

**UFRRJ
INSTITUTO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA AGRÍCOLA E
AMBIENTAL**

DISSERTAÇÃO

**DESENVOLVIMENTO DE SISTEMA DE VISÃO COMPUTACIONAL PARA
SELEÇÃO DE PRODUTOS AGRÍCOLAS A PARTIR DA APLICAÇÃO DE REDES
NEURAIS CONVOLUCIONAIS**

JOÃO CÉLIO LUNA DE CARVALHO

2025



**UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
INSTITUTO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA AGRÍCOLA E
AMBIENTAL**

**DESENVOLVIMENTO DE SISTEMA DE VISÃO COMPUTACIONAL PARA
SELEÇÃO DE PRODUTOS AGRÍCOLAS A PARTIR DA REDES NEURAIAS
CONVOLUCIONAIS**

JOÃO CÉLIO LUNA DE CARVALHO

Sob a Orientação do Professor

Dr. Anderson Gomide Costa

e Coorientação do Professor

Dr. Marcus Vinicius Moraes de Oliveira

Dissertação submetida como requisito parcial
para obtenção do grau de **Mestre em
Engenharia Agrícola e Ambiental**, no
Curso de Pós-Graduação em Engenharia
Agrícola e Ambiental, Área de Concentração
em Sistemas Agrícolas

Seropédica, RJ
Julho de 2025

Universidade Federal Rural do Rio de Janeiro
Biblioteca Central/Seção de Processamento Técnico

Ficha catalográfica elaborada
Com os dados fornecidos pelo autor

L331d Luna de Carvalho, João Célio, 1999-
Desenvolvimento de sistema de visão computacional
para seleção de produtos agrícolas a partir da aplicação
de redes neurais convolucionais / João Célio Luna de
Carvalho. - Rio de Janeiro, 2025.
99 f.: il.

Orientador: Anderson Gomide Costa.
Coorientador: Marcus Vinícius Moraes de Oliveira.
Tese(Doutorado). -- Universidade Federal Rural do
Rio de Janeiro, Programa de Pós-Graduação em Engenharia
Agrícola e Ambiental/Engenharia Agrícola e Ambiental,
2025.

1. Imagens digitais. 2. Aprendizado do computador.
3. Raspberry Pi (Computador). 4. Redes neurais
(Computação). I. Gomide Costa, Anderson, 1985-,
orient. II. Moraes de Oliveira, Marcus Vinícius, 1981
, coorient. III Universidade Federal Rural do Rio de
Janeiro. Programa de Pós-Graduação em Engenharia
Agrícola e Ambiental/Engenharia Agrícola e Ambiental.
IV. Título.

UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA AGRÍCOLA E
AMBIENTAL

JOÃO CÉLIO LUNA DE CARVALHO

Dissertação submetida como requisito parcial para obtenção do grau de **Mestre**, no Programa de Pós-Graduação em Engenharia Agrícola e Ambiental, área de concentração em Sistemas Agrícolas

DISSERTAÇÃO APROVADA EM: 29/07/2025

Prof. Dr. Anderson Gomide Costa, UFRRJ
Orientador / Presidente

Profa. Dra. Juliana Lobo Paes, UFRRJ
Membro interno

Prof. Dr. Emanuel Di Tarso Dos Santos Sousa, UFRPE
Membro externo



ATA Nº 86ª ATA DE DEFESA DE DISSERTAÇÃO DE JOÃO CÉLIO/2025 - PPGEAAMB
(12.28.01.00.00.00.40)
(Nº do Documento: 3694)

(Nº do Protocolo: NÃO PROTOCOLADO)

(Assinado digitalmente em 30/07/2025 10:39)

ANDERSON GOMIDE COSTA
PROFESSOR DO MAGISTERIO SUPERIOR
DEPTOENG (12.28.01.00.00.00.44)
Matricula: ###619#5

(Assinado digitalmente em 30/07/2025 15:41)

JULLANA LOBO PAES
PROFESSOR DO MAGISTERIO SUPERIOR
DEPTOENG (12.28.01.00.00.00.44)
Matricula: ###420#8

(Assinado digitalmente em 29/07/2025 15:54)

JOAO CELIO LUNA DE CARVALHO
DISCENTE
Matricula: 2023#####9

(Assinado digitalmente em 29/07/2025 18:40)

EMANOEL DI TARSO DOS SANTOS SOUSA
ASSINANTE EXTERNO
CPF: ###.###.574-##

AGRADECIMENTOS

Agradeço a meu avô Luciano Luna (*in memorian*), por ter me ensinado muito em seus últimos momentos de vida. Infelizmente, faleceu durante meu mestrado. Deixo este trabalho como legado da família Luna.

Agradeço a minha mãe, Luciene Pinheiro Luna, que nunca mediu esforços e me ajudou em todos os momentos, e ao meu pai, Célio Freire de Carvalho, oferecendo suporte quanto ao desenvolvimento e ideias para o projeto.

Agradeço ao meu orientador, que me acompanha desde a graduação, Anderson Gomide Costa, por acreditar no potencial dos meus projetos e por possibilitar suas realizações. Agradeço também por toda orientação, pois sem ela, eu não teria feito nada.

Agradeço a professora Juliana Lobo Paes, que mesmo sem conexão direta com o trabalho, não mediu esforços, me disponibilizando seu laboratório, sala de aula, materiais e conselhos que não só me ajudaram na realização do mestrado, mas em toda minha vida profissional.

Agradeço ao meu coorientador Marcus Vinicius Morais de Oliveira, por colaborar com a execução do trabalho.

Agradeço a Thais Machado e Camila Motta, amigas construídas no mestrado, mas que levarei para toda a vida. Obrigado pelas conversas e discussões que possibilitaram nosso crescimento pessoal e profissional. Sem o apoio de vocês, o mestrado com certeza seria sem graça.

Agradeço ao Hermano Junior e Junior Warmelling, amigas feitas há mais de 13 anos, em jogos online, de outras regiões do estado do Brasil, mas que acompanharam meu desenvolvimento até aqui. Um dia ainda irei conhecê-los pessoalmente.

Agradeço a outras amigas, construídas antes do mestrado, mas que de alguma forma contribuíram para a realização deste trabalho, mesmo que minimamente.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

RESUMO

LUNA DE CARVALHO, João Célio. **Desenvolvimento de sistema de visão computacional para seleção de produtos agrícolas a partir da aplicação de redes neurais convolucionais**. 2025. 99p. Dissertação (Mestrado em Engenharia Agrícola e Ambiental) - Instituto de Tecnologia, Universidade Federal Rural do Rio de Janeiro, Seropédica, 2024.

A demanda por produtos agrícolas de qualidade aumenta conforme o crescimento da produção alimentícia e da população mundial. Equipamentos focados na classificação destes produtos ainda são escassos e as Redes Neurais Convolucionais (CNN) têm sido aplicadas com este propósito. O objetivo desse projeto foi criar um sistema de visão computacional capaz de selecionar produtos agrícolas de forma automatizada em função de atributos de qualidade. Para isso, a câmara seletora de imagens foi composta por um *Raspberry Pi 5* acoplado ao módulo de câmera próprio, monitor LCD e lâmpadas LED para iluminação, com o software para treinamento das CNNs e classificação dos dados armazenados em sua memória interna. Três CNN foram comparadas, sendo duas com arquiteturas pré-estabelecidas (AlexNet e Mobile Net), e uma com arquitetura adaptada para este projeto (OakMoonNet). Um software foi desenvolvido para aquisição das imagens, e outro para a aplicação das CNNs e obtenção das métricas de desempenho de classificação. As CNNs foram aplicadas a seis bancos de imagens, dos quais, dois compostos por imagens obtidas por meio da câmara seletora, com iluminação controlada, e quatro compostos por imagens em condições variadas de ambiente, quantidade e iluminação. A comparação de desempenho das CNNs foi realizada por meio da avaliação dos parâmetros de Acurácia, Precisão, Sensibilidade (*Recall*) e *F1-Score*, obtidos durante o treinamento. A CCN OakMoonNet apresentou o melhor desempenho para os experimentos realizados como avaliação da câmara seletora, obtendo 77% de acurácia (tempo de processamento 23 min e 53 seg) para a classificação da severidade de danos em laranjas e 100% de acurácia para classificação dos frutos de tomates em função maturação (tempo de processamento 4 min e 12 seg). Para os experimentos complementares de validação do software, com dados obtidos fora da câmara seletora, o padrão se repetiu, com a OakMoonNet apresentando sempre a maior acurácia e menor tempo de treinamento, se mostrando mais eficiente na classificação de produtos agrícolas em comparação com as CNNs AlexNet e MobileNet, com baixo custo computacional, alta acurácia e baixo tempo de treinamento.

Palavras-chaves: imagens digitais; aprendizado de máquinas; *Raspberry Pi*, pós-colheita.

ABSTRACT

LUNA DE CARVALHO, João Célio. **Development of a computer vision system for selecting agricultural products using convolutional neural networks.** 2025. 99p. Dissertation (Master's Degree in Agricultural and Environmental Engineering) - Institute of Technology, Federal Rural University of Rio de Janeiro, Seropédica, 2024.

The demand for quality agricultural products increases as food production and the world population grows. Equipment focused on the classification of these products is still scarce, and Convolutional Neural Networks (CNN) have been applied for this purpose. The objective of this project was to create a computer vision system capable of automatically selecting agricultural products based on quality attributes. To this end, the image selection camera consisted of a Raspberry Pi 5 coupled with its own camera module, LCD monitor, and LED lamps for lighting, with software for training CNNs and classifying the data stored in its internal memory. Three CNNs were compared, two with pre-established architectures (AlexNet and Mobile Net), and one with an architecture adapted for this project (OakMoonNet). One software was developed for image acquisition, and another for applying CNNs and obtaining classification performance metrics. The CNNs were applied to six image databases, two of which consisted of images obtained using a selector camera with controlled lighting, and four of which consisted of images under varying environmental conditions, quantities, and lighting. The performance of the CNNs was compared by evaluating the parameters of Accuracy, Precision, Sensitivity (Recall), and F1-Score obtained during training. The OakMoonNet CCN performed best in the experiments conducted to evaluate the sorting camera, achieving 77% accuracy (processing time 23 min and 53 sec) for classifying the severity of damage to oranges and 100% accuracy for classifying tomatoes according to ripeness (processing time 4 min and 12 sec). For the complementary software validation experiments, with data obtained outside the sorting chamber, the pattern was repeated, with OakMoonNet always showing the highest accuracy and shortest training time, proving to be more efficient in the classification of agricultural products compared to AlexNet and MobileNet CNNs, with low computational cost, high accuracy, and short training time.

Keywords: digital images; machine learning; Raspberry Pi; post-harvest.

LISTA DE FIGURAS

Figura 1. Perceptron de McCulloch e Pitts.....	18
Figura 2. Funções de ativação de um neurônio.....	18
Figura 3. Topologias de redes neurais artificiais.....	19
Figura 4. Processamento dos filtros de uma camada convolucional. Imagem completa (A), filtro convolucional (B), nova imagem (C).....	22
Figura 5. Processamento de diferentes camadas Pooling.....	23
Figura 6. Croqui da câmara seletora. Vista superior (A), vista frontal (B), vista isométrica (C) e vista isométrica sombreada com texturas (D).....	31
Figura 7. Exemplo de imagens de frutos de laranja com danos Leves (A), Graves (B) e Muito Graves (C).	35
Figura 8. Exemplo de imagens de frutos de tomate Verde (A), Colorido (B) e Vermelho (C).	36
Figura 9. Exemplo de imagens dos frutos de Maçã (A e B), Laranja (C e D), Pêra (E e F), Morango (G e H) e Tomate (I e J) em ambientes externos e internos, utilizados para o treinamento das CNNs.....	37
Figura 10. Exemplo de imagens de frutos de tomate Verde (A), Amarelo (B) e Vermelho (C) em ambientes externos, utilizados para o treinamento das CNNs.....	38
Figura 11. Exemplo de imagens de frutos de tomate perfeitos e imperfeitos para a Coleta 1 (A e B), Coleta 2 (C e D), Coleta 3 (E e F) e Coleta 4 (G e H), utilizados para o treinamento das CNNs.	39
Figura 12. (A) Construção do padrão Time History Speckle Pattern (THSP), (B) padrão THSP obtido em uma batata na região do impacto mecânico 2 horas após o impacto mecânico e (C) padrão THSP obtido em uma batata na região do impacto mecânico 72 horas após o impacto mecânico.....	41
Figura 13. Disposição experimental para realização do impacto mecânico na batata capaz de gerar o internal bruising.....	42
Figura 14. A câmara seletora em vista frontal, com sua tampa frontal aberta (A) e fechada (B).....	44
Figura 15. Topo da câmara seletora com seus componentes (A) e durante a captura de imagens (B).....	45
Figura 16. Topo da câmara seletora com o software para treinamento das CNNs em operação.	46

Figura 17. Estrutura da OakMoonNet.....	47
Figura 18. Interface de usuário do algoritmo desenvolvido.....	49
Figura 19. Interface de usuário do algoritmo desenvolvido preenchida.....	51
Figura 20. Fluxograma de funcionamento do software desenvolvido.....	51
Figura 21. Respostas de Loss para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste.	53
Figura 22. Respostas de Acurácia para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste.	54
Figura 23. Matriz de confusão para classificação das injurias presentes no epicarpo das laranjas utilizando as CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C).....	56
Figura 24. Respostas de Loss para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste.	59
Figura 25. Respostas de Acurácia para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste. ...	60
Figura 26. Matriz de confusão para classificação dos frutos de tomate quanto a maturação utilizando as CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C).....	62
Figura 27. Respostas de Loss para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste.	64
Figura 28. Respostas de Acurácia para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste.	65
Figura 29. Matriz de confusão para classificação dos dados obtidos pela internet utilizando as CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C).....	67
Figura 30. Respostas de Loss para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste.	70
Figura 31. Respostas de Acurácia para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste. ...	71
Figura 32. Matriz de confusão para classificação dos frutos de tomate na lavoura quanto a maturação utilizando as CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C).....	72
Figura 33. Respostas de Loss para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste.	75
Figura 34. Respostas de Acurácia para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste.	76

Figura 35. Matriz de confusão para classificação dos frutos de tomate quanto a presença de injúrias no epicarpo utilizando as CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C)....	78
Figura 36. Respostas de Loss para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste.	80
Figura 37. Respostas de Acurácia para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste.	81
Figura 38. Matriz de confusão para classificação dos THSPs das batatas em horas após o impacto mecânico (HAIM) e em dias após o impacto mecânico (DAIM) utilizando as CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C).....	83

LISTA DE TABELAS

Tabela 1. Arquitetura da CNN AlexNet.....	25
Tabela 2. Arquitetura da CNN MobileNet.....	26
Tabela 3. Bibliotecas utilizadas no software desenvolvido.....	32
Tabela 4. Gravidade dos defeitos e características do fruto de laranja. Fonte: CEAGESP (2011).....	35
Tabela 5. Classificação da maturação do tomate quanto a cor. Fonte: CEAGESP (2003).....	36
Tabela 6. Divisão dos THSP obtidos durante a evolução do internal bruising em batatas nas classes Horas após o impacto mecânico (HAIM) e Dias após o impacto mecânico (DAIM).....	43
Tabela 7. Arquitetura da OakMoonNet.....	48
Tabela 8. Valores de Loss e Acurácia obtidos para os dados de treinamento e teste nas CNNs OakMoonNet, AlexNet e MobileNet.....	55
Tabela 9. Métricas de desempenho das CNNs OakMoonNet, AlexNet e MobileNet.....	57
Tabela 10. Valores de Loss e Acurácia obtidos para os dados de treinamento e teste nas CNNs OakMoonNet, AlexNet e MobileNet.....	61
Tabela 11. Métricas de desempenho das CNNs OakMoonNet, AlexNet e MobileNet.....	63
Tabela 12. Valores de Loss e Acurácia obtidos para os dados de treinamento e teste nas CNNs OakMoonNet, AlexNet e MobileNet.....	66
Tabela 13. Métricas de desempenho das CNNs OakMoonNet, AlexNet e MobileNet.....	68
Tabela 14. Valores de Loss e Acurácia obtidos para os dados de treinamento e teste nas CNNs OakMoonNet, AlexNet e MobileNet.....	71
Tabela 15. Métricas de desempenho das CNNs OakMoonNet, AlexNet e MobileNet.....	73
Tabela 16. Valores de Loss e Acurácia obtidos para os dados de treinamento e teste nas CNNs OakMoonNet, AlexNet e MobileNet.....	77
Tabela 17. Métricas de desempenho das CNNs OakMoonNet, AlexNet e MobileNet.....	79
Tabela 18. Valores de Loss e Acurácia obtidos para os dados de treinamento e teste nas CNNs OakMoonNet, AlexNet e MobileNet.....	82
Tabela 19. Métricas de desempenho das CNNs OakMoonNet, AlexNet e MobileNet.....	84

LISTA DE ABREVIATURAS E SIGLAS

Ac	Acurácia
ADAM	<i>Adaptative Moment Estimation</i>
BSL	<i>Biospeckle Laser</i>
CNN	<i>Convolutional Neural Network</i>
DAIM	Dias após o impacto mecânico
DW	<i>Depthwise</i>
fp	Falso positivo
fn	Falso negativo
HAIM	Horas após o impacto mecânico
Pe	Precisão
Re	Recall
SBC	<i>Single Board Computer</i>
THSP	<i>Time History Spacial Pattern</i>
vp	Verdadeiro positivo
vn	Verdadeiro negativo
Wj	Peso
Xj	Dado de entrada

SUMÁRIO

AGRADECIMENTOS.....	4
RESUMO.....	6
ABSTRACT.....	7
1. INTRODUÇÃO.....	15
2. OBJETIVOS.....	16
2.1. Objetivo geral.....	16
2.2. Objetivos específicos.....	16
3. REVISÃO DE LITERATURA.....	17
3.1. Histórico da neurocomputação.....	17
3.2. Estrutura das redes neurais.....	19
3.3. Rede Neural Convolucional (<i>Convolutional Neural Network</i> – CNN).....	21
3.4. Arquiteturas CNNs utilizadas.....	24
3.4.1. AlexNet.....	24
3.4.2. MobileNet.....	25
3.5. Uso da CNN na agricultura.....	26
3.6. Uso das CNNs em imagens provenientes da técnica óptica do <i>Biospeckle Laser</i> (BSL) 27	
3.7. Computadores de placa única.....	28
4. MATERIAIS E MÉTODOS.....	29
4.1. Desenvolvimento da câmara seletora.....	30
4.2. Desenvolvimento do software Classification Fruit UFRRJ.....	31
4.3. Avaliação de desempenho das CNNs.....	33
4.4. Descrição dos experimentos de teste e validação da câmara seletora.....	34
4.4.1. Primeiro experimento: Uso da câmara seletora para a classificação dos frutos de laranja em função das injúrias.....	34
4.4.2. Segundo experimento: Uso da câmara seletora para classificação dos frutos de tomate quanto a maturação.....	36
4.5. Descrição dos experimentos de teste e validação do software para outras aplicações 37	
4.5.1. Terceiro experimento: Classificação de tipos de frutos quanto utilizando	

imagens obtidas na internet.....	37
4.5.2. Quarto experimento: Classificação de frutos de tomates na lavoura quanto ao nível de maturação.....	38
4.5.3. Quinto experimento: Classificação de frutos de tomate quanto a presença de injúrias no epicarpo.....	39
4.5.4. Sexto experimento: Classificação de imagens <i>Time History Spacial Pattern</i> (THSP) para monitoramento de injúrias internas em batatas.....	40
5. RESULTADOS E DISCUSSAO.....	43
5.1. Usabilidade da Câmara seletora.....	43
5.2. Desenvolvimento da CNN Customizada.....	46
5.3. Desenvolvimento do software Classification Fruit UFRRJ.....	48
5.4. Experimentos de teste e validação da câmara seletora.....	52
5.4.1. Primeiro experimento – Uso da câmara seletora para a classificação dos frutos de laranja em função das injúrias.....	52
5.4.2. Segundo experimento – Uso da câmara seletora para classificação dos frutos de tomate quanto a maturação.....	58
5.5. Outros experimentos de validação do software.....	63
5.5.1. Terceiro experimento – Classificação de frutos quanto ao seu tipo por meio de imagens obtidas na internet.....	63
5.5.2. Quarto experimento – Classificação de frutos de tomates na lavoura quanto ao nível de maturação.....	69
5.5.3. Quinto experimento – Classificação de frutos de tomate quanto a presença de injúrias no epicarpo.....	74
5.5.4. Sexto experimento – Classificação de imagens <i>Time History Spacial Pattern</i> (THSP) de batatas para detecção do internal bruising quanto ao tempo de impacto.....	79
6. CONSIDERAÇÕES FINAIS.....	85
REFERÊNCIAS.....	86

1. INTRODUÇÃO

O avanço da tecnologia e consequente melhora da performance computacional permitiu a realização de atividades anteriormente consideradas impossíveis. O desenvolvimento tecnológico ocorre em diversas vertentes da sociedade, e consequentemente o crescimento populacional impulsiona o desenvolvimento agrícola quanto sua mecanização e automação dos processos do plantio ao pós-colheita (Lima, Silva e Iwata, 2019, Xiao, Wang e Jiang, 2024). Com isso, a utilização de redes neurais e inteligência artificial está cada vez mais aplicada a área agrícola.

A necessidade de aumento de produção de alimentos, faz com que a seleção e classificação de produtos agrícolas, conforme a normas de controle de qualidade e exigência do mercado consumidor, necessite ser mais rápida e precisa. Como um dos principais fatores de escolha de um produto agrícola é baseado na aparência e valor nutricional (Palumbo et al. 2023), a visão computacional é solução promissora com grande potencial para aplicação nas etapas de colheita e pós-colheita. O uso da visão computacional se dá por meio da entrada de imagens em modelos de aprendizagem de máquina, podendo, dentre as diversas aplicações, auxiliar na classificação e seleção dos frutos por meio da correlação dos dados de entrada com os fatores físicos e químicos de interesse (Oliveira, 2021).

Com o avanço da computação, aumento de poder de processamento e a facilidade de obtenção de microprocessadores de placa única e de baixo custo, algoritmos de aprendizagem de máquinas conhecidos como *Deep learning* (Redes Neurais Profundas) ganham destaque (Khan, 2024). Sendo uma variante das Redes Neurais Profundas, as Redes Neurais Convolucionais (*Convolutional Neural Networks* - CNN) são dedicadas exclusivamente ao processamento de imagens. Sua vantagem é identificar automaticamente as características relevantes sem a necessidade da supervisão humana (Gu et al., 2018). Há diversas arquiteturas (*Backbones*) de CNNs disponíveis para uso gratuito, como a AlexNet (Krizhevsky, Sutskever e Hinton, 2017) e a MobileNet (Howard et al., 2017) que apresentam aplicabilidade para classificação de produtos agrícolas. Uma alternativa que vem sendo testada para aplicação das CNNs para classificação de produtos agrícolas é a criação de arquiteturas autorais ou modificação de CNNs já existentes, como a SortNet (Chakraborty et al., 2023) e a YOLOv4 modificada (Chen et al., 2020), respectivamente.

A aplicação da CNN para avaliação de qualidade de produtos agrícolas, especialmente frutos, vem apresentando sucesso como ferramenta de auxílio na tomada de decisão. Abekoon

et al. (2024) criaram dois modelos CNN próprios para classificar frutos de tomate quanto a maturação, obtendo 99% e 98% de acurácia, respectivamente. De Luna et al. (2020) desenvolveram um sistema de detecção de defeitos em frutos de tomate, utilizando o *Raspberry Pi* e três diferentes arquiteturas CNN, obtendo 98,75%, 58,33% e 64,58% de acurácia, respectivamente.

Assim, a aplicação de algoritmos com arquiteturas convolucionais já existentes e arquiteturas customizadas devem ser testadas com intuito de buscar soluções para problemáticas de culturas agrícolas importantes em âmbito nacional. A possibilidade de sua implementação em sistemas ópticos mais amplos que automatizem os processos de seleção e classificação dos produtos agrícolas em função de parâmetros de qualidade, potencializa seu uso e pode ser atender a necessidade de produtores rurais, mercado de exportação, indústrias alimentícias ou o mercado consumidor interno.

2. OBJETIVOS

2.1. Objetivo geral

Desenvolver uma câmara seletora capaz de utilizar um sistema de visão computacional associada a Redes Neurais Convolucionais (CNN) para classificar produtos agrícolas em função de atributos de qualidade.

2.2. Objetivos específicos

- a) Desenvolver uma câmara seletora experimental utilizando o microcontrolador Raspberry Pi com módulo de câmera RGB acoplado para realização da aquisição coleta das imagens visando a realização de classificação utilizando arquiteturas de CNNs.
- b) Desenvolver uma CNN de baixo custo computacional, capaz de realizar a classificação de produtos agrícolas quanto a atributos de qualidade variedade, maturação e injurias conforme a necessidade do usuário.
- c) Desenvolver um software com interface visual que possa ser utilizada de forma integrada com a câmara seletora, e que seja capaz de organizar os dados de dataset

(banco de imagens), e realizar o processo de classificação das imagens utilizando a CNN customizada ou CNNs de arquiteturas pré-existentes.

- d) Aplicar as CNNs para realizar classificação de produtos agrícolas em diferentes contextos (tipo de fruto, grau de maturação e nível de injúrias) e comparar o desempenho da CNN customizada com as CNNs AlexNet e MobileNet.

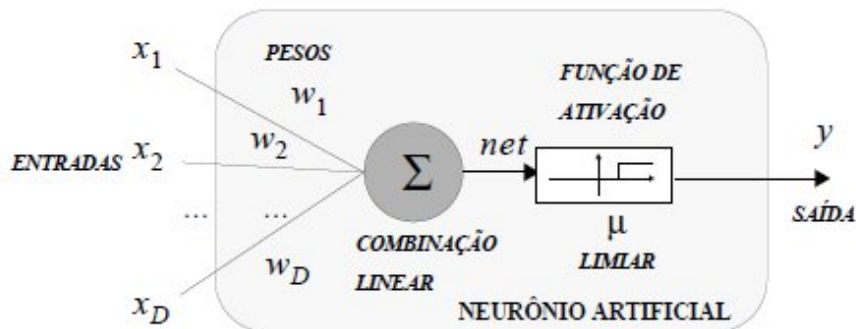
3. REVISÃO DE LITERATURA

3.1. Histórico da neurocomputação

A capacidade do cérebro humano ser robusto, tolerantes a falhas, ter alta capacidade de aprendizagem, conseguir processar informações incertas e possuir um enorme paralelismo com diversos neurônios ativos inspirou os cientistas a criarem um modelo computacional análogo ao seu funcionamento (Kopiler et al. 2019; Rauber, 2005). O primeiro modelo de neurônio artificial foi proposto por McCulloch e Pitts (1943), culminando em pesquisas na área e obtenção do modelo mais simples e conhecido de neurônio artificial, o *Perceptron* de Rosenblatt (1958). O *Perceptron* é um algoritmo com neurônios de entrada e saída, no qual toda sua “inteligência” está concentrada em seus pesos internos. É capaz de resolver somente problemas linearmente separáveis.

O neurônio artificial proposto por McCulloch e Pitts (1943) simula a realidade biológica (Figura 1). A informação recebida de outros neurônios entra em D entradas x_j (sinapses) do neurônio processados. Com cada entrada x_j associada a um peso w_j , ocorre uma combinação linear das entradas multiplicadas pelos pesos correspondentes, e o resultado é o valor net. Se esse valor ultrapassa um limiar, o neurônio responde com o valor 1, caso contrário, 0. A ativação do neurônio é baseada na função de *Heaveside*, conhecida também como função degrau ou função escada (Kopiler et al. 2019; Rauber, 2005).

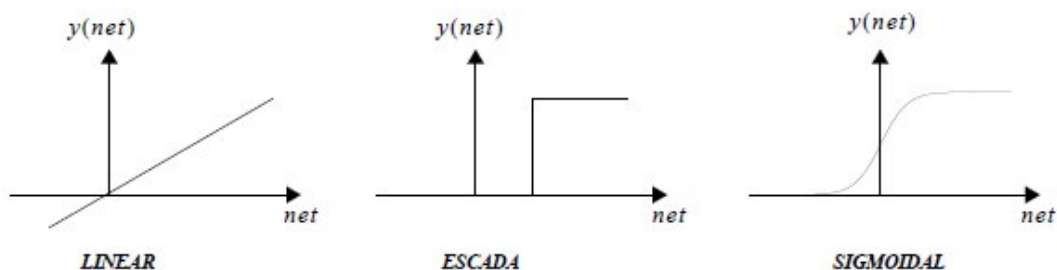
Figura 1. *Perceptron* de McCulloch e Pitts.



Fonte: (Rauber, 2005).

Há outras funções de ativação (Figura 2), como a função linear, que produz uma saída contínua, a função de escada, que produz saída não-linear discreta, e a função sigmoideal, que produz uma saída contínua e diferenciável em todo o domínio de $\mathbb{R}$, com derivada simples e estritamente monótona (Rauber, 2005).

Figura 2. Funções de ativação de um neurônio.



Fonte: (Rauber, 2005).

A limitação de resolução de problemas do *Perceptron* foi criticada por Minsky e Papert (1969). Ser inoperável em problemas de “Ou Exclusivo” (XOR) o tornava inviável para a maioria dos problemas do mundo real. Problemas XOR recebem dois valores, verdadeiros ou falsos, retornando o resultado como verdadeiro apenas quando um dos dois valores é verdadeiro, mas não ambos. A solução do problema era conhecida, bastava-se acrescentar mais uma camada de neurônios, conhecida como camada oculta. Contudo, era necessário um algoritmo capaz de treinar os pesos da rede multicamada, além de um alto poder computacional para a época (Rauber, 2005). O problema foi solucionado 19 anos depois por Rumelhat et al.

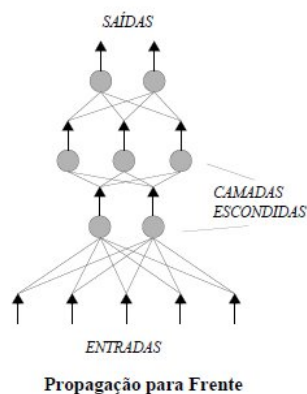
(1985), com o desenvolvimento do algoritmo de retro propagação do erro (*error backpropagation*), do qual busca “propagar” o erro da camada de saída para corrigir os erros das camadas anteriores, sendo uma retroação ou retroalimentação.

3.2. Estrutura das redes neurais

A estrutura de uma rede neural pode ser dividida em três grupos. O primeiro grupo representa as camadas de entrada, onde são apresentados os padrões para a rede. O segundo grupo compreende as camadas intermediárias, escondidas ou ocultas, onde a maior parte do processamento é realizado, implicando no poder de processamento de problemas não linearmente separáveis. O terceiro grupo é composto pelas camadas de saída, onde o resultado é apresentado.

Um dos principais modelos são as redes *feedforward* (propagação para frente) (Figura 3), que possuem o algoritmo de treinamento mais comum, o *backpropagation* (propagação para trás) (Kopiler et al. 2019). Nesses modelos, o fluxo de informação é unidirecional, com o treinamento se “propagando para frente”, com neurônios agrupados em camadas, e a atualização dos pesos sendo “propagada para trás” (Rauber, 2005).

Figura 3. Topologias de redes neurais artificiais.



Fonte: (Rauber, 2005).

A utilização de aprendizagem supervisionada em redes *feedforward* é recomendada, pois possuem um conjunto de dados de treinamentos previamente rotulado, onde o algoritmo é treinado buscando as características que diferenciam os grupos entre si (Russel e Norvig, 2020).

O processo de aprendizagem envolve a obtenção dos melhores pesos para cada conexão de neurônios. Para isso, um algoritmo de aprendizagem julga a qualidade do peso e o recalcula

quando necessário, sofrendo uma modificação no próximo treinamento, chamado de iteração. Normalmente, inicializa-se o processo com pesos aleatórios que vão se ajustando com as iterações (Rauber, 2005).

No *Perceptron* simples, com um neurônio de entrada e um de saída, o peso é feito com base no Erro e na taxa de aprendizagem definida pelo usuário. O Erro é a diferença entre a resposta correta e a resposta errada. O novo peso é definido realizando a soma do peso antigo com a taxa de aprendizagem, multiplicado pelo valor de entrada multiplicado pelo erro. O peso para todas as camadas é atualizado e o processo de iteração continua até que o resultado atingido seja satisfatório (Rauber, 2005). Existem outros algoritmos de aprendizagem, como a descida do gradiente, que associada ao Delta, atualiza os pesos por meio do *backpropagation*.

A descida do gradiente é uma solução iterativa que busca o menor erro na curva de gradiente, visando alcançar o mínimo global e evitar os mínimos locais. A regra do delta e o valor do momento definido pelo usuário buscam evitar mínimos locais. O objetivo é, portanto, encontrar a combinação de pesos em que o erro é o menor possível. A seleção da função de ativação sigmoide, neste caso, se dá pela sua fácil derivada, facilitando o processo (Rauber, 2005). A regra de Delta, ou regra de Widrow e Hoff (1960) é necessária para o cálculo do gradiente e a atualização dos pesos por meio do *backpropagation*. Utiliza-se da derivada da função sigmoide e o erro das respostas, multiplicando-se os resultados. Calcula-se o delta das camadas de saída e camadas ocultas. O valor de delta será utilizado na correção dos erros por meio do *backpropagation* (Rauber, 2005).

Com os valores de delta, é necessário a escolha manual pelos valores de momento e taxa de aprendizagem. O momento pode ser um valor fixo ou variável, por meio da utilização da Estimativa do Momento Adaptativa (*Adaptative Moment Estimation* - ADAM). A taxa de aprendizagem define o quão rápido um algoritmo vai aprender. Um valor próximo à 1 define alta convergência, mas com risco de perder o mínimo global. Já um valor próximo a 0 torna o processo mais lento, mas possui mais chances de chegar a um mínimo global. O momento busca escapar dos mínimos locais, mas nem sempre funciona. Ele define o quão confiável foi a última alteração. Um valor alto aumenta a velocidade da convergência, e um valor baixo pode evitar mínimos locais. Com os valores definidos, a atualização dos pesos ocorre por meio da soma entre a multiplicação dos valores de peso e momento em conjunto da multiplicação dos valores de entrada, delta e taxa de aprendizagem (Rauber, 2005)

3.3. Rede Neural Convolucional (*Convolutional Neural Network* – CNN)

A CNN é uma variante das redes neurais multicamadas, inspirada no processo biológico de processamento de imagens (Wang et al., 2020; Li et al., 2021; Fang et al., 2020). Sua vantagem é identificar automaticamente as características relevantes sem a necessidade de um supervisor (Gu et al., 2018). A inspiração biológica vem dos cérebros dos gatos, onde o CNN busca reproduzir a sequência complexa de células que formam o córtex visual (Hubel e Wiesel, 1962). Assim como no sistema biológico, a CNN tira proveito das cores, e experimentos psicológicos mostraram a como a cor contribui altamente para a memorização e conhecimento (Wichmann, Sharpe e Gegenfurtner, 2002)

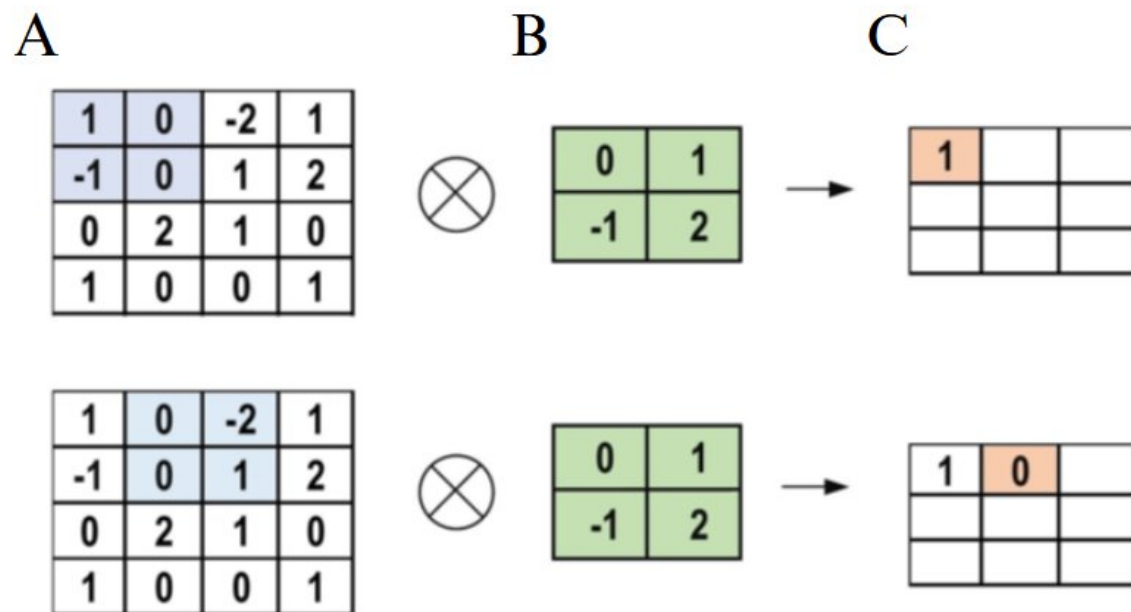
Diferente das redes neurais tradicionais, totalmente conectadas, os pesos da CNN são compartilhados e ocorrem menos conexões neurais, utilizando menos parâmetros e consequentemente simplificando seu treinamento e uso (Alzubaidi et al., 2021). Para que isso seja possível, pequenas regiões da imagem são analisadas. Uma CNN é composta por camadas convolucionais, camadas de filtro (*pooling* ou *sub-sampling*) e camadas totalmente conectada a todos os neurônios anteriores. As CNNs são capazes de aplicar filtros em dados visuais por meio da relação da vizinhança entre os pixels, onde é utilizado para detecção e reconhecimento em imagens e vídeos (Rosa et al., 2018). As CNNs são altamente organizadas e eficientes na extração de características de interesse. Sua implementação em larga escala é muito mais fácil se comparado a outras redes neurais.

Na CNN, os dados inicialmente são aplicados na camada Convolucional, composta por diversos neurônios, onde cada um é responsável pela aplicação de um “filtro convolucional”, ou kernel, na imagem. É o elemento mais importante nas CNN, sendo o nome da rede neural. O filtro aplicado trata-se de uma matriz de valores que correspondem ao seu peso. Inicialmente, números aleatórios são distribuídos no início do processo, e o treinamento da CNN ajusta os valores para extrair as características julgadas como significantes durante o treinamento (Alzubaidi et al., 2021).

A imagem aplicada nas camadas convolucionais são escaneados pelos filtros convolucionais em uma escala menor de pixels, transformando os valores dos pixels em valores correspondentes ao seu peso. O processo é realizado até o fim da varredura completa da imagem, consequentemente gerando uma imagem de menor escala. O resultado do processamento pode ser visto na Figura 4, onde a imagem completa (A) é varrida pelo filtro (B)

e os pesos são aplicados em cada pixel por meio da multiplicação. O resultado da operação gera uma nova imagem (C) (Alzubaidi et al., 2021).

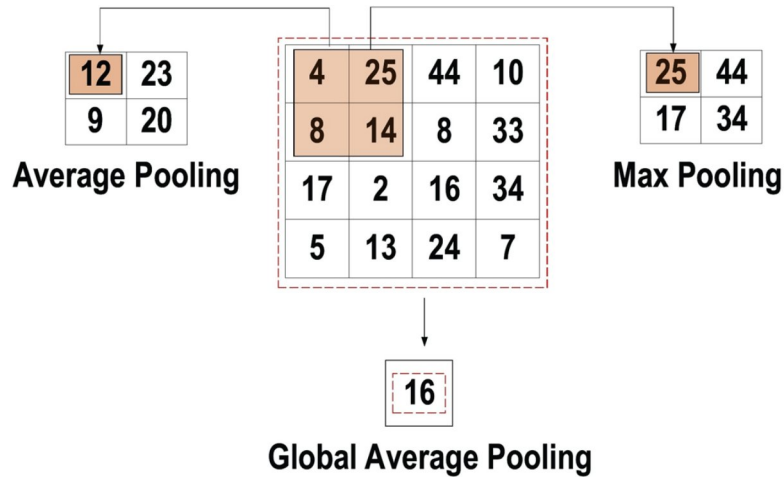
Figura 4. Processamento dos filtros de uma camada convolucional. Imagem completa (A), filtro convolucional (B), nova imagem (C).



Fonte: (Alzubaidi et al., 2021).

As camadas de Agrupamento Local ou Global (Pooling Layers) combinam as saídas de agrupamentos de outros neurônios em um único neurônio da camada posterior. A função dessa camada, portanto, é reduzir o tamanho da imagem para que seja realizado seu processamento. Dos métodos mais usados, tem-se o máximo (max), mínimo (min) e médio (GAP) pooling. O Max Pooling busca o maior valor dos neurônios, o Min pooling busca o menor valor e o GAP pooling realiza a média global dos valores (Alzubaidi et al., 2021, Rosa et al., 2018). O funcionamento das técnicas de pooling pode ser visto na Figura 5.

Figura 5. Processamento de diferentes camadas *Pooling*.



Fonte: (Alzubaidi et al., 2021).

Assim como em toda rede neural, a função de ativação é imprescindível para o funcionamento das CNNs. As funções de ativação utilizáveis são a Sigmoid, Tangencial, Softmax, ReLU, Leaky ReLU, Noisy ReLU e Parametric Linear Units. Contudo, a função mais utilizada é a ReLU, convertendo todos os valores inseridos em resultados positivos e utilizando o maior valor encontrado (Alzubaidi et al., 2021). O final da arquitetura CNN, estão presentes as Camadas Totalmente Conectadas ou Fully Connected Layers (FC). Ela é conectada com todos os neurônios da camada anterior, realizando o papel de classificador, sendo a saída final do algoritmo (Alzubaidi et al., 2021, Rosa et al., 2018).

Algumas funções são utilizadas na saída da rede neural para calcular o *Loss* durante o treinamento, calculando a diferença entre o valor estimado e real, aprimorando os pesos no próximo treinamento. A Função *Loss* quantifica o quão discrepante é a qualificação dos dados estimados em comparação com os valores reais do dataset (Shivashankar, Pandey e Zakwan, 2022).

O treinamento de uma rede neural pode apresentar três possíveis situações, sendo elas *overfitting*, *underfitting* e o *justfitting*. O *Overfitting* gera uma classificação tendenciosa para os dados de treinamento, sendo ineficaz com os dados de teste. O *Underfitting* promove uma rede neural com um treinamento e teste ineficazes. O *Justfitting* ocorre quando o modelo é bom no treinamento e teste. (Alzubaidi et al., 2021, Rosa et al., 2018). O *Overfitting* e o *Underfitting* podem ser evitados nas CNNs por meio das técnicas de *Dropout*, *Batch Normalization* e *Data augmentation*.

O *Dropout* é uma técnica altamente usada na generalização dos dados. Em cada época de treinamento, neurônios são aleatoriamente desativados. Assim, o modelo é forçado a

aprender diferentes “habilidades” com seus neurônios ativos. Vale ressaltar que o neurônio aleatório não fará parte do *backpropagation* ou *forward propagation* (Alzubaidi et al., 2021, Rosa et al., 2018). Já o *Batch Normalization* avalia a performance dos dados de saída (Ioffe e Szegedy, 2015). Ele funciona por meio da distribuição gaussiana, subtraindo a média e dividindo pelo desvio padrão para normalizar os dados de cada camada. Assim, é definido o “intervalo de covariância interno” e são ajustados os pesos. Das vantagens desse método, tem-se o fato de prevenir problemas de descida do gradiente e perda do mínimo global, controle da inicialização dos pesos de baixa qualidade, redução do tempo de processamento na busca dos mínimos globais, redução da dependência do treinamento para ótimos resultados e redução de *overfitting* (Alzubaidi et al., 2021).

O *data augmentation* aumenta artificialmente o conjunto de dados, aplicando diferentes técnicas de manipulação de imagens, como a alteração do espaço de cores, corte, rotação, translação e injeção de ruídos (Alzubaidi et al., 2021). A imagem de origem dos dados gerados não é utilizada nas etapas de treinamento ou teste. Esta técnica é útil principalmente quando existem poucos dados.

3.4. Arquiteturas CNNs utilizadas

3.4.1. AlexNet

A AlexNet (Krizhevsky, Sutskever e Hinton, 2017) foi desenvolvida com o intuito de reconhecimento de imagens, sendo pioneira neste ramo. Sua arquitetura é composta por cinco camadas convolucionais utilizando a função de ativação *ReLU*, sendo a primeira, segunda e quinta seguidas de uma camada de *Max Pooling*. Em seu final, apresenta três camadas totalmente conectadas, finalizada por um classificador utilizando a função de ativação *Softmax*, sendo o número de classes finais correspondente ao número de classes presentes no dataset utilizado. A função *Loss* utilizada é a *Sparse Categorical Crossentropy*. A imagem de entrada deve estar na proporção 227x227x3, correspondendo a altura, largura e profundidade da imagem, respectivamente. A profundidade, neste caso, representa os espaços de cor Vermelho (R), Verde (G) e Azul (B). Por ser uma rede neural com maior complexidade quanto ao número de camadas e quantidade de filtros, seu custo computacional é maior se comparado a outras redes. A Tabela 1 apresenta arquitetura da AlexNet.

Tabela 1. Arquitetura da CNN AlexNet

Camadas	Filtros	Tamanho
Imagem	1	227x227x3
<i>Convolutional</i>	96	55x55x96
<i>Max Pooling</i>	96	27x27x96
<i>Convolutional</i>	256	27x27x256
<i>Max Pooling</i>	256	13x13x256
<i>Convolutional</i>	384	13x13x384
<i>Convolutional</i>	384	13x13x384
<i>Convolutional</i>	256	13x13x256
<i>Max Pooling</i>	256	6x6x256
<i>Flatten</i>	-	9216
<i>Fully connected</i>	-	4096
<i>Fully connected</i>	-	4096
<i>Fully connected</i>	-	1000
<i>Fully connected</i>	-	CLASSES

Fonte: Elaborado pelo autor

3.4.2. MobileNet

A MobileNet (Howard et al., 2017) trabalha com detecção e classificação de objetos, sendo uma das arquiteturas mais simplificadas de CNNs, visando comprimir a etapa de convolação, fazendo com que esta seja mais leve (Santos et al., 2022). Para isso, utilizam convolações separáveis em profundidade (Howard et al., 2017). A MobileNet possui 14 camadas convolucionais e 14 camadas de convolação separável, sempre acompanhadas de uma camada de *Batch Normalization*. A camada de convolação separável (*Convolutional Depthwise* (DW)) (Kaiser, Gomez e Chollet, 2017) foi desenvolvida para a MobileNet, visando a realização de convolações mais rápidas, facilitando seu uso em dispositivos móveis. Posteriormente, a arquitetura apresenta uma camada *average pooling* e uma camada totalmente conectada de 1000 categorias, utilizando a função de ativação Softmax. A função *Loss* utilizada é a *Sparse Categorical Crossentropy*. O formato de entrada da imagem deve ser na resolução 224x224x3, e a quantidade de classes final depende do número de classes presentes no dataset

utilizado. A Tabela 2 apresenta arquitetura da MobileNet. (Howard et al., 2017; Mundim Filho, 2021).

Tabela 2. Arquitetura da CNN MobileNet

Camadas	Filtros	Tamanho
Imagem	1	224x224x3
<i>Convolutional</i>	32	112x112x32
<i>Convolutional DW</i>	32	112x112x32
<i>Convolutional</i>	64	112x112x64
<i>Convolutional DW</i>	64	56x56x64
<i>Convolutional</i>	128	56x56x128
<i>Convolutional DW</i>	128	56x56x128
<i>Convolutional</i>	128	56x56x128
<i>Convolutional DW</i>	128	28x28x128
<i>Convolutional</i>	256	28x28x256
<i>Convolutional DW</i>	256	28x28x256
<i>Convolutional</i>	256	28x28x256
<i>Convolutional DW</i>	256	14x14x256
<i>Convolutional</i>	512	14x14x512
<i>Convolutional DW</i>	512	14x14x512
<i>Convolutional</i>	512	14x14x512
<i>Convolutional DW</i>	512	7x7x512
<i>Convolutional</i>	1024	7x7x1024
<i>Convolutional DW</i>	1024	7x7x1024
<i>Convolutional</i>	1024	7x7x1024
<i>Avg. Pooling</i>	1024	1x1x1024
<i>Fully connected</i>	-	CLASSES

Fonte: Elaborado pelo autor

3.5. Uso da CNN na agricultura

A pesquisa e utilização de CNNs na agricultura tem tomado grandes proporções nos últimos anos. Uma das principais aplicações está na classificação de frutos em função de atributos de qualidade. Diversos trabalhos utilizam essas classificações para distinção de tipos

de frutos como observado em Kumar et al. (2021) e Goyal, Kumar e Verma (2023). Embora este tipo de classificação não represente uma aplicação prática, normalmente ela é utilizada para testar as capacidades de diferenciação das classes pelas CNNs, seja quanto a forma ou cor do produto agrícola.

As CNNs também têm apresentado sucesso na classificação de produtos agrícolas em função de parâmetros de qualidade como na avaliação de estágio de maturação conforme realizado no trabalho de Hendrawan et al. (2021) para classificação de pimentas e Habaragamuwa et al. (2018) na classificação de morango. A classificação em função de danos e injúrias também é uma aplicação explorada como nos trabalhos de Hadipour-Ronki et al. (2023), Elaraby, Hamdu e Alanazi (2022) e Chakraborty et al. (2023) em laranjas, e Khatun et al. (2023) e Chaturvedi, Sharma e Janghel (2023) para classificação de tomates.

Outra abordagem estudada é o aprimoramento de arquiteturas de CNNs já existentes, buscando aumentar a acurácia na classificação dos produtos agrícolas e reduzir o tempo de processamento das imagens. Exemplos destes estudos podem ser observados em Cui et al. (2021), que buscaram aprimorar a CNN YOLOv4 através da rede MobileNetv3, aplicado a classificação de frutos de morango, Tsai et al. (2023) utilizaram a associação das CNNs BoTNet, ShuffleNet e GhostNet com a YOLOv5m, aplicado a detecção de frutos de tomate, e Phan et al. (2023), que utilizaram a associação das CNNs ResNet-50, ResNet-101 e EfficientNet-B0 com a YOLOv5m, aplicado a classificação de frutos de tomate.

3.6. Uso das CNNs em imagens provenientes da técnica óptica do *Biospeckle Laser* (BSL)

A crescente demanda por técnicas precisas e eficientes para o monitoramento de qualidade de produtos agrícolas tem impulsionado a aplicação de métodos inovadores, especialmente aqueles que possibilitem a obtenção de dados de forma rápida, não invasiva e automatizada. A técnica óptica conhecida como *biospeckle laser* (BSL) tem se mostrado exitosa em aplicações na agricultura de alimentos (Zdunek et al., 2014), como no caso de identificação de mofo em citrus (Yang et al. (2024), mensuração de indicadores de qualidade de frutos (Pavarin et al., 2024), e análise do vigor e poder de germinação de sementes (Braga et al., 2024; Thakur et al., 2024).

O BSL permite caracterizar a atividade de um material biológico ou dinâmico por meio da captura de uma série temporal de imagens de padrões de interferência estática, gerados pela iluminação e reflexão de uma luz coerente, chamado de speckle (Goodman, 1975). O biospeckle laser ocorre quando a superfície é iluminada por um laser, e cada ponto de reflexão atua como

uma fonte secundária de ondas esféricas, dando origem ao padrão de interferência estática speckle, provocando o aparecimento de regiões claras e escuras, e apresentam variações temporais em caso de atividade física ou biológica (Rabal e Braga, 2018).

A busca por sistemas que permitam a avaliação automatizada a partir de técnicas ópticas, como o BSL, tem sido objeto de pesquisas recentes (Park e Ahn, 2024) e Thakur et al., (2024). Esses sistemas oferecem vantagens significativas em relação aos métodos convencionais, como a não destruição das amostras e a possibilidade de automação nos processos de controle de qualidade com mensurações instantâneas (Hu et al., 2024).

Uma das formas mais tradicionais de realizar a análise dos BSL é pela representação do *Time History Spacial Pattern* (THSP) proposto por Ouluamara et al., (1989). O THSP é uma representação gráfica dos padrões de biospeckle laser, do qual organiza cronologicamente a mesma coluna de cada imagem capturada em sequência, formando uma nova imagem, composta por elas. Assim, as alterações da intensidade de pixels ao longo do tempo são representadas na direção horizontal, enquanto as alterações espaciais se encontram na direção vertical.

Recentemente, as CNNs vem sendo estudadas como forma de análise do BSL. No entanto, vem se observando que a predominância por abordagens onde se aplica os algoritmos de aprendizagem profunda diretamente nos padrões sucessivos do biospeckle obtido ao longo do tempo durante uma iluminação a partir de algoritmos de CNN que atuam em 3 dimensões (Kaler et al., 2023; Thakur et al. 2024).

Neste contexto, o uso de CNNs aplicadas a matrizes THSP pode ser uma alternativa viável e eficiente para classificação qualitativa dos padrões de BSL, podendo ser utilizadas como ferramenta para interpretação das matrizes, sem necessidade de processamento estatístico.

3.7. Computadores de placa única

Computadores de placa única (*Single Board Computer* – SBC) são pequenos computadores, dos quais todos os seus componentes principais estão integrados em um único sistema ou *chip* (Pajankar, 2017; Richardson e Wallace, 2014; Yau, 2014). Os SBCs se popularizaram com o lançamento do *Raspberry Pi*, em 2012 (Ali et al., 2013; Basford et al., 2020) sendo que atualmente existem diversos modelos disponíveis comercialmente, como o *Raspberry Pi*, *Onion Mega*, *Banana Pi*, *Asus Tinker Board*, *Odroid*, *NVIDIA Jetson nano* e outros (Qureshi, Mehboob e Aamir, 2021; Linder et al., 2020).

Os SBCs diferenciam-se em relação as placas tradicionais quanto aos seus preços e capacidades de hardware. Enquanto a *NVIDIA Jetson nano* é equipada com uma placa gráfica com núcleo CUDA, que favorece no processamento de imagens e redes neurais, o *Raspberry Pi* é 3 vezes mais barato, apresentando menor capacidade computacional (Linder et al., 2020).

Os SBCs são capazes de rodar uma variedade de sistemas operacionais com interface visual, como Linux, Android e Windows (Raspberry Pi Foundation, 2021; Hardkernel Co. Ltd., 2021). SBCs apresentam consumo energético mais baixo se comparados aos computadores tradicionais, e podem tanto trabalhar de maneira independente quanto arranjados em conjunto (*clusters*) (Grigoriev e Martinov, 2016). Erlina e Fikri (2023) utilizaram 16 *Raspberrys* ativos ao mesmo tempo, consumindo apenas 80W.

Atualmente, o uso de redes neurais, principalmente as CNNs, tem ganhado força nos SBCs. Exemplos de aplicações podem ser observados na detecção de doenças nas folhas do tomateiro (Gonzalez-Huitron et al., 2021), na classificação de frutos diversos (Pande et al., 2019) e na classificação de folhas de melão (Rahmat, Wahjuni e Rahmawan, 2022).

As SBCs podem processar CNNs já existentes, modificadas ou customizadas. Assim, evidencia-se o potencial das SBCs no processamento de imagens e de redes neurais profundas, principalmente na agricultura, apresentando resultados satisfatórios, com baixo custo e consumo energético.

4. MATERIAIS E MÉTODOS

As CNNs foram utilizadas com o intuito de classificar produtos agrícolas, conforme a variedade, nível de maturação ou presença de injúrias, tanto em ambiente controlado, quanto em campo. Assim, foram realizados seis experimentos. O Experimento 1 e Experimento 2 tiveram como foco testar a câmara seletora desenvolvida em relação as etapas de aquisição de imagens e processamento dos dados a partir da aplicação das CNNs (classificação de tomates quanto ao nível de maturação e classificação de laranjas quanto a presença de danos no epicarpo). Os experimentos posteriores foram realizados visando avaliar as potencialidades da CNN customizada para classificar produtos agrícolas quanto a atributos de qualidade em imagens obtidas em ambientes externos à câmara seletora.

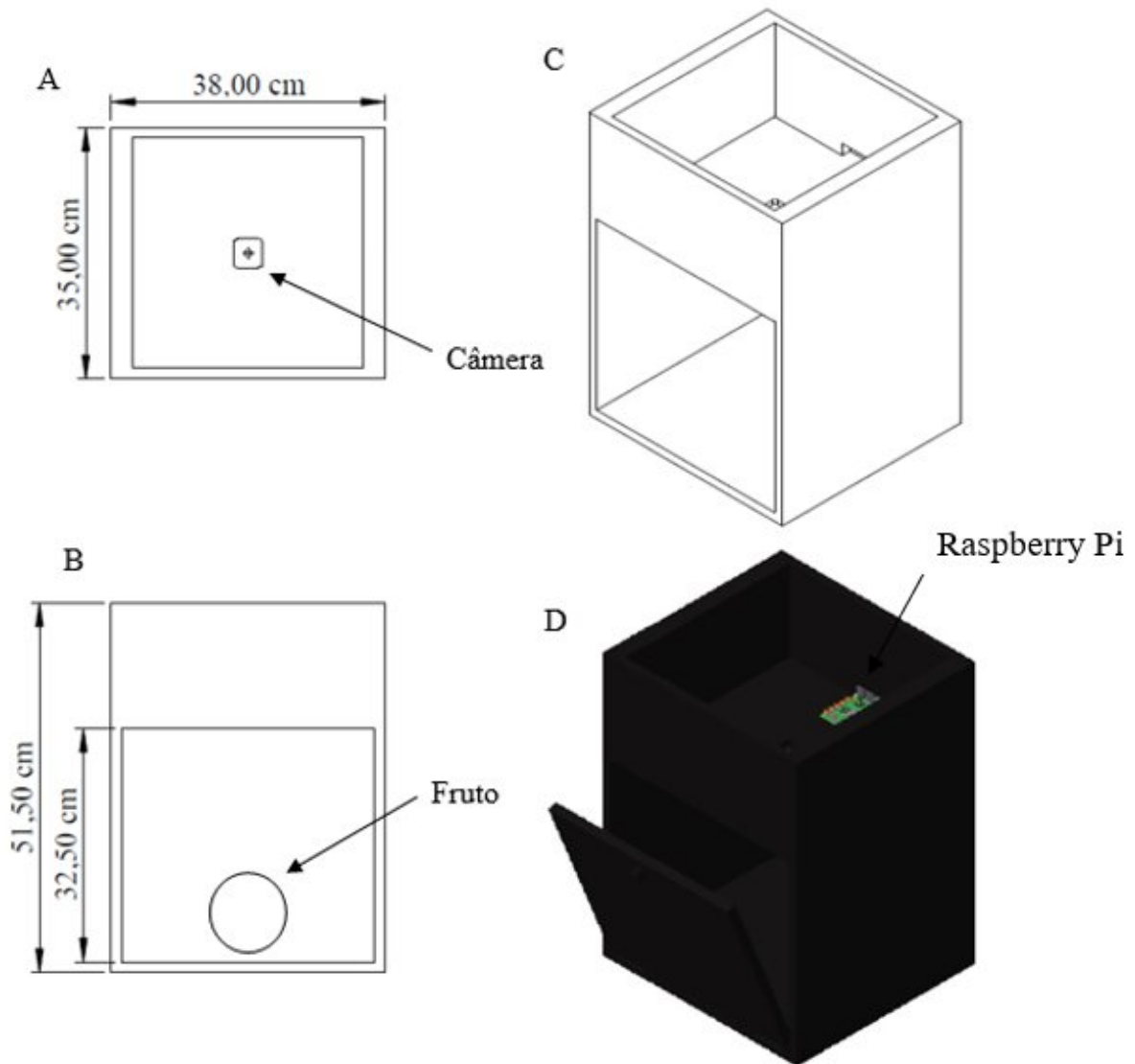
As classificações realizadas em todos os experimentos foram geradas a partir da aplicação das arquiteturas já existentes AlexNet e MobileNet. Além disso, foi desenvolvida

uma arquitetura de CNN customizada para este projeto, intitulada OakMoonNet, que será detalhada no item 4.2.

4.1. Desenvolvimento da câmara seletora

Para aquisição das imagens, processamento e classificação dos produtos agrícolas foi desenvolvido uma câmara seletora constituída de madeira e com porta de acesso, visando isolar a luz externa no momento da captura das imagens. Na parte superior, foi acoplado o *Raspberry Pi 5 8Gb* (Processador Broadcom BCM2712 Quad-core 2.4GHz 64-bit, 8 GB Ram LPDDR4X-4267 SDRAM), e um módulo de câmera para *Raspberry Pi Rev 1.3* (Sensor 5MP OV5647). Ainda na parte superior, foi fixado uma Display LCD *Touch Screen 7''* de resolução 1024 x 600, conectado ao *Raspberry Pi*. No interior foram fixadas fitas de LED 12V 6500k para iluminação. Durante a aquisição das imagens os frutos foram posicionados no interior da câmara a uma distância padronizada de 32,50 cm do módulo da câmera. A intensidade da iluminação no interior da câmara foi aferida pelo luxímetro marca Instrutherm, modelo LD-400, em escala de até 2000 Lux. As imagens foram capturadas por um software desenvolvido em Python 3.11 exclusivamente para este propósito, utilizando as bibliotecas Datetime, Libcamera, Time e Picamera2 (versão 0.3.22), sendo armazenadas em um Cartão SD SanDisk 64gb Classe A1. O croqui da câmara seletora se encontra na Figura 6.

Figura 6. Croqui da câmara seletora. Vista superior (A), vista frontal (B), vista isométrica (C) e vista isométrica sombreada com texturas (D).



Fonte: Elaborado pelo autor

4.2. Desenvolvimento do software *Classification Fruit UFRRJ*

Inicialmente, foi proposto o desenvolvimento de um software para treinamento das CNNs, conforme características específicas, como o número de épocas, a divisão e organização do banco de dados em treinamento e teste e a realização do *data augmentation*. O software deveria funcionar prioritariamente no Raspberry Pi 5, mas ser facilmente portátil para outras plataformas e sistemas operacionais, como o Linux ou Windows.

Para organizar os bancos de imagens e separar os dados em treinamento e teste, realizar pré-processamento e aplicar as CNNs, foi desenvolvido um algoritmo em Python 3.11, auxiliado pelas bibliotecas Custom tkinter, Itertools, Keras, Matplotlib, Numpy, OpenCV2, Pickle, Random, Shutil, Sklearn, Tensorflow, time e Visualkeras (Tabela 3).

Tabela 3. Bibliotecas utilizadas no software desenvolvido.

Biblioteca	Versão	Função
Custom Tkinter	2.5	Interface gráfica personalizada baseada no Tkinter.
Itertools	1.0	Ferramentas para criação de iteradores.
Keras	2.15.0	API de alto nível para redes neurais, executa sobre TensorFlow.
Matplotlib	3.8.2	Biblioteca de gráficos e visualizações em 2D.
Numpy	1.25.2	Suporte a arrays multidimensionais e operações matemáticas.
OpenCV2	4.8.0.76	Processamento de imagens e visão computacional.
Pickle	1.0	Serialização e desserialização de objetos Python.
Random	1.0	Geração de números pseudoaleatórios.
Shutil	1.0	Operações com arquivos e diretórios.
Scikit-learn (SKLearn)	1.4.1.post1	Algoritmos de aprendizado de máquina e ferramentas de modelagem.
Tensorflow	2.15.0	Computação de fluxo de dados para aprendizado profundo.
Time	1.0	Medição de tempo e cronometragem.
Visual Keras	0.14.1	Visualização gráfica de modelos criados com Keras.

Com este algoritmo, as imagens do dataset foram compiladas em arquivos para treinamento e teste, e os parâmetros de desempenho final de cada modelo, sendo ou as CNNs já existentes (AlexNet e MobileNet) ou pela CNN customizada (OakMoonNet), criada para esta pesquisa, foram extraídas e salvas em um arquivo de texto, de formato .txt.

A OakMoonNet foi desenvolvida com o intuito de ser uma CNN de menor custo computacional, eficiente em dispositivos com baixo poder de processamento, sendo capaz de classificar imagens de diversas categorias, treinando em tempos muito mais curtos do que as

arquiteturas já existentes, mas mantendo alta taxa de eficiência. O desenvolvimento desta CNN se baseou no ajuste das camadas utilizadas, possibilitando um treinamento rápido e eficiente.

Para esta arquitetura foram definidas, a partir de avaliações preliminares, os parâmetros de funcionamento da CNN como método *pooling*, a função de ativação, a função de reconhecimento dos erros, o otimizador e a taxa de aprendizado. A função *Dropout* foi utilizada na regularização dos pesos, buscando melhor distribuição na rede neural, e a função *Batch Normalization* foi utilizada para acelerar o treinamento, obtendo maior eficiência.

No algoritmo, a biblioteca OpenCV2 foi responsável pelas manipulações feitas nos arquivos de imagem, possibilitando carregar, manipular o sistema de cor, redimensionar e cortar imagens. A biblioteca Pickle foi responsável pela organização do dataset em um único arquivo, de extensão. pkl. O framework principal utilizou o TensorFlow, apoiado pela biblioteca Keras, desenvolvido por engenheiros e pesquisadores da Google Brain Team, que atuam em pesquisas de redes neurais profundas (Tensorflow, 2024). A construção da interface gráfica do usuário foi realizada a partir da biblioteca Custom Tkinter, uma versão modificada da biblioteca Tkinter, que possibilita a criação de interfaces visuais mais visualmente agradáveis.

4.3. Avaliação de desempenho das CNNs

As métricas de desempenho das CNNs foram obtidas a partir da matriz de confusão, que apresenta quatro tipos de resultados: Os Verdadeiros Positivos, dos quais foram classificados corretamente ao grupo analisado, os Verdadeiros Negativos, que não foram classificados ao grupo analisado de maneira correta, os Falso positivos, classificados como pertencentes ao grupo analisado de maneira incorreta e os Falsos negativos, classificados como não pertencentes ao grupo analisado, mas de maneira incorreta. Além disso, também é obtido os valores de *Loss* e Acurácia para os dados de treinamento e teste durante todas as épocas.

As métricas oriundas da matriz de confusão utilizadas para avaliação do desempenho das CNNs foram a Acurácia, Precisão, *Recall* e *F1-Score*. A utilização dessas métricas é comum nos estudos com redes neurais.

A acurácia demonstra a eficiência do modelo, indicando a porcentagem dos acertos obtidos para os dados de teste, sendo representado pela Equação 1. Nela, é somado os Verdadeiros Positivos e Verdadeiros Negativos, sendo o resultado dividido pela soma de todos os resultados obtidos. A precisão indica a porcentagem de dados classificados corretamente por

determinada classe, em relação a todas as suas classificações, sendo representado pela Equação 2. Assim, a precisão é o resultado da divisão entre os Verdadeiros Positivos pela soma dos Verdadeiros Positivos e Falsos Positivos. A Sensibilidade (*Recall*), indica a porcentagem total de acertos da classe em relação ao total de dados presentes da mesma classe no banco de dados para testes, sendo representado pela Equação 3. Seu cálculo se dá pela divisão dos Verdadeiros Positivos pela Soma dos Verdadeiros Positivos e Falsos Negativos. A métrica *F1-Score* utiliza-se dos valores de precisão e recall obtidos, sendo representado pela Equação 4, contemplando a acurácia na avaliação de desempenho do modelo treinado.

$$A_c = \frac{v_p + v_n}{v_p + f_p + f_n + v_n} \quad \text{Equação (1)}$$

$$P_r = \frac{v_p}{v_p + f_p} \quad \text{Equação (2)}$$

$$R_e = \frac{v_p}{v_p + f_n} \quad \text{Equação (3)}$$

$$f1 - score = \frac{2 * (v_p * r_e)}{p_e + r_e} \quad \text{Equação (4)}$$

em que,

v_p – Valores verdadeiro positivos em relação a classe analisada

v_n – Valores verdadeiro negativos em relação a classe analisada

f_p – Valores falso positivos em relação a classe analisada

f_n – Valores falso negativos em relação a classe analisada

4.4. Descrição dos experimentos de teste e validação da câmara seletora

4.4.1. Primeiro experimento: Uso da câmara seletora para a classificação dos frutos de laranja em função das injúrias

Neste experimento, foram utilizados 144 frutos de laranja, sendo que 42 frutos apresentaram injúrias leves, 38 frutos apresentaram injúrias graves e 64 frutos apresentaram

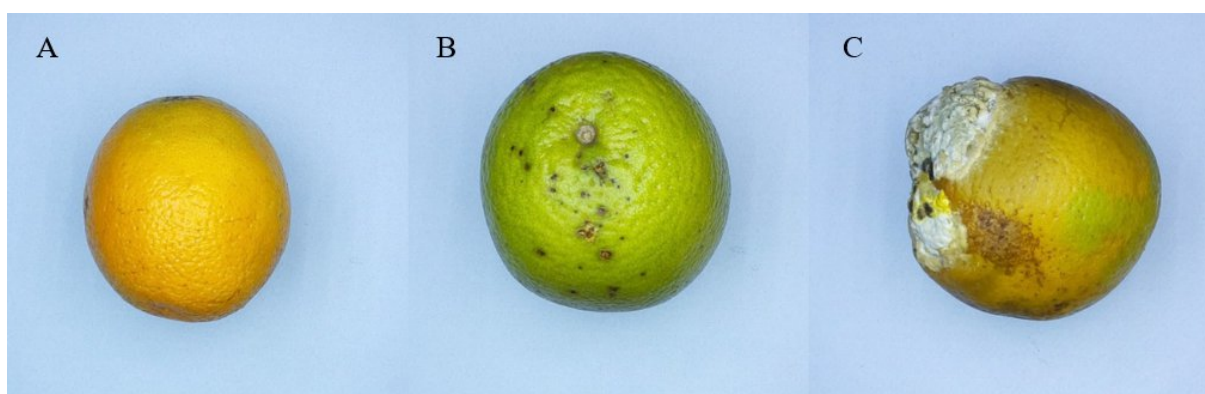
injúrias muito graves. As injúrias foram classificadas conforme a classificação definida pela CEAGESP (2011) quanto as características dos defeitos presentes no fruto (Tabela 4) Os frutos foram adquiridos no varejo, sendo da mesma cultivar (tipo Pêra) (Figura 7).

Tabela 4. Gravidade dos defeitos e características do fruto de laranja. Fonte: CEAGESP (2011).

Gravidade do defeito	Características
Leve	Dano mecânico leve
	Defeito de casca difuso leve
	Defeito de casca profundo leve
	Defeito de formato
Grave	Defeito fisiológico
	Defeito de casca difuso e profundo
	Lesão profunda
	Murcho
	Oleocelose
Muito Grave	Seco
	Podridão
	Imaturo
	Passado
	Bolor

Fonte: Elaborado pelo autor

Figura 7. Exemplo de imagens de frutos de laranja com danos Leves (A), Graves (B) e Muito Graves (C).



Fonte: Elaborado pelo autor

Uma vez que as injúrias estiveram localizadas em diferentes partes de um mesmo fruto, foram obtidas 4 imagens por fruto, totalizando 578 imagens. Por fim, as imagens foram divididas em 75% para treinamento (432 imagens) e 25% para testes (146 imagens).

O treinamento das CNNs OakMoonNet, AlexNet e MobileNet foi realizado, utilizando batch size 8 e *early stop* de 25 épocas. Assim foi avaliado a capacidade das CNNs em classificar os níveis de injúrias no epicarpo das laranjas.

4.4.2. Segundo experimento: Uso da câmara seletora para classificação dos frutos de tomate quanto a maturação

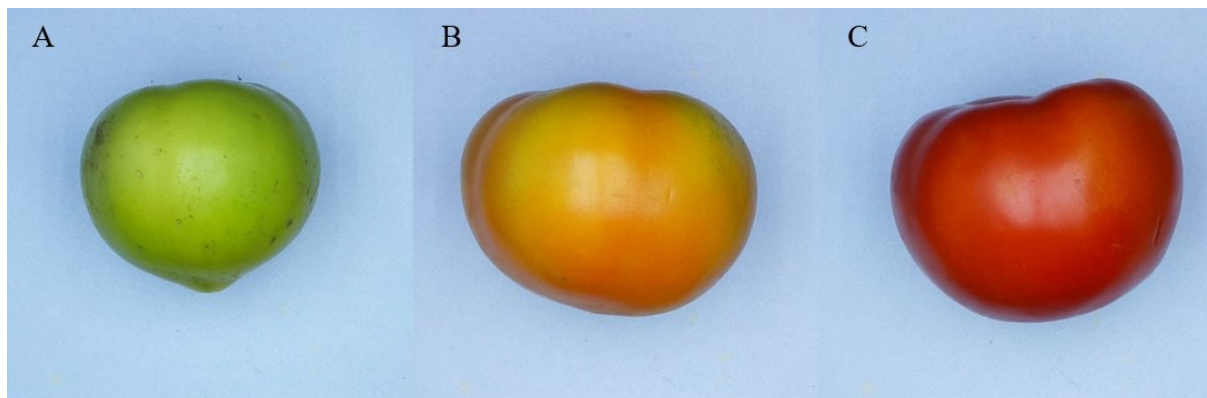
Neste experimento, foram utilizadas 600 imagens de frutos de tomate, sendo 200 de cada estágio maturação (Verde, Colorido e Maduro), conforme a classificação adaptada da CEAGESP (2003) (Tabela 5). Os frutos foram adquiridos no varejo, com tamanhos e cultivares diversos, visando aumentar a robustez do treinamento e testar a adaptabilidade da rede (Figura 8).

Tabela 5. Classificação da maturação do tomate quanto a cor. Fonte: CEAGESP (2003).

Classe	Característica
Tomate verde	Totalmente verde
Tomate colorido	Entre o amarelo e avermelhado
Tomate maduro	Com mais de 90% da coloração final

Fonte: Elaborado pelo autor

Figura 8. Exemplo de imagens de frutos de tomate Verde (A), Colorido (B) e Vermelho (C).



Fonte: Elaborado pelo autor

Os dados foram divididos em 75% para treinamento (450 imagens) e 25% para testes (150 imagens). Uma vez que o banco de dados obtidos já continha muitas imagens, não houve a necessidade da utilização do data augmentation.

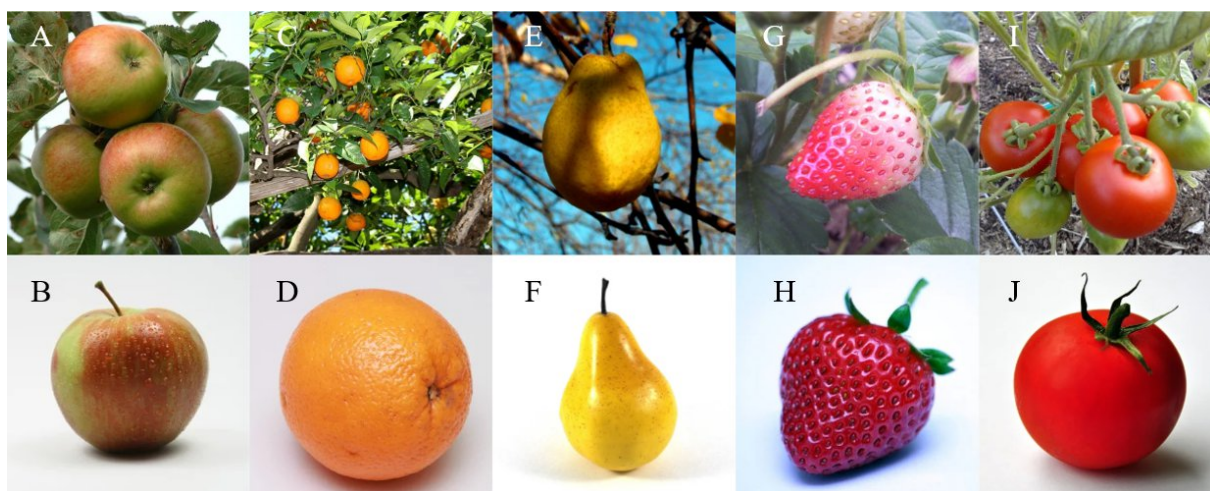
O treinamento das CNNs OakMoonNet, AlexNet e MobileNet foi realizado, utilizando *batch size* 8 e *early stop* de 25 épocas. Assim, foi avaliado a eficiência das CNNs quanto a classificação por meio de cores utilizando frutos da mesma espécie, mas de formatos diferentes.

4.5. Descrição dos experimentos de teste e validação do software para outras aplicações

4.5.1. Terceiro experimento: Classificação de tipos de frutos quanto utilizando imagens obtidas na internet

Para a realização deste experimento, foram obtidas 50 imagens dos frutos de maçã, laranja, pera, morango e tomate na internet (<https://storage.googleapis.com/openimages/web/index.html>), totalizando um banco de imagens de 250 imagens. As imagens apresentavam ao menos um fruto inteiro, em diversas condições de iluminações e ambientes, a fim de testar a robustez das redes neurais (Figura 9).

Figura 9. Exemplo de imagens dos frutos de Maçã (A e B), Laranja (C e D), Pêra (E e F), Morango (G e H) e Tomate (I e J) em ambientes externos e internos, utilizados para o treinamento das CNNs.



Fonte: Elaborado pelo autor

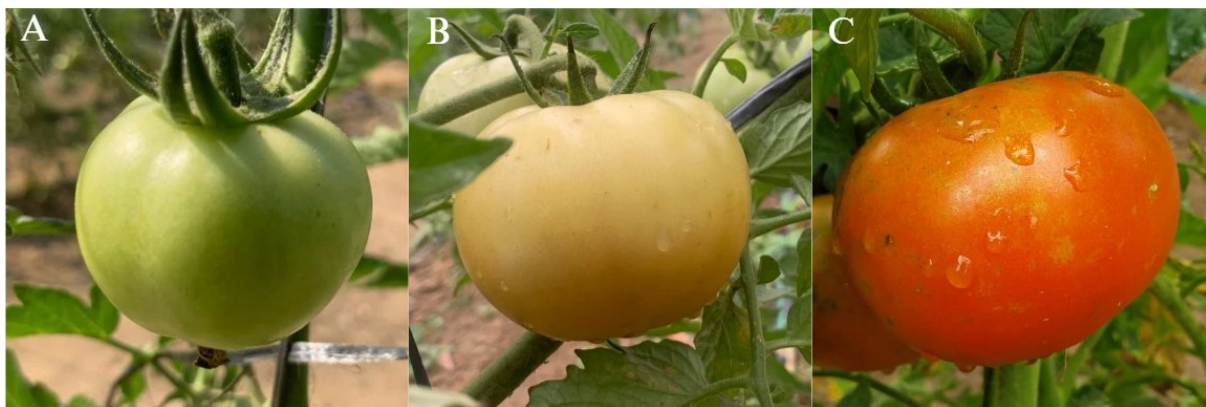
O objetivo deste experimento foi avaliar o desempenho das CNNs quanto a identificação de frutos diferentes em variados ambientes e iluminações, avaliando a performance das redes neurais convolucionais quanto a classificação em frutos de cores semelhantes (maçã, morango, pera e tomate) e formas parecidas (maçã, tomate e laranja), sejam eles na lavoura ou em ambiente controlado.

O banco de imagens foi submetido a aplicação da técnica do *data augmentation*, gerando 4 novas imagens por dado, o que possibilitou o aumento do para 1000 imagens (200 imagens de cada fruto). Os dados foram separados para treinamento e teste, na proporção de 90% (900 imagens) e 10% (100 imagens), respectivamente, sendo realizado o treinamento das três CNNs. Foi realizado o treinamento das CNNs OakMoonNet, AlexNet e MobileNet, utilizando *batch size* 8 e *early stop* de 10 épocas.

4.5.2. Quarto experimento: Classificação de frutos de tomates na lavoura quanto ao nível de maturação

Neste experimento, os frutos de tomate foram classificados ainda nas plantas quanto ao seu estágio de maturação. Assim, foram coletadas 130 imagens de frutos de tomate, sendo 45 classificados verdes, 35 classificados como intermediários e 50 classificados como maduros (Figura 10). A classificação foi realizada de forma visual seguindo as normas estabelecidas pela CEAGESP (2003). As imagens foram capturadas em campo, com diferentes iluminações, por meio de variados smartphones, visando testar o desempenho das CNNs quanto a variação de imagens e câmeras, e a eficiência em identificar um único ou múltiplos objetos.

Figura 10. Exemplo de imagens de frutos de tomate Verde (A), Amarelo (B) e Vermelho (C) em ambientes externos, utilizados para o treinamento das CNNs.



Fonte: Elaborado pelo autor

A técnica do *data augmentation* foi aplicada, gerando 221 imagens de tomates verdes, 210 imagens de tomates amarelos e 200 imagens de tomates vermelhos, totalizando 637 imagens. Com isso, 90% dos dados foram destinados a treinamento (567 imagens), e 10% para testes (64 imagens), totalizando 198, 189 e 180 imagens para treinamento e 21, 22 e 23 imagens para testes para os frutos de tomate verde, amarelo e vermelho, respectivamente. O treinamento das CNNs OakMoonNet, AlexNet e MobileNet foi realizado, utilizando *batch size* 8 e *early stop* de 10 épocas.

4.5.3. Quinto experimento: Classificação de frutos de tomate quanto a presença de injúrias no epicarpo

Este experimento classificou frutos de tomate quanto a presença de injúrias no epicarpo, sendo eles considerados Perfeitos (sem a presença de injúrias) e Imperfeitos (com presença de injúrias). O banco de imagens foi formado por 138 Frutos Perfeitos e 275 Frutos imperfeitos, obtidos em diferentes épocas de colheita mecanizada (Figura 11).

Figura 11. Exemplo de imagens de frutos de tomate perfeitos e imperfeitos para a Coleta 1 (A e B), Coleta 2 (C e D), Coleta 3 (E e F) e Coleta 4 (G e H), utilizados para o treinamento das CNNs.



Fonte: Elaborado pelo autor

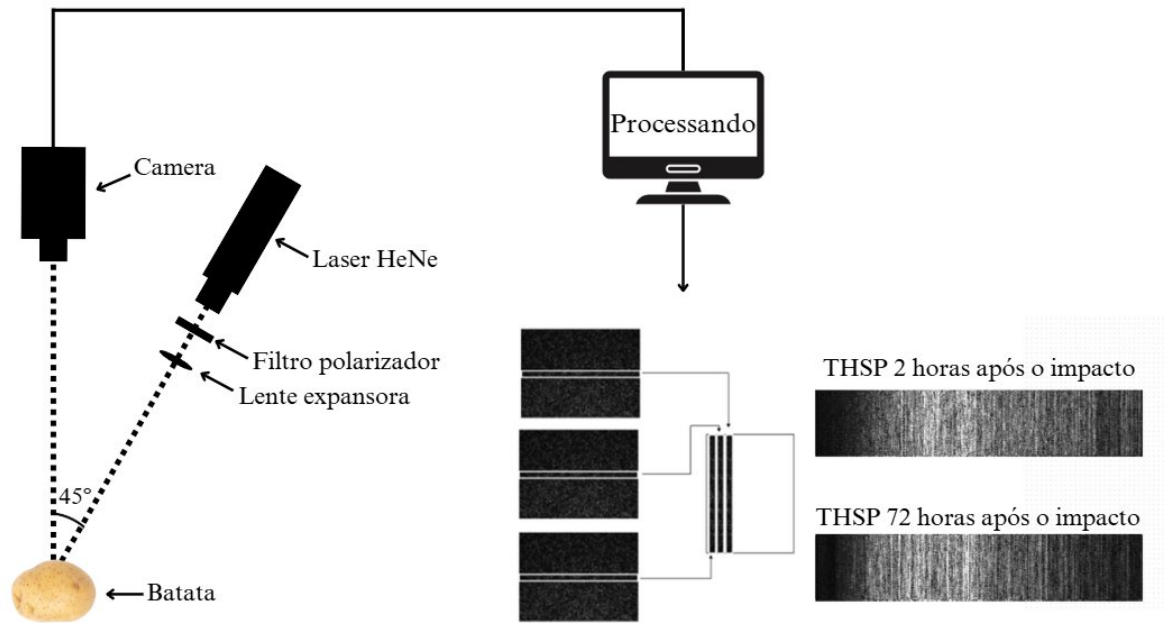
Para aproximar o número de imagens das duas classes, foi aplicada a técnica do *data augmentation* para as imagens de frutos perfeitos, gerando 2 imagens por dados, resultando em 276 imagens. Com isso, os dados foram divididos em 90% para treinamento e 10% para testes, totalizando 246 frutos perfeitos e 247 frutos imperfeitos para treinamento e 27 frutos perfeitos e 28 frutos imperfeitos para testes.

O treinamento das CNNs OakMoonNet, AlexNet e MobileNet foi realizado, utilizando *batch size* 8 e *early stop* de 25 épocas. Com isso, foi possível testar as CNNs quanto a ausência ou presença de injúrias.

4.5.4. Sexto experimento: Classificação de imagens *Time History Spatial Pattern* (THSP) para monitoramento de injúrias internas em batatas

Neste experimento, foram utilizadas imagens de batatas com presença de injúrias internas, conhecidas como internal bruising. As imagens foram obtidas por meio da técnica do *Biospeckle Laser* (BSL), sendo estas utilizadas para a formação das matrizes *Time History Speckle Pattern* (THSP). Os THSP foram formados a partir da extração da faixa horizontal central de pixels (linha central da imagem) de 128 imagens sucessivas do padrão do BSL correspondente a iluminação de cada batata, reposicionando-as em colunas lado a lado (Figura 12). Desta forma, em uma dimensão do THSP registrou-se o conjunto de pixels correspondente à distribuição espacial do padrão do BSL e na outra a evolução temporal destes pixels em instantes sucessivos.

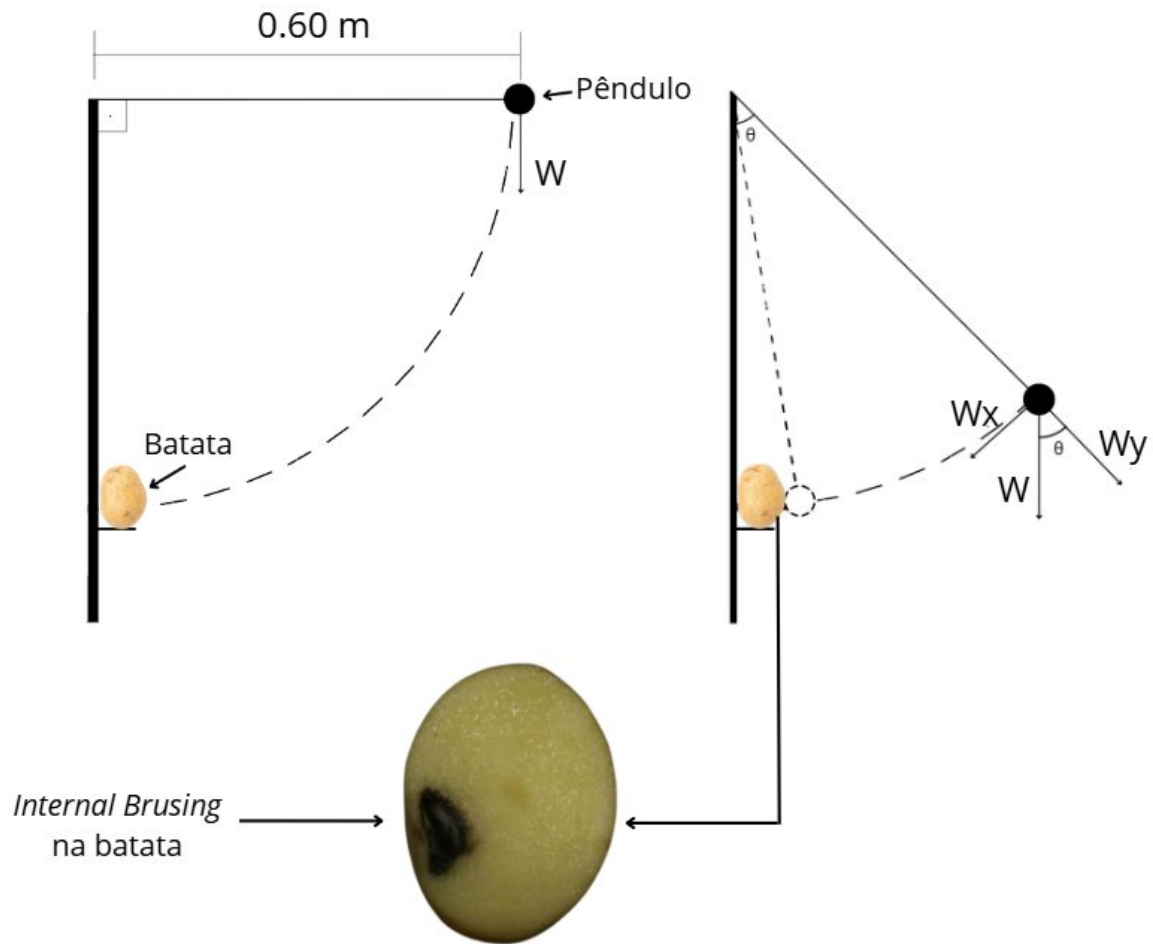
Figura 12. (A) Construção do padrão *Time History Speckle Pattern* (THSP), (B) padrão THSP obtido em uma batata na região do impacto mecânico 2 horas após o impacto mecânico e (C) padrão THSP obtido em uma batata na região do impacto mecânico 72 horas após o impacto mecânico.



Fonte: Elaborado pelo autor

O banco de imagens do BSL foi gerado por Costa (2024) e disponibilizado na plataforma Zenodo (<https://zenodo.org/records/11094575>). As imagens foram obtidas a partir da aplicação da técnica do *biospeckle laser* em 20 batatas atingidas por um pêndulo em queda livre da sua posição inicial (90° em relação a amostra) até o contato com a batata (Figura 13). As imagens do *biospeckle laser* (BSL) foram obtidas em diferentes instantes buscando acompanhar o desenvolvimento do internal bruising nos tubérculos.

Figura 13. Disposição experimental para realização do impacto mecânico na batata capaz de gerar o *internal bruising*.



Fonte: Elaborado pelo autor

As matrizes THSP foram divididas nas classes Horas após o impacto mecânico (HAIM) e Dias após o impacto mecânico (DAIM), conforme apresentado na Tabela 6. As CNNs OakMoonNet, AlexNet e MobileNet foram treinadas para realizar a classificação dos THSP associados em função do tempo de desenvolvimento do internal bruising.

Tabela 6. Divisão dos THSP obtidos durante a evolução do internal bruising em batatas nas classes Horas após o impacto mecânico (HAIM) e Dias após o impacto mecânico (DAIM).

Classe	Instante de obtenção do biospeckle	Número de THSP
Horas após o impacto mecânico (HAIM)	2 horas, 4 horas e 6 horas após a ocorrência do impacto mecânico	60
Dias após o impacto mecânico (DAIM)	24 horas, 48 horas e 72 horas após a ocorrência do impacto mecânico	60

Fonte: Elaborado pelo autor

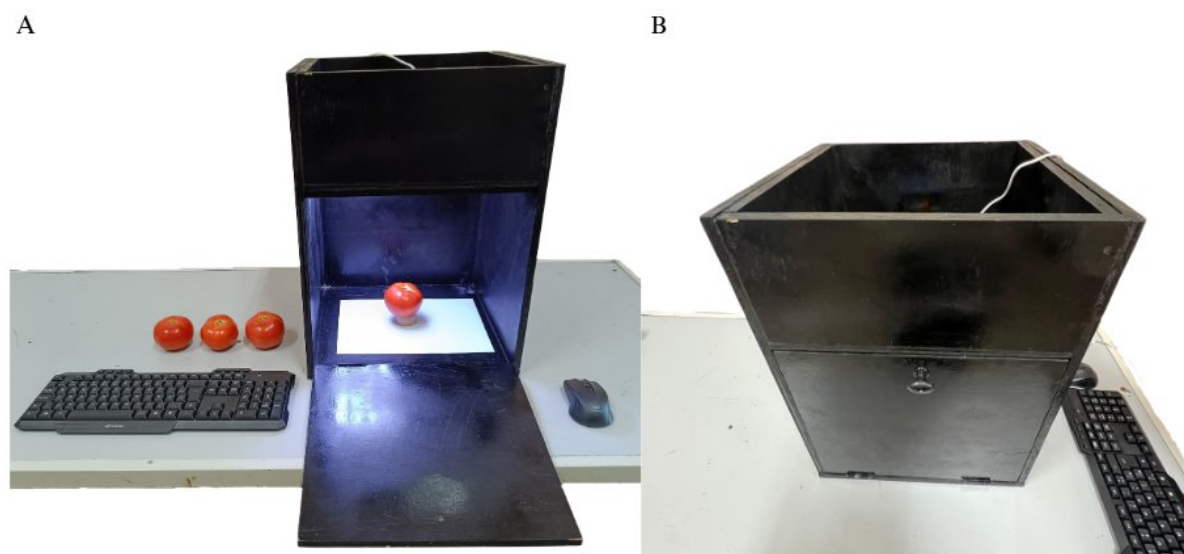
Os dados de THSP foram divididos em 66,66% para treinamento (80 imagens) e 33,33% para testes (40 imagens). O treinamento das CNNs OakMoonNet, AlexNet e MobileNet foi realizado, utilizando *batch size* 8, em 1000 épocas com *early stop* de 100 épocas.

5. RESULTADOS E DISCUSSAO

5.1. Usabilidade da Câmara seletora

A Figura 14. A câmara seletora em vista frontal, com sua tampa frontal aberta (A) e fechada (B). apresenta a câmara seletora, com sua tampa frontal aberta, LEDs ligados promovendo iluminação uniforme de 872 Lux, aferido pelo luxímetro, e com menos sombra, um fruto de tomate posicionado no meio e um fundo branco para facilitar a diferenciação do objeto de interesse na imagem capturada. A Figura 14B apresenta a câmara seletora com sua tampa frontal fechada, impedindo interferência de iluminação externa, criando um ambiente isolado.

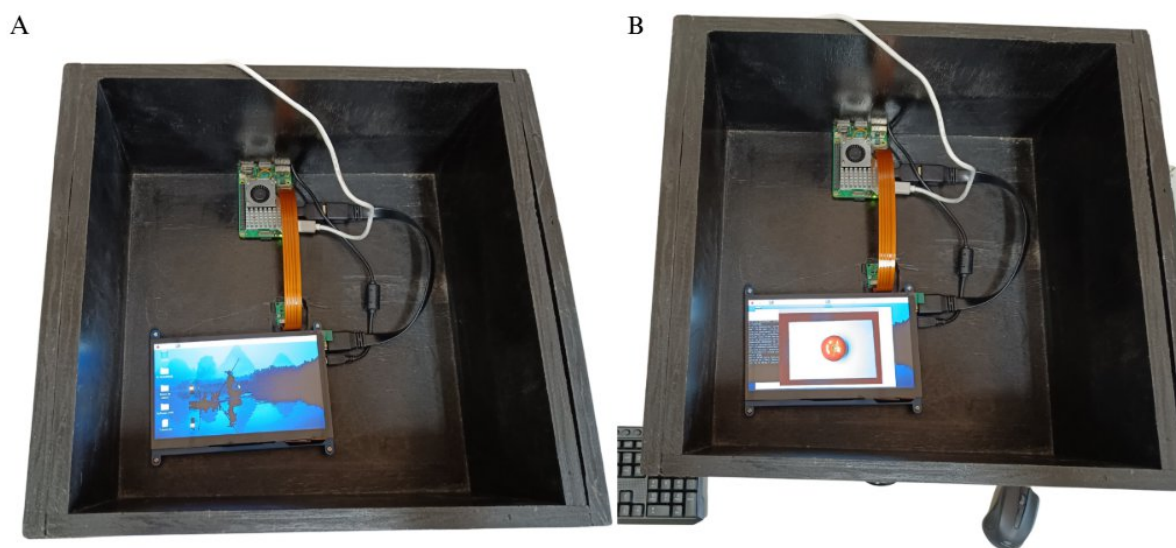
Figura 14. A câmara seletora em vista frontal, com sua tampa frontal aberta (A) e fechada (B).



Fonte: Elaborado pelo autor

No topo Da câmara seletora (Figura 15A), encontra-se a câmera, o *Raspberry Pi 5* e a Tela *Touch Screen*, do qual seu posicionamento foi feito para melhorar sua ergonomia e usabilidade do usuário. A Figura 15B mostra a tela com o software de captura em operação, exibindo a pré-visualização do fruto de tomate.

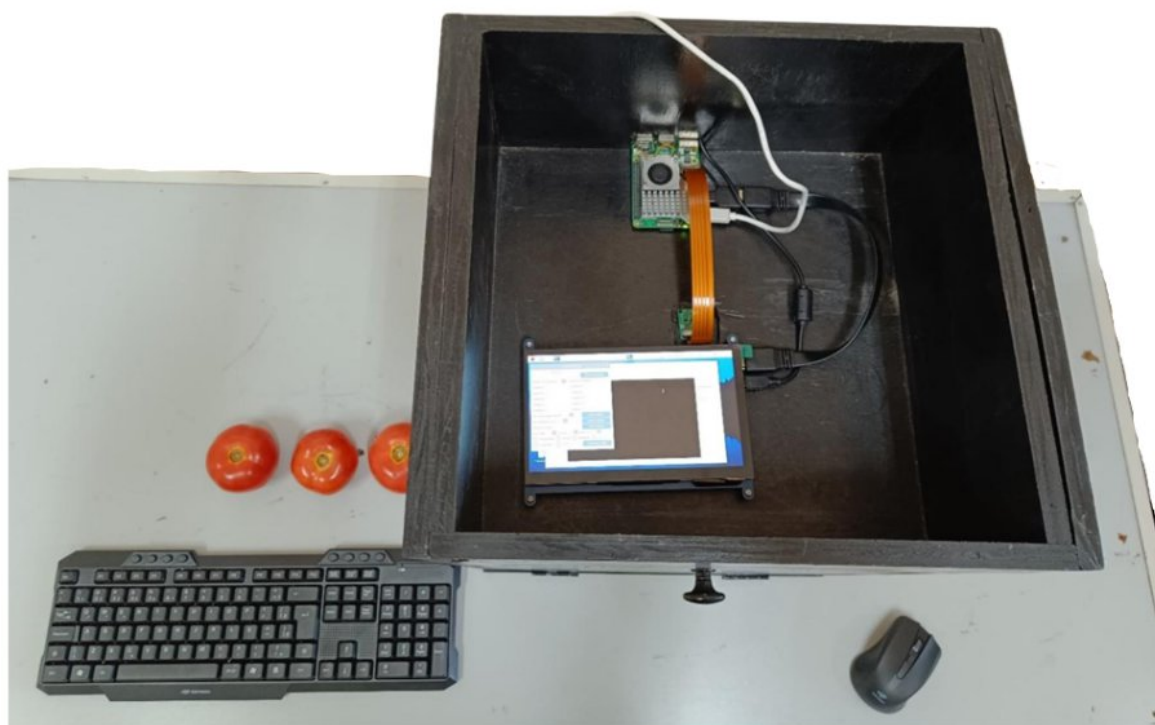
Figura 15. Topo da câmara seletora com seus componentes (A) e durante a captura de imagens (B).



Fonte: Elaborado pelo autor

Após a captura, o usuário deve organizar manualmente os dados obtidos da forma que desejar. Para treinar a CNN, o usuário deve iniciar o software desenvolvido para este propósito. A Figura 16 apresenta o topo da câmara seletora, com o software de treinamento das CNNs em operação. A extração dos dados gerados pode ser feita via dispositivos externos, como pen-drive ou HD externo, ou por meio da internet, utilizando sistemas de armazenamento em nuvem como Google Drive ou Microsoft OneDrive.

Figura 16. Topo da câmara seletora com o software para treinamento das CNNs em operação.



Fonte: Elaborado pelo autor

5.2. Desenvolvimento da CNN Customizada

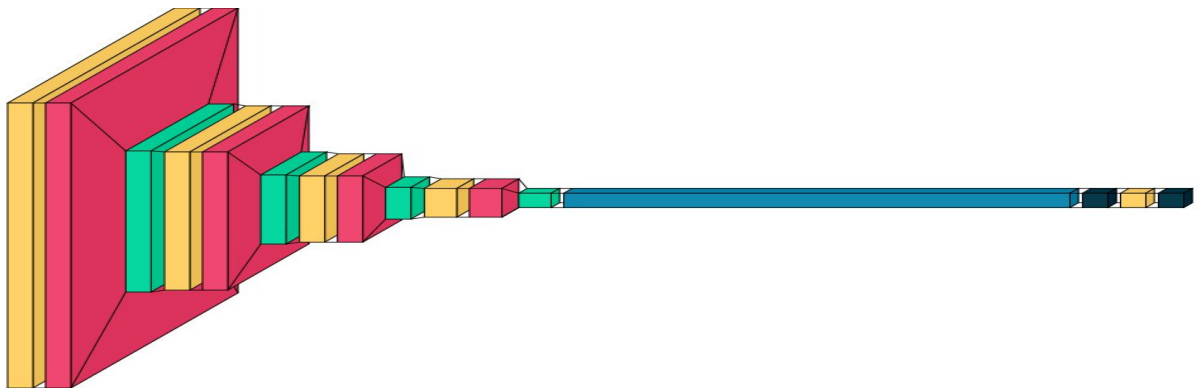
Na OakMoonNet, as imagens de entrada foram definidas na dimensão 100x100 após experimentos preliminares. Com isso, os dados de entrada são mais de 4 vezes menores em comparação com CNNs como AlexNet e MobileNet. Chakraborty et al. (2023), ao criar a CNN customizada SortNet, fez com que a imagem de entrada fosse na proporção 125x125. O tempo de treinamento utilizando um banco de dados de 1280 imagens foi de 379,2 segundos, obtendo uma acurácia de 97,60%, sendo mais rápida que a AlexNet e GoogleNet, que treinaram em 612 e 1342,8 segundos e obtiveram acurácia de 91,40% e 98,40%, respectivamente.

A OakMoonNet apresentou, em sequência, o conjunto de Camada *Convolutional*, *Batch Normalization* e *Max Pooling* quatro vezes, com a quantidade de filtros da camada convolucional variando de 32, 64, 128 e 256 respectivamente. Após, uma camada *Flatten*, seguidas de uma camada densa com 64 unidades, uma camada *dropout* e uma camada densa com unidades variáveis, conforme o número de classes do dataset utilizado. As camadas de *Batch Normalization* são o diferencial desta CNN em comparação com muitas já consagradas. Kumar et al. (2021), mesmo utilizando o tamanho tradicional para entrada de imagens em sua

CNN customizada (224x224), utilizou camadas de *Batch Normalization*, o que acelerou seu treinamento de 3 a 8 vezes, em comparação com as CNNs já existentes (Inception-Net, Mobile-Net, Mobile-Net-v2, ResNet-50-v2 e VGG-19).

O otimizador utilizado nas camadas convolucionais é a função *ReLU*, Para a última camada densa, utiliza-se o otimizador *Softmax*. A função de *Loss* utiliza a *Sparse Categorical Crossentropy* e o otimizador utilizado é o ADAM. A Figura 17 representa a arquitetura da rede neural desenvolvida, enquanto a Tabela 7 detalha as camadas da OakMoonNet. A função de ativação *ReLU* é utilizada na maioria das arquiteturas CNN, assim como Kumar et al. (2021) e Chakraborty et al. (2023) utilizaram na construção de suas arquiteturas customizadas.

Figura 17. Estrutura da OakMoonNet



Fonte: Elaborado pelo autor

Tabela 7. Arquitetura da OakMoonNet

Camadas	Filtros	Tamanho
Imagem	1	100x100x3
<i>Convolutional</i>	32	99x99x32
<i>Batch Normalization</i>	32	99x99x32
<i>Max Pooling</i>	32	49x49x32
<i>Convolutional</i>	64	48x48x64
<i>Batch Normalization</i>	64	48x48x64
<i>Max Pooling</i>	64	24x24x64
<i>Convolutional</i>	128	24x24x128
<i>Batch Normalization</i>	128	24x24x128
<i>Max Pooling</i>	128	11x11x128
<i>Convolutional</i>	256	10x10x256
<i>Batch Normalization</i>	256	10x10x256
<i>Max Pooling</i>	256	5x5x256
<i>Flatten</i>	-	6400
<i>Fully connected</i>	-	64
<i>Dropout</i>	-	64
<i>Fully connected</i>	-	CLASSES

Fonte: Elaborado pelo autor

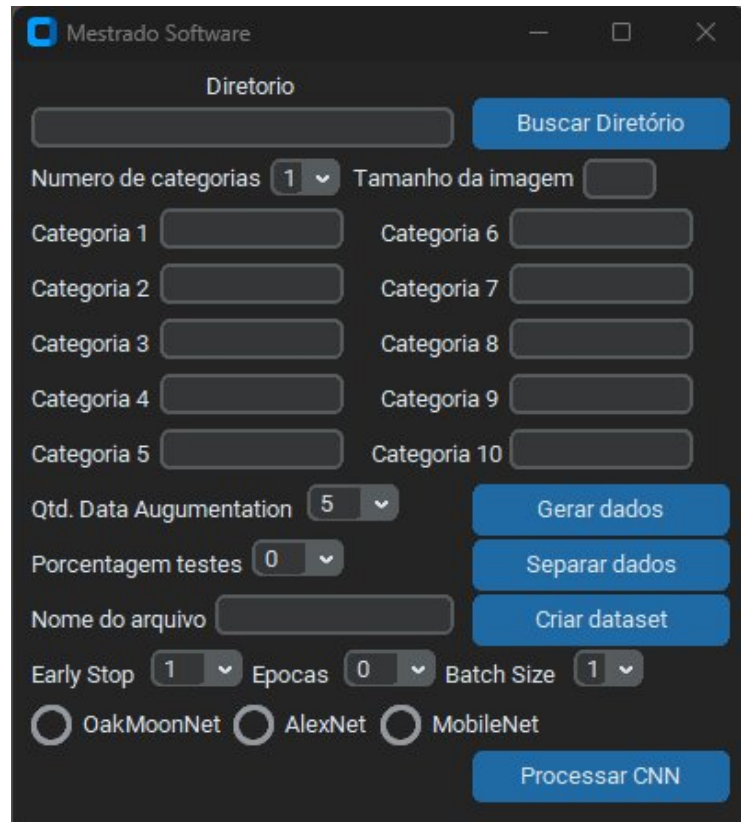
5.3. Desenvolvimento do software *Classification Fruit UFRRJ*

O Software desenvolvido, intitulado *Classification Fruit UFRRJ* (Número de registro no INPI: BR512025002316-2) foi desenvolvido e utilizado diretamente na câmara seletora, pelo *Raspberry Pi 5*. Contudo, o software pode ser facilmente portado para outras plataformas, como computadores de mesa, operando em diferentes sistemas operacionais (Linux ou Windows). Todos os experimentos foram rodados no *Raspberry Pi 5*, seja com imagens capturadas pela própria câmara seletora quanto com imagens obtidas de outras fontes.

O algoritmo desenvolvido possui uma interface visual (Figura 18), visando facilitar a utilização do usuário das funcionalidades disponibilizadas. Nele, é possível organizar um dataset de imagens com até dez diferentes categorias, realizar o *data augmentation* do banco de dados, separar os dados de treinamento e teste conforme porcentagem específica, compilar

o dataset e escolher a rede neural convolucional que será treinada (OakMoonNet, AlexNet e MobileNet), especificando o diretório do *dataset*, o número de categorias, as épocas, *batch size*, e tolerância do mecanismo de *early stop*.

Figura 18. Interface de usuário do algoritmo desenvolvido.



The image shows a software window titled "Mestrado Software" with a dark theme. It contains various input fields and buttons for configuring a CNN training process. At the top, there is a "Diretorio" label above a text input field and a "Buscar Diretório" button. Below this, "Numero de categorias" is set to 1 with a dropdown arrow, and "Tamanho da imagem" is an empty text field. There are ten input fields for "Categoria 1" through "Categoria 10". Further down, "Qtd. Data Augumentation" is set to 5 with a dropdown arrow, and "Porcentagem testes" is set to 0 with a dropdown arrow. A "Nome do arquivo" text field is followed by a "Criar dataset" button. Below that, "Early Stop" is set to 1 with a dropdown arrow, "Epocas" is set to 0 with a dropdown arrow, and "Batch Size" is set to 1 with a dropdown arrow. Three radio buttons are present for "OakMoonNet", "AlexNet", and "MobileNet", all of which are currently unselected. A large "Processar CNN" button is at the bottom right. Other buttons include "Gerar dados" and "Separar dados" located between the "Qtd. Data Augumentation" and "Porcentagem testes" settings.

Fonte: Elaborado pelo autor

Os dados para processamento devem ser separados em pastas, conforme a categoria correspondente. Essas pastas devem ser armazenadas em uma pasta principal, e o diretório, ou caminho de armazenamento dos dados, é obtido pelo usuário ao clicar no botão “Buscar Diretório”. Com isso, o usuário especifica a porcentagem de divisão dos dados, e o software aloca-os em pastas de treinamento (*train*) e teste (*test*), específicos de cada categoria. O Número de categorias representa quantas categorias do dataset serão trabalhadas, sendo elas descritas nos seus campos correspondentes (Categoria 1, Categoria 2 e Categoria N). Esta opção ainda define quantas saídas de classificação existirão no final das CNNs utilizadas. Caso haja a necessidade de geração de dados por meio do data augmentation, o usuário definirá a quantidades de imagens geradas a partir de um único arquivo, dos quais serão armazenados em uma nova pasta, cujo seu nome será a categoria, acompanhada do sufixo “DA”.

O parâmetro “Tamanho da imagem” define a dimensão, em 1:1, do reprocessamento do dataset para seu uso nas CNNs. Para a OakMoonNet, AlexNet e MobileNet, foram utilizadas as dimensões 100x100, 227x227 e 224x224, respectivamente. O parâmetro “Nome do arquivo” possibilita que os arquivos gerados no processamento recebam a nomenclatura escolhida pelo usuário. O botão “Criar dataset” inicia o processamento dos arquivos de imagem, conforme as especificações definidas anteriormente. Ao final, são obtidos dois arquivos na extensão .pkl, do qual compacta todo o dataset em treinamento e teste, tornando-os processáveis pelas CNNs.

O parâmetro “quantidade de épocas” foi implementado para definir quantas vezes, consecutivamente, será realizado o treinamento de uma rede neural. Este processo pode ser interrompido pelo parâmetro “Tolerância do Early Stop”, do qual avalia o treinamento, encerrando-o caso após um número determinado de épocas a acurácia para os dados de teste não superar uma máxima acurácia de teste previamente atingida e salvando o melhor modelo obtido. O parâmetro “*Batch size*” foi utilizado para definir quantos dados serão processados simultaneamente durante a etapa de treinamento.

O botão “Processar CNN” realizou o treinamento de alguma das CNNs selecionadas abaixo (OakMoonNet, AlexNet ou MobileNet). Ao final do treinamento, foi gerado um arquivo correspondente ao modelo treinado, um arquivo em .txt contendo os dados de *Loss* e Acurácia para os dados de treinamento e teste, além da matriz de confusão e a acurácia, precisão, sensibilidade e *F1-Score* para os dados de teste, o tempo de treinamento da CNN e a arquitetura do modelo utilizado em .png.

A Figura 19 mostra a interface preenchida para processar os dados de 5 categorias distintas (Maçã, Laranja, Pêra, Morango e Tomate) para a dimensão 100x100, gerando 5 imagens por dado pelo data augmentation, utilizando 10% dos dados para testes, 100 épocas, *batch size* 8 e tolerância de *early stop* 10, realizando o treinamento da OakMoonNet. A Figura 20 apresenta o fluxograma de funcionamento do software desenvolvido.

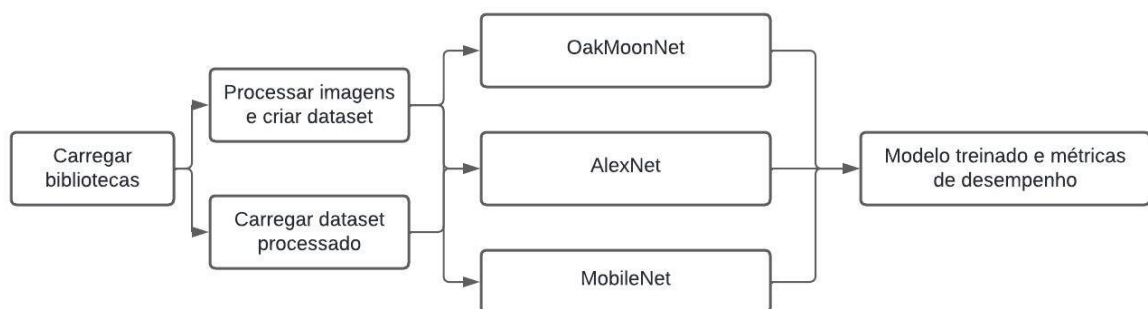
Figura 19. Interface de usuário do algoritmo desenvolvido preenchida.

The screenshot shows the 'Mestrado Software' window with the following settings:

- Diretorio:** D:/OneDrive/UFRRJ/Projeto Mestrado/
- Numero de categorias:** 5
- Tamanho da imagem:** 100
- Categorias:**
 - Categoria 1: Apple
 - Categoria 2: Orange
 - Categoria 3: Pear
 - Categoria 4: Strawberry
 - Categoria 5: Tomato
 - Categoria 6: (empty)
 - Categoria 7: (empty)
 - Categoria 8: (empty)
 - Categoria 9: (empty)
 - Categoria 10: (empty)
- Qtd. Data Augumentation:** 5
- Porcentagem testes:** 10
- Nome do arquivo:** Internet_5_100
- Early Stop:** 10
- Epocas:** 100
- Batch Size:** 8
- Model Selection:** OakMoonNet (selected), AlexNet, MobileNet
- Buttons:** Buscar Diretório, Gerar dados, Separar dados, Criar dataset, Processar CNN

Fonte: Elaborado pelo autor

Figura 20. Fluxograma de funcionamento do software desenvolvido.



Fonte: Elaborado pelo autor

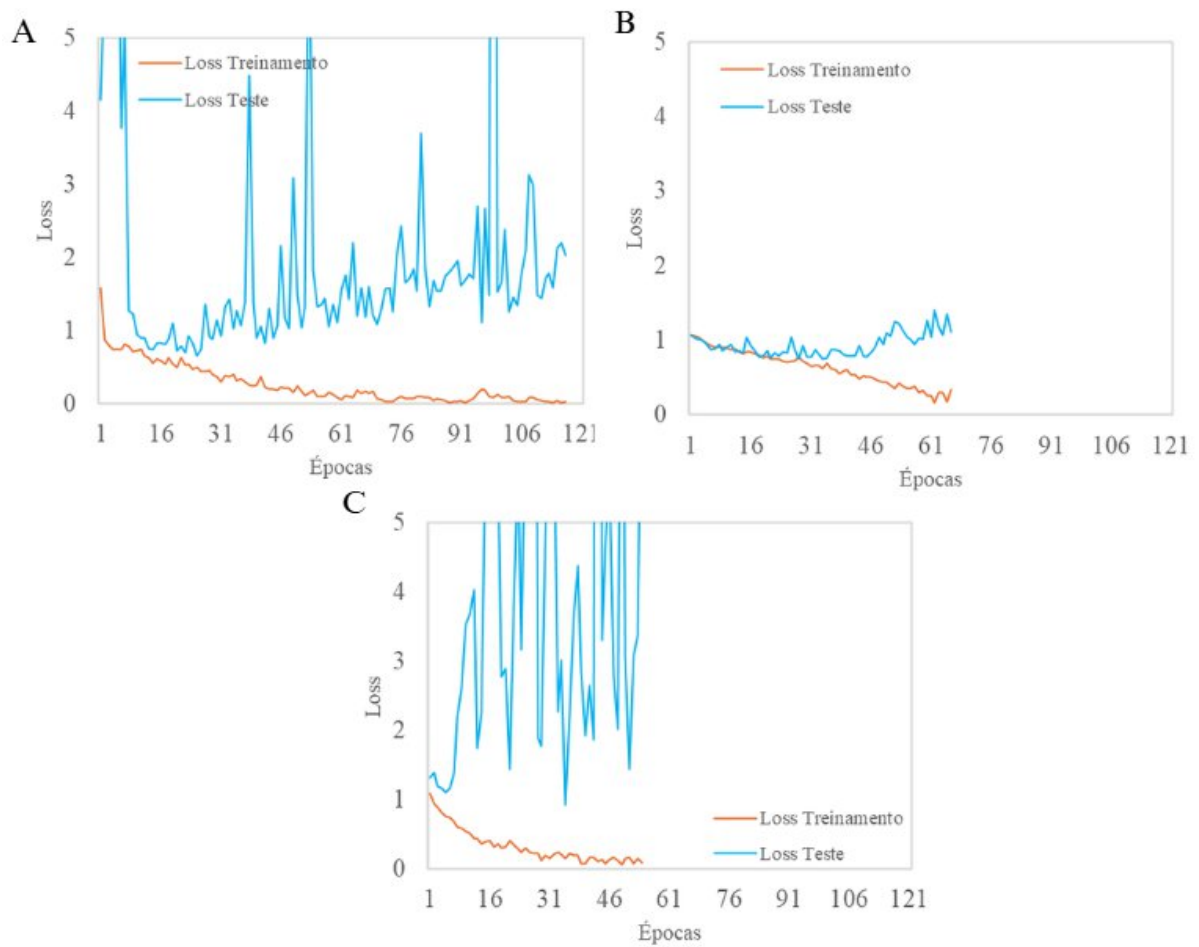
5.4. Experimentos de teste e validação da câmara seletora

5.4.1. Primeiro experimento – Uso da câmara seletora para a classificação dos frutos de laranja em função das injúrias

Durante a avaliação dos valores de *Loss* obtidos durante um treinamento, valores mais baixos (menores que 1,00) são indicativos de menos discrepância, ou seja, melhor capacidade de generalização. Picos de *Loss* representam épocas que ocorreram discrepância entre os valores previstos e os valores reais. No entanto, esta é uma análise a partir do comportamento geral ao longo das épocas. Para os casos abordados neste trabalho, uma vez que se opte por seleccionar a CNN da melhor época (maior acurácia para os dados de teste), esta questão se torna menos relevante.

Ao analisar os valores de *Loss* (Figura 21), a AlexNet apresentou estabilidade para os valores obtidos para os dados de teste, enquanto a OakMoonNet e a MobileNet apresentaram comportamento oscilatório. Além disso, para os três casos, os valores de *Loss* para os dados de treinamento apresentaram tendência crescente, enquanto para os dados de teste, apresentaram tendências decrescente. Ao analisar o *Loss* na época de melhor desempenho em cada treinamento, verificou-se os valores de 0,0170 (treinamento) e 1,6851 (teste) para a OakMoonNet (época 92), 0,5303 (treinamento) e 0,7831 (teste) para a AlexNet (época 41) e 0,1241 (treinamento) e 1,7774 (teste) para a MobileNet (época 29).

Figura 21. Respostas de *Loss* para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste.

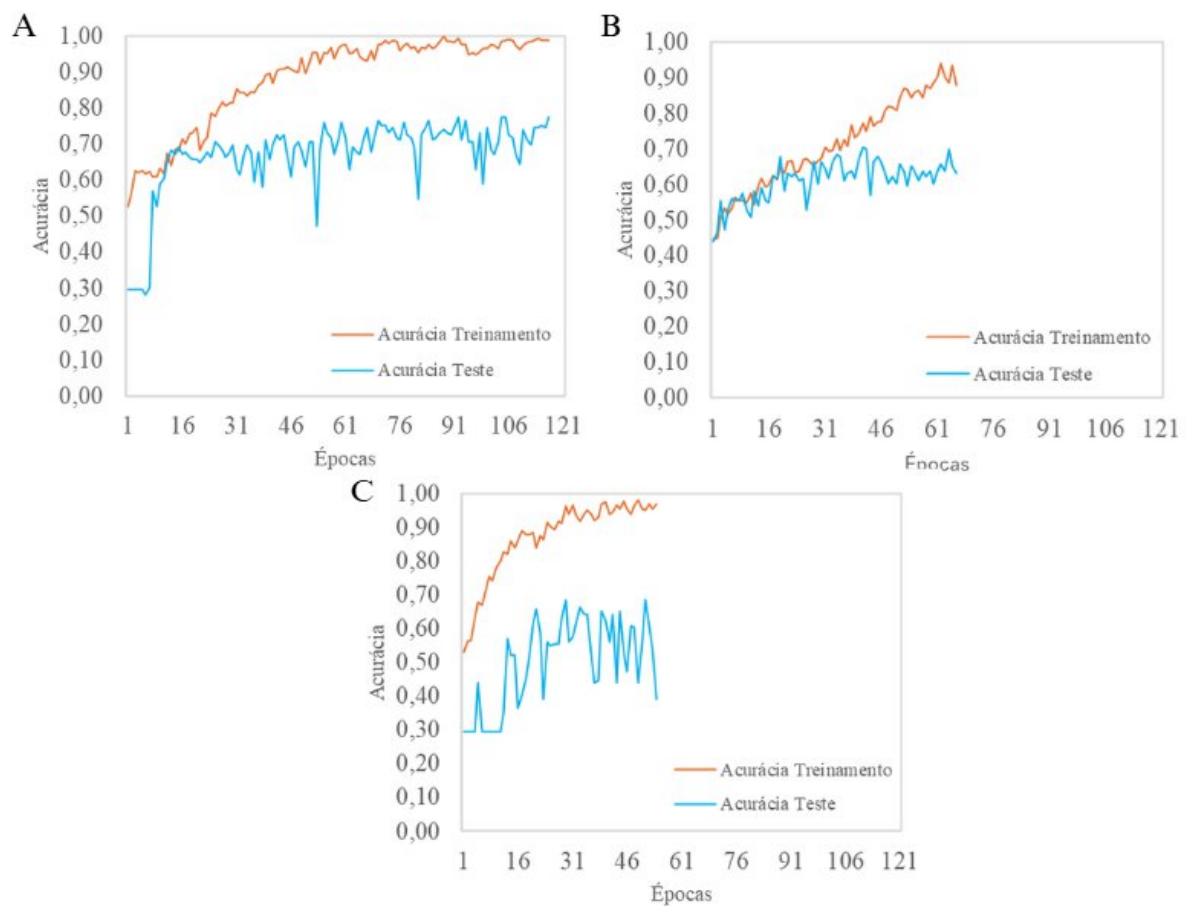


Fonte: Elaborado pelo autor

Ao analisar a acurácia foi observado comportamento similar para as três CNNs (

Figura 22). O desempenho atingiu valores próximos a 100,00% ao longo das épocas para os dados de treinamento. Para as épocas de melhor desempenho, verificou-se os valores de acurácia de 99,54% (treinamento) e 77,40% (teste) para a OakMoonNet (época 92), 77,80% (treinamento) e 70,08% (teste) para a AlexNet (época 41) e 96,30% (treinamento) e 68,49% (teste) para a MobileNet (época 29). Os resultados indicaram que as arquiteturas apresentaram limitações na capacidade de generalização quando aplicadas em um bando de imagens diferentes daquelas utilizadas para treinamento.

Figura 22. Respostas de Acurácia para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste.



Fonte: Elaborado pelo autor

No caso deste experimento, a dificuldade na distinção dos tipos de injúrias, inclusive na classificação visual, pode ter influenciado na capacidade de generalização das CNNs. O uso de um maior banco de dados, a divisão classes com diferenças mais evidentes e utilizando de

métodos de validação cruzada mais robustos (k-folds) podem ser alternativas para aproximar as acurácias obtidas a partir dos dados de treinamento e de teste.

A classificação de qualidade da laranja também foi avaliada por Asriny, Rani e Hidayatullah (2020) a partir de um dataset composto por 1000 imagens, subdivididas em 5 classes com 200 imagens. Para melhorar o desempenho de classificação, foi realizado o método da validação cruzada k-fold, além de uma variação na proporção dos dados de treinamento e teste. Já os autores Kukreja e Dhiman (2020) utilizaram a função do data augmentation para aumentar um banco de 150 imagens de frutos de laranja, para 1200 imagens com o intuito de classificá-los em duas classes (frutos com injúrias e sem injúrias), o que resultou na melhoria da acurácia da CNN.

Ao analisar as três CNNs, percebeu-se que todas apresentaram o mesmo padrão de desenvolvimento para os dados de treinamento e teste. Os valores de *Loss* apresentaram tendência decrescente para os dados de treinamento, enquanto apresentavam tendências crescentes para os dados de teste. Ao mesmo tempo, a acurácia que se mostrou crescente para os dados de treinamento, se estabilizando e não demonstrando tendência a melhoras para os dados de teste. Isso indica que, mesmo que houvesse maior tolerância de *early stop*, a acurácia não melhoraria, e os valores de *Loss* cresceriam cada vez mais.

A Tabela 8 apresenta um resumo geral dos parâmetros de desempenho gerados nas melhores épocas para cada CNN. Os resultados evidenciaram que a OakMoonNet apresentou comportamento mais equilibrado entre maiores acurácias e níveis baixos de Loss, sendo a arquitetura mais indicada para classificar laranjas em função dos níveis de injurias no epicarpo dos frutos.

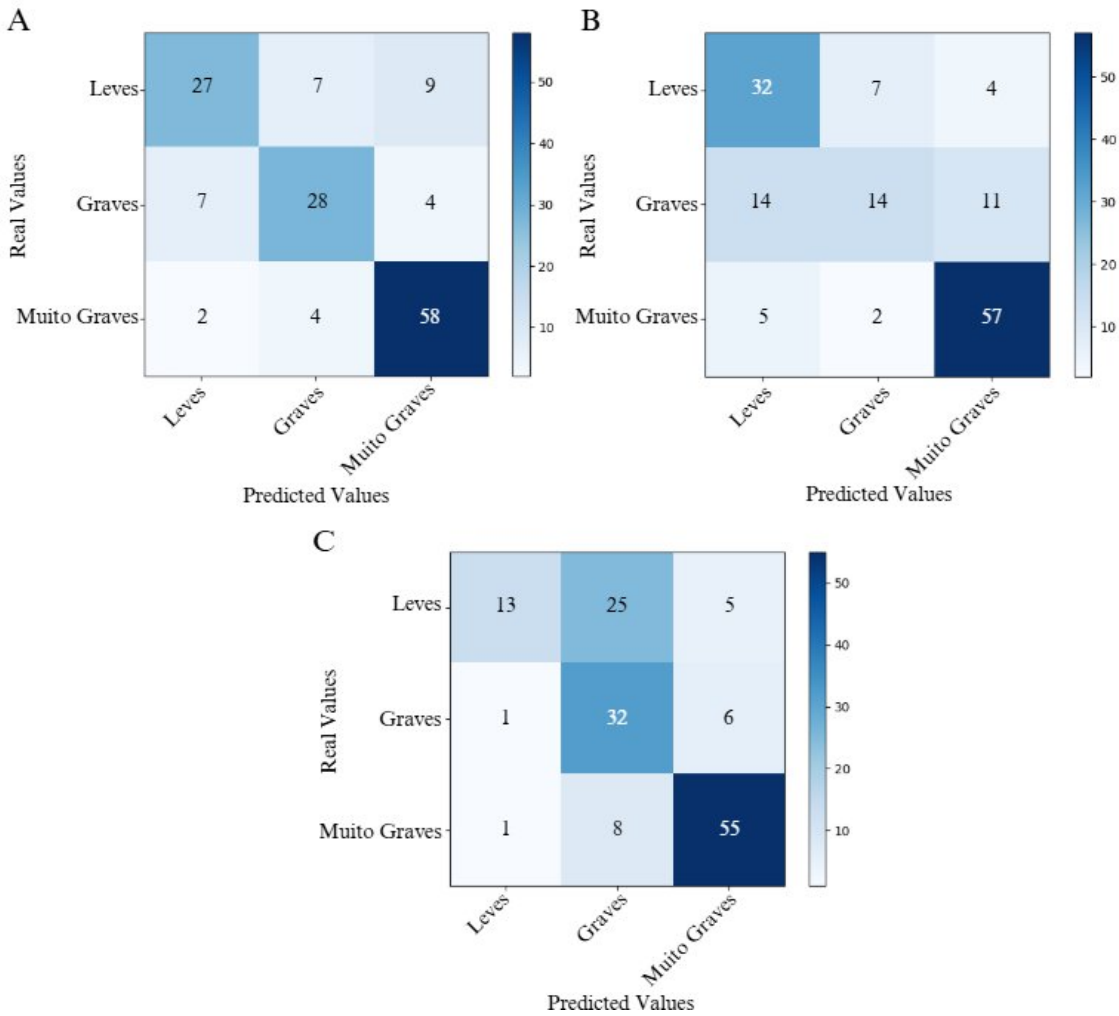
Tabela 8. Valores de Loss e Acurácia obtidos para os dados de treinamento e teste nas CNNs OakMoonNet, AlexNet e MobileNet.

Rede Neural	Melhor época	Treinamento		Teste	
		Loss	Acurácia	Loss	Acurácia
OakMoonNet	92	0,0170	99,54%	1,6851	77,40%
AlexNet	41	0,5303	77,08%	0,7831	71%
MobileNet	29	0,1241	96,30%	1,7774	68%

Fonte: Elaborado pelo autor

Ao analisar as matrizes de confusão gerada a partir das CNN das melhores épocas aplicadas aos dados de teste (Figura 23), pôde-se perceber que a OakMoonNet, AlexNet e MobileNet apresentaram maiores acertos na classificação dos frutos da classe Muito Graves.

Figura 23. Matriz de confusão para classificação das injurias presentes no epicarpo das laranjas utilizando as CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C).



Fonte: Elaborado pelo autor

Ainda pela matriz de confusão, foi possível comparar os níveis de acerto por classe entre as três CNNs. Assim, notou-se que a OakMoonNet foi a CNN com níveis de acertos mais equilibrados. dentre as três utilizadas. O desenvolvimento de CNNs customizadas tem-se mostrado uma alternativa eficiente para aumento da acurácia na detecção de injúrias em laranjas. Resultados semelhantes foram obtidos por Lu et al. (2024), que compararam uma CNN aprimorada com outras já existente, aplicada na detecção de tipos de injurias nos frutos

de laranja. A CNN aprimorada pelos autores superou as arquiteturas já existentes, obtendo acurácia média de 98,70%. O aprimoramento da CNN se deu pela implementação de uma nova camada focada nas coordenadas dos pixels.

Já no trabalho de Chen et al. (2020), o aprimoramento da CNN foi realizado na arquitetura já existente Yolov4, visando a detecção de quatro diferentes tipos de Citrus. A acurácia da CNN aprimorada apresentou-se superior em comparação com a Yolov4 original. Esse resultado foi obtido ao utilizar resoluções maiores e mais camadas na estrutura da rede neural.

As métricas apresentadas na Tabela 9 quantificaram o desempenho das CNNs e apontando a superioridade da OakMoonNet em comparação as outras CNNs (*F1-Score* superior em todas as três classes). Além do mais, o tempo de treinamento foi menor, (23 minutos e 56 segundos e 117 épocas), em comparação com os tempos obtidos pela AlexNet, (2 horas e 13 minutos e 66 épocas), e MobileNet (1 hora e 27 minutos e 54 épocas). Estes resultados demonstraram que a CNN customizada além de possibilitar os maiores níveis de acerto, também possibilitou um menor esforço computacional para o processamento dos dados.

Tabela 9. Métricas de desempenho das CNNs OakMoonNet, AlexNet e MobileNet.

Rede Neural	Classes	A _c	P _r	R _e	<i>F1-Score</i>	Épocas	Tempo de treinamento
OakMoonNet	Leves		75%	63%	68%	117	23 min e 56 seg
	Graves	77%	72%	72%	72%		
	Muito Graves		82%	91%	86%		
AlexNet	Leves		63%	74%	68%	66	2 h e 13 min
	Graves	71%	61%	36%	45%		
	Muito Graves		79%	89%	84%		
MobileNet	Leves		87%	30%	45%	54	1 h e 27 min
	Graves	73%	49%	82%	62%		
	Muito Graves		83%	86%	85%		

Fonte: Elaborado pelo autor

Os altos valores de precisão e sensibilidade para as três classes na OakMoonNet, em comparação com os valores obtidos pela AlexNet e MobileNet evidenciaram a melhor adaptabilidade da CNN customizada ao banco de dados utilizado.

A AlexNet apresentou sensibilidade de 36% para a classe Graves, e a MobileNet apresentou sensibilidade de 30% para a classe Leves, evidenciando que mesmo identificando corretamente os dados da classe, a maioria foi classificado como pertencente a outras classes. A MobileNet apresentou também precisão de 49% para a classe Graves, reforçando a tendência desta receber dados pertencentes a outras classes.

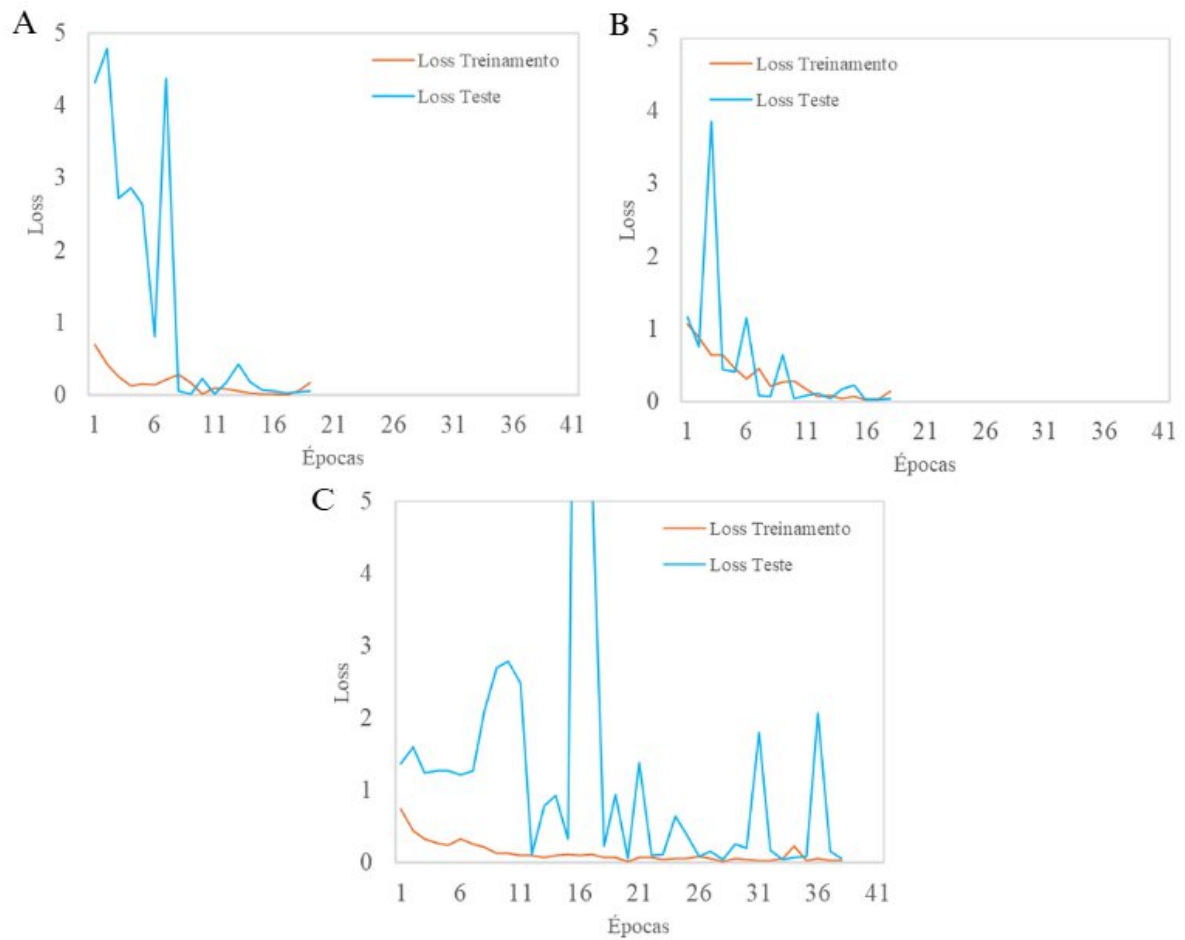
O desenvolvimento de CNNs próprias para detecção de injúrias ou doenças em laranjas pode ser aplicada no setor industrial, devido a velocidade de detecção e os altos índices de acurácia. Feng et al. (2023). Também desenvolveram uma CNN customizada (MSDD-YOLOX) e implementaram- em uma esteira seletora de laranjas. Os frutos foram categorizados em três classes (deformidade, cicatrizes e doenças). A CNN customizada apresentou desempenho superior a outras CNNs com arquiteturas tradicionais. Neste contexto, a OakMoonNet apresenta potencial prático para aplicação em sistemas de seleção de frutos em função do grau de injúrias.

No caso deste experimento ficou evidenciado que os dados pertencentes a classe Muito Graves apresentou maior diferença e quantidade de dados, enquanto os dados pertencentes a classe Leves e Graves apresentaram maior semelhança, havendo assim dificuldade na identificação destas classes. Utilizar um banco de dados mais rico ou separar os frutos em classes de mais fácil distinção poderiam gerar melhores métricas de desempenho na classificação.

5.4.2. Segundo experimento – Uso da câmara seletora para classificação dos frutos de tomate quanto a maturação

Ao analisar os valores de *Loss* (Figura 24), notou-se que todas as três CNNs conseguiram utilizar adequadamente o banco de dados de treinamento e teste, não havendo divergências. Analisando os valores de *Loss* nas melhores épocas de cada treinamento, verificou-se que em todos os casos e em todos os bancos de dados, os valores obtidos foram menores que 0,2500, sendo 0,1625 (treinamento) e 0,0096 (teste) para a OakMoonNet na época 9, 0,2022 (treinamento) e 0,0597 (teste) para a AlexNet na época 8 e 0,0260 (treinamento) e 0,0577 (teste) para a MobileNet na época 28. A MobileNet foi a única que apresentou *Loss* superior para os dados de teste.

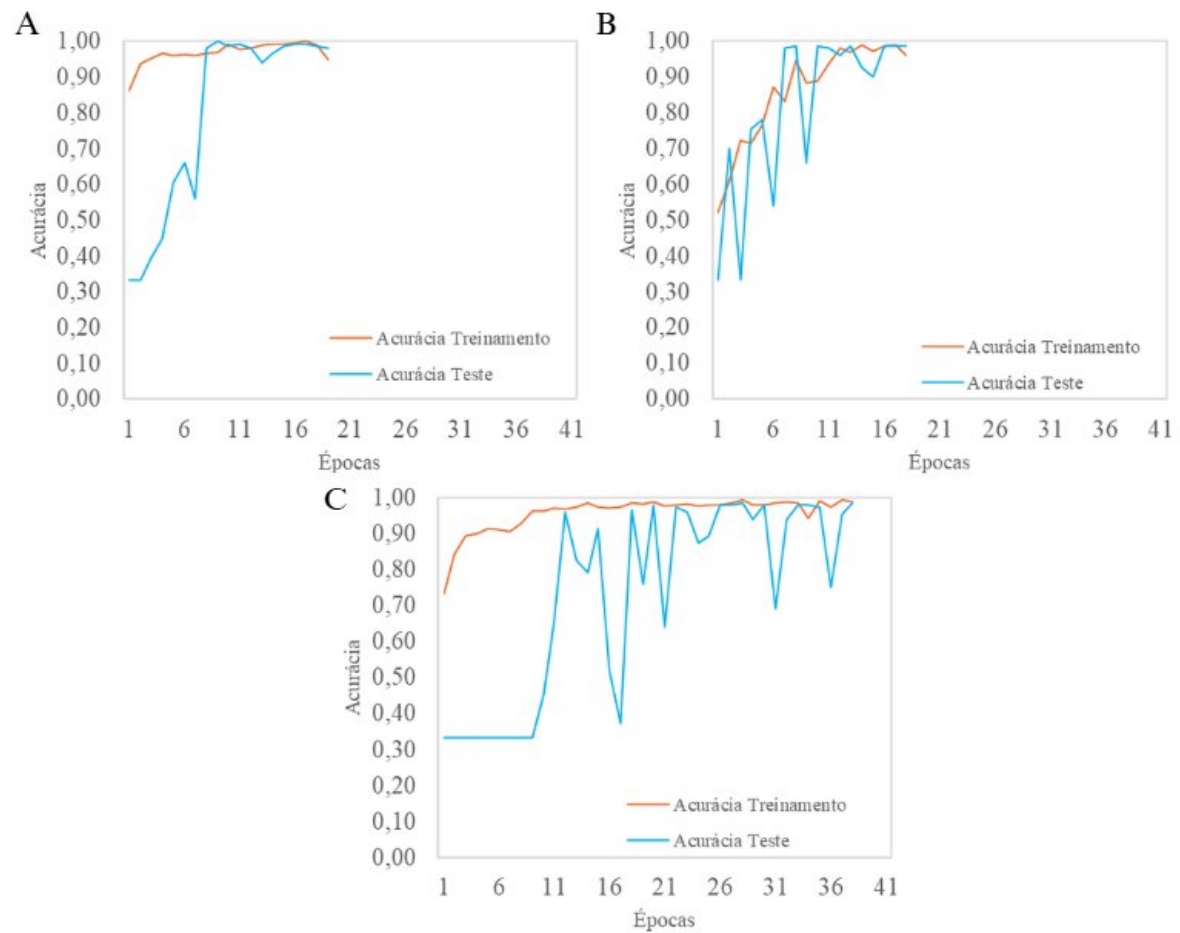
Figura 24. Respostas de *Loss* para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste.



Fonte: Elaborado pelo autor

Ao analisar a acurácia, notou-se novamente comportamento similar para as três CNNs, atingindo valores próximos a 100,00% de acurácia ao longo das épocas de treinamento e teste (Figura 25). Ao analisar as acurácias para época de melhor desempenho, verificou-se os valores de 96,89% (treinamento) e 100,00% (teste) para a OakMoonNet (época 9), 94,67% (treinamento) e 98,67% (teste) para a AlexNet (época 9) e 99,56% (treinamento) e 98,87% (teste) para a MobileNet (época 28). A MobileNet foi a única CNN que apresentou uma acurácia mais baixa para os dados de teste.

Figura 25. Respostas de Acurácia para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste.



Fonte: Elaborado pelo autor

A

Tabela 10 apresenta os valores de Loss e Acurácia obtidos para os dados de treinamento e teste na melhor época. Evidencia-se que somente a MobileNet obteve o *Loss* e Acurácia para os dados de treinamento superiores aos dados de teste. A OakMoonNet apresentou os menores valores de Loss, e em consequência, aproveitou melhor os dados de treinamento e teste, obtendo a maior acurácia.

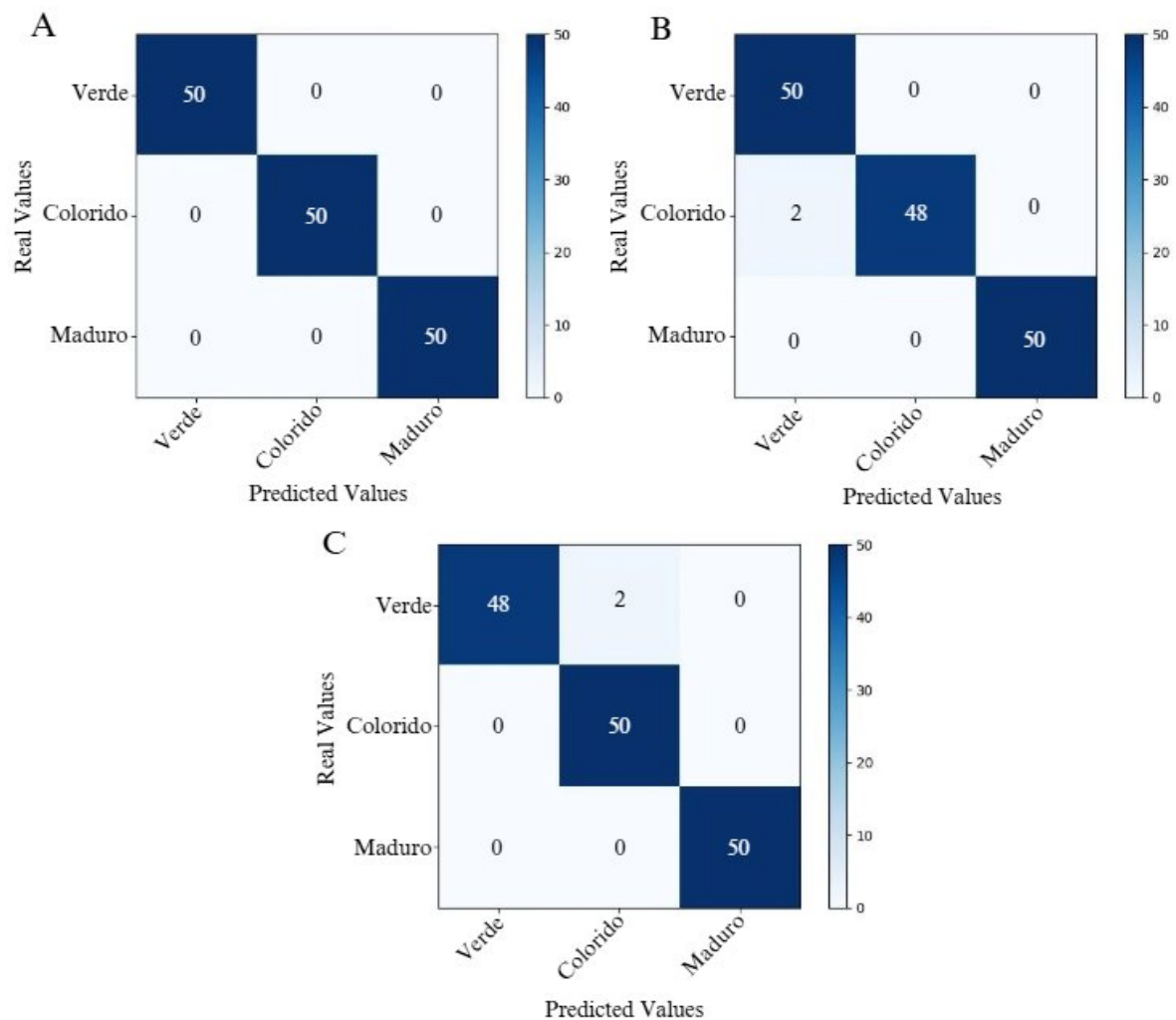
Tabela 10. Valores de Loss e Acurácia obtidos para os dados de treinamento e teste nas CNNs OakMoonNet, AlexNet e MobileNet.

Rede Neural	Melhor época	Treinamento		Teste	
		Loss	Acurácia	Loss	Acurácia
OakMoonNet	9	0,1625	96,89%	0,0096	100,00%
AlexNet	8	0,2022	94,67%	0,0597	98,67%
MobileNet	28	0,0260	99,56%	0,0477	98,87%

Fonte: Elaborado pelo autor

As matrizes de confusão demonstram que os únicos erros existentes nos treinamentos foram relacionados a classe Colorido (Figura 26). O desempenho dos três treinamentos foi influenciado pelo banco de dados, com classes facilmente diferenciáveis, devido as suas características únicas de cor.

Figura 26. Matriz de confusão para classificação dos frutos de tomate quanto a maturação utilizando as CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C).



Fonte: Elaborado pelo autor

As métricas de desempenho (Tabela 11) reforçam o alto desempenho das três CNNs. Apesar da acurácia de 100% para a OakMoonNet e 99% para a AlexNet e MobileNet, a maior diferença entre elas é vista no tempo de treinamento, a OakMoonNet exigiu somente 4 minutos e 12 segundo para obter desempenho superior a AlexNet, que treinou em 37 minutos e 26 segundos, e a MobileNet, que treinou em 1 hora e 5 minutos. A CNN customizada novamente possibilitou o maior número de acertos associado a um menor esforço computacional para o processamento dos dados.

Tabela 11. Métricas de desempenho das CNNs OakMoonNet, AlexNet e MobileNet.

Rede Neural	Classes	A_c	P_r	R_e	$F1-Score$	Épocas	Tempo de treinamento
OakMoonNet	Verdes		100%	100%	100%	19	4 min e 12 seg
	Coloridos	100%	100%	100%	100%		
	Maduros		100%	100%	100%		
AlexNet	Verdes		96%	100%	98%	18	37 min e 26 seg
	Coloridos	99%	100%	96%	98%		
	Maduros		100%	100%	100%		
MobileNet	Verdes		100%	96%	98%	19	1 h e 5 min
	Coloridos	99%	96%	100%	98%		
	Maduros		100%	100%	100%		

Fonte: Elaborado pelo autor

O desenvolvimento de CNNs customizadas para a detecção e classificação de frutos de tomate quanto ao nível de maturação tem se mostrado efetivo quanto a melhoria de desempenho de acurácia e redução de esforço computacional para o processamento das imagens. Wang et al. (2022) desenvolveram a MatDet, introduzindo a *Path Aggregation Network* (PANet) para lidar com as dificuldades de detectar a maturidade dos frutos de tomate em cenários complexos. Desta forma, obteve precisão média de 96,14% na detecção e classificação de frutos de tomate em campo. Já Das e Yadav (2020) propuseram um modelo de CNN própria para classificação de tomates em laboratório, obtendo acurácia de 99,67%. Percebe-se que o fruto de tomate é facilmente classificado por CNNs, independente do ambiente e iluminação que se encontram.

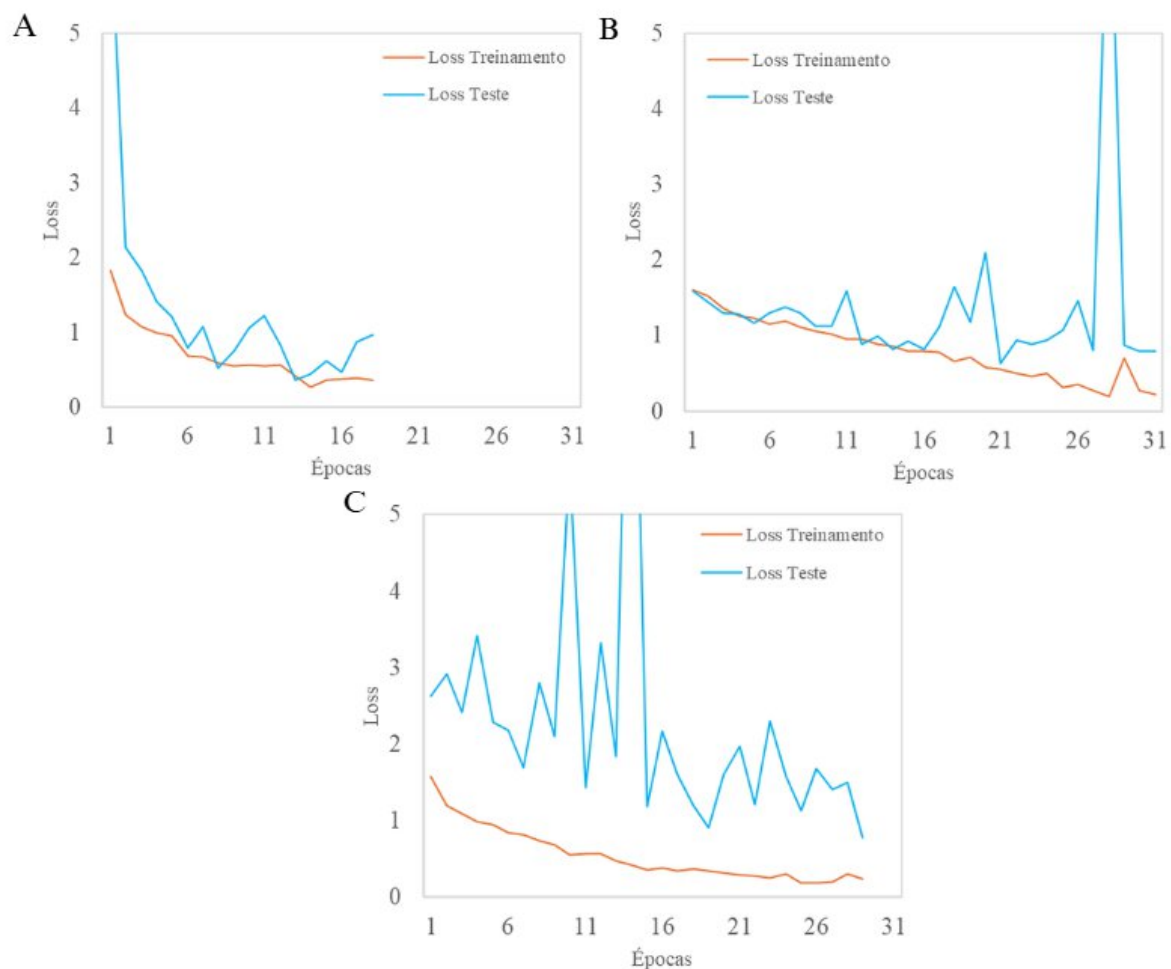
5.5. Outros experimentos de validação do software

5.5.1. Terceiro experimento – Classificação de frutos quanto ao seu tipo por meio de imagens obtidas na internet

Ao analisar os valores de *Loss* (Figura 27), nota-se que a OakMoonNet e a AlexNet lidaram de maneira eficiente com o banco de dados, apresentando baixa discrepância entre o tratamento dos dados de treinamento e teste. Contudo, a MobileNet apresentou dificuldade, apresentando uma diferença notável entre os valores de *Loss* para cada banco de dados. Ao

analisar o *Loss* nas épocas de melhor desempenho em cada treinamento, verificou-se os valores de 0,5947 (treinamento) e 0,5258 (teste) para a OakMoonNet (época 8), 0,5494 (treinamento) e 0,6276 (teste) para a AlexNet (época 21) e 0,3428 (treinamento) e 0,9160 (teste) para a MobileNet (época 19). A maior diferença entre os valores de *Loss* obtidos pela MobileNet evidencia a dificuldade desta CNN em lidar com o banco de dados.

Figura 27. Respostas de *Loss* para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste.

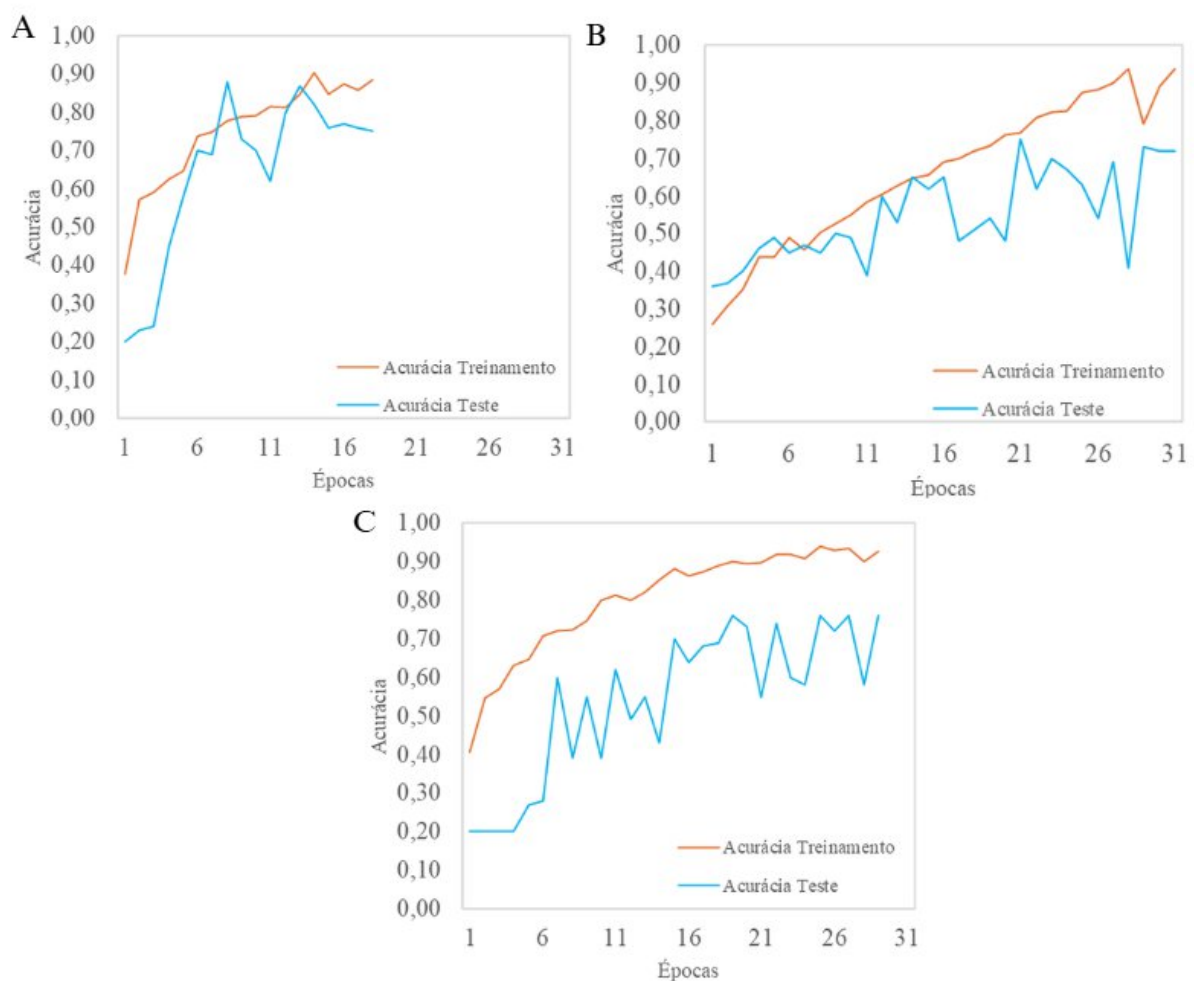


Fonte: Elaborado pelo autor

Os valores de *Loss* obtidos durante os treinamentos das três CNNs evidenciaram que um banco de dados não é necessariamente ruim, mas que uma CNNs pode apresentar dificuldade ou facilidade em lidar com os dados. Os valores de acurácia obtidos durante o treinamento das três CNNs (Figura 28) reforçaram a eficiência no tratamento dos bancos de dados pela OakMoonNet e AlexNet, e a dificuldade da MobileNet em associar as características

dos dados de treinamento na classificação dos dados de teste. Ao analisar as acurácias para época de melhor desempenho, verificou-se os valores de 77,78% (treinamento) e 88,00% (teste) para a OakMoonNet (época 8), 76,76% (treinamento) e 75,00% (teste) para a AlexNet (época 21) e 89,91% (treinamento) e 76,00% (teste) para a MobileNet (época 19). Os resultados reforçam a eficiência da CNN customizada, sendo a única a apresentar a acurácia de teste superior a de treinamento, além de ser a maior acurácia dos três treinamentos.

Figura 28. Respostas de Acurácia para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste.



Fonte: Elaborado pelo autor

A Tabela 12 apresenta os valores de *Loss* e Acurácia obtidos para os dados de treinamento e teste na melhor época. Embora a MobileNet tenha obtido o melhor aproveitamento do banco de dados para treinamento, apresentando o menor *Loss* e maior

Acurácia, a OakMoonNet performou melhor com os dados de teste, apresentando o menor Loss e maior Acurácia, treinando em menos épocas.

Tabela 12. Valores de Loss e Acurácia obtidos para os dados de treinamento e teste nas CNNs OakMoonNet, AlexNet e MobileNet.

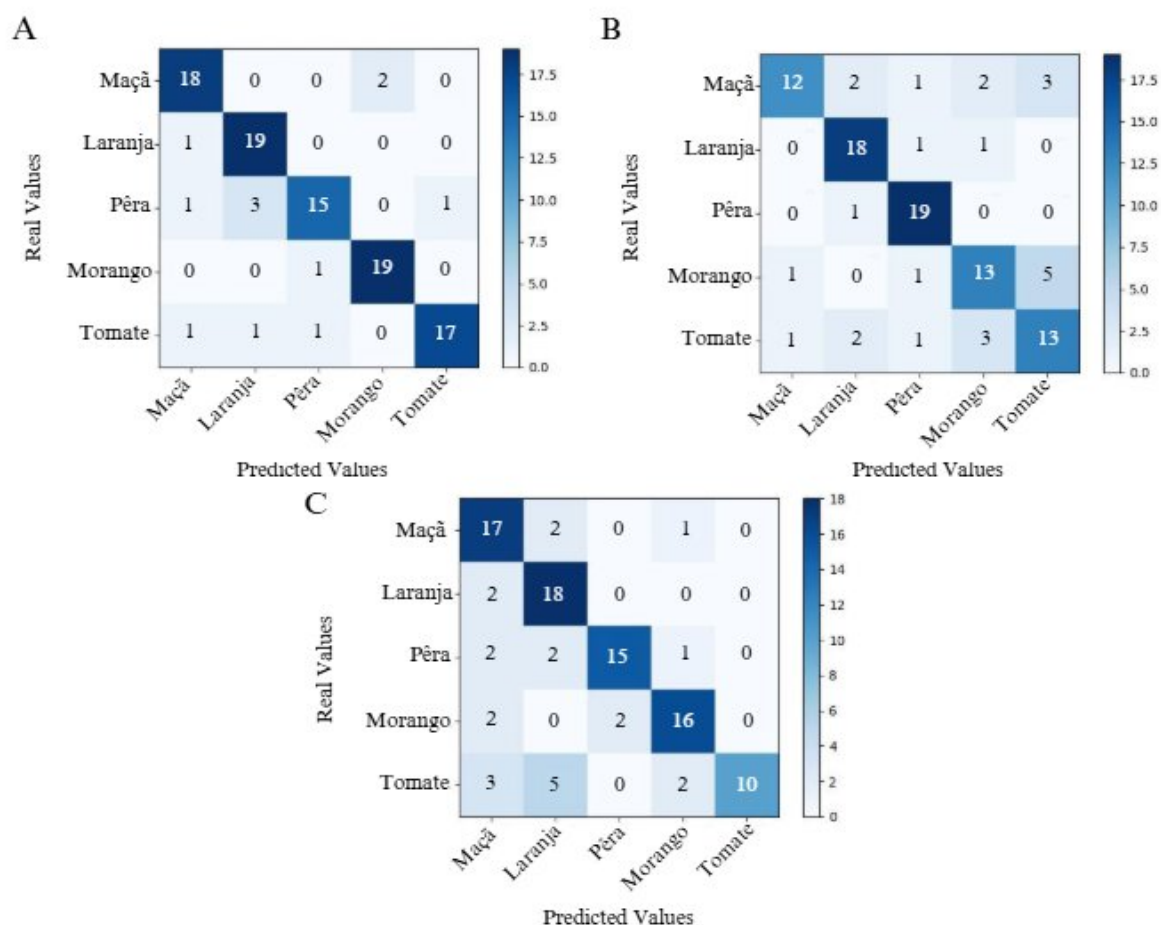
Rede Neural	Melhor época	Treinamento		Teste	
		Loss	Acurácia	Loss	Acurácia
OakMoonNet	8	0,5947	77,78%	0,5258	88,00%
AlexNet	21	0,5494	76,76%	0,6276	75,00%
MobileNet	19	0,3428	89,91%	0,9160	76,00%

Fonte: Elaborado pelo autor

Normalmente, estudos de classificação de múltiplos frutos utilizando CNN utilizam um dataset controlado, mantendo uma iluminação e posicionamento dos frutos constante, o que facilita o aprendizado. Neste experimento, as diferenças entre imagens da mesma categoria promovem maior robustez, mas gera um resultado com menor acurácia. Khatun et al. (2020) utilizou o dataset Fruits-360, do qual captura diversas imagens de um mesmo fruto rotacionado. Assim, foram utilizadas 1260 imagens, separadas em sete classes. Como as imagens de cada categoria são muito semelhantes, a acurácia obtida utilizando-se a MobileNet foi de 98.74%.

As matrizes de confusão obtidas para o treinamento das três CNNs (Figura 29) evidenciam que a OakMoonNet obteve melhor desempenho na identificação da classe Morango, e pior desempenho identificação da classe Pêra. A AlexNet obteve melhor desempenho na identificação da classe Pêra, e pior desempenho na identificação da classe Tomate. A MobileNet apresentou o melhor desempenho na identificação da classe Pêra, e pior desempenho na identificação da classe Laranja.

Figura 29. Matriz de confusão para classificação dos dados obtidos pela internet utilizando as CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C).



Fonte: Elaborado pelo autor

A diferença de aprendizado das três CNNs é evidenciada pelas matrizes de confusão. A OakMoonNet confundiu frutos de cores e formas diferentes (classes Pêra e Laranja), mesmo assim apresentou bom desempenho na identificação de todas as classes. A AlexNet apresentou dificuldade na identificação de frutos de cores e formas semelhantes, confundindo-os (classes Maçã, Morango e Tomate). A MobileNet também apresentou confusão entre os frutos de cores e formas semelhantes (classes Tomate, Laranja, Maçã e Morango). Nota-se, portanto, que as dificuldades serão diferentes para CNNs distintas.

Além disso, este experimento utilizou um banco de dados pequeno, tendo originalmente 250 imagens, sendo posteriormente expandido para 1000 imagens por meio do *data augmentation*. Goyal, Kumar e Verma (2023) treinaram a YOLOv5 para a identificação de frutos quanto ao seu tipo e posteriormente quanto a sua qualidade, utilizando um banco de dados composto por 10.545 imagens, divididos em 4 classes. A identificação do fruto quanto ao tipo

atingiu a acurácia de 92,89%, já a identificação do fruto quanto a sua qualidade atingiu acurácia média de 96,10%. Vale ressaltar que a escolha de um banco de dados pequeno se deu pelo fato do treinamento ser realizado diretamente em uma SBC. Treinamentos com bancos de dados mais robustos normalmente são realizados com computadores mais potentes.

As métricas de desempenho (Tabela 13) apontam que a OakMoonNet apresentou o melhor desempenho, destacando-se perante as demais CNNs, com acurácia de 88%, possuindo os maiores *F1-Scores* (com exceção da categoria Pêra), treinando em menos épocas e no menor tempo, de apenas 6 minutos e 24 segundos. A AlexNet apresentou acurácia de 75%, possuindo *F1-Scores* mais baixos em comparação com as outras CNNs para as categorias de Maçã, Laranja e Tomate, mas sendo a melhor para a categoria Pêra, treinando em 1 hora e 55 minutos, apresentando o maior número de épocas de treinamento. A MobileNet apresentou acurácia de 76%, treinando em 1 hora e 28 minutos, e apresentando desempenho intermediário para os valores de *F1-Score* entre as CNNs OakMoonNet e AlexNet

Tabela 13. Métricas de desempenho das CNNs OakMoonNet, AlexNet e MobileNet.

Rede Neural	Classes	A _c	P _r	R _e	<i>F1-Score</i>	Épocas	Tempo de treinamento
OakMoonNet	Maçã	88%	86%	90%	88%	18	6 min e 24 seg
	Laranja		83%	95%	88%		
	Pêra		88%	75%	81%		
	Morango		90%	95%	93%		
	Tomate		94%	85%	89%		
AlexNet	Maçã	75%	86%	60%	71%	31	1 h e 55 min
	Laranja		78%	90%	84%		
	Pêra		83%	95%	88%		
	Morango		68%	65%	67%		
	Tomate		62%	65%	63%		
MobileNet	Maçã	76%	65%	85%	74%	29	1 h e 28 min
	Laranja		67%	90%	77%		
	Pêra		88%	75%	81%		
	Morango		80%	80%	80%		
	Tomate		100%	50%	67%		

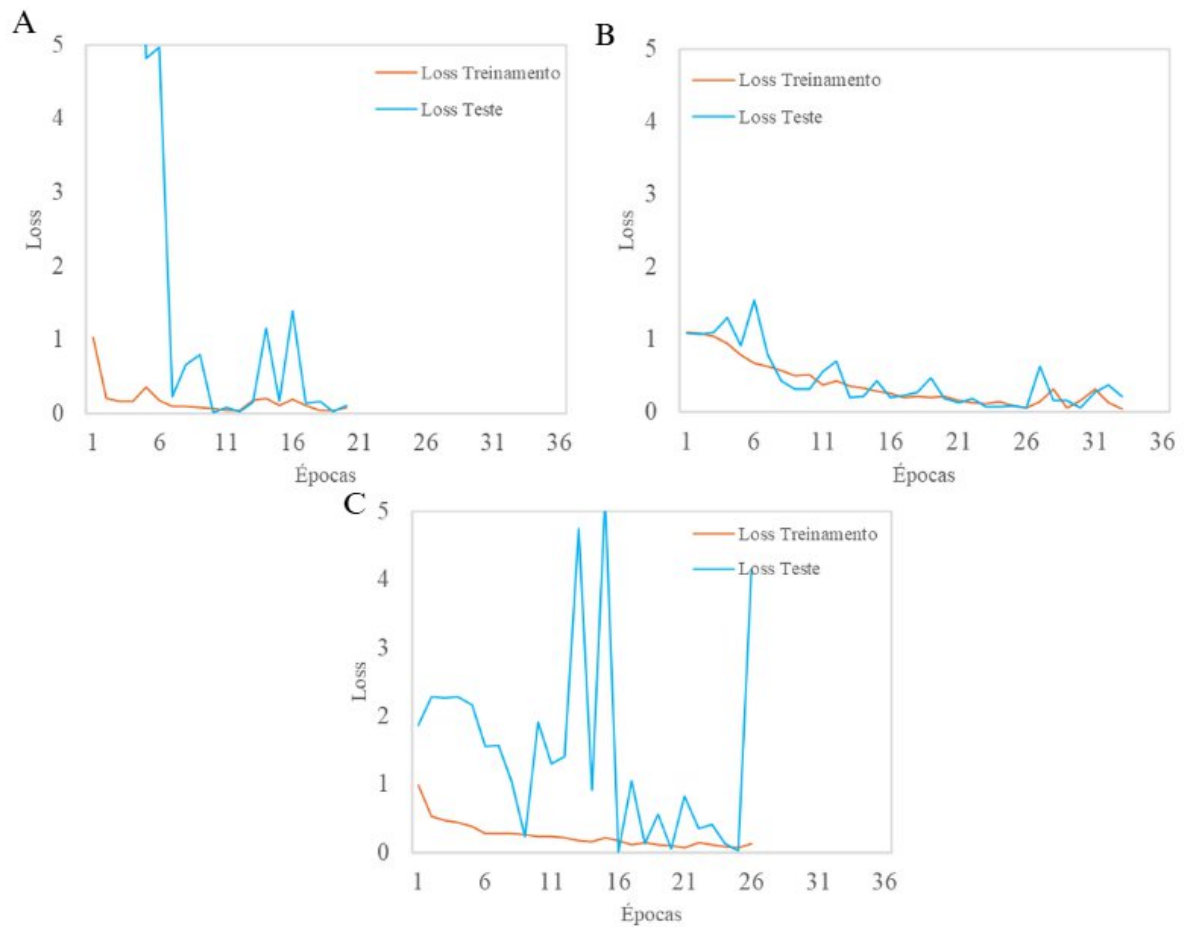
Fonte: Elaborado pelo autor

A criação de CNNs mais leves, focadas em classificação de categorias específicas tem se mostrado eficiente, principalmente em relação ao tempo de treinamento expressivamente menor, aliado ao desempenho equivalente ou superior às CNNs já existentes. Kumar et al. (2021) criou a Fruit-CNN, destinada a classificação de frutos diversos quanto ao seu tipo e quanto a sua qualidade. Utilizando o dataset FruisGB, contendo 12000 imagens, obteve 99,60% de acurácia treinando em 47 minutos. O mesmo treinamento foi realizado com a MobileNet, obtendo 99,40% de acurácia treinando em 7 horas.

5.5.2. Quarto experimento – Classificação de frutos de tomates na lavoura quanto ao nível de maturação

Ao analisar os valores de *Loss* (Figura 30), nota-se que todas as três CNNs processaram os bancos de dados de maneira eficiente. Ao analisar o *Loss* nas épocas de melhor desempenho em cada treinamento, verificou-se os valores de 0,0738 (treinamento) e 0,0125 (teste) para a OakMoonNet (época 10), 0,1114 (treinamento) e 0,0682 (teste) para a AlexNet (época 23) e 0,1766 (treinamento) e 0,0079 (teste) para a MobileNet (época 16). Em todos os três casos, os valores de *Loss* para os dados de treinamento foram menores que os obtidos para os dados de teste, evidenciando que a extração de características de distinção entre as classes foi realizada de maneira eficiente.

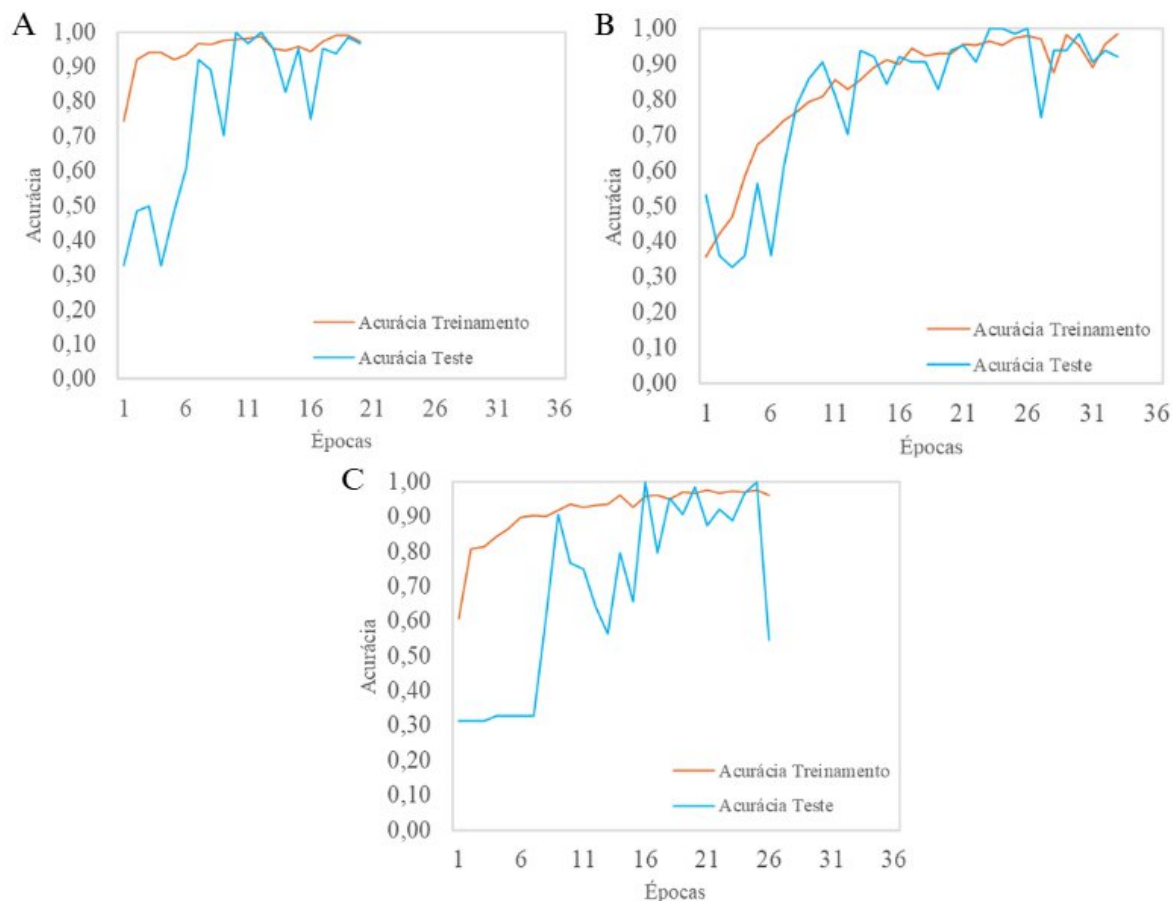
Figura 30. Respostas de *Loss* para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste.



Fonte: Elaborado pelo autor

O melhor desempenho dos dados de teste em relação aos dados de treinamento é evidenciado ao analisar a acurácia (Figura 31). Embora todas as três CNNs tenham atingido 100,00% de acurácia para os dados de teste, foram obtidos para os dados de treinamento 97,88% para a OakMoonNet (época 10), 96,65% para a AlexNet (época 23) e 95,77% para a MobileNet (época 16).

Figura 31. Respostas de Acurácia para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste.



Fonte: Elaborado pelo autor

A Tabela 14 apresenta os valores de Loss e Acurácia obtidos para os dados de treinamento e teste na melhor época. Todas as três CNNs aproveitaram melhor os dados de teste em relação aos dados de treinamento. Isso se deve, principalmente, a fácil diferenciação entre os dados de cada classe

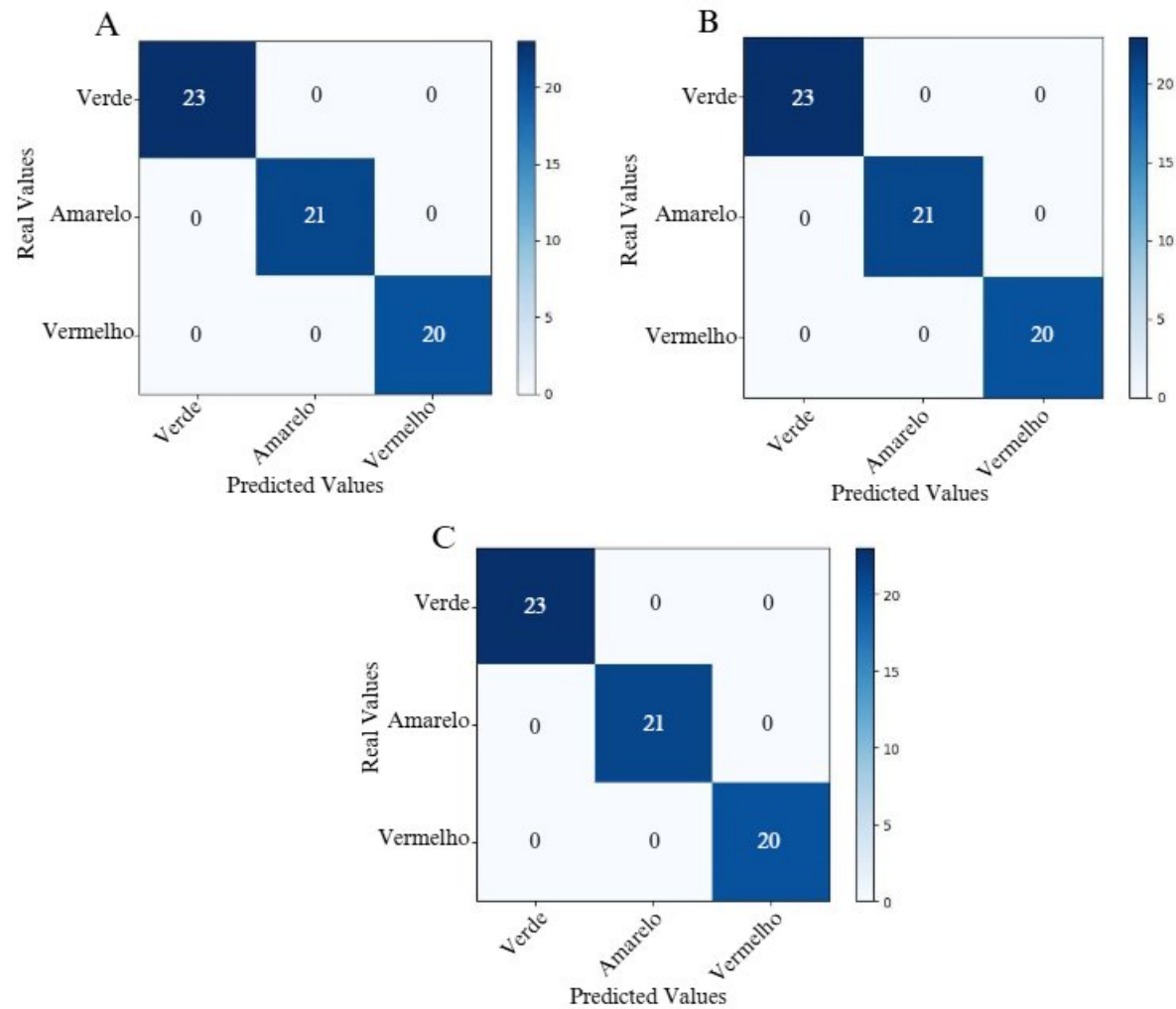
Tabela 14. Valores de Loss e Acurácia obtidos para os dados de treinamento e teste nas CNNs OakMoonNet, AlexNet e MobileNet.

Rede Neural	Melhor época	Treinamento		Teste	
		Loss	Acurácia	Loss	Acurácia
OakMoonNet	10	0,0738	97,88%	0,0125	100,00%
AlexNet	23	0,1114	96,65%	0,0682	100,00%
MobileNet	16	0,1766	95,77%	0,0079	100,00%

Fonte: Elaborado pelo autor

As matrizes de confusão obtidas durante o treinamento das três CNNs (Figura 32) evidenciam que os dados bem capturados, em condições homogêneas e facilmente diferenciáveis apresentarão melhor desempenho ao serem utilizados para treinamento de uma CNN. Além disso, a diferença expressiva de cor facilitou a diferenciação das três categorias utilizadas.

Figura 32. Matriz de confusão para classificação dos frutos de tomate na lavoura quanto a maturação utilizando as CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C).



Fonte: Elaborado pelo autor

As métricas de desempenho (Tabela 15) reforçam novamente que todas as três CNNs desempenharam igualmente, com 100% em todas as acurácias, precisões, sensibilidades e *F1-Scores*. Contudo, o tempo de treinamento mostra a diferença expressiva da OakMoonNet, que levou apenas 5 minutos e 10 segundos para alcançar o mesmo resultado da AlexNet e

MobileNet, que demoraram 1 hora e 25 minutos e 49 minutos e 2 segundos para realizar o treinamento, respectivamente. Além disso, a OakMoonNet exigiu menos épocas para encerrar seu treinamento, sendo apenas 20, em comparação com a AlexNet e MobileNet, que exigiram 33 e 26 épocas, respectivamente.

Tabela 15. Métricas de desempenho das CNNs OakMoonNet, AlexNet e MobileNet.

Rede Neural	Classes	A _c	P _r	R _e	<i>F1-Score</i>	Épocas	Tempo de treinamento.
OakMoonNet	Verdes		100%	100%	100%	20	5 min e 10 seg
	Amarelos	100%	100%	100%	100%		
	Vermelhos		100%	100%	100%		
AlexNet	Verdes		100%	100%	100%	33	1 h e 25 min
	Amarelos	100%	100%	100%	100%		
	Vermelhos		100%	100%	100%		
MobileNet	Verdes		100%	100%	100%	26	49 min e 2 seg
	Amarelos	100%	100%	100%	100%		
	Vermelhos		100%	100%	100%		

Fonte: Elaborado pelo autor

A classificação de frutos por CNN realizadas em campo, mas padronizadas quanto a forma de captura das imagens também apresentam alto desempenho, com acurácias quase sempre próximas do 100%. Huang, Wang e Basanta (2020) treinou a Mask R-CNN para detecção de tomates em campo, contando a quantidade de frutos por imagem, obtendo acurácia média foi de 98%. Hu et al. (2019) utilizou-se da Faster R-CNN para detecção de frutos de tomates maduros, dos quais posteriormente, segmentava o fundo e identificava as bordas dos frutos de tomate, estando separados, juntos, sobrepostos ou cobertos. Foram utilizadas 800 imagens, sendo separados 600 imagens para treinamento e 200 imagens para teste. A obtenção da acurácia média de 87,40% evidencia que nem sempre um banco de dados robusto é necessário para a obtenção de bons resultados, desde que estes dados sejam facilmente diferenciáveis.

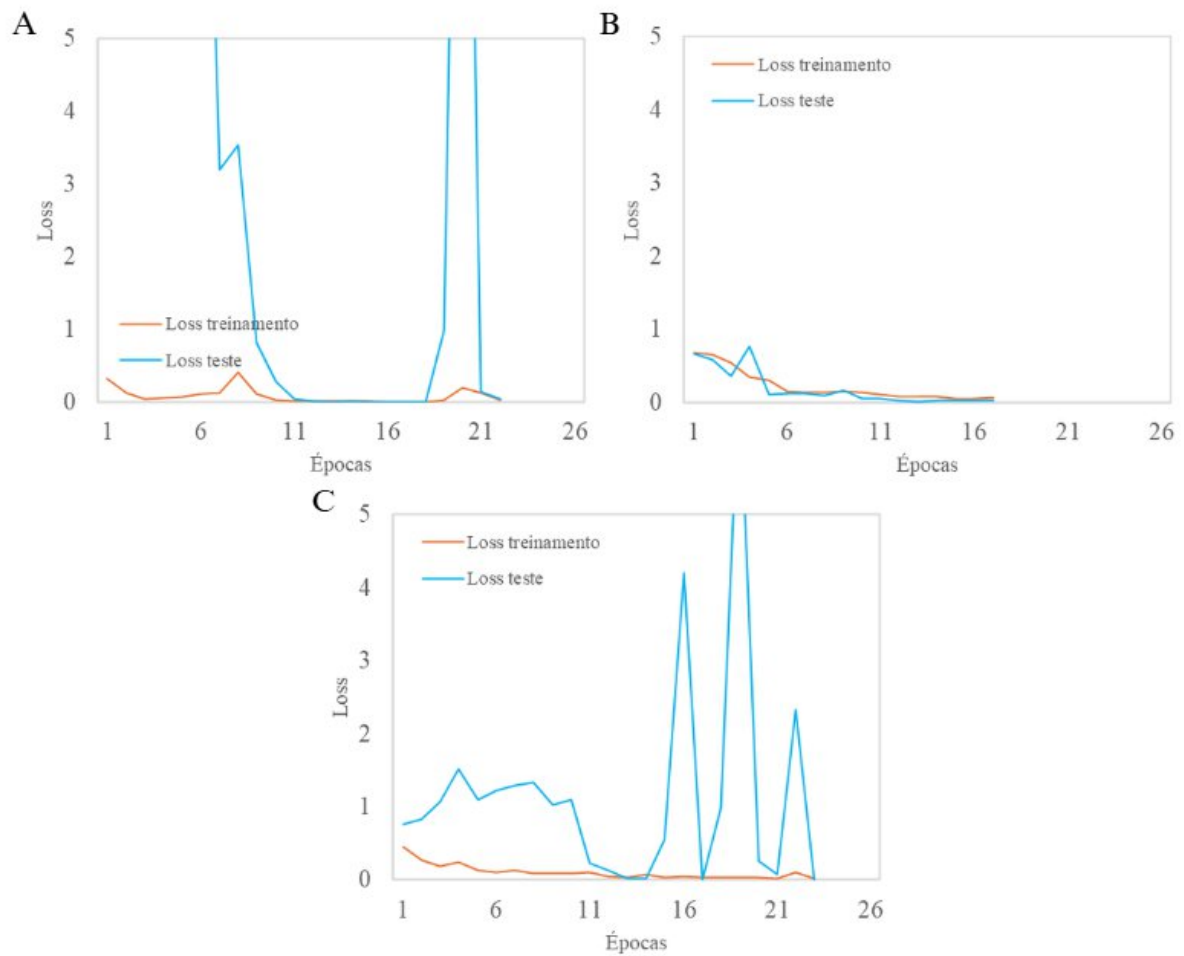
A aplicação de CNNs também vem sendo explorada para o cultivo de tomate em estufas. Zu et al. (2021) utilizou-se do Mask R-CNN para a detecção de frutos de tomates verdes, dos quais se tornam dificilmente perceptíveis por se mascararam com galhos e folhas verdes, com

imagens foram obtidas por um robô móvel, construído para este propósito, obtendo diferentes imagens em diferentes condições de ambiente. Foi atingido um *F1-Score* de 92,00%.

5.5.3. Quinto experimento – Classificação de frutos de tomate quanto a presença de injúrias no epicarpo

Ao analisar os valores de *Loss* (Figura 33), nota-se que a AlexNet foi a única que apresentou estabilidade nos valores obtidos para os dados de teste. A OakMoonNet e a MobileNet apresentaram diversos picos nos valores obtidos para os dados de teste ao longo do treinamento. Entretanto, todas as três CNNs apresentaram bom desenvolvimento e uso dos dados. Ao analisar o *Loss* nas épocas de melhor desempenho em cada treinamento, verificou-se os valores de 0,0050 (treinamento) e 0,0095 (teste) para a OakMoonNet (época 12), 0,1388 (treinamento) e 0,1207 (teste) para a AlexNet (época 7) e 0,0369 (treinamento) e 0,0184 (teste) para a MobileNet (época 13). A OakMoonNet apresentou os menores valores dentre as três, mostrando seu melhor aproveitamento do banco de dados.

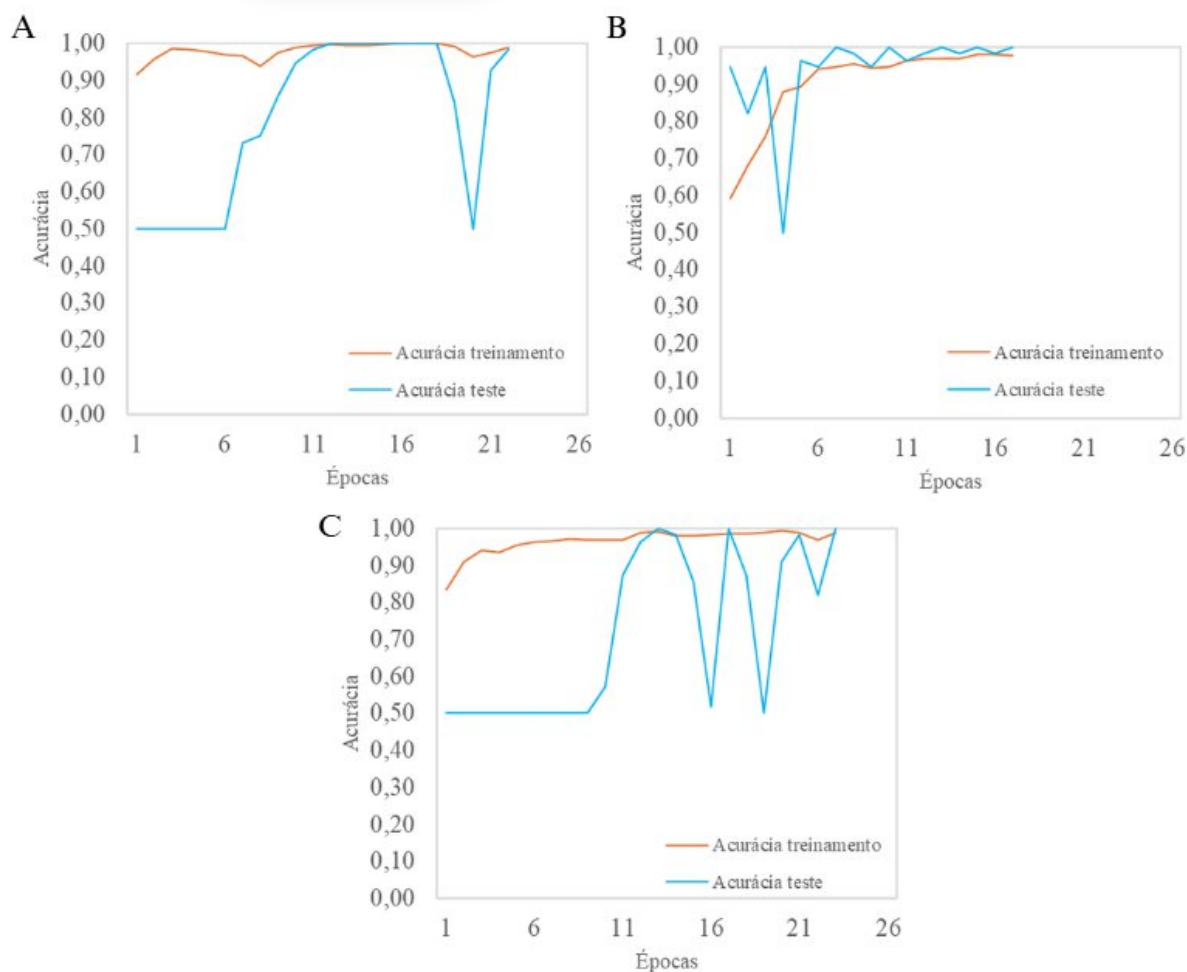
Figura 33. Respostas de *Loss* para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste.



Fonte: Elaborado pelo autor

Ao analisar a acurácia obtida durante os treinamentos (Figura 34), nota-se que todas as três CNNs alcançaram 100% de acurácia para os dados de teste, mas foram obtidos para treinamento 99,76% para a OakMoonNet (época 12), 94,73% para a AlexNet (época 7) e 99,18% para a MobileNet (época 13). Tal resultado pode ser justificado pela presença de somente duas classes, com imagens bem diferenciáveis.

Figura 34. Respostas de Acurácia para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste.



Fonte: Elaborado pelo autor

A Tabela 16 apresenta os valores de Loss e Acurácia obtidos para os dados de treinamento e teste na melhor época. Os valores de Loss, tanto para treinamento quanto teste, demonstram que a OakMoonNet lidou com mais facilidade com os dados em relação as outras CNNs. Contudo, isso não impediu com que todas alcançassem 100,00% de acurácia para os dados de teste.

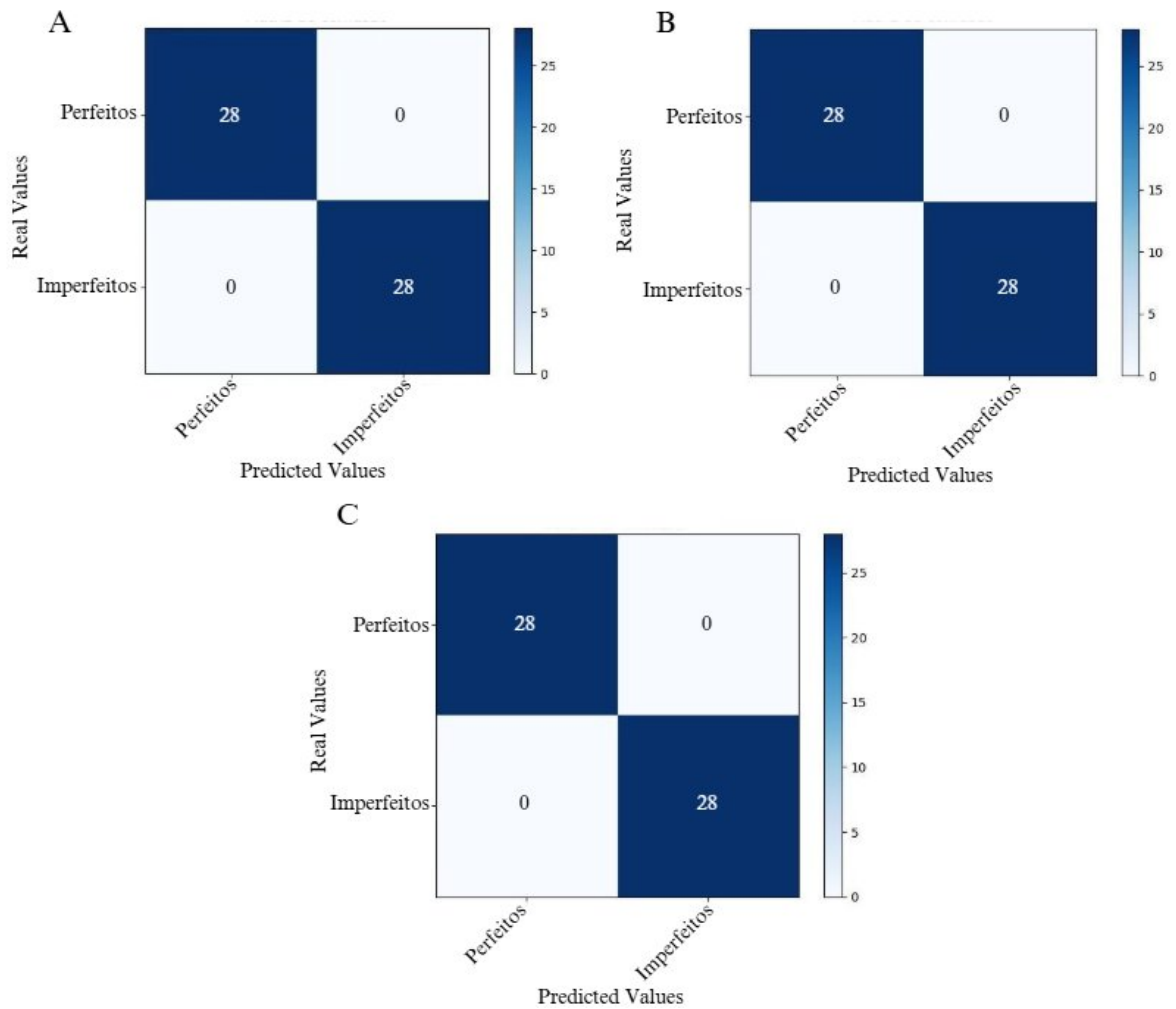
Tabela 16. Valores de Loss e Acurácia obtidos para os dados de treinamento e teste nas CNNs OakMoonNet, AlexNet e MobileNet.

Rede Neural	Melhor época	Treinamento		Teste	
		Loss	Acurácia	Loss	Acurácia
OakMoonNet	12	0,0050	99,80%	0,0095	100,00%
AlexNet	7	0,1388	94,73%	0,1207	100,00%
MobileNet	13	0,0369	99,19%	0,0184	100,00%

Fonte: Elaborado pelo autor

As matrizes de confusão obtidas durante o treinamento das três CNNs (Figura 35) evidenciaram que dados facilmente diferenciáveis, mesmo que em condições de iluminações não controlada, são passíveis de classificação com altos níveis de acurácia. Neste caso, as cores e formas semelhantes entre os frutos perfeitos e imperfeitos não impediu que a rede neural identificasse as características, mostrando que a cor não é um fator decisivo para o desempenho das CNNs.

Figura 35. Matriz de confusão para classificação dos frutos de tomate quanto a presença de injúrias no epicarpo utilizando as CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C).



Fonte: Elaborado pelo autor

As métricas de desempenho (Tabela 17) reforçam novamente que todas as três CNNs desempenharam igualmente, com 100% em todas as acurácias, precisões, sensibilidades e *F1-Scores*. Contudo, o tempo de treinamento mostra a diferença expressiva da OakMoonNet, que levou apenas 5 minutos e 24 segundos para treinar 22 épocas e alcançar o mesmo resultado da AlexNet e MobileNet, que demoraram 38 minutos e 30 segundos com 17 épocas e 38 minutos e 14 segundos com 23 épocas para realizar o treinamento, respectivamente.

Tabela 17. Métricas de desempenho das CNNs OakMoonNet, AlexNet e MobileNet.

Rede Neural	Classes	A _c	P _r	R _e	F1-Score	Épocas	Tempo de
-------------	---------	----------------	----------------	----------------	----------	--------	----------

							treinamento
OakMoonNet	Perfeito	100%	100%	100%	100%	22	5 min e 24 seg
	Imperfeito		100%	100%	100%		
AlexNet	Perfeito	100%	100%	100%	100%	17	38 min e 30 seg
	Imperfeito		100%	100%	100%		
MobileNet	Perfeito	100%	100%	100%	100%	23	38 min e 14 min
	Imperfeito		100%	100%	100%		

Fonte: Elaborado pelo autor

Com isso, percebe-se que as CNNs também identificam frutos de cor e forma semelhantes, mas com variações no epicarpo de forma eficiente. Em todos os treinamentos, não houve erros de classificação, mas sim uma diferença na tratativa dos dados de treinamento e teste.

A classificação dos frutos de tomate feita de maneira isolada para seu nível de maturação ou estado do epicarpo apresenta menor dificuldade do que quando estas classes estão associadas. Khatun et al. (2023) realizou a coleta de dois diferentes bancos de dados de frutos de tomate, diferenciando os frutos de tomate quanto ao nível de maturação (4000 imagens), e outro que diferenciava os frutos de tomate quanto a presença de danos no epicarpo (6000 imagens). Ao utilizar a MobileNetV2, obteve 98,00% e 96,00% de acurácia para os dados de maturação e presença de danos, respectivamente.

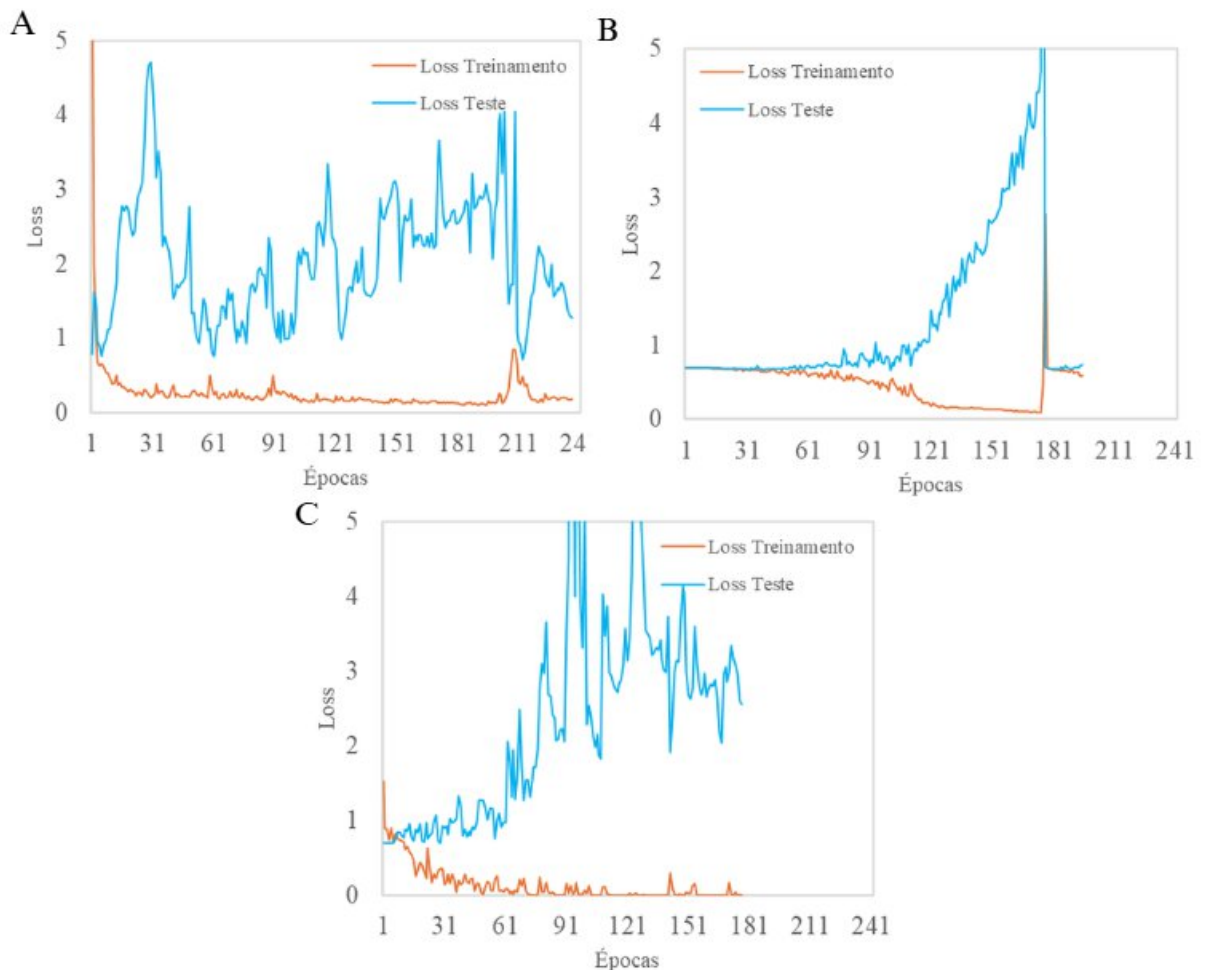
Chaturvedi, Sharma e Janghel (2023) realizaram a classificação de frutos de tomate quanto ao estado do epicarpo, classificando-os em Saudável e Defeituoso, utilizando um banco de dados de 43.849 imagens dos frutos obtidas em laboratório, com iluminação controlada. A melhor acurácia obtida foi 97.97%, e as outras CNNs treinadas apresentaram desempenho parecido. Percebe-se que quanto maior o banco de dados, maior a propensão de obter acurácia abaixo de 100,00%, indicando maior robustez do algoritmo.

5.5.4. Sexto experimento – Classificação de imagens *Time History Spacial Pattern* (THSP) de batatas para detecção do *internal bruising* quanto ao tempo de impacto

Ao analisar os valores de *Loss* (Figura 36), notou-se que enquanto a AlexNet apresenta um desenvolvimento estável, ocorrendo o encontro dos valores obtidos para ambos os bancos de dado ao final do treinamento, as outras CNNs apresentaram *Loss* decrescente para os dados

de treinamento e crescente e oscilante para os dados de teste. Ao analisar o *Loss* na época de melhor desempenho em cada treinamento, verificou-se os valores de 0,1673 (treinamento) e 1,5747 (teste) para a OakMoonNet (época 137), 0,5052 (treinamento) e 0,7920 (teste) para a AlexNet (época 95) e 0,0066 (treinamento) e 1,9727 (teste) para a MobileNet (época 77). Portanto, além da notável dificuldade de associação de características em comum para os dados de treinamento e teste, percebe-se que, mesmo que houvesse maior treinamento, não seriam obtidas melhoras.

Figura 36. Respostas de *Loss* para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste.

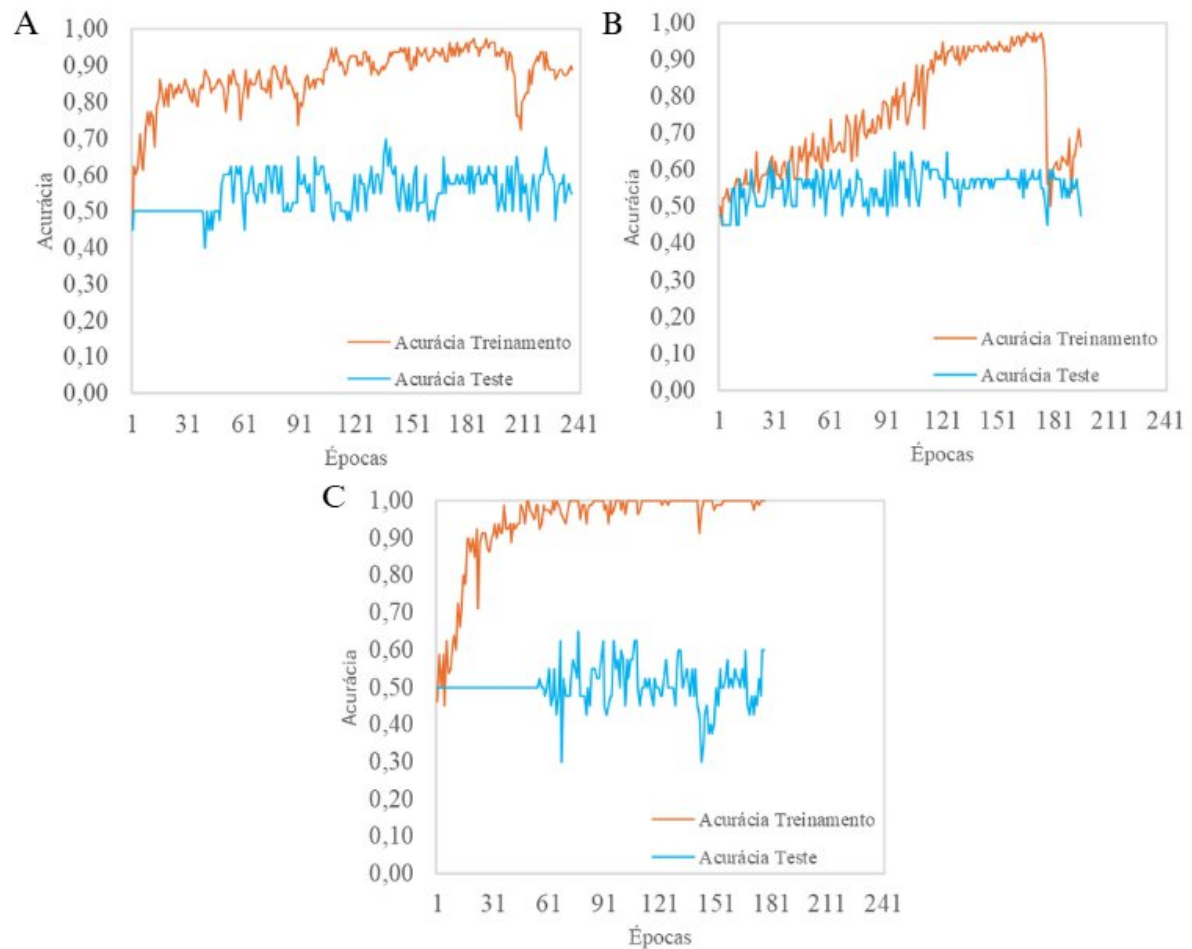


Fonte: Elaborado pelo autor

Ao analisar os valores de acurácia (Figura 37), notou-se que mesmo obtendo valores próximos a 100,00% para os dados de treinamento, os valores de teste ficaram abaixo e se

estabilizaram durante todo o treinamento das três CNNs. Ao analisar as acurácias para época de melhor desempenho, verificou-se os valores de 90,00% (treinamento) e 70,00% (teste) para a OakMoonNet (época 137), 77,55% (treinamento) e 65,00% (teste) para a AlexNet (época 95) e 100,00% (treinamento) e 65,00% (teste) para a MobileNet (época 77). É reforçado, portanto, que devido ao pequeno banco de dados, não foi possível obter melhores resultados.

Figura 37. Respostas de Acurácia para as épocas de treinamento das CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C) aplicadas ao conjunto de dados de treinamento e teste.



Fonte: Elaborado pelo autor

A

Tabela 14 apresenta os valores de Loss e Acurácia obtidos para os dados de treinamento e teste na melhor época. A MobileNet apresentou muito mais facilidade em lidar com os dados de treinamento em relação as outras CNNs, sendo isso evidenciado pelo valor de Loss, e consequentemente a maior acurácia para os dados de treinamento. Contudo, a OakMoonNet apresentou a maior acurácia para os dados de treinamento.

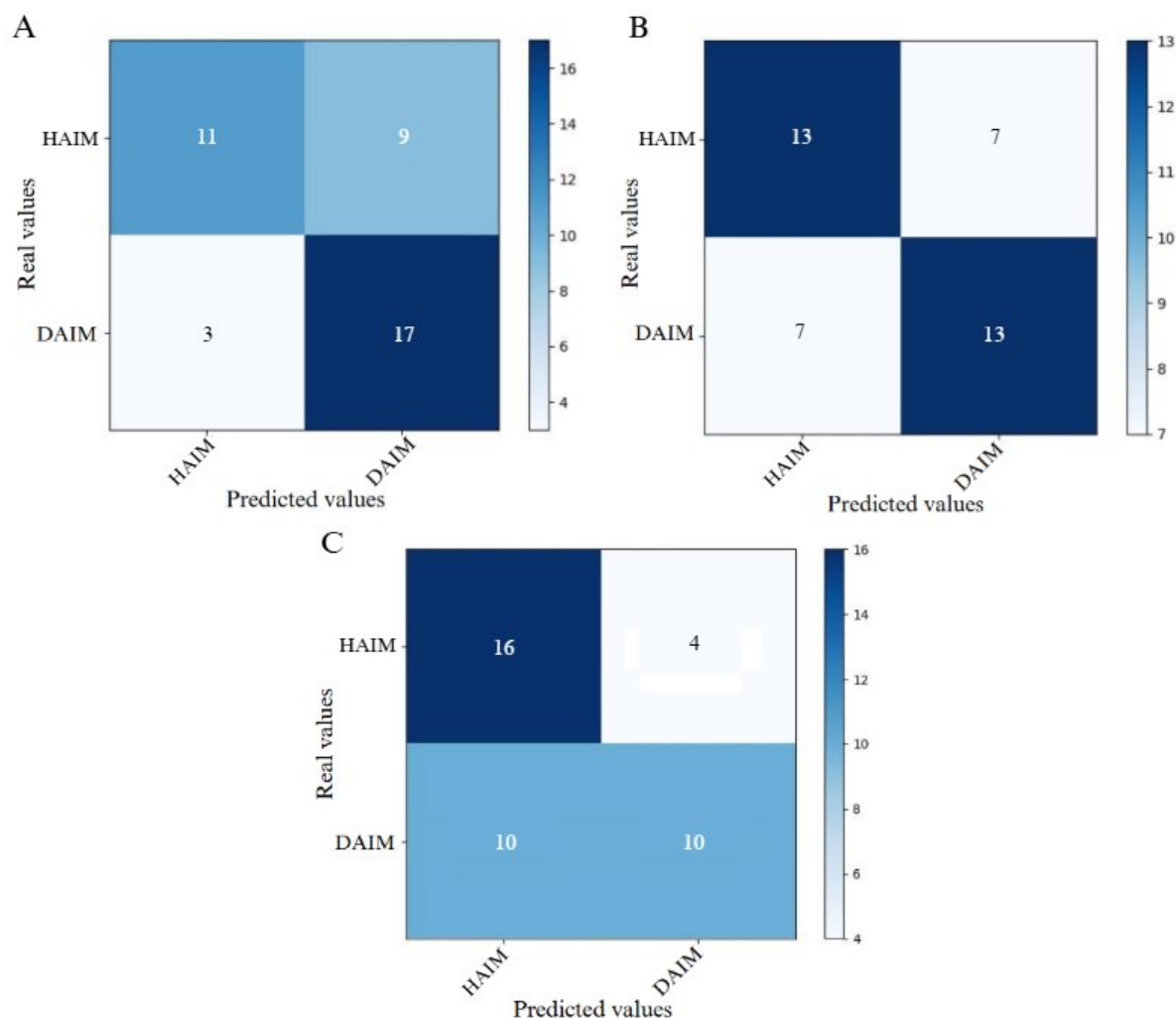
Tabela 18. Valores de Loss e Acurácia obtidos para os dados de treinamento e teste nas CNNs OakMoonNet, AlexNet e MobileNet.

Rede Neural	Melhor época	Treinamento		Teste	
		Loss	Acurácia	Loss	Acurácia
OakMoonNet	137	0,1673	90,00%	1,5747	70,00%
AlexNet	95	0,5052	77,55%	0,7920	65,00%
MobileNet	77	0,0066	100,00%	1,9427	65,00%

Fonte: Elaborado pelo autor

As matrizes de confusão (Figura 38) evidenciam as diferenças que cada CNN teve ao lidar com cada classe. A OakMoonNet apresentou maior tendência em classificar os dados como DAIM, a AlexNet apresentou um resultado não tendencioso, e a MobileNet apresentou tendência em classificar os dados como HAIM. Classificações tendenciosas são ruins, pois embora acerte com maior precisão determinada classe, também classifica erroneamente a outra classe, afetando, por consequência, a acurácia.

Figura 38. Matriz de confusão para classificação dos THSPs das batatas em horas após o impacto mecânico (HAIM) e em dias após o impacto mecânico (DAIM) utilizando as CNNs OakMoonNet (A), AlexNet (B) e MobileNet (C).



Fonte: Elaborado pelo autor

Como demonstrado por Costa et al. (2018) áreas de internal bruising de batatas tendem a apresentar maiores níveis de atividade biológica nas primeiras horas após a realização do impacto mecânico. Uma análise em intervalos mais amplos (mensurações logo após o impacto e após 72 horas após a ocorrência do impacto) podem contribuir para o aumento do acerto da classificação. Contudo, esta distinção não afetou os resultados obtidos pelo treinamento das três CNNs.

As métricas de desempenho (Tabela 19) evidenciam que mesmo que a OakMoonNet apresente a maior acurácia (70,00%) em comparação com a AlexNet e MobileNet (65,00%), a AlexNet foi a única que apresentou um desempenho totalmente equilibrado em ambas as

classes. Em consequência, apresentou o maior tempo de treinamento, de 13 minutos e 10 segundos, enquanto a OakMoonNet treinou em apenas 2 minutos e 14 segundos, e a MobileNet treinou em 9 minutos.

Tabela 19. Métricas de desempenho das CNNs OakMoonNet, AlexNet e MobileNet.

Rede Neural	Classe	A_c	P_r	R_e	$F1-Score$	Épocas	Tempo de Treinamento
OakMoonNet	HAIM	70%	79%	55%	65%	237	2 min e 14 seg
	DAIM		65%	85%	74%		
AlexNet	HAIM	65%	65%	65%	65%	195	13 min e 10 seg
	DAIM		65%	65%	65%		
MobileNet	HAIM	65%	62%	80%	70%	177	9 min
	DAIM		71%	50%	59%		

Fonte: Elaborado pelo autor

Abordagens para adaptação dos modelos quando a quantidade de dados de treinamento é limitada (Wang, Ramanan e Hebert, 2017) e uso de modelos de CNN de baixa complexidade, como as arquiteturas testadas nesta pesquisa, podem alcançar desempenho comparável a arquiteturas mais complexas em conjuntos de dados menores (Brigato e Iocchi, 2020). Isso se prova com o fato da OakMoonNet ter obtido a melhor acurácia e o menor tempo de treinamento. Características específicas da arquitetura como a menor profundidade e o número reduzido de filtros, tendem a favorecer a generalização sem a necessidade de grandes volumes de dados para treinamento, aumentando a eficiência computacional em comparação com outras CNNs no atual estado da arte (Chakraborty et al., 2023)

Neste experimento, não se utilizou a técnica do *data argumentation* em imagens de THSP pois a espacialidade dos pixels está diretamente associada aos padrões do *biospeckle* obtidos ao longo do tempo durante o monitoramento da atividade biológica da amostra. Por consequência, os resultados foram afetados negativamente. A utilização de um volume maior de dados de entrada possibilitaria que as CNNs obtivessem modelos mais robustos e generalizáveis a partir de uma variedade mais ampla de características (Alomar, Aysel e Cai, 2023).

As CNNs vêm sendo aplicadas na análise de padrões de *biospeckle* por meio de diferentes abordagens. Kaler, Bhatia e Chollet (2023) desenvolveram um modelo baseado em aprendizado profundo tridimensional (3D-CNN e ConvLSTM) para analisar dados de

biospeckle laser, visando a detecção e classificação de doenças em sementes sob diferentes parâmetros experimentais e níveis de ruído. Nesse estudo, a rede neural utilizou como entrada sequências temporais de padrões de BSL (frames sucessivos), permitindo a extração de informações espaço-temporais para a classificação. Além disso, foram empregados outros algoritmos de aprendizado de máquina para auxiliar na etapa de classificação. Embora essa abordagem tenha alcançado alta acurácia, o método pode ser considerado com certa complexidade por utilizar de forma combinada de diferentes técnicas de machine learning e *deep learning*. Outros estudos aplicando CNNs em 3 dimensões nos padrões do *biospeckle* podem ser observadas na detecção precoce de alterações biológicas em produtos agrícolas (Thakur et al. 2024)

Em contraste, a metodologia proposta neste estudo adota uma abordagem inovadora e mais simplificada, ao utilizar as CNNs sobre os padrões de THSP, que são gerados a partir das sequências temporais dos padrões de *biospeckle*. Dessa forma, busca-se reduzir a complexidade do processamento e obter uma classificação eficiente e direta pela CNN implementada.

6. CONSIDERAÇÕES FINAIS

A câmara seletora para coleta de dados e processamento utilizando CNNs foi desenvolvida, e se mostrou funcional e de fácil operação.

As CNNs AlexNet e MobileNet funcionaram adequadamente. A AlexNet apresentou um padrão de aprendizado que melhora com o tempo, com poucas oscilações de *Loss* durante os treinamentos. Já a MobileNet apresentou oscilações nos seus valores de *Loss* durante os treinamentos, mas isso não a impediu de também obter bons resultados.

A CNN desenvolvida, intitulada OakMoonNet, apresentou desempenho igual ou superior se comparado a AlexNet e MobileNet em todos os testes realizados, tanto em acurácia, precisão, recall e *F1-Score*. Seu tempo de treinamento, em todos os casos, foi expressivamente menor. Portanto, ela se mostrou uma CNN Eficiente e adequada para ser utilizada em microcontroladores, como o *Raspberry Pi*.

A interface visual e o software desenvolvido apresentaram facilidade em seu uso, sendo intuitivo e possibilitando o uso das CNNs mesmo por usuário com pouco conhecimento sobre o algoritmo. O *Classification fruit UFRRJ* possibilitada que o usuário somente indique o diretório de armazenamento das imagens e defina alguns *hiperparametros* que serão utilizados, o que dá ao programa um caráter de fácil usabilidade.

REFERÊNCIAS

ABEKOON, T.; SAJINDRA, H.; JAYAKODY, J. A. D. C. A.; SAMARAKOON, E. R. J.; RATHNAYAKE, U. Image processing techniques to identify tomato quality under market conditions. **Smart Agricultural Technology**, v. 7, n. 100433, p. 100433, 2024. ISSN 2772-3755. DOI: 10.1016/j.atech.2024.100433. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2772375524000388?via%3Dihub>. Acesso em: 10 abr. 2025.

ALI, M.; VLASKAMP, J. H. A.; NASSER EDDINY, N.; FALCONER, B.; ORAM, C. Technical development and socioeconomic implications of the Raspberry Pi as a learning tool in developing countries. *In*: COMPUTER SCIENCE AND ELECTRONIC ENGINEERING CONFERENCE, 5., 2013, Colchester. **Anais [...]**. Colchester: IEEE, 2013. p. 103–108

ALOMAR, K.; AYSEL, H. I.; CAI, X. Data Augmentation in Classification and Segmentation: A Survey and New Strategies. **Journal of Imaging**, v. 9, n. 2, p. 46, 2023. DOI: 10.3390/jimaging9020046. Disponível em: <https://www.mdpi.com/2313-433X/9/2/46>. Acesso em: 20 set. 2025.

ALZUBAIDI, L.; ZHANG, J.; HUMAIDI, A. J.; AL-DUJAILI, A.; DUAN, Y.; AL-SHAMMA, O.; SANTAMARÍA, J.; FADHEL, M. A.; AL-AMIDIE, M.; FARHAN, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. **Journal of Big Data**, v. 8, n. 1, p. 1–74, 2021. ISSN 2196-1115. DOI: 10.1186/s40537-021-00444-8. Disponível em: <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00444-8>. Acesso em: 07 out. 2023.

ASRINY, D. M.; RANI, S.; HIDAYATULLAH, A. F. Orange fruit images classification using convolutional neural networks. **IOP Conference Series: Materials Science and Engineering**, v. 803, n. 1, p. 012020, 2020. Trabalho apresentado no IOP Conference Series: Materials Science and Engineering, 2020, Yogyakarta.

BASFORD, P. J.; JOHNSTON, S. J.; PERKINS, C. S.; GARNOCK-JONES, T.; TSO, F. P.; PEZAROS, D.; MULLINS, R. D.; YONEKI, E.; SINGER, J.; COX, S. J. Performance analysis

of single board computer clusters. **Future Generation Computer Systems**, v. 102, p. 278–291, 2020. DOI: 10.1016/j.future.2019.07.040. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167739X1833142X>. Acesso em: 31 ago. 2025.

BRAGA, R. A., Jr; CONTADO, J. L.; DUCATTI, K. R.; SILVA, E. A. A. Analysis of seed vigor using the biospeckle laser technique. **AgriEngineering**, v. 7, n. 1, p. 3, 2024. ISSN 2624-7402. DOI: 10.3390/agriengineering7010003. Disponível em: <https://www.mdpi.com/2624-7402/7/1/3>. Acesso em: 05 mar. 2025.

BRIGATO, L.; IOCCHI, L. A close look at deep learning with small data. **Proceedings of the International Conference on Pattern Recognition**, Milan, p. 2490–2497, 2021. Supl. 1. Trabalho apresentado no 25º International Conference on Pattern Recognition, 2020, Milan.

CEAGESP – Centro de Qualidade em Horticultura. **Classificação do tomate**. São Paulo: ABH, 2003. Disponível em: <https://ceagesp.gov.br/wp-content/uploads/2015/07/tomate.pdf>. Acesso em: 05 out. 2023.

CEAGESP – Companhia de Entrepósitos e Armazéns Gerais de São Paulo. **Normas para classificação de citros de mesa**. São Paulo: CEAGESP, 2011. 12 p. Disponível em: <https://ceagesp.gov.br/wp-content/uploads/2015/07/citros.pdf>. Acesso em: 05 out. 2023.

CHAKRABORTY, S. K.; SUBEESH, A.; DUBEY, K.; JAT, D.; CHANDEL, N. S.; POTDAR, R.; RAO, N. R. N. V. G.; KUMAR, D. Development of an optimally designed real-time automatic citrus fruit grading–sorting machine leveraging computer vision-based adaptive deep learning model. **Engineering Applications of Artificial Intelligence**, v. 120, n. 105826, p. 105826, 2023. DOI: 10.1016/j.engappai.2023.105826. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0952197623000106?via%3Dihub>. Acesso em: 15 out. 2023.

CHATURVEDI, A.; SHARMA, S.; JANGHEL, R. R. Detection of external defects in tomatoes using deep learning. **Journal of Ambient Intelligence and Humanized Computing**, v. 14, n.

3, p. 2709–2721, 2023. DOI: 10.1007/s12652-023-04514-y. Disponível em: <https://link.springer.com/article/10.1007/s12652-023-04514-y>. Acesso em: 18 nov. 2024.

CHEN, W.; LU, S.; LIU, B.; LI, G.; QIAN, T. Detecting citrus in orchard environment by using improved YOLOv4. **Scientific Programming**, v. 2020, p. 1–13, 2020. DOI: 10.1155/2020/8859237. Disponível em: <https://onlinelibrary.wiley.com/doi/10.1155/2020/8859237>. Acesso em: 09 jan. 2025.

COSTA, A. G.; BRAGA Jr., R. A.; VILAS BOAS, E. V. B.; RISSO, M. Early prediction of internal bruising in potatoes by biospeckle laser technique. **African Journal of Agricultural Research**, v. 13, n. 14, p. 691–697, 2018. DOI: 10.5897/AJAR2017.12959. Disponível em: <https://academicjournals.org/journal/AJAR/article-abstract/6C3128356571>. Acesso em: 20 mar. 2025

CUI, M.; JIANG, Q.; LI, N.; XUE, X. Research on strawberry maturity detection technology based on improved YOLOv4. **Journal of Physics: Conference Series**, v. 2138, n. 1, p. 012012, 2021. DOI: 10.1088/1742-6596/2138/1/012012. Disponível em: <https://iopscience.iop.org/article/10.1088/1742-6596/2138/1/012012>. Acesso em: 18 nov. 2023.

DAS, P.; YADAV, J. P. S. Automated tomato maturity grading system using CNN. **Proceedings of the International Conference on Smart Electronics and Communication**, Tirunelveli, p. 136–142, 2020. Trabalho apresentado no 2020 International Conference on Smart Electronics and Communication (ICOSEC), 2020, Tirunelveli.

DE LUNA, R. G.; DADIOS, E. P.; BANDALA, A. A.; VICERRA, R. R. P. Tomato fruit image dataset for deep transfer learning-based defect detection. **Proceedings of the IEEE International Conference on Cybernetics and Intelligent Systems and Robotics, Automation and Mechatronics**, Singapore, p. 356–361, 2020. Trabalho apresentado no 2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), 2019, Singapore.

ELARABY, A.; HAMDY, W.; ALANAZI, S. Classification of citrus diseases using optimization deep learning approach. **Computational Intelligence and Neuroscience**, v. 2022, p. 9153207, 2022. DOI: 10.1155/2022/9153207. Disponível em: <https://onlinelibrary.wiley.com/doi/10.1155/2022/9153207>. Acesso em: 09 jan. 2025.

ERLINA, T.; FIKRI, M. Y. Yolo Algorithm-Based Visitor Detection System for Small Retail Stores Using Single Board Computer. **J. Appl. Eng. Technol. Sci**, v. 4, n. 2, p. 908–920, 2023. DOI: 10.37385/jaets.v4i2.1872. Disponível em: <https://journal.yrpioku.com/index.php/jaets/article/view/1872>. Acesso em: 31 ago. 2025.

FANG, W.; LOVE, P. E. D.; LUO, H.; DING, L. Computer vision for behaviour-based safety in construction: A review and future directions. **Advanced Engineering Informatics**, v. 43, n. 100980, p. 100980, 2020. DOI: 10.1016/j.aei.2019.100980. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S1474034619305531?via%3Dihub>. Acesso em: 20 set. 2023.

FENG, J.; WANG, Z.; WANG, S.; TIAN, S.; XU, H. MSDD-YOLOX: An enhanced YOLOX for real-time surface defect detection of oranges by type. **European Journal of Agronomy: The Journal of the European Society for Agronomy**, v. 149, n. 126918, p. 126918, 2023. DOI: 10.1016/j.eja.2023.126918. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S1161030123001867?via%3Dihub>. Acesso em: 09 jan. 2025.

GONZALEZ-HUITRON, V.; LEÓN-BORGES, J. A.; RODRIGUEZ-MATA, A. E.; AMABILIS-SOSA, L. E.; RAMÍREZ-PEREDA, B.; RODRIGUEZ, H. 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. DOI: 10.1016/j.compag.2020.105951. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168169920331562>. Acesso em: 31 ago. 2025.

GOODMAN, J. W. **Statistical properties of laser speckle patterns**. In: DAINITY, J. C. (ed.). *Laser speckle and related phenomena*. Berlin Heidelberg: Springer-Verlag, 1975. p. 9–75.

Disponível em: https://link.springer.com/content/pdf/10.1007/978-3-662-43205-1_2.pdf.
Acesso em: 10 mar. 2025.

GOYAL, K.; KUMAR, P.; VERMA, K. AI-based fruit identification and quality detection system. **Multimedia Tools and Applications**, v. 82, n. 16, p. 24573–24604, 2023. DOI: 10.1007/s11042-022-14188-x. Disponível em: <https://link.springer.com/article/10.1007/s11042-022-14188-x>. Acesso em: 15 abr. 2024.

GRIGORIEV, S. N.; MARTINOV, G. M. An ARM-based multi-channel CNC solution for multi-tasking turning and milling machines. **Procedia CIRP**, v. 46, p. 525–528, 2016. DOI: 10.1016/j.procir.2016.04.036. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2212827116301913>. Acesso em: 31 ago. 2025.

GU, J.; WANG, Z.; KUEN, J.; MA, L.; SHAHROUDY, A.; SHUAI, B.; LIU, T.; WANG, X.; WANG, G.; CAI, J.; CHEN, T. Recent advances in convolutional neural networks. **Pattern Recognition**, v. 77, p. 354–377, 2018. DOI: 10.1016/j.patcog.2017.10.013. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0031320317304120?via%3Dihub>. Acesso em: 13 jan. 2024.

HABARAGAMUWA, H.; OGAWA, Y.; SUZUKI, T.; SHIIGI, T.; ONO, M.; KONDO, N. Detecting greenhouse strawberries (mature and immature), using deep convolutional neural network. **Engineering in Agriculture, Environment and Food**, v. 11, n. 3, p. 127–138, 2018. DOI: 10.1016/j.eaef.2018.03.001. Disponível em: https://www.jstage.jst.go.jp/article/eaef/11/3/11_127/_article/-char/en. Acesso em: 10 nov. 2023.

HADIPOUR-ROKNI, R.; ASKARI ASLI-ARDEH, E.; JAHANBAKHSHI, A.; ESMAEILI-AFKAROTI, I.; SABZI, S. Intelligent detection of citrus fruit pests using machine vision system and convolutional neural network through transfer learning technique. **Computers in Biology and Medicine**, v. 155, n. 106611, p. 106611, 2023. DOI: 10.1016/j.combiomed.2023.106611. Disponível em:

<https://www.sciencedirect.com/science/article/abs/pii/S0010482523000768?via%3Dihub>.

Acesso em: 09 jan. 2025.

HARDKERNEL CO. LTD. ODROID—GO SUPER. Hardkernel Co. Ltd., 2021. Disponível em: <https://www.hardkernel.com>. Acesso em: 31 ago. 2025

HENDRAWAN, Y.; ROHMATULLOH, B.; PRAKOSO, I.; LIANA, V.; FAUZY, M. R.; DAMAYANTI, R.; HERMANTO, M. B.; AL RIZA, D. F.; SANDRA. Classification of large green chilli maturity using deep learning. IOP Conference Series: Earth and Environmental Science, v. 924, n. 1, p. 012009, 2021. DOI: 10.1088/1755-1315/924/1/012009. Disponível em: <https://iopscience.iop.org/article/10.1088/1755-1315/924/1/012009>. Acesso em: 10 out. 2023.

HOWARD, A. G.; ZHU, M.; CHEN, B.; KALENICHENKO, D.; WANG, W.; WEYAND, T.; ANDREETTO, M.; ADAM, H. **Mobilenets: Efficient convolutional neural networks for mobile vision applications**. arXiv preprint arXiv:1704.04861, 2017. Disponível em: <https://arxiv.org/abs/1704.04861>. Acesso em: 15 set. 2023.

HU, C.; LIU, X.; PAN, Z.; LI, P. Automatic detection of single ripe tomato on plant combining faster R-CNN and intuitionistic fuzzy set. **IEEE Access: Practical Innovations, Open Solutions**, v. 7, p. 154683–154696, 2019. DOI: 10.1109/ACCESS.2019.2949343. Disponível em: <https://ieeexplore.ieee.org/document/8882212>. Acesso em: 03 dez. 2024.

HU, D.; JIA, T.; SUN, X.; ZHOU, T.; HUANG, Y.; SUN, Z.; ZHANG, C.; SUN, T.; ZHOU, G. Applications of optical property measurement for quality evaluation of agri-food products: a review. *Critical Reviews in Food Science and Nutrition*, v. 64, n. 33, p. 12599–12619, 2024. DOI: 10.1080/10408398.2023.225526. Disponível em: <https://www.tandfonline.com/doi/full/10.1080/10408398.2023.2255260>. Acesso em: 09 jan. 2024.

HUANG, Y.-P.; WANG, T.-H.; BASANTA, H. Using fuzzy mask R-CNN model to automatically identify tomato ripeness. **IEEE Access: Practical Innovations, Open Solutions**,

v. 8, p. 207672–207682, 2020. DOI: 10.1109/ACCESS.2020.3038184. Disponível em: <https://ieeexplore.ieee.org/document/9260129>. Acesso em: 18 nov. 2024.

HUBEL, D. H.; WIESEL, T. N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. **The Journal of Physiology**, v. 160, n. 1, p. 106–154, 1962. DOI: 10.1113/jphysiol.1962.sp006837. Disponível em: <https://physoc.onlinelibrary.wiley.com/doi/10.1113/jphysiol.1962.sp006837>. Acesso em: 06 out. 2023.

IBGE – Instituto Brasileiro de Geografia e Estatística. **Estatística Mensal da Produção Agrícola. Levantamento Sistemático da Produção agrícola**. 2024. Disponível em: <https://www.ibge.gov.br/estatisticas/economicas/agricultura-e-pecuaria/9201-levantamento-sistemico-da-producao-agricola.html>. Acesso em: 15 fev. 2024

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. **Proceedings of the International Conference on Machine Learning**, Lille, p. 448–456, 2015. Trabalho apresentado no International Conference on Machine Learning, 2015, Lille.

KAISER, L.; GOMEZ, A. N.; CHOLLET, F. **Depthwise separable convolutions for neural machine translation**. arXiv preprint arXiv:1706.03059, 2017. Disponível em: <https://arxiv.org/abs/1706.03059>. Acesso em: 04 fev. 2024.

KALER, N.; BHATIA, V.; MISHRA, A. K. Deep learning-based robust analysis of laser bio-speckle data for detection of fungal-infected soybean seeds. **IEEE Access: Practical Innovations, Open Solutions**, v. 11, p. 89331–89348, 2023. DOI: 10.1109/ACCESS.2023.3305273. Disponível em: <https://ieeexplore.ieee.org/document/10216973>. Acesso em: 13 mar. 2025.

KHAN, M. R. Advances in architectures for deep learning: A thorough examination of present trends. **Journal of Artificial Intelligence General Science**, v. 1, n. 1, p. 24–30, 2024. DOI: 10.60087/jaigs.v1i1.p30. Disponível em: <https://jaigs.org/index.php/JAIGS/article/view/14>. Acesso em: 03 mai. 2025.

KHATUN, T.; RAZZAK, A.; ISLAM, M. S.; UDDIN, M. S. et al. An extensive real-world in field tomato image dataset involving maturity classification and recognition of fresh and defect tomatoes. **Data in Brief**, v. 51, n. 109688, p. 109688, 2023. DOI: 10.1016/j.dib.2023.109688. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/37920387/>. Acesso em: 03 nov. 2024.

KHATUN, M.; ALI, M. F.; TURZO, N. A.; NINE, J.; SARKER, P. Fruits classification using convolutional neural network. **GRD Journals – Global Research and Development Journal for Engineering**, v. 5, n. 8, p. 1–6, 2020. DOI: 10.13140/RG.2.2.16997.22244. Disponível em: <https://www.academia.edu/download/64177538/GRDJEV05I080015.pdf>. Acesso em: 18 nov. 2024.

KOPIER, A. A.; SILVA, V. N. A. L. da; OLIVEIRA, L. A. A. de; LINDEN, R.; SILVA, L. R. A. de A.; FONSECA, B. L. da C. Redes Neurais Artificiais e suas aplicações no setor elétrico. **Revista de Engenharias da Faculdade Salesiana**, v. 1, n. 9, p. 27–33, 2019. Disponível em: https://www.fsma.edu.br/RESA/Edicao9/FSMA_RESA_2019_1_04.pdf. Acesso em: 07 set 2023.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet classification with deep convolutional neural networks. **Communications of the ACM**, v. 60, n. 6, p. 84–90, 2017. DOI: 10.1145/3065386. Disponível em: <https://dl.acm.org/doi/10.1145/3065386>. Acesso em: 02 fev. 2024.

KUKREJA, V.; DHIMAN, P. A deep neural network based disease detection scheme for citrus fruits. **Proceedings of the International Conference on Smart Electronics and Communication**, Tirunelveli, p. 97–101, 2020. Trabalho apresentado no 2020 International Conference on Smart Electronics and Communication (ICOSEC), 2020, Tirunelveli.

KUMAR, A.; JOSHI, R. C.; DUTTA, M. K.; JONAK, M. BURGET, R. Fruit-CNN: An efficient deep learning-based fruit classification and quality assessment for precision agriculture. **Proceedings of the International Congress on Ultra Modern Telecommunications and Control Systems and Workshops**, Moscow, p. 60–65, 2021.

Trabalho apresentado no 13th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2021, Moscow.

LI, G.; ZHANG, M.; LI, J.; LV, F.; TONG, G. Efficient densely connected convolutional neural networks. *Pattern Recognition*, v. 109, n. 107610, p. 107610, 2021. DOI: 10.1016/j.patcog.2020.107610. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0031320320304131?via%3Dihub>. Acesso em: 08 out. 2023.

LIMA, A. F.; SILVA, E. G. DE A.; IWATA, B. DE F. Agriculturas e agricultura familiar no Brasil: uma revisão de literatura. *Retratos de Assentamentos*, v. 22, n. 1, p. 50–68, 2019. DOI: 10.25059/2527-2594/retratosdeassentamentos/2019.v22i1.332. Disponível em: <https://retratosdeassentamentos.com/index.php/retratos/article/view/332>. Acesso em: 12 out. 2023.

LINDNER, T.; WYRWAL, D.; BIAŁEK, M.; NOWAK, P. Face recognition system based on a single-board computer. In: *INTERNATIONAL CONFERENCE MECHATRONIC SYSTEMS AND MATERIALS*, 2020, Opole. *Anais [...]*. Opole: IEEE, 2020. p. 1–6

LU, J.; CHEN, W.; LAN, Y.; QIU, X.; HUANG, J.; LUO, H. Design of citrus peel defect and fruit morphology detection method based on machine vision. *Computers and Electronics in Agriculture*, v. 219, n. 108721, p. 108721, 2024. DOI: 10.1016/j.compag.2024.108721. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0168169924001121?via%3Dihub>. Acesso em: 09 jan. 2025.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, v. 5, n. 4, p. 115–133, 1943. DOI: 10.1007/BF02478259. Disponível em: <https://link.springer.com/article/10.1007/BF02478259>. Acesso em: 04 set. 2023.

MINSKY, M.; PAPERT, S. *An introduction to computational geometry*. Cambridge: TIASS, HIT, v. 479, n. 480, p. 104, 1969.

MUNDIM FILHO, A. C.; ALVARENGA, C.; CARNEIRO, M. G. Colher ou não colher? Uma rede neural convolucional para apoiar cafeicultores na decisão da colheita. **Anais Estendidos do XVII Simpósio Brasileiro de Sistemas de Informação**, p. 57–60, 2021. Trabalho apresentado no XVII Simpósio Brasileiro de Sistemas de Informação, 2021, Uberlândia.

OLIVEIRA, C. T. de. **Monitoramento da maturação dos frutos e de doenças do cafeeiro utilizando modelos de deep learning**. Dissertação (Mestrado em Engenharia Agrícola) – Programa de Pós-Graduação em Engenharia Agrícola, Universidade Federal de Viçosa, Viçosa, 2021.

OULOMARA, G.; TRIBILLON, J.; DUVERNOY, J. Biological Activity Measurement on Botanical Specimen Surfaces Using a Temporal Decorrelation Effect of Laser Speckle. **Journal of Modern Optics**, v. 36, n. 2, p. 165–179, 1989. DOI: 10.1080/09500348914550221. Disponível em: <https://www.tandfonline.com/doi/abs/10.1080/09500348914550221>. Acesso em: 1 out. 2025.

PALUMBO, M.; CEFOLA, M.; PACE, B.; ATTOLICO, G.; COLELLI, G. Computer vision system based on conventional imaging for non-destructively evaluating quality attributes in fresh and packaged fruit and vegetables. **Postharvest Biology and Technology**, v. 200, n. 112332, p. 112332, 2023. DOI: 10.1016/j.postharvbio.2023.112332. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0925521423000935?via%3Dihub>. Acesso em: 08 set. 2023.

PARK, H. S.; AHN, Y. C. Method for measuring blood flow and depth of blood vessel based on laser speckle contrast imaging using 3D convolutional neural network: A preliminary study. **Optics & Laser Technology**, v. 179, n. 111367, p. 111367, 2024. DOI: 10.1016/j.optlastec.2024.111367. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0030399224008259?via%3Dihub>. Acesso em: 20 set. 2025.

PAVARIN, F. F. A.; FRACAROLLI, J. A.; FALCÃO, A. X. Biospeckle laser for assessing tomatoes ripeness indexes. **Revista Ciência Agronômica**, v. 55, p. e20207677, 2024. DOI:

10.5935/1806-6690.20240022. Disponível em:
<https://periodicos.ufc.br/revistacienciaagronomica/article/view/91892>. Acesso em: 09 mar. 2025.

PANDE, A.; MUNOT, M.; SREEEMATHY, R.; BAKARE, R. V. An efficient approach to fruit classification and grading using deep convolutional neural network. *In: IEEE 5th International Conference for Convergence in Technology (I2CT)*, 2019, Pune. **Anais [...]** Pune: IEEE, 2019.

PHAN, Q.-H.; NGUYEN, V.-T.; LIEN, C.-H.; DUONG, T.-P.; HOU, M. T.-K.; LE, N.-B. Classification of tomato fruit using YOLOv5 and convolutional neural network models. **Plants**, v. 12, n. 4, 2023. DOI: 10.3390/plants12040790. Disponível em: <https://www.mdpi.com/2223-7747/12/4/790>. Acesso em: 20 mar. 2024.

QURESHI, R.; MEHBOOB, S. H.; AAMIR, M. Sustainable green fog computing for smart agriculture. *Wireless Personal Communications*, v. 121, n. 2, p. 1379–1390, 2021. DOI: 10.1007/s11277-021-09059-x. Disponível em: <https://link.springer.com/article/10.1007/s11277-021-09059-x>. Acesso em: 31 ago. 2025.

RABAL, H. J.; BRAGA JR, R. A. **Dynamic laser speckle and applications**. Boca Raton, FL: CRC Press, 2018.

RAHMAT, H.; WAHJUNI, S.; RAHMAWAN, H. Performance analysis of deep learning-based object detectors on raspberry pi for detecting melon leaf abnormality. *Int. J. Adv. Sci. Eng. Inf. Technol.*, v. 12, n. 2, p. 572–579, 2022. DOI: 10.18517/ijaseit.12.2.13801. Disponível em:
https://www.researchgate.net/publication/360294751_Performance_Analysis_of_Deep_Learning-based_Object_Detectors_on_Raspberry_Pi_for_Detecting_Melon_Leaf_Abnormality. Acesso em: 31 ago. 2025.

RASPBERRY PI FOUNDATION. **Raspberry Pi 400**. Raspberry Pi Foundation in association with Broadcom, 2021. Disponível em: <https://www.raspberrypi.org/>. Acesso em: 31 ago. 2025.

RAUBER, T. W. **Redes neurais artificiais**. Universidade Federal do Espírito Santo, 2005.

RICHARDSON, M.; WALLACE, S. **Getting Started with Raspberry Pi: Electronic Projects with Python, Scratch, and Linux**. Sebastopol, CA: Maker Media, Inc., 2014.

RUSSEL, S.; NORVIG, P. **Artificial intelligence: a modern approach**. 4. ed. New Jersey: Prentice Hall, 2020.

ROSA, R. P. Método de classificação de pragas por meio de rede neural convolucional profunda. 2018. Dissertação (Mestrado em Computação Aplicada) — Programa de Pós-Graduação em Ciência Aplicada, Universidade Estadual de Ponta Grossa, Ponta Grossa, 2018.

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological Review**, v. 65, n. 6, p. 386–408, 1958. DOI: 10.1037/h0042519. Disponível em: <https://psycnet.apa.org/doiLanding?doi=10.1037%2Fh0042519>. Acesso em: 15 nov. 2023.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. **Learning internal representations by error propagation**. In: MCCLELLAND, J. L.; RUMELHART, D. E. (org.). *Parallel distributed processing: explorations in the microstructure of cognition*. v. 1. Cambridge, MA: MIT Press, 1985. p. 318–362.

SANTOS, D. M. **Redes neurais convolucionais para classificar a maturação de frutos de café**. 2022. Dissertação (Mestrado em Agricultura e Informações Geoespaciais) — Instituto de Ciências Agrárias, Universidade Federal de Uberlândia, Monte Carmelo, MG, 2022.

SHIVASHANKAR, M.; PANDEY, M.; ZAKWAN, M. Estimation of settling velocity using generalized reduced gradient (GRG) and hybrid generalized reduced gradient–genetic algorithm (hybrid GRG-GA). **Acta Geophysica**, v. 70, n. 5, p. 2487–2497, 2022. DOI: 10.1007/s11600-021-00706-2. Disponível em: <https://link.springer.com/article/10.1007/s11600-021-00706-2>. Acesso em: 19 out. 2023.

TENSORFLOW. **Tensorflow**. 2024. Disponível em: <https://www.tensorflow.org/>. Acesso em: 1 abr. 2024.

THAKUR, P. S.; KREJCAR, O.; BHATIA, V.; PRAKASH, S. Deep learning based processing framework for spatio-temporal analysis and classification of laser biospeckle data. **Optics and Laser Technology**, v. 169, n. 110138, p. 110138, 2024. DOI: 10.1016/j.optlastec.2023.110138. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0030399223010319?via%3Dihub>. Acesso em: 20 mar. 2025.

TSAI, F.-T.; NGUYEN, V.-T.; DUONG, T.-P.; PHAN, Q.-H.; LIEN, C.-H. Tomato fruit detection using modified Yolov5m model with convolutional neural networks. **Plants**, v. 12, n. 17, 2023. DOI: 10.3390/plants12173067. Disponível em: <https://www.mdpi.com/2223-7747/12/17/3067>. Acesso em: 03 jan. 2024.

WANG, T.; LU, C.; YANG, M.; HONG, F.; LIU, C. A hybrid method for heartbeat classification via convolutional neural networks, multilayer perceptrons and focal loss. **PeerJ Computer Science**, v. 6, n. e324, p. e324, 2020. DOI: 10.7717/peerj-cs.324. Disponível em: <https://peerj.com/articles/cs-324/>. Acesso em: 13 dez. 2023.

WANG, Y. X.; RAMANAN, D.; HEBERT, M. Growing a brain: Fine-tuning by increasing model capacity. **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**, Honolulu, p. 2471–2480, 2017. Trabalho apresentado no IEEE Conference on Computer Vision and Pattern Recognition, 2017, Honolulu.

WANG, Z.; LING, Y.; WANG, X.; MENG, D.; NIE, L.; AN, G.; WANG, X. An improved Faster R-CNN model for multi-object tomato maturity detection in complex scenarios. **Ecological Informatics**, v. 72, n. 101886, p. 101886, 2022. DOI: 10.1016/j.ecoinf.2022.101886. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S1574954122003363?via%3Dihub>. Acesso em: 19 ago. 2024.

WICHMANN, F. A.; SHARPE, L. T.; GEGENFURTNER, K. R. The contributions of color to recognition memory for natural scenes. **Journal of Experimental Psychology: Learning, Memory, and Cognition**, v. 28, n. 3, p. 509–520, 2002. DOI: 10.1037/0278-7393.28.3.509. Disponível em: <https://psycnet.apa.org/record/2002-12651-010>. Acesso em: 14 dez. 2023.

XIAO, X.; WANG, Y.; JIANG, Y. Review of research advances in fruit and vegetable harvesting robots. **Journal of Electrical Engineering and Technology**, v. 19, n. 1, p. 773–789, 2024. DOI: 10.1007/s42835-023-01596-8. Disponível em: <https://link.springer.com/article/10.1007/s42835-023-01596-8>. Acesso em: 15 out. 2023.

WIDROW, B.; HOFF, M. E. **Adaptive switching circuits**. In: IRE WESCON CONVENTION RECORD, 1960, Los Angeles, CA. p. 96–104.

YANG, S.; ZHANG, Z.; LI, J.; WANG, X.; ZHOU, B. Early detection of fungal infection in citrus using biospeckle imaging. **Computers and Electronics in Agriculture**, v. 225, p. 109293, 2024. DOI: 10.1016/j.compag.2024.109293. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0168169924006847?via%3Dihub>. Acesso em: 20 mar. 2025.

YAU, H. **Learning BeagleBone**. Birmingham, UK: Packt Publishing Ltd, 2014.

ZDUNEK, A.; KADLEC, K.; BORKOWSKI, A.; LUKASZEWSKA, A. The biospeckle method for the investigation of agricultural crops: A review. **Optics and Lasers in Engineering**, v. 52, p. 276–285, 2014. DOI: 10.1016/j.optlaseng.2013.06.017. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0143816613002030?via%3Dihub>. Acesso em: 20 mar. 2025.

ZU, L.; ZHAO, Y.; LIU, J.; SU, F.; ZHANG, Y.; LIU, P. Detection and segmentation of mature green tomatoes based on mask R-CNN with automatic image acquisition approach. **Sensors (Basel, Switzerland)**, v. 21, n. 23, p. 7842, 2021. DOI: 10.3390/s21237842. Disponível em: <https://www.mdpi.com/1424-8220/21/23/7842>. Acesso em: 09 out. 2024.