

UNIVERSIDADE PAULISTA  
PROGRAMA DE MESTRADO EM ENGENHARIA DE PRODUÇÃO

**IDENTIFICAÇÃO DE VARIEDADES DE PIMENTAS COM  
YOLOV8: UMA APLICAÇÃO DE APRENDIZADO PROFUNDO  
PARA O SETOR AGRIFOOD**

Exame de Defesa de Dissertação  
apresentada ao Programa de Pós-  
Graduação em Engenharia de  
Produção da Universidade Paulista  
– UNIP, para obtenção do título de  
Mestre em Engenharia de  
Produção.

**MADALENA DE OLIVEIRA BARBOSA**

**SÃO PAULO**

**2025**

# **IDENTIFICAÇÃO DE VARIEDADES DE PIMENTAS COM YOLOV8: UMA APLICAÇÃO DE APRENDIZADO PROFUNDO PARA O SETOR AGRIFOOD**

Exame de defesa de dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Paulista – UNIP, para obtenção do título de Mestre em Engenharia de Produção.

Orientador: Prof. Dr. Marcelo Tsugui Okano

Área de Conhecimento: Engenharia de Produção

Área de Concentração: Gestão de Sistemas de Operação

Linha de Pesquisa: Redes de Empresas e Planejamento da Produção

Projeto de Pesquisa: Gestão da Produção e Inovação na Agricultura, Indústria e Serviços

**MADALENA DE OLIVEIRA BARBOSA**

**SÃO PAULO**

**2025**

## FICHA CATALOGRÁFICA

Barbosa, Madalena de Oliveira.

Identificação de variedades de pimentas com YOLOv8:  
uma aplicação de aprendizado profundo para o setor *agrifood* /  
Madalena de Oliveira Barbosa. – 2025.

106 f. : il. color. + CD-ROM.

Dissertação de Mestrado Apresentada ao Programa de  
Pós- Graduação em Engenharia de Produção da Universidade  
Paulista, São Paulo, 2025.

Área de concentração: Gestão de Sistemas de Operação.  
Orientador: Prof. Dr. Marcelo Tsugui Okano.

1. Agricultura 4.0. 2. Aumento de dados. 3. Identificação de  
pimentas. 4. Visão computacional. 5. YOLOv8. I. Okano,  
Marcelo Tsugui Okano (orientador). II. Título.

## DEDICATÓRIA

À minha mãe Rosa, por sua força.  
Aos meus filhos, Rafaela e Pedro, por serem meu maior orgulho.  
Tudo é por vocês.

## **AGRADECIMENTOS**

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e ao Programa de Suporte à Pós-Graduação de Instituições de Ensino Particulares (PROSUP) pelo apoio concedido.

Agradeço ao meu orientador, Prof. Dr. Marcelo Tsuguio Okano, pela orientação e contribuições essenciais, e à Prof.<sup>a</sup> Dra. Irenilza de Alencar Nääs, pela generosa contribuição em pesquisa e em escrita acadêmica.

Agradeço também ao Prof. Dr. Mario Mollo, por sua disponibilidade em participar da banca de qualificação, por seus apontamentos e sugestões tão valiosos.

Agradeço ainda a Deus, à minha mãe, a meus irmãos, a meus filhos - Rafaela e Pedro - e aos amigos todos pelo apoio, e pelo incentivo ao longo desta trajetória.

## LISTA DE FIGURAS

Figura 1- Relação entre objetivos específicos e resultados da pesquisa .....	19
Figura 2 - Exemplo de imagens do <i>dataset</i> por classe (8 variedades de pimentas) .....	36
Figura 3 - Exemplo de anotação com <i>bounding boxes</i> no <i>Roboflow</i> .....	37
Figura 4 - Exemplo de técnicas de <i>data augmentation</i> aplicadas ao dataset.....	38
Figura 5 - Fluxograma de preparação do <i>dataset</i> e seu particionamento .....	39
Figura 6 - Fluxograma de treinamento do modelo <i>YOLO8m</i> no <i>Google Colab Pro</i> .....	42
Figura 7- Representação das métricas de avaliação (Precisão, <i>Recall</i> , <i>mAP50</i> e <i>mAP95</i> ) .....	43
Figura 8 - Tela inicial de identificação de pimentas do <i>App Pimentas</i> .....	50
Figura 9 - Tela de <i>chat</i> de informações do <i>App Pimentas</i> (com exemplo para a pimenta <i>Habanero-Pepper</i> ).....	51

## LISTA DE TABELAS

**Tabela 1** - Parâmetros de treinamento do YOLOv8 com e sem *data augmentation*..... 41

## LISTA DE ABREVIAÇÕES

API	Interface de Programação de Aplicações ( <i>Application Programming Interface</i> )
CEAGESP	Companhia de Entrepósitos e Armazéns Gerais de São Paulo
CNN	Rede Neural Convolucional ( <i>Convolutional Neural Network</i> )
CPU	Unidade Central de Processamento ( <i>Central Processing Unit</i> )
CSS	<i>Cascading Style Sheets.</i>
DSR	<i>Design Science Research</i>
GPU	Unidade de Processamento Gráfico ( <i>Graphics Processing Unit</i> )
HTML	<i>HyperText Markup Language.</i>
IA	Inteligência Artificial ( <i>Artificial Intelligence</i> )
IoT	Internet das Coisas ( <i>Internet of Things</i> )
IoU	Intersecção sobre União ( <i>Intersection over Union</i> )
INPI	Instituto Nacional da Propriedade Industrial.
JPEG	Grupo Conjunto de Especialistas em Fotografia ( <i>Joint Photographic Experts Group</i> )
JSON	Notação de Objetos JavaScript ( <i>JavaScript Object Notation</i> )
LLM	Large Language Models

mAP	Precisão Média ( <i>mean Average Precision</i> )
mAP50	Precisão Média com IoU $\geq 50\%$
mAP95	Precisão Média com IoU de 50% a 95%
ODS	Objetivos de Desenvolvimento Sustentável ( <i>Sustainable Development Goals</i> )
ONU	Organização das Nações Unidas
PaaS	Platform as a Service
PWA	<i>Progressive Web App</i>
PIL	Biblioteca de Imagens Python ( <i>Python Imaging Library</i> )
RAG	Retrieval-Augmented Generation
REST	Transferência de Estado Representacional ( <i>Representational State Transfer</i> )
SHU	<i>Scoville Heat Units</i> .
YOLO	Você Só Olha Uma Vez ( <i>You Only Look Once</i> )
YOLOv8m	Configuração 'médio' do modelo YOLOv8

## RESUMO

A crescente demanda por eficiência e padronização no setor agroalimentar tem impulsionado a adoção de tecnologias da Agricultura 4.0. Nesse contexto, a aplicação de técnicas de visão computacional e aprendizado profundo desponta como alternativa estratégica para modernizar cadeias agroalimentares. Este estudo avaliou a aplicação da arquitetura YOLOv8 na identificação automática de variedades de pimentas (*Capsicum spp.*) com o desenvolvimento adicional de um aplicativo móvel (PWA) que integra o modelo treinado, oferecendo suporte interativo por meio de um *chatbot*. Para tanto, foi construído um *dataset* inicial de 1.476 imagens coletadas na CEAGESP e em repositórios públicos, anotadas e expandidas por meio de técnicas de *data augmentation*, o que totaliza 3.964 imagens. O modelo foi treinado em dois cenários (com e sem *augmentation*) e avaliado pelas métricas *Box Precision*, *Recall*, mAP50 e mAP95. Os resultados evidenciaram que o uso de *augmentation* elevou a robustez do modelo, alcançando um *mAP50* de 0,694 e mitigando riscos de *overfitting*. Além do desempenho experimental, a pesquisa apresenta a implementação funcional de um aplicativo móvel capaz de classificar pimentas em tempo real, integrando informações adicionais sobre pungência e usos culinários.

Palavras-chave: Agricultura 4.0; Aumento de dados; Identificação de Pimentas; Visão Computacional; YOLOv8.

## ABSTRACT

The growing demand for efficiency and standardization in the agri-food sector has accelerated the adoption of Agriculture 4.0 technologies. Within this context, computer vision and deep learning have emerged as strategic tools to modernize food supply chains. This study evaluated the application of the YOLOv8 architecture for the automatic identification of pepper varieties (*Capsicum spp.*) and developed and implemented a mobile application (PWA) that integrates the trained model, providing interactive support through a chatbot. To this end, an initial dataset of 1,476 images was collected from CEAGESP and public repositories, annotated, and expanded using data augmentation techniques, resulting in 3,964 images. The model was trained under two scenarios (with and without augmentation) and evaluated using Precision, Recall, mAP50, and mAP95 metrics. Results demonstrated that augmentation significantly improved model robustness, achieving a mAP50 of 0.694 and reducing the risk of overfitting. Beyond experimental outcomes, this research presents the functional implementation of a mobile application capable of classifying peppers in real-time while providing additional information on pungency and culinary applications..

Keywords: Agriculture 4.0; Computer Vision; Data augmentation; Pepper varieties classification; YOLOv8.

## UTILIDADE

A utilidade desta dissertação está em disponibilizar um conjunto de dados e um modelo de classificação de pimentas, modelo que pode apoiar pesquisas futuras e aplicações práticas na Agricultura 4.0. O estudo pode contribuir para maior padronização em mercados atacadistas, para redução de perdas e para alinhamento às metas dos Objetivos de Desenvolvimento Sustentável. Além disso, esta pesquisa se alinha a compromissos globais estabelecidos pela Organização das Nações Unidas (ONU) por meio dos Objetivos de Desenvolvimento Sustentável (ODS). A proposta de automatizar a identificação de variedades de pimentas contribui diretamente para o ODS 2: Fome Zero e Agricultura Sustentável, ao apoiar a modernização da produção agrícola com maior eficiência e qualidade. Também está relacionada ao ODS 9: Indústria, Inovação e Infraestrutura, por promover a adoção de tecnologias digitais emergentes, como visão computacional e inteligência artificial, no setor agroalimentar. Finalmente, colabora com o ODS 12: Consumo e Produção Responsáveis, ao fortalecer mecanismos de rastreabilidade e reduzir perdas decorrentes de falhas humanas na classificação de produtos (ONU, 2015).

## SUMÁRIO

CAPÍTULO I .....	16
1      CONSIDERAÇÕES INICIAIS .....	16
1.1    Introdução .....	16
1.2    Justificativa .....	17
1.3    Objetivos .....	18
1.3.1    Objetivo Geral .....	18
1.3.2    Objetivos Específicos .....	18
1.4    Metodologia Resumida .....	18
1.5    Organização estrutural da pesquisa .....	18
CAPÍTULO II .....	20
2      REFERENCIAL TEÓRICO .....	20
2.1    Agricultura 4.0 e contexto global .....	20
2.2    Visão computacional na Agricultura .....	21
2.3    Aprendizado Profundo e Redes Neurais Convolucionais (CNNs) .....	23
2.4    Modelos de Detecção de Objetos .....	24
2.5    Evolução da Arquitetura YOLO .....	25
2.6    Pimentas do gênero Capsicum .....	27
2.7    Desafios na Classificação Automática de Variedades de <i>Capsicum</i> .....	28
CAPÍTULO III .....	31
3      METODOLOGIA .....	31
3.1    Caracterização da Pesquisa .....	31
3.2    Protocolo de Revisão de Literatura .....	31
3.3 <i>Design Science Research (DSR)</i> .....	32
3.3.1    Fase 1 – Identificação do problema e definição dos objetivos .....	32
3.3.2    Fase 2 – Análise de requisitos .....	33
3.3.3    Fase 3 – Design e desenvolvimento do artefato .....	33
3.3.4    Fase 4 – Demonstração .....	33
3.3.5    Fase 5 – Avaliação .....	34
3.3.6    Fase 6 – Comunicação .....	34
3.4    Coleta de dados .....	35

3.5	Preparação do <i>dataset</i> .....	36
3.6	Técnicas de <i>data augmentation</i> .....	38
3.7	Configuração Computacional .....	40
3.8	Procedimentos de treinamento.....	40
3.9	Métricas de Avaliação .....	43
3.10	Desenvolvimento do protótipo do aplicativo <i>App Pimentas</i> .....	44
3.10.1	Arquitetura geral e tentativa inicial com Kodular .....	45
3.10.2	<i>Backend</i> de inferência em <i>Python</i> , <i>FastAPI</i> e <i>ONNX</i> .....	46
3.10.3	<i>Interface web</i> e <i>Progressive Web App (PWA)</i> .....	47
3.10.4	Versionamento no <i>GitHub</i> e implantação na <i>Render</i> .....	51
3.10.5	Procedimento de replicação do protótipo.....	52
3.11	Procedimento de validação do protótipo .....	54
CAPÍTULO IV .....	56	
4	RESULTADOS E DISCUSSÕES .....	56
4.1	<i>Dataset</i> .....	56
4.2	Artigo Científico - Identificação Automatizada de Variedades de Pimentas com <i>YOLOv8</i> .....	56
4.3	Protótipo do aplicativo <i>App Pimentas</i> .....	59
4.4	Tentativa de validação do protótipo com a equipe da CEAGESP .....	61
4.4.1	Procedimento de coleta .....	61
4.4.2	Retorno obtido e implicações pra análise .....	62
4.4.3	Síntese da etapa de validação.....	62
CAPÍTULO V .....	64	
5	CONCLUSÕES.....	64
5.1	Considerações gerais sobre a pesquisa .....	64
5.2	Contribuições da pesquisa .....	64
5.2.1	Contribuições científicas .....	64
5.2.2	Contribuições metodológicas .....	65
5.2.3	Contribuições práticas .....	66
5.3	Limitações e implicações .....	67
5.4	Recomendações para trabalhos futuros .....	69
5.5	Considerações finais.....	70
REFERÊNCIAS BIBLIOGRÁFICAS .....	72	

APÊNDICE A – Questionário de validação do protótipo do aplicativo <i>App Pimentas</i> .....	79
APÊNDICE B – Acesso ao repositório <i>GitHub</i> do <i>App Pimentas</i> .....	82
ANEXOS A .....	83
ANEXO B .....	102

## CAPÍTULO I

### 1 CONSIDERAÇÕES INICIAIS

A agricultura desempenha papel estratégico na economia nacional e internacional; destaca-se como um dos setores mais competitivos do país. De acordo com a FAO (2024), no setor dos condimentos, para exemplo, a produção de pimentas (*Capsicum spp.*) alcançou mais de 812 mil toneladas globalmente em 2022, e o Brasil é o responsável por aproximadamente 128 mil toneladas do produto.

Nos últimos anos, a pressão por aumento de produtividade, por redução de custos e por adoção de práticas mais sustentáveis tem intensificado a busca por soluções inovadoras, em consonância com os Objetivos de Desenvolvimento Sustentável (ODS) estabelecidos pela ONU (2015). Nesse contexto, a Agricultura 4.0 desponta como paradigma que integra tecnologias digitais, como internet das coisas (IoT), inteligência artificial (IA) e *big data*, o que amplia o alcance da Indústria 4.0 para os sistemas agroalimentares (TIAN *et al.*, 2020; JAVAID *et al.*, 2022).

Entre as tecnologias emergentes, a visão computacional baseada em inteligência artificial tem se consolidado como uma ferramenta estratégica para a agricultura, ao possibilitar a automatização de tarefas tradicionalmente dependentes de mão de obra intensiva. Estudos recentes apresentam a aplicação de técnicas de visão computacional com aprendizado profundo em diferentes contextos agrícolas, com evidência de redução de erros associados ao trabalho manual e de aumento da consistência nos resultados. Por exemplo, Dhanya (2022) apresentou abordagens profundas que automatizam operações agrícolas com alta precisão, enquanto Taneja *et al.* (2023) destacaram como a inteligência artificial melhora a eficiência, reduz o desperdício e fortalece a qualidade no setor agroalimentar.

No caso das pimentas, a elevada diversidade morfológica entre variedades, somada ao grande volume comercializado em centrais atacadistas, como a CEAGESP, torna a classificação manual especialmente vulnerável a inconsistências, fato que reforça a necessidade de soluções tecnológicas inovadoras.

#### 1.1 Introdução

Esta pesquisa insere-se no contexto da Agricultura 4.0, marcada pela integração de tecnologias digitais e inteligência artificial ao setor agroalimentar. Para manter o

foco da Introdução e evitar redundâncias, o aprofundamento conceitual e histórico sobre Agricultura 4.0, visão computacional e a evolução dos detectores de objetos, com a inclusão da família YOLO, encontra-se detalhado no Capítulo II. Em termos de relevância social, o trabalho dialoga diretamente com os Objetivos de Desenvolvimento Sustentável, em especial o ODS 2, o ODS 9 e o ODS 12, ao promover inovação tecnológica, eficiência produtiva e padronização de processos.

## 1.2 Justificativa

A aplicação de técnicas de visão computacional na agricultura não se restringe à identificação de variedades de pimentas. Diversos estudos têm explorado o potencial das CNNs em tarefas como classificação de sementes (KURTULMUŞ, ALIBAŞ e KAVDIR, 2016), detecção de doenças em folhas (ZENG *et al.*, 2021) e sistemas inteligentes de classificação em linhas de produção (MOHI-ALDEN *et al.*, 2022). Esses avanços reforçam a aplicabilidade do aprendizado profundo para resolver gargalos de eficiência e qualidade na agricultura de precisão.

Nesse sentido, o desenvolvimento de um protótipo de aplicativo móvel que integre o modelo treinado amplia a relevância prática da investigação, pois permite a aplicação direta em campo e em ambientes comerciais. Essa proposta também se alinha à Engenharia de Produção, ao contribuir para processos de padronização, redução de falhas humanas e otimização de cadeias agroalimentares intensivas em mão de obra (KAMILARIS e PRENAFETA-BOLDÚ, 2018).

Por fim, embora a literatura recente evidencie avanços em visão computacional aplicada à agricultura, observa-se escassez de estudos dedicados especificamente à classificação automática de variedades de *Capsicum spp.* em ambientes comerciais de larga escala, como a CEAGESP. Essa lacuna fundamenta o presente trabalho, que busca a preencher com a aplicação do YOLOv8, associada ao desenvolvimento de um protótipo de aplicativo móvel.

Diante disso, surge a questão que norteia esta pesquisa: Como aplicar e avaliar a arquitetura YOLOv8 na identificação automática de variedades de *Capsicum spp.*

em ambientes comerciais de alta complexidade e integrá-la a um protótipo de aplicativo móvel voltado ao uso prático?

### 1.3 Objetivos

#### 1.3.1 Objetivo Geral

Avaliar a aplicabilidade da arquitetura YOLOv8 na identificação automática de variedades de pimentas e desenvolver um aplicativo móvel que integre o modelo treinado, com a permissão de seu uso prático em tempo real.

#### 1.3.2 Objetivos Específicos

- A. Construir e organizar um *dataset* de imagens de variedades de pimentas** coletadas na CEAGESP e complementadas com bases abertas.
- B. Avaliar o desempenho da arquitetura YOLOv8** em cenários com e sem *data augmentation*, com a consideração de métricas, como *Box Precision*, *Recall*, *mAP50* e *mAP95*.
- C. Desenvolver e validar um protótipo de aplicativo móvel** que integre o modelo YOLOv8, com classificação em tempo real e informações adicionais por meio de um *chatbot*.

### 1.4 Metodologia Resumida

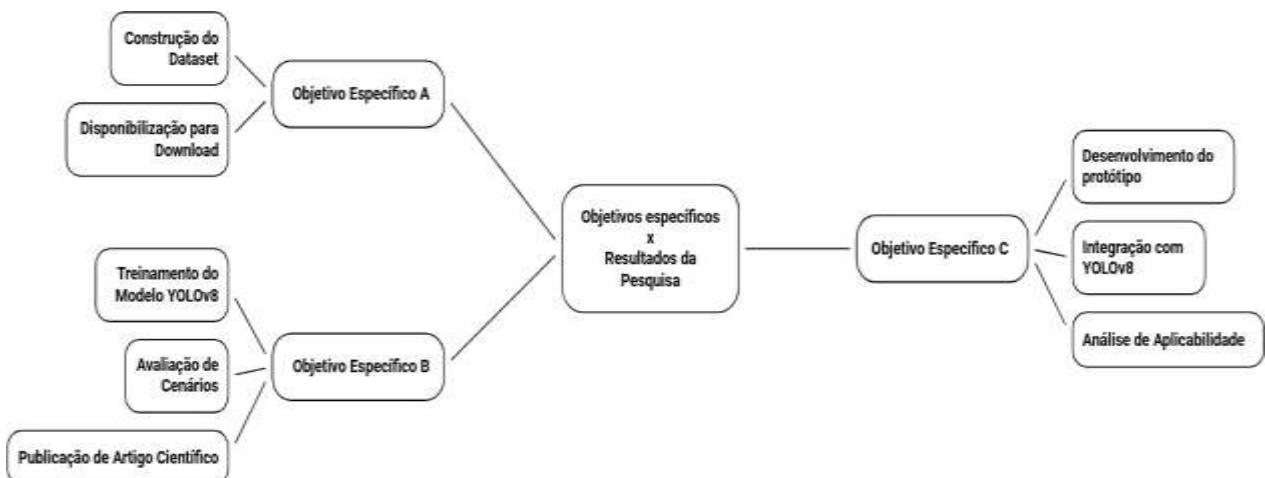
Esta pesquisa adota uma abordagem aplicada e experimental, voltada à avaliação de técnicas de visão computacional aplicadas à agricultura, especificamente à classificação automática de variedades de *Capsicum spp.*. O estudo envolve três etapas principais: a construção de um *dataset* de imagens, a avaliação experimental da arquitetura YOLOv8 em diferentes cenários de treinamento e o desenvolvimento de um protótipo de aplicativo móvel. A opção por detalhar minuciosamente cada etapa metodológica encontra respaldo em Gil (2018), que postula que a clareza dos procedimentos é fundamental para contribuir com a replicabilidade do estudo, um pilar da pesquisa científica.

### 1.5 Organização estrutural da pesquisa

A dissertação está organizada em cinco capítulos. O Capítulo 1 apresenta a contextualização do tema, a justificativa, os objetivos e uma metodologia resumida. O

Capítulo 2 reúne o referencial teórico sobre Agricultura 4.0, visão computacional, aprendizado profundo, redes neurais convolucionais e a evolução da família YOLO, além de abordar as características das pimentas e os desafios da classificação automática. O Capítulo 3 descreve os procedimentos de revisão de literatura e de coleta de dados que fundamentam a pesquisa. Conforme ilustrado na Figura 1, o Capítulo 4 apresenta os resultados parciais e a discussão. Ele abrange a disponibilização do *dataset* (objetivo específico A), a aplicação e a avaliação experimental do modelo YOLOv8 (objetivo específico B) e o desenvolvimento do protótipo de aplicativo móvel (objetivo específico C). O artigo científico que detalha parte desses resultados encontra-se integralmente apresentado no Anexo A. O Capítulo 5 apresenta as conclusões parciais.

**Figura 1-** Relação entre objetivos específicos e resultados da pesquisa



Fonte: A autora (2025).

## CAPÍTULO II

### 2 REFERENCIAL TEÓRICO

#### 2.1 Agricultura 4.0 e contexto global

No setor agroalimentar, a Agricultura 4.0 apresenta-se como vetor estratégico para a modernização das cadeias de suprimento, o que permite o uso de ferramentas como internet das coisas (IoT), inteligência artificial (IA), sensores inteligentes e análise em tempo real para apoiar a tomada de decisão (SUBEESH e MEHTA, 2021). Essas tecnologias possibilitam maior rastreabilidade, redução de perdas e otimização no uso de recursos naturais; alinharam-se às diretrizes globais de sustentabilidade (OLIVEIRA e SILVA, 2023).

A convergência entre a Indústria 4.0 e os sistemas agroalimentares ampliou a consolidação do termo *AgriFood*, que enfatiza a transformação digital das cadeias agroalimentares em toda a sua extensão, da produção agrícola ao processamento, da distribuição ao consumo final. Essa perspectiva destaca o papel das tecnologias digitais e verdes como elementos essenciais para acelerar o cumprimento dos Objetivos de Desenvolvimento Sustentável (ODS), ao promover práticas agrícolas mais resilientes, eficientes e ambientalmente responsáveis (LEZOCHE *et al.*, 2020; HASSOUN *et al.*, 2022).

O Brasil, enquanto uma das maiores potências agroalimentares globais, desempenha papel central nessa transformação. A adoção de soluções digitais apresenta potencial significativo de impacto econômico e social, ao se considerar a extensão territorial e a diversidade de cultivos do país (EMBRAPA, 2020). Nesse contexto, mercados atacadistas, como a Companhia de Entrepótos e Armazéns Gerais de São Paulo (CEAGESP), assumem papel estratégico, dado o volume e a diversidade de hortifrutigranjeiros movimentados. A classificação e a padronização de produtos nesses entrepostos configuram um gargalo logístico que pode ser mitigado pela aplicação de tecnologias da Agricultura 4.0.

Assim, a Agricultura 4.0 e, em sentido ampliado, o conceito de *Agri-Food* 4.0, devem ser compreendidos não apenas como instrumentos de ganho de produtividade, mas também como vetores de sustentabilidade, inovação e

competitividade. Ao integrar inteligência artificial, visão computacional e sistemas móveis, tais iniciativas contribuem para a eficiência operacional e para a transformação das cadeias agroalimentares em direção a modelos mais sustentáveis e inovadores.

## 2.2 Visão computacional na Agricultura

A visão computacional (VC) constitui um campo da inteligência artificial voltado para a interpretação automatizada de imagens e vídeos digitais, com o objetivo de extrair informações relevantes para processos de tomada de decisão. Nos últimos anos, avanços em capacidade computacional e algoritmos de aprendizado profundo possibilitaram a aplicação da VC em contextos complexos e isto supera limitações de métodos tradicionais baseados em extração manual de atributos (WANG e KANG, 2025).

A ascensão da Agricultura 4.0 é intrinsecamente ligada aos avanços em inteligência artificial, notadamente no campo da VC. Esta disciplina consolidou-se como um pilar tecnológico para a automação e a otimização de processos agrícolas, o que permite que sistemas computacionais interpretem e extraiam significado de dados visuais para a execução de tarefas com elevada precisão. Revisões recentes destacam a diversidade de aplicações; incluem identificação de doenças em plantas, classificação de frutos e sementes, detecção de plantas daninhas e estimativa de produtividade (CAO, SUN e BAO, 2025).

A manutenção de elevada precisão e velocidade em sistemas de detecção aplicados a ambientes agrícolas complexos constitui um importante eixo de investigação. Nesse contexto, Li *et al.* (2021) propuseram uma versão aprimorada do modelo *YOLOv4-tiny* para a detecção em tempo real de pimentas verdes, fato que busca mitigar problemas decorrentes de oclusões severas por galhos e folhas, bem como da significativa variação de escala dos alvos. O modelo aprimorado incorporou predição em múltiplas escalas e um mecanismo híbrido de atenção e isto alcançou desempenho expressivo, com 95,11% de precisão média, o que indica adequação para implementação embarcada em sistemas robóticos agrícolas.

No campo do diagnóstico de fitopatologias, definido como o estudo de doenças que comprometem o ciclo biológico das plantas, Bezabh *et al.* (2023) indicaram o modelo *CPD-CCNN*, uma rede convolucional concatenada que se deriva das

arquiteturas *VGG16* e *AlexNet*. O objetivo consistiu na classificação de doenças que afetam plantas de pimenta, entre elas ferrugem comum, mancha foliar e podridão do fruto, com a superação da limitação de abordagens anteriores centradas em classificações binárias. O modelo alcançou precisão de 95,82 por cento.

Por sua vez, Zeng *et al.* (2021) desenvolveram um modelo de *deep learning* baseado em redes neurais convolucionais (CNNs) para a detecção de doenças em folhas de pimenta. O estudo utilizou um conjunto de dados que contém 2.478 imagens e aplicou técnicas de transferência de aprendizado para aprimorar a precisão diagnóstica em condições reais de cultivo. O modelo atingiu 99,55% de acurácia na identificação de folhas sadias e doentes, o que demonstra sua aplicabilidade prática para detecção precoce de doenças na agricultura.

Embora estudos recentes ampliem o uso de visão computacional na cadeia produtiva da pimenta, trabalhos anteriores já indicavam a aplicabilidade dessas técnicas. Kurtulmuş *et al.* (2016) analisaram sementes de pimenta por meio de extração de características de cor, forma e textura e classificaram os dados com um *perceptron multicamadas*. A acurácia obtida foi de 84,94 por cento, o que indica a aplicabilidade de métodos de aprendizado profundo desde etapas iniciais da produção.

Pesquisas também sugerem que a integração da VC com redes neurais convolucionais e algoritmos de detecção de objetos potencializa a robustez das análises e permite o seu uso em ambientes não controlados. Upadhyay (2025), em uma revisão abrangente, aponta que técnicas de *deep learning* aplicadas à detecção de doenças em plantas, com imagens RGB, multiespectrais e hiper espectrais, oferecem resultados de alta precisão, com potencial para diagnósticos precoces e não destrutivos. Essa característica é particularmente relevante para cadeias de produção agrícola que demandam rapidez e confiabilidade na tomada de decisão.

Apesar dos avanços, a adoção da VC na agricultura enfrenta desafios significativos. Entre eles destacam-se a necessidade de grandes conjuntos de dados anotados, a variabilidade de condições ambientais (iluminação, ângulo de captura, clima) e a elevada demanda computacional de alguns modelos, o que pode dificultar a implementação em dispositivos móveis e sistemas distribuídos (MIN, 2025). Para mitigar essas limitações, a literatura recente aponta para soluções, como aprendizado

auto supervisionado, modelos leves otimizados para *edge computing* e a fusão multimodal de dados provenientes de diferentes sensores (SORNAPUDI e SINGH, 2024).

Portanto, a visão computacional na agricultura representa uma das áreas mais promissoras da Agricultura 4.0, ao possibilitar ganhos em eficiência, sustentabilidade e padronização de processos, ainda que dependa de contínuos avanços técnicos e da superação de barreiras operacionais para sua plena integração em ambientes de produção.

### **2.3 Aprendizado Profundo e Redes Neurais Convolucionais (CNNs)**

O aprendizado profundo, ou *deep learning*, consolidou-se como paradigma dominante nas aplicações de VC, o que viabiliza a extração automatizada de características visuais sem depender de processamento manual por parte do pesquisador (EL SAKKA, 2025).

Entre as arquiteturas de *deep learning*, as redes neurais convolucionais (CNNs) destacam-se por sua capacidade de aprender hierarquias espaciais complexas, por meio de operações de convolução, ativação e redução de dimensionalidade. Elas permitem reconhecer padrões visuais, desde bordas até estruturas sintáticas de alto nível, diretamente em dados de imagem (LEITE, 2024).

No âmbito agrícola, as CNNs têm sido aplicadas com sucesso em tarefas como detecção de doenças em plantas, identificação de pragas, classificação de estágios de crescimento e monitoramento de colheitas. El Sakka (2025) aponta que modelos baseados em *deep learning* superam métodos tradicionais como *k-means* e Máquina de Vetores de Suporte (SVM) em diversos cenários agroalimentares, especialmente quando se busca robustez e inovação em condições visuais adversas.

Em atividades, como a identificação de estresse hídrico ou sanidade vegetal, CNNs têm se mostrado eficazes ao processar imagens obtidas via drones, sensores remotos e câmeras multiespectrais. As redes aceleram diagnósticos e favorecem decisões em tempo real (KESKES, 2025). No entanto, a adoção dessas redes no campo ainda enfrenta limitações relevantes, uma vez que a coleta e a anotação de grandes bases de dados demandam alto esforço e conhecimento especializado. Além disso, os modelos costumam ser computacionalmente intensivos, o que restringe sua execução em dispositivos de borda (*edge computing*). A revisão de Hossen (2025)

sobre transferência de aprendizado destaca que o uso de modelos pré-treinados reduz esses obstáculos e viabiliza seu emprego em aplicações com menos dados e infraestrutura limitada.

Em síntese, as CNNs representam a espinha dorsal do *deep learning* aplicado à agroecologia e são decisivas para conectar a teoria de IA à prática da Agricultura 4.0. No entanto, seu uso exige mecanismos de adaptação (como transferência de aprendizado e modelos leves), sobretudo em contextos com dados escassos ou recursos computacionais limitados.

## 2.4 Modelos de Detecção de Objetos

Os modelos de detecção de objetos representam um avanço significativo da visão computacional, pois combinam tarefas de localização e classificação em imagens digitais. Essa capacidade é fundamental em contextos agrícolas, em que se busca identificar não apenas a presença de um fruto, folha ou praga, mas também sua posição e sua quantidade em uma cena (ZHANG *et al.*, 2021).

Historicamente, os detectores de objetos foram divididos em duas categorias principais. O primeiro grupo inclui modelos *two-stage*, como o *Faster R-CNN*, que, primeiro, gera regiões de interesse; depois, realiza a classificação e apresenta elevado nível de precisão, mas com custo computacional mais alto (REN *et al.*, 2015). O segundo grupo engloba os modelos *one-stage*, como *YOLO*, *SSD*, *RetinaNet* e *EfficientDet*, que realizam detecção e classificação em uma única etapa, o que proporciona maior velocidade de inferência com desempenho competitivo (TAN e LE, 2020).

No setor agrícola, os modelos de dois estágios (*two-stage*), como o *Faster R-CNN*, têm sido aplicados com sucesso no diagnóstico de doenças e na identificação de pragas e o fato demonstra alta precisão para alvos pequenos e com características complexas. Já os modelos de um estágio (*one-stage*), como o *YOLO*, destacam-se por sua vantagem em velocidade, o que viabiliza o processamento rápido na detecção de doenças e pragas. A eficácia desses modelos para aplicações em cenários do mundo real e em tempo real é evidenciada pela integração em aplicativos para plataformas móveis. Ela atinge taxas de quadros por segundo (FPS) adequadas para diagnósticos preliminares. A escolha entre uma arquitetura *one-stage* e *two-stage* depende do equilíbrio necessário entre precisão, tempo de resposta e recursos

computacionais disponíveis (LI *et al.*, 2023).

A literatura recente também aponta para uma tendência de integração entre eficiência e escalabilidade. Modelos como o *EfficientDet* alcançam bom desempenho em cenários de uso agrícola com recursos limitados, enquanto o *RetinaNet*, com sua função de perda focal *loss*, tem mostrado robustez em *datasets* desbalanceados, comuns no agro (TAN e LE, 2020; CAO, SUN e BAO, 2025). Essa adaptabilidade é crucial para aplicações em ambientes reais, onde a coleta de dados enfrenta variações de iluminação, oclusões e ângulos não controlados.

Essas inovações confirmam a tendência de busca por detectores que combinem alta precisão com eficiência computacional, essenciais para ambientes agroalimentares de alta variabilidade. Tais avanços oferecem base técnica para compreendermos a evolução posterior dos detectores *YOLO* e sua aplicabilidade no contexto das pimentas.

## 2.5 Evolução da Arquitetura *YOLO*

A arquitetura *YOLO* (*You Only Look Once*), proposta por Redmon *et al.* em 2016, foi estruturada como um método de detecção de objetos em que a imagem é dividida em uma grade *SxS*, e o modelo prevê simultaneamente as *bounding boxes*, que são retângulos que delimitam a posição dos objetos na cena, e as probabilidades de classe em uma única avaliação da rede, e isto caracteriza um processo de detecção de etapa única (REDMON *et al.*, 2016). Em contraste com abordagens *two-stage*, como o *Faster R-CNN*, o *YOLO* executa o processo em um único passo; ele reduz o tempo de inferência e possibilita o uso em sistemas com restrições computacionais. Desde sua formulação, a arquitetura *YOLO* passou por versões sucessivas voltadas ao aprimoramento da precisão e da eficiência.

O *YOLOv1*, apresentado em 2016, introduziu o paradigma da detecção em tempo real, ainda com limitações em objetos pequenos (REDMON *et al.*, 2016). Já o *YOLOv2* (*YOLO9000*), lançado em 2017, incorporou o uso de *anchor boxes* e a capacidade de detectar milhares de categorias simultaneamente (REDMON e FARHADI, 2017). Em 2018, foi lançado o *YOLOv3* que trouxe a *backbone Darknet-53* e que ampliou a profundidade da rede e a robustez em diferentes escalas (REDMON e FARHADI, 2018).

Segundo Bochkovskiy *et al.* (2020), o *YOLOv4* introduziu as estratégias *bag of*

*freebies* e *bag of specials*, que otimizaram o desempenho sem aumento significativo do custo computacional. No mesmo ano, a *Ultralytics* lançou o YOLOv5, amplamente adotado pela comunidade científica por sua facilidade de implementação em *PyTorch* e ampla documentação. O YOLOv6, lançado em 2022 pela Meituan, foi voltado a aplicações industriais, com otimizações para dispositivos de borda e maior velocidade de inferência, e mantém precisão competitiva (LI *et al.*, 2022).

O YOLOv7 apresentou o módulo *E-ELAN*, que aprimorou a convergência e a precisão e consolidou-se como o detector mais rápido e preciso de sua geração (WANG *et al.* 2022). Publicado em 2023 pela *Ultralytics*, o YOLOv8 reformulou a arquitetura com melhorias no *head* de detecção, além de introduzir modelos mais leves e integrados a fluxos de implantação em dispositivos móveis.

O YOLOv9, em 2024, apresentou o conceito de *Programmable Gradient Information* (PGI) e a espinha dorsal GELAN e alcançou melhor aproveitamento de características em diferentes escalas (WANG, YEH e LIAO, 2024). O YOLOv10, em 2024, propôs um desenho livre de *NMS* com atribuições duais consistentes; obteve maior eficiência energética e latências menores; é adequado a aplicações em *edge computing* (WANG *et al.*, 2024).

No setor agrícola, Khan, Shen e Liu (2025) reforçam que arquiteturas YOLO são particularmente relevantes para detecção de pragas, frutos e doenças devido à necessidade de diagnósticos rápidos e escaláveis. Em complementação, Venkateswara e Padmanabhan (2025) mostraram que o YOLOv10 atinge taxas de quadros por segundo elevadas e desempenho robusto em tarefas agroalimentares complexas. Ele supera alternativas, como *EfficientDet* e *RetinaNet*, em cenários de campo.

Finalmente, em 2024, a *Ultralytics* lançou o YOLOv11, a versão mais recente até o momento, que foca no equilíbrio entre precisão e velocidade. A atualização melhora a eficiência da arquitetura e a integração para implantação em dispositivos móveis e na nuvem. Segundo a empresa, o YOLOv11 aprimora o treinamento, o desempenho em múltiplos *datasets* e a escalabilidade para aplicações do mundo real (ULTRALYTICS, 2024).

Cabe destacar que várias versões do YOLO foram inicialmente divulgadas em formato de *preprint*, como ocorre com o YOLOv4, YOLOv6, YOLOv7, YOLOv9 e

YOLOv10. Embora esses trabalhos não tenham passado pelo processo tradicional de revisão por pares, sua relevância científica está consolidada por sua ampla adoção na comunidade acadêmica, pelo elevado número de citações e por serem frequentemente analisados em revisões sistemáticas indexadas em *Scopus* e *Web of Science* (KHAN, SHEN e LIU, 2025; DALAL e MITTAL, 2025). Dessa forma, o uso de *preprints* neste referencial teórico reflete a própria dinâmica de disseminação da pesquisa em visão computacional, em que muitas contribuições fundamentais são divulgadas primeiramente em repositórios de acesso aberto.

A trajetória do YOLO, do *v1* ao *v11*, deve ser entendida como um exemplo de avanço científico incremental e colaborativo, no qual cada versão incorpora melhorias arquiteturais que viabilizam novas aplicações. Esse movimento consolida a arquitetura como uma das mais adequadas para cenários agroalimentares complexos, como a classificação automática de variedades de *Capsicum spp.* na CEAGESP, onde robustez e processamento em tempo real são requisitos fundamentais.

## 2.6 Pimentas do gênero *Capsicum*

O gênero *Capsicum L.* (*Solanaceae*) reúne espécies de ampla relevância econômica, gastronômica e científica. É originário das Américas, com centro de diversidade nos Andes e distribuição que se estende do sul dos Estados Unidos até a Argentina e o Brasil (BARBOZA *et al.*, 2022). Atualmente, são reconhecidas 43 espécies, das quais cinco foram domesticadas e alcançaram cultivo em escala global: *C. annuum*, *C. chinense*, *C. frutescens*, *C. baccatum* e *C. pubescens* (BARBOZA *et al.*, 2022).

As pimentas são valorizadas não apenas por seu papel na alimentação, mas também por atributos nutricionais e funcionais. Os frutos apresentam elevado teor de compostos bioativos, entre os quais se destacam os capsaicinóides, responsáveis pela pungência característica (BASITH *et al.*, 2016). Dentre esses, a capsaicina e a dihidrocapsaicina respondem por até 90% da ardência e atuam como principais marcadores químicos de intensidade sensorial (BASITH *et al.*, 2016). A intensidade da pungência é mensurada pela Escala de *Scoville*, proposta em 1912, que varia de 0 SHU (pimentões doces) a 16 milhões SHU (capsaicina pura) (SCOVILLE, 1912).

Além da utilização como condimento e conservante natural, estudos recentes

destacam o potencial farmacológico e industrial da capsaicina. Entre suas atividades biológicas estão efeitos analgésicos, anti-inflamatórios, antioxidantes e anticancerígenos, com aplicações no tratamento de dores neuropáticas, distúrbios metabólicos e inflamações crônicas (BASITH *et al.*, 2016). O uso ornamental de algumas espécies e o emprego de derivados de pimentas em formulações industriais, repelentes e produtos de defesa pessoal também ampliam o espectro de aplicações (BARBOZA *et al.*, 2022).

No contexto brasileiro, as pimentas apresentam relevância tanto pelo consumo interno, quanto pela presença em cadeias de distribuição atacadista, como a Companhia de Entrepósitos e Armazéns Gerais de São Paulo (CEAGESP), onde o grande volume de comercialização evidencia a importância socioeconômica do gênero (BARBOZA *et al.*, 2022). Entretanto, os processos tradicionais de classificação e padronização, baseados em inspeção visual, ainda constituem um gargalo logístico, marcado pela subjetividade e pela suscetibilidade a erros (CRUZ-DOMÍNGUEZ *et al.*, 2021).

Pesquisas recentes indicam que ferramentas de visão computacional e inteligência artificial podem superar essas limitações e proporcionar maior precisão e escalabilidade na identificação varietal. Cruz-Domínguez *et al.* (2021) desenvolveram um sistema baseado em redes neurais artificiais para classificação de pimentas secas. Obtiveram acurácia superior a 82% na diferenciação entre classes de qualidade, o que evidencia o potencial de tecnologias automatizadas para modernizar a cadeia produtiva.

## **2.7 Desafios na Classificação Automática de Variedades de *Capsicum***

A classificação automática de variedades de *Capsicum spp.* enfrenta desafios específicos que decorrem tanto das características intrínsecas da cultura, quanto das limitações tecnológicas ainda presentes nas abordagens atuais de visão computacional. Diferentemente de outras hortaliças ou frutos de maior variação morfológica, as pimentas apresentam elevado grau de similaridade visual entre variedades, o que dificulta a distinção por algoritmos de aprendizado profundo (BARBOZA *et al.*, 2022).

Outro desafio relevante refere-se às condições ambientais sob as quais as imagens são capturadas. Em ambientes comerciais como a CEAGESP, a iluminação

artificial irregular, a presença de sombras, reflexos e a sobreposição de frutos no mesmo espaço reduzem a qualidade das imagens e prejudicam a generalização dos modelos. Estudos recentes apontam que a variabilidade de iluminação e ângulo de captura pode impactar significativamente a precisão de sistemas baseados em CNNs e YOLO (KHAN, SHEN e LIU, 2025).

A disponibilidade e a qualidade de *datasets* também constituem barreira crítica. Muitos estudos em visão computacional aplicada ao agro utilizam bases de dados limitadas, com amostras desbalanceadas entre classes, o que compromete a robustez e aumenta o risco de *overfitting* (DALAL e MITTAL, 2025). A escassez de imagens rotuladas de forma padronizada para diferentes variedades de pimentas reforça a necessidade de estratégias como *data augmentation* e *transfer learning* para compensar a falta de dados em ambientes reais (PADILLA *et al.*, 2021).

Outro aspecto a ser considerado é o custo computacional das arquiteturas mais avançadas. Embora versões recentes da família YOLO tenham sido otimizadas para uso em dispositivos móveis e de borda, sua aplicação em mercados atacadistas demanda processamento rápido em tempo real, o que exige balancear a escolha de modelos mais leves com a manutenção da precisão necessária para a classificação varietal (VENKATESWARA e PADMANABHAN, 2025).

Além disso, a aceitação prática da tecnologia depende da integração com ferramentas acessíveis e compatíveis com o cotidiano de produtores e comerciantes. Sistemas complexos ou que exigem infraestrutura computacional avançada tendem a enfrentar barreiras de adoção em contextos como a agricultura familiar e os mercados regionais. Esse desafio é enfatizado por Porciello *et al.* (2022), que apontam que a agricultura digital depende da capacidade dos sistemas de compartilhar dados. Muitas regiões, entretanto, ainda apresentam limitações estruturais básicas, como acesso instável à eletricidade ou a redes de telefonia móvel, além da ausência de arquiteturas adequadas para o compartilhamento de dados. Nesse cenário, soluções que conciliem precisão técnica, simplicidade operacional e baixo custo tornam-se essenciais para sua efetiva implementação.

Esses desafios evidenciam uma lacuna científica e prática ainda pouco explorada. Embora os modelos da família YOLO tenham alcançado avanços expressivos em tarefas de detecção de objetos em diferentes domínios, sua aplicação

direcionada à classificação automática de variedades de *Capsicum spp.* em ambientes comerciais de alta complexidade, como a CEAGESP, permanece limitada. A ausência de *datasets* específicos, as dificuldades de generalização em condições reais de mercado e a escassez de soluções integradas em plataformas acessíveis reforçam a necessidade de estudos que aliem robustez técnica e aplicabilidade prática. Nesse contexto, a presente dissertação busca preencher essa lacuna ao avaliar o desempenho do YOLOv8 na identificação varietal de pimentas e ao propor sua integração em um protótipo de aplicativo móvel voltado a usuários do setor agroalimentar.

A escolha do YOLOv8 como modelo base deve-se a um conjunto de fatores técnicos e práticos. Em comparação com versões anteriores da família YOLO, o YOLOv8 apresenta melhorias arquiteturais que resultam em maior precisão e maior estabilidade de treinamento, o que, ao mesmo tempo, mantém estrutura mais leve e escalável (ULTRALYTICS, 2023). Na família, a versão “m” (medium) representa um equilíbrio entre desempenho e custo computacional: é mais robusta que as versões “n” (nano) e “s” (small), mas ainda viável para ser executada em dispositivos móveis e de borda, o que a torna especialmente adequada para o desenvolvimento do protótipo de aplicativo proposto nesta pesquisa. Essa decisão metodológica está alinhada ao objetivo de oferecer uma solução prática e acessível, que combine rigor científico com aplicabilidade em ambientes comerciais como a CEAGESP.

É importante destacar que versões posteriores da família YOLO, como o YOLOv9, YOLOv10 e YOLOv11, foram lançadas ao longo de 2024, quando esta pesquisa já estava em andamento. Embora representem avanços relevantes, essas arquiteturas ainda careciam de validações consolidadas em cenários agrícolas, o que poderia comprometer a reproduzibilidade no cronograma da pesquisa. Dessa forma, o YOLOv8 foi selecionado como a alternativa mais estável e viável, pois garante equilíbrio entre inovação, eficiência computacional e aplicabilidade prática.

## CAPÍTULO III

### 3 METODOLOGIA

#### 3.1 Caracterização da Pesquisa

De acordo com Gil (2018) e Yin (2015), esta pesquisa pode ser caracterizada como:

**Natureza** aplicada, pois busca propor uma solução prática para o problema da identificação de variedades de pimentas, por meio do uso de visão computacional e do desenvolvimento de um aplicativo.

**Abordagem** quantitativa, já que envolve a mensuração do desempenho do modelo YOLOv8 por métricas objetivas *recall* e precisão média (mAP); e qualitativa em caráter complementar, ao discutir a aplicabilidade prática do protótipo do aplicativo.

**Objetivos** exploratórios, ao investigar o potencial de aplicação da arquitetura YOLOv8 em um novo contexto agrícola; explicativos, ao analisar os resultados obtidos e suas limitações.

**Procedimento técnico** experimental, por empregar treinamento e testes de rede neural em diferentes cenários de *dataset* (com e sem *data augmentation*), e de desenvolvimento tecnológico, por englobar a elaboração de um protótipo de aplicativo móvel que integra o modelo treinado.

#### 3.2 Protocolo de Revisão de Literatura

A revisão bibliográfica foi conduzida de forma sistemática para garantir uma base teórica sólida e atualizada. Foram consultadas as bases *Scopus*, *Web of Science* e *Google Scholar*, com a priorização de artigos publicados entre 2019 e 2024:

**Critérios de inclusão:** estudos aplicados de visão computacional na agricultura, uso de CNNs e evolução da família YOLO.

**Critérios de exclusão:** trabalhos não indexados em bases de alta credibilidade ou com escopo divergente.

**Relevância científica:** a seleção priorizou artigos publicados em periódicos de alto impacto, predominantemente classificados em Q1 e Q2 nas bases *Scopus/Web of Science*, e nos estratos P1 a P3 do *Qualis CAPES* (2022), que correspondem aos níveis mais elevados de qualidade da produção científica nacional.

### **3.3 *Design Science Research (DSR)***

A presente pesquisa adota como orientação metodológica o *Design Science Research (DSR)*, abordagem que vem sendo discutida e atualizada nos últimos anos, especialmente no contexto de pesquisas aplicadas que buscam gerar simultaneamente conhecimento científico e soluções práticas (VOM BROCKE, HEVNER e MAEDCHE, 2020). O DSR estrutura-se em fases iterativas que incluem a identificação do problema, a definição dos objetivos, o *design* e desenvolvimento do artefato, sua demonstração e avaliação, e a comunicação dos resultados, com a garantia de rigor científico e relevância prática.

No campo das ciências aplicadas, a adoção do *DSR* tem se mostrado especialmente útil para pesquisas que envolvem inovações tecnológicas e complexidade de implementação. Prost (2021) destaca o potencial do *DSR* para revitalizar práticas em ciências agrícolas, ao alinhar metodologias científicas com a construção de artefatos capazes de enfrentar problemas reais. Da mesma forma, Tuunanen *et al.* (2025) propõem que, em cenários de alta complexidade, como aqueles que envolvem múltiplos atores e recursos limitados, a aplicação do *DSR* deve considerar níveis de *design*, para assegurar maior clareza e coerência entre etapas.

No contexto desta dissertação, as fases do *DSR* foram operacionalizadas da seguinte forma:

#### **3.3.1 Fase 1 – Identificação do problema e definição dos objetivos**

O problema central foi definido a partir da dificuldade de classificação manual de variedades de *Capsicum spp.* em ambientes de grande escala, como a CEAGESP, o que gera inconsistências, perdas e subjetividade no processo. Dessa lacuna derivaram os objetivos da pesquisa: (i) desenvolver um *dataset* específico de pimentas, (ii) treinar e avaliar o modelo YOLOv8 em cenários distintos e (iii) propor um protótipo de aplicativo móvel para aplicação prática.

### 3.3.2 Fase 2 – Análise de requisitos

Nesta fase, foram definidos os requisitos técnicos, funcionais e científicos necessários ao desenvolvimento do artefato. Do ponto de vista técnico e científico, estabeleceu-se a necessidade de um *dataset* com diversidade morfológica adequada, a seleção de um modelo de detecção de objetos que apresentasse equilíbrio entre acurácia e viabilidade de uso em dispositivos móveis. Foi adotado o *YOLOv8* e a definição das métricas de avaliação, o que contempla Precisão, *Recall*, *mAP50* e *mAP95*.

Além desses aspectos, foram definidos os requisitos funcionais que orientam o fluxo operacional do protótipo. Conceitualmente, o sistema deverá seguir o seguinte processo:

- **Interface do usuário:** permitir que o usuário capture uma imagem diretamente pela câmera do dispositivo ou carregue uma foto já existente.
- **Processamento:** a imagem deverá ser enviada ao modelo *YOLOv8* hospedado em servidor próprio e acessado por meio de uma *API REST*.
- **Retorno dos resultados:** o sistema deverá exibir a variedade identificada, juntamente com informações adicionais pertinentes.
- **Chatbot de apoio:** o protótipo deverá incluir um módulo conversacional para fornecer informações complementares sobre a pimenta identificada, tais como nível de pungência, características botânicas, formas de consumo culinário e recomendações de armazenamento, fato que amplia a utilidade do sistema no contexto agroalimentar.

### 3.3.3 Fase 3 – Design e desenvolvimento do artefato

Nesta fase foram conduzidas as etapas de coleta de dados (seção 3.4), preparação do *dataset* (seção 3.5), aplicação de técnicas de *data augmentation* (seção 3.6), configuração computacional e procedimentos de treinamento (seções 3.7 e 3.8). Essas atividades resultaram no desenvolvimento do modelo *YOLOv8* treinado em dois cenários distintos, com e sem *data augmentation*.

### 3.3.4 Fase 4 – Demonstração

A demonstração do artefato ocorreu em duas etapas complementares. A primeira consistiu na aplicação experimental do modelo *YOLOv8* treinado sobre o

conjunto de teste, o que permite avaliar seu desempenho em condições controladas. A segunda etapa correspondeu à implementação e à integração do modelo em um protótipo funcional de aplicativo móvel, desenvolvido a partir dos *mockups* conceituais apresentados na Seção 3.10. Nessa fase, o protótipo foi integrado ao modelo por meio de uma *API REST*, fato que incorporou funcionalidades de captura de imagem, classificação automática e *chatbot* informativo. Concluída essa integração, o aplicativo foi disponibilizado à equipe técnica da CEAGESP para validação prática em condições reais de uso no ambiente comercial da unidade.

### **3.3.5 Fase 5 – Avaliação**

A avaliação inicial concentrou-se nas métricas de desempenho do modelo YOLOv8, amplamente reconhecidas na literatura, inclusive em *Precisão*, *Recall*, *mAP50* e *mAP95*. Em etapa posterior, prevista como continuidade da pesquisa, será realizada a avaliação prática do protótipo móvel em ambiente real, com participação da equipe parceira da CEAGESP.

### **3.3.6 Fase 6 – Comunicação**

Os resultados já obtidos foram comunicados por meio de artigo científico publicado na revista *Applied Sciences* (2025) e estão sistematizados nesta dissertação. Após a implementação da versão funcional, o aplicativo foi disponibilizado à equipe técnica da CEAGESP, acompanhada de um formulário de avaliação que buscou registrar a percepção dos usuários sobre aspectos operacionais, clareza da interface e adequação das funcionalidades às rotinas do entreposto. Essa ação teve como objetivo comunicar o artefato ao público diretamente envolvido na cadeia comercial de pimentas e coletar contribuições que possam orientar ajustes futuros.

A combinação entre a documentação apresentada nesta dissertação e o processo de validação junto aos profissionais da CEAGESP permite que o artefato seja apresentado de forma transparente e contextualizada, que ofereça subsídios para sua análise, seu uso e seus possíveis aprimoramentos em trabalhos posteriores.

Essa organização metodológica contribui para o alinhamento entre problema, objetivos, procedimentos e resultados; ela reforça a replicabilidade da pesquisa e atende as recomendações de Gil (2018) quanto à clareza no detalhamento de etapas metodológicas.

### 3.4 Coleta de dados

A coleta de dados foi realizada em duas frentes:

**Coleta primária:** A coleta de dados foi realizada na CEAGESP, principal centro de comercialização hortifrutigranjeira do país. O objetivo foi registrar imagens representativas das oito variedades de *Capsicum* selecionadas, capturando a variabilidade visual observada em condições reais de mercado.

As fotografias foram obtidas diretamente nas bancas. Utilizou-se a iluminação natural do ambiente e preservaram-se os elementos característicos do cenário comercial, como fundos variados, embalagens, caixas, sombras e diferentes disposições dos frutos. Essa estratégia buscou refletir o contexto genuíno de uso do sistema, o que permitiu que o modelo fosse treinado com imagens compatíveis com aquelas posteriormente encontradas no ambiente de aplicação.

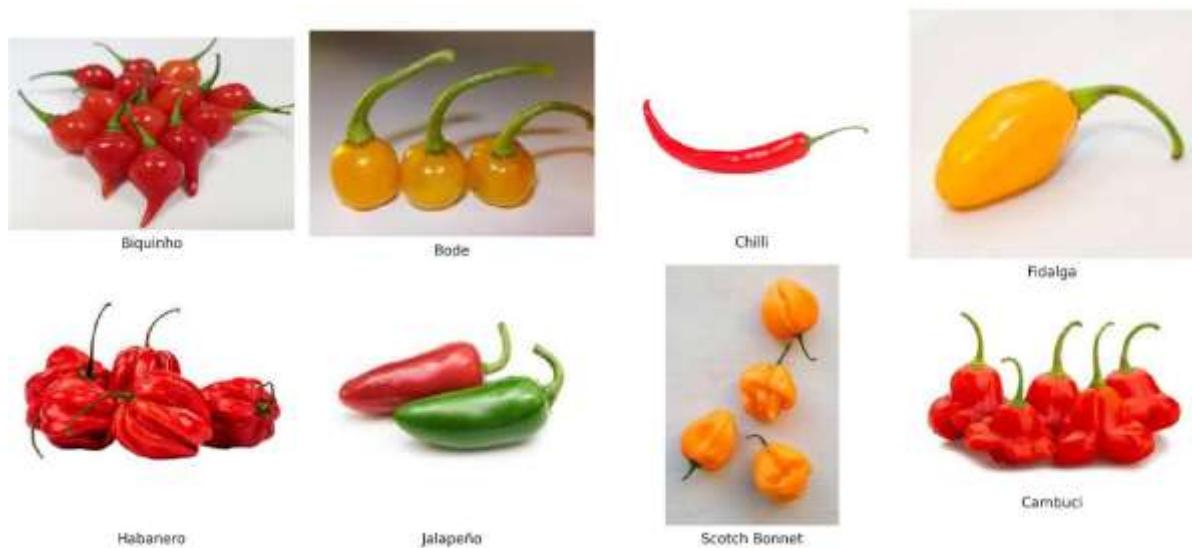
Foram utilizados três dispositivos para a captura das imagens, com a garantia de resolução, textura e profundidade de campo:

- Samsung Galaxy Tab A8, com câmera traseira de 8 MP;
- Nikon Coolpix L120, no modo automático, com sensor CCD de 14,1 MP e lente com zoom óptico de 21x;
- iPhone 11, equipado com sistema de câmeras duplas de 12 MP e capacidade de gravação em 4K.

**Coleta secundária:** Foi feita manualmente por busca *on-line* em bases abertas; o *Google Imagens* (com filtro de licenças *Creative Commons*) como fonte principal e complementada por coletas no *Pixabay*. Selecionaram-se arquivos em alta resolução ( $\geq 300$  dpi), nos formatos *JPEG* (.jpg) ou *PNG* (.png), com diversidade de ângulos, iluminação e enquadramentos para ampliar a variabilidade do conjunto.

Foram selecionadas 1.476 imagens iniciais, distribuídas entre as variedades *Biquinho*, *Bode*, *Chili*, *Fidalga*, *Habanero*, *Jalapeño*, *Scotch Bonnet* e *Cambuci*, representadas no mosaico de exemplos apresentado na Figura 2.

**Figura 2** - Exemplo de imagens do *dataset* por classe (8 variedades de pimentas)

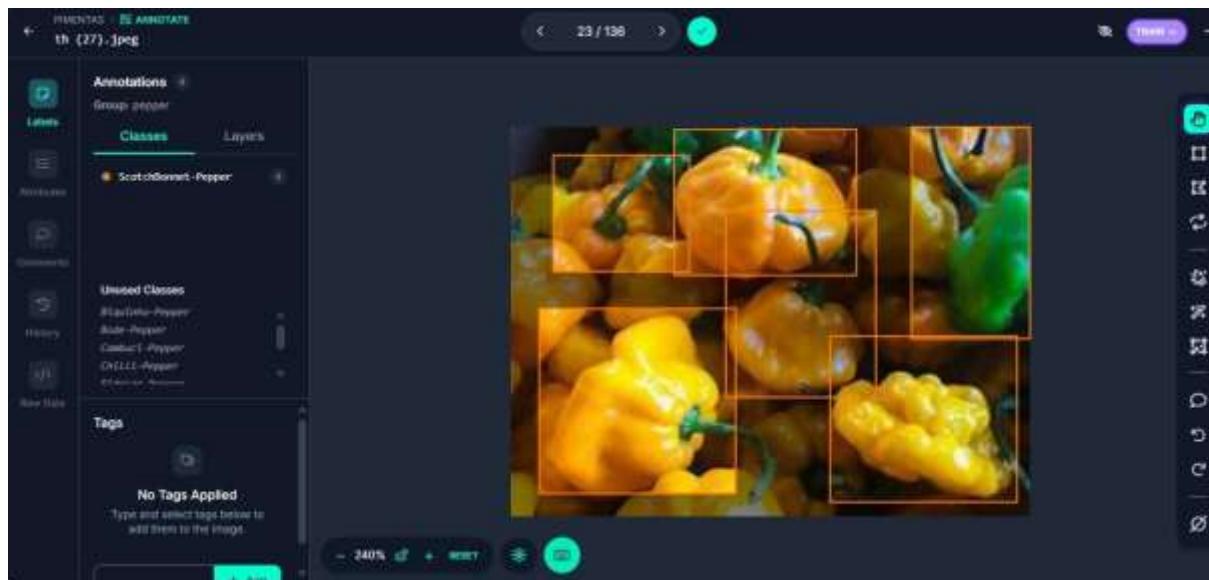


Fonte: Elaboração própria (2025).

### 3.5 Preparação do *dataset*

A rotulagem das imagens coletadas na etapa anterior foi realizada manualmente na plataforma *Roboflow*. Utilizou-se a ferramenta retangular para delimitar cada fruto com uma *bounding box* aderente às bordas visíveis e, em seguida, associou-se a classe correspondente a partir da lista previamente definida. Em cenas com múltiplos frutos, cada instância recebeu uma caixa e um rótulo próprios; em oclusões parciais, anotou-se a porção visível quando suficiente para identificação, conforme Figura 3. Imagens com desfoque severo, exposição inadequada ou ambiguidade de classe foram excluídas. Esse procedimento seguiu as orientações operacionais do *Roboflow* para anotação supervisionada com *bounding boxes* (ROBOFLOW, 2023).

**Figura 3** - Exemplo de anotação com *bounding boxes* no *Roboflow*



Fonte: Elaboração própria (2025).

A padronização do tamanho foi executada no próprio *Roboflow*, com a normalização de todas as imagens para  $640 \times 640$  pixels, em extensão *JPG*, conforme recomendação do *YOLOv8* para assegurar uniformidade e eficiência computacional (Jocher *et al.*, 2022).

Na etapa de pré-processamento foi exportado o conjunto de imagens anotadas no perfil *YOLOv8*. O formato de rótulo adotado foi o nativo do *YOLO*: para cada imagem, um arquivo *.txt* homônimo é criado; cada linha descreve um objeto no padrão *class x\_center y\_center width height*, com as quatro grandezas geométricas normalizadas em  $[0,1]$  em relação à largura e à altura da imagem. A lista de classes e seus nomes é definida no arquivo *data.yaml*, que também referencia os caminhos das partições *train/ val/ test* (ULTRALYTICS, 2023; ROBOFLOW, 2023).

Após a anotação, o *dataset* inicial, composto por 1.476 imagens, foi dividido em 84% para treinamento (1.245 imagens), 14% para validação (201 imagens) e 2% para teste (30 imagens). Priorizou-se maximizar exemplos de treino e manter, em validação e teste, amostras visualmente diversas (iluminação, ângulos e morfologia). A estratégia adotada encontra respaldo nos estudos de Frizzi *et al.* (2021), os quais mostram que, em cenários caracterizados pela escassez de dados, a priorização do volume do conjunto de treinamento, aliada a técnicas de *data augmentation*, atua eficazmente na mitigação do *overfitting* e pressupõe a manutenção da representatividade nos subconjuntos de validação e teste. Apesar da adesão a este princípio, ressalta-se que a dimensão reduzida do conjunto de teste atual impõe

limitações à avaliação definitiva quanto à capacidade de generalização do modelo.

### 3.6 Técnicas de *data augmentation*

Com o objetivo de aumentar a diversidade do *dataset* e melhorar a capacidade de generalização do modelo *YOLOv8*, foram aplicadas técnicas de *data augmentation* diretamente na plataforma *Roboflow*, com a inclusão da rotação aleatória e do desfoque gaussiano. Essas transformações resultaram a geração de um novo *dataset* expandido, o que totalizou 3.964 imagens. A Figura 4 apresenta exemplos dessas transformações.

**Figura 4** - Exemplo de técnicas de *data augmentation* aplicadas ao *dataset*



Fonte: Elaboração própria (2025).

O uso de *data augmentation* é amplamente reconhecido na literatura como estratégia eficaz para reduzir o risco de *overfitting* e aumentar a robustez de modelos de visão computacional (YILMAZ e KUTBAY, 2024). O *overfitting* ocorre quando um modelo apresenta desempenho elevado no conjunto de treinamento, mas perde capacidade de generalização em novos dados, por ter memorizado padrões específicos em vez de aprender características representativas (ZHANG *et al.*, 2021). No presente estudo, foram utilizados os seguintes parâmetros de transformação:

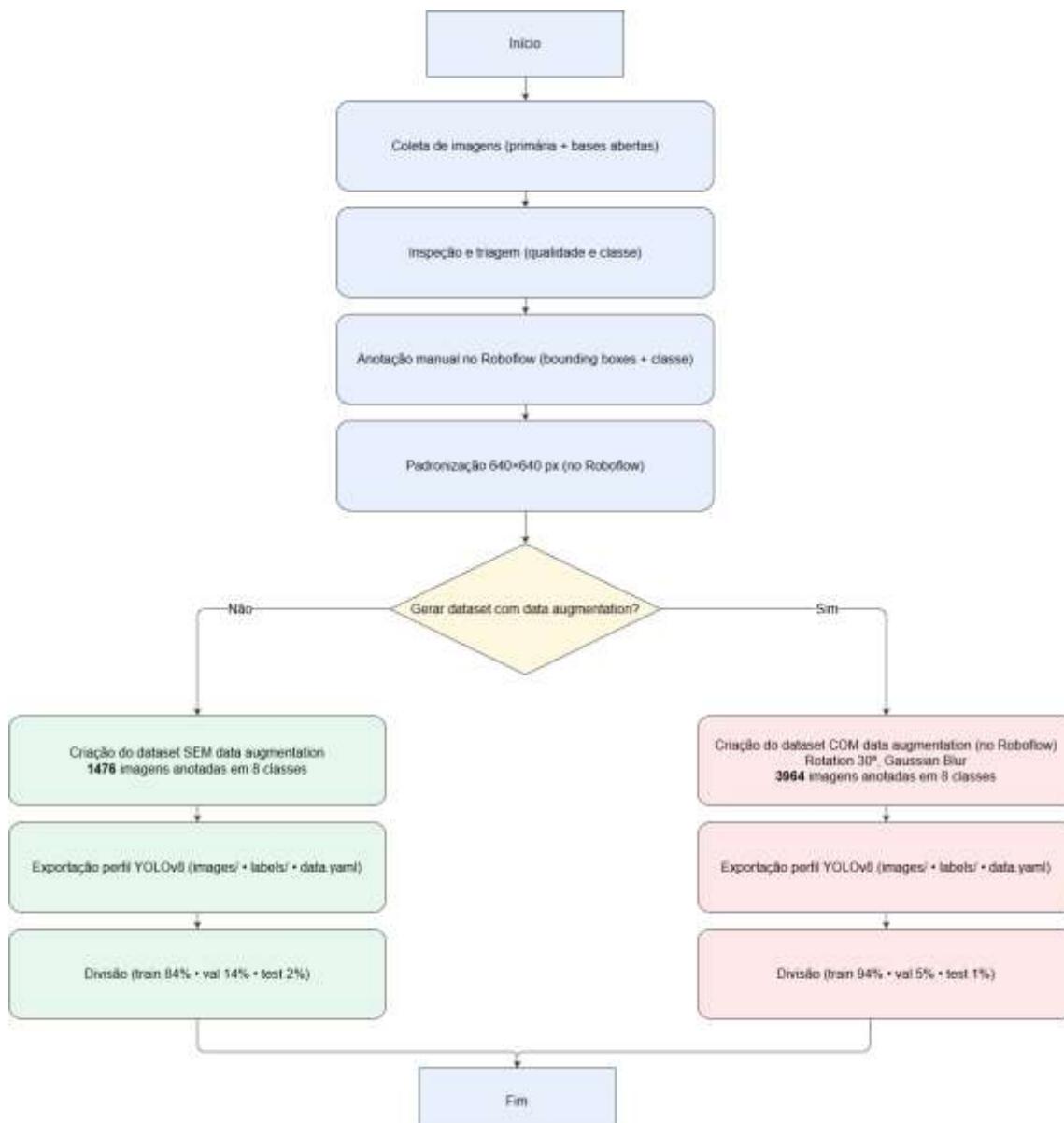
**Rotação aleatória:** variação entre  $-30^\circ$  e  $+30^\circ$ , o que permite simular diferentes ângulos de captura das pimentas;

**Desfoque gaussiano (*gaussian blur*):** o desfoque gaussiano foi aplicado com *kernel*  $3 \times 3$ , com produção de um efeito leve, suficiente para simular pequenas variações de foco que ocorrem em condições reais de captura de imagens, sem comprometer a legibilidade das características visuais das variedades de pimentas.

**Recorte (cropping):** avaliado inicialmente, foi descartado, pois ocasionava a perda de características importantes em variedades com frutos menores.

Essas técnicas ampliaram o *dataset* inicial, com a garantia de maior diversidade visual e de uma base mais robusta para o treinamento do modelo. A Figura 5 sintetiza, de forma gráfica, o fluxo de criação do *dataset*.

**Figura 5** - Fluxograma de preparação do *dataset* e seu particionamento



Fonte: Elaboração própria (2025).

### 3.7 Configuração Computacional

O treinamento foi realizado no *Google Colab Pro*, serviço hospedado de *notebooks Jupyter* que permite executar *Python* no navegador com acesso a *GPUs/TPUs* e instalação de bibliotecas por sessão, o que favorece reproduzibilidade, baixo custo e rápida disponibilização de *hardware* acelerado (*GPU*) para *deep learning* (GOOGLE, 2025a; GOOGLE, 2025b).

A opção pelo *Colab Pro* se justifica por (i) acesso sob demanda a *GPUs NVIDIA* (com a inclusão de T4, L4, V100 e A100, conforme plano), com cobrança previsível e documentação pública de preços; (ii) ambiente gerenciado, com *CUDA* e *drivers* compatíveis; (iii) facilidade de compartilhamento do *notebook* para auditoria do experimento (GOOGLE, 2025c).

No presente estudo, as execuções utilizaram *GPU NVIDIA Tesla T4* (16 GB GDDR6), adequada a tarefas de treinamento de CNNs graças aos Tensor Cores e ao suporte a cálculo multi-precisão (FP32/FP16/INT8), que aumentam a eficiência em *training/inference* (NVIDIA, 2019; NVIDIA, 2024).

O *runtime* foi *Python 3.10* com *PyTorch 2.0* e *Ultralytics YOLOv8*, instalados no próprio *Colab* via *pip*, com atenção aos requisitos mínimos do projeto (*Python*  $\geq 3.8$ ; *PyTorch*  $\geq 1.8$ ; *GPU CUDA*-compatível para aceleração) e ao guia de instalação/uso do *Ultralytics* (ULTRALYTICS, 2023a; ULTRALYTICS, 2023b; PYTORCH, 2025). As execuções iniciavam com verificação de acesso à *GPU* e *seed* para reproduzibilidade.

### 3.8 Procedimentos de treinamento

O treinamento do modelo *YOLOv8* foi conduzido em dois cenários distintos, com e sem *data augmentation*, com o objetivo de comparar o impacto da aplicação dessa técnica sobre o desempenho do modelo e sobre o custo computacional.

Utilizou-se o *YOLOv8m* (detecção *one-stage*) com *imgsz*=640, *batch size* = 16 e *learning rate* inicial = 0,01, mantendo otimizador/agendador padrão do *framework*. O treinamento foi invocado pela API do *Ultralytics* via *model.train* (...), conforme documentação oficial (ULTRALYTICS, 2023a; ULTRALYTICS, 2023e). No Cenário 1 (sem *data augmentation*), foram utilizadas 200 épocas de treinamento, *batch size* de 16 e *learning rate* de 0,01. Já no Cenário 2 (com *data augmentation*), o modelo foi treinado por 170 épocas, com a manutenção do mesmo *batch size* (16) e *learning rate* (0,01).

O parâmetro *batch size* refere-se ao número de amostras processadas simultaneamente antes da atualização dos pesos, enquanto o *learning rate* controla a taxa de atualização dos parâmetros do modelo durante o processo de retropropagação. Ambos os parâmetros exercem influência direta sobre a estabilidade e a eficiência do treinamento (GOODFELLOW, BENGIO e COURVILLE, 2016; ZHANG *et al.*, 2021). Os parâmetros adotados em cada cenário estão resumidos na Tabela 1.

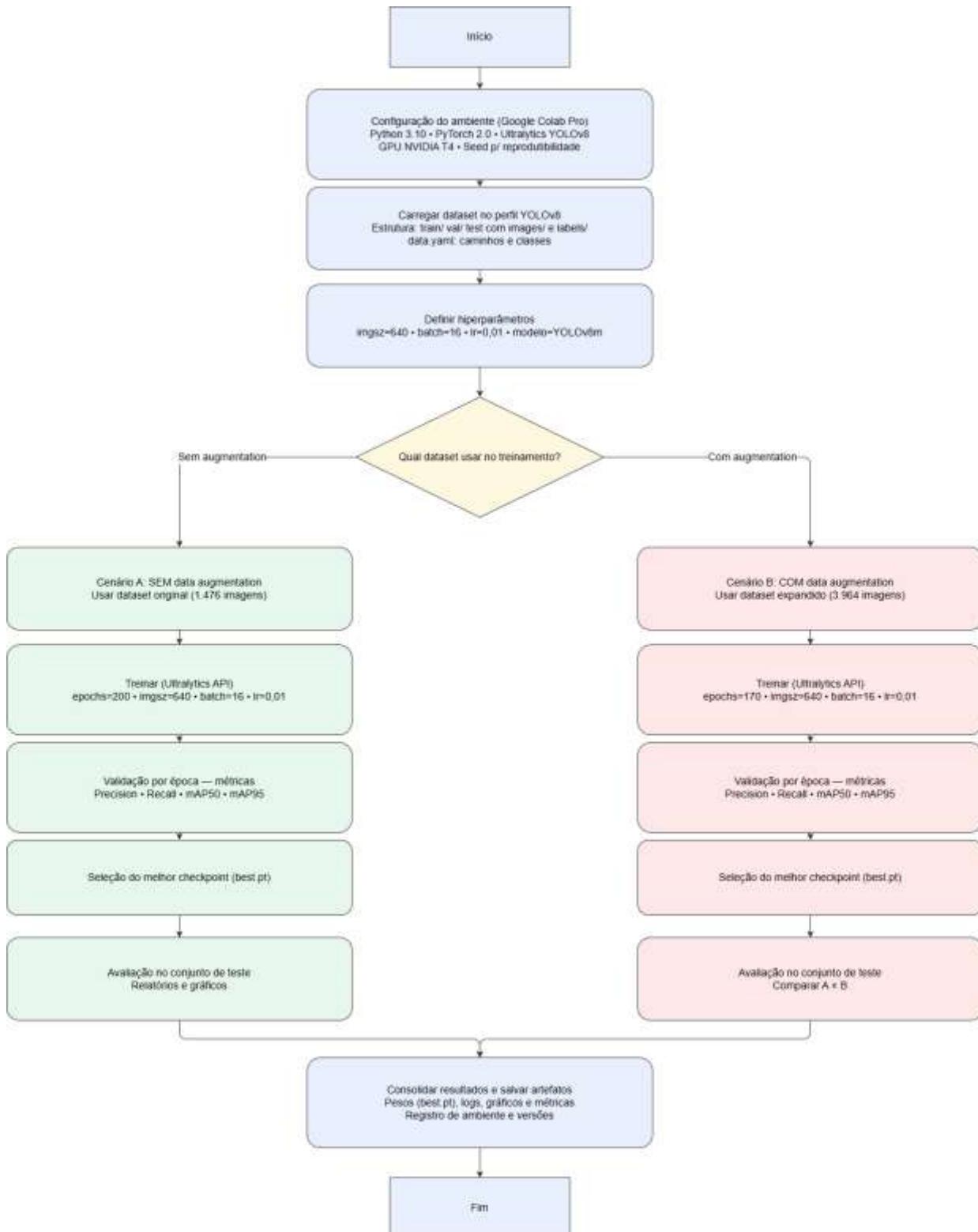
**Tabela 1** - Parâmetros de treinamento do YOLOv8 com e sem *data augmentation*

size	Cenários	Épocas	Batch size	Learning rate	Tempo total
	Sem <i>data augmentation</i>	200	16	0,01	3h04min
	Com <i>data augmentation</i>	170	16	0,01	6h34min

Fonte: Elaboração própria (2025).

Para tornar explícitos os procedimentos empregados no treinamento, a Figura 6 sintetiza o fluxo adotado.

**Figura 6 - Fluxograma de treinamento do modelo YOLO8m no Google Colab Pro**



Fonte: Elaboração própria (2025).

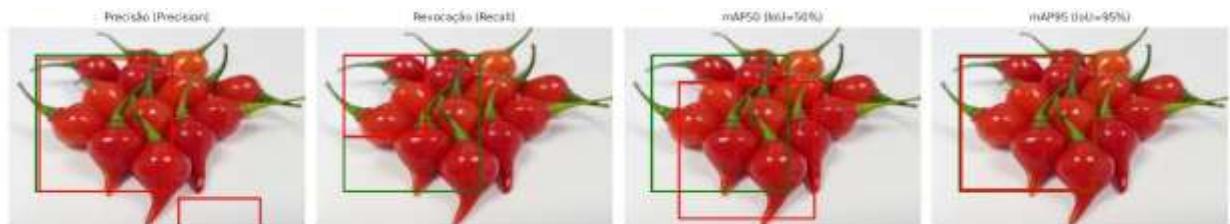
### 3.9 Métricas de Avaliação

A mensuração do desempenho do modelo YOLOv8 foi realizada no ambiente *Google Colab Pro*, com a utilização dos módulos de validação da biblioteca *Ultralytics*, extraídos diretamente dos relatórios de execução gerados pelo *framework*.

Esses resultados são apresentados pela ferramenta sob duas formas complementares: (i) registros tabulares (*logs*), que detalham os valores de precisão e *recall* ao final de cada época de treinamento; (ii) artefatos gráficos, com inclusão de curvas de aprendizado (*F1-score*, *Precision-Recall*) e matrizes de confusão, salvos automaticamente ao término do processo. Com base nesses dados brutos, foi possível compilar as métricas consagradas na literatura de visão computacional (PADILLA *et al.*, 2021) para avaliar a qualidade das detecções.

A Figura 7 ilustra visualmente os conceitos de Precisão, Revocação e os limiares de Interseção sobre União (IoU) adotados nesta pesquisa. Nela, os retângulos verdes representam as caixas delimitadoras reais (anotadas manualmente); os retângulos vermelhos indicam as predições realizadas pelo modelo.

**Figura 7-** Representação das métricas de avaliação (Precisão, *Recall*, *mAP50* e *mAP95*)



Fonte: Adaptado de Padilla *et al.* (2021).

Com base na representação visual da Figura 7 e nas definições estabelecidas por Padilla *et al.* (2021), as métricas são definidas como:

**Precisão (Precision):** ilustrada no primeiro quadro da Figura 7, esta métrica avalia a confiabilidade das detecções positivas. Ela mensura a proporção de predições corretas (Verdadeiros Positivos) em relação ao total de caixas geradas pelo modelo. Em termos práticos, uma alta precisão indica que o modelo minimiza a ocorrência de falsos positivos, ou seja, "não inventa" pimentas onde elas não existem.

**Revocação (Recall):** representada no segundo quadro, a revocação (ou sensibilidade) quantifica a capacidade do modelo em encontrar todas as instâncias

dos objetos de interesse presentes na imagem. Ela é calculada pela razão entre os acertos do modelo e o número total de objetos reais (*ground truth*). Um alto valor de *recall* indica que o sistema é eficaz em não omitir pimentas (minimiza falsos negativos)

***mAP50* (Média da Precisão com IoU  $\geq 50\%$ ):** conforme observado no terceiro quadro da Figura 7, o IoU (Interseção por União) calcula a área de interseção dividida pela área de união entre a caixa predita e a real. O *mAP50* representa a média da precisão para todas as classes; considera como "acerto" apenas as previsões que apresentam uma sobreposição de área superior a 50% com a anotação original. Este é o limiar padrão para avaliações de modelos de identificação.

***mAP95* (Média da Precisão com IoU 50-95%):** Ilustrada no quarto quadro, esta é uma métrica de rigor elevado. Ela calcula a média da precisão na variação do limiar de IoU de 0,50 a 0,95, com incrementos de 0,05. Visualmente, isso exige que a caixa vermelha (previsão) esteja quase perfeitamente alinhada à caixa verde (real). Esta métrica é fundamental para avaliar a precisão da localização geométrica do objeto. Ela penaliza detecções que, embora classifiquem corretamente a variedade, não delimitam perfeitamente as bordas do fruto.

### 3.10 Desenvolvimento do protótipo do aplicativo *App Pimentas*

Esta seção descreve, em nível operacional, os procedimentos adotados para desenvolvimento e disponibilização do protótipo do aplicativo *web App Pimentas*. Ao longo do texto, o termo “aplicativo” é utilizado para se referir a esse protótipo funcional, que constitui o artefato da pesquisa em *Design Science*.

Buscou-se registrar o percurso técnico seguido na pesquisa de forma que ele possa ser compreendido e, em princípio, repetido por outros interessados, com foco na transparência e na reproduzibilidade do artefato. No âmbito da *Design Science Research (DSR)*, a explicitação do processo de construção, do ambiente técnico e das decisões de projeto integra o rigor da pesquisa (HEVNER *et al.*, 2004; HEVNER, 2007; HEVNER, 2024).

O aplicativo, em si, tem como objetivo principal identificar 8 variedades de pimentas a partir de imagens e disponibilizar informações associadas ao uso culinário e à ardência. A metodologia, por sua vez, busca descrever este processo desde o uso do modelo *YOLOv8* exportado em *ONNX* até a interface *web progressiva* e a

implantação em nuvem (MAXWELL; BESTER; RAMEZAN, 2022).

### 3.10.1 Arquitetura geral e tentativa inicial com Kodular

Inicialmente, foi investigada a possibilidade de disponibilizar o modelo em um aplicativo Android nativo criado no *Kodular*. *Kodular* é uma plataforma visual *no-code* para desenvolvimento de aplicativos Android, baseada em componentes gráficos e blocos de programação, o que permite criar e compilar pacotes de instalação (arquivos *APK*, *Android Package*) sem necessidade de escrever código textual (KODULAR, 2025).

Durante os testes em dispositivos Android, observou-se, porém, que as chamadas de câmera realizadas via navegador embutido (*WebView*) não acionavam corretamente o pedido de permissão do sistema operacional, o que impedia a captura de imagem no *App*, apesar de a mesma página funcionar normalmente no navegador *Chrome* do mesmo aparelho. Essa limitação é compatível com relatos técnicos sobre restrições de acesso à câmera e uso de *getUserMedia* em alguns contextos de *WebView*, que podem exigir configurações adicionais ou não oferecer suporte completo à *API* (ROUMELIOTIS; TSELIKAS, 2022).

Diante dessa evidência empírica, optou-se por não prosseguir com o empacotamento via *Kodular*, mas adotar uma arquitetura em que o usuário acessa diretamente a aplicação pelo navegador do dispositivo, sem intermediação de *WebViews*. O registro dessa tentativa e de sua limitação técnica contribui para explicitar o caminho que levou à solução adotada.

Na configuração consolidada, o protótipo passou a ser organizado em três camadas:

**Backend de inferência** em *Python*, com a execução do modelo *YOLOv8m* exportado para o formato *ONNX*;

**Interface web em formato *Progressive Web App (PWA)***, responsável pela interação com o usuário, captura/envio de imagens e exibição dos resultados;

**Infraestrutura de versionamento e implantação**, baseada em repositório *GitHub* e serviço em nuvem do tipo *Platform as a Service (PaaS)*, neste caso a plataforma *Render*.

### 3.10.2 Backend de inferência em *Python*, *FastAPI* e *ONNX*

O *backend*, isto é, a parte da aplicação responsável pelo processamento no servidor e pela disponibilização das rotas *HTTP*, foi desenvolvido em *Python*, com a utilização do *framework* *FastAPI*. Este é um *framework* moderno para criação de *APIs REST* em *Python*, que, na sua vez, se apoia em anotações de tipo (*type hints*) para validação automática de dados. Ele oferece boa performance em servidores assíncronos, além de gerar documentação automática em formato *OpenAPI* (TIANGOLO, 2018).

O modelo de detecção utilizado é o YOLOv8m, na configuração padrão de arquitetura disponibilizada pela *Ultralytics*, previamente treinado e ajustado neste estudo para a detecção das classes de pimentas consideradas (Seção 3.x), e posteriormente exportado para o formato *Open Neural Network Exchange* (*ONNX*).

O *ONNX* é um padrão aberto para representação de modelos de aprendizado de máquina, concebido para permitir interoperabilidade entre diferentes *frameworks* e facilitar a execução em ambientes variados, como servidores, dispositivos de borda e aceleradores de *hardware* (BAI *et al.*, 2019; MICROSOFT, 2024). A execução do modelo em formato *ONNX* é realizada por meio da biblioteca *Ultralytics YOLO*, em conjunto com o *ONNX Runtime*, *runtime* de alto desempenho para inferência de modelos *ONNX*, mantido pela Microsoft (MICROSOFT, 2024).

No arquivo *main.py* são definidos:

- o caminho local do modelo (*MODEL\_PATH*) e a *URL* remota (*MODEL\_URL*), da qual o arquivo *best.onnx* é baixado na primeira execução;
- a função *ensure\_model\_file()*, responsável por verificar a existência do arquivo local e efetuar o *download* quando necessário;
- uma rotina de carregamento em segundo plano (*background*), que é executada automaticamente ao iniciar o servidor, que baixa e carrega o modelo *ONNX* em memória, com o registro de variáveis globais de estado (READY e LOAD\_ERR).

Dessa forma, ao utilizar o mesmo código-fonte e a mesma *URL* do arquivo *best.onnx*, configurada em *MODEL\_URL*, diferentes instalações do sistema passam

a usar exatamente o mesmo modelo de detecção, o que está em linha com recomendações de boas práticas para reproduzibilidade e replicabilidade em experimentos de *deep learning* (MAXWELL; BESTER; RAMEZAN, 2022).

As principais rotas expostas pelo *backend* são

- POST `/predict`: recebe um arquivo de imagem (*UploadFile*); converte-o para o formato de cores *RGB* (vermelho, verde e azul); reduz a imagem de modo que o maior lado tenha no máximo 1.024 pixels; executa a inferência com o modelo *YOLOv8m*. A resposta da *API* é um objeto *JSON* que contém, para cada detecção, a classe prevista, o escore de confiança e as coordenadas da caixa delimitadora (*bounding box*);
- *GET /ui*: entrega o documento *HTML* que compõe a interface de identificação de pimentas;
- *GET /info*: entrega a interface do tipo *chat* que consome o arquivo estático *pepper\_info.json* para exibir informações detalhadas sobre cada variedade.

As rotas foram organizadas de modo a permitir que a *API* seja acessada e testada de forma direta (por exemplo, por meio de ferramentas de inspeção de requisições *HTTP*). Isto facilita a verificação do seu funcionamento em diferentes ambientes, em linha com recomendações de reproduzibilidade em *workflows* de aprendizado de máquina (MAXWELL; BESTER; RAMEZAN, 2022).

### 3.10.3 Interface web e Progressive Web App (PWA)

A interface do *Aplicativo Pimentas* foi desenvolvida como um *Progressive Web App (PWA)*. Nessa abordagem, a aplicação é escrita com tecnologias *web* (*HTML*, *CSS* e *JavaScript*), mas passa a se comportar de forma semelhante a um aplicativo nativo. Pode ser instalada na tela inicial, pode funcionar com conectividade limitada. Isso é viabilizado pelo uso de um *service worker*, *script* que roda em segundo plano no navegador e controla o *cache* da aplicação, em conjunto com um mecanismo de armazenamento em *cache* e uma estrutura básica de interface (*app shell*) (ROUMELIOTIS; TSELIKAS, 2022).

No projeto, a *PWA* é composta por

- um arquivo de manifesto (*manifest.webmanifest*), configurado com o nome

do aplicativo, ícones, cores de tema e *URL* de início;

- um *service worker* (*sw.js*), responsável por interceptar requisições e gerenciar o *cache* dos recursos essenciais;
- arquivos *HTML* e *JavaScript* que controlam a interação no navegador: exibem os botões para tirar foto com a câmera ou selecionar uma imagem da galeria; os arquivos obtêm essa imagem, reduzem sua resolução no próprio navegador (com o uso de um elemento *canvas*) e a empacotam em um formulário (*FormData*) para enviá-la, por meio de uma requisição *HTTP POST*, ao *endpoint* */predict*. Em seguida, o código *JavaScript* recebe a resposta da *API* em formato *JSON*; interpreta os campos retornados (classe prevista, escore de confiança e, quando disponível, a *URL* da imagem anotada) e atualiza a interface; mostra o nome da pimenta identificada e a respectiva imagem ao usuário.

O *service worker* utiliza uma estratégia de *cache* do tipo *app shell*, na qual os arquivos básicos da interface são guardados localmente no navegador. No momento em que o *service worker* é instalado (evento *install*), ele pré-carrega um conjunto de caminhos essenciais, como */ui* e */info*, o arquivo de manifesto, os ícones e o *pepper\_info.json*. Em seguida, os armazena em um *cache* identificado pelo nome *CACHE\_NAME*.

Quando o usuário navega pelo aplicativo, cada solicitação de recurso faz com que o navegador dispare o evento *fetch*. Nessa etapa, o código do *service worker* verifica o tipo de requisição e decide se o recurso será obtido diretamente da rede ou atendido a partir do conteúdo já armazenado em cache, com a aplicação de regras distintas para as rotas de interface e para o *endpoint* de predição (*/predict*).

As regras implementadas no *service worker* para o atendimento das requisições são

- **Chamadas *POST* para */predict*:** são sempre encaminhadas diretamente à rede, sem uso de *cache*, de modo que cada requisição de predição utilize o resultado atual do modelo, sem reutilizar respostas anteriores.
- **Requisições para as rotas de interface (*/ui*, */info*):** adotam uma estratégia em que o navegador tenta obter o recurso primeiro na rede (*network first*); se a

conexão falhar, o *service worker* retorna a versão armazenada em *cache*, o que permite que a interface continue acessível mesmo com instabilidade de conexão.

- **Requisições de recursos estáticos (ícones, arquivos JSON, CSS):** utilizam uma estratégia em que o *service worker* procura primeiro no *cache* (*cache first*); se o recurso não estiver disponível localmente, a busca é feita na rede (*fallback*), e o resultado é então armazenado em *cache* para usos futuros.

Essa forma de uso do *service worker* e do *cache* segue a ideia de *PWA* descrita na literatura, em que aplicações *web* são configuradas para funcionar em diferentes dispositivos e em situações de conexão instável, e elas se mantêm baseadas em tecnologias *web* (ROUMELIOTIS; TSELIKAS, 2022).

Na tela principal da *PWA*, o usuário encontra

- botão para captura de imagem pela câmera do dispositivo;
- botão para envio de uma imagem da galeria;
- área para visualização da imagem enviada e, quando habilitado, da imagem anotada com *bounding boxes*;
- texto com a classe de pimenta identificada e o respectivo nível de confiança.

Na Figura 8, apresenta-se a tela inicial do *App Pimentas*, com os botões de escolha de imagem e abertura da câmera, bem como a visualização da imagem original e do resultado da detecção de pimentas.

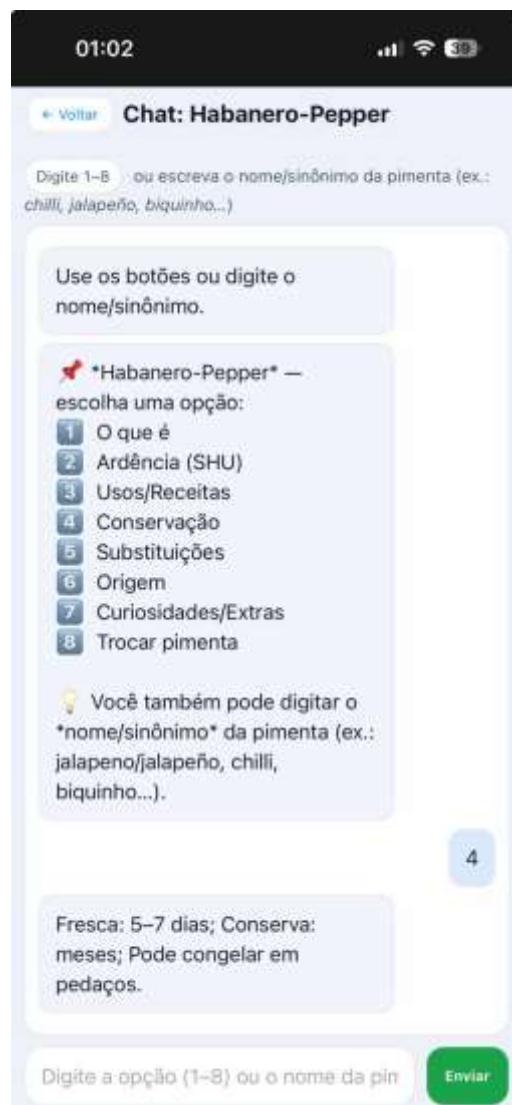
**Figura 8** - Tela inicial de identificação de pimentas do App Pimentas



Fonte: Elaboração própria (2025).

Na tela de informações complementares, acessada após o processo de identificação, o protótipo abre automaticamente um *chat* associado à variedade de pimenta reconhecida pelo modelo na etapa anterior. Quando apenas uma classe é detectada, o *chat* já é iniciado diretamente com essa pimenta; caso mais de uma classe seja identificada na mesma imagem, o usuário pode escolher, na própria interface, qual variedade deseja explorar primeiro. A partir desse ponto, o usuário pode navegar entre diferentes tipos de pimenta por meio de opções no próprio *chat*, com consulta aos tópicos, como definição, ardência, usos culinários, conservação e curiosidades, sem necessidade de voltar à tela de identificação. A Figura 9 apresenta a interface do *chat* de informações para a pimenta *Habanero-Pepper*.

**Figura 9** - Tela de chat de informações do *App Pimentas* (com exemplo para a pimenta Habanero-Pepper)



Fonte: Elaboração própria (2025).

### 3.10.4 Versionamento no *GitHub* e implantação na *Render*

Para organizar o código e apoiar a replicação do protótipo, foi utilizado um repositório na plataforma **GitHub**, baseada no sistema de controle de versão distribuído *Git*. Repositórios *Git* têm sido amplamente adotados em ciência de dados e aprendizado de máquina justamente por permitirem rastrear mudanças, associar resultados a *commits* específicos e facilitar a colaboração (MAXWELL; BESTER; RAMEZAN, 2022).

No repositório estão incluídos

- o arquivo principal da *API* (*main.py*);

- os arquivos de configuração do ambiente (*requirements.txt* e *runtime.txt*);
- os arquivos estáticos da *PWA* (*manifest.webmanifest*, *sw.js*, ícones, *HTML* e *JavaScript*);
- o arquivo *pepper\_info.json*, com as informações textuais sobre as pimentas.

O aplicativo foi implantado na plataforma *Render*, que é um serviço de *Platform as a Service (PaaS)* para hospedagem de aplicações *web*. Plataformas *PaaS* abstraem parte da complexidade de configuração de servidores (sistema operacional, servidor de aplicação, bibliotecas) e permitem reproduzir ambientes de execução a partir de configurações declarativas (MICROSOFT, 2024).

Na configuração adotada, a plataforma *Render* está vinculada ao repositório do projeto no *GitHub*. Sempre que há uma nova atualização no *branch* configurado, a *Render* cria um ambiente com *Python 3.10*, instala as dependências listadas em *requirements.txt* e inicia o servidor *FastAPI* com um comando do tipo *unicorn main:app*. Variáveis de ambiente, como *MODEL\_URL* e *HF\_TOKEN*, são registradas diretamente no painel da *Render*, o que evita que esses valores apareçam nos arquivos do repositório. Essa combinação entre código versionado e uso de um serviço *PaaS* contribui para organizar o processo de implantação e oferece um caminho relativamente simples para que outro pesquisador configure um ambiente de execução equivalente.

### 3.10.5 Procedimento de replicação do protótipo

1. Obter o código-fonte
  - Acessar o repositório do projeto no *GitHub* (conforme Apêndice B), referente ao *commit* utilizado nesta dissertação, e clonar o repositório ou baixar o pacote de código.
2. Configurar o ambiente *Python*
  - Instalar *Python 3.10*;
  - Criar um ambiente virtual e instalar as dependências com *pip install -r requirements.txt*.

### 3. Configurar o modelo ONNX

- Definir a variável de ambiente *MODEL\_URL* apontando para o arquivo *best.onnx* (modelo treinado em *YOLOv8m*);
- Opcionalmente, configurar *token* de acesso para o repositório do modelo, se necessário.

### 4. Executar o *backend* localmente

- Iniciar o servidor com *uvicorn main:app --host 0.0.0.0 --port 8000*;
- verificar, em *http://localhost:8000/*, o status de carregamento do modelo.

### 5. Acessar a interface *PWA*

- Abrir *http://localhost:8000/ui* em um navegador;
- Instalar o *app* na tela inicial, caso o navegador ofereça essa opção;
- Testar o envio de imagens e o retorno dos resultados de inferência.

### 6. Validar a tela de informações complementares

- Acessar *http://localhost:8000/info* e verificar o carregamento do conteúdo de *pepper\_info.json*;
- Conferir se as informações exibidas correspondem às classes de pimentas utilizadas no modelo.

Embora o *App Pimentas* tenha sido registrado como programa de computador, o código-fonte utilizado nesta pesquisa foi disponibilizado em repositório *Git* na plataforma *GitHub*, conforme indicado no Apêndice B. Essa opção permite que outros pesquisadores accessem a implementação do protótipo, com a inclusão de arquivos de código, configurações de ambiente e o *commit* específico associado a esta dissertação, o que facilita a reconstrução do ambiente descrito nos passos anteriores e a realização de estudos posteriores a partir do mesmo ponto de partida.

A descrição detalhada desses passos, associada à disponibilização do código e à indicação da versão específica do modelo ONNX, vai ao encontro de recomendações recentes sobre transparência e reproduzibilidade em DSR e em

*workflows* de aprendizado de máquina (HEVNER; VOM BROCKE, 2023; MAXWELL; BESTER; RAMEZAN, 2022).

Adicionalmente, o protótipo descrito nesta seção foi objeto de pedido de registro como programa de computador junto ao Instituto Nacional da Propriedade Industrial (INPI), sob o título ***App Pimentas***, processo nº **512025006150-1**, com petição eletrônica nº **870250108484**, protocolada em 26 nov. 2025. Esse registro tem a finalidade de resguardar a autoria e os direitos relativos ao software e não impede a disponibilização do código-fonte em repositório *Git* para fins de pesquisa, conforme indicado no Anexo B.

Concluída a etapa de implementação, o protótipo será disponibilizado para validação prática junto à equipe parceira da CEAGESP, a fim de validar sua aplicabilidade em ambiente ~~real~~, coletar *feedback* de usuários e orientar melhorias futuras.

### **3.11 Procedimento de validação do protótipo**

Para uma avaliação inicial do protótipo no contexto de uso, foi realizado contato com a equipe do Centro de Qualidade Hortigranjeira da CEAGESP. Foi enviado um e-mail ao chefe da seção, com o *link* de acesso ao aplicativo *App Pimentas* e o *link* para um questionário eletrônico de avaliação elaborado na plataforma *Microsoft Forms* (ver Apêndice A).

O instrumento utilizado foi um questionário estruturado, em formato digital, composto por uma breve apresentação dos objetivos do estudo e instruções de preenchimento, seguido de uma escala de concordância de cinco pontos (1 = discordo totalmente; 5 = concordo totalmente). O questionário foi organizado em quatro blocos temáticos: (i) **Usabilidade e experiência do usuário**, com itens sobre facilidade de aprendizagem, navegação entre telas e uso da câmera nas condições de trabalho (galpão/box); (ii) **Desempenho e precisão da inteligência artificial**, com abordagem de acerto na identificação do tipo de pimenta, na capacidade de classificar lotes mistos ou parcialmente cobertos e no alinhamento com a classificação padrão utilizada na CEAGESP; (iii) **Desempenho operacional e confiança**, com questões relativas à latência entre a captura da foto e a apresentação do resultado e ao grau de confiança do usuário na classificação para fins de decisão (por exemplo, destinação ou preço); (iv) **Utilidade percebida e intenção de uso**, que trata da

estabilidade do aplicativo, da utilidade da informação de classificação para a atividade profissional, da percepção de agregação de valor à conferência ou negociação e da intenção de uso futuro do aplicativo.

O questionário foi configurado para não coletar automaticamente dados pessoais dos respondentes (como nome ou endereço de e-mail) e não incluiu questões específicas de caracterização de perfil (por exemplo, idade, escolaridade ou função detalhada). O foco do instrumento esteve na avaliação da usabilidade, do desempenho percebido do modelo e da utilidade prática do protótipo no contexto de trabalho.

Até a data de fechamento desta dissertação, o questionário não havia recebido respostas válidas por parte da equipe do Centro de Qualidade Hortigranjeira da CEAGESP. Dessa forma, não foram produzidos resultados quantitativos de validação em campo, e a avaliação sistemática com usuários externos permanece como etapa pendente. Ainda assim, a descrição do instrumento e do procedimento de convite é mantida nesta seção como registro metodológico, e é retomada no Capítulo V como limitação do estudo e recomendação para trabalhos futuros.

## CAPÍTULO IV

### 4 RESULTADOS E DISCUSSÕES

#### 4.1 *Dataset*

Este item atende ao objetivo específico A, que consistiu em construir e organizar um *dataset* de imagens de variedades de *Capsicum spp.*. Foram coletadas 1.476 imagens na CEAGESP e em repositórios públicos, posteriormente anotadas e expandidas por técnicas de *data augmentation*, o que resultou 3.964 imagens. O *dataset* foi organizado em partições para treinamento, validação e teste, e está disponível para *download* no *link*: <https://figshare.com/s/22c9578a30a4c5409bc7>, fato que contribui para a replicabilidade da pesquisa. Esse resultado reforça a relevância do estudo ao fornecer uma base inédita dedicada especificamente à classificação automática de pimentas.

A criação do *dataset* preenche uma lacuna identificada na literatura (DALAL e MITTAL, 2025; KHAN, SHEN e LIU, 2025), em que a ausência de bases de dados específicas para pimentas limitava experimentos reproduzíveis. Além disso, o resultado conecta-se ao ODS 9 (Indústria, Inovação e Infraestrutura) ao disponibilizar infraestrutura científica aberta que pode ser reutilizada por outros pesquisadores.

A construção deste *dataset* não apenas atendeu ao objetivo específico A, mas também foi o alicerce para o desenvolvimento do objetivo específico B. O artigo científico apresentado na seção seguinte só pôde ser realizado graças à existência desse repositório de imagens, que viabilizou o treinamento, a avaliação e a validação experimental do modelo YOLOv8.

#### 4.2 Artigo Científico - Identificação Automatizada de Variedades de Pimentas com YOLOv8

Este item responde ao objetivo específico B, que consistiu em avaliar o desempenho da arquitetura YOLOv8 em cenários com e sem *data augmentation*. Apresenta o artigo científico intitulado "YOLOv8 for Automated Pepper Variety Identification: Improving Accuracy with Data Augmentation". O trabalho foi desenvolvido com o objetivo de aplicar e avaliar o desempenho da arquitetura de rede

neural convolucional YOLOv8 na identificação automatizada de variedades de pimentas no contexto pós-colheita.

A pesquisa valida o uso do modelo YOLOv8 para aprimorar o controle de qualidade no setor agroalimentar e, por isto, aborda a necessidade de identificação de oito variedades de pimentas: *Biquinho*, *Bode*, *Chili*, *Fidalga*, *Habanero*, *Jalapeño*, *Scotch Bonnet* e *Cambuci*. Para isso, foi construído e utilizado um *dataset* com 1.476 imagens anotadas, que foi se expandido por meio de técnicas de aumento de dados para otimizar a performance do modelo. Conforme detalhado no artigo, o treinamento com o *dataset* aumentado resultou melhorias significativas em indicadores-chave de desempenho, como *box precision*, *recall* e *mean average precision (mAP)*.

Esse resultado indica que, apesar de os desafios de variabilidade morfológica e iluminação em ambientes comerciais serem significativos (KHAN, SHEN e LIU, 2025), técnicas de *augmentation* podem ampliar a capacidade de generalização do modelo e mitigar riscos de *overfitting*. A contribuição científica é reforçada pelo reconhecimento em publicação revisada por pares, enquanto a contribuição prática está em oferecer um modelo mais robusto para ambientes reais. Em termos de impacto, conecta-se ao ODS 2 (Fome Zero e Agricultura Sustentável), ao apoiar processos de classificação mais confiáveis que podem reduzir perdas pós-colheita.

O artigo foi aceito e publicado na revista *Applied Sciences* (v. 15, e7024, 2025.) em 22 de junho de 2025. A íntegra do artigo publicado está disponível no ANEXO A.

Os resultados obtidos apresentados no artigo indicam a aplicabilidade da arquitetura YOLOv8 para a identificação de pimentas. A métrica geral de *mAP50* alcançou 0.694 com o uso de data *augmentation*. Este valor deve ser analisado no contexto da complexidade da tarefa, que envolve a classificação de grão fino (*fine-grained classification*), um desafio em visão computacional em que as classes apresentam elevada similaridade visual e a distinção depende de características sutis (WEI *et al.*, 2022), entre oito variedades com elevada similaridade visual. Diante deste cenário, um desempenho de 0.694 sugere que o modelo foi capaz de aprender características distintivas e estabeleceu um ponto de partida para a validação da tecnologia.

A análise por classe indica diferenças relevantes de desempenho entre as variedades. A Cambuci-Pepper, cuja morfologia é mais singular em relação às demais, apresentou mAP50 de 0,827. Essa diferença é compatível com resultados de tarefas de classificação de imagens em que classes visualmente mais distintas tendem a ser melhor discriminadas. Já as categorias com alta semelhança visual concentram as principais confusões de classe, como relatado em problemas de reconhecimento de caracteres manuscritos e de espécies vegetais com grande similaridade morfológica (SAYEED et al., 2021; RANI et al., 2025).

Em contraste, no caso da Fidalga-Pepper, observou-se redução da BoxP de 0,768 para 0,652 após a aplicação do data augmentation. Uma hipótese plausível é que, em variedades muito semelhantes entre si, determinadas transformações de aumento de dados possam enfatizar padrões visuais comuns e atenuar pistas morfológicas distintivas, o que aumenta a confusão entre classes. Estudos recentes destacam que o efeito do data augmentation é dependente do contexto: embora frequentemente melhore o desempenho, em alguns cenários pode não produzir ganhos ou pode degradar métricas específicas. Isto dependerá da combinação de técnicas utilizada e da arquitetura do modelo (ALOMAR; AYSEL; CAI, 2023; OMONIYI et al., 2025).

Esses achados são consistentes com evidências de que a combinação entre alta similaridade visual entre classes e estratégias de data augmentation não adaptadas ao domínio pode aumentar confusões em tarefas de classificação fina (ALOMAR; AYSEL; CAI, 2023; RANI et al., 2025).

A pesquisa contribui para a literatura ao apresentar um *dataset* público para a classificação de *Capsicum spp.*, uma lacuna apontada por Khan, Shen e Liu (2025) e Dalal e Mittal (2025). A disponibilização dos dados alinha-se ao ODS 9 (Indústria, Inovação e Infraestrutura), ao fomentar a replicabilidade e o avanço de novas pesquisas.

A validação do YOLOv8 como ferramenta aplicável reforça o potencial da visão computacional na Agricultura 4.0, conforme discutido por Javaid et al. (2022) e Lezoche et al. (2020). A automação da classificação em centrais, como a CEAGESP, pode mitigar gargalos logísticos e erros de inspeção manual e impactar positivamente o ODS 12 (Consumo e Produção Responsáveis).

As limitações da pesquisa devem ser reconhecidas. O conjunto de teste, embora diverso, é numericamente reduzido, uma vez que a estratégia priorizou um maior volume de dados para o treinamento do modelo. Adicionalmente, mesmo que a coleta de imagens tenha ocorrido em um ambiente comercial real (CEAGESP), a metodologia buscou amostras com maior "uniformidade de iluminação e enquadramento". Essa abordagem, necessária para garantir a qualidade do *dataset* inicial, pode não ter capturado toda a extensão de condições adversas encontradas no ambiente do entreposto, como sombras intensas, oclusões parciais ou baixa luminosidade. Portanto, a generalização do modelo para condições de captura totalmente irrestritas ainda representa um campo para validação futura.

Apesar dessas limitações, os resultados alcançados indicam o potencial do YOLOv8 como solução viável para a classificação de pimentas. Essa validação experimental constitui a base para o terceiro objetivo específico da pesquisa, que consiste no desenvolvimento de um protótipo de aplicativo móvel. A transição do ambiente de testes para uma solução prática busca ampliar a utilidade do estudo. Com isto, disponibiliza-se o modelo em uma ferramenta acessível para produtores, comerciantes e consumidores na CEAGESP, com funcionalidades adicionais de apoio por meio de um *chatbot* informativo.

### 4.3 Protótipo do aplicativo *App Pimentas*

Este item apresenta os resultados relacionados ao objetivo específico C, que consistiu em desenvolver um protótipo de aplicativo integrado ao modelo YOLOv8m para identificação automática de variedades de pimentas comercializadas na CEAGESP. O protótipo foi implementado como aplicação *web progressiva (PWA)*, conforme descrito no Capítulo 3, e implantado em ambiente de produção na plataforma *Render*, a partir do repositório versionado do projeto.

Do ponto de vista funcional, o protótipo permite que o usuário capture uma imagem pela câmera do dispositivo ou selecione uma foto previamente armazenada na galeria, envie essa imagem ao *backend* de inferência e visualize, na própria interface, a variedade de pimenta identificada e o nível de confiança associado à predição. Quando a opção está habilitada, a aplicação também exibe a imagem anotada com as caixas delimitadoras (*bounding boxes*) sobre os frutos detectados.

Na tela principal do aplicativo, o usuário encontra (i) botão para captura de

imagem pela câmera do dispositivo; (ii) botão para envio de uma imagem da galeria; (iii) área para visualização da imagem enviada e, quando habilitado, da imagem anotada com *bounding boxes*; (iv) texto com a classe de pimenta identificada e o respectivo nível de confiança. Essa tela inicial do protótipo já foi apresentada na Figura 8 (Seção 3.10), que ilustra o fluxo de identificação de pimentas em execução em dispositivo móvel.

Depois de realizada a predição, o protótipo abre automaticamente a tela de informações complementares, em formato de *chat*, vinculada à variedade de pimenta identificada. Quando há apenas uma classe detectada, o *chat* é carregado diretamente com essa pimenta. Quando o modelo retorna mais de uma classe, a interface oferece ao usuário a possibilidade de escolher com qual variedade deseja iniciar a consulta. A qualquer momento, o usuário pode alternar entre diferentes tipos de pimentas no próprio *chat*, no acesso a informações sobre definição, ardência, usos culinários e outras características, a partir do conteúdo estruturado no arquivo *pepper\_info.json*. Esse comportamento da interface pode ser observado na Figura 9 (Seção 3.10), que mostra um exemplo de diálogo com o aplicativo para uma variedade específica de pimenta.

Durante o desenvolvimento, foram realizados testes funcionais internos, conduzidos pela autora, com o objetivo de verificar o fluxo de uso do protótipo em condições controladas: captura ou seleção de imagens, envio ao *endpoint* */predict*, retorno dos resultados em formato JSON e atualização da interface com a variedade identificada. Esses testes permitiram corrigir problemas de *layout* e de tratamento de erros (por exemplo, envio de arquivos em formato incorreto ou ausência de conexão no momento da requisição), bem como verificar a integração entre a *PWA* e o *backend* de inferência.

Em síntese, ao final da etapa de desenvolvimento, o protótipo do *App Pimentas* encontra-se funcional e acessível via navegador, a partir de *URL* pública, com suporte à captura de imagens em dispositivos móveis, execução do modelo *YOLOv8m* em formato *ONNX* no servidor e apresentação dos resultados em interface voltada ao uso prático em ambientes de comercialização de hortigranjeiros, e a *CEAGESP* tem o estudo de caso inicial.

#### 4.4 Tentativa de validação do protótipo com a equipe da CEAGESP

Esta seção descreve a etapa de validação do protótipo *App Pimentas* planejada no Centro de Qualidade Hortigranjeira da CEAGESP e a situação efetivamente observada até o fechamento desta dissertação. Em linha com o procedimento metodológico apresentado na Seção 3.11, previu-se a aplicação de um questionário estruturado a profissionais da CEAGESP, de modo a captar percepções sobre usabilidade, desempenho percebido do modelo e utilidade prática do protótipo no contexto de trabalho.

Entretanto, apesar do envio do convite e da disponibilização do aplicativo em ambiente de produção, não foram registradas respostas ao questionário durante o período de coleta. Assim, não há resultados empíricos suficientes para uma análise quantitativa ou estatística da aceitação do protótipo pela equipe da CEAGESP. Nesta seção, são apresentados o procedimento de coleta adotado e as implicações dessa ausência de retorno para a interpretação dos resultados da pesquisa.

##### 4.4.1 Procedimento de coleta

A validação do protótipo no contexto de uso foi planejada por meio de um questionário eletrônico elaborado na plataforma *Microsoft Forms*, conforme detalhado na Seção 3.11 e apresentado integralmente no Apêndice A. O *link* de acesso ao *App Pimentas*, hospedado na plataforma *Render*, e o *link* do questionário foram encaminhados por e-mail ao chefe da seção de qualidade hortigranjeira da CEAGESP, com solicitação de divulgação aos demais membros da equipe técnica.

O instrumento manteve a estrutura originalmente proposta: uma breve apresentação dos objetivos do estudo, seguida por itens organizados em blocos temáticos (usabilidade, desempenho percebido da inteligência artificial, desempenho operacional e confiança, utilidade percebida e intenção de uso), todos avaliados em escala de concordância de cinco pontos (1 = discordo totalmente a 5 = concordo totalmente), além de um campo aberto opcional para comentários.

O formulário permaneceu disponível para respostas de 19/11/2025 a 02/12/2025, data de fechamento desta dissertação. Nesse intervalo, não foram registradas respostas válidas, o que impossibilitou a consolidação de resultados numéricos ou a análise comparativa entre respondentes.

#### 4.4.2 Retorno obtido e implicações pra análise

Diante da ausência de respostas ao questionário, não foi possível realizar a análise descritiva planejada na Seção 3.11. Não há, portanto, distribuição de frequências por item, médias por bloco temático ou síntese quantitativa das percepções da equipe da CEAGESP sobre o protótipo.

Essa situação restringe a avaliação do *App Pimentas* sob a perspectiva dos usuários finais previstos, de modo que a discussão desta dissertação, no que se refere à etapa de validação, permanece concentrada em dois níveis:

- **testes funcionais internos**, realizados pela autora, que verificaram o fluxo básico de uso (captura ou seleção de imagens, envio ao *endpoint /predict*, retorno do JSON de predição e atualização da interface);
- **análise conceitual** da adequação do protótipo ao contexto de uso, com base na literatura sobre Agricultura 4.0, visão computacional e serviços digitais em agricultura.

Em outras palavras, o estudo conseguiu avançar na validação técnica (desempenho do modelo *YOLOv8m* e funcionamento da arquitetura *PWA/Backend*), mas não obteve evidências empíricas suficientes para uma validação organizacional do protótipo com a equipe da CEAGESP. Essa distinção ajuda a situar o alcance dos resultados: o protótipo mostrou-se funcional e tecnicamente viável nas condições avaliadas, mas sua aceitação e sua utilidade prática no ambiente do entreposto permanecem como questões em aberto para investigações futuras.

#### 4.4.3 Síntese da etapa de validação

Em síntese, a etapa de validação planejada com a equipe do Centro de Qualidade Hortigranjeira da CEAGESP não se concretizou em termos de respostas ao questionário, o que impede a apresentação de resultados estatísticos ou de conclusões baseadas em evidências coletadas em campo.

Ainda assim, o registro do procedimento adotado, o convite formal por e-mail, a disponibilização de link público para o protótipo e uso de questionário estruturado em plataforma digital podem ser úteis por pelo menos três motivos:

1. **Transparência metodológica**: explicita-se que houve uma tentativa de validação com usuários reais, bem como suas limitações práticas, o que evita que se atribua

- ao protótipo um grau de validação que ele ainda não possui;
2. **Delimitação do escopo dos resultados:** reforça-se que, nesta dissertação, o foco recai sobre o desempenho do modelo *YOLOv8m*, a construção do *dataset* e a implementação do protótipo em arquitetura PWA, enquanto a avaliação sistemática por usuários permanece pendente;
  3. **Agenda para pesquisas futuras:** o questionário e o procedimento descritos podem ser reaproveitados em ciclos posteriores de *Design Science Research*, seja com a própria equipe da CEAGESP, seja com outros atores da cadeia de valor das pimentas, o que permite aprofundar a análise sobre usabilidade, utilidade percebida e impacto do protótipo na rotina de trabalho.

Essa leitura é retomada no Capítulo V, em que a ausência de respostas ao questionário é explicitada como limitação do estudo e como oportunidade para pesquisas subsequentes.

## CAPÍTULO V

### 5 CONCLUSÕES

#### 5.1 Considerações gerais sobre a pesquisa

Este trabalho partiu da seguinte questão orientadora: Como aplicar e avaliar a arquitetura YOLOv8 na identificação automática de variedades de *Capsicum spp.* em ambientes comerciais de alta complexidade e integrá-la a um protótipo de aplicativo móvel voltado ao uso prático? A investigação foi estruturada sob a perspectiva da *Design Science Research (DSR)*, com o entendimento do protótipo *dataset*, modelo e aplicação *web*, como artefato por meio do qual se constrói e se organiza conhecimento de projeto (*design knowledge*) sobre o problema estudado.

As etapas de construção do *dataset*, o treinamento do modelo YOLOv8m, a avaliação dos resultados e o desenvolvimento do protótipo *App Pimentas* em arquitetura *PWA* permitiram explorar questões técnicas (desempenho, generalização e impactos do *data augmentation*). Em linha com a perspectiva da *DSR*, aplicada às ciências agrárias, o artefato não é tratado como solução “final”, mas como um arranjo experimental que ajuda a explicitar condições, limitações e possibilidades de uso de visão computacional na cadeia

De forma geral, os resultados indicaram que a combinação entre um modelo YOLOv8m treinado em um *dataset* específico de pimentas e uma arquitetura *web*, baseada em *PWA*, constitui uma alternativa tecnicamente viável para apoiar a identificação automática de variedades de *Capsicum spp.*, desde que se reconheçam os limites de generalização do modelo e o caráter exploratório do protótipo. O trabalho também reforçou que, em tarefas de classificação fina (*fine-grained*), a construção do *dataset*, o desenho das estratégias de aumento de dados e a definição do contexto de uso têm impacto direto nas métricas obtidas e na utilidade prática do sistema.

#### 5.2 Contribuições da pesquisa

##### 5.2.1 Contribuições científicas

No plano científico, a pesquisa se soma aos estudos que exploram o uso de visão computacional na agricultura e na cadeia de alimentos, em especial aqueles que tratam de detecção e classificação de plantas, frutos ou doenças em imagens capturadas em cenário de campo ou de pós-colheita.

A análise dos resultados do modelo contribui para a literatura de classificação de grão fino ao ilustrar, em um caso aplicado, como diferenças de morfologia entre classes favorecem ou dificultam a discriminação por redes profundas. Contribuem ainda ao ilustrar como certas combinações de *data augmentation* podem, em contextos muito sensíveis, não produzir ganhos homogêneos entre as classes. Esses achados dialogam com revisões sobre análise de imagens de alta granularidade, que apontam a importância de considerar cuidadosamente quais transformações são compatíveis com as pistas visuais que distinguem cada grupo de interesse.

Além disso, o estudo operacionaliza, em um domínio concreto, proposições recentes sobre o papel da *DSR* na renovação das ciências agrárias ao articular a construção de um artefato tecnológico com a compreensão do contexto produtivo e institucional em que ele pode ser utilizado. Essa articulação é particularmente relevante em um cenário em que a Agricultura 4.0 e os serviços digitais em agricultura são vistos como promissores, mas ainda convivem com lacunas de infraestrutura, de dados e de desenho de soluções ajustadas às rotinas dos usuários finais.

### **5.2.2 Contribuições metodológicas**

Do ponto de vista metodológico, uma preocupação central deste trabalho foi documentar, de forma transparente e detalhada, o processo de construção do protótipo *App Pimentas*, bem como as condições necessárias para sua reconstrução em outros ambientes técnicos. Essa postura dialoga com a perspectiva da *Design Science Research (DSR)*, que enfatiza a necessidade de tornar explícitos o ciclo de construção e de avaliação do artefato, o contexto de aplicação e os artefatos gerados, como condição para o rigor e para a utilidade científica da pesquisa (HEVNER *et al.*, 2004; HEVNER, 2007).

No caso específico do *App Pimentas*, essa preocupação se materializou na organização de uma cadeia de implementação que explicita os principais elementos do ambiente: (i) um repositório *Git* que contém o código-fonte do *backend* em *FastAPI* e da interface *PWA*; (ii) um arquivo *requirements.txt* com as dependências necessárias ao ambiente *Python*; (iii) arquivos de configuração da interface *web* (*HTML*, *JavaScript*, manifesto e *service worker*); (iv) o uso de variáveis de ambiente para parametrizar o caminho do modelo *best.onnx* e outras configurações sensíveis. A literatura em inteligência artificial e aprendizado de máquina tem apontado, de forma recorrente, que a ausência de informação sobre código, dados, ambiente e

procedimentos de execução é um dos principais obstáculos à reproduzibilidade de estudos empíricos (GUNDERSEN e KJENSMO, 2018; PINEAU *et al.*, 2021; DESAI; ABDELHAMID; PADALKAR, 2025).

Ao explicitar o uso de repositório *Git*, arquivo *requirements.txt*, variáveis de ambiente, *logs* de carregamento do modelo e procedimentos passo a passo para reconstrução do ambiente, o trabalho responde a recomendações da literatura de ciência de dados que vêm destacando a importância do versionamento e da documentação de *pipelines* para a reproduzibilidade de experimentos de aprendizado de máquina (GUNDERSEN e KJENSMO, 2018; PINEAU *et al.*, 2021). Nesse sentido, a metodologia proposta pode servir como referência para outros pesquisadores que desejem implantar protótipos semelhantes, seja em agricultura, seja em outros domínios que demandem classificação de imagens em tempo quase real.

A opção por uma arquitetura *PWA* também tem dimensão metodológica. Ao empregar apenas tecnologias padrão da *web* (*HTML*, *CSS* e *JavaScript*), complementadas por *service worker* e mecanismos de *cache*, o protótipo pôde ser utilizado em diferentes dispositivos sem exigir a instalação de um aplicativo nativo. Sendo assim, o fato reduz barreiras de entrada e se mostra condizente com descrições de *PWAs* como aplicações *web* instaláveis, baseadas em tecnologias padrão, capazes de oferecer uso em múltiplas plataformas e funcionamento em modo *offline* ou com conectividade instável (CHERUKURI, 2024; MUAWWAL, 2024).

Além disso, a metodologia descreve, em passos operacionais, os procedimentos necessários para reconstruir o protótipo em outro ambiente: obtenção do código-fonte (via repositório *Git* ou pacote associado a um *commit* específico), instalação do *Python* e das dependências por meio do *requirements.txt*, configuração da variável de ambiente *MODEL\_URL*, o que aponta para o arquivo *best.onnx*, para a inicialização do servidor *FastAPI* e para o acesso às rotas principais (*/ui*, */predict* e */info*). A descrição desses passos busca atender as recomendações de que estudos em aprendizado de máquina explicitem não apenas os resultados, mas também o *pipeline* técnico que os produz, o que facilita a repetição e a extensão de experimentos por outros grupos de pesquisa (GUNDERSEN e KJENSMO, 2018; PINEAU *et al.*, 2021; DESAI; ABDELHAMID; PADALKAR, 2025).

### 5.2.3 Contribuições práticas

No plano prático, a pesquisa resultou um protótipo funcional que ilustra como

modelos de detecção de objetos podem ser integrados ao cotidiano de um entreposto hortigranjeiro. Embora ainda não se trate de uma solução pronta para adoção em escala, o *App Pimentas* fornece um exemplo concreto de como organizar, em uma mesma interface, a identificação automática de variedades de pimentas e a disponibilização de informações sobre ardência, uso culinário, conservação e curiosidades, em formato de *chat* acessível de dispositivos móveis.

Essa contribuição prática tem potencial para apoiar não apenas a equipe técnica da CEAGESP, mas também estudantes, pesquisadores e outros profissionais interessados em testar, adaptar ou expandir o artefato para outros produtos agrícolas ou outros contextos de comercialização. Ao longo do processo, foram identificados pontos de atenção que podem orientar iniciativas futuras, como a necessidade de calibrar o fluxo de interação às condições reais de uso (tempo disponível, qualidade da rede, familiaridade com tecnologias digitais) e de considerar desde cedo estratégias para manutenção e atualização contínua do modelo à medida que outras variedades de pimentas sejam implementadas.

### 5.3 Limitações e implicações

Como em todo estudo empírico, os resultados apresentados devem ser interpretados à luz de limitações específicas.

A primeira diz respeito ao próprio *dataset*. As imagens utilizadas foram coletadas em condições e contextos limitados, com foco em oito variedades de *Capsicum spp.* comercializadas em um único entreposto. Isso significa que o modelo foi exposto a um conjunto relativamente restrito de variações de iluminação, disposição dos frutos, tipos de embalagem e combinações de produtos. Em cenários de aplicação diferentes, outros mercados, outras regiões, outros arranjos de bancas, é plausível que o desempenho se altere, o que reforça a necessidade de processos contínuos de atualização do *dataset* e de re-treinamento do modelo quando se pretende ampliar o escopo de uso.

Uma segunda limitação está associada às estratégias de *data augmentation*. Embora tenham sido importantes para aumentar a diversidade aparente das imagens de treinamento, os resultados mostraram que, em classes visualmente muito próximas, algumas transformações podem reforçar semelhanças em vez de destacar diferenças e produzir efeitos distintos entre as classes. Isso sugere que abordagens

de aumento de dados em classificação de grão fino talvez exijam configurações mais cuidadosas, possivelmente adaptadas a cada grupo de interesse, em linha com análises recentes sobre efeitos contextuais do *data augmentation* em redes profundas.

Do ponto de vista arquitetural, a opção por realizar a inferência no servidor, e não diretamente no dispositivo, implica dependência de conexão de rede para uso do protótipo. Embora a arquitetura *PWA* permita algum nível de funcionamento em condições de conectividade instável (*cache* de interface, mensagens de erro mais controladas), a identificação de pimentas em si não ocorre *offline*. Isso pode limitar o uso em ambientes com cobertura precária ou em situações em que a rede local do mercado esteja sobrecarregada.

Por fim, a avaliação com usuários não pôde ser efetivamente realizada. Essa ausência de retorno configura uma limitação importante do estudo e indica que a etapa de validação com usuários externos ainda precisa ser conduzida em ciclos futuros de pesquisa, seja com a CEAGESP, seja com outros contextos de comercialização de hortigranjeiros.

Em consequência, o objetivo específico C (“desenvolver e validar um protótipo de aplicativo móvel que integre o modelo *YOLOv8*”), foi atendido de forma parcial: o protótipo foi desenvolvido e colocado em funcionamento em ambiente *web* (*PWA*), com integração ao modelo *YOLOv8m* em formato *ONNX*, mas a validação com usuários finais permaneceu restrita a testes funcionais internos, sem a etapa de avaliação estruturada no contexto da CEAGESP.

Essas limitações não invalidam os achados, mas delimitam o alcance das conclusões. Elas reforçam a importância de tratar o protótipo como ponto de partida para estudos subsequentes e de integrar, em ciclos de DSR, avaliações técnicas e organizacionais mais amplas, como sugerem discussões recentes sobre a adoção de serviços digitais em agricultura.

Uma limitação adicional refere-se ao módulo de *chat* do protótipo. Nesta etapa, o *chat* foi implementado de forma propositalmente simples, a partir de um conjunto fixo de respostas estruturadas em arquivo *JSON* e regras de navegação entre tópicos, sem utilização de modelos de linguagem de grande porte (*large language models*, *LLMs*) nem de técnicas de geração aumentada por recuperação (*retrieval-augmented*

*generation, RAG).*

Modelos de linguagem combinados com mecanismos de recuperação em bases externas têm sido investigados como alternativa promissora para tarefas de pergunta-resposta intensivas em conhecimento. Eles permitem recuperar documentos relevantes de um repositório especializado e utilizá-los como contexto para a geração de respostas mais específicas e fundamentadas (LEWIS *et al.*, 2020).

No contexto deste trabalho, abordagens baseadas em *LLM* e *RAG* poderiam, em estudos futuros, ser exploradas para oferecer explicações mais ricas e atualizadas sobre características agronômicas, nutricionais ou de mercado das pimentas, a partir de um repositório ampliado de fontes. Ao mesmo tempo, a adoção desse tipo de solução exigiria lidar com questões de custo computacional, curadoria e atualização de conteúdo, controle de alucinações e governança das fontes utilizadas, o que justifica a opção inicial por um *chat* determinístico e controlado nesta fase da pesquisa.

#### **5.4 Recomendações para trabalhos futuros**

No eixo de dados e modelagem, uma prioridade é a ampliação do *dataset* para incluir mais variedades de pimentas, diferentes estágios de maturação, múltiplos mercados e condições sazonais diversas. A incorporação de imagens capturadas por diferentes dispositivos e em diferentes horários pode contribuir para tornar o modelo mais *resistente* a variações de iluminação e enquadramento. Estudos posteriores também podem investigar estratégias de *data augmentation* específicas para classificação fina e avaliar, de forma sistemática, quais transformações preservam ou reforçam pistas morfológicas relevantes para a distinção entre classes semelhantes.

Outra possibilidade é a exploração de arquiteturas alternativas ou complementares ao *YOLOv8m*, como versões mais recentes da família *YOLO* ou abordagens multimodais que integrem informações visuais e textuais (por exemplo, rótulos, anotações de caixas, descrição do lote). A comparação entre diferentes modelos poderia ser conduzida com base em métricas padronizadas e em cenários de uso alinhados à literatura de avaliação de detectores em agricultura.

Quanto à interação e à adoção, recomenda-se aprofundar estudos com usuários em diferentes perfis (classificadores, comerciantes, técnicos de qualidade, estudantes), com a combinação de métodos quantitativos (questionários

estruturados, experimentos controlados) e qualitativos (entrevistas, observação em campo) para compreender não apenas se o protótipo é percebido como útil, mas como ele se integra, ou não, às rotinas de trabalho. Esses estudos poderiam explorar, por exemplo, o papel do aplicativo em ações de formação, na comunicação com compradores ou na padronização de terminologias de pimentas.

Por fim, ao considerar o debate sobre ciência aberta e serviços digitais em agricultura, trabalhos futuros podem avançar na disponibilização do *dataset* e do código-fonte em repositórios públicos, com documentação mais detalhada e discussão de modelos de governança para manutenção, atualização e eventual ampliação colaborativa do repositório de imagens e do conjunto de classes.

## 5.5 Considerações finais

Os resultados obtidos ao longo desta pesquisa evidenciam que os objetivos propostos foram alcançados. A utilização da arquitetura YOLOv8 permitiu o desenvolvimento e a avaliação de um modelo de detecção automática para a identificação de diferentes variedades de *Capsicum spp.* em ambientes comerciais de maior complexidade, caracterizados por variações de iluminação, sobreposição de objetos e diversidade visual. As análises realizadas sugerem um desempenho adequado do modelo diante dos desafios observados nesses cenários. Ademais, a integração da solução a um protótipo de aplicativo móvel demonstrou a possibilidade de uso prático da abordagem proposta, apontando para seu potencial de aplicação em contextos reais.

Ao longo deste trabalho, a construção e o teste do protótipo *App Pimentas* permitiram discutir, em um caso concreto, como técnicas recentes de visão computacional podem ser articuladas a necessidades específicas de um contexto agroalimentar, que envolve classificação de produtos e disseminação de informações de forma acessível. Em vez de oferecer uma solução fechada, o estudo buscou organizar um caminho de projeto, desde o planejamento do *dataset* até a implantação em ambiente *web*, que possa ser reutilizado, criticado e estendido por outros pesquisadores e praticantes.

Sob a lente da *Design Science Research*, o artefato desenvolvido funciona como um mediador entre o conhecimento técnico sobre redes neurais convolucionais, modelos YOLO e arquiteturas *web*, e o conhecimento prático de quem lida

diariamente com a comercialização de pimentas e outros hortigranjeiros. Nessa mediação, algumas questões foram respondidas, como a viabilidade de uso de YOLOv8 nesse domínio e a possibilidade de oferecer uma interface *PWA* acessível. Outras foram abertas, sobretudo no que diz respeito ao escalonamento da solução, à ampliação do conjunto de classes e à consolidação de evidências de impacto em processos de trabalho.

Espera-se que as reflexões e os procedimentos aqui sistematizados possam apoiar não apenas futuras pesquisas acadêmicas, mas também iniciativas práticas interessadas em aproximar visão computacional, agricultura e serviços digitais de forma cuidadosa, gradual e sensível ao contexto.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ALOMAR, K.; AYSEL, H. I.; CAI, X. Data augmentation in classification and segmentation: a survey and new strategies. *Journal of Imaging*, Basel, v. 9, n. 2, art. 46, 2023. DOI: 10.3390/jimaging9020046.
- BAI, J. et al. ONNX: Open Neural Network Exchange. 2019. Disponível em: <https://github.com/onnx/onnx>. Acesso em: 29 nov. 2025.
- BARBOSA, M. Conjunto de dados com e sem aumento de dados de pimentas. **figshare**. Databse, 2025. <https://doi.org/10.6084/m9.figshare.30082318.v3>
- BRASIL. INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL. **App Pimentas: pedido de registro de programa de computador**. Processo nº 512025006150-1, petição nº 870250108484, 26 nov. 2025.
- BARBOZA, G. E. et al. Taxonomy of the genus Capsicum: origin, domestication, diversity and applications. *Plants*, v. 11, n. 23, p. 3264, 2022.
- BASITH, S. et al. Harnessing the therapeutic potential of capsaicin and its analogues. *Molecules*, v. 21, n. 15, p. 5179, 2016.
- BEZABH, Yohannes Agegnehu et al. **CPD-CCNN: classification of pepper disease using a concatenation of convolutional neural network models**. *Scientific Reports*, [s. l.], v. 13, n. 1, p. 15581, 20 set. 2023.
- BOCHKOVSKIY, A.; WANG, C. Y.; LIAO, H. Y. M. YOLOv4: Optimal speed and accuracy of object detection. **arXiv preprint arXiv:2004.10934**, 2020. Disponível em: <https://arxiv.org/abs/2004.10934>. Acesso em: 11 ago. 2024.
- CAO, Z.; SUN, S.; BAO, X. A Review of Computer Vision and Deep Learning Applications in Crop Growth Management. *Applied Sciences*, v. 15, n. 15, p. 8438, 2025.
- CHERUKURI, B. R. Progressive Web Apps (PWAs): enhancing user experience through modern web development. *International Journal of Science and Research (IJSR)*, v. 13, n. 10, 2024.
- CRUZ-DOMÍNGUEZ, H. T.; VAN, H. T. et al. A novel method for dried chili pepper classification using artificial intelligence. *Journal of Agriculture and Food Research*, v. 4, p. 100-110, 2021.
- DALAL, M.; MITTAL, P. A systematic review of deep learning-based object detection in agriculture: methods, challenges, and future directions. *Computers, Materials & Continua*, v. 76, n. 3, p. 3517-3539, 2025.
- DESAI, A.; ABDELHAMID, M.; PADALKAR, N. R. What is reproducibility in artificial intelligence and machine learning research? *AI Magazine*, v. 46, n. 2, 2025. DOI: 10.1002/aaai.70004.
- DHANYA, V. G. et al. Deep learning based computer vision approaches for smart

agricultural applications. **Artificial Intelligence in Agriculture**, v. 6, p. 211-229, 2022. DOI: 10.1016/j.aiia.2022.09.007.

EL SAKKA, M. A Review of CNN Applications in Smart Agriculture Using Deep Learning. **Sensors**, 2025.

EMBRAPA. **Agricultura Digital: inovação e tendências**. Brasília: Embrapa, 2020.

FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS (FAO). **Production** (t). FAO Statistics Database, 2024. Disponível em: <https://www.fao.org/faostat/en/#data/QCL>. Acesso em: 1 jul. 2024.

GIL, A. C. **Métodos e técnicas de pesquisa social**. 7. ed. São Paulo: Atlas, 2018.

GOOGLE. **Welcome to Colab**. 2025a. Disponível em: <https://colab.research.google.com/>. Acesso em: 23 nov. 2025.

GOOGLE. **Colaboratory FAQ**. 2025b. Disponível em: <https://research.google.com/colaboratory/faq.html>. Acesso em: 23 nov. 2025.

GOOGLE. **Preços dos serviços pagos do Colab**. 2025c. Disponível em: <https://colab.research.google.com/signup?hl=pt-BR>. Acesso em: 23 nov. 2025.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge: MIT Press, 2016.

GUNDERSEN, O. E.; KJENSMO, S. State of the art: Reproducibility in artificial intelligence. **Proceedings of the AAAI Conference on Artificial Intelligence**, v. 32, n. 1, 2018.

HASSOUN, A. et al. Exploring the role of green and Industry 4.0 technologies in achieving sustainable development goals in food sectors. **Food Research International**, v. 162, p. 112068, 2022.

HEVNER, A. R. A three cycle view of design science research. **Scandinavian Journal of Information Systems**, v. 19, n. 2, p. 87-92, 2007.

HEVNER, A. R. et al. Design science in information systems research. **MIS Quarterly**, v. 28, n. 1, p. 75-105, 2004.

HEVNER, A. R.; VOM BROCKE, J. A proficiency model for design science research education. **Journal of Information Systems Education**, v. 34, n. 3, p. 264-278, 2023.

HOSSEN, M. I. Transfer learning in agriculture: a review. **Artificial Intelligence Review**, 2025.

JAVAID, M. et al. Enhancing smart farming through the applications of Agriculture 4.0 technologies. **International Journal of Intelligent Networks**, v. 3, p. 150-164, 2022.

JOCHER, G.; CHAURASIA, A.; QIU, J. Ultralytics YOLOv8 — model overview and configuration guide. 2023. Disponível em: <https://docs.ultralytics.com/models/yolov8/>. Acesso em: 7 set. 2024.

KAMILARIS, A.; PRENAFETA-BOLDÚ, F. X. Deep learning in agriculture: A survey. **Computers and Electronics in Agriculture**, v. 147, p. 70-90, 2018.

KEDRON, P.; FRAZIER, A. E. How to improve the reproducibility, replicability, and extensibility of remote sensing research. **Remote Sensing**, v. 14, n. 21, art. 5471, 2022. DOI: 10.3390/rs14215471.

KESKES, M. I. Review of The Current State of Deep Learning Applications in Agriculture. **Preprints**, 2025.

KHAN, Z.; SHEN, Y.; LIU, H. Object Detection in Agriculture: A Comprehensive Review of Methods, Applications, Challenges, and Future Directions. **Agriculture**, v. 15, n. 13, p. 1351, 2025.

KURTULMUŞ, F.; ALIBAŞ, L.; KAVDIR, I. Classification of pepper seeds using machine vision based on neural network. **International Journal of Agricultural and Biological Engineering**, v. 9, p. 51-62, 2016.

LEITE, D. Advancements and outlooks in utilizing Convolutional Neural Networks for plant disease detection. **Theoretical and Natural Science**, 2024.

LEWIS, P. et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. 2020. Disponível em: <https://arxiv.org/abs/2005.11401>. DOI: 10.48550/arXiv.2005.11401.

LEZOCHE, M. et al. Agri-Food 4.0: A survey of the supply chains and technologies for the future agriculture. **Computers in Industry**, v. 117, p. 103187, 2020.

LI, C. et al. YOLOv6: A single-stage object detection framework for industrial applications. **arXiv preprint arXiv:2209.02976**, 2022. Disponível em: <https://arxiv.org/abs/2209.02976>.

LI, M. et al. High-performance plant pest and disease detection based on model ensemble with inception module and cluster algorithm. **Plants**, Basel, v. 12, n. 1, p. 200, 2023.

LI, Xu et al. Fast and accurate green pepper detection in complex backgrounds via an improved Yolov4-tiny model. **Computers and Electronics in Agriculture**, Amsterdam, v. 191, p. 106503, 22 Oct. 2021.

LÓPEZ-BARRIOS, J. D. et al. Green sweet pepper fruit and peduncle detection using Mask R-CNN in greenhouses. **Applied Sciences**, v. 13, p. 6296, 2023.

MIN, X. Computer Vision Meets Generative Models in Agriculture. **Applied Sciences**, 2025.

MAXWELL, A. E.; BESTER, M. S.; RAMEZAN, C. A. Enhancing reproducibility and replicability in remote sensing deep learning research and practice. **Remote**

**Sensing**, v. 14, n. 22, art. 5760, 2022. DOI: 10.3390/rs14225760.

MOHI-ALDEN, K.; OMID, M.; SOLTANI FIROUZ, M.; NASIRI, A. Design and evaluation of an intelligent sorting system for bell pepper using deep convolutional neural networks. **Journal of Food Science**, v. 87, n. 1, p. 289-301, 2022.

MUAWWAL, A. The implementation of PWA (Progressive Web App) technology in enhancing website performance & mobile accessibility. **Buletin Pos dan Telekomunikasi**, v. 22, n. 1, p. 25-36, 2024. DOI: 10.17933/bpostel.v22i1.395.

NVIDIA. **NVIDIA T4 Tensor Core GPU — Datasheet**. 2019. Disponível em: <https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/tesla-t4/t4-tensor-core-datasheet-951643.pdf>. Acesso em: 23 nov. 2025.

NVIDIA. **NVIDIA T4 Tensor Core GPU**. 2024. Disponível em: <https://www.nvidia.com/en-us/data-center/tesla-t4/>. Acesso em: 23 nov. 2025

OLIVEIRA, R. C. d.; SILVA, R. D. d. S. e. Artificial Intelligence in Agriculture: Benefits, Challenges, and Trends. **Applied Sciences**, v. 13, n. 13, p. 7405, 2023.

OMONIYI, T. M. et al. The effect of data augmentation on performance of custom and pre-trained CNN models for crack detection. **Applied Sciences**, Basel, v. 15, n. 22, 12321, 2025. DOI: 10.3390/app152212321.

ORGANIZAÇÃO DAS NAÇÕES UNIDAS (ONU). **Transformando nosso mundo: a Agenda 2030 para o Desenvolvimento Sustentável**. Nova Iorque: ONU, 2015.

PADILLA, R. et al. A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit. **Electronics**, v. 10, n. 3, p. 279, 2021.

PINEAU, J. et al. Improving reproducibility in machine learning research (a report from the NeurIPS 2019 reproducibility program). **Journal of Machine Learning Research**, v. 22, p. 1-20, 2021.

PORCIELLO, Jaron; COGGINS, Sam; MABAYA, Edward; OTUNBA-PAYNE, Gabriella. Digital agriculture services in low-and middle-income countries: a systematic scoping review. **Global Food Security**, v. 34, p. 100640, 2022. DOI: 10.1016/j.gfs.2022.100640.

PROST, L. Revitalizing agricultural sciences with design sciences. **Agricultural Systems**, v. 193, p. 103225, 2021. DOI: <https://doi.org/10.1016/j.agsy.2021.103225>.

PYTORCH. **Get Started Locally**. 2025. Disponível em: <https://pytorch.org/get-started/locally/>. Acesso em: 23 nov. 2025

RANI, N. S. et al. CISCS: Classification of inter-class similarity based medicinal plant species groups with machine learning. **MethodsX**, v. 15, 103652, 2025. DOI: 10.1016/j.mex.2025.103652.

REDMON, J. et al. You Only Look Once: Unified, real-time object detection. **arXiv preprint arXiv:1506.02640**, 2016. Disponível em: <https://arxiv.org/abs/1506.02640>.

REDMON, J.; FARHADI, A. YOLO9000: Better, faster, stronger. **arXiv preprint** arXiv:1612.08242, 2017. Disponível em: <https://arxiv.org/abs/1612.08242>.

REDMON, J.; FARHADI, A. YOLOv3: An incremental improvement. **arXiv preprint** arXiv:1804.02767, 2018. Disponível em: <https://arxiv.org/abs/1804.02767>.

REN, S. et al. Faster R-CNN: Towards real-time object detection with region proposal networks. *In: ADVANCES IN NEURAL INFORMATION PROCESSING*

ROBOFLOW. Preprocess images. 2024. Disponível em: <https://docs.roboflow.com/datasets/image-preprocessing>. Acesso em: 29 nov. 2025.

ROBOFLOW. Image augmentation. 2024. Disponível em: <https://docs.roboflow.com/datasets/image-augmentation>. Acesso em: 29 nov. 2025.

ROBOFLOW. Roboflow: computer vision tools for developers and enterprises. 2024. Disponível em: <https://roboflow.com/>. Acesso em: 29 nov. 2025.

ROUMELIOTIS, K. I.; TSELIKAS, N. D. Evaluating progressive web app accessibility for people with disabilities. **Network**, v. 2, n. 2, p. 350-369, 2022. DOI: 10.3390/network2020022.

SYSTEMS (NeurIPS), 28., 2015, Montreal. **Anais...** Montreal, 2015. p. 91-99.

SAYEED, A. et al. BengaliNet: a low-cost novel convolutional neural network for Bengali handwritten characters recognition. *Applied Sciences*, Basel, v. 11, n. 15, 6845, 2021. DOI: 10.3390/app11156845.

SCOVILLE, W. L. Note on capsicums. **Journal of the American Pharmaceutical Association**, v. 1, n. 5, p. 453-454, 1912.

SORNAPUDI, S.; SINGH, R. Self-supervised backbone framework for diverse agricultural vision tasks. **arXiv preprint** arXiv:2403.15248, 2024. Disponível em: <https://arxiv.org/abs/2403.15248>.

SUBEESH, A.; MEHTA, C. R. Automation and digitization of agriculture using artificial intelligence and internet of things. **Artificial Intelligence in Agriculture**, v. 5, p. 278-291, 2021.

TAN, M.; LE, Q. EfficientDet: Scalable and efficient object detection. *In: IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR)*, 2020, Seattle. **Anais...** Piscataway: IEEE, 2020. p. 10781-10790.

TANEJA, A. et al. Artificial Intelligence: Implications for the Agri-Food Sector. **Agronomy**, v. 13, n. 5, p. 1397, 2023. DOI: 10.3390/agronomy13051397.

TERVEN, J.; CÓRDOVA-ESPARZA, D. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. **arXiv preprint** arXiv:2304.00501, 2023. Disponível em: <https://arxiv.org/abs/2304.00501>.

TIAN, H. et al. Computer vision technology in agricultural automation: a review.

**Information Processing in Agriculture**, v. 7, n. 1, p. 1-19, 2020.

TIANGOLO. *FastAPI*. 2018. Disponível em: <https://fastapi.tiangolo.com/>. Acesso em: 29 nov. 2025.

TUUNANEN, T. et al. Dealing with complexity in design science research: a methodology using design echelons. *Information Systems Journal*, Early View, 2025. DOI: <https://doi.org/10.1111/isj.12510>.

ULTRALYTICS. **Train mode (YOLO)**. 2023a. Disponível em: <https://docs.ultralytics.com/modes/train/>. Acesso em: 23 nov. 2025.

ULTRALYTICS. **FAQ — System Requirements**. 2023b. Disponível em: <https://docs.ultralytics.com/help/FAQ/>. Acesso em: 23 nov. 2025

ULTRALYTICS. **Object Detection Datasets Overview**. 2023c. Disponível em: <https://docs.ultralytics.com/datasets/detect/>. Acesso em: 23 nov. 2025.

ULTRALYTICS. **Datasets Overview (data.yaml)**. 2023d. Disponível em: <https://docs.ultralytics.com/datasets/>. Acesso em: 23 nov. 2025.

ULTRALYTICS. **Explore Ultralytics YOLOv8**. 2023e. Disponível em: <https://docs.ultralytics.com/models/yolov8/>. Acesso em: 23 nov. 2025.

ULTRALYTICS. **Ultralytics YOLOv8 Documentation: Datasets and Labels; Configuration (imgsz) and Training**. Disponível em: <https://docs.ultralytics.com>.

ULTRALYTICS. **YOLOv5**. 2020. Disponível em: <https://github.com/ultralytics/yolov5>.

ULTRALYTICS. **YOLOv8**. 2023. Disponível em: <https://github.com/ultralytics/ultralytics>.

ULTRALYTICS. **YOLO11: Modelos e documentação oficial**. 2024. Disponível em: <https://docs.ultralytics.com/pt/models/yolo11/>.

UPADHYAY, A. Deep Learning and Computer Vision in Plant Disease Detection: a comprehensive review of techniques, models, and trends in precision agriculture. **Artificial Intelligence Review**, 2025.

VENKATESWARA, S. M.; PADMANABHAN, J. Deep learning based agricultural pest monitoring and classification. **Scientific Reports**, v. 15, art. 8684, 2025.

VOM BROCKE, J.; HEVNER, A. R.; MAEDCHE, A. Introduction to design science research. In: VOM BROCKE, J.; HEVNER, A. R.; MAEDCHE, A. (orgs.). **Design Science Research. Cases**. Cham: Springer, 2020. p. 1–13. DOI: [https://doi.org/10.1007/978-3-030-46781-4\\_1](https://doi.org/10.1007/978-3-030-46781-4_1).

WANG, A. et al. YOLOv10: Real-time end-to-end object detection. **arXiv preprint arXiv:2405.14458**, 2024. Disponível em: <https://arxiv.org/abs/2405.14458>.

WANG, C. Y. et al. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. **arXiv preprint arXiv:2207.02696**, 2022. Disponível em: <https://arxiv.org/abs/2207.02696>.

WANG, C. Y.; YEH, I. H.; LIAO, H. Y. M. YOLOv9: Learning what you want to learn using programmable gradient information. **arXiv preprint arXiv:2402.13616**, 2024. Disponível em: <https://arxiv.org/abs/2402.13616>.

WANG, W.; KANG, Y. A Review of Computer Vision Technologies in Precision Agriculture. **Theoretical and Natural Science**, v. 101, p. 35-40, 2025.

WEI, X.-S. et al. Fine-Grained Image Analysis with Deep Learning: A Survey. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 44, n. 12, p. 8927-8948, 2022.

YILMAZ, B.; KUTBAY, U. YOLOv8 Based Drone Detection: Performance Analysis and Optimization. **Computers**, v. 13, n. 9, p. 234, 2024.

YIN, R. K. **Estudo de caso: planejamento e métodos**. 5. ed. Porto Alegre: Bookman, 2015.

ZENG, Y. et al. Pepper disease detection model based on convolutional neural network and transfer learning. *In: IOP CONFERENCE SERIES: EARTH AND ENVIRONMENTAL SCIENCE*, 2021. **Anais...** IOP Publishing, 2021. v. 792, n. 1, p. 012001.

ZHANG, Z. et al. Comparative study of object detection algorithms in agriculture. **Computers and Electronics in Agriculture**, v. 182, p. 106055, 2021.

## APÊNDICE A – Questionário de validação do protótipo do aplicativo *App Pimentas*

### Questionário de Validação do *App de Identificação de Pimentas*

Este questionário faz parte de um estudo científico para validar o aplicativo móvel de identificação de pimentas desenvolvido para auxiliar no seu trabalho. Sua opinião é fundamental para avaliarmos a usabilidade, o desempenho do modelo de Inteligência Artificial e a aplicabilidade da ferramenta em sua rotina.

**Instruções:** Responda a cada afirmação utilizando a seguinte escala de concordância, baseada na sua experiência durante o teste do aplicativo:

1. Discordo Totalmente	2. Discordo Parcialmente	3. Neutro	4. Concordo Parcialmente	5. Concordo Totalmente
------------------------	--------------------------	-----------	--------------------------	------------------------

Quando você enviar este formulário, ele não coletará automaticamente seus detalhes, como nome e endereço de email, a menos que você mesmo o forneça.

\* Obrigatória

**Bloco 1: Usabilidade e Experiência do Usuário (UX) \***

	1	2	3	4	5
Eu aprendi a usar o aplicativo rapidamente (menos de 5 minutos).	<input type="radio"/>				
A navegação entre as telas e a função de tirar fotos são intuitivas.	<input type="radio"/>				
Foi fácil manusear o celular e o aplicativo nas condições de iluminação do local (galpão/box).	<input type="radio"/>				

**Bloco 2: Desempenho e Precisão da Inteligência Artificial \***

	1	2	3	4	5
O aplicativo identificou corretamente o tipo de pimenta (ex: Chilli, Fidalga, Biquinho).	<input type="radio"/>				
O aplicativo conseguiu classificar as pimentas mesmo quando estavam misturadas ou parcialmente escondidas/cobertas.	<input type="radio"/>				
A identificação fornecida pelo aplicativo está de acordo com a classificação padrão utilizada no CEAGESP.	<input type="radio"/>				

**Bloco 3: Desempenho Operacional e Confiança \***

	1	2	3	4	5
O tempo entre tirar a foto e receber o resultado (latência) foi rápido o suficiente.	<input type="radio"/>				
Eu confio no resultado da classificação deste aplicativo para tomar decisões sobre a pimenta (ex: destinação ou preço).	<input type="radio"/>				
O aplicativo funcionou de forma estável, sem travar ou perder a conexão durante o uso.	<input type="radio"/>				

**Bloco 4: Utilidade Percebida e Intenção de Uso \***

	1	2	3	4	5
A informação de classificação fornecida pelo aplicativo é útil para a minha atividade (seja compra, venda ou conferência de produto).	<input type="radio"/>				
A identificação precisa agregada pelo aplicativo agrupa valor à conferência ou negociação da pimenta.	<input type="radio"/>				
Eu utilizaria este aplicativo sempre que precisasse confirmar o tipo ou a qualidade de uma pimenta.	<input type="radio"/>				

**Enviar**

Nunca forneça sua senha. [Relatar abuso](#)



Este conteúdo foi criado pelo proprietário do formulário. Os dados que você enviar serão enviados ao proprietário do formulário. A Microsoft não é responsável pela privacidade ou práticas de segurança de seus clientes, incluindo aqueles do proprietário deste formulário. Nunca forneça sua senha.

[Microsoft Forms](#) | Pesquisas, questionários e votações com tecnologia de IA. [Crie meu próprio formulário](#)

O proprietário deste formulário não forneceu uma política de privacidade sobre como usaria seus dados de resposta. Não forneça informações pessoais ou confidenciais. | [Condições de uso](#)

## APÊNDICE B – Acesso ao repositório *GitHub* do *App Pimentas*

Este apêndice apresenta as informações necessárias para acesso ao repositório de código-fonte utilizado no desenvolvimento do protótipo do aplicativo App Pimentas.

- Plataforma: GitHub
- Endereço do repositório: <https://github.com/divinomadalena8-crypto/pimentas-api>
- Branch de referência: main
- Commit utilizado na dissertação:  
203a1cfdc499f1ac1124c9de7e906881aee98743
- Data de acesso/validação do repositório: 11 nov. 2025

O repositório contém:

- código-fonte do backend em FastAPI;
- arquivos da interface web (HTML, CSS, JavaScript, manifesto e service worker);
- arquivo requirements.txt com as dependências necessárias;
- arquivos de configuração e documentação mínima para execução local.

## ANEXOS A



Article

# YOLOv8m for Automated Pepper Variety Identification: Improving Accuracy with Data Augmentation

Madalena de Oliveira Barbosa, Fernanda Pereira Leite Aguiar , Suely dos Santos Sousa, Luana dos Santos Cordeiro, Irenilza de Alencar Nääs \* and Marcelo Tsugui Okano

Graduate Program in Production Engineering, Universidade Paulista, Rua Dr. Bacelar 1212, São Paulo 04026-002, SP, Brazil; madalena.divino@etec.sp.gov.br (M.d.O.B.); fernanda.p.aguiar@gmail.com (F.P.L.A.); suely.sousa2@etec.sp.gov.br (S.d.S.S.); profluana.loureiro@gmail.com (L.d.S.C.); marcelo.okano1@docente.unip.br (M.T.O.)  
\* Correspondence: irenilza.naas@docente.unip.br

### Featured Application

The study indicates the practical application of YOLOv8m, a deep learning architecture, for real-time identification and classification of pepper varieties, providing a practical solution for automating quality control and post-harvest sorting processes in food supply chains.

### Abstract

This research addresses the critical need for an efficient and precise identification of *Capsicum* spp. fruit varieties within the post-harvest contexts to enhance quality control and ensure consumer satisfaction. Employing the YOLOv8m convolutional neural network, the study identified eight distinct pepper varieties: Pimento, Bode, Cambuci, Chilli, Fidalga, Habanero, Jalapeno, and Scotch Bonnet. A dataset comprising 1476 annotated images was utilized and significantly expanded through data augmentation techniques, including rotation, flipping, and contrast adjustments. Comparative analysis reveals that training with the augmented dataset yielded significant improvements across key performance indicators, particularly in box precision, recall, and mean average precision (mAP<sub>50</sub> and mAP<sub>95</sub>), underscoring the effectiveness of data augmentation. These findings underscore the considerable potential of CNNs to advance the AgriFood sector through increased automation and efficiency. While acknowledging the constraints of a controlled image dataset, subsequent research should prioritize expanding the dataset and conducting real-world testing to confirm the model's robustness across various environmental factors. This study contributes to the field by illustrating the application of deep learning methodologies to enhance agricultural productivity and inform decision-making.

**Keywords:** pepper classification; YOLOv8m; computer vision in food supply; deep learning; object detection



Academic Editors: Claudio Perone and Roberto Romanillo

Received: 16 May 2025

Revised: 18 June 2025

Accepted: 20 June 2025

Published: 22 June 2025

**Citation:** Barbosa, M.d.O.; Aguiar, F.P.L.; Sousa, S.d.S.; Cordeiro, L.d.S.; Nääs, I.d.A.; Okano, M.T. YOLOv8m for Automated Pepper Variety Identification: Improving Accuracy with Data Augmentation. *Appl. Sci.* **2025**, *15*, 7024. <https://doi.org/10.3390/app15137024>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

### 1. Introduction

The international trade of spicy peppers (*Capsicum* species) constitutes a vibrant and economically significant sector within global agriculture and commerce [1]. Worldwide, the demand for spice peppers continues to rise, fueled by their role in ethnic cuisines, food preservation, and the growing recognition of their health-promoting properties, such as capsaicin content [1–3]. Major producing countries, including Brazil, India, China, and Mexico, utilize pepper cultivation as a vital agricultural activity that supports rural

livelihoods, exports, and local economies [1,4]. The stability of the pepper production chain ensures a constant market supply. The Food and Agriculture Organization (FAO) reported that global pepper production in 2022 reached 812,673 tons, with Brazil producing approximately 128,331 tons [4]. Peppers are essential spices and widely used horticultural crops with global cultivation origins in South America [1]. Peppers, commonly used in the culinary, pharmaceutical, and cosmetic industries, vary in color, heat, size, and shape. In Brazil, varieties such as Malagueta (*Capsicum frutescens*), Dedo-de-Moça (*Capsicum baccatum*), and Bode (*Capsicum chinense*) exhibit distinct traits, with seed variation influencing their characteristics [5].

Consumer satisfaction with spicy peppers is a multifaceted concept that significantly influences market demand and product differentiation. This satisfaction is not solely derived from heat level but also encompasses flavor profiles, aroma, freshness, and perceived health benefits [6,7]. Accurate identification of spicy varieties is crucial for optimizing production, enhancing quality, and minimizing errors in post-harvest management and market. Traditional post-harvest and market management screening methods are manual, time-consuming, and prone to errors in peppers (*Capsicum* spp.), which are widely cultivated and valued for their diverse flavors, shapes, and heat levels. These limitations underscore the need to adopt advanced technologies, such as computer vision, to streamline post-harvest practices [8,9].

The accurate identification of pepper (*Capsicum* spp.) varieties is a critical concern in agricultural production, as misclassification can result in substantial economic losses for producers, diminished product quality, and compromised supply chain efficiency. Given the increasing demand for specific pepper varieties in both domestic and international markets, failure to correctly identify and classify peppers can result in reduced market value, increased post-harvest losses, and a loss of consumer trust [1,4,5]. The complexity of distinguishing between morphologically similar varieties, particularly in large-scale operations, further worsens the inherent risks of errors associated with manual sorting and screening processes. Consequently, there is a need for advanced, automated identification methods to improve both the accuracy and efficiency of varietal discrimination in pepper production systems [6]. Recent developments in image processing have sparked significant interest in agricultural research [7]. The development and implementation of robust computer vision-based detection systems hold considerable potential for transforming post-harvest handling, quality control, and overall agricultural productivity by reducing human error, lowering operational costs, and ensuring consistent product standards [10,11].

Adopting computer vision methods provides greater accuracy and speed in the identification process. Recent advances in object detection methods, particularly those based on deep learning, have shown remarkable progress in agriculture, with several models producing positive results in identification tasks [12]. These technologies automate processes, reduce errors, enhance agricultural operations, and facilitate research advancements, as well as the integration of monitoring systems, contributing to more precise and sustainable agriculture. With the advancement of AI, deep learning has shown promise in addressing complex computer vision tasks in agriculture, with several deep learning algorithms, such as convolutional neural networks (CNNs), being successfully applied to recognition activities [13].

Integrating computer vision and artificial intelligence (AI), particularly with deep learning techniques, has revolutionized agricultural practices. Object detection models, such as convolutional neural networks (CNNs), have revealed substantial progress in automating tasks previously performed manually, enhancing speed and accuracy [12]. The YOLO (you only look once) model family stands out due to its real-time detection capabilities. However, image segmentation accuracy remains a critical factor for computer

vision applications, as it directly impacts the performance of identification models. Recent studies emphasize the importance of striking a balance between computational efficiency and segmentation fidelity, thereby ensuring robust results without compromising system performance [14].

Current research indicates that YOLO-based models are highly effective for detecting and classifying Capsicum (pepper) fruits, including tasks such as growth stage classification, segmentation, and real-time identification [10,15]. While YOLO and data augmentation are well established for general *Capsicum* detection and classification tasks [15,16], their application to efficiently and accurately identify pepper varieties remains underexplored.

The YOLOv8m (medium) model offers a favorable trade-off between detection accuracy and computational efficiency. It outperforms the smaller variants (nano and small) in precision while requiring significantly fewer resources than the larger models (large and extra-large), as previously verified in the current literature [16]. This outcome makes YOLOv8m a robust and efficient choice for deployment on hardware-constrained devices. The authors emphasize that the YOLO architecture was explicitly designed to be scalable, allowing researchers and practitioners to adjust model complexity according to the specific requirements of both the application and the available hardware. Moreover, YOLO-based models have been increasingly applied in the agriculture sector for tasks such as crop and pest detection, contributing to advancements in precision agriculture and process automation.

To address these challenges, the present study develops an automated method for identifying pepper varieties post-harvest using the YOLOv8m architecture and data augmentation. The main contributions include the construction of an annotated image dataset, the application of advanced deep-learning techniques, and a comprehensive evaluation of detection performance. The paper is structured as follows: the subsequent sections present the theoretical background, describe the materials and methods, report the results, and discuss the findings and implications for automated quality control in the post-harvest scenario.

## 2. Related Work

Computer vision is a branch of artificial intelligence that enables machines to interpret and analyze images to make decisions. This process utilizes cameras and computers to identify, track, and measure targets, followed by image processing [17]. This technology has increased automation and efficiency in various agricultural fields. Critical applications include crop growth monitoring, disease control, product identification and classification, automated harvesting, quality testing, and modern farm management automation. Unmanned aerial vehicles (UAVs) equipped with computer vision technology are used for aerial surveillance of agricultural lands, providing valuable data for decision-making. Automating identification processes through computer vision offers significant benefits, including increased efficiency and productivity, enhanced accuracy and consistency, early problem detection, cost reduction, and continuous monitoring [17].

Deep learning (DL), a subfield of machine learning (ML) and artificial intelligence (AI), is considered a core technology of the Fourth Industrial Revolution, also known as Industry 4.0 [18,19]. The word "deep" refers to the multiple layers through which data is processed to build a hierarchical data-driven model. DL models can automatically extract features from data, enabling them to handle complex tasks accurately. These models are widely applied in healthcare, cybersecurity, business intelligence, and especially agricultural visual recognition tasks.

According to Li et al. [20], convolutional neural networks (CNNs) represent a fundamental class of deep learning models, particularly significant in the field of computer vision.

The architecture of CNNs is designed to process grid-structured data, such as images. This process is achieved through convolutional layers, which apply convolution operations to extract local features from images, followed by pooling layers that reduce the dimensionality of the features while retaining essential representations. These networks have shown remarkable capabilities in learning complex hierarchical representations directly from data, making them highly effective in pattern recognition tasks such as image classification, object detection, and semantic segmentation. The capacity of CNNs to capture detailed and abstract visual patterns makes them ideal for identifying various visual characteristics of agricultural products, such as shape, color, and texture, which are critical for distinguishing between different pepper varieties [21].

Recent studies show that deep learning methods substantially enhance the detection and classification of agricultural diseases and renewable energy sites. Ioannou and Myronidis [22] used convolutional neural networks (CNNs) with high-resolution satellite imagery to automatically and accurately detect photovoltaic farms, offering a scalable tool for energy monitoring. In agriculture, Arnob et al. [23] found that ResNet50 outperformed other deep-learning models in classifying cauliflower diseases, achieving an accuracy of 90.85%. Roy and Kukreja [24] applied vision transformers (ViTs) for rice leaf disease detection and severity estimation, surpassing some limitations of traditional CNNs and achieving strong performance in both classification and severity assessment. Together, these findings highlight the value of advanced AI models in both environmental monitoring and precision agriculture.

Hussain [25] explains that the YOLO (you only look once) model represents an innovative approach to real-time object detection, using a single convolutional neural network that simultaneously predicts bounding boxes and class probabilities from an image divided into a grid. This architecture enables the model to process the entire image simultaneously, resulting in high frame-per-second rates suitable for critical applications such as surveillance, autonomous driving, and industrial automation. In post-harvest agricultural contexts, YOLO's efficiency is emphasized by its ability to perform accurate real-time detections, supporting real-time monitoring and identification essential for dynamic agricultural environments, such as fields or conveyor systems sorting produce [26].

The YOLO architecture has evolved substantially since its inception, with each version introducing significant enhancements to improve object detection accuracy, speed, and computational efficiency. YOLOv1 marked the initial step toward real-time detection using a grid-based approach to predict bounding boxes. Building on this, YOLOv2 incorporated anchor boxes, significantly improving localization accuracy. YOLOv3 further refined the model by adopting the Darknet-53 architecture and integrating feature pyramid networks (FPNs), enabling better generalization across object scales. In YOLOv4, additional architectural innovations were introduced, including the Bag of Specials (BoS), which improved accuracy without sacrificing speed. A significant shift occurred with YOLOv5, implemented in PyTorch, where optimizations in training processes and the integration of Cross Stage Partial Network (CSPNet) and Path Aggregation Network (PANet) enhanced performance and modularity. YOLOv6 introduced reparameterization techniques and novel backbone designs for greater efficiency. Continuing the trend of architectural improvements, YOLOv7 employed Efficient Layer Aggregation Network (ELAN) blocks to improve feature learning. With YOLOv8m, the focus turned to computational optimization and more refined training strategies, making it suitable for resource-constrained environments. YOLOv9 introduced programmed gradient information (PGI) to further enhance detection precision and efficiency. Most recently, YOLOv10 eliminated the traditional non-maximum suppression (NMS) step, streamlining the detection of multiple overlapping objects and enhancing throughput for dense scenes. This progression reflects a continuous drive to balance speed,

accuracy, and model simplicity, particularly in real-time agricultural applications such as pepper variety identification [11].

These developments position YOLO as a leading architecture for real-time visual classification systems in the agricultural sector. In parallel, other deep learning approaches, such as convolutional neural networks (CNNs) combined with transfer learning, have also indicated strong practical potential in the agricultural sector. Previous studies have shown their effectiveness, particularly in the pepper production chain. Ren et al. [27] developed a model to classify bell peppers (*Capsicum annuum*) based on external quality, utilizing pre-trained neural networks and image capture tools. Their approach achieved high accuracy (98.14%) and low loss (0.0669), showing excellent performance and generalization without requiring manual feature extraction.

Kurtulmuş et al. [28] investigated the classification of pepper seeds using computer vision and neural networks. Their study involved extracting color, shape, and texture features from the seeds and using a multilayer perceptron (MLP) for classification, achieving an accuracy of 84.94%. This advancement underscores the feasibility of applying CNNs at various stages of the pepper production chain, from seed to final product. Furthermore, a deep learning model utilizing convolutional neural networks (CNNs) for disease detection in pepper leaves was developed based on a dataset of 2478 images.

Previous research [29] has emphasized the use of transfer learning to enhance diagnostic accuracy under cultivation conditions. Their results presented an accuracy of 99.55% in distinguishing between healthy and diseased leaves, highlighting the model's practical applicability for early disease detection in agriculture.

Presenting additional application lines, Mohi-Alden et al. [30] developed an intelligent classification system for bell peppers using deep convolutional neural networks (DCNN). By employing a modified ResNet50 architecture, the system achieved an accuracy of 96.89% in classifying bell peppers based on maturity stage and size. The system was integrated into a computer vision-based sorting machine, showing significant potential to improve classification processes in the food industry, with an overall sorting rate of approximately 3000 samples per hour.

Research on applying CNNs and transfer learning to classify and detect quality, types, and diseases in bell peppers reveals significant potential to transform food quality assessment practices. Previous studies have proven that integrating these technologies can enhance the accuracy, reduce costs, and increase the efficiency of agricultural processes [31].

### 3. Materials and Methods

#### 3.1. Data Recording

Figure 1 presents a schematic overview of the five-step methodology employed for automated pepper variety identification using the YOLOv8m model. Step 1 involved image collection, where photographic images of eight *Capsicum* fruit varieties were acquired to establish the initial dataset. Step 2 involved image annotation, where the collected images were standardized to a  $640 \times 640$  pixel resolution and annotated with bounding boxes to identify *Capsicum* fruit instances using the Roboflow platform. In Step 3, we performed the initial model training using the YOLOv8m model. Step 4 involved data augmentation, where we enhanced the model's generalization and expanded the dataset to include larger images through data augmentation techniques. In Step 5, the YOLOv8m model underwent final training using the full and augmented dataset.

The first step involved training the YOLOv8m model using a dataset comprising 1476 images representing eight distinct pepper varieties. Initially, the dataset consisted of approximately 300 images, collected exclusively through in-person visits to the Company of Warehouses and General Stores of São Paulo (CEAGESP) (a major public enterprise

in Brazil and one of the largest wholesale food centers in the world and the largest in Latin America). Located in São Paulo, Brazil, CEAGESP plays a crucial role in the supply chain of horticultural products, fish, flowers, and other food items. The main unit in São Paulo is the largest such center in South America and handles a significant volume of food products annually, where photographs of the eight pepper types were captured under real market conditions using various devices, including the Samsung Galaxy Tab A8 tablet, which features an 8 MP rear camera; the Samsung Nikon Coolpix L120, a compact camera with a 14.1 MP CCD sensor and a  $21 \times$  optical zoom lens; and the iPhone 11 smartphone, which has a dual 12 MP camera and 4 K video recording capability. These devices ensured high-quality, versatile image capture in diverse market settings. However, as the research progressed, it became evident that expanding the number of pepper varieties was essential to enhance the model's generalization capacity and practical applicability [32].



Figure 1. Steps adopted in the current research to identify *Capsicum* fruits.

Another critical factor was the need to incorporate greater product variability. Observations at CEAGESP indicated that the peppers available for commercial sale exhibited high-quality standards, resulting in limited variation in size, coloration, and overall product condition. This high level of visual uniformity could potentially restrict the model's ability to accurately identify peppers in more diverse real-market conditions, where variations in texture, ripeness, lighting, and shape are more common. An alternative approach was adopted to expand the dataset to address this limitation: sourcing images from online platforms, including open-access databases such as Google Images under the Creative Commons license. Image selection was based on quality criteria, prioritizing high resolution ( $\geq 300$  dpi) and a diversity of angles to ensure a dataset that is more representative of the commercial reality of peppers across different contexts.

In the second step, the images were normalized to 640 px  $\times$  640 px and annotated with bounding box annotations using the Roboflow platform [33]. As Matuck et al. [34] described, Roboflow is a computer vision tool that streamlines the training and deployment of real-time object detection models.

The platform supports image importation, labeling, data preprocessing, model training, and deployment, making the development of computer vision models more accessible and efficient. Through this platform, the images were annotated, and distinct classes were created for each pepper variety: Biquinho Pepper, Bode Pepper, Cambuci Pepper, Chilli Pepper, Fidalga Pepper, Habanero Pepper, Jalapeno Pepper, and Scotch Bonnet Pepper.

Table 1 presents a photographic reference and concise morphological description for the eight pepper cultivars contained in the study (Biquinho, Bode, Cambuci, Chilli, Fidalga, Habanero, Jalapeno, and Scotch Bonnet). Each cultivar is provided with its common name, scientific designation, and key visual traits (shape, color, color variation, and size). These descriptors justify the class labels employed during image annotation and highlight the fine-grained features the YOLOv8m network must learn to distinguish during automated detection.

Table 1. Image, name, variety, and morphological description of the eighth studied pepper.

Image of the Peppers	Name/Variety	Description
	Biquinho/ <i>Capsicum chinense</i>	Small, teardrop-shaped, glossy red fruits measuring 2–3 cm long and 1–2 cm in diameter.
	Bode/ <i>Capsicum chinense</i>	Round and bulbous, yellowish, typically yellow or red, 2–4 cm in both height and diameter, with a distinctly wrinkled surface.
	Cambuci/ <i>Capsicum baccatum</i>	Bell-shaped with pronounced lobes, green in color, measuring 4–6 cm in diameter and 3–5 cm in height.
	Chilli/ <i>Capsicum annuum</i>	Elongated and slender, bright red or chocolate, typically 8–12 cm long and 1–2 cm in diameter.
	Fidalga/ <i>Capsicum chinense</i>	Small and round, shiny yellow, about 2–3 cm in diameter and 2–4 cm in height.
	Habanero/ <i>Capsicum chinense</i>	Oval and lantern-shaped, orange, 3–6 cm long and 2–4 cm in diameter, with a wrinkled texture.
	Jalapeno/ <i>Capsicum annuum</i>	Medium-sized, cylindrical, dark green, measuring 5–9 cm in length and 2–4 cm in diameter, with a tapered end.
	Scotch Bonnet/ <i>Capsicum chinense</i>	Squashed-round shape, yellow, 4–6 cm in both height and diameter, with a wrinkled appearance.

### 3.2. Dataset Partitioning and Preprocessing Strategies

In the third step, the dataset was then divided into three distinct subsets to increase the volume and diversity of training samples, considering the limitations of the initial dataset:

- Training Set: 1245 images (84% of the total)
- Validation Set: 201 images (14% of the total)
- Test Set: 30 images (2% of the total)

Although the final test set comprised only 30 images (1% of the total dataset), this decision prioritized maximizing the volume and diversity of training data, which is critical for enhancing deep learning performance in object detection tasks. Although the test set was relatively small (30 images), it was carefully constructed to ensure visual diversity, encompassing variations in lighting, angles, and fruit morphology, thereby reflecting real-world application scenarios. To reduce the risk of overfitting, data augmentation techniques such as rotation and Gaussian blur were employed to introduce controlled variability and improve the model's generalization capacity. The stabilization of performance metrics after 100 epochs, along with consistent improvements observed with the use of augmented data, indicates a well-adjusted model. Approximately 900 instances were generated for each pepper variety at the end of the process. Subsequently, a second round of model training was conducted using an augmented dataset.

This methodological approach is consistent with the findings of Frizzi et al. [35], who verified that in data-scarce scenarios, a larger training partition combined with data augmentation effectively mitigates overfitting, provided the validation and test sets retain

representative variability. While this principle guides our methodology, we acknowledge that the limited size of the current test set constrains a definitive evaluation of the risk of overfitting. Future research should incorporate strategies such as cross-validation and an expanded test dataset to enhance the robustness and reliability of the findings.

### 3.3. Data Augmentation

In the fourth step, we employed data augmentation. This process enlarges a training dataset through techniques such as rotation, flipping, cropping, and contrast adjustment, thereby enhancing the model's generalization and performance [35]. Data augmentation techniques were applied to the training set to improve the generalization capacity of the YOLOv8m model and optimize its accuracy in pepper identification. This strategy aimed to simulate natural variations in real-world image capture conditions, making the model more robust to changes in lighting, angles, and image sharpness. The augmentation methods, such as rotation and blurring, were selected based on prior studies indicating their positive impact on YOLOv8m accuracy. In contrast, cropping was excluded due to its tendency to eliminate critical features in small objects [36].

In this study, data augmentation techniques were applied strategically in two distinct phases. Initially, the YOLOv8m model was trained on the original dataset to establish a baseline for performance. Subsequently, data augmentation methods such as rotation and Gaussian blur were introduced during a second training cycle, allowing for a more controlled assessment of their specific impact on the model's generalization capabilities. Although it is common practice to apply data augmentation before the initial training phase, as noted by Ioannou and Myronidis [22], the adopted approach offered clearer insights into how augmentation influenced convergence behavior and class-level detection performance. The observed improvements in metrics such as box precision and mAP during the second phase demonstrated that post-baseline augmentation not only enhanced generalization but also contributed to the model's robustness in recognizing subtle morphological differences among pepper varieties. Additionally, this phased methodology enabled more efficient use of computational resources by avoiding unnecessary dataset expansion before validating the model's initial learning performance.

Based on these indications, the following procedures were applied to the training dataset:

- Rotation: Images were randomly rotated at angles between  $-30^\circ$  and  $+30^\circ$ , increasing perspective variability and reducing the model's dependence on fixed angles.
- Blurring: A soft Gaussian blur filter ( $3 \times 3$  kernel) was applied to simulate focus variations that may arise due to camera movement or differences in image quality.
- Following data augmentation, the dataset increased to a total of 3964 images, with the final data split structured as follows:
- Training Set: 3733 images (94% of the total)
- Validation Set: 201 images (5% of the total)
- Test Set 30 images (1% of the total)

The dataset was divided into training (94%), validation (5%), and test (1%) subsets to increase the volume and diversity of training samples, considering the limitations of the initial dataset. Tables 2 and 3 provide the specifications necessary for reproducibility of the training process. This configuration represents a standard cloud-based training setup, ensuring that the model can be reproduced under similar computational conditions.

Hyperparameters were defined based on the default configurations provided in the `defaults.yaml` file of the official Ultralytics YOLOv8 implementation. These values reflect established best practices for training object detection models [37]. The use of standardized configurations ensures methodological consistency with the YOLOv8 framework and facilitates the comparability of results across related studies.

**Table 2.** Hardware specifications and relevant software environments for the reproducibility of the training process.

Parameter	IT Environment
Operating system	Ubuntu 20.04/Google Colab VM
Processor	Intel(R) Xeon(R) CPU @ 2.20 GHz
RAM	13 GB
GPU	NVIDIA Tesla T4 (16 GB VRAM)
CUDA version	11.8
Python version	3.10
YOLOv8 version	Ultralytics 8.1.0
PyTorch version	2.0.1
Training environment	Google Colab

**Table 3.** Key hyperparameters adopted for training the YOLOv8m model.

Hyperparameter	Value	Description
epochs	200	Number of training iterations
batch_size	16	Training samples per batch
imgsz	640	Input image resolution (square)
lr0	0.01	Initial learning rate
lrf	0.01	Final learning rate decay factor
momentum	0.937	SGD momentum
weight_decay	0.0005	Regularization parameter
optimizer	auto	Optimizer automatically selected (typically SGD)
patience	100	Early stopping patience (epochs without gain)
mosaic	1.0	Mosaic augmentation enabled
flplr	0.5	Horizontal flip probability
translate	0.1	Image shift augmentation
scale	0.5	Image scaling range for augmentation

### 3.4. YOLO Usage

The selection of the YOLOv8m model is justified by its superior performance in agricultural contexts, particularly in terms of accuracy, inference speed, and model compactness, compared to alternatives such as EfficientDet and CenterNet. YOLO's architecture, which frames object detection as a single-pass regression problem, provides a significant velocity advantage crucial for real-time applications [12].

This claim is substantiated by a systematic review of 30 studies by Badgugar et al. [26], which concluded that YOLO models consistently outperform frameworks such as SSD, Faster R-CNN, and EfficientDet in agricultural tasks. Given its verified efficiency and accuracy, YOLOv8m is well suited for the present study on varietal pepper classification in uncontrolled environments. Figure 2 illustrates the flow of features through convolutional layers, C2f modules, and the anchor-free detection head.

The fifth step involved model training conducted using the Google Colab Pro infrastructure [38], which ensured access to hardware acceleration for deep learning tasks. The computational configuration comprised an NVIDIA Tesla T4 GPU with 15 GB of VRAM, driver version 550.54.15, and CUDA version 12.4. Available memory was 15.36 GB, with GPU power consumption ranging from 9 W (minimum usage) to 70 W (maximum capacity).

The frameworks used were Python 3.10.12 [39], PyTorch 2.5.1 [40], and Ultralytics YOLOv8m [37], along with OpenCV [41] and Roboflow [33]. Google Colab Pro was selected due to its access to high-performance GPUs, which enables the efficient training of the YOLOv8m model within a reduced time frame. The programming language used was Python [39], in combination with several specialized libraries, including:

- OpenCV: For image manipulation.
- TensorFlow: For model construction and training.
- PyTorch: An alternative deep learning framework [40].

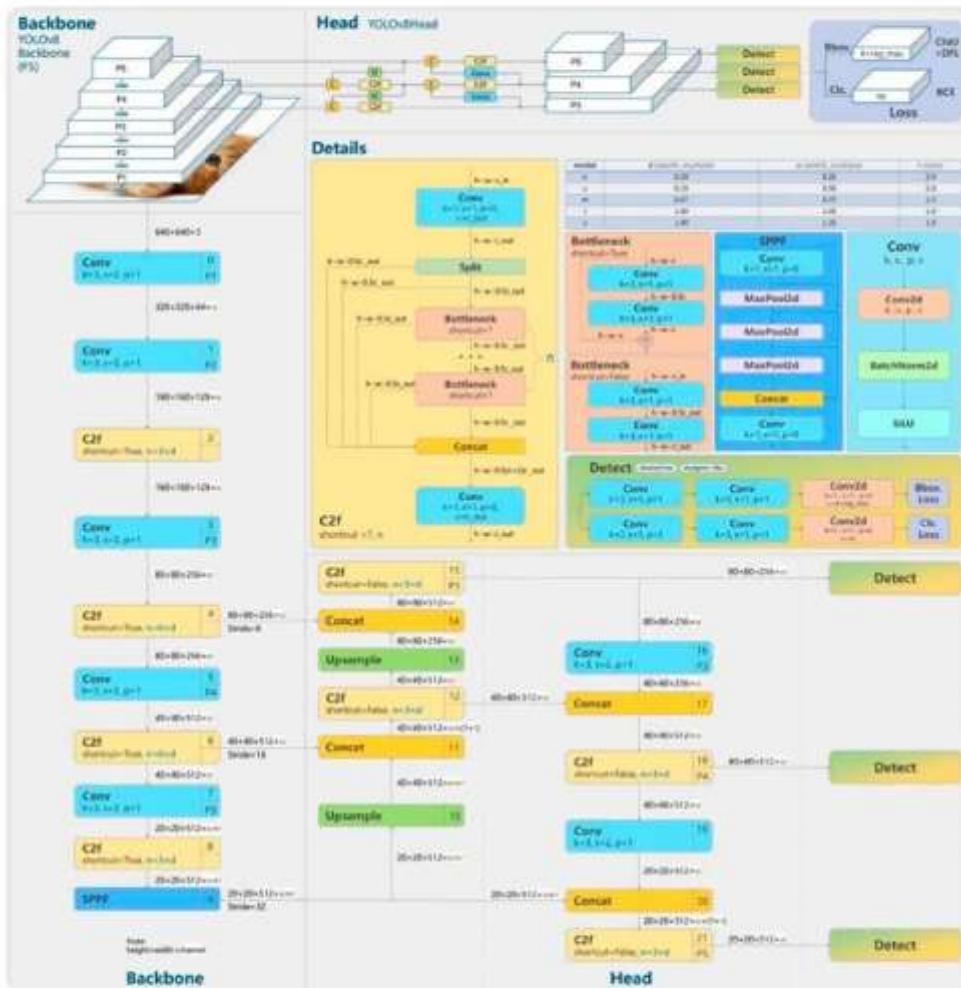


Figure 2. Diagram of the YOLOv8 architecture, including backbone, neck, and head components. Adapted from [33].

After training, the model's performance was evaluated using the following metrics provided by the YOLOv8m CNN (Equations (1)–(3)):

- Box Precision (BoxP): Measures the accuracy of bounding box predictions relative to the annotated boxes (Equation (1)).

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad (1)$$

where  $TP$  = true positive, and  $FP$  = false positive. The  $TP$  is a predicted bounding box that correctly matches a ground truth bounding box of the same class.  $FP$  is a predicted bounding box that does not correspond to a real object.

- Recall (R): Measures the proportion of true positives among all positive instances in the dataset (Equation (2)).

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2)$$

where  $TP$  = true positive, and  $FN$  = false negative. The  $FN$  is a ground truth bounding box that the model failed to detect.

- Mean Average Precision at 50% IoU (mAP<sub>50</sub>): Calculates the average precision across different intersection-over-union (IoU) thresholds, focusing on a 50% threshold (Equation (3)). The mAP<sub>50</sub> is a standard metric for evaluating object detection models. It represents the average of the average precision (AP) calculated for each object class. For this metric, a predicted bounding box is considered a true positive only if its intersection over union (IoU) with a ground truth box of the same class is 0.50 or greater. The AP for each class is the area under its corresponding precision–recall curve, reflecting the trade-off between precision and recall at varying detection confidence thresholds under the 0.50 IoU criterion.

$$\text{mAP}_{50} = \frac{1}{N} \times \sum [\text{AP}_{c, 0.50}] \text{ from } c = 1 \text{ to } N \quad (3)$$

where  $N$  = total number of object classes, and  $\text{AP}_{c, 0.50}$  = average precision for class  $c$ , calculated using an IoU threshold of 0.50. This metric refers to the area under the precision–recall curve for class  $c$ , where predictions are counted as true positives if their IoU with ground truth is greater than 0.50.

- Mean average precision at 95% IoU (mAP<sub>95</sub>): Calculates the average precision at a stricter 95% IoU threshold (Equation (4)).

$$\text{mAP}_{95} = \frac{1}{N} \times \sum [\text{AP}_{c, 0.95}] \text{ from } c = 1 \text{ to } N \quad (4)$$

where  $N$  = total number of object classes, and  $\text{AP}_{c, 0.95}$  = average precision for class  $c$ , calculated using a fixed IoU threshold of 0.95. This metric is the area under the precision–recall curve for class  $c$ , where predictions are counted as true positives if their IoU with ground truth is greater than 0.95.

The YOLOv8m model was chosen for the present study due to its proven efficiency in real-time object detection tasks, aligning with the precision and speed requirements for identifying pepper varieties. This version represents a significant advancement within the YOLO family, combining architectural enhancements and advanced training techniques, making it particularly suited for agricultural applications. According to Terven and Cordova-Esparza [11], YOLOv8m achieved superior performance in metrics such as mean average precision (mAP) compared to earlier models, including YOLOv5, and even competing approaches like Faster R-CNN. The results were compared between datasets with and without augmentation, highlighting the impact of data augmentation on the YOLOv8m model's performance in pepper detection. Figure 3 shows the workflow of the methodology.

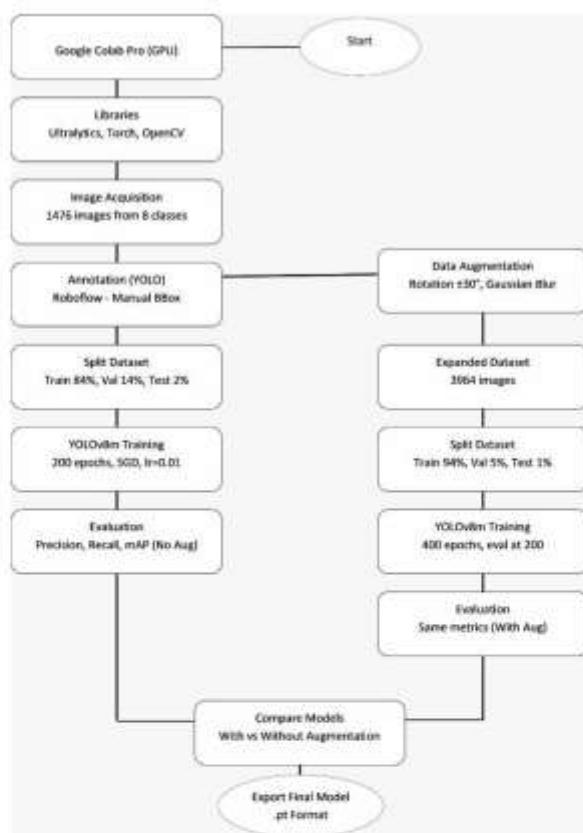


Figure 3. Methodology workflow.

#### 4. Results

This study investigated the training of the YOLOv8m convolutional neural network (CNN) to identify eight distinct varieties of peppers. The model's performance was evaluated both with and without the application of data augmentation techniques, considering key metrics such as bounding box precision (BoxP), recall, and mean average precision across different intersection over union (IoU) thresholds (mAP<sub>50-95</sub>).

Table 4 summarizes the evolution of global detection metrics—bounding-box precision (BoxP), recall, mAP<sub>50</sub>, and mAP<sub>50-95</sub>—at four representative training epochs (10, 50, 100, 170) for models trained with and without augmentation. The results indicate that rotation and Gaussian-blur augmentation accelerated convergence and yielded substantial gains.

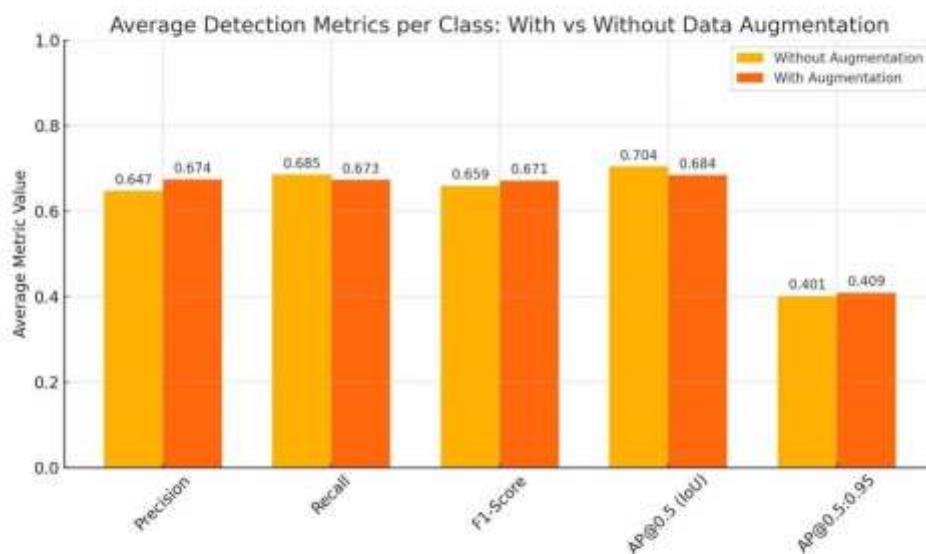
Table 4. Summary of the model training to identify peppers.

Epochs	BoxP	BoxP with Aug.	Recall	Recall with Aug.	mAP <sub>50</sub>	mAP <sub>50</sub> with Aug.	mAP <sub>50-95</sub>	mAP <sub>50-95</sub> with Aug.
10	0.169	0.340	0.206	0.387	0.160	0.360	0.061	0.163
50	0.526	0.705	0.570	0.605	0.562	0.696	0.278	0.369
100	0.626	0.749	0.599	0.576	0.641	0.676	0.334	0.403
170	0.694	0.684	0.600	0.638	0.673	0.694	0.361	0.371

The results presented in Table 4 indicate the model's performance progression throughout the training process, highlighting the positive effects of data augmentation. The application of these techniques led to improvements across all evaluated metrics.

Continuing training beyond 100 epochs revealed the stabilization of performance metrics, indicating that the model had reached an appropriate point of convergence for the dataset. This behavior suggests that increasing the number of epochs beyond this threshold would not yield significant improvements in the model's learning performance.

Figure 4 shows that the results display consistent improvements in box precision, recall, and mean average precision (mAP) when augmentation techniques were applied, indicating enhanced generalization capabilities of the model.



**Figure 4.** Comparison of object detection performance metrics between training with and without data augmentation.

The comparative analysis between models trained with and without data augmentation techniques revealed consistent improvements in the average values of precision, F1-score, and AP@0.5:0.95. These results underscore the importance of incorporating data augmentation methods to enhance the robustness and generalization capacity of object detection models. As emphasized by Abdulkareem et al. [42], applying geometric transformations such as rotation, flipping, and scaling increases the visual diversity of the training dataset. This broader variability enables convolutional neural networks, particularly those based on the YOLO architecture, to detect objects more accurately across different orientations, scales, and visual perspectives.

Table 5 provides the class performance values (BoxP, recall, mAP<sub>>50</sub>, and mAP<sub>>50-95</sub>) for the best non-augmented and augmented models after 170 epochs. The table highlights how varietal morphology affects detector sensitivity and where further targeted data enrichment may be necessary.

**Table 5.** Result of each class in the last training.

Class	BoxP	BoxP with Aug.	Recall	Recall with Aug.	mAP50	mAP50 with Aug.	mAP50-95	mAP50-95 with Aug.
Biquinho-Pepper	0.693	0.658	0.516	0.486	0.636	0.562	0.692	0.386
Bode-Pepper	0.618	0.638	0.436	0.564	0.523	0.397	0.566	0.293
Cambuci-Pepper	0.747	0.852	0.758	0.777	0.816	0.827	0.884	0.596
Chilli-Pepper	0.713	0.750	0.595	0.729	0.648	0.790	0.622	0.345
Fidalga-Pepper	0.768	0.652	0.579	0.592	0.701	0.679	0.626	0.377
Habanero-Pepper	0.636	0.577	0.613	0.667	0.641	0.693	0.748	0.675
Jalapeno-Pepper	0.622	0.595	0.775	0.75	0.702	0.721	0.422	0.489
ScotchBonnet-Pepper	0.756	0.746	0.527	0.542	0.720	0.708	0.404	0.415

The data show that the impact of data augmentation varied across classes, with some varieties benefiting more from the technique than others (Supplementary material). Peppers that showed improvement with data augmentation:

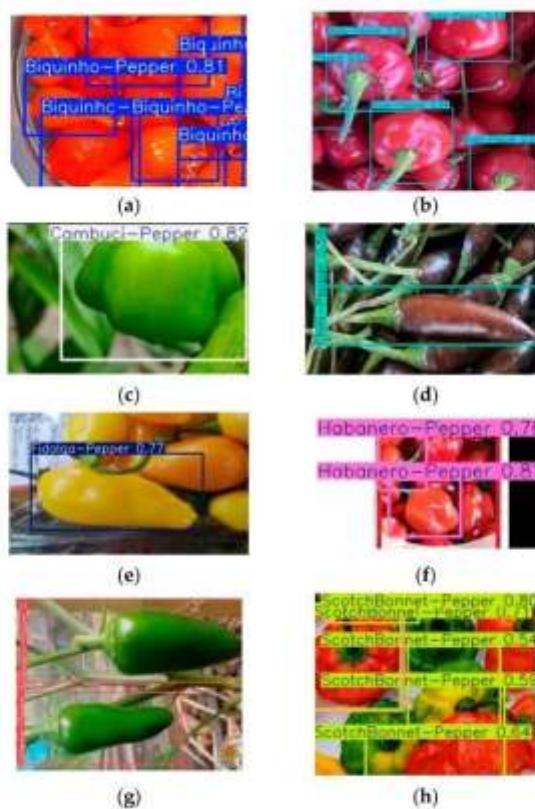
- Cambuci-Pepper achieved the highest gains through data augmentation, reaching 0.852 in BoxP (+14%) and 0.827 in mAP50 (+1.1%).
- Chilli-Pepper also identified significant improvements, particularly in recall (+22%) and mAP50-95 (+31%).

The peppers with more stable or decreased performance with data augmentation were the Fidalga Pepper, which showed a reduction in precision (BoxP dropped from 0.768 to 0.652), suggesting that the model may have been affected by noise in the newly generated samples, and Habanero-Pepper, which also showed a decline in BoxP (from 0.636 to 0.577); however, it showed an improvement in recall (+8.8%), indicating that more objects were detected, even though with lower precision.

The peppers that consistently performed well under both conditions were the Jalapeno pepper and Scotch Bonnet pepper, which exhibited minimal variation between tests. This result suggests that the model effectively captures their visual characteristics, regardless of the training strategy.

Observations of the results on test images also raise several considerations. Figure 5 presents eight subfigures (a to h) showing the YOLOv8m model's detection of each pepper class on test images. The eight pepper varieties evaluated in the study are (a) Biquinho, (b) Bode, (c) Cambuci, (d) Chilli, (e) Fidalga, (f) Habanero, (g) Jalapeno, and (h) Scotch Bonnet. The bounding boxes inside the images highlight the detections made by the model, along with the confidence scores assigned to each prediction.

The confidence values vary from image to image, as they depend on the specific conditions of each sample evaluated by the model. This variation in confidence arises from multiple factors, including lighting and contrast, the position and angle of the peppers, instance overlap, image quality, and resolution. Such fluctuations in confidence values are expected and reflect the model's robustness when dealing with diverse capture conditions. However, the variability in prediction confidence suggests that the model can still be optimized to reduce fluctuations in values across different images. Strategies such as adjusting the detection threshold, refining class balance, and implementing post-processing techniques could help improve the consistency of predictions.



**Figure 5.** Example of results of the model tests using samples of the *Capsicum* fruits: (a) Biquinho; (b) Bode; (c) Cambuci; (d) Chilli; (e) Fidalga; (f) Habanero; (g) Jalapeno; and (h) Scotch Bonnet. The bounding box defines the spatial location and extent of the detected object.

## 5. Discussion

The reduction in precision observed for the Fidalga-Pepper and Habanero-Pepper classes may be attributed to the introduction of variations that do not accurately represent the actual distribution of the training images. The reduced performance observed in certain pepper varieties, such as Fidalga, is likely attributable to their visual similarity to other yellow, rounded types like Habanero and Scotch Bonnet, combined with the possible introduction of unrealistic variations during data augmentation, which may have impaired the model's ability to distinguish subtle inter-class differences. This effect has already been documented in the literature, as noted by Yilmaz and Kutbay [36], who indicate that data augmentation techniques can, in some cases, generate unrealistic instances, hindering the model's generalization capacity. The variation in performance across classes suggests that some pepper varieties possess visual characteristics that facilitate detection, while others require further refinement of the model to improve detection.

The application of data augmentation also impacted the total training time of the model. Without augmentation, training was completed in 200 epochs, totaling 3 h and 43 min. Using the augmentation technique, there was a reduction in epochs (170), but the processing time more than doubled, reaching 6 h and 54 min. This increase in processing

time, from 3 h and 43 min without augmentation to 6 h and 54 min with augmentation, can be attributed to the larger volume of images and the increased computational complexity introduced by greater sample diversity. This impact on training time aligns with the findings of Yilmaz and Kutbay [36], who report that applying data augmentation techniques can significantly increase model execution time, especially when implemented on large datasets. The increase in training time needs to be weighed against the gains in model accuracy. Although data augmentation yielded better performance for some pepper classes, the computational cost remains a critical factor for deploying the model in real-world applications.

The present study's findings confirm previous research [43], indicating the effectiveness of computer vision in agricultural product identification while highlighting the challenges associated with automated pepper detection. An earlier study [17] highlights the increasing application of computer vision in agricultural automation, noting that CNN-based models can replace traditional screening methods, thereby reducing time and minimizing human error. Similarly, Li et al. [10] show that the performance of CNNs is highly dependent on the quantity and diversity of the dataset, which explains the positive impact of data augmentation observed in the present study.

The choice of YOLOv8m for this study aligns with research by Hussain [25] and Terven and Cordova-Esparza [11], who highlight the YOLO architecture's advances in efficiency and speed in agricultural object detection. This study reinforces these findings, demonstrating that YOLOv8m can accurately identify pepper varieties, especially when trained on a diverse dataset.

We observed a clear trend based on the experimental results: model performance, as indicated by metrics such as mAP50 and AP50-95, generally improves with increased training from 10 to 200 epochs without data augmentation. However, further training to 400 epochs yields diminishing returns and introduces the risk of overfitting. In contrast, data augmentation, particularly evident after 50 epochs, substantially enhances the model's generalization capabilities across the eight distinct pepper varieties. This result points out the crucial role of data augmentation in enabling extended training durations to achieve robust and accurate fine-grained classification.

The YOLOv8m model indicated a notable improvement in performance after applying data augmentation techniques. Key metrics such as box precision (BoxP) and recall showed significant gains after 50 epochs, with further stabilization observed after 100 epochs. For instance, the mAP50 for Chilli-Pepper increased from 0.648 to 0.790 with augmentation, while the recall for Bode peppers improved from 0.436 to 0.564. The precision and recall values across the eight pepper varieties indicate the model's capacity to differentiate effectively between them. These results underscore the crucial role of data augmentation in enhancing the performance of CNN models in agricultural contexts, laying a solid foundation for future advancements in automated crop identification and management systems.

Future research should explore advanced architectural enhancements, such as integrating attention mechanisms (e.g., SE-Net, CBAM), to improve spatial and channel feature refinement. Alternatively, employing ensemble approaches that combine predictions from multiple models could increase robustness across heterogeneous field conditions. Investigating lightweight architectures (e.g., MobileNet, YOLO-NAS) may also support deployment in edge devices used in precision agriculture.

## 6. Conclusions

The present study presents a practical application of the YOLOv8m CNN in identifying eight distinct *Capsicum* pepper varieties. Implementing data augmentation techniques significantly enhanced the model's precision, recall, and mean average precision (mAP)

metrics. These findings underscore the potential of deep learning to revolutionize food quality practices, particularly in automating management tasks.

Although this research identified the effectiveness of the YOLOv8m model for pepper identification, further research is essential to enhance the model's robustness in real-world scenarios by including images captured under diverse real-world conditions, such as varying lighting intensities, partial occlusions from overlapping produce, heterogeneous backgrounds, and differences in image resolution and camera angles. Exploring advanced techniques, such as transfer learning and alternative object detection models, could improve the model's generalization capacity and adaptability.

**Supplementary Materials:** The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/app15137024/s1>. All results with and without the augmentation procedure.

**Author Contributions:** Conceptualization, M.d.O.B. and L.d.S.C.; methodology, M.d.O.B.; software, F.P.L.A.; validation, S.d.S.S., M.d.O.B. and F.P.L.A.; formal analysis, I.d.A.N.; investigation, M.d.O.B.; resources, M.d.O.B.; data curation, S.d.S.S.; writing—original draft preparation, M.d.O.B.; writing—review and editing, I.d.A.N.; visualization, L.d.S.C.; supervision, M.T.O. and I.d.A.N. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data will be made available upon request.

**Acknowledgments:** The authors thank the Coordination for the Advancement of Higher Education—CAPES for the scholarship and the cooperation between Universidade Paulista and CEAGESP, which made the images available.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

YOLO	You only look once (real-time object detection framework)
CNN	Convolutional neural network
MobileNet	CNN architecture developed by Google, designed for mobile applications
SE-Net	Squeeze-and-excitation network (a channel attention mechanism for improving CNN performance)
CBAM	Convolutional block attention module (a lightweight attention module combining spatial and channel attention)
YOLO-NAS	YOLO with neural architecture search (an optimized variant of YOLO using automated architecture search for improved accuracy and efficiency)

## References

1. Riaz, U.; Iqbal, S.; Sohail, M.I.; Samreen, T.; Ashraf, M.; Akmal, F.; Siddiqui, A.; Ahmad, I.; Naveed, M.; Khan, N.I.; et al. A comprehensive review on emerging importance and economical potential of medicinal and aromatic plants (MAPs) in current scenario. *Pak. J. Agri. Res.* **2021**, *34*, 381–392. [[CrossRef](#)]
2. Ganguly, S.; Praveen, K.P.; Para, P.A.; Sharma, V. Medicinal properties of chili pepper in human diet. *AJPHCM* **2017**, *2*, 6–7.
3. Shi, Z. Chilli intake is inversely associated with hypertension among adults. *Clin. Nutr. ESPEN* **2018**, *23*, 67–72. [[CrossRef](#)]
4. Food and Agriculture Organization of the United Nations (FAO). Production (t). FAO Statistics Database. Available online: <https://www.fao.org/statistics/en> (accessed on 18 December 2023).
5. da Costa Ribeiro, C.S.; Amaro, G.B.; Reifsneider, F.J.B.; Carvalho, S.I.C. Cultivars. Embrapa Hortaliças. Available online: <https://www.embrapa.br/agencia-de-informacao-tecnologica/cultivos/pimenta/pre-producao/caracteristicas/cultivares> (accessed on 18 January 2024).

6. Lillywhite, J.; Robinson, C. Understanding chile pepper consumers' preferences: A discrete choice experiment. *Agriculture* **2023**, *13*, 1792. [[CrossRef](#)]
7. Johnson, N.A.N.; Adade, S.Y.S.S.; Ekumah, J.N.; Kwadzokpui, B.A.; Xu, J.; Xu, Y.; Chen, Q. A comprehensive review of analytical techniques for spice quality and safety assessment in the modern food industry. *Crit. Rev. Food Sci. Nutr.* **2025**, *1*–26. [[CrossRef](#)]
8. Huynh, Q.K.; Nguyen, C.N.; Vo-Nguyen, H.P.; Tran-Nguyen, P.L.; Le, P.H.; Le, D.K.L.; Nguyen, V.C. Crack identification on the fresh chilli (*Capsicum*) fruit destemmed system. *J. Sens.* **2021**, *1*, 8838247. [[CrossRef](#)]
9. Zhong, Y.; Cheng, Y.; Ruan, M.; Ye, Q.; Wang, R.; Yao, Z.; Zhou, G.; Liu, J.; Yu, J.; Wan, H. High-throughput SSR marker development and the analysis of genetic diversity in *Capsicum frutescens*. *Horticulture* **2021**, *7*, 187. [[CrossRef](#)]
10. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Trans. Neural Netw. Learn.* **2021**, *33*, 6999–7019. [[CrossRef](#)]
11. Terven, J.; Córdova-Esparza, D.M.; Romero-González, J.A. A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Mach. Learn. Knowl.* **2023**, *5*, 1680–1716. [[CrossRef](#)]
12. Yue, X.; Li, H.; Song, Q.; Zeng, F.; Zheng, J.; Ding, Z.; Kang, G.; Cai, Y.; Lin, Y.; Xu, X.; et al. YOLOv7-GCA: A Lightweight and High-Performance Model for Pepper Disease Detection. *Agronomy* **2024**, *14*, 618. [[CrossRef](#)]
13. Dai, M.; Sun, W.; Wang, L.; Dorjoy, M.M.H.; Zhang, S.; Miao, H.; Han, L.; Zhang, X.; Wang, M. Pepper leaf disease recognition based on enhanced lightweight convolutional neural networks. *Front. Plant Sci.* **2023**, *14*, 1230886. [[CrossRef](#)]
14. Aguiar, E.P.L.; Niñas, I.A.; Okano, M.T. Bridging the Gap Between Computational Efficiency and Segmentation Fidelity in Object-Based Image Analysis. *Animals* **2024**, *14*, 3626. [[CrossRef](#)] [[PubMed](#)]
15. Li, X.; Pan, J.; Xie, F.; Zeng, J.; Li, Q.; Huang, X.; Liu, D.; Wang, X. Fast and accurate green pepper detection in complex backgrounds via an improved YOLOv4-tiny model. *Comput. Electron. Agr.* **2021**, *191*, 106503. [[CrossRef](#)]
16. Paul, A.; Machavaram, R.A.; Kumar, D.; Nagar, H. Smart solutions for capsicum Harvesting: Unleashing the power of YOLO for Detection, Segmentation, growth stage Classification, Counting, and real-time mobile identification. *Comput. Electron. Agr.* **2024**, *219*, 108832. [[CrossRef](#)]
17. Tian, H.; Wang, T.; Liu, Y.; Qiao, X.; Li, Y. Computer vision technology in agricultural automation—A review. *Inf. Process Agric.* **2020**, *7*, 1–19. [[CrossRef](#)]
18. Sarker, I.H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Comput. Sci.* **2021**, *2*, 420. [[CrossRef](#)]
19. Eunice, J.; Popescu, D.E.; Chowdary, M.K.; Hernanth, J. Deep learning-based leaf disease detection in crops using images for agricultural applications. *Agronomy* **2022**, *12*, 2395. [[CrossRef](#)]
20. Li, X.; Wang, L.; Miao, H.; Zhang, S. Aphid Recognition and Counting Based on an Improved YOLOv5 Algorithm in a Climate Chamber Environment. *Insects* **2023**, *14*, 839. [[CrossRef](#)]
21. López-Barrios, J.D.; Escobedo Cabello, J.A.; Gómez-Espinosa, A.; Montoya-Cávera, L.-E. Green sweet pepper fruit and peduncle detection using Mask R-CNN in greenhouses. *Appl. Sci.* **2023**, *13*, 6296. [[CrossRef](#)]
22. Ioannou, K.; Myronidis, D. Automatic Detection of Photovoltaic Farms Using Satellite Imagery and Convolutional Neural Networks. *Sustainability* **2021**, *13*, 5323. [[CrossRef](#)]
23. Armob, A.S.; Kausik, A.K.; Islam, Z.; Khan, R.; Rashid, A.B. Comparative result analysis of cauliflower disease classification based on deep learning approach VGG16, inception v3, ResNet, and a custom CNN model. *Hybrid. Adv.* **2025**, *10*, 100440. [[CrossRef](#)]
24. Roy, P.S.; Kukreja, K.V. Vision transformers for rice leaf disease detection and severity estimation: A precision agriculture approach. *J. Saudi Soc. Agric. Sci.* **2025**, *24*, 3. [[CrossRef](#)]
25. Hussain, M. YOLOv1 to v8: Unveiling Each Variant—A Comprehensive Review of YOLO. *IEEE Access* **2024**, *12*, 42816–42833. [[CrossRef](#)]
26. Badgujar, C.M.; Poulose, A.; Gan, H. Agricultural Object Detection with You Only Look Once (YOLO) Algorithm: A Bibliometric and Systematic Literature Review. *Comput. Electron. Agr.* **2024**, *223*, 109090. [[CrossRef](#)]
27. Ren, R.; Zhang, S.; Sun, H.; Gao, T. Research on pepper external quality detection based on transfer learning integrated with convolutional neural network. *Sensors* **2021**, *21*, 5305. [[CrossRef](#)]
28. Kurtulmus, F.; Alibas, I.; Kavdir, I. Classification of pepper seeds using machine vision based on neural network. *Int. J. Agric. Biol. Eng.* **2016**, *9*, 51–62. [[CrossRef](#)]
29. Zeng, Y.; Zhao, Y.; Yu, Y.; Tang, Y.; Tang, Y. Pepper disease detection model based on convolutional neural network and transfer learning. *IOP Conf. Ser. Earth Environ. Sci.* **2021**, *792*, 012001. [[CrossRef](#)]
30. Mohi-Alden, K.; Omid, M.; Soltani Firouz, M.; Nasiri, A. Design and evaluation of an intelligent sorting system for bell pepper using deep convolutional neural networks. *J. Food Sci.* **2022**, *87*, 289–301. [[CrossRef](#)]
31. Thakur, P.; Chug, A.; Singh, A.P. Plant Disease Detection of Bell Pepper Using Transfer Learning over Different Models. In Proceedings of the 8th International Conference on Signal Processing and Integrated Networks (SPIN 2021), Noida, India, 26–27 August 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 384–389. [[CrossRef](#)]

32. Antunes, S.N.; Okano, M.T.; Nääs, I.A.; Lopes, W.A.C.; Aguiar, F.P.L.; Vendrametto, O.; Fernandes, J.C.L.; Fernandes, M.E. Model Development for Identifying Aromatic Herbs Using Object Detection Algorithm. *AgriEngineering* **2024**, *6*, 1924–1936. [CrossRef]
33. Roboflow. Available online: <https://roboflow.com/> (accessed on 12 January 2024).
34. Matuck, G.R.; Castro, A.J.A.; da Silva, L.E.; Carvalho, E.G. Facial Recognition With Artificial Intelligence Using The Roboflow Platform. *Rev. Prociências* **2023**, *6*, 114–131. [CrossRef]
35. Frizzi, S.; Bouchouicha, M.; Gineux, J.M.; Moreau, E.; Sayadi, M. Convolutional neural network for smoke and fire semantic segmentation. *IET Image Process.* **2021**, *15*, 634–647. [CrossRef]
36. Yilmaz, B.; Kutbay, U. YOLOv8 based drone detection: Performance analysis and optimization. *Computers* **2024**, *13*, 234. [CrossRef]
37. Ultralytics. Ultralytics YOLOv8m Docs. Available online: <https://docs.ultralytics.com/> (accessed on 12 November 2024).
38. Google Colab Pro. Available online: <https://colab.google.com> (accessed on 12 November 2024).
39. Python, Version 3.10.12. Python Software Foundation. 2023. Available online: <https://www.python.org/> (accessed on 12 November 2024).
40. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32, pp. 8024–8035. [CrossRef]
41. Open Source Computer Vision Library. OpenCV. Available online: <https://opencv.org/> (accessed on 15 November 2024).
42. Abdulkareem, I.M.; AL-Shamuri, E.K.; Khalid, N.A.A.; Omran, N.A. Proposed approach for object detection and recognition by deep learning models using data augmentation. *Int. J. Online Biomed. Eng.* **2024**, *20*, 5. [CrossRef]
43. Zhang, F.; Fu, L.S. Application of computer vision technology in agricultural field. *Appl. Mech. Mater.* **2014**, *462*, 72–76. [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

## ANEXO B



870250108484  
26/11/2025 23:09  
29409192348831013

Pedido de Registro de Programa de Computador - RPC

Número do Processo: 512025006150-1

Dados do Titular

Titular 1 de 2

Nome ou Razão Social: MARCELO TSUGUO OKANO  
Tipo de Pessoa: Pessoa Física  
CNPJ/CNPJ: 10094147859  
Nacionalidade: Brasileira  
Qualificação Física: Pesquisador  
Endereço: Rua Engenheiro Massonet Scocimelli, 54  
Cidade: Santo André  
Estado: SP  
CEP: 09760290  
País: Brasil  
Telefone: (11)41256096  
Fax:  
Email: piano@vap.com.br

**PETICIONAMENTO** Esta solicitação foi enviada pelo sistema Peticionamento Eletrônico em  
ELETRÔNICO 26/11/2025 às 23:09, Pedido 870250108484

Pedido 870250108484, de 26/11/2025, pág. 05

## Titular 2 de 2

Nome ou Razão Social: MADALENA DE OLIVEIRA BARBOSA  
 Tipo de Pessoa: Pessoa Física  
 CPF/CNPJ: 32866000832  
 Nacionalidade: Brasileira  
 Qualificação Física: Mestramento  
 Endereço: Pedro Galvão, 8 Apt 106, Tomé I - Parque São Vicente  
 Cidade: Mauá  
 Estado: SP  
 CEP: 09371-605  
 País: BRASIL  
 Telefone: (11) 993 931151  
 Fax:  
 Email: madalena.oliveira@elec.ap.gov.br

## Dados do Programa

Data de Publicação: 1/1/2025

Data de Criação: 1/1/2025

• § 2º do art. 2º da Lei 9.609/98: "Fica assegurada a tutela dos direitos relativos a programa de computador pelo prazo de cinquenta anos contados a partir de 1º de janeiro do ano subsequente ao da sua publicação ou, na ausência desta, da sua criação"

Título: APP Primatas

Algoritmo hash: SHA-512 - Secure Hash Algorithm

Resumo digital hash: 390b6300289883e4703b43c12a4687670094ac9443090a86159  
 3e1013a45c75942a6220f593233893d3f3f8e6277fe66785c  
 e54591af130cc01

§1º e Incisos VI e VII do §2º do Art. 2º do Instrução Normativa: O titular é o responsável único pela transformação, em resumo digital hash, dos trechos do programa de computador e demais dados considerados suficientes para identificação e caracterização, que serão motivo do registro. O titular terá a inteira responsabilidade pela guarda da informação sigilosa definida no inciso II, § 1º, art. 2º da Lei 9.609 de 19 de fevereiro de 1998.

PETICIONAMENTO ELETRÔNICO Esta solicitação foi enviada pelo sistema Peticionamento Eletrônico em 26/11/2025 às 23:09, Petição 570250106454

Petição 570250106454, de 26/11/2025, pág. 25

Linguagens: PYTHON

JAVA SCRIPT

HTML

JSON

Campo de Aplicação: IN02 - TECNOLOGIA (POLÍTICA TECNOLÓGICA, COOPERAÇÃO TÉCNICA, PESQUISA TECNOLÓGICA, INovaÇÃO TECNOLÓGICA, TECNOLOGIA APROPRIADA, QUÍMICA TECNOLÓGICA) IN03-ENGENHARIA (DESENHO TÉCNICO, ENGENHARIA METALURGICA, ENGENHARIA QUÍMICA, MECÂNICA, AUTOMOTIVA, AERONAUTICA, NAVAL, DE PRODUÇÃO, DE TESTES)

Tipo de Programa: AP01 - APLICATIVOS

**PETIÇÃO** Esta solicitação foi enviada pelo sistema Petição Eletrônica em  
ELETRÔNICO 26/11/2025 às 23:09, Pequeno 570250108484

Petição 570250108484, de 26/11/2025, pag. 35

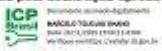
<b>Dados do Autor</b>
Autor 1 de 2
Nome: MADALENA DE OLIVEIRA BARBOSA CPF: 32680502052 Nacionalidade: Brasileira Qualificação Física: Mestrandra Endereço: Pedro Galvão, 8 Apt 188, Torre I - Parque São Vicente Cidade: Mauá Estado: SP CEP: 09371-005 País: BRASIL Telefone: (11) 993 931151 Fax: Email: madalena.olivinha@elec.ca.gov.br
Autor 2 de 2
Nome: MARCELO TSUGUÍD OKANO CPF: 10094147859 Nacionalidade: Brasileira Qualificação Física: Pesquisador Endereço: Rua Engenheiro Masantel Scocimelli, 54 Cidade: São Bernardo do Campo Estado: SP CEP: 09760-290 País: BRASIL Telefone: (11) 991 096575 Fax: Email: okano@vap.com.br
<b>Declaração de Veracidade - DV</b>
Nome: declaracaoVeracidade-1_assinado.pdf
<b>PETICIONAMENTO ELETRÔNICO</b> Esta solicitação foi enviada pelo sistema Peticionamento Eletrônico em 26/11/2025 às 23:09. Peticão 870250108454.
Processo 870250108454, de 26/11/2025, pág. 45

## DECLARAÇÃO DE VERACIDADE - CLIENTE

Em atendimento à Instrução Normativa em vigor eu, MARCELO TSUGUO OKANO, CPF: 100.941.478-09, declaro, para fins de direito, sob as penas da Lei e em atendimento ao art. 2º do Decreto nº 2.556<sup>a</sup>, de 20 de abril de 1998, que as informações feitas no formulário eletrônico de programa de computador – e-Software, são verdadeiras e autênticas.

Fico ciente através desse documento que a falsidade dessa declaração configura crime previsto no Código Penal Brasileiro e passível de apuração na forma da Lei.

Cliente das responsabilidades pela declaração apresentada, firmo a presente.



.....(assinado digitalmente).....

DECRETO Nº 2.556, DE 20 DE ABRIL DE 1998  
Art. 2º Os programas de computador perderão, a critério do titular dos respectivos direitos, ser registrados no Instituto Nacional da Propriedade Industrial - INPI.

Art. 2º A veracidade das informações de que trata o artigo anterior não se imputa responsabilidade do requerente, não prejudicando eventuais direitos de terceiros nem acarretando qualquer responsabilidade do Governo.

20-4001829-00026.0.2