



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS CORRENTE
CURSO TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

JOSÉ GUILHERME MOUTA FERREIRA

A DETECÇÃO DE DOENÇAS EM FOLHAS DO TOMATEIRO:
UMA AVALIAÇÃO COMPARATIVA ENTRE MODELOS DE APRENDIZADO
PROFUNDO

CORRENTE

2024



JOSÉ GUILHERME MOUTA FERREIRA

A DETECÇÃO DE DOENÇAS EM FOLHAS DO TOMATEIRO: UMA AVALIAÇÃO
COMPARATIVA DE MODELOS DE APRENDIZADO PROFUNDO

Trabalho de Conclusão de Curso apresentado como exigência parcial para obtenção do diploma do Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Corrente.

Orientador: Prof. Me. Carlos Estevão Bastos Sousa.

Coorientador: Prof. Dr. Matheus Emerick de Magalhães.

CORRENTE

2024

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD

Ferreira, José Guilherme Mouta

F383d A detecção de doenças em folhas do tomateiro : uma avaliação comparativa entre modelos de aprendizado profundo / José Guilherme MoutaFerreira. - 2024.
44 p.: il. color.

Trabalho de Conclusão de Curso (Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Corrente, 2024.

Orientador : Prof Me. Carlos Estevão Bastos Sousa..

Coorientador : Prof Dr. Matheus Emerick de Magalhães..

1. Analise e Desenvolvimento de Sistemas - estudo e ensino. 2. Tomates -Solanum lycopersicum. 3. Detecção de doenças. 4. Aprendizagem profundo. I.Título.

CDD - 004

Elaborado por Tefischer Huanderson Soares e Sousa CRB 3/1251

LISTA DE FIGURAS

Figura 1 - Esquema matemático de um Neurônio Artificial.	16
Figura 2 - Estrutura de uma Rede Neural Convolucional.	17
Figura 3 - Bloco residual	19
Figura 4 - Arquitetura do classificador MobileNet V2	20
Figura 5 - Arquitetura do classificador Inception V3	21
Figura 6 - Pipeline da metodologia	26
Figura 7 - Amostra de imagens da base de dados	28
Figura 8 - Arquitetura do sistema	29

LISTA DE TABELAS

Tabela 1 - Representação das doenças e sua descrição	12
Tabela 2 - Representação das doenças e sua descrição	13
Tabela 3 - Matriz de confusão.	22
Tabela 4 - Acurácia média e desvio padrão das arquiteturas de RNC implementadas	32
Tabela 5 - Matriz de confusão do classificador ResNet50	33
Tabela 6 - Resultado das métricas de avaliação do classificador ResNet50	33
Tabela 7 - Matriz de confusão do classificador MobileNet V2	34
Tabela 8 - Resultado das métricas de avaliação do classificador MobileNet V2	35
Tabela 9 - Matriz de confusão do classificador Inception V3	36
Tabela 10 - Resultado das métricas de avaliação do classificador Inception V3	36

SUMÁRIO

1 INTRODUÇÃO	9
1.1 OBJETIVOS	10
1.1.1 OBJETIVOS ESPECÍFICOS	10
1.2 ORGANIZAÇÃO DESTE TRABALHO	11
2 FUNDAMENTAÇÃO TEÓRICA	12
2.1 DOENÇAS FOLIARES DO TOMATE	12
2.2 AGRICULTURA DE PRECISÃO	14
2.3 REDES NEURAIS ARTIFICIAIS (RNA)	15
2.4 REDES NEURAIS CONVOLUCIONAIS (RNC)	16
2.5 RESNET50	18
2.6 MOBILENET V2	19
2.7 INCEPTION V3	20
2.8 MÉTRICAS DE AVALIAÇÃO	22
2.9 TRABALHOS RELACIONADOS	23
3 METODOLOGIA	26
3.1 BASE DE DADOS	27
3.2 ARQUITETURA	28
3.3 TECNOLOGIAS UTILIZADAS	29
3.3.1 PYTHON	30
3.3.2 TENSORFLOW	30
3.3.3 DJANGO REST FRAMEWORK (DRF)	30
4 RESULTADOS E DISCUSSÕES	32
4.1 RESULTADOS OBTIDOS PELO RESNET50	32
4.2 RESULTADOS OBTIDOS PELO MOBILENET V2	34
4.3 RESULTADOS OBTIDOS PELO INCEPTION V3	35
4.4 DISCUSSÕES	37
5 CONSIDERAÇÕES FINAIS	38
5.1 SUGESTÕES DE TRABALHOS FUTUROS	38
REFERÊNCIAS	40

RESUMO

Os tomates (*Solanum lycopersicum*) são uma fonte rica de nutrientes essenciais, como vitamina C e licopeno, desempenhando um papel crucial na dieta da população e na economia local. Este estudo surge da necessidade de enfrentar os desafios específicos no cultivo dessas frutas, concentrando-se na identificação de doenças foliares por meio de modelos de aprendizado profundo. Utilizando uma base de dados com 11.000 imagens de folhas do tomateiro, foram exploradas diversas classes de patologias, incluindo Mancha Bacteriana, Pinta Preta, Requeima, Molde de Folha, Septoriose, Ácaro-Aranha Bicolor, Ponto Alvo, Ondulação da Folha Amarela, Vírus do Mosaico, além de amostras saudáveis. Três modelos foram treinados: ResNet50, MobileNet V2 e Inception V3. Os resultados revelam que o Inception V3 se destaca, alcançando uma acurácia de 98%, seguido pelo ResNet50 com 96% e o MobileNet V2 com 95%. Esses achados enfatizam a consistência do Inception V3 na classificação de imagens de folhas do tomateiro. A partir do melhor modelo, foi desenvolvida uma API utilizando o Framework Django Rest, capaz de processar imagens e oferecer informações sobre pragas e métodos de controle. Testes validaram a eficácia da API, representando uma contribuição significativa para o avanço na detecção de doenças em plantações de tomate.

Palavras-chave: *solanum lycopersicum*; detecção de doenças; aprendizado profundo.

ABSTRACT

Tomatoes (*Solanum lycopersicum*) are a rich source of essential nutrients, such as vitamin C and lycopene, playing a crucial role in the population's diet and local economy. This study arises from the need to address specific challenges in the cultivation of these fruits, focusing on the identification of foliar diseases through deep learning models. Utilizing a database with 11,000 images of tomato leaves, various classes of pathologies were explored, including Bacterial Spot, Black Spot, Late Blight, Leaf Mold, Septoria Leaf Spot, Two-Spotted Spider Mite, Target Spot, Yellow Leaf Curl, Mosaic Virus, along with healthy samples. Three models were trained: ResNet50, MobileNet V2, and Inception V3. The results reveal that Inception V3 stands out, achieving an accuracy of 98%, followed by ResNet50 with 96%, and MobileNet V2 with 95%. These findings emphasize the consistency of Inception V3 in classifying tomato leaf images. From the best model, an API was developed using the Django Rest Framework, capable of processing images and providing information about pests and control methods. Tests validated the effectiveness of the API, representing a significant contribution to the advancement in detecting diseases in tomato plantations.

Keywords: *Solanum lycopersicum*; disease detection; deep learning.

1 INTRODUÇÃO

O tomate tem origem na América do Sul, mas é amplamente empregado como um vegetal em variadas tradições culinárias globais. Segundo Santos et al. (2023), o cultivo desempenha um papel fundamental na subsistência de diversas famílias em todo o Brasil, especialmente na região Nordeste. Para pequenos agricultores, é uma fonte significativa de sustento e sua importância econômica motiva o plantio em grandes latifúndios.

Durante o processo de cultivo, os agricultores preparam o solo, realizam espaçamento adequado e, geralmente, adotam técnicas de semeadura em fileiras. O plantio de sementes ou o transplante de mudas é realizado com acompanhamento próximo para garantir condições ideais de crescimento. A implementação de práticas sustentáveis e o manejo integrado são essenciais para assegurar uma produção eficiente e de qualidade.

Conforme Naika et al. (2006), o tomate, uma das olerícolas mais relevantes e pertencente à família das Solanáceas, teve sua origem na região Andina da América do Sul, embora tenha sido domesticado no México e introduzido na Europa em 1544 pelos navegadores portugueses. De acordo com Brito Júnior (2012), a maior parte da produção de tomate é destinada ao consumo in natura.

Segundo o compêndio de estudos da Rubin (2019), estima-se que a maioria dos agricultores que cultivam o tomate plantam cerca de 1 a 2 hectares, com produção média de 60 toneladas/hectare e com um custo de aproximadamente R\$80.000/hectare. O custo de produção elevado é justificado pela grande exigência em termos de adubação, tratamentos culturais e fitossanitário (VIANNA; CRUZ, 2012).

De acordo com Corrêa (2020), os maiores produtores mundiais da cultura são a China, Índia e Estados Unidos. O Brasil se encontra entre os dez maiores produtores de tomate, com um crescimento acima da média mundial. Para Shijie et al. (2017), esta produção é afetada por uma variedade de pragas e doenças do tomateiro que comprometem seriamente a produção do tomate, tais como: mancha bacteriana, mancha precoce, mancha tardia, mosaico vírus e folhas amarelas e curvadas.

Durmuş, Güneş e Kirci (2017), afirmam que, para prevenir essas doenças e pragas, métodos caros e vários pesticidas são usados na agricultura. O uso generalizado desses métodos químicos prejudicam a saúde das plantas e a saúde humana, além de afetar negativamente o meio ambiente. Para controlar com eficácia as pragas e doenças, é importante identificá-las precisamente.

Para Silva et al. (2022), os métodos diretos de identificação de doenças em folhas envolvem a observação e análise minuciosa das folhas infectadas por profissionais especializados. Embora altamente precisos, esses métodos tradicionais podem ser desafiadores, demorados e invasivos, demandando a coleta de amostras para análise. Uma alternativa que proporciona avaliação ágil, precisa e automatizada envolve a utilização de técnicas computacionais para análise de imagens.

Diante do exposto, no presente trabalho é proposto o desenvolvimento de uma Interface de Programação de Aplicações (API) baseada em Aprendizado Profundo, capaz de analisar imagens de folhas de tomate em busca de sinais de doenças. Utilizando algoritmos avançados de aprendizado profundo, a API identifica padrões específicos associados a diferentes patologias, proporcionando diagnósticos rápidos e precisos.

1.1 OBJETIVOS

O objetivo deste trabalho é propor uma API para classificar doenças e pragas que afetam o cultivo de tomates, analisando imagens das folhas das plantas. A finalidade central é estabelecer um *pipeline* eficaz e confiável para identificar doenças e pragas que afetam o cultivo de tomates a partir da análise fotográfica das folhas das plantas. Isso permitirá que desenvolvedores criem diversos aplicativos que utilizem essa API para auxiliar na identificação de pragas em plantações e aplicar o tratamento adequado.

1.1.1 OBJETIVOS ESPECÍFICOS

1. Adquirir conjuntos de dados contendo imagens que retratam doenças presentes nas folhas de tomate;
2. Compreender e implementar os modelos de redes neurais convolucionais para identificação de doenças;
3. Testar e avaliar o modelo com diferentes conjuntos de dados de imagens, para verificar sua capacidade de generalização;
4. Implementar a API a partir do modelo, com os melhores resultados obtidos no treinamento, para que permita o *upload* de imagens possa classificá-las corretamente.

1.2 ORGANIZAÇÃO DESTE TRABALHO

Este trabalho é composto por 5 capítulos. Neste, é apresentada a introdução e os objetivos. No Capítulo 2, além da revisão de literatura abordando os conceitos fundamentais da pesquisa, são apresentadas as métricas de avaliação utilizadas no estudo e uma análise dos trabalhos relacionados. No Capítulo 3, são apresentados a metodologia do trabalho, a base de dados utilizada, e a arquitetura da API. No Capítulo 4, os resultados provenientes do treinamento da base de dados por três classificadores são apresentados, permitindo uma comparação entre eles. No Capítulo 5, são destacadas as considerações finais do estudo, nas quais são resumidas as conclusões derivadas da análise dos resultados.

2 FUNDAMENTAÇÃO TEÓRICA

Neste Capítulo são introduzidos os conceitos e definições fundamentais sobre doenças foliares do tomate, agricultura de precisão, Redes Neurais Artificiais (RNA) e redes neurais convolucionais. Além disso, são abordadas as métricas de avaliação utilizadas para mensurar o desempenho dos modelos implementados. Também são discutidos os trabalhos relacionados que serviram como base para o desenvolvimento desta pesquisa

2.1 DOENÇAS FOLIARES DO TOMATE

A saúde das folhas desempenha um papel fundamental no desenvolvimento e na produção do tomateiro, sendo essencial compreender as diferentes infecções e seus sintomas para implementar estratégias de manejo e controle. Desde infecções bacterianas, como a Mancha Bacteriana, até vírus, como o Vírus do Mosaico, passando por diversas infecções fúngicas, como Pinta-Preta, Requeima, Mofo das folhas, Septoriose, Mancha Alvo, e até mesmo problemas causados por ácaros, como os Ácaros-rajados, cada doença apresenta características distintas que demandam atenção específica.

Na Tabela 1, é apresentada uma representação visual e descritiva das doenças foliares do tomate, oferecendo uma visão abrangente das características distintas de cada patologia, conforme descrito anteriormente.

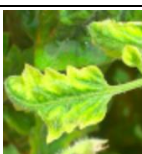
Tabela 1 - Representação das doenças e sua descrição.

Imagem	Nome	Agente causador	Descrição
	Mancha bacteriana	bactérias do gênero <i>Xanthomonas</i>	manchas necróticas, concentrando-se nas bordas das folhas, que evoluem para uma queima da folhagem
	Pinta preta	fungo <i>Guignardia citricarpa</i>	bordas bem definidas, frequentemente com anéis concêntricos mais evidentes nas folhas
	Requeima	oomiceto <i>Phytophthora infestans</i>	lesões grandes, marrom escuras, aparecendo primeiro nas folhas mais novas

Fonte: Adaptada de Lopes (2000)

Na Tabela 2, é apresentada uma continuação do conjunto de informações sobre doenças foliares do tomate, complementando a Representação das doenças e sua descrição fornecida na tabela 1.

Tabela 2 - Representação das doenças e sua descrição

Imagem	Nome	Agente causador	Descrição
	Mofo das folhas	fungo <i>Fulvia fulva</i>	lesões irregulares, verde claras ou amareladas, mais evidentes nas folhas mais velhas
	Septoriose	fungo <i>Septoria lycopersici</i>	manchas arredondadas, marrom escuras com o centro cinza claro
	Ácaros rajados	ácaro rajado (<i>Tetranychus urticae</i>)	folhas inicialmente infestadas tornam-se amareladas na face oposta à colônia; posteriormente, as áreas ficam necrosadas
	Mancha Alvo	fungo <i>Corynespora cassicola</i>	lesões que se iniciam por pontuações pardas, com halo amarelado, evoluindo para grandes manchas circulares, de coloração castanho-clara
	Vírus do mosaico	<i>Tomato mosaic virus</i> (ToMV)	manchas amarelas ou verdes nas folhas, deformação das folhas
	Enrolamento das folhas amarelas	<i>Tomato yellow leaf curl virus</i> (TYLCV)	amarelecimento e enrolamento das folhas

Fonte: Adaptada de Lopes (2000)

É importante ressaltar que, embora as Tabelas 1 e 2 tenham abordado nove doenças foliares específicas do tomate, há uma ampla gama de outras patologias que também podem afetar as plantas de tomateiro. A escolha de focar nessas nove doenças específicas neste trabalho foi estratégica, visando otimizar o treinamento do modelo. Ao concentrar-se em um conjunto

mais restrito, foi possível aprofundar a análise, proporcionar maior precisão na identificação dessas doenças específicas e, conseqüentemente, aprimorar a capacidade preditiva do modelo.

2.2 AGRICULTURA DE PRECISÃO

A Agricultura de Precisão (AP), conforme destacado por Blackmore (1996), representa uma abordagem inovadora no manejo agrícola, buscando aprimorar a eficiência por meio da integração de tecnologias avançadas. Ao modificar e incorporar novas ferramentas, essa prática visa não apenas otimizar a produção, mas também reduzir o desperdício de insumos, alinhando-se com a sustentabilidade e a maximização dos recursos disponíveis. O constante desenvolvimento dessa tecnologia reflete um compromisso contínuo com a melhoria dos métodos agrícolas, impulsionando a agricultura para patamares mais eficientes e sustentáveis.

Conforme as informações do Ministério da Agricultura, Pecuária e Abastecimento (2013), a AP é abordada como um campo multidisciplinar, envolvendo um sistema de manejo integrado de informações e tecnologias de hardware e software. A abordagem é fundamentada na compreensão de que as variabilidades de tempo e espaço exercem influência significativa nos rendimentos das lavouras. A AP visa ao gerenciamento planejado do sistema de produção agrícola como um todo, indo além das simples aplicações de insumos ou mapeamentos, e faz uso de diversas tecnologias, incluindo o Sistema de Posicionamento Global (GPS), Sistema de Informação Geográfica (SIG), bancos de dados, sensores para medidas ou detecção de parâmetros e alvos de interesse no agroecossistema (solo, planta, insetos e doenças), além de geoestatística e mecatrônica.

Segundo a Empresa Brasileira de Pesquisa Agropecuária (2022), o uso de ferramentas adequadas da agricultura de precisão contribui para a diminuição de perdas na agricultura. Por meio de tais ferramentas de agricultura de precisão, é possível obter dados provenientes da análise da propriedade subdividida em pequenas áreas, como informações geográficas georreferenciadas, relativos à irrigação, propriedades físicas do solo e necessidades de aplicação de defensivos. Quanto mais subdividida a propriedade rural, mais útil será a informação georreferenciada. O controle das variáveis que influenciam o cultivo depende do maior detalhamento das informações.

De acordo com o Serviço Nacional de Aprendizagem Rural (2023), o Brasil é hoje um gigante no cenário agrícola mundial. A cada ano a agricultura brasileira vem batendo recordes de produção, com grande representatividade na balança comercial e exportação de produtos agrícolas. Todo esse avanço é graças ao esforço e dedicação diária dos produtores e

trabalhadores rurais e aos avanços tecnológicos que estão sendo disponibilizados nas atividades agropecuárias.

O trabalho de Pinheiro et al. (2021) destaca que, com a chegada da tecnologia ao campo, surge a necessidade premente de obter respostas rápidas e eficientes. Nesse contexto, a capacidade intelectual humana mostra-se insuficiente para atender à velocidade da demanda de um grande volume de dados que requerem alocação adequada na execução das atividades.

Kim et al. (2008) salientam que, diante da elevada demanda por análise de lotes, a interpretação de todas as informações dos testes torna-se inviável para a mente humana. As tecnologias baseadas em inteligência artificial (IA) desempenham um papel fundamental na melhoria da eficiência em diversas áreas de estudo, incluindo o enfrentamento de desafios nas indústrias. Isso abrange setores como o agrícola, onde a IA contribui para otimizar aspectos como rendimento da safra, irrigação, detecção de conteúdo do solo, monitoramento de safra, capina e estabelecimento da cultura.

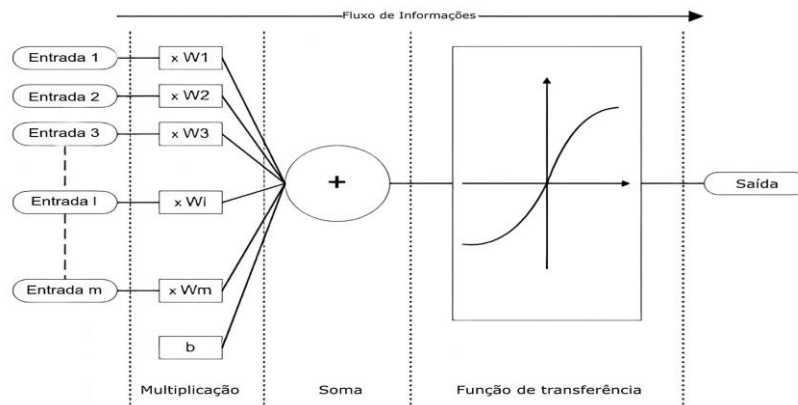
2.3 REDES NEURAIS ARTIFICIAIS (RNA)

De acordo com Haykin (1999), uma Rede Neural pode ser vista como “um processador distribuído massivamente paralelo feito de simples unidades de processamento, as quais possuem uma propensão natural para guardar conhecimento de experiências e tornar isso disponível para uso”. Neste sentido, a Rede Neural Artificial (RNA) se assemelha com o cérebro humano em dois aspectos:

- I. Conhecimento é adquirido pela rede através do processo de aprendizado.
- II. Forças de conexão entre neurônios, conhecidas como pesos sinápticos, são usados para guardar o conhecimento adquirido.

A RNA é considerada como uma das ferramentas inseridas no campo do *Machine Learning* e é considerada como um modelo matemático que tenta simular a estrutura e funcionalidades de redes neurais biológicas (KRENKER et al., 2011). Na Figura 1, é apresentado um esquema matemático de um neurônio artificial.

Figura 1 - Esquema matemático de um Neurônio Artificial.



Fonte: Krenker et al. (2011)

Na Figura 1, é representado o bloco de construção básico de um neurônio artificial, sendo essa composta por um modelo matemático simples, isto é, uma função. O neurônio artificial constitui a unidade elementar, enquanto a rede neural artificial é composta por uma coleção interconectada desses neurônios. Neste modelo, o funcionamento da RNA se baseia em três conjuntos simples de regras. Primeiramente, as entradas da RNA são multiplicadas por pesos, que representam valores numéricos determinando a contribuição de cada entrada para a saída da rede. Em seguida, o somatório das entradas ponderadas é calculado e submetido a uma função de ativação. As funções de ativação são essenciais, pois modificam a forma da saída da rede. Elas podem ser lineares ou não lineares, sendo estas últimas fundamentais para a aprendizagem de relações complexas entre as entradas e saídas da rede neural.

Krenker et al. (2011), afirma que na entrada do neurônio artificial, os *inputs* são ponderados, o que significa que cada *input* é multiplicado por um peso individual. Na seção intermediária do neurônio artificial, existe a função que soma todas as entradas ponderadas e o viés. Na saída do neurônio artificial, a soma dos *inputs* previamente ponderados e do viés passa pela função de ativação, que também é chamada de função de transferência.

O processo de aprendizado em uma RNA é conduzido pelo algoritmo de aprendizado, cuja função é adaptar os pesos sinápticos. Esse ajuste é realizado ao longo de um número finito de iterações, buscando otimizar o desempenho da rede para se aproximar o máximo possível do objetivo estabelecido. Além disso, de acordo com Rezende (2003), considera-se que o processo de aprendizado tem como característica a ocorrência de estímulo da rede pelo meio externo através da apresentação do conjunto de dados.

2.4 REDES NEURAIS CONVOLUCIONAIS (RNC)

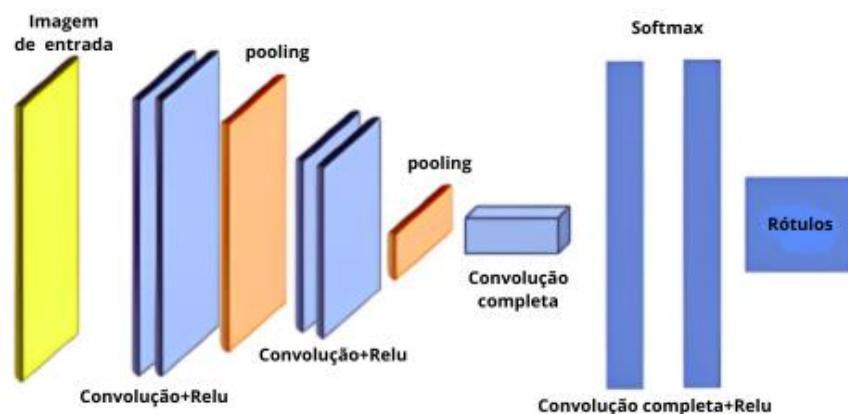
De acordo com Dumoulin e Visin (2016), as Redes Neurais Convolucionais têm desempenhado um papel fundamental nos avanços alcançados no campo do aprendizado profundo. A sua arquitetura é utilizada na aplicação de matrizes bidimensionais, como imagens, por exemplo. A Rede Neural Convolucional (RNC) surgiu na década de 1960, com ideias como percepção local, pesos do compartilhamento e amostragem no espaço ou no tempo. O benefício das RNCs é que são mais fáceis de treinar e têm muito menos parâmetros do que redes totalmente conectadas com a mesma quantidade de unidades ocultas.

A convolução é uma operação linear especializada que é utilizada para extrair características dos dados de entrada aplicando diferentes filtros. Esses filtros são treinados para reconhecer padrões específicos como linhas, curvas ou formas geométricas (GOODFELLOW et al. 2016).

Silva (2023) afirma que outra importante operação executada pelas RNCs são as camadas de *pooling*. Elas são responsáveis pela redução do tamanho dos mapas de recursos através do uso de alguma função para resumir as sub-regiões, como obter a média ou o valor máximo. O *pooling* funciona deslizando uma janela pela entrada e alimentando o conteúdo da janela para uma função de *pooling*. De certa forma, o *pooling* funciona como uma convolução discreta, mas substitui a combinação linear descrita pelo *kernel* por alguma outra função.

Na Figura 2, é ilustrada a arquitetura de uma Rede Neural Convolucional, destacando suas camadas e conexões, fundamentais para a compreensão do processamento de imagens.

Figura 2 - Estrutura de uma Rede Neural Convolucional.



Fonte: Adaptada de AFRIDI et al. (2021).

A camada convolucional, mostrada na figura acima, desempenha um papel importante em várias etapas da RNC. A camada de entrada recebe os dados, geralmente imagens. A camada de convolução aplica filtros para extrair características relevantes da entrada, sendo essencial para a detecção de padrões. A camada de ativação introduz não-linearidades através de funções

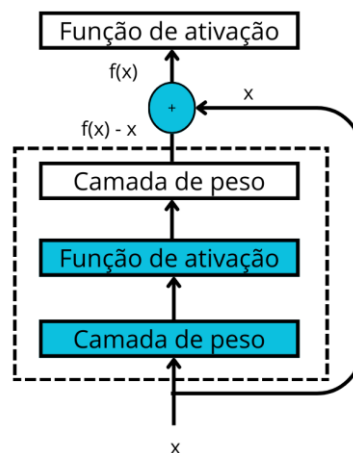
como ReLU. A camada de *pooling* reduz a dimensionalidade espacial, diminuindo a carga computacional. As camadas totalmente conectadas (*Full Convolution*) realizam a classificação final, combinando características para produzir a saída desejada. A convolução completa é um tipo de operação onde zeros, ou outros valores, são adicionados ao redor dos dados de entrada antes de aplicar os filtros. Isso é feito para garantir que o tamanho da saída da convolução seja maior ou igual ao tamanho da entrada. A camada de saída (Rótulos) gera as probabilidades associadas a cada classe.

2.5 RESNET50

A Residual Neural Network (ResNet), proposta por He et al. (2016), faz o uso de conexões de salto para reduzir, de forma significativa, a dificuldade de treinamento, o que resultou em grandes aumentos de desempenho em termos de erro de treinamento e generalização. He et al. (2016) afirma que a ideia básica é pular blocos de camadas convolucionais usando conexões de atalho. Os blocos básicos, chamados de blocos "bottleneck", seguem duas regras de design simples: (i) para o mesmo tamanho de mapa de características de saída, as camadas têm o mesmo número de filtros; e (ii) se o tamanho do mapa de características for reduzido pela metade, o número de filtros é dobrado.

Na Figura 3, é apresentada a arquitetura do bloco residual. Conforme He et al. (2016), essas camadas adicionadas são o mapeamento de identidade e as outras camadas são copiadas do modelo mais raso aprendido. Esta solução indica que um modelo mais profundo não deve produzir um erro de treinamento maior do que sua versão da rede mais rasa .

Figura 3 - Bloco residual



Fonte: Adaptada de He et al. (2016).

Na figura acima, é apresentado um bloco residual, na qual pode ser expresso em código Python usando a expressão $\text{output} = F(x) + x$ onde x é uma entrada para o bloco residual e uma saída da camada anterior, e $F(x)$ é parte de uma RNC que consiste em vários blocos convolucionais.

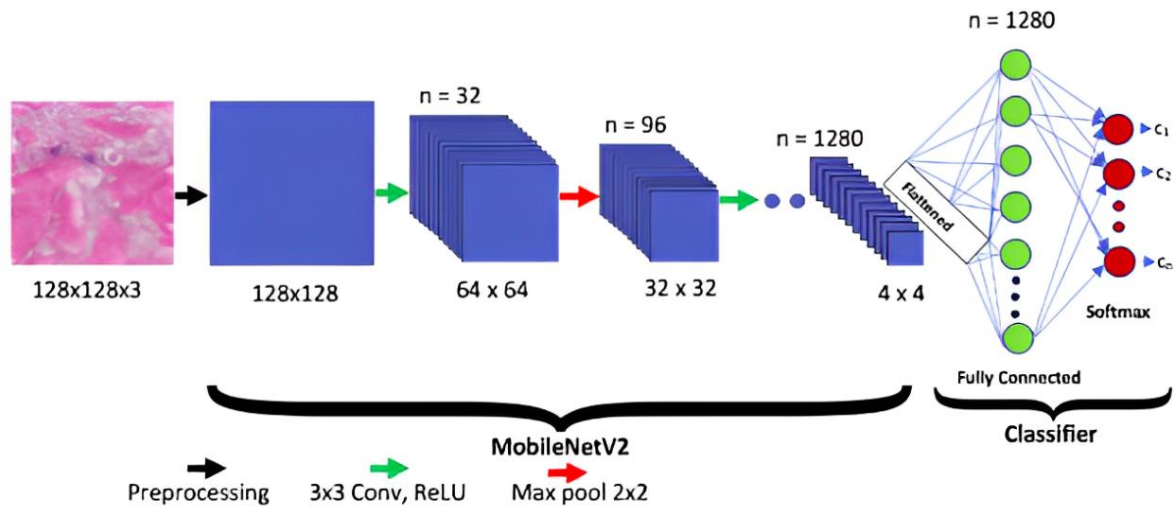
2.6 MOBILENET V2

A MobileNet V2, desenvolvida por Sandler et al. (2018), é uma arquitetura de rede neural desenvolvida para aplicações de visão computacional em dispositivos móveis e sistemas com recursos limitados. Criada como uma evolução do MobileNet original, a versão 2 busca melhorar ainda mais a eficiência computacional e o desempenho em tarefas como classificação de imagens e detecção de objetos.

Sandler et al. (2018) afirma que uma das características notáveis do MobileNet V2 é a incorporação de blocos residuais, inspirados na arquitetura ResNet, que facilitam o treinamento de redes mais profundas sem enfrentar problemas de degradação no desempenho. Além de incluir técnicas como expansão linear e projeção não linear, permitindo um equilíbrio eficiente entre a capacidade de representação e a carga computacional.

Na Figura 4 é apresentada a arquitetura do classificador MobileNet V2. Conforme Akay et al. (2021), ela consiste em dois componentes principais: um caminho de contração (lado esquerdo) e uma unidade classificadora (lado direito). O caminho de contração segue uma estrutura convolucional padrão, utilizando repetidas convoluções 3x3, cada uma seguida por uma unidade linear retificada (ReLU) e uma operação de *pooling* máximo 2x2 para amostragem descendente. Esses blocos são iterados várias vezes, formando um conjunto profundo de camadas totalmente conectadas (classificador estágio). As convoluções aplicam filtros para obter mapas de recursos locais, enquanto as não lineares aumentam as propriedades não lineares desses mapas.

Figura 4 - Arquitetura do classificador MobileNet V2.



Fonte: Adaptado de Akay et al. (2021)

A Figura 4 fornece uma visão esquemática da arquitetura, que é composta por uma série de etapas. A convolução separável em profundidade é empregada para reduzir o custo computacional, fragmentando a convolução padrão em operações distintas, enquanto os resíduos invertidos introduzem uma estrutura de gargalo que amplia a representação de características. O projeto de gargalo, por meio de convoluções 1×1 e os gargalos lineares aprimoram a eficiência do modelo, preservando informações importantes. Além disso, a inclusão de blocos de compressão e excitação (SE) destaca-se por aprimorar adaptativamente as respostas dos recursos do canal, enfocando recursos mais informativos e suprimindo os menos relevantes. Esses componentes trabalham sinergicamente, resultando em uma arquitetura eficiente e eficaz para a classificação de imagens, especialmente em contextos de recursos limitados.

2.7 INCEPTION V3

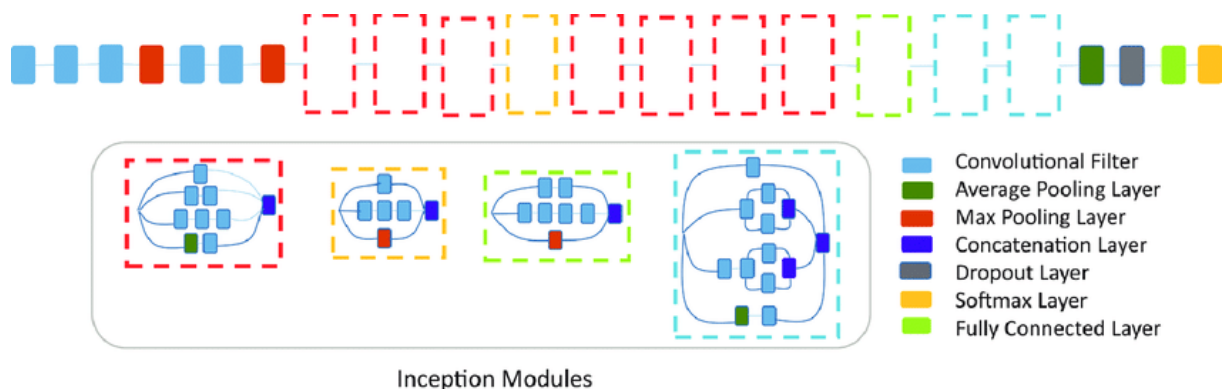
A arquitetura Inception V3, proposta por Szegedy et al. (2016), tem como diferencial a presença de módulos chamados *inception*. Segundo Leonardo e Faria (2019), estes módulos servem como extratores de características convolucionais, visando aprender, com um número reduzido de parâmetros, representações informativas.

Segundo Google (2023), o conceito central da arquitetura é a utilização de diferentes tamanhos de filtros de convolução (1×1 , 3×3 , 5×5) em paralelo para capturar informações em

várias escalas. Esses filtros são aplicados em uma única camada de entrada, e suas saídas são concatenadas, possibilitando que a rede capture tanto características de nível elevado quanto detalhes mais sutis.

Na Figura 5 a seguir, é apresentada a estrutura do Inception V3. Google (2023) afirma que as camadas iniciais realizam convoluções 3x3 e 1x1 com diferentes passos (*strides*) para extrair características em escalas variadas. Em seguida, o modelo incorpora módulos Inception, nos quais convoluções de diferentes tamanhos são empregadas simultaneamente para capturar informações em diversas resoluções espaciais. Intercaladas entre essas convoluções, estão operações de *pooling* e convoluções 1x1 para preservar a dimensionalidade. A estrutura é enriquecida com blocos de atenção '*squeeze-and-excitation*', realçando características importantes. Conexões residuais facilitam o treinamento profundo e a rede culmina com uma camada global de *pooling* e uma camada totalmente conectada, produzindo a saída final para tarefas como classificação ou detecção de objetos. Essa arquitetura destaca-se por sua capacidade de capturar representações ricas e hierárquicas de imagens complexas.

Figura 5 - Arquitetura do classificador Inception V3.



Fonte: Adaptado de Athyrson e Francisco (2019).

As várias camadas do Inception V3 foram projetadas para extrair e hierarquizar informações em diferentes escalas, resultando em uma arquitetura eficiente para tarefas de visão computacional. A combinação de convoluções diversificadas, módulos Inception, atenção "squeeze-and-excitation" e conexões residuais confere ao Inception V3 a capacidade única de capturar características complexas, o que o torna uma escolha valiosa para uma variedade de aplicações de aprendizado profundo.

2.8 MÉTRICAS DE AVALIAÇÃO

No contexto do aprendizado de máquina, é essencial que os modelos tenham a capacidade de fazer previsões consistentes em conjuntos de dados que não foram previamente

observados. Neste documento são utilizada a matriz de confusão apresentada na Tabela 3, composta por:

VP (Verdadeiro Positivo) - Refere-se aos casos em que o modelo identifica corretamente um exemplo como positivo.

VN (Verdadeiro Negativo) - Refere-se aos casos em que o modelo identifica corretamente um exemplo como negativo.

FP (Falso Positivo) - Refere-se aos casos em que o modelo identifica erroneamente um exemplo negativo como positivo.

FN (Falso Negativo) - Refere-se aos casos em que o modelo identifica erroneamente um exemplo positivo como negativo.

E também métricas como: acurácia, recall, F1-score, as quais serão esclarecidas a seguir.

Tabela 3 - Matriz de confusão.

		Valor predito	
		Sim	Não
Real	Sim	VP	FN
	Não	FP	VN

Fonte: Autor (2024)

A matriz de confusão é uma tabela que descreve o desempenho de um modelo de classificação em relação aos exemplos de um conjunto de dados. Ela é particularmente útil para entender como o modelo está acertando ou errando em diferentes classes.

A análise da matriz de confusão fornece uma visão detalhada do desempenho do modelo em diferentes cenários, ajudando na identificação de áreas de melhoria e na compreensão de como o modelo lida com cada classe.

A acurácia, dada pela equação 2.1, é obtida dividindo-se o número de previsões corretas pelo número total de previsões. Esta métrica oferece uma visão geral da precisão do modelo, fornecendo uma medida simples e direta de seu desempenho em termos de acertos em relação ao universo de previsões efetuadas.

$$\text{acc} = \frac{VP + VN}{VP + VN + FP + FN} \quad (2.1)$$

A precisão, dada pela equação 2.2, é uma métrica de avaliação que mensura a proporção de instâncias corretamente classificadas como positivas em relação ao total de instâncias

previstas como positivas. Em outras palavras, ela indica a capacidade do modelo em evitar classificações incorretas de exemplos negativos como positivos.

$$\text{prec} = \frac{VP}{VP + FP} \quad (2.2)$$

Recall ou Sensibilidade, dada pela equação 2.3, representa a proporção de exemplos corretamente identificados de uma classe específica em relação ao total dessa classe no conjunto de dados. Essencial em situações onde a identificação de todos os casos positivos é prioritária, mesmo que isso leve a mais falsos positivos, o recall destaca a capacidade do modelo em capturar efetivamente todos os exemplos relevantes

$$\text{rec} = \frac{VP}{VP + FN} \quad (2.3)$$

O F1-score, dado pela equação 2.4, é uma métrica de avaliação que combina a precisão e o recall, proporcionando uma medida do desempenho geral de um modelo. Sua fórmula calcula a média harmônica entre precisão e recall, destacando a importância de ambas as métricas. O F1-score é particularmente sensível à métrica que tem o valor mais baixo, refletindo um equilíbrio entre a capacidade do modelo em realizar classificações precisas (precisão) e identificar corretamente todos os exemplos positivos (recall).

$$F1 = 2 * \frac{\text{prec} * \text{rec}}{\text{prec} + \text{rec}} \quad (2.4)$$

Como mencionado anteriormente, além da matriz da confusão são utilizadas quatro métricas de avaliação. Neste contexto, essas métricas são empregadas com o propósito de selecionar o melhor classificador entre os utilizados neste trabalho, possibilitando a tomada de decisões mais fundamentadas nos procedimentos de detecção de doenças nas folhas do tomateiro.

2.9 TRABALHOS RELACIONADOS

É fundamental integrar trabalhos relacionados à temática mencionada para garantir uma base sólida de conhecimento específico no campo do desenvolvimento de software e das mais recentes inovações relacionadas a este projeto. Esses estudos apontam para a aplicação conjunta de técnicas de visão computacional e processamento de imagens, demonstrando que essa abordagem pode oferecer soluções precisas e eficientes na identificação de doenças em plantas. Abaixo estão apresentados três trabalhos com uma abordagem similar e que serviram como base para este.

Silva et. al. (2022), no trabalho intitulado “Detecção de Patologias em Folhas de Tomateiros Utilizando Dois Classificadores”, propuseram uma abordagem baseada na utilização de Redes Neurais Convolucionais (RNCs) para discriminar folhas saudáveis de folhas afetadas por patologias. A base de dados utilizada neste trabalho foi a PlantVillage, amplamente utilizada na identificação e classificação de doenças que acometem diversos tipos de plantas. A base contém exclusivamente imagens de folhas saudáveis e afetadas por várias doenças. O pré-processamento das imagens incluiu correção de gama e aplicação de um filtro médio com tamanho de *kernel* de 5 x 5. As imagens foram segmentadas usando uma técnica de análise de cores baseada na conversão das imagens para o espaço de cores Matiz, Valor, Saturação (HSV) e na definição de limites entre as cores do fundo e das folhas. Foram utilizados dois classificadores, onde um se concentrou na forma das folhas, enquanto o outro considerou a textura e o aspecto da superfície das folhas. Após o treinamento, os dois classificadores foram utilizados de forma conjunta na classificação do conjunto de testes, permitindo uma análise abrangente das amostras. Além da classificação baseada na utilização dos dois classificadores, também foi realizada uma classificação utilizando a ferramenta *Teachable Machine*, gerando outros dois modelos para a classificação do conjunto de testes. A acurácia alcançada com a utilização de um único classificador baseado na arquitetura proposta pela ferramenta *Teachable Machine* foi de 97,43%, enquanto a sensibilidade atingiu 98,21%. A abordagem com dois classificadores baseados em redes neurais convolucionais (RNCs) apresentou uma acurácia de 91,63% e sensibilidade de 94,00%.

Viana (2023), em seu trabalho intitulado, “Diagnóstico de doenças em tomateiros com Deep Learning”, um modelo de redes neurais convolucionais com aprendizado por transferência, juntamente com um sistema de pré-processamento de imagens que mostra-se promissor na detecção precoce de doenças foliares do tomate. As abordagens propostas são validadas por meio de experimentos com conjuntos de dados de imagens PlantVillage e PlantDoc, que contêm amostras de folhas de tomate com e sem doenças. Foi apresentada duas abordagens para diagnosticar doenças: um modelo RNC baseado em *Transfer Learning* (TL) usando a arquitetura MobileNet V2 e a aplicação da mesma arquitetura em conjunto com pré-processamento de imagens que combina técnicas como suavização, segmentação de cores e segmentação adaptativa. O modelo de arquitetura MobileNet V2, atinge uma acurácia de 98,56 % quando testado com o conjunto de dados PlantVillage e 26,72 % para o conjunto PlantDoc. O desempenho da proposta com pré-processamento de dados alcança acurácia de 97,21 % no conjunto PlantVillage e 42,31 % no PlantDoc.

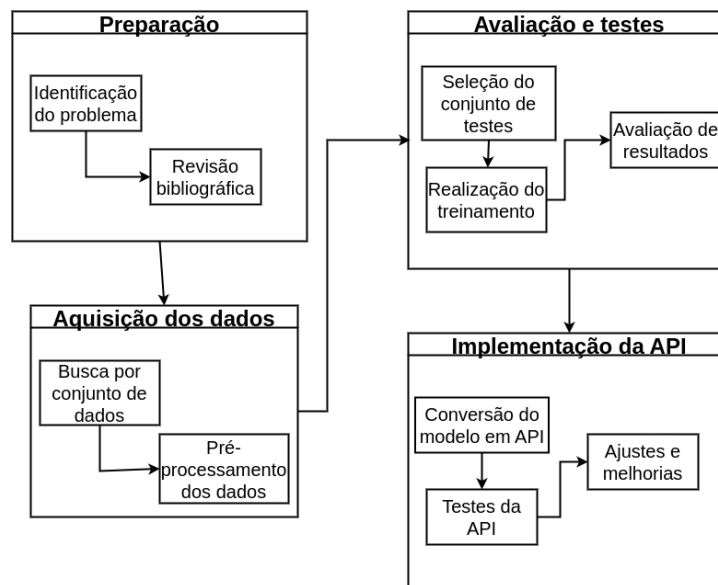
No trabalho de Crispi (2022), intitulado “Uso de redes neurais profundas na avaliação do ataque de mosca minadora ao tomateiro”, foi desenvolvido um modelo de inteligência artificial para detecção e estimativa automática da severidade dos sintomas do ataque da mosca-minadora em folhas do tomate. O conjunto de dados utilizados reuniu 1.932 imagens capturadas nas condições de campo contendo uma folha com o sintoma da praga em evidência. Foi abordada a segmentação semântica multiclasse, um processo onde se dividem imagens em diferentes regiões ou classes. Três arquiteturas de redes neurais convolucionais foram escolhidas para esse propósito, todas baseadas em redes neurais totalmente convolucionais: U-Net, FPN (Feature Pyramid Network) e LinkNet. Os três modelos utilizaram a técnica Transfer Learning, onde aproveitou-se dos pesos pré-treinados de espinhas dorsais (*backbones*) de outras arquiteturas bem estabelecidas, como VGG 16, ResNet 34, Inception V3 e DenseNet 121. No melhor cenário, a aplicação do modelo FNP resultou em índices de acurácia notáveis para diferentes arquiteturas de redes neurais convolucionais. Em particular, a VGG 16 alcançou uma acurácia de 89,27%, a ResNet 34 demonstrou um desempenho de 90,53%, a Inception V3 atingiu 91,10%, e a DenseNet 121 destacou-se com uma acurácia de 91,56%.

3 METODOLOGIA

Para alcançar os objetivos deste trabalho, foi utilizada uma metodologia estruturada em quatro etapas distintas: preparação, aquisição dos dados, implementação do modelo e implementação da API. Foi empregado um método de pesquisa exploratória para investigar e explorar tecnologias similares, visando alcançar os objetivos estabelecidos.

Na Figura 6, é apresentada a estrutura da metodologia aplicada para implementar a API destinada à classificação de doenças em plantações de tomate. Na fase inicial, a atenção foi direcionada para a preparação, que compreende a identificação do problema e a revisão bibliográfica. A identificação do problema envolve uma análise das necessidades relacionadas à classificação de doenças e pragas no cultivo de tomates. Paralelamente, a revisão bibliográfica foi conduzida para aprofundar o entendimento das técnicas de classificação de imagens, redes neurais convolucionais e, especificamente, das enfermidades que afetam as folhas de tomate.

Figura 6 - Pipeline da metodologia



Fonte: Autor (2024)

Na etapa subsequente, voltada à aquisição de dados, iniciou-se a busca por conjuntos de dados. Este processo envolveu a identificação e seleção de conjuntos de dados que contenham imagens representativas das doenças que afetam as folhas de tomate. Também foi realizado o pré-processamento do conjunto fornecido, onde técnicas como reescala, inclinação, zoom e inversão horizontal foram aplicadas. Vale ressaltar que o conjunto de dados já possuía um formato padronizado, compartilhando o mesmo plano de fundo, o que proporciona uma uniformidade visual nas imagens. Essas transformações adicionais, aplicadas de maneira uniforme, contribuem para a diversificação do conjunto.

Em seguida, realizou-se a implementação das RNCs, iniciando pelo treinamento do modelo utilizando 7000 imagens do conjunto e os classificadores ResNet50, MobileNet V2 e Inception V3. Inicialmente, os classificadores são carregados com pesos pré-treinados no conjunto de dados ImageNet. Em seguida, um modelo sequencial é construído, incluindo o classificador como camada base, seguida por camadas densas para classificação. Os modelos são compilados com o otimizador Adam, função de perda de entropia cruzada categórica e métrica de acurácia. Dois *callbacks* são implementados para monitorar o desempenho do modelo durante o treinamento, um é responsável por salvar os melhores pesos do modelo durante cada época, com base no valor da acurácia. E o outro permite a parada antecipada se não houver melhorias significativas na métrica de acurácia de validação. O treinamento é conduzido por 500 épocas, com a opção de interrupção antecipada para evitar *overfitting*. No teste foram selecionadas 1000 imagens do conjunto, não utilizadas durante o treinamento, para avaliar a capacidade de generalização do modelo. Na avaliação de desempenho, métricas como acurácia, precisão, recall, F1-score e matriz de confusão, foram utilizadas para realizar uma análise crítica da eficácia do modelo. Essa avaliação permitirá identificar as capacidades e limitações

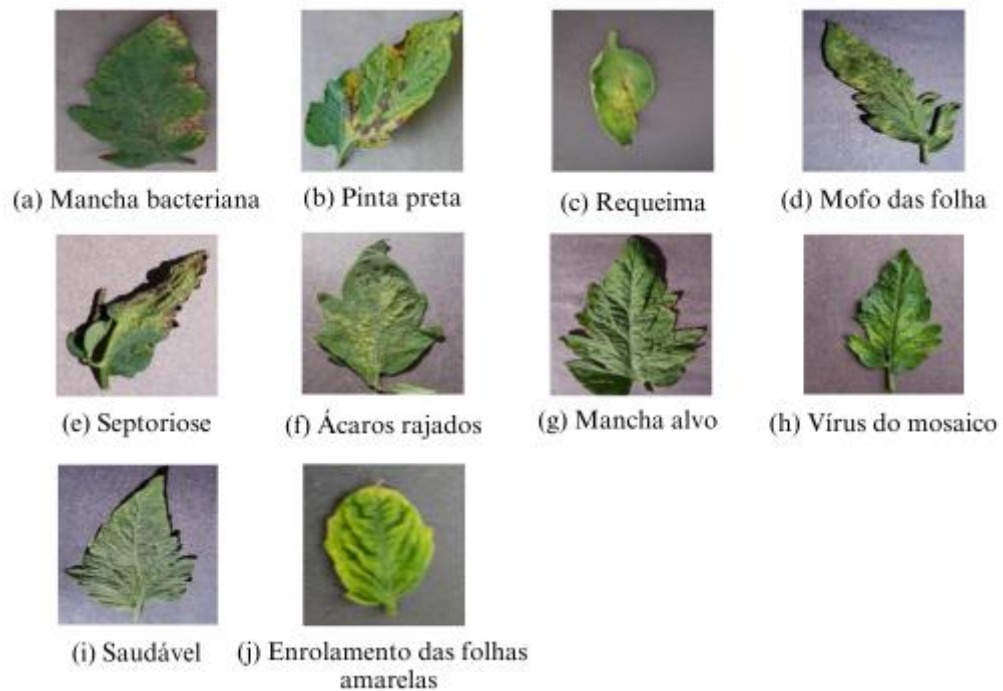
Por fim, no processo de implementação da API, foi realizada a conversão do modelo treinado em uma API funcional. Isso implica na criação de *endpoints* que permitam a entrada de imagens para classificação, além de fornecer informações sobre as pragas e formas de combatê-las. Também foram conduzidos testes para validar a funcionalidade e eficácia da API. Diversos conjuntos de imagens representativas foram submetidos à API, avaliando sua capacidade de classificar doenças nas folhas de tomate. Com base nos resultados dos testes, foram realizados ajustes e melhorias. Isso envolveu otimizações para melhorar o desempenho, ajustes de parâmetros ou incorporação de *feedback* para lidar com casos específicos não considerados anteriormente.

No processo de definir a arquitetura, treinar o modelo e desenvolver a API, adotou-se uma abordagem qualitativa. Cada tecnologia escolhida para o desenvolvimento foi selecionada com base em sua adequação específica para esta aplicação.

3.1 BASE DE DADOS

O banco de dados de imagens deste estudo foi composto por 11.000 imagens de folhas de tomateiro, disponibilizadas na plataforma Kaggle, com dimensões de 256 x 256 pixels. A base é dividida em 10 classes que contém imagens de folhas com algum tipo de patologia ou folhas saudáveis, como é apresentado na Figura 7.

Figura 7 - Amostra de imagens da base de dados.



Fonte: Adaptada de Kaggle (2023)

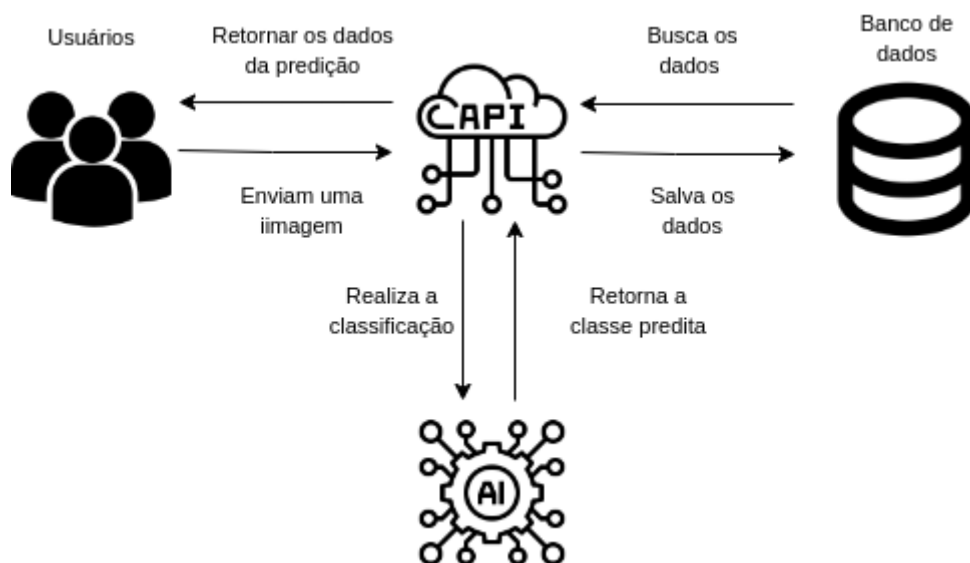
As imagens de cada categoria seguem um formato padronizado, compartilhando o mesmo plano de fundo, orientação e quantidade específica de folhas exibidas. No contexto deste estudo, é relevante destacar que o conjunto completo de imagens foi empregado de maneira integral em todas as fases do processo, incluindo treinamento, validação e teste. Essa abordagem assegura uma avaliação mais precisa da capacidade do modelo de generalizar para novos dados, além de tornar as comparações entre diferentes modelos mais consistentes e confiáveis.

3.2 ARQUITETURA

Ao delinear a arquitetura, observou-se que a estrutura mais adequada para atender às demandas do projeto é a apresentada na Figura 8. Nessa configuração, uma aplicação cliente assume a responsabilidade de consumir as funcionalidades disponibilizadas por meio de uma API. Essa abordagem proporciona uma organização eficiente e alinhada com os requisitos do projeto, estabelecendo clareza na distribuição de responsabilidades entre a aplicação cliente e a

API, otimizando assim a implementação das funcionalidades necessárias. Essa API, por sua vez, utiliza um modelo de *machine learning* para realizar a classificação de imagens de folhas de tomateiro, identificando uma doença específica. Ao receber informações do cliente, a API processa as imagens por meio do modelo de *machine learning* designado e, em seguida, consulta um banco de dados que armazena registros históricos de diagnósticos, imagens associadas e informações relevantes sobre as doenças identificadas. O banco de dados existe para fornecer uma fonte centralizada de dados, permitindo o armazenamento eficiente e o acesso rápido a informações sobre diagnósticos anteriores, contribuindo para a melhoria contínua do modelo de *machine learning* e oferecendo um histórico completo das análises realizadas. O servidor, encarregado de gerenciar a API, retorna as informações processadas ao cliente. Para concluir o fluxo, a aplicação cliente apresenta as informações processadas, originárias da resposta fornecida pela API. A arquitetura delineada pode ser visualizada na representação da Figura 8 apresentada a seguir.

Figura 8 - Arquitetura do sistema.



Fonte: Autor (2024)

Em suma, a arquitetura apresentada estabelece uma abordagem eficaz para atender às demandas do projeto. Com a divisão de responsabilidades entre a API de *machine learning* e o servidor de gerenciamento, o sistema está organizado para realizar a classificação de imagens de tomateiro, identificando possíveis doenças.

3.3 TECNOLOGIAS UTILIZADAS

Nesta seção, são descritas as tecnologias empregadas, abrangendo configurações e outras características pertinentes ao processo de desenvolvimento.

3.3.1 PYTHON

Conforme documentado na Python Software Foundation (2023), a linguagem oferece uma sintaxe clara e expressiva, permitindo aos desenvolvedores concentrarem-se mais na lógica do problema em questão do que na complexidade da programação. Além disso, a extensa documentação oficial fornece recursos detalhados que facilitam a compreensão e implementação de bibliotecas científicas, como NumPy, SciPy e Pandas, essenciais para análise de dados e modelagem estatística. A capacidade de integrar facilmente módulos de terceiros, juntamente com uma comunidade ativa de desenvolvedores, contribui para a rápida resolução de problemas e atualizações frequentes.

3.3.2 TENSORFLOW

O TensorFlow é uma biblioteca de código aberto desenvolvida pela equipe do Google Brain. De acordo com a ABADI et al. (2016), a biblioteca é projetada para oferecer flexibilidade e eficiência, sendo amplamente utilizada tanto em ambientes acadêmicos quanto industriais. Sua arquitetura modular permite a construção e treinamento de modelos complexos em uma variedade de domínios, desde visão computacional e processamento de linguagem natural. Ela oferece suporte a uma variedade de linguagens de programação, incluindo Python e C++, facilitando a integração em diferentes ecossistemas de desenvolvimento.

3.3.3 DJANGO REST FRAMEWORK (DRF)

O Django Rest Framework (DRF) representa uma poderosa extensão do *framework* web Django, projetado especificamente para facilitar o desenvolvimento de APIs RESTful em aplicações Python. De acordo com DJANGO Software Foundation, este *framework* oferece uma abordagem pragmática e eficiente para a criação de interfaces de programação de aplicativos robustas, permitindo aos desenvolvedores implementar operações Criação, Leitura, Atualizar, Excluir (CRUD) de forma intuitiva e escalável. O DRF integra-se harmoniosamente com os conceitos fundamentais do Django, como modelos, *views* e Localizador Padrão de Recursos (URLs), proporcionando uma estrutura coerente para a exposição de recursos através

de *endpoints* RESTful. Além disso, o DRF oferece funcionalidades avançadas, como autenticação, autorização, serialização e suporte a formatos de dados variados, incluindo Notação de Objeto em Javascript (JSON) e Linguagem de Marcação Extensível (XML).

4 RESULTADOS E DISCUSSÕES

Neste capítulo, são apresentados os desempenhos obtidos por cada modelo, acompanhados de discussões detalhadas sobre as considerações que influenciaram a escolha do modelo a ser implementado na API. Além disso, é realizada uma análise das métricas, destacando as vantagens e limitações de cada modelo.

Ao analisar os resultados de acurácia para os modelos ResNet50, MobileNet V2 e Inception V3, é possível identificar padrões interessantes, como é apresentado na Tabela 4. A ResNet50 demonstrou um desempenho notável durante o treinamento, alcançando uma acurácia média de 100 % com um desvio padrão de 0.06. Este resultado sugere uma convergência estável do modelo durante o treinamento. No entanto, ao avaliar o desempenho na validação, observa-se uma pequena redução para uma média de 96 %, indicando uma possível leve sobreajuste.

Tabela 4 - Acurácia média e desvio padrão das arquiteturas de RNC implementadas.

Arquitetura	Acurácia média (%) / Desvio Padrão	
	Treinamento	Validação
ResNet50	100 ± 0,06	96 ± 0,18
MobileNet V2	99 ± 0,08	95 ± 0,21
Inception V3	99 ± 0,08	98 ± 0,14

Fonte: Autor (2024)

O MobileNet V2 apresentou resultados igualmente notáveis, com uma acurácia média de 99 % durante o treinamento e um desvio padrão de 0,08. Embora o desvio padrão seja ligeiramente maior em comparação com a ResNet50, ainda é uma indicação de estabilidade no treinamento. Na fase de validação, a acurácia média foi de 95 %, com um desvio padrão de 0,21. Isso sugere uma sensibilidade maior do modelo a variações nos dados de validação.

A Inception V3 também demonstrou um bom desempenho durante o treinamento, atingindo uma acurácia média de 99 % com um desvio padrão de 0,08. Na validação, a acurácia média foi de 98 %, indicando uma melhor capacidade de generalização. O desvio padrão de 0,14 na fase de validação, embora ligeiramente mais alto do que durante o treinamento, sugere uma boa estabilidade.

9	0	0	0	0	4	0	0	0	996	0	0	0	0	0	19	0	0	0	81	0
10	0	0	0	0	1	0	0	0	0	999	0	0	0	0	3	0	0	0	0	97

Fonte: Autor (2024)

A seguir, na Tabela 6, são apresentados os demais dados sobre a matriz de confusão apresentados na Tabela 5.

Tabela 6 - Resultado das métricas de avaliação do classificador ResNet50.

Classe	Treinamento				Validação			
	Acurácia	Precisão	Recall	F1	Acurácia	Precisão	Recall	F1
1	1,00	0,99	1,00	0,99	0,96	0,98	1,00	0,99
2		1,00	0,99	1,00		1,00	1,00	1,00
3		1,00	0,99	1,00		1,00	1,00	1,00
4		1,00	1,00	1,00		1,00	0,97	0,98
5		0,99	1,00	0,99		0,77	1,00	0,87
6		1,00	0,99	1,00		0,99	0,98	0,98
7		0,99	0,99	0,99		0,99	0,94	0,96
8		1,00	0,99	1,00		1,00	0,98	0,99
9		1,00	1,00	1,00		0,99	0,81	0,89
10		1,00	1,00	1,00		1,00	0,97	0,98

Fonte: Autor (2024)

Na Tabela 6, é apresentado um relatório das métricas de avaliação em cada classe do classificador ResNet50. No conjunto de treinamento, as classes apresentaram, em sua maioria, altas precisões, indicando que o modelo teve sucesso em classificar corretamente a maioria das instâncias. No entanto, na validação, a classe 5, isto é, ao problema Septoriose, obteve uma precisão significativamente menor, sugerindo que o modelo pode ter dificuldades em discernir essa classe durante a avaliação. Ao analisar o recall no conjunto de treinamento, foram obtidos valores elevados, indicando que o modelo foi capaz de recuperar a maioria das instâncias positivas para cada classe. Na validação, no entanto, algumas classes, como a 9, que representa Vírus do Mosaico, tiveram valores mais baixos, sugerindo que o modelo pode ter dificuldades em identificar adequadamente essas instâncias durante a avaliação. As pontuações F1-score no treinamento são consistentemente altas, indicando um bom equilíbrio entre precisão e recall para a maioria das classes. Na validação, as classes 5, 9 e 10, que representa, Septoriose, Vírus do Mosaico e Saudável, de modo respectivo, mostraram pontuações F1 mais baixas.

4.2 RESULTADOS OBTIDOS PELO MOBILENET V2

Na Tabela 7, é apresentada a matriz de confusão gerada a partir do treinamento e validação utilizando o classificador MobileNet V2.

Tabela 7 - Matriz de confusão do classificador MobileNet V2.

Predito	Treinamento										Validação									
Real	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
1	996	0	3	0	1	0	0	0	0	0	99	0	0	0	0	0	0	1	0	0
2	6	981	2	0	9	0	2	0	0	0	0	100	0	0	0	0	0	0	0	0
3	0	1	997	0	1	0	0	0	0	1	0	2	98	0	0	0	0	0	0	0
4	0	1	0	998	0	1	0	0	0	0	0	1	1	97	1	0	0	0	0	0
5	0	1	0	0	999	0	0	0	0	0	0	1	0	0	99	0	0	0	0	0
6	0	1	0	0	0	998	0	0	0	1	0	3	0	2	2	93	0	0	0	0
7	0	2	0	0	7	2	977	0	0	12	0	3	0	1	6	0	87	0	0	3

8	0	0	0	0	0	2	0	998	0	0	1	0	0	0	0	0	99	0	0
9	0	3	0	0	9	1	2	0	985	0	0	3	0	0	12	0	0	85	0
10	0	0	0	0	0	0	0	0	0	1000	0	0	0	0	2	0	0	0	98

Fonte: Autor (2024)

A seguir, na Tabela 8, são apresentados os demais dados sobre a matriz de confusão apresentados na tabela 7.

Tabela 8 - Resultado das métricas de avaliação do classificador MobileNet V2.

Classe	Treinamento				Validação			
	Acurácia	Precisão	Recall	F1	Acurácia	Precisão	Recall	F1
1	0,99	0,99	1,00	1,00	0,95	0,99	0,99	0,99
2		0,99	0,98	0,99		0,88	1,00	0,94
3		1,00	1,00	1,00		0,99	0,98	0,98
4		1,00	1,00	1,00		0,97	0,97	0,97
5		0,97	1,00	0,99		0,81	0,99	0,89
6		0,99	1,00	1,00		1,00	0,93	0,96
7		1,00	0,98	0,99		1,00	0,87	0,93
8		1,00	1,00	1,00		0,99	0,99	0,99
9		1,00	0,98	0,99		1,00	0,85	0,92
10		0,99	1,00	0,99		0,97	0,98	0,98

Fonte: Autor (2024)

Na tabela 8, é apresentado o relatório do classificador MobileNet V2. A precisão mostra valores consistentemente elevados no treinamento, indicando que o modelo foi capaz de realizar classificações precisas para a maioria das classes. Entretanto, na validação, a classe 2 e a classe 5 que representam, respectivamente, Pinta Preta e Septoriose, apresentam precisões mais

baixas. O recall mostra resultados satisfatórios tanto no treinamento quanto na validação. As classes 6 e 7 que representam, respectivamente, Ácaro-Aranha Bicolor e Ponto Alvo, apresentam recalls ligeiramente inferiores na validação, indicando uma possível área de melhoria para o modelo. O F1-score revela um equilíbrio satisfatório no treinamento, mas novamente evidencia desafios nas classes 2, 5 e 9, representados por Pinta Preta, Septoriose e Vírus do Mosaico, durante a validação. Essas classes podem ser particularmente sensíveis a falsos positivos ou negativos.

4.3 RESULTADOS OBTIDOS PELO INCEPTION V3

Na Tabela 9, é apresentada a matriz de confusão gerada a partir do treinamento e validação utilizando o classificador Inception V3.

Tabela 9 - Matriz de confusão do classificador Inception V3.

Predito	Treinamento										Validação									
Real	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
1	998	1	0	1	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
2	1	990	8	0	1	0	0	0	0	0	1	99	0	0	0	0	0	0	0	0
3	0	2	997	0	0	0	0	0	0	1	0	0	100	0	0	0	0	0	0	0
4	0	6	0	994	0	0	0	0	0	0	0	2	0	98	0	0	0	0	0	0
5	0	2	5	0	991	1	1	0	0	0	0	1	4	0	95	0	0	0	0	0
6	0	0	0	1	0	986	12	0	0	1	0	0	0	2	0	95	1	0	2	0
7	5	4	0	0	1	0	988	0	0	2	4	1	0	0	2	0	93	0	0	0
8	0	0	0	0	0	1	0	998	0	0	1	0	0	0	0	0	0	99	0	0
9	0	0	0	0	0	0	0	0	1000	0	0	0	0	0	0	0	0	0	100	0
10	0	0	0	0	0	0	0	0	0	1000	0	0	0	0	0	0	0	0	0	100

Fonte Autor (2024)

A seguir, na Tabela 10, são apresentados os demais dados sobre a matriz de confusão apresentados na Tabela 9.

Tabela 10 - Resultado das métricas de avaliação do classificador Inception V3.

Classe	Treinamento				Validação			
	Acurácia	Precisão	Recall	F1	Acurácia	Precisão	Recall	F1
1	0,99	0,99	1,00	1,00	0,98	0,94	1,00	0,97
2		0,99	0,99	0,99		0,96	0,99	0,98
3		0,99	1,00	0,99		0,96	1,00	0,98
4		1,00	0,99	1,00		0,98	0,98	0,98
5		1,00	0,99	0,99		0,98	0,95	0,96
6		1,00	0,99	0,99		1,00	0,95	0,97
7		0,99	0,99	0,99		0,99	0,93	0,96
8		1,00	1,00	1,00		1,00	0,99	0,99
9		1,00	1,00	1,00		0,98	1,00	0,99
10		1,00	1,00	1,00		1,00	1,00	1,00

Fonte: Autor (2024)

Na Tabela 10, é apresentado o relatório do classificador Inception V3. A precisão demonstra valores elevados em ambas as fases. Isso sugere que o modelo foi capaz de realizar classificações precisas para a maioria das classes, com apenas algumas discrepâncias notáveis, como a classe 1, isto é, Mancha Bacteriana, na validação, que apresentou uma precisão um pouco mais baixa. O recall também demonstra resultados altos em ambas as fases. Algumas variações notáveis incluem os recalls mais baixos nas classes 5, 6 e 7, que representa, Septoriose, Ácaro-Aranha Bicolor e Ponto Alvo, respectivamente, na validação, indicando uma menor capacidade do modelo em recuperar todas as instâncias positivas dessas classes. A pontuação F1 reflete um equilíbrio relevante no treinamento, com desempenho robusto nas classes avaliadas. No entanto, na validação, algumas classes, como a 5 e a 7, Septoriose e Ponto Alvo, de modo respectivo, apresentam pontuações F1 um pouco mais baixas, indicando desafios específicos nessas categorias.

4.4 DISCUSSÕES

Ao comparar os resultados dos três classificadores foi observado diferentes padrões de desempenho em suas métricas. No geral, os três modelos demonstram um bom desempenho. No entanto, há variações notáveis em seus desempenhos relativos. No geral, o Inception V3 se destaca por apresentar consistentemente altos níveis de precisão, recall e F1-score em ambas as fases de treinamento e validação. Isso sugere que o Inception V3 tem uma melhor capacidade de realizar classificações precisas e identificar corretamente instâncias positivas para as diferentes classes.

Diante dos resultados promissores apresentados pela avaliação dos classificadores, a escolha do Inception V3 destacou-se como a opção ideal para o desenvolvimento de uma API utilizando Django Rest Framework. Além disso, a decisão de utilizar este *framework* oferece benefícios adicionais em termos de desenvolvimento rápido, escalabilidade e a capacidade de criar *endpoints* API de maneira estruturada.

5 CONSIDERAÇÕES FINAIS

Este trabalho propôs e desenvolveu uma API dedicada à detecção de doenças em tomateiros, utilizando modelos de aprendizado profundo para análise de imagens. A implementação da API contempla dois *endpoints*: o primeiro, responsável pelo envio de imagens e pela detecção de doenças, e o segundo, destinado a disponibilizar as imagens processadas. Diversos modelos foram explorados, incluindo ResNet50, MobileNet V2 e Inception V3.

Os resultados indicam que o modelo Inception V3 se destaca, oferecendo uma performance superior em termos de acurácia e capacidade de generalização, obtendo o valor de 98 %, enquanto o ResNet50 alcançou 96 % e o MobileNet V2 95 %. A escolha deste modelo para o desenvolvimento da API, integrada ao Django Rest Framework, foi fundamentada nas métricas de avaliação, destacando-se como a melhor opção para a classificação de imagens de tomateiros.

Por fim, é possível concluir que a aplicação alcançou resultados adequados, atendendo plenamente aos objetivos estabelecidos neste projeto. A pesquisa bibliográfica desempenhou um papel fundamental ao proporcionar um aprofundamento no entendimento das tecnologias empregadas. Essa base teórica não apenas respaldou as decisões durante o desenvolvimento da aplicação, mas também contribuiu para a compreensão do contexto envolvido. O sucesso obtido reforça a eficácia da abordagem adotada, ressaltando a importância da integração entre a prática e a fundamentação teórica para alcançar resultados sólidos e alinhados com os propósitos do projeto.

5.1 SUGESTÕES DE TRABALHOS FUTUROS

Em perspectiva aos trabalhos futuros, uma expansão da base de dados de treinamento surge como um passo crítico para aprimorar ainda mais a eficácia da API. A inclusão de um conjunto mais diversificado de imagens, abrangendo diferentes condições de crescimento, variações climáticas e múltiplos estágios de desenvolvimento do tomateiro, permitirá ao modelo aprender padrões mais abrangentes e generalizáveis.

Outro ponto é a otimização dos parâmetros de treinamento do modelo. A condução de estudos sistemáticos para ajuste e refinamento dos hiperparâmetros, como taxa de aprendizado, tamanho do lote e número de épocas, pode resultar em melhorias significativas no desempenho do modelo.

Também, considera-se a implementação de um aplicativo móvel voltado para auxiliar os agricultores no acesso e utilização da API de detecção de doenças em tomateiros. Este aplicativo poderá oferecer uma interface intuitiva para o envio de imagens diretamente do dispositivo móvel, permitindo aos agricultores realizarem o monitoramento em tempo real de suas plantações. Ademais, poderá incluir funcionalidades adicionais, como a apresentação de relatórios de diagnósticos, alertas sobre condições de saúde das plantas, e até mesmo sugestões de práticas agrícolas com base nos resultados obtidos.

REFERÊNCIAS

- ABADI, M. et al. **TensorFlow**: a system for Large-Scale machine learning. In: 12th USENIX symposium on operating systems design and implementation (OSDI 16), p. 265-283, 2016.
- AFRIDI, T. H., ALAM, A., KHAN, M. N., KHAN, J., & LEE, Y. K. **A multimodal memes classification**: A survey and open research issues. In Innovations in Smart Cities Applications Volume 4: The Proceedings of the 5th International Conference on Smart City Applications pp. 1451-1466, 2021.
- AKAY, M.; DU, Y.; SERSHEN, C.; WU, M.; CHEN, T.; ASSASSI, S.; MOHAN, C.; AKAY, Y. **Deep Learning Classification of Systemic Sclerosis Skin Using the MobileNet V2 Model**. IEEE Open Journal of Engineering in Medicine and Biology, [S.l.], v. 3, n. 1, p. 1-1, 2021. DOI: 10.1109/OJEMB.2021.3066097.
- ATHYRSON M. R.; FRANCISCO DE PAULA S. A. J. **Rede Inception V3 Voltada a Identificação do Glaucoma**: Comparação entre Métodos de Otimização. 2019.
- BLACKMORE, B. S. **An information system for precision farming**. Silsoe. Inglaterra : The Centre for Precision Farming. Cranfield University, 1996. p.09. Disponível em: <<http://www.silsoe.cranfield.ac.uk/cpf/papers/ISPF/ispf3.pdf>>. Trabalho apresentado em: The Brighton Conference Pests and Diseases, 1996. Acesso em 01 Fev. 2024
- BRITO JÚNIOR, F. P. de. **Produção de tomate (Solanum lycopersicum L.) reutilizando substratos sob cultivo protegido no município de Iranduba-AM**. Dissertação (Mestrado em Agronomia/Produção Vegetal) - Universidade Federal do Amazonas, Manaus. 2012.
- RUBIN, C. **Tomate**: Análise dos Indicadores da Produção e Comercialização no Mercado Mundial, Brasileiro e Catarinense. Compêndio de estudos Conab, Brasília, v. 21. 2019.
- CORRÊA, D. **Produção de mudas de tomate com bioestimulante**. Rev. Agronomia Brasileira. Universidade do Estado de Mato Grosso–UNEMAT, 2020.
- CRISPI, G. M. **Uso de redes neurais profundas na avaliação do ataque de mosca minadora ao tomateiro**. Dissertação (Mestrado em Engenharia Agrícola) - Universidade Federal de Viçosa, Viçosa. 2022.
- DJANGO Software Foundation. **Django REST framework**. Versão 3.14.0. Disponível em: <https://www.django-rest-framework.org/>. Acesso em: 6 fev. 2024.
- DUMOULIN, V.; VISIN, F. **A guide to convolution arithmetic for deep learning**. arXiv preprint arXiv:1603.07285, 2016
- DURMUŞ, H.; GÜNEŞ, E. O.; KIRCI, M. **Disease detection on the leaves of the tomato plants by using deep learning**. 6th International Conference on AgroGeoinformatics. p. 1–5, 2017.
- EMPRESA BRASILEIRA DE PESQUISA AGROPECUÁRIA - EMBRAPA. **Agricultura de precisão**. Disponível em: <<https://www.embrapa.br/agencia-de-informacao>>

tecnologica/cultivos/cana/producao/avanco-tecnologico/agricultura-de-precisao>. Brasília, 2022. Acesso em: 02 nov. 2022.

GOODFELLOW, I; BENGIO, Y; COURVILLE, A. **Deep Learning**. Cambridge, MA: MIT Press, 2016. Disponível em: <http://www.deeplearningbook.org>. Acesso em: 12 dez. 2023.

GOOGLE. **Guia avançado do Inception v3**. <<https://cloud.google.com/tpu/docs/inception-v3-advanced?hl=pt-br>>. 2023. Acesso em: 05 dez. 2023.

GOOGLE. **TensorFlow Documentation**. Disponível em: <<https://www.tensorflow.org/>>. Acesso em: 10 jan. 2024.

HAYKIN, S. **Neural networks: a comprehensive foundation**. Upper Saddle River: Prentice Hall, 1999.

HE, K.; ZHANG, X.; REN, S.; SUN, J. **Deep residual learning for image recognition**. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, 2016.

KIM, R. G.; EVANS; W. M. IVERSEN. **Remote sensing and control of an irrigation system using adistributed wireless sensor network**. IEEE Transactions on Instrumentation And Measurement, v. 57, n. 7, p. 1379-1387, 2008.

KRENKER, A.; BESTER, J.; & KOS, A. **Introduction to the Artificial Neural Networks**. InTech. p. 1-18, 2011.

LEONARDO, M. M.; FARIA, F. A. **Um sistema de reconhecimento de espécies de moscas-das-frutas**. In Anais do XXXVIII Concurso de Trabalhos de Iniciação Científica da SBC. 2019.

LOPES, C. A. **Guia de Identificação das Doenças do Tomateiro**. Brasília: Embrapa Hortícolas / ZENECA, 2000.

MINISTÉRIO DA AGRICULTURA, PECUÁRIA E ABASTECIMENTO. **Agricultura de precisão**. Secretaria de Desenvolvimento Agropecuário e Cooperativismo. Brasília, 2013. Disponível em: <http://www.agricultura.gov.br/arq_editor/Boletim%20tecnico.pdf>. Acesso em: 02 nov. 2023.

NAIKA, S.; JEUDE, J. v. L. de; GOFFAU, M. de.; HILMI, M.; DAM, B. v. **A Cultura do tomate: produção, processamento e comercialização**. Wageningen: Fundação Agromisa e CTA, p. 06, 2006.

PINHEIRO, R. M.; GADOTTI, G. I.; MONTEIRO, R. D. C. M.; BERNARDY, R. **Inteligência artificial na agricultura com aplicabilidade no setor sementeiro**. Diversitas Journal, p. 2996-3012, 2021.

PYTHON SOFTWARE FOUNDATION. **Python 3 Documentation**. Disponível em: <<https://docs.python.org/3/>>. Acesso em: 10 jan. 2024.

REZENDE, S. O. **Sistemas inteligentes: fundamentos e aplicações**. Editora Manole Ltda, 2003.

SANDLER, M.; HOWARD, A.; ZHU, M.; ZHMOGINOV, A.; & CHEN, L. C. **MobileNet V2: Inverted residuals and linear bottlenecks**. In Proceedings of the IEEE conference on computer vision and pattern recognition p. 4510-4520, 2018.

SANTOS, J. A. dos; MONTEBELLO, A. E. S.; MARJOTTA-MAISTRO, M. C.; PEDROSO, M. T. M. **Comercialização e análise dos preços no mercado do tomate nas principais CEASAs da Região Sudeste de 2011 a 2021**. In: FÓRUM INTERNACIONAL ON-LINE DE EMPREENDEDORISMO E INOVAÇÃO NO AGRO, 4., 2023. Anais... São Paulo: FZEA/USP, 2023.

SERVIÇO NACIONAL DE APRENDIZAGEM RURAL. **Programa Agricultura de Precisão**. 2023. Disponível em: <<https://cnabrazil.org.br/projetos-e-programas/agricultura-de-precisao>>. Acesso em: 02 nov. 2023.

SHIJIE, J., PEIYI, J., SIPING, H., & HAIBO, S. (2017). **Automatic detection of tomato diseases and pests based on leaf images**. Chinese Automation Congress (CAC). p. 2537–2510, 2017.

SILVA, E. **Adivinhe quem é o maior produtor mundial de tomate**. 2022. Disponível em: <<https://revistagloborural.globo.com/Noticias/Agricultura/Hortifruti/noticia/2022/01/adivinhe-quem-e-o-maior-produtor-mundial-de-tomate.html>>. Acesso em: 01 nov. 2023.

SILVA, L. M. D.; BARROS, L. R.; DOS SANTOS, M. L. F.; REGIS, C. D. M. **Deteção de Patologias em Folhas de Tomateiros Utilizando Dois Classificadores**. XL SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES E PROCESSAMENTO DE SINAIS, 2022.

SILVA, T. H. L. D. **Análise comparativa de redes neurais convolucionais na classificação de tumores pulmonares**. 2023.

SZEGEDY, C.; VANHOUCKE, V.; IOFFE, S.; SHLENS, J.; & WOJNA, Z. **Rethinking the inception architecture for computer vision**. In Proceedings of the IEEE conference on computer vision and pattern recognition p. 2818-2826, 2016.

VIANA, R. F. **Diagnóstico de doenças em tomateiros com Deep Learning**. Monografia (Graduação em Engenharia Elétrica) - Instituto de Ciências Exatas e Aplicadas, Universidade Federal de Ouro Preto, João Monlevade, 2023.

VIANNA, G. K.; CRUZ, S. M. S. da. **Redes neurais artificiais aplicadas ao monitoramento inteligente de doenças e pragas em tomateiros**. 2012.