resultando em saídas do tipo 0 ou 1.

A camada *FC* é responsável por traçar um caminho de decisão utilizando como base as respostas obtidas dos filtros das camadas anteriores para cada classe. Depois da camada totalmente conectada, o último passo é a função de classificação. Essa camada tem papel fundamental no treinamento, pois influencia diretamente no aprendizado dos filtros e no resultado da rede. A função *SoftMax*, por possuir bons

resultados e simplicidade, pode ser utilizada nessa etapa, contribuindo para um treinamento mais rápido e sem perda de qualidade (VARGAS; CARVALHO; VASCONCELOS, 2016).

2.6 Estado da arte

Yang *et al.* (2022) descrevem a aplicação de uma Rede Neural Convolucional (*CNN*) para a detecção de objetos em equipamentos de corte de raízes de alho. A metodologia envolveu o treinamento da *CNN* com um conjunto de dados contendo imagens de alho e outras raízes, bem como a implementação da *CNN* em um sistema para classificar e separar os alhos cortados. Os resultados mostraram que a *CNN* foi capaz de detectar, com precisão, os alhos e separá-los dos outros tipos de raízes, com uma taxa de acerto superior a 95%. Os autores concluem que essa tecnologia pode melhorar significativamente a eficiência e precisão do processo de corte e classificação das raízes, o que pode ter implicações importantes na indústria agrícola.

Anh, Thuyet e Kobayashi (2022) apresentaram um estudo que consistiu em utilizar uma rede neural convolucional profunda para classificar imagens de alho após o processo de corte das raízes. Foram criados dois modelos: um multiclasse, para classificar quatro tipos de alho (bom, ruim, não cortado e danificado), e um modelo multirrótulo, para classificar duas características do alho (limpo ou sujo). Os resultados mostraram que ambos os modelos tiveram desempenho satisfatório na classificação das imagens, com acurácia média de 92% para o modelo multiclasse e 96% para o modelo multirrótulo.

Oliveira *et al.* (2019) realizaram um estudo que comparou a eficácia de duas técnicas para classificação de grãos de café: redes neurais convolucionais e processamento digital de imagens com *MLP*. Foram realizadas 30 execuções para cada algoritmo, variando os conjuntos de treinamento e teste, com o objetivo de obter um valor médio da acurácia e seu respectivo desvio padrão. Os resultados mostraram que a técnica de redes neurais convolucionais apresentou uma acurácia média maior do que a técnica de processamento digital de imagens com *MLP*. A *CNN* obteve acurácia de 96.60%, com desvio padrão de 0.0091, e a *MLP* obteve acurácia de 78.13%, com desvio padrão de 0.0199.

O trabalho de Moreira (2023) aborda o uso de Redes Neurais Convolucionais (*CNNs*) para a detecção da região radicular do bulbilho de alho, buscando aprimorar o plantio automatizado da cultura. A pesquisa se destaca pela construção de um invólucro iluminado, permitindo capturar imagens em condições controladas, e pelo desenvolvimento de uma base de dados para o treinamento da *CNN*. Foram empregadas duas abordagens: imagens completas dos bulbilhos de alho e fragmentos de alho contendo ou não a região radicular. O modelo *MobileNet*, utilizado no

estudo, apresentou resultados com alta acurácia nos testes, com médias de 92% para as imagens completas e 93% para os fragmentos de alho. Assim, o trabalho demonstra a capacidade e aplicabilidade da visão computacional na mecanização agrícola, podendo contribuir para maior precisão no plantio e redução dos custos de plantio da cultura.

Silva e Oliveira Junior (2024) apresentaram um estudo que teve como objetivo analisar o impacto de diferentes posições de plantio de bulbilhos de alho na produtividade e qualidade deste. O estudo foi realizado em São Gotardo-MG entre março e julho de 2023. O plantio dos bulbilhos foi realizado em cinco posições diferentes, definidas como: correta, invertida, dorsal, ventral e lateral. O plantio foi realizado de forma manual, para garantir que as posições dos bulbilhos fossem cuidosamente definidas. Entre os parâmetros, foram avaliadas a produtividade e a qualidade dos bulbos do alho. Os resultados mostraram que o plantio na posição correta apresentou a maior produção, chegando a 17,78 toneladas por hectare (ton/ha), enquanto, na posição invertida, foi obtida a menor produtividade, cerca de 8,99 ton/ha. As outras posições apresentaram resultados intermediários. Quanto à qualidade, a posição correta também obteve os melhores resultados, apresentando a maior porcentagem de alho “extra”. A posição invertida teve os piores resultados, obtendo pior qualidade e apresentando a maior quantidade de bulbos de calibre menor.

O estudo demonstra a importância de se realizar o plantio dos bulbilhos de alho na posição correta para se obter maior produtividade e qualidade. A posição invertida deve ser evitada, e a mecanização do plantio deve atender a esse requisito de posicionamento, visando a melhores resultados.

Liu *et al.* (2022) apresentaram um pesquisa que propõe um método baseado em aprendizado profundo, utilizando imagens do contorno do bulbilho de alho para otimizar a identificação da sua orientação. Para garantir a generalização do modelo, foi criado um conjunto de dados que abrange variações como resíduos de casca de alho e desfoque de movimento. Ao todo, foram coletadas 1.470 imagens, sendo que 1.172 foram usadas para treinamento, e 298, para validação.

A etapa de pré-processamento incluiu a ampliação de dados por meio de operações como inversão horizontal, estiramento, cisalhamento e translação, aumentando a diversidade das imagens. Após essas transformações, o conjunto de dados foi ampliado para 29.400 imagens. Para a classificação, cada imagem foi rotacionada aleatoriamente em até 30 graus e dividida em quatro classes (superior, esquerda, inferior e direita), totalizando 117.600 imagens. O treinamento do modelo utilizou diferentes arquiteturas, incluindo como base a *MobileNetV3* e redes totalmente conectadas, com diversos otimizadores e funções de ativação.

Os testes indicaram que o modelo baseado em *MobileNetV3-Large* obteve a maior taxa de reconhecimento, atingindo 98,71% de precisão no conjunto de valida-

ção. No entanto, sua complexidade computacional foi considerada alta para implementação em sistemas embarcados. O modelo *CNN* simplificado apresentou desempenho semelhante com menor custo computacional, tornando-se uma opção mais viável.

O teste embarcado foi realizado utilizando-se uma placa *OrangePi 3 LTS*, demonstrando que a rede totalmente conectada atingiu uma taxa de processamento de 151,40 FPS, garantindo rapidez e eficiência para aplicações em semeadoras de alta velocidade. Os resultados mostram que a abordagem proposta oferece uma solução eficaz e de baixo custo para o reconhecimento da orientação das sementes de alho.

Os principais trabalhos analisados apresentam diferentes abordagens para o uso de redes neurais convolucionais na detecção e classificação de alho, como pode ser visto na Tabela 1. O trabalho de Yang *et al.* (2022) tem como foco a separação automática de raízes de alho, enquanto Anh, Thuyet e Kobayashi (2022) ampliam a análise para classificação pós-corte. Moreira (2023) propõe uma abordagem para detecção da região radicular dos bulbilhos de alho, e Liu *et al.* (2022) se concentra na orientação das sementes de alho.

Tabela 1 – Comparativo entre os trabalhos do estado da arte

Referência	Metodologia	Principais Resultados
Yang *et al.* (2022)	*CNN* para classificar e separar alhos cortados	95% de precisão
Anh, Thuyet e Kobayashi (2022)	*CNN* profunda para classificar imagens de alho	Precisão de 92% (multi classe) e 96% (multi rótulo)
Moreira (2023)	*CNN* para a detecção da região radicular do bulbilho de alho	92% (imagens completas) e 93% (fragmentos)
Liu *et al.* (2022)	*CNN* para identificação da orientação das sementes de alho	98,71% de precisão
Este trabalho	*CNN* para classificação e posicionamento automático de bulbilhos de alho.	90% de precisão

Fonte: Elaborado pelo autor, 2025

O presente trabalho se distingue por desenvolver um sistema completo, que integra a detecção da orientação do bulbilho, classificando sua orientação em ápice ou radicular, utilizando *CNNs*, com um mecanismo físico capaz de posicionar a semente corretamente para o plantio. Já outros estudos focam na classificação ou em etapas específicas; e esse projeto busca uma solução integrada para a automação do plantio de alho.

3 METODOLOGIA

Neste capítulo, apresentam-se a classificação da pesquisa, a solução proposta e os materiais e métodos que serão empregados na realização do trabalho. Este capítulo busca descrever, de forma clara e objetiva, como a pesquisa será conduzida, permitindo futuramente a reprodução deste estudo por outros pesquisadores interessados.

3.1 Classificação do trabalho

De acordo com Gerhardt e Silveira (2009), a pesquisa científica é resultado de um questionamento e estudo minucioso, realizados com o objetivo de se resolver um problema, utilizando procedimentos científicos.

A pesquisa do presente trabalho pode ser classificada, quanto à natureza, como aplicada, pois visou gerar conhecimentos para aplicação prática, com a finalidade de resolver o problema de posicionamento dos bulbilhos de alho presente nos sistemas de plantio mecanizados (GERHARDT; SILVEIRA, 2009).

Quanto aos objetivos, é exploratória, pois tem como objetivo proporcionar maior entendimento do problema, utilizando levantamento bibliográfico para o desenvolvimento de possíveis soluções e condução do projeto (GERHARDT; SILVEIRA, 2009).

Quanto aos procedimentos, é experimental, pois foram coletados dados, como as imagens dos bulbilhos, para o treinamento da *CNN*, e desenvolvido um mecanismo para fazer a rotação do bulbilho caso seja identificado que ele esteja na posição incorreta.

Quanto à abordagem, é quantitativa, já que foi realizada a coleta de dados objetivos e mensuráveis, como imagens capturadas pelo sistema de visão computacional, que permitiram analisar os resultados. Essa abordagem forneceu uma estrutura para o desenvolvimento e aprimoramento do sistema de posicionamento dos bulbilhos de alho com base nos resultados obtidos.

De acordo com a classificação proposta por Wazlawick (2009), a pesquisa deste trabalho pode ser classificada como a “Apresentação de algo diferente”, pois busca apresentar uma solução tecnológica e eficiente para uma tarefa específica, o posicionamento dos bulbilhos de alho.

3.2 Materiais e métodos

Esta seção descreve os materiais utilizados, como o ambiente de desenvolvimento, equipamentos e bibliotecas que foram empregadas para se desenvolver o

trabalho.

3.2.1 Ambiente de desenvolvimento e equipamentos

Para o desenvolvimento deste trabalho, foi utilizado um computador pessoal, além de outros equipamentos. Suas especificações estão detalhadas no Quadro 1.

Quadro 1 – Equipamentos

Componente	Especificação
Processador	AMD Ryzen 5 5600
Placa de vídeo	Geforce RTX 3060 Ti 8GB
Memória Ram	32,0 GB DDR4
Armazenamento	SSD NVMe 1 TB
Sistema operacional	Windows 11 Pro
Arquitetura do Sistema Operacional	64-bits
Câmera digital	HDR 12 Mp
Arduino	UNO R3
Servomotor	Mg995
Motor de passo	Nema 16
Módulo driver de motor de passo	Easy Driver

Fonte: Elaborado pelo Autor, 2025.

Para a configuração do ambiente de desenvolvimento, foram empregadas as ferramentas detalhadas no Quadro 2.

Inicialmente, planejou-se embarcar o modelo na placa *Jetson Nano* para permitir a detecção diretamente no dispositivo e, futuramente, integrar o sistema a um equipamento para plantio automático, buscando garantir maior autonomia ao sistema. No entanto, durante a implementação, foram identificados problemas de compatibilidade com algumas bibliotecas essenciais ao funcionamento do modelo. Diversas tentativas de ajuste foram realizadas, incluindo a instalação de versões específicas das bibliotecas e ajustes no ambiente de desenvolvimento. Porém, devido ao prazo de finalização do trabalho, optou-se por reduzir o escopo do projeto, executando o modelo em um computador desktop, garantindo as funcionalidades necessárias para os testes e validação do sistema.

3.2.2 Projeto elétrico e protótipo do dispositivo

O protótipo foi inicialmente idealizado para que seu desenvolvimento fosse feito de forma iterativa, com o objetivo de validar as principais funcionalidades e a vi-

Quadro 2 – Ferramentas

Tipo	Software/Biblioteca	Versão
Linguagem de programação	Python	3.10.10
Editor de código	Visual Studio Code	1.96.4
Modelagem de projeto mecânico	AutoCAD	V.116.0.0 2025.1
Modelagem de projeto eletrônico	Proteus	8
Biblioteca de Visão Computacional	OpenCV	4.8.1.78
Biblioteca de IA	TensorFlow	2.10.0
Biblioteca de IA	Keras	2.10.0
Biblioteca de Visualização	Seaborn	0.12.2
Biblioteca de Visualização	Matplotlib	3.7.2
Biblioteca Numérica	Numpy	1.26.4
Biblioteca de Dados	Pandas	2.0.3
Biblioteca de Machine Learning	Scikit-learn	1.3.0

Fonte: Elaborado pelo Autor, 2025.

abilidade do projeto. A primeira versão foi feita utilizando-se materiais simples, como papel e papelão, o que permitiu que fossem visualizados alguns requisitos como medidas, que foram definidas para que o protótipo fosse capaz de receber o bulbilho de alho sempre na posição dorsal, facilitando a análise e o treinamento do modelo.

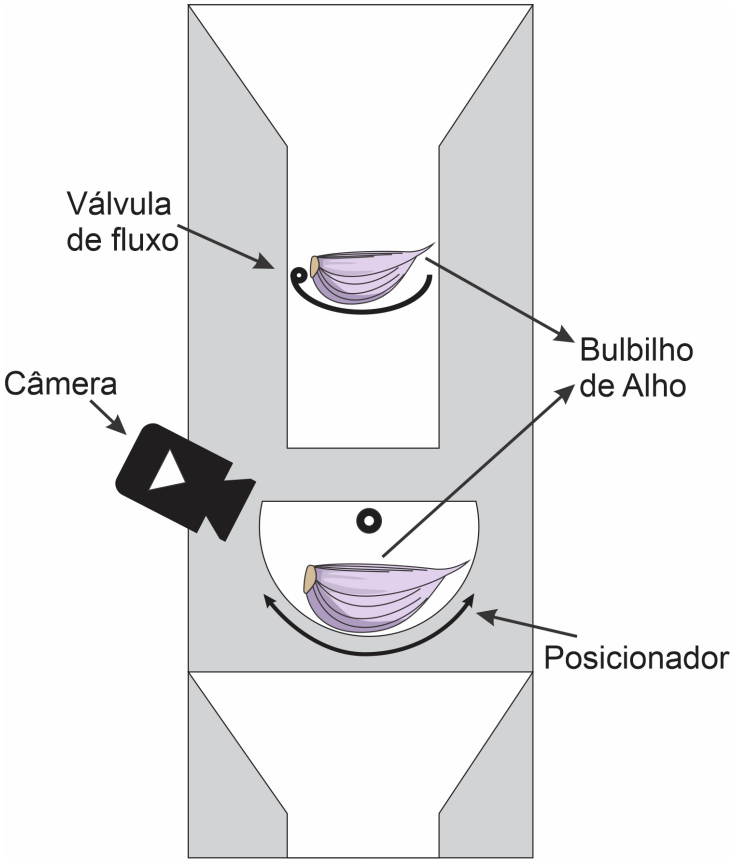
Na Figura 7, é apresentado o design do protótipo, desenvolvido para atender aos requisitos de usabilidade do projeto do sistema de posicionamento de bulbilhos de alho.

No desenho, podemos observar que o protótipo possui uma válvula de fluxo, que, através de um mecanismo rotativo, libera um bulbilho por vez para a câmara de classificação. Essa válvula também garante que não haja aglomeração dos bulbilhos e evita que ocorra congestionamento, o que pode causar travamento do dispositivo ou quebra dos bulbilhos, permitindo que cada bulbilho seja analisado individualmente.

O bulbilho avança, então, para a câmara de classificação, onde é posicionado de maneira dorsal, que é a posição inicial para análise da semente. A câmera captura uma imagem do bulbilho, a qual é processada pelo algoritmo para a identificação da orientação da semente. Essa classificação tem como objetivo determinar qual lado do bulbilho está voltado para a câmera, para, então, corrigir a posição. O berço posicionador, através de um movimento rotativo, alinha o bulbilho, liberando-o com a parte radicular orientada para baixo.

O projeto do protótipo foi desenvolvido utilizando-se o software *AutoCAD*, Figura 8. O projeto foi, então, exportado no formato *DXF* para ser utilizado na máquina de corte a laser. As peças para a montagem do protótipo foram, inicialmente, cortadas

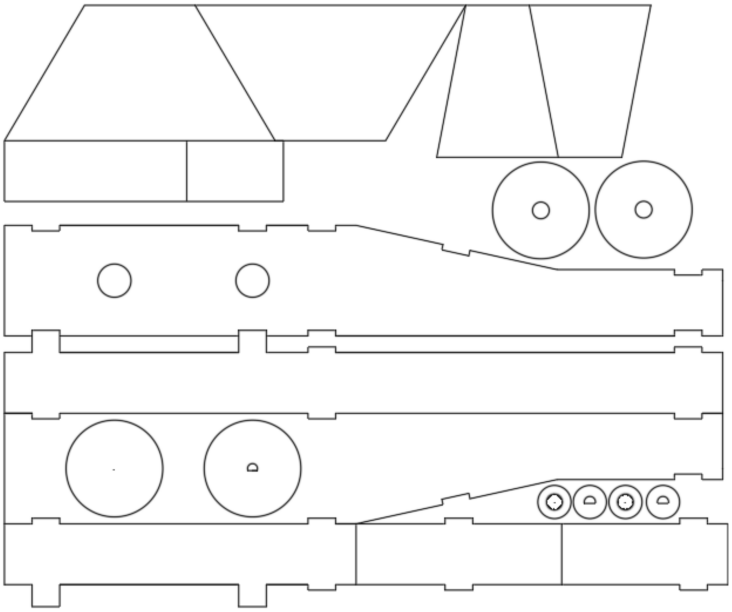
Figura 7 – Desenho do protótipo



Fonte: Elaborado pelo autor, 2025.

em acrílico transparente de 2 mm. Após o corte, o protótipo foi montado utilizando-se uma cola de secagem rápida. Durante a montagem, foi verificado que o material de 2 mm era frágil demais para os requisitos do projeto. Por isso, foram implementadas melhorias, e o projeto foi adaptado para as novas medidas e cortado novamente em acrílico mais resistente, de 4 mm. A espessura de 4 mm foi escolhida por garantir maior resistência, necessária para suportar os componentes do protótipo, garantindo, assim, sua estabilidade e segurança.

Figura 8 – Projeto do protótipo no *AutoCad*.

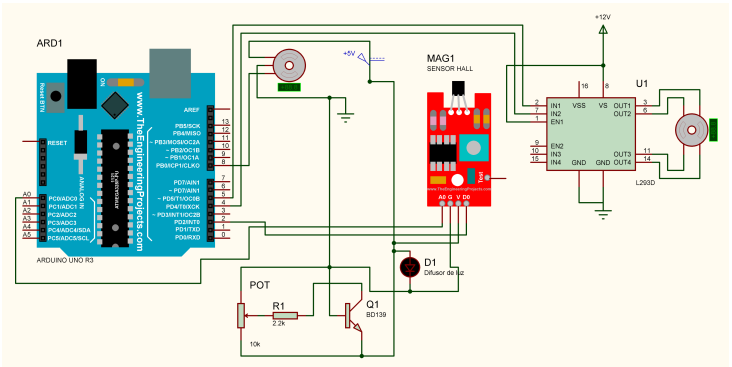


Fonte: Elaborado pelo autor, 2025.

Na construção do protótipo, priorizou-se a utilização de componentes de baixo custo e fácil aquisição, com o objetivo de otimizar o tempo e o custo da implementação. A plataforma de prototipagem escolhida foi o Arduino UNO R3, devido à sua facilidade de aquisição, disponibilidade no mercado, programação e flexibilidade para testar diferentes configurações.

O projeto elétrico/eletrônico foi desenvolvido utilizando-se o software Proteus 8, um ambiente de simulação e modelagem eletrônica, conforme ilustrado na Figura 9. Esse projeto teve como objetivo garantir a correta integração entre todos os componentes necessários ao protótipo, verificando sua compatibilidade e configuração ideal para que funcionassem de forma integrada e como esperado.

Figura 9 – Esquema elétrico do protótipo



Fonte: Elaborado pelo autor, 2025.

Foi desenvolvido um programa para Arduino que tem como objetivo controlar um motor de passo usando um driver (*Easydriver*) e um servomotor, além de implementar uma lógica de detecção de travamento do servomotor. O código utiliza

interrupções, controle de tempo e funções para fazer a detecção de travamentos, Figura 10.

Figura 10 – Lógica de detecção de travamento

```
102     // Lógica de travamento
103     if ((millis() - t3) > 6000) {
104       if ((vel == 0) && (travou == 0)) {
105         travou = 1;
106         vel = 180;
107         Serial.println("Travou!");
108         t2 = millis();
109         t3 = millis();
110       }
111     }
112
113     if (travou == 1 && (millis() - t2) > 3000) {
114       travou = 0;
115       t2 = millis();
116       vel = 0;
117     }
118   }
```

Fonte: Elaborado pelo autor, 2025.

3.3 Construção do *dataset*

A câmera foi posicionada no dispositivo de forma que possa capturar as imagens de forma mais eficaz, visando obter imagens que priorizam destacar duas posições de interesse nos bulbilhos de alho, destacando a visualização destas.

Para a coleta dos dados, foi desenvolvido um *script* em *Python*, Figura 13, que utiliza a biblioteca *OpenCV* (*cv2*) para capturar imagens de uma câmera conectada ao sistema. O programa organiza as imagens capturadas em um diretório de saída definido no código, podendo ser “Apice” ou “Radicular”. Caso o diretório ainda não exista, ele é criado automaticamente.

A captura das imagens ocorre por meio de um loop que monitora continuamente o feed da câmera. Durante a execução, o usuário pode pressionar a tecla ’s’ para salvar o quadro atual, que será armazenado no diretório com um nome sequencial, garantindo a organização. Além disso, o *script* permite a saída segura do programa ao pressionar a tecla ’q’. Ao final da execução, o *script* realiza a liberação dos recursos utilizados, incluindo o encerramento da câmera e o fechamento de janelas de exibição, assegurando a estabilidade e o correto funcionamento do sistema.

Para garantir que todas as imagens seguissem um padrão de nomes, foi criado um *script* em *Python*, como mostrado na Figura 12. Este *script* utiliza o módulo os para renomear arquivos de imagem dentro de uma pasta específica. A função renomear_imagens percorre cada arquivo na pasta, verifica se ele possui uma das

Figura 11 – *Script* para captura das imagens

```python
import cv2
import os
import keyboard

def tirar_foto(nome_arquivo, moldura):
    caminho_completo = os.path.join(pasta_saida, nome_arquivo)
    cv2.imwrite(caminho_completo, moldura)
    print(f"Foto salva em {caminho_completo}")

# Cria a pasta de saída, se ela não existir
pasta_saida = "Apice" # "Radicular"
if not os.path.exists(pasta_saida):
    os.makedirs(pasta_saida)

# Inicializa a câmera
camera = cv2.VideoCapture(0)  # 0 indica a câmera padrão

if not camera.isOpened():
    print("Erro: Não foi possível abrir a câmera.")
    exit()
```

Fonte: Elaborado pelo autor, 2025

extensões de imagem comuns (.png. jpg, etc.) e, então, extrai o nome e a extensão do arquivo. Em seguida, gera um novo nome sequencial para cada imagem, no formato “apice_XX.extensao”, onde XX é um número com dois dígitos e a extensão original do arquivo. Este código tem como objetivo facilitar a organização de um banco de dados de imagens, padronizando os nomes dos arquivos.

Figura 12 – *Script* para renomear as imagens do banco de dados

```python
import os

def renomear_imagens(caminho_da_pasta):
  """
  Renomeia todas as imagens em uma pasta sequencialmente, começando com "radicular01".
  Args:
    caminho_da_pasta: O caminho para a pasta que contém as imagens.
  """
  i = 1
  for nome_do_arquivo in os.listdir(caminho_da_pasta):
    # Verifica se o arquivo é uma imagem
    if nome_do_arquivo.endswith(('.png', '.jpg', '.jpeg', '.gif', '.bmp')):
      # Separa o nome do arquivo e a extensão
      nome_base, extensao = os.path.splitext(nome_do_arquivo)

      # Cria o novo nome do arquivo
      novo_nome = f"apice_{i:02d}{extensao}"

      # Cria os caminhos completos para o arquivo antigo e o novo
      caminho_antigo = os.path.join(caminho_da_pasta, nome_do_arquivo)
      caminho_novo = os.path.join(caminho_da_pasta, novo_nome)

      # Renomeia o arquivo
      os.rename(caminho_antigo, caminho_novo)
      print(f"Arquivo '{nome_do_arquivo}' renomeado para '{novo_nome}'")

      i += 1
```

Fonte: Elaborado pelo autor, 2025

No conjunto de imagens obtidas, existem duas situações possíveis: a região virada para a câmera será a região radicular ou a região do ápice, conforme a Figura 13.

Todas as imagens utilizadas para a formação da base de dados foram imagens reais coletadas diretamente no dispositivo, sendo a resolução utilizada de 640x480 pixels. Para a formação da base de dados, coletaram-se 1084 imagens de

Figura 13 – Exemplo de captura no dispositivo.

Fonte: Elaborado pelo autor, 2025

bulbilhos de alho, sendo 542 da região radicular e 542 da região do ápice. Com o objetivo de garantir maior diversidade das amostras, foram utilizados bulbilhos de alho com variações de tamanho, cor e formato. A coleta das imagens no dispositivo foi realizada em um ambiente com iluminação controlada, com a finalidade de assegurar imagens de maior qualidade, seguindo-se um padrão de iluminação. A posição da câmera é fixa no dispositivo; porém, foram consideradas pequenas variações de ângulo e inclinação na captura das imagens dos bulbilhos de alho.

Para a preparação dos dados para o treinamento do modelo, na Figura 14, podemos ver um *script* em *Python* que foi desenvolvido para garantir que as imagens fossem divididas de forma aleatória para os conjuntos de treino e validação. Este *script* divide um conjunto de imagens recebidas de uma pasta de entrada em duas pastas distintas: uma para treino e outra para validação. A divisão é feita de forma aleatória, garantindo que cada conjunto represente bem a diversidade de dados originais. Os conjuntos de treino e validação são criados na proporção de 80% e 20%, respectivamente, permitindo que o modelo seja treinado com uma parte dos dados e sua performance seja validada com os dados que não foram utilizados no treinamento. As pastas de treino e teste são criadas automaticamente se não existirem, e as quantidades de imagens em cada conjunto são exibidas ao final da execução do programa.

3.4 Treinamento do modelo

Inicialmente, o *script* importa as bibliotecas necessárias para o desenvolvimento do projeto, incluindo o *tensorflow* e *keras*, para a construção e treinamento do modelo de rede neural; *numpy* e *pandas*, para manipulação de dados; *seaborn* e *matplotlib*, para visualização; e *sklearn*, para métricas de avaliação. Em seguida, são

Figura 14 – *Script* para realizar a divisão dos dados

```python
import os
import random
import shutil
from PIL import Image

def split_images(input_folder, train_folder, valid_folder, train_ratio=0.8):
    # Cria as pastas de treino e validação se elas não existirem
    if not os.path.exists(train_folder):
        os.makedirs(train_folder)
    if not os.path.exists(valid_folder):
        os.makedirs(valid_folder)

    # Lista todos os arquivos na pasta de entrada
    filenames = [f for f in os.listdir(input_folder) if os.path.isfile(os.path.join(input_folder, f))]

    # Filtra para manter apenas arquivos de imagem válidos
    image_filenames = []
    for filename in filenames:
        try:
            # Tenta abrir a imagem para verificar se é válida
            with Image.open(os.path.join(input_folder, filename)) as img:
                img.verify()  # Verifica se a imagem não está corrompida
            image_filenames.append(filename)
        except Exception as e:
            print(f"Arquivo ignorado (não é uma imagem válida ou está corrompido): {filename} - {e}")

    # Embaralha a lista de imagens
    random.shuffle(image_filenames)

    # Calcula o número de imagens para treino e validação
    num_train = int(len(image_filenames) * train_ratio)
    num_valid = len(image_filenames) - num_train

    # Divide as imagens em conjuntos de treino e validação
    train_images = image_filenames[:num_train]
    valid_images = image_filenames[num_train:]
```

Fonte: Elaborado pelo autor, 2025

definidas as variáveis *PATH, train_dir e validation_dir* para especificar os caminhos dos diretórios contendo as imagens de treinamento e validação, sendo definidos, também, o *batch size* e as dimensões das imagens de entrada para o modelo, como pode ser observado na Figura 15.

Figura 15 – Criação das classes a definições iniciais

```python
PATH = "C:/data"
train_dir = os.path.join(PATH, 'train')
validation_dir = os.path.join(PATH, 'validation')

BATCH_SIZE = 10
IMG_SIZE = (100, 100)
```

Fonte: Adaptado de Chollet (2023).

Em seguida, as imagens do banco de dados são carregadas na memória e passam por um processo de *data augmentation* (*DA*), onde são espelhadas horizontal e verticalmente, além de rotacionadas em ângulos aleatórios, ampliando o número total de imagens, como demonstrado na Figura 16.

Figura 16 – Carregamento das imagens e *data augmentation*

```python
train_dataset = train_dataset.prefetch(buffer_size=AUTOTUNE)
validation_dataset = validation_dataset.prefetch(buffer_size=AUTOTUNE)
test_dataset = test_dataset.prefetch(buffer_size=AUTOTUNE)

data_augmentation = tf.keras.Sequential([
    tf.keras.layers.experimental.preprocessing.RandomFlip('horizontal'),
    tf.keras.layers.experimental.preprocessing.RandomFlip('vertical'),
    tf.keras.layers.experimental.preprocessing.RandomRotation(0.2),
])
```

Fonte: Adaptado de Chollet (2023).

A arquitetura do modelo é construída utilizando-se o modelo pré-treinado *MobileNetV2* como base, e as imagens são normalizadas com a função *prepro-*

cess_input. O modelo-base é pré-treinado no banco de imagens *ImageNet*, que é composto por 1,4 milhão de imagens e 1000 classes. O atributo *trainable*, do modelo-base, é definido como *False*, para que seus pesos não sejam alterados durante o treinamento inicial. Camadas adicionais são adicionadas para adaptar o modelo à tarefa específica de classificação binária, incluindo uma nova camada chamada *GlobalAveragePooling2D*, para reduzir a dimensionalidade, e também uma camada *Dense* para realizar a classificação. O modelo final é compilado utilizando-se o *Keras* e o otimizador *Adam*. Aplicam-se a função de perda *BinaryCrossentropy* e a métrica de avaliação *accuracy*. O modelo é, então, treinado com 100 épocas e taxa de aprendizado de 0,0001, como apresentado na Figura 17.

Figura 17 – Construção e Configuração do Modelo

```
preprocess_input = tf.keras.applications.mobilenet_v2.preprocess_input

IMG_SHAPE = IMG_SIZE + (3,)
base_model = tf.keras.applications.MobileNetV2(input_shape=IMG_SHAPE, include_top=False, weights='imagenet')

image_batch, label_batch = next(iter(train_dataset))
feature_batch = base_model(image_batch)
print(feature_batch.shape)

base_model.trainable = False

base_model.summary()

global_average_layer = tf.keras.layers.GlobalAveragePooling2D()
feature_batch_average = global_average_layer(feature_batch)
print(feature_batch_average.shape)

prediction_layer = tf.keras.layers.Dense(1)
prediction_batch = prediction_layer(feature_batch_average)
print(prediction_batch.shape)

inputs = tf.keras.Input(shape=(100, 100, 3))
x = data_augmentation(inputs)
x = preprocess_input(x)
x = base_model(x, training=False)
x = global_average_layer(x)
x = tf.keras.layers.Dropout(0.2)(x)
outputs = prediction_layer(x)
model = tf.keras.Model(inputs, outputs)

base_learning_rate = 0.0001
model.compile(optimizer=tf.keras.optimizers.Adam(lr=base_learning_rate),
              loss=tf.keras.losses.BinaryCrossentropy(from_logits=True), metrics=['accuracy'])

model.summary()

initial_epochs = 100
loss0, accuracy0 = model.evaluate(validation_dataset)
```

Fonte: Adaptado de Chollet (2023).

O modelo treinado é avaliado no conjunto de dados de validação, que é composto por 20% das imagens do *dataset*. Isso é necessário para determinar a performance final do modelo, conforme evidenciado na Figura 18.

Figura 18 – Avaliação do modelo

```
loss, accuracy = model.evaluate(test_dataset)
print('Test Accuracy :', accuracy)

image_batch, label_batch = test_dataset.as_numpy_iterator().next()
predictions = model.predict_on_batch(image_batch).flatten()

predictions = tf.nn.sigmoid(predictions)
predictions = tf.where(predictions < 0.5, 0, 1)

print('Predictions:\n', predictions.numpy())
print('Labels:\n', label_batch)
```

Fonte: Adaptado de Chollet (2023).

Para avaliar o desempenho do modelo, é utilizada uma matriz de confusão, que compara as previsões feitas pelo modelo com os rótulos do conjunto de valida-

ção. Essa matriz organiza os acertos e erros do modelo em categorias específicas, permitindo realizar uma análise detalhada do desempenho do modelo. Além disso, são calculadas três principais métricas de classificação:

Precisão (*Precision*): essa métrica indica a proporção de previsões reais corretas em relação ao total de previsões feitas pelo modelo, exibindo quantas classificações realmente estavam corretas.

Recall: essa métrica mede a capacidade do modelo de identificar acertadamente todas as ocorrências positivas e calcula a proporção de casos positivos previstos corretamente em relação ao total de casos positivos reais.

F1-score: é a métrica que a média entre precisão e *Recall*, proporcionando um equilíbrio entre essas duas métricas; é útil para avaliar quando há um desequilíbrio entre as classes.

Essas métricas são obtidas por meio da função *classification_report*, da biblioteca *Scikit-learn*, conforme pode ser visto na Figura 19.

Figura 19 – Matriz de Confusão e Métricas de Classificação

```
list1 = list()
list2 = list()
for i in range(5):
    loss, accuracy = model.evaluate(test_dataset)
    print('Test accuracy :', accuracy)
    image_batch, label_batch = test_dataset.as_numpy_iterator().next()
    predictions = model.predict_on_batch(image_batch).flatten()
    predictions = tf.nn.sigmoid(predictions)
    predictions = tf.where(predictions < 0.5, 0, 1)
    print ('Predictions: \n', predictions.numpy())
    print ('Labels: \n', label_batch)
    list1 = np.append(list1, predictions.numpy())
    list2 = np.append(list2, label_batch)

y_actu = pd.Series(list2, name='Actual')
y_pred = pd.Series(list1, name='Predicted')

df_confusion = pd.crosstab(y_actu, y_pred)

df_confusion_pct = df_confusion.applymap(lambda x: x/df_confusion.values.sum()*100)

sns.heatmap(df_confusion_pct, cmap='Blues', annot=True, fmt='.2f', xticklabels=class_names, yticklabels=class_names)
plt.title('Matriz de Confusão')
plt.show()

print('Classification Report:')
print(classification report(y actu, y pred, target names=class names))
```

Fonte: Adaptado de Chollet (2023).

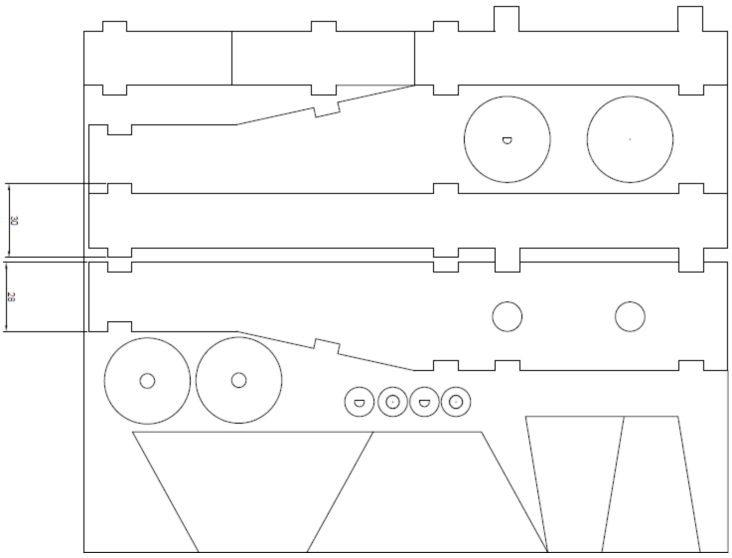
4 RESULTADOS E DISCUSSÃO

Neste capítulo, serão apresentados os resultados obtidos a partir da aplicação da metodologia, materiais e métodos descritos no capítulo 3. Os resultados obtidos serão analisados e discutidos, relacionando-os aos objetivos propostos anteriormente.

4.1 Mecanismo de posicionamento do bulbilho de alho

Todas as peças do mecanismo de posicionamento dos bulbilhos de alho foram projetadas e fabricadas no laboratório *IFMaker*, utilizando-se a máquina de corte a laser deste. O material empregado foram placas acrílicas de 4 mm. O plano de corte é apresentado na Figura 20.

Figura 20 – Plano de corte

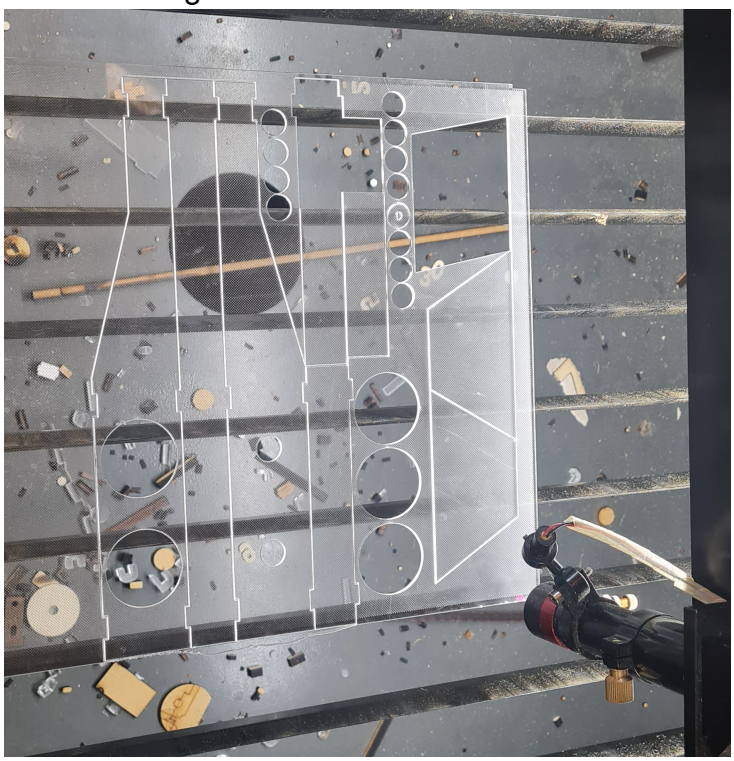


Fonte: Elaborado pelo autor, 2025

O processo de corte com a máquina de corte a laser é ilustrado na Figura 21. Todas as partes foram dispostas da melhor forma para otimizá-lo, reduzindo o tempo de trabalho e gerando economia de material.

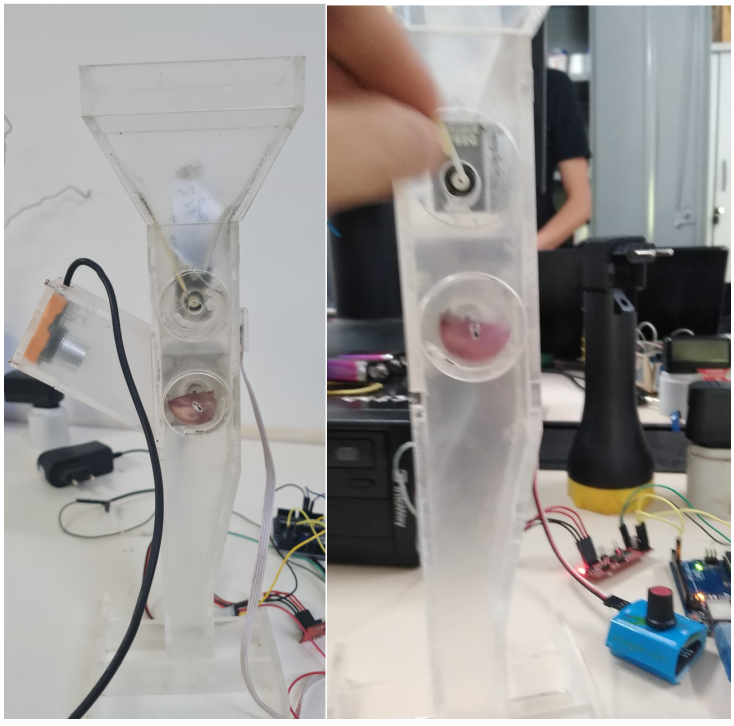
Em seguida, efetuaram-se a montagem e o teste integrado do projeto mecânico, eletrônico e programa do microcontrolador. Após os ajustes necessários, o sistema permitiu a separação de bulbilhos, a deposição na câmara de análise e a rotação do bulbilho de alho, como esperado. Após os primeiros testes, verificou-se que algumas dimensões precisaram ser alteradas, e, depois da montagem com os ajustes, o sistema funcionou como esperado, acionando o mecanismo de giro. Também foi testado o sistema de detecção de travamentos, que funcionou como previsto. A Figura 22 ilustra a montagem final e o teste integrado do mecanismo.

Figura 21 – Processo de corte



Fonte: Elaborado pelo autor, 2025

Figura 22 – Teste integrado (mecânico, eletrônico e programa de controle)



Fonte: Elaborado pelo autor, 2025

Foi instalado um sistema de iluminação para a câmera de análise das imagens, e o mecanismo foi pintado com tinta preta fosca para facilitar a segmentação das imagens. O mecanismo foi desenvolvido utilizando-se o microcontrolador Arduíno

UNO R3, para fazer o acionamento dos motores de passo, responsáveis pelo ajuste de posicionamento do bulbilho de alho.

Foi utilizado um sensor *hall* para detectar a posição inicial e final do tambor rotativo, o qual, por sua vez, realiza a função de válvula de bloqueio, impedindo que novos bulbilhos sejam depositados na câmara de análise de imagem, se houver um bulbilho sendo analisado.

Caso haja travamento, o programa detecta, por meio do sinal do sensor de efeito *hall*, em estado lógico 0 ou 1, por um determinado tempo definido, e aciona a reversão, destravando o canal de passagem do bulbilho. O sensor de efeito *hall* é ligado a uma entrada de interrupção do microcontrolador, assumindo prioridade em caso de acionamento. Na Figura 23, ilustram-se o tambor rotativo (2), a câmara de análise de imagens (3), o sensor de efeito *hall* (4) e a câmera instalada (1).

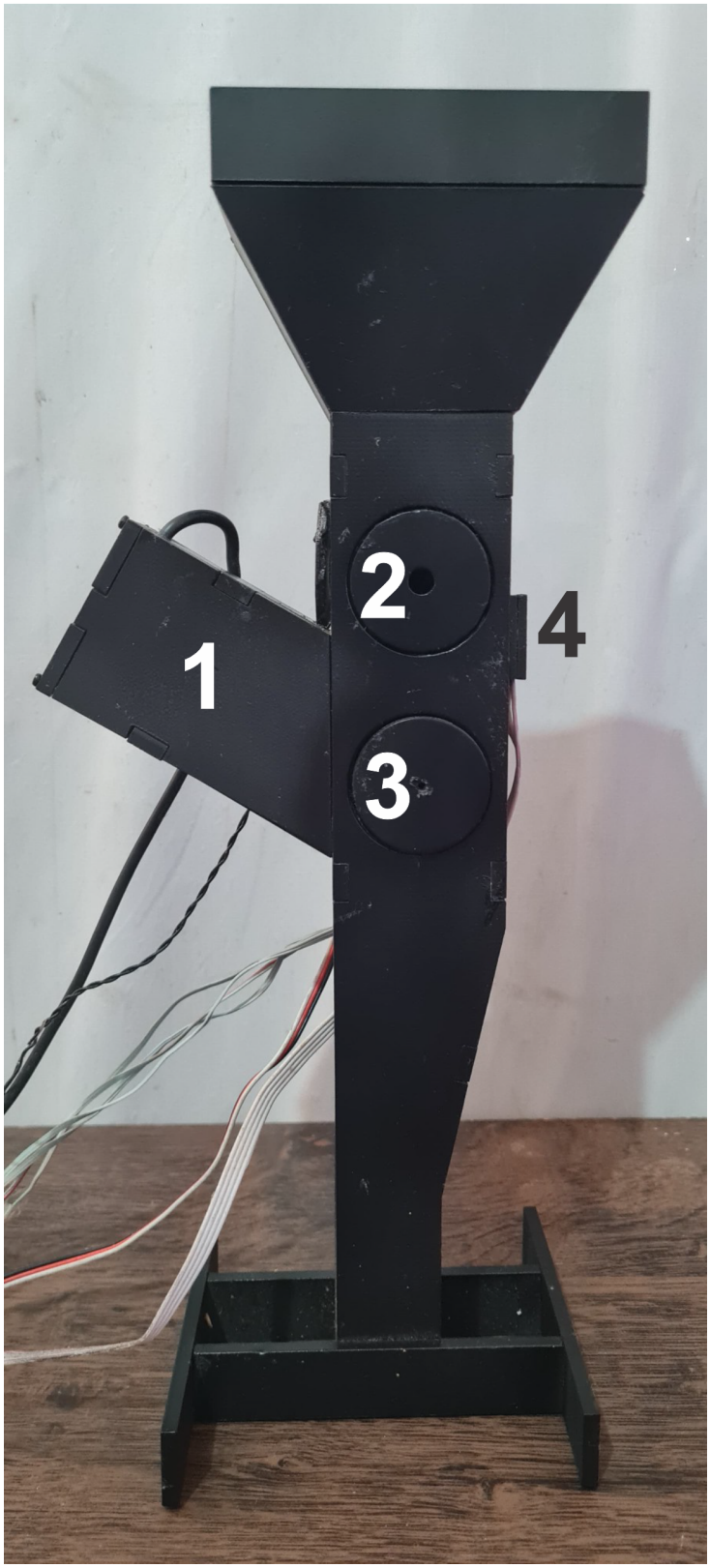
Com o dispositivo finalizado e o modelo treinado e validado, foi feita a integração do sistema utilizando-se um computador para rodar o modelo treinado, e, com a câmera conectada, efeuou-se a detecção, em tempo real, no dispositivo. Com o sistema sendo executado, a classificação foi feita diretamente no dispositivo, o que garante alta velocidade de resposta, permitindo detectar, em tempo real, qual classe está sendo identificada, o que é essencial para o processo de posicionamento automático do bulbilho de alho. O resultado da classificação é disponibilizado, em tempo real, para acionar o mecanismo de posicionamento, que é responsável por corrigir a orientação dos bulbilhos, garantindo que estejam sempre na posição correta para o plantio. Na Figura 24, é possível visualizar exemplos da detecção, em tempo real, das classes Radicular e Ápice.

Os testes foram realizados de forma individual. Primeiramente, foi testada a capacidade de processamento de frames por segundo (*FPS*) do modelo e, depois, foram realizados os testes para verificar com qual velocidade o mecanismo conseguiria realizar um giro de 90 do posicionador, necessário para corrigir a posição dos bulbilhos de alho.

O modelo demonstrou uma capacidade de processamento de 6.18 FPS, o que indica uma velocidade considerável para a classificação dos bulbilhos. Porém, a taxa de plantio é limitada pela velocidade do mecanismo físico, que realiza um giro de 90 graus, necessário para a liberação do bulbilho, a cada 0.67 segundos. Com base nesse tempo de giro, o mecanismo consegue liberar, aproximadamente, 1.49 sementes por segundo, atingindo uma taxa de plantio de cerca de 89.4 sementes por minuto. Portanto, embora o modelo possua alta capacidade de processamento, o sistema fica restrito à parte física do mecanismo, limitando a sua taxa de plantio.

A abordagem de detecção, em tempo real, no dispositivo é essencial para a melhora no desempenho e eficiência do sistema. A capacidade de realizar a detecção, em tempo real, das classes do modelo diretamente no dispositivo garante uma

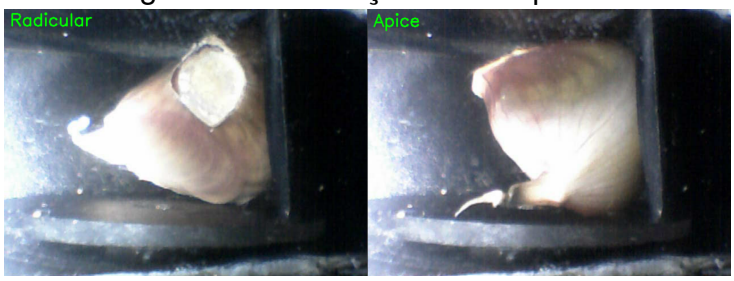
Figura 23 – Dispositivo de posicionamento montado



Fonte: Elaborado pelo autor, 2025

vantagem significativa para o projeto, proporcionando uma solução mais autônoma, confiável e eficiente, podendo ser aplicada em cenários reais de plantio.

Figura 24 – Detecção em tempo real



Fonte: Elaborado pelo autor, 2025.

4.2 Rede neural

Foi utilizado um modelo de *CNN MobileNetV2* disponibilizado por Moreira (2023), em que foram utilizados 20% das imagens do banco de dados para validação, e os outros 80%, para o treinamento da rede. Com os resultados obtidos do treinamento, o autor obteve acurácia superior a 92% na identificação da região radicular.

Com a finalidade de melhorar a acurácia do modelo, foi feito um novo treinamento do modelo, utilizando-se um novo banco de dados com imagens coletadas diretamente no dispositivo, e, das imagens coletadas, foram separadas 20% para validação, e os outros 80%, para o treinamento da rede. O conjunto de testes foi definido como sendo 1/5 do tamanho do conjunto de validação.

Com as alterações feitas, foram obtidos novos resultados, que se mostraram promissores, obtendo-se acurácia de cerca de 96% utilizando-se o método de validação *Hold-out*, como pode ser visto na Figura 2. Nesse método, são definidos dois conjuntos separados de treinamento e validação, e uma divisão adicional do conjunto de validação para criar um conjunto de testes separado para avaliação final.

Tabela 2 – Resultado do treinamento utilizando a validação *Hold-out*

Test accuracy:	0.949999988079071			
Predictions:	[0 1 0 0 0 0 1 0 0 1]			
Labels:	[0 1 0 0 1 0 1 0 0 1]			
Classification Report:				
Classe/Métrica	Precisão	Recall	F1-Score	Suporte
Ápice	0.92	1.00	0.96	24
Radicular	1.00	0.92	0.96	26
Acurácia	-	-	0.96	50
Média Macro	0.96	0.96	0.96	50
Média Ponderada	0.96	0.96	0.96	50

Fonte: Elaborado pelo autor, 2025

Buscando obter resultados melhores e mais confiáveis, foi realizada a vali-

dação cruzada *K-Fold*, utilizando-se 5 treinamentos e separando os dados em 5 pastas, para alimentar o modelo. Cada treinamento empregou 80% das imagens, para o treinamento, e 20%, para validação. Ao final de todos os 5 treinamentos, foi realizada uma média da acurácia obtida dos modelos, resultando em 90% de acurácia, o que representa um resultado satisfatório em relação ao método *Hold-out*. Com os resultados obtidos de todos os treinamentos, elaborou-se uma tabela e foi determinada a média dos resultados, como pode ser visto na Tabela 3. A precisão média para a classe Ápice foi de 91%, e, para radicular, de 90%, indicando que o modelo é capaz de classificar, de forma correta, as amostras de cada classe com uma boa acurácia.

Tabela 3 – Média das métricas de classificação

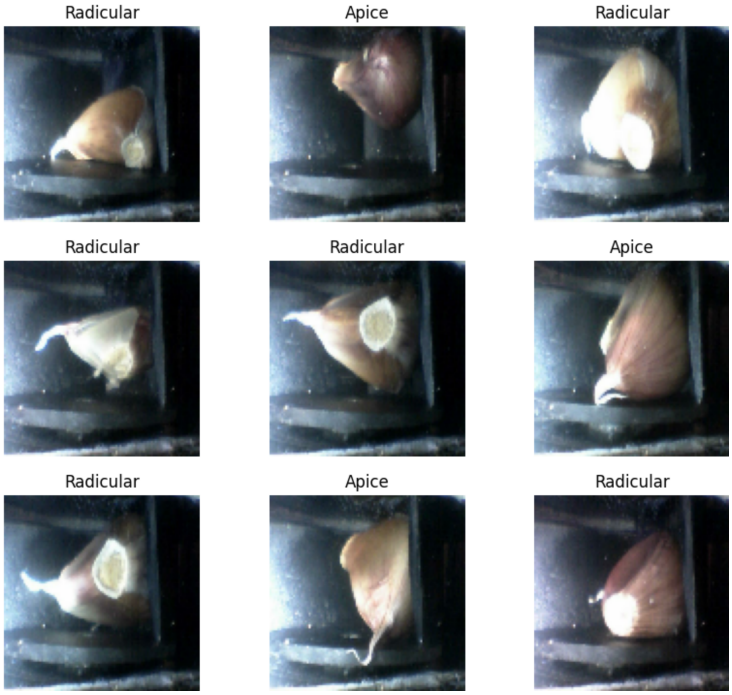
Classe/Métrica	Precisão	Recall	F1-Score	Suporte
Ápice	0.91	0.90	0.90	26.6
Radicular	0.90	0.90	0.90	23.4
Acurácia	-	-	0.90	50
Média Macro	0.90	0.90	0.90	50
Média Ponderada	0.90	0.90	0.90	50

Fonte: Elaborado pelo autor, 2025

Na Figura 25, pode-se observar um exemplo de classificação do modelo em que a rede foi capaz de classificar corretamente todas as imagens dos bulbilhos de alho em Ápice e Radicular. Porém, na Figura 26, podemos observar que a rede classificou, de forma incorreta, uma imagem de ápice como radicular - o que pode ter ocorrido devido ao brilho incidente na imagem.

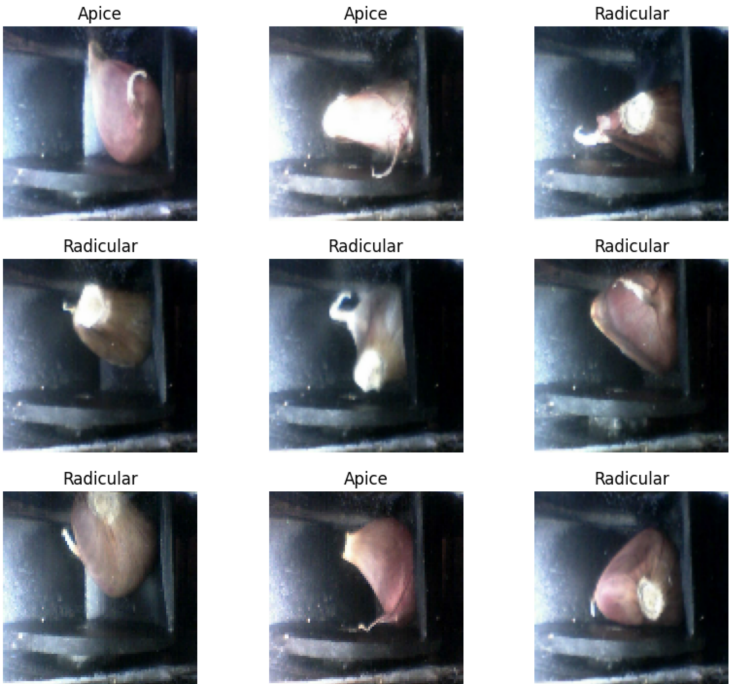
No geral, os resultados indicam que o modelo tem uma performance muito boa na classificação entre as duas classes de Ápice e Radicular, apresentando altos valores para todas as métricas analisadas. Para garantir a robustez do modelo, o treinamento e os testes foram realizados com bulbilhos de alho de diferentes formatos, cores e tamanhos. Essa diversidade no conjunto de dados permitiu avaliar a capacidade do modelo de generalizar e classificar corretamente as classes de forma eficiente, mesmo com essas variações.

Figura 25 – Exemplo de classificação correta



Fonte: Elaborado pelo autor, 2025.

Figura 26 – Exemplo de classificação incorreta



Fonte: Elaborado pelo autor, 2025.

5 CONSIDERAÇÕES FINAIS

O trabalho demonstrou o desenvolvimento de um protótipo para o posicionamento automático de bulbilhos de alho, utilizando visão computacional. A combinação de um mecanismo físico e eletrônico e a utilização de redes neurais convolucionais (*CNNs*) resultou em um sistema capaz de identificar, em tempo real, com boa precisão, a orientação (radicular ou ápice) dos bulbilhos de alho, possibilitando corrigir a posição para o plantio.

Com os resultados obtidos, verificou-se que o modelo treinado alcançou uma acurácia média de 90% na classificação dos bulbilhos, indicando a eficácia da abordagem proposta. A capacidade do sistema de realizar a detecção e a classificação dos bulbilhos, em tempo real, diretamente no dispositivo, representa um avanço em relação às soluções tradicionais de plantio. Além disso, o projeto demonstra um grande potencial de ser implementado em um sistema de plantio automatizado, tendo potencial de reduzir custos de produção e aumentar a produtividade da cultura do alho. O uso da mecanização no plantio e o posicionamento correto dos bulbilhos de alho podem contribuir para uma melhor germinação e desenvolvimento das plantas, reduzindo as perdas causadas pelo plantio incorreto das sementes.

O objetivo geral do trabalho foi alcançado, e os objetivos específicos, parcialmente alcançados, sendo desenvolvido um sistema para o posicionamento automático de bulbilhos de alho utilizando-se visão computacional. O protótipo desenvolvido é capaz de rotacionar os bulbilhos para a posição correta de plantio e, com um algoritmo de processamento digital de imagens e inteligência artificial, classificar a posição dos bulbilhos em radicular e ápice. Entretanto, o algoritmo para corrigir a posição do bulbilho de alho, que deveria receber o resultado da classificação e acionar os atuadores mecânicos para liberar a semente na posição correta, não foi implementado.

Para trabalhos futuros, recomendam-se a utilização de atuadores mais rápidos e o desenvolvimento de um algoritmo capaz de fazer a integração entre o sistema de classificação e o mecanismo de posicionamento, permitindo uma integração completa do sistema e sua implementação em um sistema embarcado, possibilitando a utilização em um sistema de plantio automatizado.

O estudo buscou contribuir com projetos futuros, como o desenvolvimento de sistemas de plantio totalmente automatizados, com a integração de outras tecnologias e sensores, aumentando a eficiência e a qualidade da produção.

REFERÊNCIAS

ANH, P. T. Q.; THUYET, D. Q.; KOBAYASHI, Y. Image classification of root-trimmed garlic using multi-label and multi-class classification with deep convolutional neural network. **Postharvest Biology and Technology**, v. 190, p. 111956, 2022. Disponível em: https://www.sciencedirect.com/science/article/pii/S0925521422001247. Acesso em: 23/05/2023.

BARRETO, J. M. Introdução às Redes Neurais Artificiais. **Florianópolis: UFSC**, 2002.

CHOLLET, F. **Transfer learning and fine-tuning**. 2023. Disponível em: https://www.tensorflow.org/tutorials/images/transfer_learning. Acesso em: 15/06/2024.

DATA SCIENCE ACADEMY. **Deep Learning Book**. 2022. Disponível em: https://www.deeplearningbook.com.br/. Acesso em: 28/05/2023.

GERHARDT, T. E.; SILVEIRA, D. T. **Métodos de pesquisa**. Porto Alegre: Editora da UFRGS, 2009.

GONZALEZ, R. C.; WOODS, R. E. **Processamento digital de imagens**. 3. ed.: Pearson, 2010.

GRÜNDLING, R. D. P.; GAZZOLA, R.; ARAGÃO, A. A. Mercado mundial do alho: tendências gerais e as implicações para o Brasil. In: 59. CONGRESSO da Sociedade Brasileira de Economia, Administração e Sociologia Rural – SOBER. 2021.

IBM. **Redes Neurais Convolucionais (CNNs)**. 2024. Disponível em: https://www.ibm.com/br-pt/topics/convolutional-neural-networks. Acesso em: 26/01/2025.

LIU, J. *et al.* Contour resampling-based garlic clove bud orientation recognition for high-speed precision seeding. **Agriculture**, MDPI, v. 12, n. 9, p. 1334, 2022.

LUDERMIR, T. B. Inteligência Artificial e Aprendizado de Máquina: estado atual e tendências. **Estudos Avançados**, SciELO Brasil, Pernambuco, v. 35, p. 85–94, 2021.

MARQUES FILHO, O.; VIEIRA NETO, H. **Processamento digital de imagens**. Brasport, 1999.

MENEGUZZO, R. **Efeito da irrigação, posição de plantio e aplicação de silício sobre a cultura do alho na Serra Gaúcha.** 2020. Trabalho de Conclusão de Curso (Graduação em Engenheira Agrônoma) – Universidade de Caxias do Sul, Caxias do Sul, 2020.

MENEZES SOBRINHO, J. A. de *et al.* A cultura do alho. **Brasília, DF: EMBRAPA-SPI**, 1993.

MOREIRA, F. A. C. **Detecção da região radicular do bulbo de alho utilizando redes neurais convolucionais: avaliação de uma RNC.** 2023. Trabalho de Conclusão de Curso (Graduação em Engenheira de Computação) – Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais, Bambuí - MG, 2023.

OLIVEIRA, B. A. S. *et al.* Sistema para Classificação de Grãos de Café: Uma Comparação Entre Redes Neurais Convolucionais e Processamento Digital de Imagens com MLP. **SBAI**, Ouro Preto, 2019.

RESENDE, F. V.; HABER, L. L.; PINHEIRO, J. B. A cultura do alho. **Embrapa**, 2015.

ROSSINI JUNIOR, C. R.; CAMOLESI, A. R. Um estudo no uso de Redes Neurais Artificiais associada com conceitos de Tecnologia Adaptativa na solução de problemas complexos. **XI Fórum Científico Fema-Anais**, Assis, 2018.

SCHMIDT, A. S. **Desenvolvimento do sistema mecanizado para a cultura do alho(Allium sativum L.) com enfase no plantio**. 1997. Tese (Doutor em Engenharia) – Universidade Federal de Santa Catarina, Florianópolis, 1997.

SILVA, M. T.; OLIVEIRA JUNIOR, V. C. de. Impactos da posição de bulbilhos no plantio, sobre produtividade e qualidade do alho. **Revista FT**, 2024.

VARGAS, A. C. G.; CARVALHO, A. M. P.; VASCONCELOS, C. N. Um estudo sobre Redes Neurais Convolucionais e sua aplicação em detecção de pedestres. In: 4. PROCEEDINGS of the xxix conference on graphics, patterns and images. 2016. v. 1.

WAZLAWICK, R. S. **Metodologia de Pesquisa para Ciência da Computação**. Elsevier Rio de Janeiro, 2009. v. 2.

YANG, K. *et al.* Convolutional Neural Network for Object Detection in Garlic Root Cutting Equipment. **Foods**, MDPI, v. 11, n. 15, p. 2197, 2022.