WESLEY RIBEIRO DE SOUZA

IDENTIFICAÇÃO DE FERRUGEM, CERCOSPORIOSE E BICHO-MINEIRO EM FOLHAS DE CAFEEIRO POR MEIO DE TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL

LAVRAS – MG

2025

WESLEY RIBEIRO DE SOUZA

IDENTIFICAÇÃO DE FERRUGEM, CERCOSPORIOSE E BICHO-MINEIRO EM FOLHAS DE CAFEEIRO POR MEIO DE TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL

Projeto apresentado à Universidade Federal de Lavras, como parte das exigências do Programa de Pós Graduação em Engenharia de Sistemas e Automação, área de Engenharia de Sistemas e Automação, para a obtenção do título de Mestre.

Prof. DSc. Danton Diego Ferreira
Orientador

LAVRAS – MG
2025

Ficha Catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).

Souza, Wesley Ribeiro de .
Identificação de ferrugem, cercospora e bicho-mineiro em folhas de cafeeiro por meio de técnicas de inteligência artificial / Wesley Ribeiro de Souza. - 2025.
77 p. : il.

Orientador: Danton Diego Ferreira

Dissertação (Mestrado Acadêmico) - Universidade Federal de Lavras, 2025.
Bibliografia.

1. Hemileia vastatrix. 2. Cercospora coffeicola. 3. Leucoptera coffeella. 4. Aprendizagem de Maquinas. 5. Visão Computacional. I. Diego Ferreira, Danton. II. Universidade Federal de Lavras. III. Título.

WESLEY RIBEIRO DE SOUZA

IDENTIFICAÇÃO DE FERRUGEM, CERCOSPORIOSE E BICHO-MINEIRO EM FOLHAS DE CAFEEIRO POR MEIO DE TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL

Projeto apresentado à Universidade Federal de Lavras, como parte das exigências do Programa de Pós Graduação em Engenharia de Sistemas e Automação, área de Engenharia de Sistemas e Automação, para a obtenção do título de Mestre.

APROVADA em 16 de Dezembro de 2025.

Prof. DSc. Danton Diego Ferreira	UFLA
Dr. Margarete Marin Lordelo Volpato	EPAMIG
Dr. Rogério Antônio Silva	EPAMIG
Prof. DSc. Giovani Bernades Vitor	UNIFEI

gov.br
Documento assinado digitalmente
DANTON DIEGO FERREIRA
Data: 06/03/2026 09:23:15-0300
Verifique em https://validar.iti.gov.br

Prof. DSc. Danton Diego Ferreira
Orientador

LAVRAS – MG
2025

Dedico este trabalho in memoriam de meu pai Eugênio Cláudio de Souza, ao qual foi minha referência ao longo dos meus anos de vida por optar e gostar de trabalhar com tecnologia.

AGRADECIMENTOS

Agradeço aos orientadores DSc. Danton Diego Ferreira e Dr. Margarete Marin Lordelo Volpato, aos vários colegas que me ajudaram ao longo do trabalho feito, ao programa de pós graduação em engenharia de sistemas e automação, a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), ao Consórcio Pesquisa Café (CP&D Café), a Empresa de Pesquisa Agropecuária de Minas Gerais (EPAMIG) pelo apoio técnico e permissão da coleta de dados. O presente trabalho foi realizado com apoio do Fundação de Amparo à Pesquisa de Minas Gerais (FAPEMIG)

"Ora, a vida não é senão um modo de ensaiar a construção de metáforas ou de experimentar metáforas construídas?"
(Manoel de Barros)

RESUMO

Eventos fitossanitários em folhas de cafeeiro figuram entre os principais desafios da cafeicultura, especialmente a ferrugem-do-cafeeiro, a cercosporiose e o bicho-mineiro, responsáveis por provocar desfolha precoce, redução da área fotossintética e consequente queda na produtividade. Os métodos tradicionais de diagnóstico baseados em inspeção visual apresentam limitações relacionadas à subjetividade, ao tempo de análise e à necessidade de especialistas, comprometendo a eficiência do monitoramento das lavouras. Diante desse cenário, este trabalho propõe o desenvolvimento e a avaliação de modelos para classificação automática de doenças foliares do cafeeiro empregando técnicas de visão computacional e aprendizado de máquina. São investigadas arquiteturas consolidadas de redes neurais convolucionais, como ResNet, DenseNet e EfficientNet, com a finalidade de identificar aquelas capazes de diferenciar, com maior precisão, folhas sadias daquelas afetadas por ferrugem, cercosporiose ou bicho-mineiro. Adicionalmente, são aplicadas as técnicas de Análise de Componentes Principais (PCA) e Minimally Random Convolutional Kernel Transform (MiniRocket) para redução de dimensionalidade e otimização do processamento, possibilitando o uso de classificadores tradicionais, como SVM e Random Forest, e promovendo soluções mais leves do ponto de vista computacional. A avaliação dos modelos é conduzida por meio de métricas como accuracy, precision, recall e F1-score, além da análise do impacto de carbono e do tempo de processamento associado ao treinamento e à inferência, buscando promover abordagens de inteligência artificial mais parcimoniosas e sustentáveis no contexto da agricultura digital. Os resultados obtidos indicam bom desempenho dos modelos (accuracy variando de 0.7311 à 0.9750, precision variando de 0.7453 à 0.9751, recall 0.7410 à 0.9751 e f1-score variando de 0.7323 à 0.9784 entre treinamento e validação final, respectivamente) em condições controladas, embora persistam desafios significativos quando aplicados a cenários reais de campo.

Palavras-chave: Hemileia vastatrix; Cercospora coffeicola; Leucoptera coffeella; Aprendizagem de Maquinas; Visão computacional.

ABSTRACT

Phytosanitary events affecting coffee leaves are among the main challenges in coffee production, especially coffee leaf rust, cercospora leaf spot, and the coffee leaf miner, which cause premature defoliation, reduction of the photosynthetic area, and consequently decreased productivity. Traditional diagnostic methods based on visual inspection present limitations related to subjectivity, analysis time, and the need for specialists, compromising the efficiency of crop monitoring. In this context, this study proposes the development and evaluation of models for the automatic classification of coffee leaf diseases using computer vision and machine learning techniques. Established convolutional neural network architectures, such as ResNet, DenseNet, and EfficientNet, are investigated in order to identify those capable of more accurately distinguishing healthy leaves from those affected by rust, cercospora, or leaf miner. Additionally, Principal Component Analysis (PCA) and the Minimally Random Convolutional Kernel Transform (MiniRocket) are applied for dimensionality reduction and processing optimization, enabling the use of traditional classifiers such as Support Vector Machines (SVM) and Random Forest, and promoting computationally lighter solutions. Model evaluation is conducted using metrics such as accuracy, precision, recall, and F1-score, along with an analysis of carbon impact and processing time associated with training and inference, aiming to promote more parsimonious and sustainable artificial intelligence approaches within the context of digital agriculture. The results indicate good model performance (accuracy ranging from 0.7311 to 0.9750, precision from 0.7453 to 0.9751, recall from 0.7410 to 0.9751, and F1-score from 0.7323 to 0.9784 between training and final validation, respectively) under controlled conditions, although significant challenges remain when applied to real field scenarios.
Keywords: Hemileia vastatrix; Cercospora coffeicola; Leucoptera coffeella; Machine Learning; Computer Vision.

INDICADORES DE IMPACTO

O uso de visão computacional na classificação de doenças e pragas em folhas de café representa um avanço significativo para a cafeicultura, pois permite identificar precocemente problemas fitossanitários como ferrugem, cercosporiose e bicho-mineiro com maior rapidez, precisão e padronização em comparação à inspeção visual tradicional. Ao reduzir a subjetividade da avaliação humana e possibilitar o monitoramento automatizado em larga escala, essa tecnologia contribui para decisões mais assertivas no manejo, diminui perdas produtivas, otimiza a aplicação de defensivos e reduz custos operacionais. Além disso, favorece práticas agrícolas mais sustentáveis ao evitar aplicações excessivas de produtos químicos, melhorar a eficiência do controle e apoiar sistemas de agricultura de precisão baseados em dados.

IMPACT INDICATORS

The use of computer vision for the classification of diseases and pests in coffee leaves represents a significant advancement for coffee production, as it enables the early identification of phytosanitary problems such as coffee leaf rust, cercospora leaf spot, and leaf miner with greater speed, accuracy, and standardization compared to traditional visual inspection. By reducing the subjectivity of human assessment and enabling largescale automated monitoring, this technology supports more assertive management decisions, decreases yield losses, optimizes the application of agrochemicals, and reduces operational costs. Furthermore, it promotes more sustainable agricultural practices by preventing excessive chemical applications, improving control efficiency, and supporting data-driven precision agriculture systems.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

1.1 Contexto, Motivação e Desafios

Segundo a matéria publicada em 2023 pelo Ministério da Agricultura e Pecuária, o café é a segunda bebida mais consumida no mundo, ficando atrás apenas da água (PECUáRIA, 2023). De acordo com a Organização Internacional do Café (OIC), entre outubro de 2021 e setembro de 2022, a produção mundial de café foi de 170,83 milhões de sacas de 60 quilos, enquanto o consumo da bebida alcançou 164,9 milhões de sacas.

O Brasil se destaca como o maior produtor global e o segundo maior consumidor de café, atrás apenas dos Estados Unidos (PECUáRIA, 2023). Em 2022, o país exportou aproximadamente 2,2 milhões de toneladas, equivalentes a 39,4 milhões de sacas de café, para 145 países, com os principais destinos sendo Estados Unidos, Alemanha, Itália, Bélgica e Japão. O valor elevado do café no mercado internacional permitiu que as exportações brasileiras de café verde, solúvel e extratos alcançassem US$ 9,2 bilhões no ano passado.

A importância cultural do café se faz presente na história do Brasil desde o século XIX, na época do Império. Após os ciclos econômicos do pau-brasil, da cana-de-açúcar e do ouro, o ciclo do café, que foi até o século XX, representou a maior fonte de riqueza do país e o principal produto de exportação.

Devido à grande importância do café na economia, seu monitoramento torna-se fundamental. Pragas e doenças afetam tanto a qualidade quanto a quantidade de produção, além de causarem danos fisiológicos à plantação. Por esses motivos, o monitoramento dos cafeeiros em campo é imprescindível para haja uma tomada de decisão rápida quanto ao manejo do cafeeiro. Entretanto, uma das principais dificuldades para o sucesso do manejo integrado é a identificação oportuna de pragas e doenças nos cafeeiros em campo (SOUZA et al., 2004).

As pragas mais recorrentes considerado praga chave da cultura são: bicho-mineiro-do-cafeeiro (*Leucoptera coffeella*) e broca-do-café (*Hypothenemus hampei*). Por sua vez, as doenças mais recorrentes são: ferrugem-do-cafeeiro (*Hemileia vastatrix*) e cercosporiose (*Cercospora coffeicola*) (REIS; CUNHA, 2010).

A incidência de fungos, bactérias, nematoides e vírus representa um desafio significativo para a produção agrícola, podendo reduzir os rendimentos em até 20% se não forem devidamente controlados. Esses agentes patogênicos afetam diretamente a saúde das plantas, comprometendo tanto a quantidade quanto a qualidade dos produtos cultivados. Nesse cenário, intervenções rápidas e eficazes tornam-se essenciais para mitigar os danos e garantir a produtividade, destacando a importância de estratégias preventivas e tecnologias de monitoramento para detecção precoce (ALBUQUERQUE; GUEDES, 2024).

Porém, a identificação dessas pragas e doenças no campo nem sempre é fácil. Métodos convencionais de diagnóstico de doenças, como inspeção visual, são demorados, imprecisos e dependem de especialistas para serem confirmados, o que pode comprometer toda a produção caso não haja uma intervenção em tempo hábil para conter os danos à planta e, consequentemente, à produção de café.

Os diagnósticos manuais, baseados na observação visual, são frequentemente subjetivos e suscetíveis a erros, o que dificulta a realização de intervenções rápidas e eficazes. Nesse contexto, a implementação de soluções que acelerem o diagnóstico e aumentem a precisão na identificação de problemas em campo é essencial para aprimorar a gestão agrícola e minimizar perdas.

Ferramentas computacionais de reconhecimento das pragas e doenças podem auxiliar no monitoramento e na tomada de decisão quanto ao manejo dos cafeeiros e têm potencial

de evitar danos à produção e à qualidade do café, bem como o uso incorreto de defensivos agrícolas, proporcionando economia financeira, segurança alimentar e menor impacto ao meio ambiente.

Dessa forma, o uso da visão computacional pode proporcionar um modelo confiável para a identificação e alerta do início das ocorrências de pragas e da incidência de doenças no cafeeiro, viabilizando uma atuação eficaz na contenção e controle de sua propagação (TAJANE et al., 2024).

A visão computacional tem sido aplicada com sucesso no setor agrícola, especialmente na identificação de doenças em plantas. As redes neurais convolucionais (CNNs), combinadas com técnicas de segmentação e detecção, permitem a análise automatizada de grandes volumes de imagens de plantações, facilitando o monitoramento da saúde das culturas e a identificação precoce de doenças.

A visão computacional destaca-se pela capacidade de extrair e processar informações a partir de dados de imagens. Em termos gerais, essa técnica da computação combina hardware e software com o objetivo de replicar, ainda que de forma simplificada, as complexas habilidades da visão humana (QIU et al., 2021). Um dos principais propósitos da visão computacional é interpretar e extrair informações de imagens 2D, permitindo a classificação e o reconhecimento do conteúdo representado. Essa característica tem possibilitado sua aplicação em diversas áreas, como veículos autônomos, inspeção de elementos, sistemas de segurança e vigilância, análises esportivas e classificação de itens (CHEN; LITTLE, 2017; PINTO et al., 2017).

Devido à ampla variedade de aplicações, a visão computacional tem ganhado notoriedade e tem sido frequentemente integrada a outras técnicas computacionais. Seu objetivo principal é unir hardware e software para permitir, por exemplo, a análise de cenários por meio de câmeras, extraindo informações como distância de obstáculos, detecção de pessoas ou objetos, e, com base nesses dados, auxiliar na tomada de decisões por outros componentes do sistema. Essa integração tem ampliado o potencial da visão computacional, consolidando-a como uma ferramenta essencial em diversos contextos tecnológicos para a aplicação na agricultura.

Este estudo tem como propósito desenvolver e avaliar um sistema de classificação de doenças foliares em cafeeiro por meio de técnicas avançadas de visão computacional e aprendizado de máquina. Para isso, são investigadas arquiteturas de redes neurais convolucionais, métodos de extração de features e estratégias de redução de dimensionalidade, aplicadas à identificação de ferrugem-do-cafeeiro, cercosporiose e bicho-mineiro. A abordagem busca não apenas analisar o desempenho e a capacidade de generalização dos modelos em cenários reais, mas também contribuir para o aprimoramento de ferramentas de apoio ao manejo fitossanitário, favorecendo práticas agrícolas mais eficientes e tecnicamente embasadas.

1.2 Objetivo

O objetivo deste trabalho foi desenvolver e avaliar um sistema de classificação de doenças foliares baseado em técnicas avançadas de visão computacional e aprendizado de máquina, integrando arquiteturas de redes neurais convolucionais (como EfficientNet, DenseNet e ResNet) com métodos de redução de dimensionalidade e classificadores tradicionais, a fim de investigar sua capacidade de generalização em cenários reais por meio de validações internas e externas.

1.2.1 Objetivos Específicos

- Construção de um banco de imagens de folhas de café infectadas com ferrugem-do-cafeeiro *(Hemileia vastatrix)*, cercosporiose *(Cercospora coffeicola* e bicho-mineiro-do-cafeeiro *(Leucoptera coffeella)*, coletadas em campos experimetrais da EPAMIG (Empresa de Pesquisa Agropecuária de Minas Gerais), para compor os conjuntos de treinamento, teste e validação;
- Analisar o desempenho de diferentes arquiteturas de CNNs aplicadas ao reconhecimento de doenças e pragas, considerando métricas de acurácia, precisão, sensibilidade e f1-score.
- Implementar estratégias de extração de features por meio de modelos pré-treinados, visando representar de forma eficiente os padrões visuais presentes nas imagens.
- Aplicar técnicas de redução de dimensionalidade, como a Análise de Componentes Principais (PCA) e transformações rápidas como o MiniRocket, buscando otimização computacional e preservação de informação relevante.
- Treinar e comparar classificadores supervisionados utilizando features reduzidas, avaliando sua robustez e estabilidade por meio de validação cruzada (K-Fold).
- Realizar validação externa com banco de dados coletado em campo, a fim de examinar o nível de generalização dos modelos e identificar limitações inerentes ao domínio real.
- Comparar de forma crítica os resultados internos e externos, destacando os desafios enfrentados pelos modelos diante de variações reais de iluminação, textura, presença de múltiplas doenças e ausência de padronização nas imagens.
- Identificar as combinações de métodos mais promissoras para aplicações práticas, considerando desempenho, custo computacional e viabilidade de uso em campo.

1.3 Estrutura do Texto

No capítulo 2, será realizada uma revisão bibliográfica sobre os conceitos fundamentais de Inteligência Artificial (IA), Aprendizado de Máquina (Machine Learning), Aprendizado Profundo (Deep Learning) e Visão Computacional. No capítulo 3, serão apresentados materiais e métodos utilizados, divididos em banco de dados, modelos de classificação, treinamento do modelo e os parâmetros de avaliação. No capítulo 4, serão detalhados os resultados e discussões dos modelos propostos. Por fim, no capítulo 5, serão apresentadas as conclusões e referências bibliográficas utilizadas ao longo do trabalho, garantindo a fundamentação teórica e técnica do estudo.

2 REVISÃO BIBLIOGRÁFICA

2.1 Inteligência Artificial

A inteligência artificial (IA) é definida como o campo da ciência da computação que se concentra no desenvolvimento de sistemas e máquinas capazes de executar tarefas que tradicionalmente exigiriam inteligência humana, como reconhecimento de fala, tomada de decisão, tradução de idiomas e percepção visual (RUSSELL; NORVIG, 2016).

O conceito de IA abrange diversas interpretações, dependendo do contexto. Em um sentido estrito, a IA refere-se a sistemas que simulam processos cognitivos humanos, como raciocínio, aprendizado e adaptação, utilizando algoritmos que analisam grandes quantidades de dados (GOODFELLOW et al., 2016a).

A IA pode ser dividida em duas grandes categorias: IA estreita (ou fraca) e IA geral (ou forte). A IA estreita é desenvolvida para realizar uma tarefa específica, como reconhecimento facial ou recomendação de produtos, sendo o tipo de IA mais comum atualmente. Por outro lado, a IA geral tem como objetivo replicar a inteligência humana em uma escala mais ampla, sendo capaz de realizar qualquer tarefa intelectual que um ser humano conseguiria (LEGG; HUTTER, 2007).

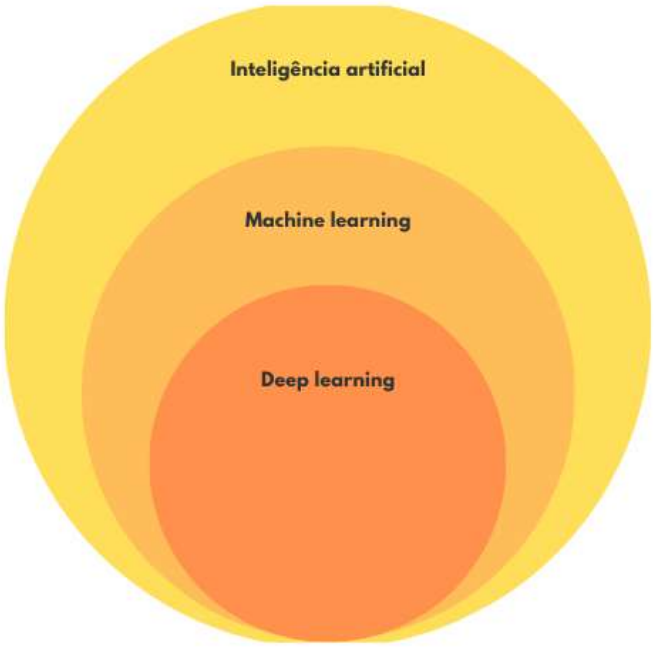
Outra perspectiva importante sobre a IA é o aprendizado de máquina (*machine learning*), uma subárea da IA que permite que os sistemas aprendam e melhorem automaticamente a partir de experiências, sem a necessidade de programação explícita. O *machine learning*, juntamente com o aprendizado profundo (*deep learning*), que utiliza redes neurais artificiais, tem proporcionado muitos dos avanços recentes em IA (SCHMIDHUBER, 2015).

Além disso, a IA pode ser vista de uma perspectiva funcional e prática, na qual sistemas de IA são considerados inteligentes na medida em que demonstram comportamentos adaptativos que resultam em soluções eficazes para problemas complexos no mundo real (NILSSON, 2009).

Dentro do vasto universo da IA, destacam-se áreas específicas como redes neurais artificiais, aprendizado profundo (*deep learning*), processamento de linguagem natural (PLN) e robótica (GOODFELLOW et al., 2016a). Cada uma dessas áreas contribui de maneira única para o desenvolvimento de soluções inovadoras e atuações em setores diversos, como saúde, finanças, transporte e entretenimento (JURAFSKY; MARTIN, 2008).

O aprendizado profundo (*deep learning*), em particular, tem mostrado resultados impressionantes em tarefas complexas, como a identificação de padrões em grandes conjuntos de dados e a tomada de decisões em tempo real. No entanto, essa capacidade de adaptação também pode levar a comportamentos inesperados ou indesejados, especialmente em sistemas de IA que operam em ambientes dinâmicos e imprevisíveis (LECUN et al., 2015). Por exemplo, o uso de IA em veículos autônomos, diagnósticos médicos e sistemas financeiros exige garantias de que esses sistemas operem de maneira segura e eficaz (SILVER et al., 2016).

Figura 2.1 – Hierarquia IA x ML x DL.



Fonte: do Autor

2.1.1 Aprendizado de Máquina (*Machine Learning*)

O campo de *machine learning* envolve uma variedade de métodos e técnicas do qual permitem que sistemas aprendam a partir de dados, realizar previsões ou tomar decisões sem a necessidade de programação explícita para cada tarefa. As máquinas podem aprender de três formas: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço. Cada uma dessas abordagens tem características distintas e é utilizada para diferentes finalidades no campo da inteligência artificial.

2.1.1.1 Aprendizado Supervisionado

No aprendizado supervisionado, o modelo é treinado usando um conjunto de dados rotulados, onde cada entrada está associada a uma saída desejada. Esse processo permite que o modelo aprenda a mapear entradas para saídas corretas com base nos exemplos fornecidos. As técnicas mais comuns nesse tipo de aprendizado incluem:

- **Regressão Linear e Regressão Logística:** Utilizadas para prever valores contínuos e probabilidades binárias, respectivamente. A regressão linear tenta encontrar a melhor linha que se ajusta aos dados, enquanto a regressão logística modela a probabilidade de um evento binário ocorrer (JAMES et al., 2013).
- **Árvores de Decisão:** Um método que utiliza uma estrutura em árvore para dividir os dados em subconjuntos baseados em critérios específicos, o que facilita a tomada de decisões. Árvores de decisão são amplamente usadas pela sua simplicidade e interpretabilidade (QUINLAN, 1986).

- **Máquinas de Vetores de Suporte (SVM):** Uma técnica poderosa que cria hiperplanos em um espaço multidimensional para separar diferentes classes de dados. As SVMs são eficazes em alta dimensionalidade e quando a separação de classes é clara (VAPNIK, 1995)

- **Redes Neurais:** Inspiradas na estrutura do cérebro humano, as redes neurais artificiais consistem em camadas de neurônios artificiais que processam a informação e aprendem a realizar tarefas complexas, como reconhecimento de padrões e imagens (GOODFELLOW et al., 2016b).

2.1.1.2 Aprendizado Não Supervisionado

O aprendizado não supervisionado lida com dados não rotulados e busca descobrir padrões ou estruturas ocultas dentro dos dados. Técnicas comuns incluem:

- ***Clustering* (Agrupamento):** Um método para agrupar objetos semelhantes em *clusters*. O algoritmo *K-means* é uma técnica popular que divide os dados em *K clusters* por meio de suas características (MACQUEEN, 1967).

- **Análise de Componentes Principais (PCA):** Uma técnica de redução de dimensionalidade que transforma variáveis altamente correlacionadas em um conjunto menor de variáveis não correlacionadas, chamadas de componentes principais. PCA é usada para simplificar modelos e visualizar dados complexos (PEARSON, 1901).

- **Associação:** Utilizada para encontrar relações entre variáveis em grandes conjuntos de dados. Os algoritmos de regras de associação, como o Apriori, são amplamente utilizados em análise de cestas de compras para identificar produtos que são frequentemente comprados juntos (AGRAWAL et al., 1994).

2.1.1.3 Aprendizado por Reforço

O aprendizado por reforço é uma abordagem onde o modelo aprende a tomar decisões sequenciais por meio de interações com um ambiente dinâmico. O agente recebe recompensas ou punições com base nas ações que realiza e ajusta sua estratégia para maximizar a recompensa total ao longo do tempo. Técnicas comuns incluem:

- ***Q-Learning*:** Um algoritmo de aprendizado por reforço que busca a melhor política de ação para maximizar a recompensa esperada. O agente aprende a estimar o valor das ações em diferentes estados e utiliza essas estimativas para guiar suas decisões (WATKINS; DAYAN, 1992).

- **Política Gradiente:** Um método que otimiza diretamente as políticas de decisão, em vez de estimar valores de ação. Essa técnica é particularmente útil em ambientes complexos onde as ações influenciam não apenas recompensas imediatas, mas também estados futuros (SUTTON, 2018).

2.1.2 Aprendizado Profundo (*Deep Learning*)

Deep Learning (Aprendizado Profundo) é uma subárea do *machine learning* que se concentra no uso de redes neurais artificiais profundas, compostas por várias camadas de neurônios,

para modelar e entender dados complexos. Esse campo emergiu como uma das áreas mais dinâmicas da inteligência artificial, impulsionando avanços significativos em diversos domínios, como reconhecimento de fala, visão computacional, processamento de linguagem natural e jogos.

2.1.2.1 Estrutura das Redes Neurais Profundas

A base do *deep learning* está nas redes neurais artificiais, que são compostas por camadas de unidades de processamento, chamadas neurônios, que estão organizadas em uma estrutura em camadas: camada de entrada, camadas ocultas (*deep layers*) e camada de saída. Cada neurônio em uma camada está conectado a neurônios da camada seguinte através de pesos ajustáveis, e o aprendizado ocorre através da atualização desses pesos para minimizar a diferença entre a saída prevista pela rede e a saída real (RUMELHART et al., 1986).

As redes neurais profundas distinguem-se das redes neurais tradicionais por possuírem um grande número de camadas ocultas. Essa profundidade permite que as redes capturem características complexas e abstratas dos dados, tornando-as particularmente eficazes para tarefas como reconhecimento de padrões e classificação de imagens (LECUN et al., 2015).

2.1.2.2 Técnicas e Arquiteturas de Deep Learning

Existem várias arquiteturas e técnicas no campo do *deep learning*, cada uma adequada para diferentes tipos de problemas:

- **Redes Neurais Convolucionais (CNNs):** CNNs são especialmente eficazes para tarefas de visão computacional, como reconhecimento de imagens e detecção de objetos. Elas utilizam camadas convolucionais que aplicam filtros a pequenos segmentos dos dados de entrada, capturando características locais, como bordas e texturas. O trabalho seminal de LeCun et al. (1998) introduziu as CNNs com sucesso no reconhecimento de dígitos manuscritos, o que veio a se tornar um marco na área.

- **Redes Neurais Recorrentes (RNNs):** RNNs são utilizadas principalmente para processamento de dados sequenciais, como séries temporais e linguagem natural. Elas possuem conexões que formam ciclos dentro da rede, permitindo que informações de etapas anteriores sejam mantidas e usadas nas etapas subsequentes. No entanto, as RNNs tradicionais enfrentam desafios com dependências de longo prazo, o que levou ao desenvolvimento de variantes como LSTM (*Long Short-Term Memory*) e GRU (*Gated Recurrent Units*), que foram introduzidas por Hochreiter (1997) para lidar com esses problemas.

- **Redes Adversárias Generativas (GANs):** GANs são uma arquitetura inovadora introduzida por Goodfellow et al. (2014), composta por duas redes neurais que competem entre si: uma rede geradora, que tenta criar dados falsos que imitam os dados reais, e uma rede discriminadora, que tenta distinguir entre os dados reais e os dados gerados. As GANs têm sido amplamente utilizadas para geração de imagens, vídeos, música e até dados sintéticos para treinamento de outros modelos de machine learning.

- **Redes Neurais Profundas (DNNs):** DNNs são redes neurais com múltiplas camadas ocultas, que podem ser aplicadas a uma ampla gama de problemas, desde reconhecimento de fala até previsões financeiras. A profundidade das DNNs permite que elas aprendam representações hierárquicas dos dados, capturando tanto características simples quanto complexas.

2.1.2.3 Treinamento e Desafios

O treinamento de redes neurais profundas requer grandes volumes de dados e poder computacional significativo. Uma das técnicas cruciais para o sucesso do deep learning é o uso de grandes conjuntos de dados anotados, como o ImageNet, que contém milhões de imagens rotuladas e foi fundamental para os avanços em reconhecimento de imagem (DENG et al., 2009).

Outro desafio importante é o fenômeno do *overfitting*, onde o modelo aprende muito bem os dados de treinamento, mas não consegue generalizar para novos dados. Técnicas como regularização, *dropout* (SRIVASTAVA et al., 2014) e o uso de dados aumentados ajudam a mitigar esse problema, melhorando a capacidade de generalização dos modelos.

2.1.2.4 Aplicações e Impacto

Deep learning revolucionou diversas áreas ao possibilitar avanços antes inatingíveis com técnicas tradicionais de machine learning. Em visão computacional, *deep learning* é a tecnologia por trás de sistemas de reconhecimento facial, diagnóstico médico assistido por IA, e veículos autônomos (KRIZHEVSKY et al., 2017). No processamento de linguagem natural, *deep learning* tem sido crucial no desenvolvimento de tradutores automáticos, *chatbots* e assistentes virtuais, como os sistemas baseados em *transformers* (VASWANI, 2017).

O campo do *deep learning* continua a evoluir rapidamente, com novas técnicas e arquiteturas sendo desenvolvidas continuamente. A pesquisa atual se concentra em tornar os modelos mais interpretáveis, eficientes em termos de energia e capazes de aprender com menos dados. A combinação de *deep learning* com outras tecnologias emergentes, como computação quântica, também é uma área de intenso interesse e potencial.

2.2 Visão Computacional

A visão computacional tem se destacado como uma das áreas mais promissoras da inteligência artificial (IA), desempenhando um papel crucial na automação de tarefas que envolvem a análise de imagens e vídeos. Com a evolução de algoritmos de aprendizado de máquina, particularmente o aprendizado profundo (*deep learning*), a capacidade de máquinas em interpretar e processar imagens tem se tornado cada vez mais precisa, atingindo níveis de desempenho comparáveis e, em alguns casos, superiores aos humanos em determinadas tarefas.

A visão computacional possui diversas aplicações, sendo amplamente utilizada em três principais áreas:

- Classificação de imagens;
- Detecção de Objetos;
- Segmentação de imagens.

2.2.1 Classificação de Imagens

A classificação de imagens é uma das tarefas fundamentais no campo da visão computacional, uma área da inteligência artificial que busca permitir que máquinas "enxerguem"e interpretem o mundo visual. A classificação de imagens é um processo que envolve a análise de uma imagem e a atribuição de uma ou mais categorias predefinidas a ela. Esse processo é amplamente utilizado em aplicações como reconhecimento facial, diagnóstico médico por

imagens, veículos autônomos, monitoramento de segurança e muito mais (SZELISKI, 2022). O desafio principal é ensinar uma máquina a reconhecer padrões visuais em imagens, como formas, cores, texturas e objetos, e associá-los a categorias específicas.

Dentre as principais técnicas de classificação de imagens, destacam-se a **Classificação Supervisionada** e a **Classificação Não Supervisionada**. A principal diferença entre elas está no uso de dados rotulados: enquanto a classificação supervisionada utiliza um conjunto de dados com rótulos predefinidos, a classificação não supervisionada não depende de rótulos, buscando padrões e estruturas nos dados de forma autônoma. No caso da classificação supervisionada, o algoritmo aprende a partir dos exemplos fornecidos, ajustando suas respostas para se adequar às informações rotuladas. No entanto, essa abordagem exige um esforço prévio significativo para rotular o conjunto de dados, o que pode ser uma desvantagem em comparação com métodos não supervisionados. É importante ressaltar que não existe uma técnica universalmente ideal para todos os problema e a escolha do método mais adequado depende das características específicas do problema e dos dados disponíveis (TUIA et al., 2011)

No processo de classificação de imagens, independentemente da técnica escolhida, uma etapa crucial é a extração de características. Essa etapa consiste em analisar cada pixel da imagem e transformar as informações visuais em um formato que possa ser processado por algoritmos de aprendizado de máquina. Para isso, são utilizadas técnicas como **HOG (Histogram of Oriented Gradients)** e **LBP (Local Binary Patterns)**, que convertem a imagem em um vetor de características, capturando aspectos como texturas, bordas e padrões locais. Esse vetor é então submetido a um modelo classificador, como por exemplo Support Vector Machine (SVM) ou uma rede neural artificial, cujo objetivo é interpretar os dados de entrada, analisar as características extraídas e estimar a probabilidade de um conjunto de pixels pertencer a uma determinada classe. Esses modelos são projetados para identificar padrões complexos nos dados e fornecer uma classificação precisa com base nas informações disponíveis.

2.2.2 Detecção de Objetos

A detecção de objetos envolve a localização de um ou mais objetos em uma imagem, o que inclui identificar sua posição, suas dimensões e demarcar sua área de ocorrência. Diferentemente da classificação de imagens, que se concentra em prever a classe de uma imagem como um todo, a detecção de objetos exige não apenas a identificação da classe do objeto, mas também a definição precisa de sua localização. Para isso, utiliza-se um retângulo delimitador (conhecido como bounding box) ao redor do objeto detectado (WU et al., 2020; ZHENG; WANG, 2017; GIRSHICK et al., 2014). Essa abordagem permite, por exemplo, contar quantos objetos de uma determinada classe estão presentes em uma imagem, ampliando as possibilidades de análise e aplicação em cenários como monitoramento, reconhecimento de padrões e automação de processos.

Atualmente, as técnicas de detecção de objetos podem ser divididas em duas principais vertentes: métodos de uma etapa e métodos de duas etapas.

Os algoritmos de uma etapa, como o YOLO (You Only Look Once) e o SSD (Single Shot Detector), são projetados para priorizar a velocidade de inferência. Eles realizam a detecção de objetos em uma única passagem pela rede neural, o que os torna altamente eficientes e adequados para aplicações que exigem processamento em tempo real, como sistemas de vigilância ou veículos autônomos.

Por outro lado, os algoritmos de duas etapas, como o R-CNN (Region-based Convolutional Neural Networks) e o Cascade R-CNN, seguem uma abordagem mais detalhada. Na primeira etapa, eles identificam e delimitam as regiões de interesse na imagem. Em seguida, na

segunda etapa, refinam e classificam essas regiões. Essa abordagem prioriza a acuracidade na detecção, sendo mais precisa em cenários que exigem alto nível de detalhamento. No entanto, devido às etapas adicionais e ao foco no refinamento, esses métodos tendem a ser mais lentos em comparação com os de uma etapa (LIU et al., 2016)

2.2.3 Segmentação de Imagens

No contexto da visão computacional, a segmentação de imagens é uma área dedicada à extração de características de uma imagem por meio da subdivisão de suas partes constituintes ou objetos. Esse processo de dividir a imagem em subgrupos de características tem como principal objetivo aumentar a confiabilidade e a precisão ao extrair informações relevantes de um conjunto de pixels.

Existem diferentes tipos de segmentação, sendo os principais a **segmentação por descontinuidade** e a **segmentação por similaridade**, conforme destacado em (CHU et al., 2020) e (WU et al., 2024). A segmentação por descontinuidade busca identificar pontos na imagem que apresentem mudanças abruptas em suas características, como bordas, linhas, contornos ou pontos isolados. Já a segmentação por similaridade tem como foco agrupar pixels com características semelhantes, seguindo critérios pré-definidos. Dentre as técnicas mais comuns para esse tipo de segmentação estão a limiarização (thresholding), o crescimento de regiões (Region Growing) e a divisão e conquista (Split and Merge), conforme descrito em (RONCERO, 2005; PAVLIDIS; LIOW, 1990)

Além da divisão por tipo de técnica, a segmentação de imagens também pode ser classificada em três categorias principais:

- segmentação semântica;
- segmentação por instâncias;
- segmentação panóptica.

As técnicas de segmentação diferenciam-se por seus objetivos finais, embora todas classifiquem objetos em grupos. A **segmentação semântica** rotula pixels e agrupa-os por similaridade, focando na identificação e separação de objetos com características comuns. Já a **segmentação por instâncias** vai além, identificando individualmente cada objeto detectado. Por fim, a **segmentação panóptica** combina ambas: classifica todos os pixels, agrupa-os por classe e rotula cada elemento individualmente (HU et al., 2018; HWANG et al., 2025; LIAO et al., 2025; SHIM; LEE, 2025; SILVA, 2025)

2.2.4 Processo Embedding

O processo de *embedding* consiste em transformar dados complexos como imagens, palavras, sons ou categorias em vetores numéricos densos que representam, em um espaço contínuo, as características semânticas mais relevantes desses dados. Essa transformação é fundamental porque os modelos de aprendizado profundo operam sobre valores numéricos; portanto, o *embedding* converte dados de alta dimensionalidade e natureza simbólica em uma representação que as redes neurais possam processar e aprender padrões de forma eficiente (BENGIO et al., 2013)

Em essência, um *embedding* é uma representação vetorial compacta que preserva relações de similaridade entre os elementos. Por exemplo, em modelos de linguagem, palavras com

significados próximos como “café” e “xícara” possuem vetores próximos no espaço de *embedding* (MIKOLOV et al., 2013). Da mesma forma, em visão computacional, imagens de folhas com a mesma doença produzem *embedding* similares, enquanto imagens de classes distintas ocupam regiões mais afastadas do espaço vetorial (LECUN et al., 2015).

O processo de *embedding* é aprendido automaticamente durante o treinamento das redes neurais. Nas redes convolucionais (CNNs), as camadas convolucionais extraem gradualmente características visuais de baixo e alto nível (bordas, texturas, formas e padrões), e as camadas finais, normalmente antes da camada de classificação (fully connected ou softmax), produzem um vetor de características que representa a imagem de forma densa. Esse vetor, geralmente de algumas centenas ou milhares de dimensões, constitui o *embedding* da imagem (SIMONYAN; ZISSERMAN, 2015). Ele pode ser utilizado como descritor para tarefas de classificação, busca por similaridade, agrupamento (clustering) ou detecção de anomalias.

Na prática, gerar *embedding* envolve o uso de uma rede previamente treinada (por exemplo, DenseNet, ResNet ou EfficientNet), removendo sua camada de saída e capturando a penúltima camada, que contém as características aprendidas pelo modelo. Essa abordagem é amplamente utilizada em aprendizado por transferência (*transfer learning*), permitindo aproveitar representações otimizadas em grandes bases de dados, como o ImageNet, para resolver outros problemas com menos dados (CUI et al., 2018).

Assim, o embedding atua como uma forma de compressão semântica dos dados, preservando informações essenciais e reduzindo a dimensionalidade de modo significativo. Ele é, portanto, um elemento central no aprendizado de representações (*representation learning*), servindo de base para a generalização e a interpretação de padrões em redes neurais profundas (LECUN et al., 2015; BENGIO et al., 2013).

2.2.5 Trabalhos Relacionados

A visão computacional tem sido aplicada com sucesso no setor agrícola, especialmente na identificação de doenças em plantas. As redes neurais convolucionais, combinadas com técnicas de segmentação e detecção, permitem a análise automatizada de grandes volumes de imagens de plantações, facilitando o monitoramento da saúde das culturas e a identificação precoce de doenças (FERENTINOS, 2018), desta forma, os estudos recentes mostram que modelos baseados em CNN conseguem identificar doenças em plantas com alta precisão, superando métodos tradicionais de análise visual (FERENTINOS, 2018). Esse avanço é particularmente relevante no contexto do cultivo de café, onde a detecção precoce de doenças como a ferrugem pode minimizar perdas e melhorar a eficiência do manejo agrícola.

O estudo realizado por Türkoğlu e Hanbay (2019) propôs a mensuração do grau de severidade utilizando um sistema de telefone móvel que pode reconhecer e classificar instantaneamente o minador e a ferrugem das folhas do café . Eles empregaram a técnica de K-means, a técnica de Otsu e uma técnica iterativa de limiarização em dois espaços de cores (HSV e YCbCr) para segmentar as áreas afetadas por doenças e pragas nas folhas de café das áreas saudáveis. Para treinar o modelo, foram usadas máquinas de aprendizado extremo (ELM), redes neurais de retropropagação (BPNN) e redes neurais artificiais - ANN (AlexNet, VGG16, VGG19, GoogleNet, ResNet50, ResNet101, InceptionV3, Inception-ResNetV2), como . No final, a ELM superou a ANN e a BPNN por uma pequena margem.

No estudo conduzido por Paulos e Woldeyohannis (2022) apresenta o uso de redes neurais convolucionais (CNNs) para a detecção e classificação de doenças em folhas de café, incluindo ferrugem, manchas marrons e murcha. Os autores utilizaram um conjunto de dados contendo 1.120 imagens, e após o uso de técnicas de aumento de dados, chegou a 3.360 ima-

gens. Compararam treinamento a partir de pesos iniciais aleatórios com aprendizado por transferência, utilizando as arquiteturas MobileNet e ResNet50. O modelo baseado em ResNet50 alcançou 99,89% de acurácia, superando os métodos alternativos. O estudo demonstra a viabilidade de aplicar aprendizado profundo para identificar doenças em folhas de café de forma eficiente, enfatizando o potencial para otimizar a produtividade e qualidade no setor cafeeiro.

Já estudo conduzido por Işik e Eskicioglu (2022) foi investigar a classificação de folhas de café não saudáveis usando Redes Neurais Convolucionais (CNNs). Utilizaram um conjunto de dados do Kaggle (2025) com 542 amostras, empregando oito arquiteturas diferentes de CNNs (como VGG-16 e ResNet50). Obtiveram acurácias superiores a 95%, destacando o impacto do pré-processamento no desempenho do modelo.

No trabalho de Novtahaning et al. (2022) encontramos um método ensemble com arquiteturas CNN pré-treinadas (VGG-16, EfficientNetB0 e ResNet152) para classificar e detectar com precisão quatro classes entre doenças e pragas, nas folhas do cafeeiro (phoma, minador, ferrugem e cercosporiose) em imagens. Utilizando uma estratégia de votação, alcançaram uma precisão de 97,3% e um F1-Score de 95,1%. No entanto, a abordagem apresentou alta complexidade computacional devido ao grande número de parâmetros.

Aufar et al. (2023) utilizaram os conjuntos de dados JMuBEN e JMuBEN2, contendo 58.550 amostras, para treinar arquiteturas como ResNet50 e DenseNet169. Obtiveram uma precisão experimental de 100% em algumas arquiteturas, mas não demonstrou as medidas estatísticas de dispersão.

Yamashita e Leite (2023) propuseram um sistema baseado em aprendizado profundo para classificar doenças em folhas de café, visando a implementação eficiente em dispositivos de borda com recursos limitados. O estudo utilizou a arquitetura MobileNetV2 devido ao seu bom equilíbrio entre precisão e leveza computacional, treinando o modelo com um conjunto de dados composto por imagens de folhas saudáveis e com três tipos principais de doenças: ferrugem, phoma e cercosporiose. Após o treinamento e avaliação, o modelo alcançou uma acurácia de 98,83%, demonstrando alta eficácia na classificação das doenças. Além disso, os testes mostraram que o modelo é adequado para execução em dispositivos embarcados, como o NVIDIA Jetson Nano, com tempos de inferência satisfatórios, o que reforça seu potencial para aplicação prática em campo por produtores agrícolas.

Abuhayi e Mossa (2023) propõem um método baseado em Redes Neurais Convolucionais (CNNs) para classificação de doenças em plantas de café, utilizando a concatenação de características extraídas das arquiteturas GoogLeNet e RESNET. O processo envolveu três etapas principais: pré-processamento de imagens (aplicação de filtro Gaussiano e aumento de dados), extração de características (combinação de GoogLeNet e RESNET) e classificação (usando MLPs, algoritmos clássicos de aprendizado de máquina e técnicas de ensemble). O modelo alcançou uma precisão de teste de 99,08%, superando classificadores individuais e outros métodos existentes na literatura. Além disso, técnicas como filtragem Gaussiana e segmentação personalizada demonstraram ser eficazes para redução de ruído e extração de regiões de interesse. O estudo também destacou a importância do balanceamento entre complexidade do modelo e tempo de treinamento, mostrando que o aumento excessivo de parâmetros pode levar a overfitting. Os resultados indicam que a abordagem proposta é promissora para a detecção precoce de doenças em cafeeiros, com potencial aplicação prática na agricultura.

O artigo de Albuquerque e Guedes (2024) apresenta tecnicas sobre uso de Visão Computacional e Inteligência Artificial no diagnóstico automatizado de doenças em folhas de café. Ele explora técnicas como segmentação baseada no algoritmo JSEG e o uso de redes neurais convolucionais (CNNs) para classificação de amostras. Estudos prévios abordados incluem o uso de métodos ensemble, aprendizado por transferência e arquiteturas pré-treinadas, como

VGG-16 e ResNet50, com acurácias superiores a 95%. O trabalho destaca a eficácia dessas abordagens, apesar dos desafios relacionados à complexidade computacional e à generalização em condições reais.

Na revisão bibliográfica de Signo et al. (2024) apresentam sobre o uso de técnicas de aprendizado profundo, especialmente redes neurais convolucionais (CNNs), na detecção de doenças em plantas por meio de imagens de folhas. A revisão explora diversas abordagens e estudos anteriores que aplicam CNNs para identificar sintomas visuais de doenças foliares, destacando a eficácia dessas técnicas na agricultura de precisão. Além disso, o artigo demonstra a implementação prática de um modelo baseado em CNN, que foi treinado e testado com um conjunto de dados contendo 38 classes de doenças. Como resultado, o modelo alcançou uma acurácia de 98,56% no treinamento e 94,10% na fase de teste, reforçando o potencial das CNNs como ferramenta confiável para o diagnóstico automatizado de doenças em plantas.

O artigo de revisão bibliográfica de Leite et al. (2024) propõe o modelo leve e eficiente chamado GreenDiseaseNet para a classificação de doenças em folhas, com foco em aplicações em dispositivos com recursos computacionais limitados. Inspirado em arquiteturas como MobileNet e ShuffleNet, o modelo foi treinado com conjuntos de dados públicos (PlantVillage e folhas de milho) e demonstrou desempenho excepcional, alcançando uma acurácia de 99,63%, superando modelos mais complexos como ResNet50 e VGG-16 em precisão, tempo de inferência e uso de memória. A pesquisa destaca a viabilidade prática do GreenDiseaseNet para uso em campo, reforçando sua contribuição para a agricultura de precisão e o monitoramento automatizado de doenças em plantas.

A partir da análise dos trabalhos relacionados, fica evidente que o uso de técnicas de visão computacional e aprendizado profundo tem avançado significativamente no diagnóstico de doenças em folhas de café. Esses estudos demonstram não apenas a eficácia das CNNs e de modelos leves para aplicações em campo, mas também a relevância de métodos capazes de operar em dispositivos com recursos limitados, atendendo às demandas práticas do setor cafeeiro. Com base nesse panorama tecnológico, torna-se fundamental compreender quais são as principais doenças e pragas que afetam o cafeeiro, uma vez que o conhecimento detalhado desses problemas é essencial para orientar tanto o desenvolvimento de soluções computacionais quanto a adoção de estratégias de manejo eficientes. A seguir, são apresentadas as principais enfermidades que impactam a cultura do café e suas implicações para a produtividade.

Tabela 2.1 – Comparação das arquiteturas e resultados dos artigos relacionados

Artigo	**Arquitetura Utilizada**	**Resultado**	**Acurácia**
Türkoğlu e Hanbay (2019)	CNNs (AlexNet, VGG16, VGG19, GoogleNet, ResNet50, ResNet101, InceptionV3, SqueezeNet)	Melhor resultado com ResNet50 + SVM: desempenho eficiente com aprendizado por transferência.	97,86%
Işik e Eskicioglu (2022)	CNNs (VGG-16, ResNet50, MobileNetV2)	Acurácia superior a 95% utilizando CNNs pré-treinadas.	>95%
Novtahaning et al. (2022)	Ensemble (VGG-16, EfficientNetB0, ResNet152)	Acurácia de 97,3% e F1-Score de 95,1%; custo computacional elevado.	97,3%
Aufar et al. (2023)	CNNs (ResNet50, DenseNet169)	Precisão experimental de até 100% em algumas arquiteturas; sem estatísticas de dispersão.	100%
Yamashita e Leite (2023)	MobileNetV2	Modelo leve para execução em dispositivos embarcados; adequado para uso em campo.	98,83%
buhayi e Mossa (2023)	MobileNetV2 + Adam Optimizer	Alta eficiência computacional, bom desempenho com três classes de folhas de café (saudável, cescospora, ferrugem).	96,23%
buhayi e Mossa (2023)	MobileNetV2 + Adam Optimizer	Alta eficiência computacional, bom desempenho com três classes de folhas de café (saudável, cescospora, ferrugem).	96,23%
Signo et al. (2024)	CNN personalizada)	Revisão bibliográfica com implementação prática; classifica 38 tipos de doenças foliares.	94,10% (teste), 98,56% (treino)
Leite et al. (2024)	GreenDiseaseNet (modelo leve custom)	Revisão bibliográfica. Modelo eficiente e compacto, ideal para dispositivos móveis; superou modelos pesados como ResNet50 e VGG-16 em tempo e precisão.	99,63%

Fonte: Autor (2025).

2.3 Principais Pragas do Cafeeiro

Para se manter competitivo em um mercado cada vez mais exigente, o cafeicultor precisa buscar conhecimento para aprimorar a produção e a qualidade do café, de forma que os custos permaneçam compatíveis com a atividade.

A adoção de práticas culturais é essencial e acessível aos produtores, desde que sejam apresentadas alternativas tecnológicas adaptadas às escalas e possibilidades de suas produções.

Entre essas práticas, o controle fitossanitário é uma das mais importantes e deve ser aplicado com critério, por meio de estratégias que combinem o sucesso no combate a pragas e doenças com o desenvolvimento sustentável da atividade cafeeira.

O cafeeiro é vulnerável a diversas doenças, tanto na fase de viveiro quanto no campo. A ocorrência dessas enfermidades é um dos fatores que impactam negativamente a produtividade e a qualidade do café, além de elevar os custos de produção (REIS; CUNHA, 2010).

2.3.1 Ferrugem do Cafeeiro

A ferrugem, causada pelo fungo ***Hemileia vastatrix Berk. & Br.***, é considerada a doença mais importante na cafeicultura devido aos grandes prejuízos que provoca. O fungo afeta diversas cultivares de café, no entanto, há diferenças de patogenicidade dentro do gênero ***Coffea***. A espécie ***Coffea canephora*** possui cultivares resistentes à ferrugem, enquanto a maioria das cultivares comerciais da espécie ***C. arabica*** é suscetível à doença (REIS; CUNHA, 2010).

Os primeiros sinais da ferrugem são pequenas manchas circulares de cor amarelo-alaranjada, com diâmetros que podem chegar a 0,5 cm ou mais, localizadas na face inferior das folhas. Nessas manchas, forma-se uma massa pulverulenta de uredósporos. Em estágios mais avançados, partes do tecido foliar são destruídas e se tornam necrosadas (REIS; CUNHA, 2010).

A doença causa intensa queda das folhas do cafeeiro, resultando em uma redução significativa da produtividade no ano seguinte, o que compromete a qualidade final do café (REIS; CUNHA, 2010).

A ocorrência da ferrugem é influenciada por fatores relacionados ao hospedeiro (cafeeiro), ao patógeno (fungo) e ao ambiente. Elementos como o nível de folhagem, a carga pendente (produção) e a densidade de plantas por área são determinantes, pois afetam o microclima da plantação, influenciando a incidência e a severidade da doença. Esses fatores são cruciais para a definição das estratégias de controle da enfermidade (REIS; CUNHA, 2010). A figura 2.2 ilustra um exemplo de folha de cafeeiro com sintoma de Ferrugem.

Figura 2.2 – Folha de cafeeiros com sintoma de Ferrugem

Fonte: Do Autor

2.3.2 Cercosporiose

A cercosporiose é uma das doenças mais antigas que afetam os cafeeiros. No Brasil, conhecida como "mancha-de-olho-pardo", é causada pelo fungo Cercospora coffeicola Berk. & Cooke, sendo uma das principais doenças que acometem o cafeeiro C. arabica L. na fase de viveiro. As mudas afetadas pela cercosporiose sofrem com desfolha, desenvolvimento reduzido e raquitismo, tornando-as inadequadas para o plantio (REIS; CUNHA, 2010).

Os sintomas característicos da cercosporiose são manchas circulares de coloração que varia do castanho-claro ao escuro, com o centro branco-acinzentado, geralmente cercadas por um halo amarelado. A doença está presente em todas as regiões cafeeiras do Brasil e causa prejuízos tanto na fase de viveiro (mudas) quanto no campo, afetando plantas jovens e adultas (REIS; CUNHA, 2010). A figura 2.3 ilustra um exemplo de folha de cafeeiro com sintoma de Cercosporiose.

Figura 2.3 – Folha de cafeeiros com sintoma de Cercosporiose

Fonte: Do Autor

2.3.3 Bicho-mineiro

O bicho-mineiro (Leucoptera coffeella), também conhecido como minador das folhas do cafeeiro, é uma das principais pragas que afetam a cultura do café, especialmente em regiões de clima quente e com períodos prolongados de estresse hídrico. Essa praga causa danos significativos à produtividade, uma vez que suas larvas se alimentam do tecido foliar, reduzindo a capacidade fotossintética da planta e, consequentemente, a produção de grãos (REIS; CUNHA, 2010).

O adulto do bicho-mineiro é um microlepidóptero cuja mariposa mede cerca de 6,5 mm de envergadura, possui coloração branco-prateada e asas anteriores e posteriores franjadas. As fêmeas depositam seus ovos na superfície superior das folhas, e as pequenas lagartas, ao eclodirem, penetram diretamente no interior do tecido foliar, sem contato com o ambiente externo. As larvas vivem dentro de galerias ou "minas"que elas mesmas cavam, alimentando-se do parênquima foliar. Ao completarem seu desenvolvimento, atingem aproximadamente 3,5 mm de comprimento. Em seguida, abandonam as folhas pela parte superior das minas e, com o auxílio de um fio de seda por elas produzido, descem até as folhas mais baixeiras da planta para empupar em casulos, onde se transformam em pupas antes de emergirem como adultos (REIS; CUNHA, 2010).

As lesões causadas pelo bicho-mineiro nas folhas são inconfundíveis: apresentam o centro mais escuro devido ao acúmulo de excrementos das larvas, enquanto o contorno geralmente assume um formato arredondado. A epiderme superior da folha, na região afetada, solta-se com facilidade quando pressionada. Essas lesões são encontradas, em sua maioria, no terço superior da planta, pois essa área é mais arejada e favorece o desenvolvimento da praga. Em infestações severas, as folhas podem cair prematuramente, enfraquecendo a planta e reduzindo sua capacidade produtiva (REIS; CUNHA, 2010).

Além dos danos diretos, o bicho-mineiro também pode facilitar a entrada de patógenos, como fungos e bactérias, que se aproveitam das lesões para infectar o cafeeiro. O controle dessa praga envolve monitoramento constante, manejo integrado com uso de inseticidas seletivos, conservação de inimigos naturais (como vespas parasitoides) e práticas culturais, como adubação equilibrada e irrigação adequada, que reduzem o estresse das plantas e as tornam menos suscetíveis ao ataque (REIS; CUNHA, 2010). A figura 2.4 ilustra um exemplo de folha de cafeeiro com minas do bicho-mineiro.

Figura 2.4 – Folha de cafeeiros com Bicho Mineiro

Fonte: Do Autor

A figura 2.5 representa a complexidade que se pode ter em uma unica imagems, contendo todos os objetos a ser classificados em uma unica folha.

Figura 2.5 – Folha de cafeeiros com todos os sintomas

Fonte: Do Autor

3 MATERIAL E MÉTODOS

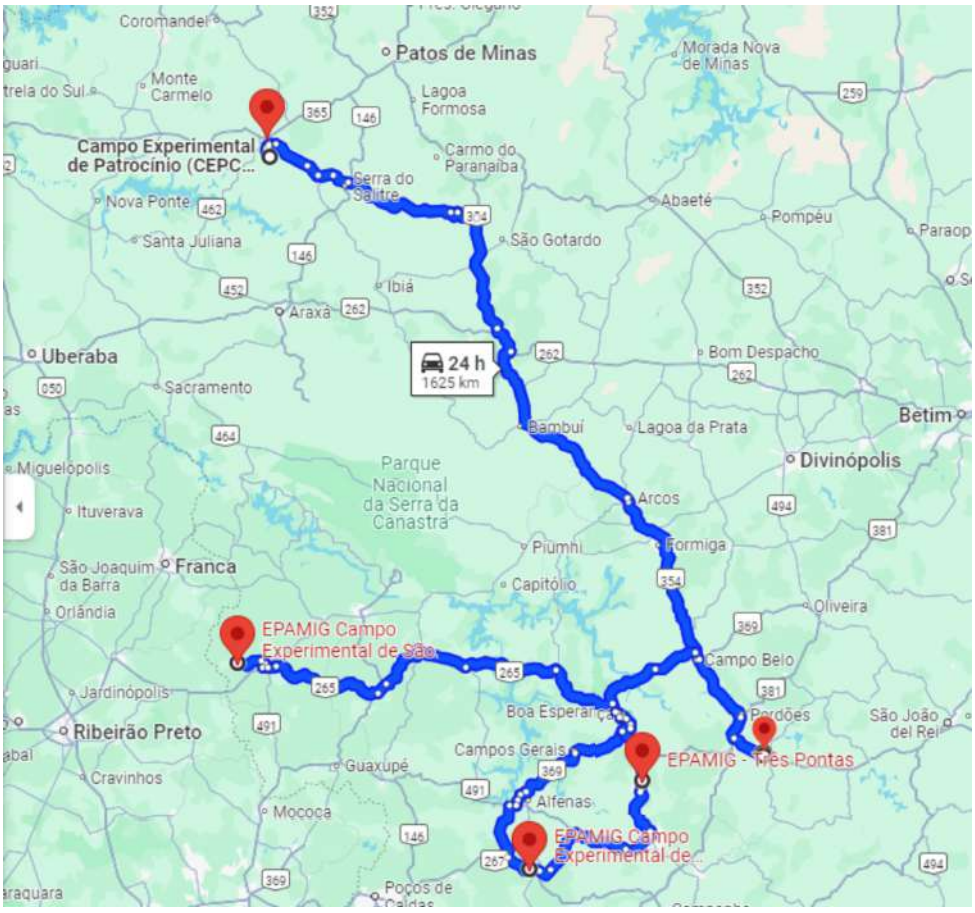
3.1 Banco de Dados

3.1.1 Descrição da Coleta de Fotos em Campo

Em um primeiro momento, foram coletadas folhas com doenças e pragas relacionados ao trabalho nos campos de estudos da EPAMIG, nos municípios de Patrocínio, São Sebastião do Paraíso, Machado, Três Pontas e Lavras.

As fotos foram coletadas através de aparelhos *smartphone* modelo Xiaomi Redmi Note 8 e Xiaomi Redmi Note 13c, ambos capturaram fotos distintas entre si, não havendo repetição de nenhuma folha.

Figura 3.1 – Trajetória Percorrida e Local de Coleta



Fonte: Adaptado (GOOGLE, 2024)

Os aparelho possuem câmeras fotográficas acopladas, respectivamente de 48 MP, 8 MP e 2 MP. Foi utilizado o recurso Modo Retrato, do qual cria um efeito *bokeh* (desfoque) para destacar a folha do cafeeiro.

Foram coletadas, em campo, imagens de folhas que apresentavam sintomas de ferrugem, cercosporiose e bicho-mineiro, registradas tanto sob iluminação natural quanto em ambientes internos com luz artificial.

Uma folha de café após ser retirada da planta, tendem a perder suas características naturais em questão de dias, dependendo das condições em que é armazenada. Se mantida em

condições de alta umidade e temperaturas baixas, a folha pode manter boa parte de sua integridade visual e coloração por até uma semana, mas ainda assim começará a secar e mudar de aparência (TAIZ et al., 2017).

Em condições normais de temperatura ambiente e sem o controle de umidade, uma folha de café começa a mostrar sinais de desidratação e mudança de coloração dentro de 1 a 3 dias. A perda de água provoca a secagem e o surgimento de manchas amarronzadas, o que dificulta a identificação de características visuais como padrões de lesão ou coloração para análise (TAIZ et al., 2017).

Como as condições descritas anteriormente são inviáveis, a captura das imagens foram feitas no mesmo dia.

Por padrão, as resoluções das fotos são de 3000 x 4000 pixels, totalizando 12 megapixels (MP), sendo em media 4 megabytes (MB) de tamanho.

Não houve padronização quanto a luminosidade e enquadramento das fotos, tanto em campo aberto quanto as folhas recolhidas para serem capturadas por foto posteriormente em ambiente fechado, pois pretendia-se realizar o treinamento do sistema com as condições normais no campo através de um *smartphone* para esta aplicação em campo.

3.1.2 Montagem do banco de dados

Inicialmente, foi constituido um banco de dados de imagens de doenças e pragas do cafeeiro a partir de duas fontes principais:

- (i) imagens coletadas do site Roboflow (DWYER; AL., 2025), utilizadas para o treinamento inicial dos modelos.
- (ii) imagens obtidas em campo, capturadas diretamente em plantações de café sob diferentes condições de iluminação e estágio de desenvolvimento foliar.

Todos os bancos de dados utilizados neste estudo passaram por processos de ***data augmentation***, técnica essencial para aumentar a variabilidade das amostras e melhorar a capacidade de generalização do modelo. Essa estratégia foi aplicada de forma sistemática para simular diferentes condições de captura das imagens e ampliar o conjunto de dados disponível para o treinamento. Além disso, em diversas imagens, é possível observar mais de uma classe presente na mesma fotografia, o que torna o processo de classificação mais desafiador, exigindo que o modelo aprenda a identificar múltiplos padrões simultaneamente e classifique de forma correta a mesma imagem.

Para garantir a robustez e a generalização dos modelos propostos, o conjunto total de imagens foi organizado em três grupos (A, B, C).

3.1.3 Banco de dados de treinamento (A):

O banco de dados de treinamento é formado majoritariamente pelas imagens provenientes do Roboflow, utilizadas no ajuste inicial dos parâmetros dos modelos. Contém ao total 6438 imagens. As imagens, foram divididas em duas pastas, denominadas TRAIN e VALID.

A pasta denominado TRAIN, foi formado por três classes, sendo cercospora (folhas com cercosporiose), *leaf-miner* (Folhas com bicho-mineiro) e leaf-rust (folhas com ferrugem). Do mesmo modo, o banco de dados denominado VALID é formado por três classes, sendo cercospora (folhas com cercosporiose), *leaf-miner* (Folhas com bicho-mineiro) e leaf-rust (folhas com ferrugem). A quantidade de cada imagens por cada classe e por cada pasta se dá por:

Pasta TRAIN:

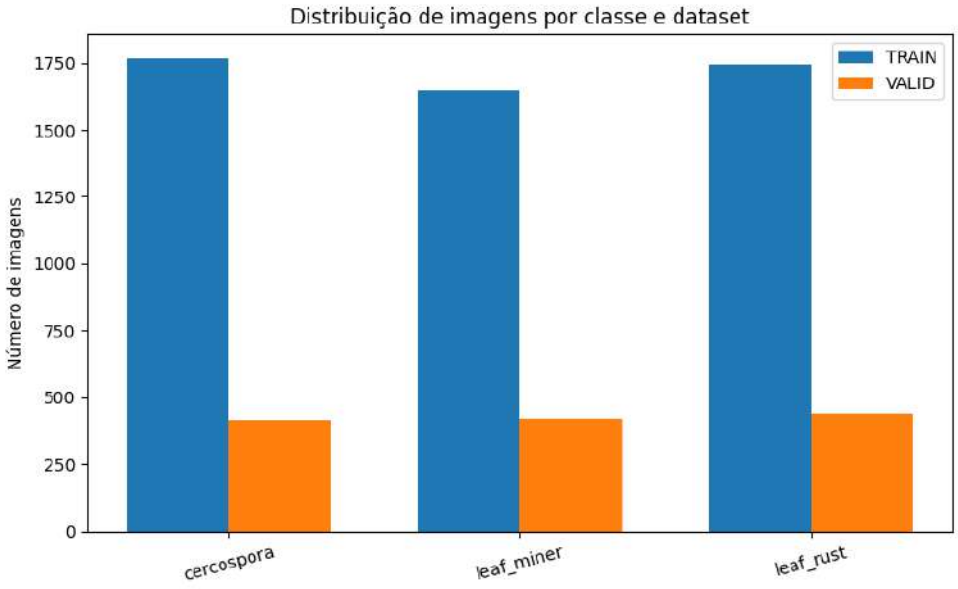
- Folhas com cercosporiose: 1768 imagens;
- Folhas com bicho mineiro: 1647 imagens;
- Folhas com ferrugem: 1744 imagens.

Pasta VALID:

- Folhas com cercosporiose: 419 imagens;
- Folhas com bicho mineiro: 420 imagens;
- Folhas com ferrugem: 440 imagens.

A figura 3.2 mostra como foi distribuído o número de fotos para treinamento e validação nas modelagens.

Figura 3.2 – Distribuição de imagens por Classe e pasta TRAIN e VALID do banco de dados A.



Fonte: do Autor

3.1.4 Banco de dados para validação cruzada K-Fold (B)

Composto por uma junção de amostras provenientes tanto do Roboflow quanto da coleta de imagens em campo, esse conjunto foi utilizado para analisar a estabilidade, a robustez e a consistência dos modelos sob diferentes particionamentos dos dados. Para esse fim, empregou-se um procedimento de validação cruzada K-Fold estratificada, garantindo que a proporção das classes fosse preservada em cada divisão e que o desempenho avaliado refletisse de forma mais fidedigna o comportamento do modelo diante de distribuições reais. No total, o conjunto é constituído por 8798 imagens.

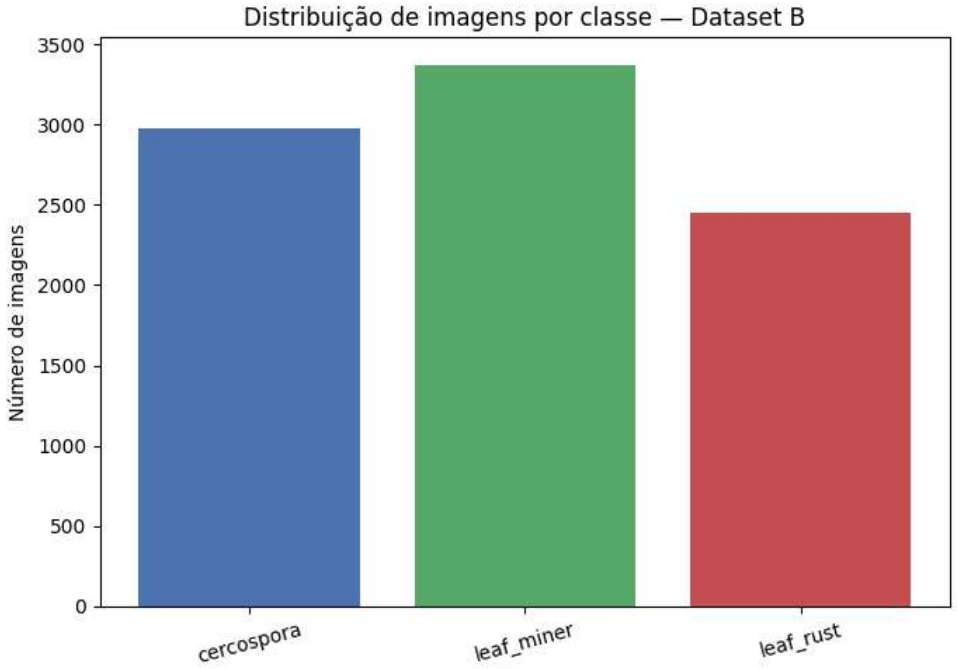
Para a composição deste banco de dados para a validação cruzada k-fold, foram coletados 6438 da pasta de treinamento e incluidas 2360 novas fotos,sendo a quantidade de cada imagens por cada classe dada por:

- Folhas com cercosporiose: 2975 imagens;

- Folhas com bicho mineiro: 3373 imagens;
- Folhas com ferrugem: 2450 imagens.

A figura 3.3 mostra como foi distribuído o numero de imagens de validação cruzada K-Fold do banco de dados B.

Figura 3.3 – Distribuição de imagens do banco de dados B



Fonte: do Autor

3.1.5 Banco de dados de Validação (C)

Constituído exclusivamente por imagens coletadas em campo, utilizadas para testar o desempenho real dos modelos em condições não vistas durante o treinamento. Contém ao total 6588 imagens.

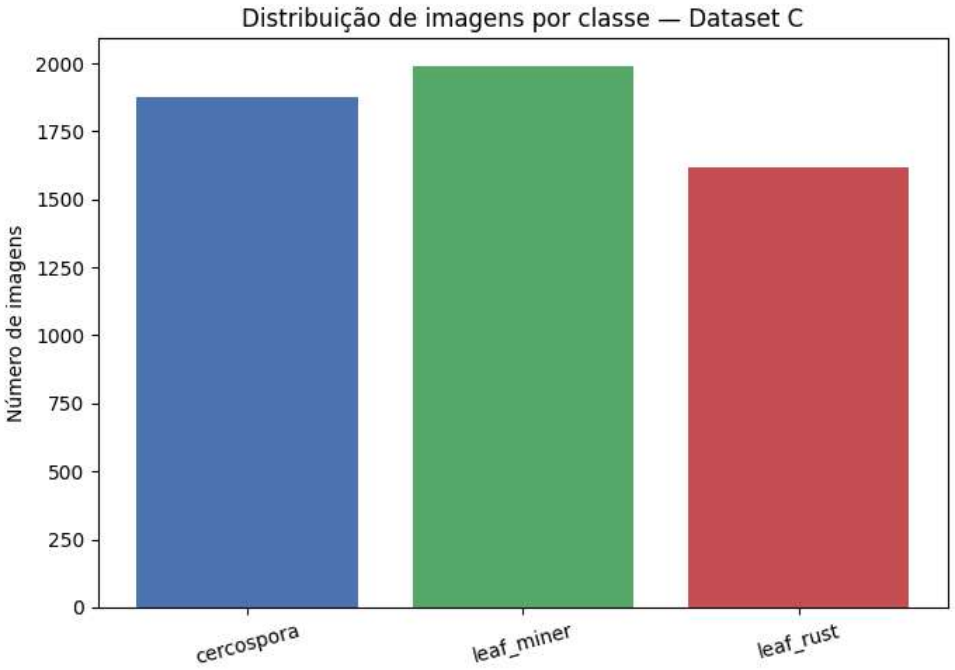
Essa abordagem possibilitou uma avaliação mais abrangente da capacidade de generalização do modelo, considerando tanto dados padronizados quanto imagens reais de campo, com maior variabilidade de fatores ambientais e visuais.

Para a composição deste banco de dados de validação , foram coletados 6588, sendo a quantidade de cada imagens por cada classe dada por:

- Folhas com cercosporiose: 1875 imagens;
- Folhas com bicho mineiro: 1992 imagens;
- Folhas com ferrugem: 1618 imagens.

A figura 3.4 mostra como foi distribuído o numero de fotos de validação do banco de dadoos C.

Figura 3.4 – Distribuição de imagens do banco de dados C.



Fonte: do Autor

Outro aspecto importante considerado foi o equilíbrio no número de amostras por classe, garantindo que nenhuma categoria fosse sub-representada e evitando vieses que pudessem comprometer a capacidade da rede neural de aprender e distinguir adequadamente cada tipo de doença e praga. Por fim, o conjunto de dados não passou por padronização nas condições de captura das fotos, apresentando variações significativas de incidência de luz, fundo da imagem, brilho e contraste. Essa ausência de uniformidade reflete a diversidade de situações encontradas em ambientes reais, contribuindo para o desenvolvimento de um modelo mais robusto e adaptável.

A seguir nas figuras 3.5, 3.6, 3.7, 3.8, 3.9, 3.10 são dados os exemplos de variabilidade nas imagens contidas em todos os bancos de dados (A, B e C).

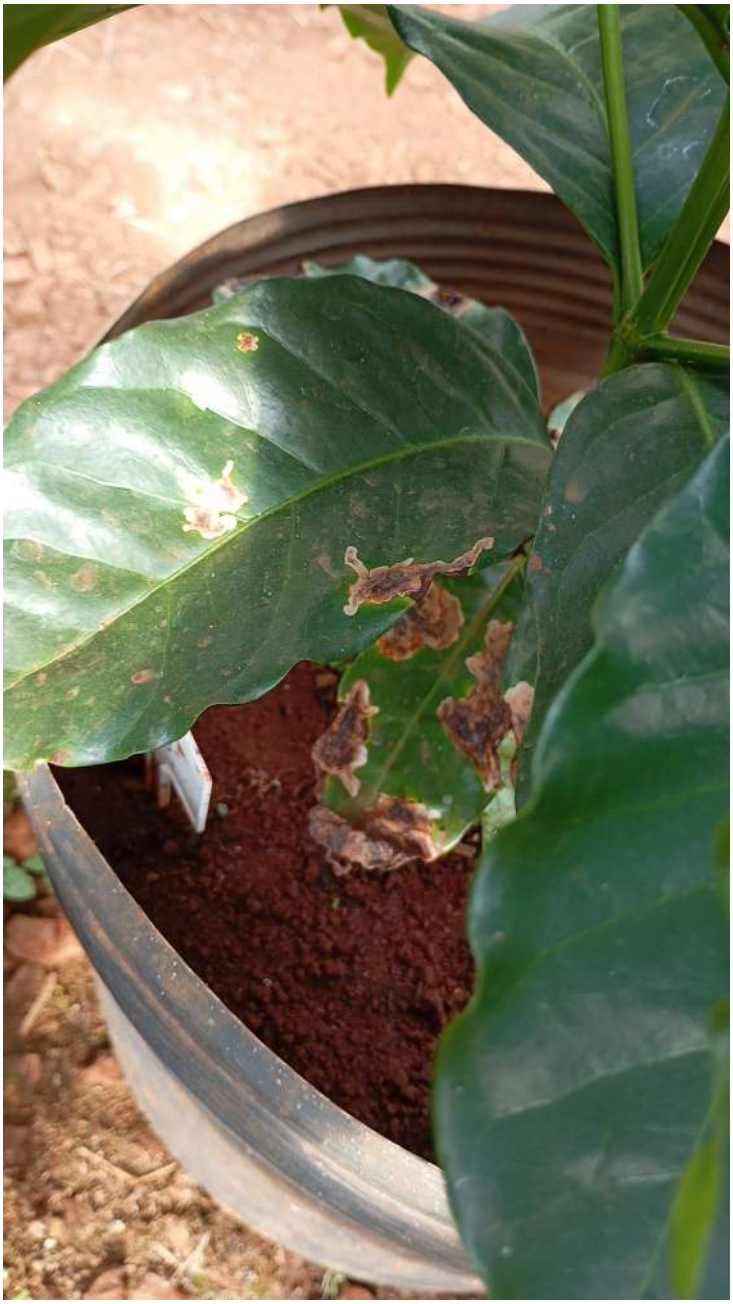
Figura 3.5 – Exemplo de imagem da folha em campo com tratamento de cor e contraste.

Fonte: (DWYER; AL., 2025)

Figura 3.6 – Exemplo de imagem da folha destacada da planta e com fundo padronizado

Fonte: (DWYER; AL., 2025)

Figura 3.7 – Exemplo de imagem da folha coletada do viveiro.



Fonte: do Autor

Figura 3.8 – Exemplo de imagem coletada em campo com fundo padronizado

Fonte: (DWYER; AL., 2025)

Figura 3.9 – Exemplo de imagem da folha destacada da planta sem fundo padronizado.

Fonte: do Autor

Figura 3.10 – Exemplo de imagem da folha em campo sem tratamento de cor, contraste e sombra

Fonte: do Autor

3.2 Modelo de Classificação

No decorrer deste estudo, foram empregados quatro redes convolucionais (DenseNet, ResNet e EfficientNet) e duas técnicas de redução de dimensionalidade (PCA e MiniRocket) para realizar a classificação das doenças e praga nas imagens analisadas. A metodologia foi estruturada em quatro etapas principais, cada uma desempenhando um papel específico na construção de um sistema eficiente, escalável e de baixo custo computacional.

3.2.1 Treinamento inicial utilizando DenseNet e ResNet.

O processo se inicia com o treinamento supervisionado das arquiteturas DenseNet e ResNet, selecionadas para o modelos de referência e permitindo estabelecer um desempenho-base para o problema. Ambas as arquiteturas são amplamente reconhecidas por sua eficiência na extração hierárquica de características visuais, o que contribui para a identificação precisa dos padrões associados às doenças foliares.

3.2.2 Transfer learning para a EfficientNet como estratégia de redução de custo computacional.

Com base na etapa anterior, empregou-se *transfer learning* para adaptar a EfficientNet ao domínio do conjunto de dados utilizado, pois apresenta melhor relação entre desempenho e custo computacional, devido ao seu escalonamento composto que equilibra profundidade, largura e resolução. Assim, ao inicializar a EfficientNet com pesos derivados do treinamento prévio, o sistema se beneficia de uma rede mais leve e eficiente, mantendo elevada capacidade discriminativa. Essa abordagem reduz significativamente o tempo de treinamento e o uso de recursos computacionais, sem comprometer a qualidade das representações geradas.

3.2.3 Uso de PCA e MiniRocket como etapas de pré-processamento e extração de características para reduzir o custo computacional.

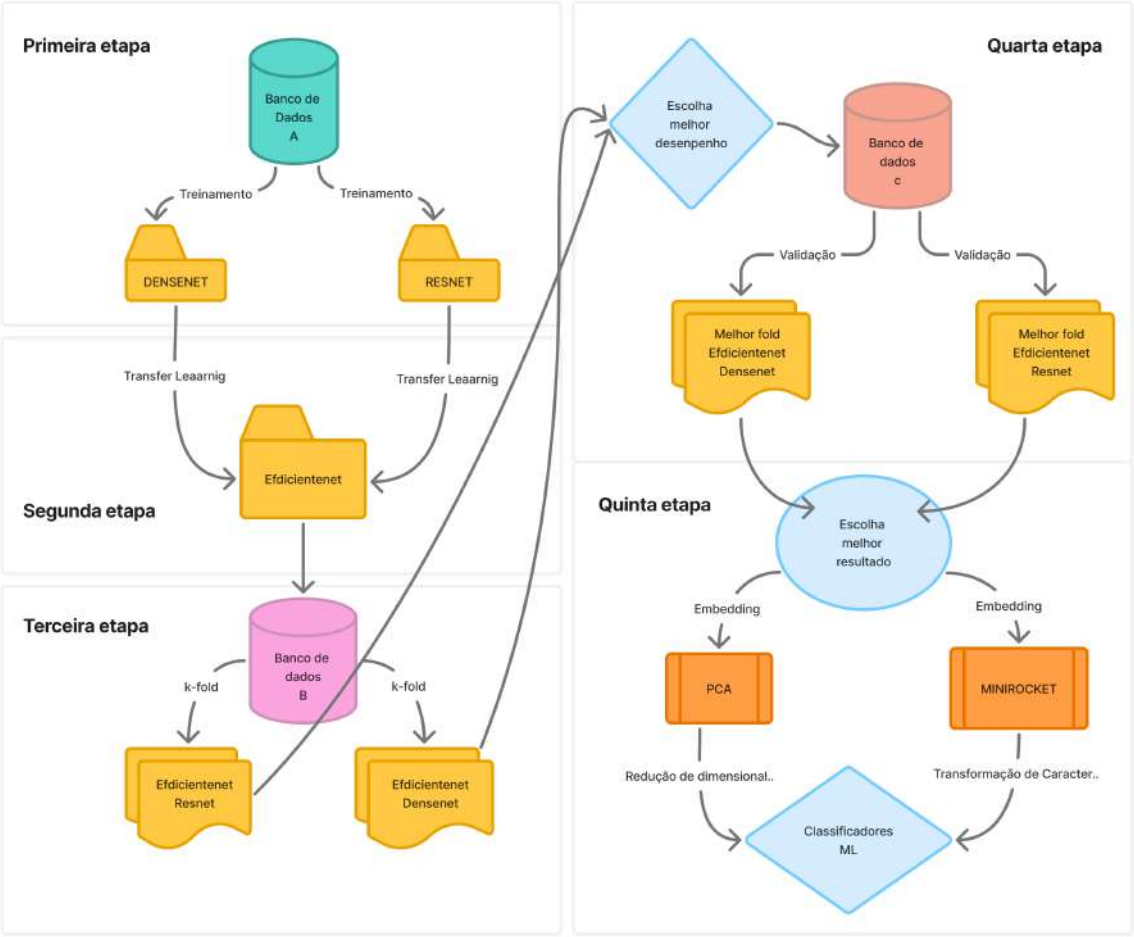
Após o refinamento das redes convolucionais, aplicaram-se duas técnicas de extração e compactação de características: o PCA (Principal Component Analysis) e o MiniRocket. O PCA foi utilizado para gerar representações lineares reduzidas, preservando a variância essencial das imagens e minimizando redundâncias nos dados. O MiniRocket, por sua vez, permitiu extrair características discriminativas de maneira extremamente rápida e eficiente, ao utilizar convoluções determinísticas e estatísticas simples, como a *Proportion of Positive Values* (PPV). Ambas as técnicas atuam como formas de pré-processamento voltadas à diminuição da dimensionalidade e do custo computacional, preparando as representações visuais para classificadores subsequentes.

3.2.4 Emprego de modelos de Deep Learning sobre as *features* extraídas para avaliar o ganho de resolução computacional sem perda de desempenho.

As características geradas pelo PCA e pelo MiniRocket foram então utilizadas como entrada para modelos de *deep learning* e classificadores avançados, com o objetivo de verificar se representações mais compactas permitiriam manter — ou até melhorar — a qualidade das classificações. Essa etapa teve como finalidade confirmar se a redução computacional obtida nas etapas anteriores não comprometeu a capacidade preditiva do sistema.

Em conjunto, essas quatro etapas resultaram em um sistema robusto, eficiente e escalável, capaz de classificar as doenças cercosporiose, ferrugem e a praga bicho-mineiro em folhas do cafeeiro com alta precisão e baixo custo computacional. A figura 3.11 mostra o diagrama de todo o processo descrito no capitulo 3.2.

Figura 3.11 – Diagrama de treinamento do sistema



Fonte: do Autor

3.2.5 Arquiteturas de Rede Neural Convolucional

3.2.5.1 Residual Network de 50 camadas

A ResNet50 (Residual Network de 50 camadas) é uma arquitetura de rede neural convolucional proposta por He et al. (2016), que revolucionou o aprendizado profundo ao introduzir o conceito de conexões residuais (*skip connections*). Esse conceito foi criado para resolver um problema comum em redes muito profundas, chamado de o desvanecimento do gradiente (*vanishing gradient*), que dificultava o treinamento à medida que o número de camadas aumentava.

A ideia central da ResNet é permitir que algumas camadas aprendam funções residuais, ou seja, em vez de aprender diretamente a saída desejada $H(x)$, elas aprendem a diferença $F(x) = H(x) - x$. Dessa forma, a saída de um bloco é expressa como $y = F(x) + x$, onde o termo x é a entrada original somada novamente à saída do conjunto de camadas. Essa adição direta da entrada à saída cria um caminho alternativo para o fluxo do gradiente durante o treinamento, facilitando o aprendizado e permitindo o uso de redes com dezenas ou até centenas de camadas sem perda significativa de desempenho.

A arquitetura ResNet50 é composta por 50 camadas profundas, incluindo convoluções, camadas de normalização em lote (Batch Normalization) e funções de ativação *ReLU*. Sua estrutura é organizada em blocos residuais, sendo dois tipos principais: os blocos simples (*basic blocks*) e os blocos de gargalo (*bottleneck blocks*). A versão de 50 camadas utiliza os blocos de gargalo, que têm três camadas convolucionais:

- uma convolução 1×1 para reduzir a dimensionalidade,
- uma convolução 3×3 para processar as características,
- e outra convolução 1×1 para restaurar a dimensionalidade original.

Essa configuração reduz o custo computacional e mantém a profundidade e a expressividade da rede.

A ResNet50 começa com uma camada convolucional inicial (7×7), seguida por uma camada de *max pooling* para redução da resolução espacial. Depois disso, há quatro estágios principais de blocos residuais, geralmente com 3, 4, 6 e 3 blocos, respectivamente, onde extraem e refinam as características da imagem em diferentes níveis de abstração. Por fim, a rede termina com uma camada de *pooling global* e uma camada totalmente conectada (*fully connected*) responsável pela classificação final.

Graças às conexões residuais e à eficiência do design em blocos de gargalo, a ResNet50 é capaz de alcançar grande profundidade com excelente desempenho e estabilidade de treinamento, sendo uma das arquiteturas mais utilizadas em visão computacional. Ela é amplamente empregada em tarefas de classificação de imagens, extração de características, detecção de objetos e transferência de aprendizado, servindo como base para diversas outras arquiteturas modernas.

3.2.5.2 Densely Connected Convolutional Network

A DenseNet (Densely Connected Convolutional Network) é uma arquitetura de rede neural convolucional proposta por Huang et al. (2017), que se destaca por estabelecer conexões diretas entre todas as camadas de uma mesma parte da rede. Em vez de cada camada receber como entrada apenas a saída da camada anterior, como ocorre em redes tradicionais, cada camada em uma DenseNet recebe como entrada os mapas de características (*feature maps*) de todas as camadas anteriores dentro do mesmo bloco denso (*dense block*). Essa estratégia cria conexões densas entre as camadas, promovendo um forte reaproveitamento de informações e gradientes, o que melhora a eficiência e a capacidade de aprendizado da rede.

Na prática, isso significa que, se houver L camadas em um bloco, haverá $\frac{L(L+1)}{2}$ conexões diretas entre elas. Cada camada produz um número fixo de novos mapas de características, chamado de taxa de crescimento (*growth rate*), e esses novos mapas são concatenados com todos os anteriores para formar a entrada da próxima camada. Esse mecanismo reduz o problema de desvanecimento do gradiente (*vanishing gradient*), pois o fluxo de informação e gradiente é facilitado entre camadas distantes.

A arquitetura da DenseNet é organizada em blocos densos intercalados por camadas de transição (*transition layers*). Dentro de um bloco denso, ocorre a concatenação das saídas, enquanto as camadas de transição, compostas geralmente por uma convolução 1x1 seguida de *pooling* 2x2, têm a função de reduzir a dimensionalidade e controlar a complexidade computacional da rede. Essa combinação de conexões densas e compressão gradual dos dados torna a DenseNet uma rede profunda, porém eficiente, utilizando menos parâmetros do que arquiteturas tradicionais de profundidade equivalente, como ResNet.

Além disso, a DenseNet apresenta uma excelente eficiência de uso de parâmetros, pois evita a redundância na extração de características á camadas posteriores. Essa característica também favorece a generalização e reduz a necessidade de grandes quantidades de dados para treinamento.

De forma geral, a DenseNet é uma arquitetura que alia profundidade, eficiência e estabilidade de treinamento, sendo amplamente utilizada em tarefas de classificação de imagens, segmentação, e extração de características (*embeddings*) em aplicações de visão computacional.

3.2.5.3 EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks

A EfficientNet é uma arquitetura de rede neural convolucional proposta por Tan e Le (2019), desenvolvida com o objetivo de alcançar alto desempenho com eficiência computacional, ou seja, obter grande precisão com menor número de parâmetros e operações. Ela se destaca por introduzir uma forma sistemática e balanceada de escalonamento da rede ao invés de simplesmente aumentar arbitrariamente a profundidade, a largura ou a resolução das imagens de entrada, a EfficientNet aplica um método composto de escalonamento (compound scaling), que ajusta essas três dimensões de maneira conjunta e otimizada.

O ponto de partida da arquitetura é o modelo EfficientNet-B0, obtido por meio de um processo automatizado de busca de arquitetura neural (Neural Architecture Search – NAS) utilizando a técnica MnasNet, desenvolvida pelo Google AI. A partir desse modelo base, são geradas as versões EfficientNet-B1 a B7, cada uma maior e mais precisa que a anterior, conforme o fator de escala aplicado. O método de compound scaling utiliza um conjunto de coeficientes ϕ, α, β e γ, que controlam respectivamente o aumento da profundidade, largura e resolução, de forma que o custo computacional total (medido em FLOPs) cresça de maneira equilibrada. Em outras palavras, ao dobrar o número de recursos disponíveis, a rede é escalada de modo a aproveitar esse poder adicional de maneira otimizada, evitando desperdício de processamento.

A arquitetura interna da EfficientNet é baseada em blocos convolucionais chamados MBConv (Mobile Inverted Bottleneck Convolution), originalmente introduzidos no MobileNetV2. Esses blocos combinam convoluções 1×1 e 3×3, camadas de normalização, funções de ativação Swish (ou SiLU) e conexões residuais, o que permite alta eficiência e reutilização de características. Além disso, a EfficientNet utiliza a técnica Squeeze-and-Excitation (SE), que recalibra dinamicamente a importância dos canais de cada mapa de características, permitindo que a rede aprenda quais informações são mais relevantes.

Uma característica marcante da EfficientNet é sua eficiência em termos de número de parâmetros e FLOPs: comparada a arquiteturas tradicionais como ResNet ou Inception, ela alcança maior acurácia com um número significativamente menor de operações. Por exemplo, a EfficientNet-B7, uma das versões maiores, supera a ResNet-152 em precisão no ImageNet, utilizando cerca de 8 vezes menos parâmetros e 10 vezes menos computação.

A EfficientNet representa um avanço importante no design de redes convolucionais, pois combina otimização automática de arquitetura, blocos convolucionais eficientes e escalonamento composto balanceado, resultando em uma família de modelos altamente precisos e econômicos em termos de recursos. Essa eficiência faz com que a EfficientNet seja amplamente utilizada em aplicações de classificação de imagens, detecção de objetos e aprendizado por transferência, especialmente em contextos onde o desempenho computacional é crítico.

3.2.6 Redução de Dimensionalidade

3.2.6.1 Análise de Componentes Principais

A PCA (Principal Component Analysis) é uma das técnicas estatísticas mais utilizadas para redução de dimensionalidade e extração de características em conjuntos de dados multivariados. Proposta originalmente por Pearson (1901) e posteriormente generalizada por Hotelling (1933), a PCA tem como objetivo transformar um conjunto de variáveis possivelmente correlacionadas em um novo conjunto de variáveis não correlacionadas, chamadas de componentes principais (PEARSON, 1901; HOTELLING, 1933)

Esses componentes principais são combinações lineares das variáveis originais e são ordenados de modo que o primeiro componente retém a maior quantidade possível de variância dos dados, o segundo retém a maior variância remanescente, e assim por diante. O processo é baseado na decomposição espectral da matriz de covariância ou, de forma equivalente, na decomposição em valores singulares (SVD — Singular Value Decomposition), o que permite identificar as direções no espaço de atributos onde os dados apresentam maior dispersão (JOLLIFFE; CADIMA, 2016).

A principal vantagem do PCA é sua capacidade de reduzir a dimensionalidade dos dados sem perder informações relevantes. Isso é especialmente importante em cenários com grandes volumes de variáveis (alta dimensionalidade), nos quais o ruído ou a redundância podem prejudicar o desempenho de algoritmos de aprendizado de máquina. Ao projetar os dados em um espaço de menor dimensão, preservando a variância mais significativa, a PCA melhora tanto a eficiência computacional quanto a capacidade de generalização dos modelos.

Em aplicações práticas de visão computacional e aprendizado profundo, a PCA é frequentemente usada após a extração de características por redes neurais, servindo para reduzir o tamanho dos vetores de *embedding* e facilitar o uso de classificadores tradicionais, como o Support Vector Machine (SVM) ou o XGBoost (ABDI; WILLIAMS, 2010). Além disso, ela é amplamente utilizada para visualização de dados em espaços bidimensionais ou tridimensionais, permitindo interpretar melhor as relações entre amostras em espaços originalmente de alta dimensão.

Do ponto de vista matemático, o PCA busca os autovetores da matriz de covariância dos dados. Cada autovetor corresponde a um eixo de máxima variância, e os autovalores associados indicam a quantidade de variância explicada por cada componente. Assim, o número de componentes principais escolhidos depende da proporção de variância acumulada que se deseja preservar (frequentemente, 90% ou 95% da variância total) (JOLLIFFE; CADIMA, 2016).

Em síntese, a PCA é uma técnica fundamental no campo da análise de dados e aprendizado de máquina, permitindo compressão, visualização e pré-processamento eficiente, e servindo de base para diversas abordagens modernas de representação e redução de dimensionalidade.

3.2.6.2 Random Convolutional Kernel Transform e MiniRocket

O Random Convolutional Kernel Transform (ROCKET) é uma técnica recente desenvolvida para classificação de séries temporais que combina a simplicidade de modelos lineares com o poder expressivo de transformações convolucionais. Proposta por Dempster et al. (2020), a abordagem aplica um grande número de kernels convolucionais gerados aleatoriamente sobre as séries temporais originais, extraindo estatísticas simples, como o proporção de valores positivos (PPV) e o máximo resultante de cada convolução. Essas estatísticas formam um vetor de características de alta dimensionalidade, que é então utilizado como entrada para um classifica-

dor linear, geralmente uma regressão ridge. A principal vantagem do ROCKET é sua eficiência computacional e excelente desempenho preditivo, comparável a métodos muito mais complexos, como redes neurais profundas ou ensembles de árvores, mas com custo de treinamento significativamente menor (DEMPSTER et al., 2020).

Apesar de seu sucesso, o ROCKET original apresenta alto custo de tempo e memória devido ao grande número de kernels necessários para atingir bom desempenho (geralmente 10.000 ou mais). Para resolver essa limitação, Dempster et al. (2021) propuseram o MiniRocket, uma versão otimizada e determinística do método. O MiniRocket reduz drasticamente o custo computacional ao fixar um conjunto pequeno e padronizado de kernels convolucionais e ao eliminar a aleatoriedade na geração dos parâmetros. Em vez de depender de milhares de combinações aleatórias, o MiniRocket utiliza um número reduzido de dilatações e pesos convolucionais cuidadosamente definidos para capturar padrões temporais relevantes. Essa abordagem mantém a acurácia do ROCKET original, mas com velocidade até 75 vezes superior e menor variabilidade entre execuções (DEMPSTER et al., 2021).

Do ponto de vista metodológico, o MiniRocket introduz uma forma extremamente eficiente de pré-processamento e extração de características para séries temporais. O método aplica convoluções 1D com kernels fixos de valores binários (tipicamente no intervalo $[-1, +1]$) combinados a diferentes dilatações, permitindo captar padrões em múltiplas escalas temporais sem a necessidade de normalização explícita ou parametrização adicional. Em seguida, cada mapa de ativação gerado pelas convoluções é resumido por meio da *of Positive Values* (PPV), métrica que quantifica a fração de ativações maiores que zero. Esses valores funcionam como descritores estatísticos altamente discriminativos, formando vetores de características compactos, estáveis e adequados para modelos lineares. A PPV, além de computacionalmente simples, é robusta a variações de escala e amplitude, o que contribui para a generalização do método (DEMPSTER et al., 2021; RUIZ et al., 2021).

No contexto deste trabalho, o MiniRocket não é aplicado diretamente sobre séries temporais clássicas, mas sim sobre vetores de embeddings extraídos por uma rede convolucional profunda. Esses embeddings, que representam semanticamente cada imagem, são reinterpretados como séries unidimensionais artificiais, permitindo que o MiniRocket atue como um transformador de características sobre representações latentes. Dessa forma, o método explora padrões estruturais ao longo da dimensão do embedding, funcionando como um mecanismo eficiente de projeção não linear antes da classificação.

3.3 Parâmetros de Avaliação do Desempenho

3.3.1 *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) e *False Negative* (FN)

Os parâmetros de desempenho *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) e *False Negative* (FN) são métricas essenciais para avaliar a eficácia de um modelo em tarefas de classificação e detecção de objetos. Esses parâmetros ajudam a entender como o modelo está se comportando em termos de acertos e erros, fornecendo uma base sólida para o cálculo de métricas mais complexas, como precisão, recall e F1-score. A seguir, detalham-se esses parâmetros:

- ***True Positive (TP)* - Verdadeiro Positivo:** Representa os casos em que o modelo identificou corretamente a área de infecção de ferrugem do cafeeiro. Ou seja, é o número de instâncias positivas que foram corretamente previstas pelo modelo. Um resultado é considerado TP quando a *interseção sobre união (IoU)* entre a predição e a anotação real é superior a um limiar pré definido, indicando uma detecção correta.

- ***False Positive (FP)* - Falso Positivo:** Ocorre quando o modelo detecta incorretamente uma área de infecção da ferrugem do cafeeiro, na realidade, não existe nenhuma doença ou é um outro patógeno . Isso significa que o modelo fez uma detecção equivocada, identificando algo que não é uma infecção de ferrugem do cafeeiro como se assim o fosse. Um resultado é classificado como FP quando a *IoU* é inferior ao limiar definido, indicando uma detecção incorreta.

- ***True Negative (TN)* - Verdadeiro Negativo:** Refere-se aos casos em que o modelo corretamente não detecta uma a área de infecção de ferrugem do cafeeiro, e de fato, não há nenhuma ferrugem do cafeeiro na área em questão. Isso demonstra que o modelo acertou ao não detectar uma ferrugem do cafeeiro onde ela não está presente. No entanto, esse parâmetro geralmente não se aplica diretamente em tarefas de detecção de objetos, pois o foco está em identificar a presença do objeto, e não na ausência.

- ***False Negative (FN)* - Falso Negativo:** Ocorre quando o modelo falha em detectar uma ferrugem do cafeeiro que está presente na imagem. Isso significa que o modelo cometeu um erro ao não identificar uma área de ferrugem do cafeeiro que realmente existe no local. Nenhuma predição é gerada para essa marcação, o que indica uma falha na detecção.

Esses parâmetros são fundamentais para calcular métricas como precisão, recall, especificidade e F1-score, que oferecem uma visão abrangente do desempenho do modelo na tarefa específica de segmentação de área da ferrugem do cafeeiro. Essas métricas permitem avaliar o equilíbrio entre a capacidade do modelo de identificar corretamente as palmeiras e a minimização de erros, garantindo que ele seja eficaz na aplicação prática.

3.3.2 Acurácia

A métrica accuracy representa a proporção total de previsões corretas feitas pelo modelo em relação a todas as previsões realizadas. Em outras palavras, ela indica o quanto o modelo acertou no conjunto completo de amostras, considerando tanto as classificações positivas quanto as negativas.

Trata-se de uma métrica global, que mede o desempenho geral do classificador, avaliando quantas instâncias foram rotuladas corretamente independentemente da classe.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Onde:

- **TP (True Positives):** Casos corretamente classificados como positivos
- **TN (True Negatives):** Casos corretamente classificados como negativos.
- **FP (False Positives):** Casos incorretamente classificados como positivos.
- **FP (False Positives):**: Casos incorretamente classificados como negativos.

A acurácia é especialmente útil quando o conjunto de dados possui classes balanceadas, pois reflete diretamente o percentual de acertos do modelo. Entretanto, em cenários com forte desbalanceamento entre classes, sua interpretação pode se tornar limitada, já que um modelo pode apresentar alta acurácia mesmo ignorando completamente uma classe minoritária.

3.3.3 Precision

O parâmetro *precision* é uma métrica que avalia a proporção de instâncias positivas identificadas corretamente em relação ao total de instâncias positivas previstas pelo modelo. Isso significa que a precisão mede a qualidade das previsões positivas do modelo, indicando a proporção de casos positivos que são verdadeiramente positivos entre todas as previsões positivas

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3.1}$$

Onde:

- **TP (True Positives):** Casos que foram corretamente classificados como positivos.
- **FP (False Positives):** Casos que foram incorretamente classificados como positivos.

3.3.4 Recal

A equação do *Recall* (ou Sensibilidade) é utilizada para medir a capacidade de um modelo de detectar todos os casos positivos. Ela é definida como a razão entre o número de verdadeiros positivos (TP) e a soma do número de verdadeiros positivos (TP) e falsos negativos (FN).

A equação é dada por:

$$\text{Recall} = \frac{TP}{TP+FN} \tag{3.2}$$

Onde:

- **TP (True Positives):** Casos que foram corretamente classificados como positivos.
- **FN (False Negatives):** Casos que foram incorretamente classificados como negativos.

4 RESULTADOS E DISCUSSÕES

Os treinamentos, validação k-fold e validação do bancos de dados C foram divididos em cinco etapas, sendo:

4.1 Treinamento dos moldelos DenseNet e ResNet

Na primeira etapa, foram treinados os modelos das arquiteturas DenseNet e ResNet com treinamento supervisionado direto sobre o bancos de dados de treinamento. As tabelas 4.1 e 4.2 mostram os resultados dos treinamento com detalhes das métricas *accutacy*, *precision*, *recall* e *f1-score* de cada classe, juntamente com a apegada de carbono e consumo de energia elétrica do treinamento. As matrizes de confusão da figuras 4.1 e figuras 4.2 indicaram que, de modo geral, os modelos apresentaram acurácia alta nos dois modelos (acima de 99%).

Tabela 4.1 – Métricas alcançadas pelo treinamento da rede Densenet

Classe	**precision**	**recall**	**f1-score**
cercospora	0.9952	0.9857	0.9904
leaf-miner	1.0000	1.0000	1.0000
leaf-rust	0.9865	0.9910	0.9910
Accuracy total = 0.9937			
Tempo de execução (hh:mm:ss) = 00:13:09			
Emissões de CO_2		**Consumo estimado de energia**	
0.000628 kg		0.006389 kWh	

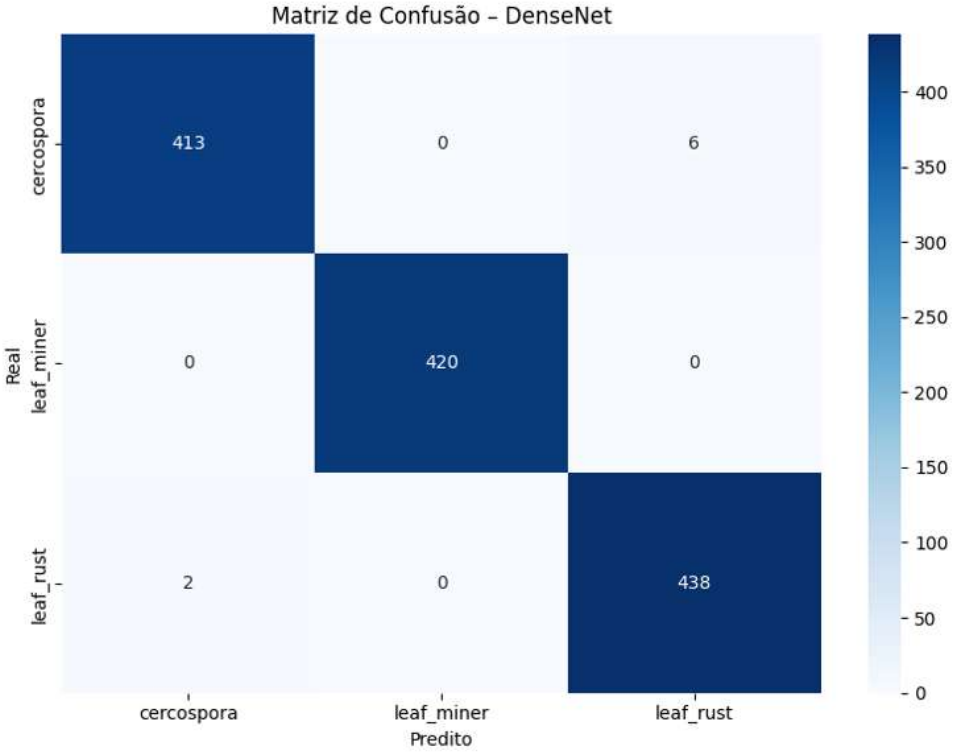
Fonte: Autor (2025).

Tabela 4.2 – Métricas alcançadas pelo treinamento da rede ResNet

Classe	**precision**	**recall**	**f1-score**
cercospora	1.0000	0.9785	0.9891
leaf-miner	1.0000	0.9976	0.9988
leaf-rust	0.9778	1.0000	0.9988
Accuracy total = 0.9922			
Tempo de execução (hh:mm:ss) = 00:10:46			
Emissões de CO_2		**Consumo estimado de energia**	
0.000489 kg		0.004969 kWh	

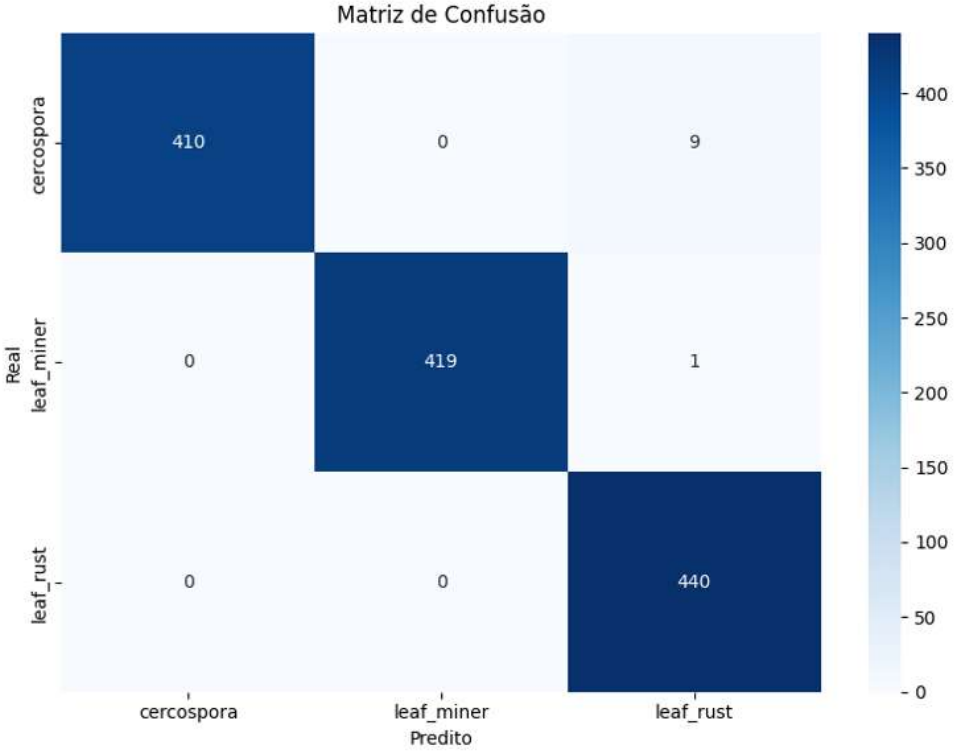
Fonte: Autor (2025).

Figura 4.1 – Matriz de confusão alcançadas pelo treinamento da rede Densenet



Fonte: do Autor

Figura 4.2 – Matriz de confusão alcançadas pelo treinamento da rede Resnet



Fonte: do Autor

Os resultados obtidos nos treinamentos com as arquiteturas **DenseNet** e **ResNet** apresentaram desempenho altamente satisfatório, demonstrando que ambas as redes foram capazes de aprender de forma consistente os padrões presentes nos dados. Esse desempenho sólido serviu como base confiável para avançar para a etapa seguinte do estudo, que consistiu em aplicar *transfer learning* em uma arquitetura mais leve e eficiente, a EfficientNet. A decisão de prosseguir para essa fase foi motivada pela intenção de manter altos níveis de acurácia, ao mesmo tempo reduzindo o custo computacional e o tempo de inferência por meio de uma rede mais compacta.

4.2 Transfer Learning e EfficientNet

Nesta segunda etapa, foi feita o *transfer learning* dos modelos DenseNet e ResNet para o um novo modelo da arquitetura EfficientNet, afim de acelerar o processo do treinamento do mesmo. A tabelas 4.3 e 4.4 mostram os resultados dos treinamento com detalhes das métricas *accutacy*, *precision*, *recall* e *f1-score* de cada classe, juntamente com a pegada de carbono e consumo de energia elétrica do treinamento. As matrizes de confusão da figuras 4.3 e figuras 4.4 indicaram que, de modo geral, os modelos apresentaram acurácia alta nos dois modelos (acima de 96%).

Tabela 4.3 – Métricas alcançadas pelo treinamento da rede EfficientNet (backbone DenseNet)

Classe	**precision**	**recall**	**f1-score**
cercospora	0.9690	0.9714	0.9702
leaf-miner	0.9857	0.9833	0.9945
leaf-rust	0.9705	0.9705	0.9705
Accuracy total = 0.9750			
Tempo de execução (hh:mm:ss) = 00:27:29			
Emissões de CO_2		**Consumo estimado de energia**	
0.002902 kg		0.029507 kWh	

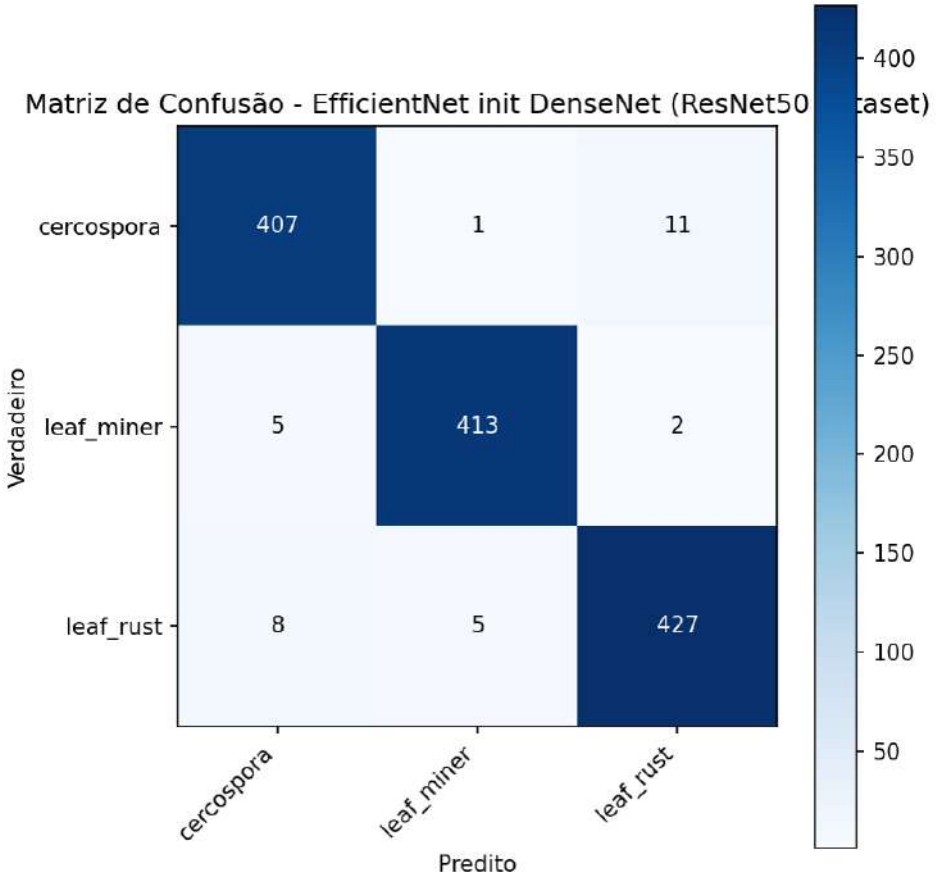
Fonte: Autor (2025).

Tabela 4.4 – Métricas alcançadas pelo treinamento da rede EfficientNet (backbone Resnet)

Classe	**precision**	**recall**	**f1-score**
cercospora	0.9689	0.9666	0.9677
leaf-miner	0.9976	0.9881	0.9928
leaf-rust	0.9663	0.9773	0.9718
Accuracy total = 0.9773			
Tempo de execução (hh:mm:ss) = 00:38:59			
Emissões de CO_2		**Consumo estimado de energia**	
0,013701 kg		0,139309 kWh	

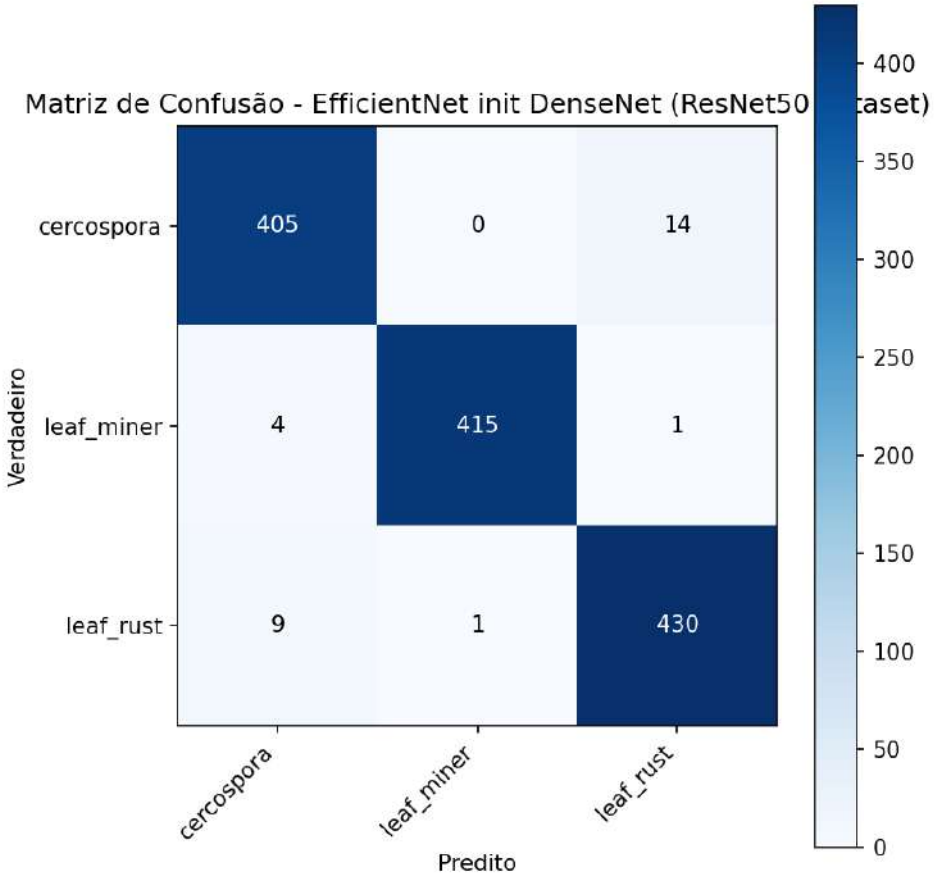
Fonte: Autor (2025).

Figura 4.3 – Matriz de confusão alcançadas pelo treinamento da rede EfficientNet (Backbone Densenet)



Fonte: do Autor

Figura 4.4 – Matriz de confusão alcançadas pelo treinamento da rede EfficientNet (Backbone Resnet)



Fonte: do Autor

Os resultados obtidos nos treinamentos com a arquitetura EfficientNet, ao receber o transfer learning proveniente dos modelos DenseNet e ResNet, demonstrou excelente capacidade de adaptação. Ao incorporar o conhecimento previamente extraído pelas redes mais profundas, o modelo conseguiu não apenas manter o alto nível de desempenho observado nas etapas anteriores, mas também otimizar o processo de inferência graças à sua estrutura mais leve. Os resultados obtidos confirmaram que a transferência do aprendizado foi bem assimilada, permitindo que a EfficientNet alcançasse métricas robustas mesmo com menor demanda computacional, consolidando-se como uma solução eficiente e precisa para a classificação das imagens.

4.3 K-folds dos modelos EfficientNet

Nesta terceira etapa, foi feita os ***k-fold*** do modelos EfficientNet (Backcone DenseNet e Resnet), para avaliar os desempenhos dos mesmos de forma mais confiável, reduzindo o risco de que a avaliação dependa de uma única divisão específica dos dados, diminuindo a variância dos resultados e revelando se o modelo realmente generaliza bem utilizando o banco de dados A. A tabela 4.5 mostra os resultados dos treinamento com acurácia dos treinamentos do modelo

EfficienteNet (backbone Densenet) e a tabela 4.6 mostra os resultados dos treinamento com acurácia dos treinamentos do modelo EfficienteNet (backbone Resnet)

Tabela 4.5 – Melhores acurácias obtidas para cada fold de validação cruzada EfficienteNet (backbone Densenet)

K Folds	**Melhor Acurácia**
Fold 1	0.8375
Fold 2	0.8222
Fold 3	0.8420
Fold 4	0.8260
Fold 5	0.8385

Tempo de execução (hh:mm:ss): 01:53:48
Emissões de CO_2: 0.003759 kg
Consumo estimado de energia: 0.038224 kWh
Fonte: Autor (2025).

Tabela 4.6 – Melhores acurácias obtidas para cada fold de validação cruzada EfficienteNet (backbone Resnet)

K Folds	**Melhor Acurácia**
Fold 1	0.8347
Fold 2	0.8205
Fold 3	0.8357
Fold 4	0.8249
Fold 5	0.8455

Tempo de execução (hh:mm:ss): 02:01:49
Emissões de CO_2: 0.004024 kg
Consumo estimado de energia: 0.040914 kWh
Fonte: Autor (2025).

Os resultados apresentados na tabela 4.5 e tabela 4.6 de validação cruzada revelam a consistência do desempenho da EfficientNet utilizando ao longo dos cinco folds avaliados. Em ambas as execuções, observa-se que as acurácias se mantiveram próximas, com variações discretas entre os folds, o que demonstra estabilidade no processo de treinamento e boa capacidade de generalização do modelo.

No primeiro conjunto de resultados (tabela 4.5), as acurácias variaram entre 0.8222 e 0.8420, enquanto no segundo conjunto (tabela 4.6) oscilaram entre 0.8205 e 0.8455. Essas pequenas diferenças entre os folds são esperadas em validação cruzada e reforçam que o modelo não apresentou comportamentos anômalos ou queda de desempenho significativa em nenhuma das partições e manteve desempenho robusto mesmo com diferentes divisões de dados. Essa

convergência reforça a confiabilidade do modelo e valida a eficácia do uso do transfer learning aliado a uma arquitetura mais leve.

4.4 Validação dos K-folds EfficienteNet

Nesta quarta etapa, foram avaliados os melhores modelos de cada **k-fold** em sua respectiva modelagem. Ao testar o modelo final em um banco de validação completamente externo ao treinamento (Banco de dados C) é possível verificar se o comportamento observado durante o **k-fold** realmente se mantém em condições reais de uso. Esse procedimento permite identificar problemas como *overfitting*, viés de distribuição ou dependência de padrões específicos do dataset original, especialmente em contextos nos quais há variações de iluminação, fundo, qualidade da imagem ou características das doenças e pragas.

Os modelos escolhidos foram:

- **FOLD-3** EfficienteNet (backbone DENSENET)
- **FOLD-5** EfficienteNet (backbone RESNET)

As tabelas 4.7 e 4.8 mostram os resultados dos treinamento com detalhes das métricas *accuracy*, *precision*, *recall* e *f1-score* de cada modelo. As matrizes de confusão da figuras 4.5 e figuras 4.6 indicaram uma queda brusca nas métricas e uma forte confusão do modelo de muitas predições incorretas como cercospora e leaf-miner.

Tabela 4.7 – Métricas alcançadas na validação do fold-3 EfficientNet (Backbone Densenet)

Classe	**precision**	**recall**	**f1-score**
cercospora	0.6482	0.7008	0.6735
leaf-miner	0.8739	0.6230	0.7274
leaf-rust	0.7139	0.8993	0.7960
Accuracy total = 0.7311			
Tempo de execução (hh:mm:ss) = 00:02:31			
Emissões de CO_2		**Consumo estimado de energia**	
0,000082 kg		0,000830 kWh	

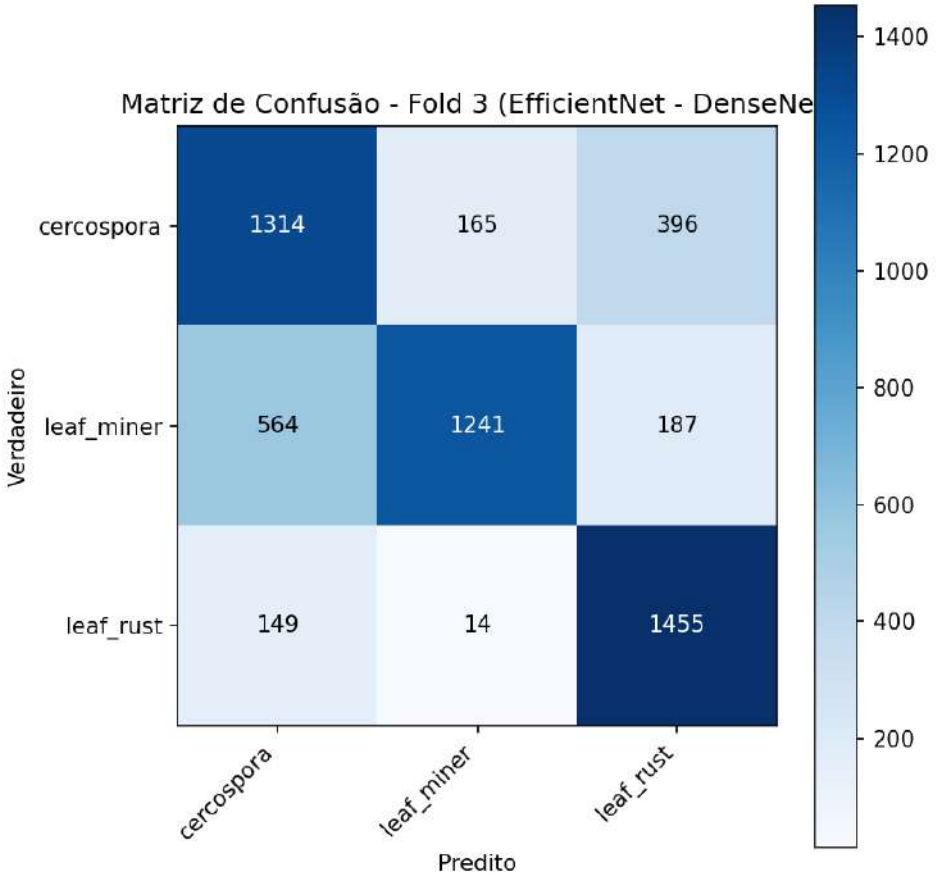
Fonte: Autor (2025).

Tabela 4.8 – Métricas alcançadas na validação do fold-5 EfficientNet (Backbone Resnet)

Classe	**precision**	**recall**	**f1-score**
cercospora	0.4403	0.8187	0.5727
leaf-miner	0.6384	0.0718	0.1291
leaf-rust	0.5454	0.5983	0.5706
Accuracy total = 0.4822			
Tempo de execução (hh:mm:ss) = 00:02:22			
Emissões de CO_2		**Consumo estimado de energia**	
0,000077 kg		0,000781 kWh	

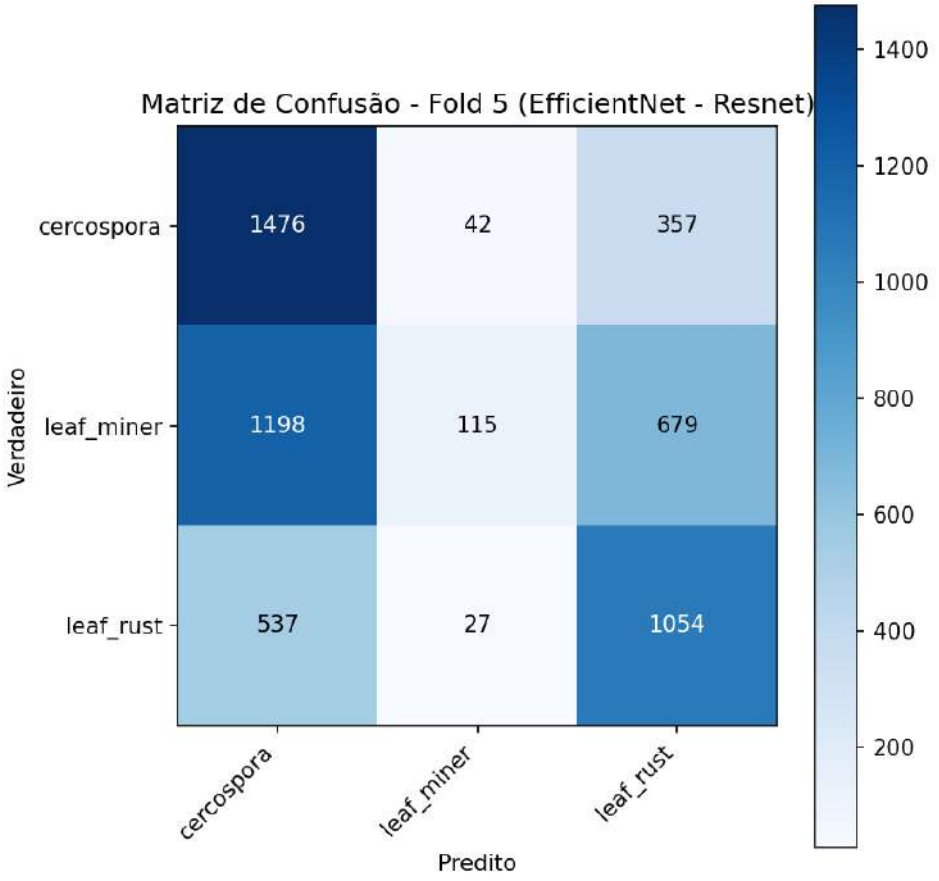
Fonte: Autor (2025).

Figura 4.5 – Matriz de confusão alcançadas pela validação do fold-3 EfficientNet (Backbone Desnsenet)



Fonte: do Autor

Figura 4.6 – Matriz de confusão alcançadas pela validação do fold-5 EfficientNet (Backbone Resnet)



Fonte: do Autor

A validação externa do *FOLD-3* da EfficientNet com backbone DenseNet demonstrou uma queda perceptível no modelo quando testado com o banco de dados C. Os resultados indicam um desempenho moderado do modelo, com acurácia global de 0,7311, o que significa que cerca de 73% das imagens foram classificadas corretamente, mas com um comportamento desbalanceado entre as classes. A classe leaf_rust foi a mais bem reconhecida em termos de cobertura, apresentando recall muito alto (0,8993), ou seja, o modelo consegue encontrar a grande maioria dos casos reais dessa doença; contudo, sua precision (0,7139) sugere que ainda há um número relevante de falsos positivos (o modelo "chama de leaf_rust" imagens que na verdade pertencem a outra classe), o que é consistente com um modelo que tende a "puxar" previsões para essa classe. Já leaf_miner mostra o padrão oposto: precision muito alta (0,8739) com recall baixo (0,6230), indicando que quando o modelo prevê leaf_miner ele geralmente acerta, porém deixa muitos casos reais passarem, classificando-os como outra doença (alto falso negativo), o que é crítico se o objetivo for detecção sensível. Por fim, cercospora aparece como a classe mais frágil, com precision (0,6482) e f1-score (0,6735) mais baixos, sugerindo confusão com as demais e dificuldade em capturar bem suas características visuais; apesar disso, o recall de 0,7008 mostra que ainda consegue recuperar parte considerável dos casos.

A validação externa do *FOLD-5* da EfficientNet com backbone DenseNet apresentou novamente um desempenho reduzido ao ser aplicada ao conjunto de imagens do banco de dados C, evidenciando as dificuldades de generalização do modelo diante das condições reais de

coleta. As métricas globais confirmam que o modelo teve desempenho abaixo do observado na validação interna (banco de dados de treinamento), refletindo a complexidade das imagens de campo, o que podemos deduzir ser em razão de diferenças de iluminação, diferentes padrões de fundo e sobreposições de sintomas. A baixa taxa de acertos indica que o modelo ainda não consegue lidar com todas essas variabilidades externas de maneira robusta. A matriz de confusão reforça esse comportamento ao mostrar uma forte confusão entre as classes, principalmente entre leaf-miner e cercospora, além de um grande número de classificações incorretas envolvendo leaf-rust. Observa-se que cercospora continua sendo a classe mais superestimada pelo modelo, enquanto leaf-miner sofreu elevado número de erros, com mais de mil imagens rotuladas incorretamente como cercospora.

Esses resultados reforçam a necessidade de ampliar e diversificar o treinamento, além de ajustar o modelo para lidar melhor com condições reais de campo.

4.5 Treinamentos de modelos de Machine Learning

Nesta quinta etapa, buscou-se reduzir a dimensionalidade dos ***vetores de features*** extraídos pela EfficientNet (Backbone Densenet), de modo a diminuir o custo computacional das etapas posteriores de análise e classificação. Para isso, foram aplicadas técnicas de redução de dimensionalidade, permitindo compactar a representação dos dados sem perda significativa de informação. Em seguida, realizou-se o treinamento e a validação por meio de K-Fold, mantendo o mesmo padrão metodológico adotado nos modelos de CNNs, garantindo assim uma comparação consistente entre as abordagens.

As etapas dentro deste processo foram:

- **Modelos treinados da EfficientNet:** Os modelos EfficientNet previamente treinados são utilizados como extratores de características. A EfficientNet não é mais treinada aqui; ela atua como uma rede capaz de transformar cada imagem em um vetor de alta dimensionalidade, que contém os padrões aprendidos durante o treinamento.

- **Extração das "features"(vetores de características):**Cada imagem é passada pela EfficientNet, e em vez de pegar a predição final, são coletadas as ativações internas da rede. Esses vetores podem ter milhares de dimensões e são considerados uma representação compacta e rica dos padrões visuais.

- **Aplicação do PCA sobre as features extraídas:** Com os vetores de características em mãos, aplica-se o PCA (Principal Component Analysis), que reduz a dimensionalidade desses dados mantendo a maior parte da variância. O PCA identifica combinações lineares das características originais e gera um novo espaço reduzido que preserva o máximo possível de informação.

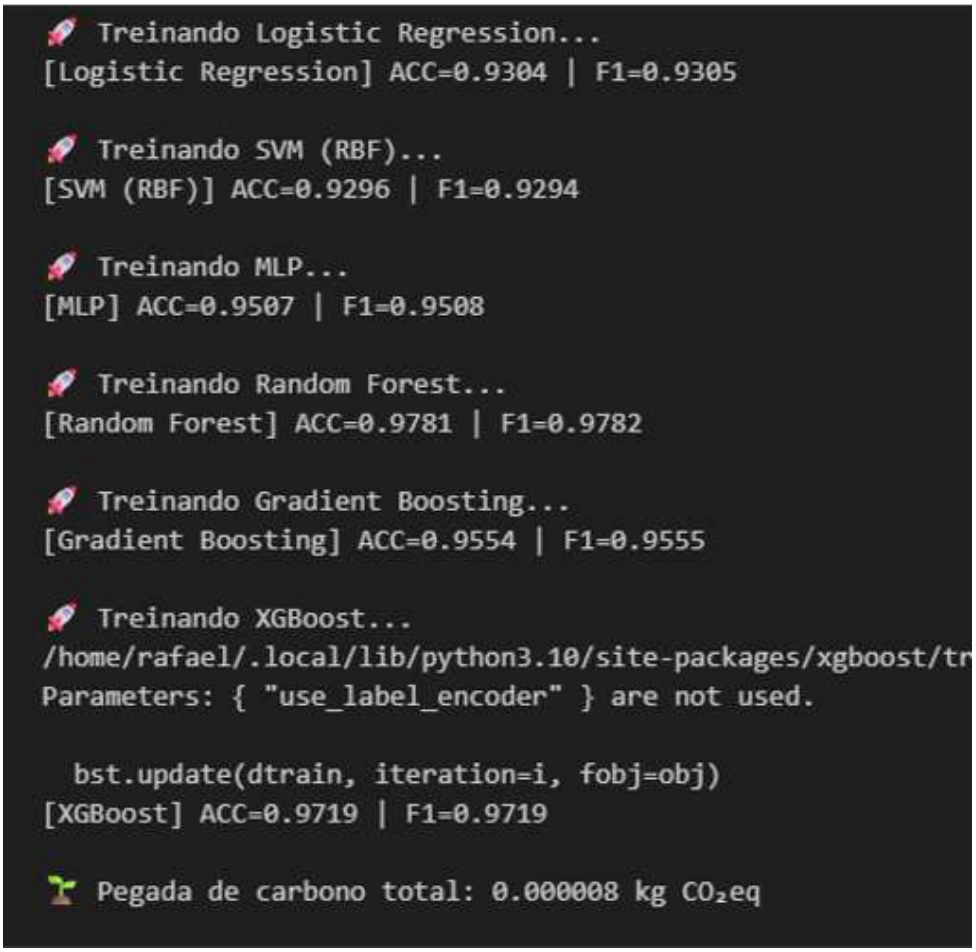
- **Geração das features reduzidas**: Após a aplicação do PCA, obtém-se um novo conjunto de vetores com muito menos dimensões, porém ainda representativos dos padrões aprendidos pela EfficientNet.

- **Utilização das features reduzidas em classificadores:** Por fim, os vetores reduzidos são usados como entrada para diferentes modelos de Machine Learning, como MLP, Random Forest, XGBoost e etc.

4.5.1 Redução de dimensionalidade via Análise de Componentes Principais (PCA)

Foram extraídos da técnica de redução de dimensionalidade PCA do bancos de dados de treinamento 5159 imagens, com 5 dimensões cada. A figura 4.7 mostra os resultados de acurácia e *f1-score* dos modelos *Logistic Regression*, *Suporte vector Machine (RBF)*, *Multi Layer Percepton (MLP)*, *Random Florest*, *Gradient Doosting* e *XGBosst*.

Figura 4.7 – Métricas alcançadas pelas ML da técnica de redução de dimensionalidade via PCA



Fonte: do Autor

Em síntese, todos os modelos obtiveram resultados expressivos, mas o **XGBoost** foi o que melhor explorou a combinação EfficientNet + PCA, alcançando o maior desempenho geral. Estes resultados mostram que, após a redução dimensional, classificadores baseados em *boosting* tendem a extrair o máximo das features condensadas, superando os demais métodos tradicionais.

4.5.1.1 K-fold do melhor modelo (XGboost) usando redução de dimensionalidade via PCA

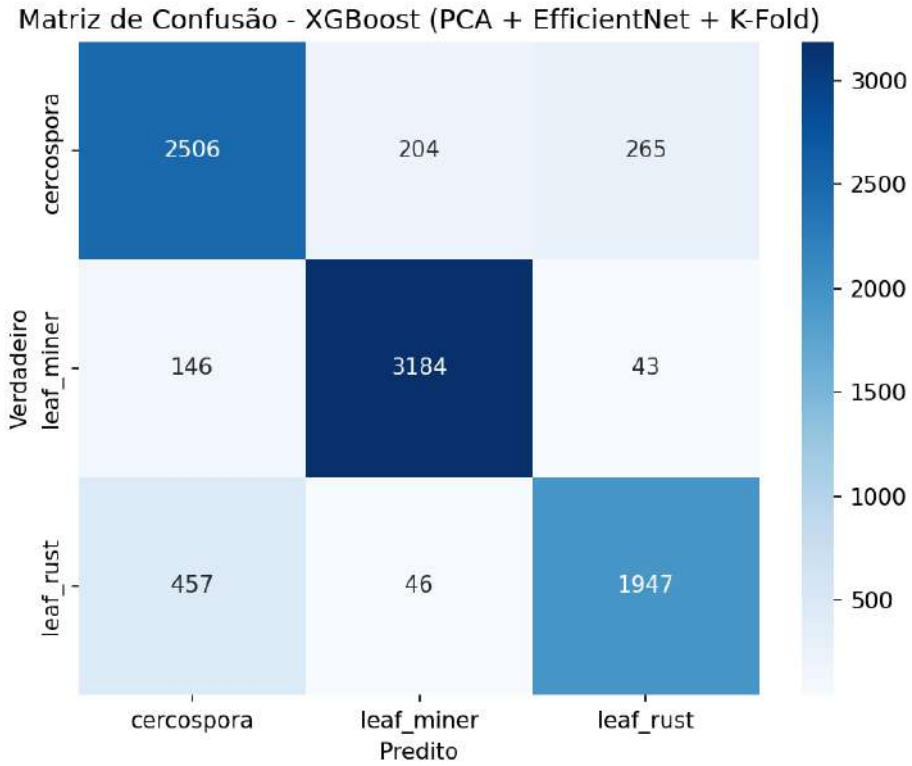
A tabela 4.9 mostram os resultados do melhor treinamento com detalhes das métricas *accuracy*, *precision*, *recall* e *f1-score* de do melhor modelo obtido. A matriz de confusão do treinamento da figuras 4.8 indicaram um resultado satisfatório semelhantes aos modelos aqui já descritos anteriormente.

Tabela 4.9 – Melhores acurácias obtidas para cada fold de validação cruzada modelo XGboost via PCA

K Folds	Accuracy	Precision	Recall	f1-score
Fold 1	0.781250	0.773021	0.772101	0.772527
Fold 2	0.748295	0.738410	0.739511	0.738909
Fold 3	0.769886	0.761467	0.762082	0.761670
Fold 4	0.754406	0.747850	0.743211	0.744908
Fold 5	0.766913	0.757772	0.759398	0.758454

Fonte: Autor (2025).

Figura 4.8 – Métricas alcançadas do modelo XGboost utilizando a técnica de redução de dimensionalidade via PCA



Fonte: do Autor

O modelo XGBoost apresentou desempenho sólido e homogêneo, confirmando sua capacidade de extrair padrões relevantes mesmo após o PCA compactar as features geradas pela EfficientNet. A proximidade entre precisão, sensibilidade e f1-score reforça que o modelo manteve um bom equilíbrio entre acertos positivos e negativos, evitando tanto excessos de falsos positivos quanto de falsos negativos. Assim, os resultados indicam que o XGBoost se comportou de maneira robusta durante todo o processo de validação cruzada, consolidando-se como um classificador eficiente dentro dessa etapa metodológica.

4.5.1.2 Validação do modelo XGboost (FOLD-1) usando redução de dimensionalidade via PCA

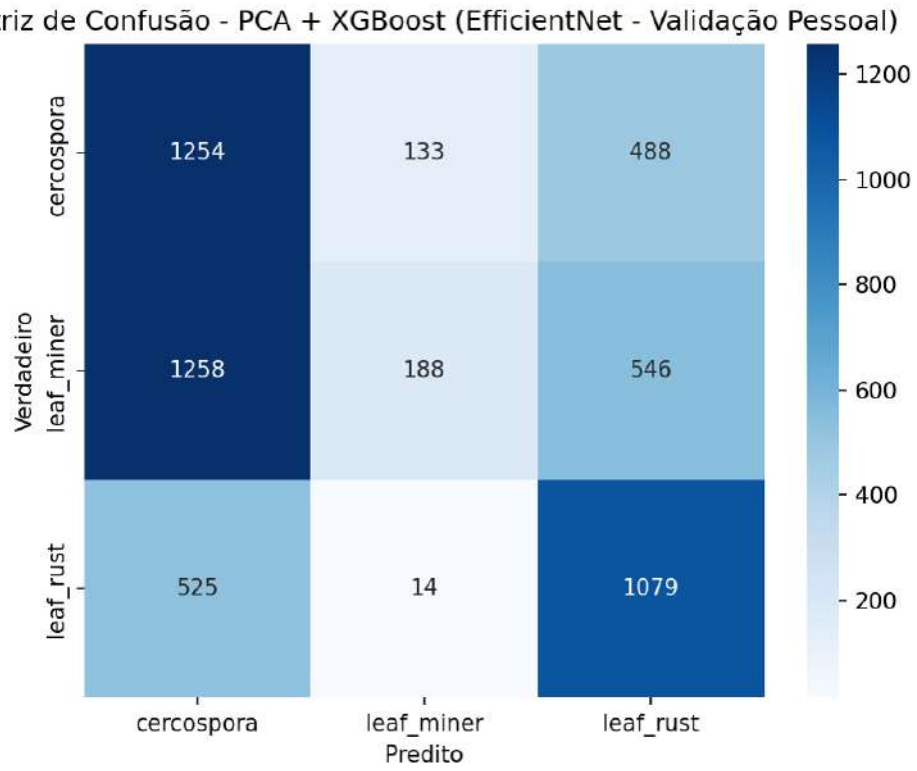
A tabela 4.10 mostra o resultado de validação com detalhes das métricas *accuracy precision*, *recall* e *f1-score* do modelo. A matriz de confusão da figuras 4.9 indicaram uma queda brusca nas métricas e uma forte confusão do modelo de muitas predições incorretas como cercospora e leaf-miner, semelhante ao já constatado em modelos anteriores.

Tabela 4.10 – Métricas alcançadas na validação do fold-5 EfficientNet (Backbone Densenet)

Metrica	**Valor Aproximado**
accuracy	0.479125
precision	0.546075
recall	0.493544
f1-score	0.423758

Fonte: Autor (2025).

Figura 4.9 – Métricas alcançadas do modelo XGboost utilizando a técnica de redução de dimensionalidade via PCA



Fonte: do Autor

A validação externa do *FOLD-1* do modelo XGBoost (após PCA e features da EfficientNet) mostra uma queda significativa de desempenho quando o classificador é aplicado ao conjunto de imagens do banco de dados C, que apresenta condições reais e não controladas

de campo. As métricas globais indicam que o modelo teve dificuldade em generalizar para esse domínio mais complexo, apesar do excelente desempenho observado na validação interna. Esse comportamento reforça que as imagens externas (banco de dados pessoal) possuem maior variabilidade de cenário, iluminação e características visuais que não estavam completamente representadas no conjunto de treinamento original.

A matriz de confusão mostra um padrão claro de erro. Há uma forte tendência do modelo em classificar diversas amostras como cercospora, especialmente aquelas que pertencem à classe leaf-miner, resultando em 1.258 classificações incorretas. A classe leaf-rust também foi afetada, com 525 amostras sendo classificadas como cercospora. Apesar disso, o modelo conseguiu manter um desempenho razoável para leaf-rust, com 1.079 acertos, o que sugere que essa classe apresenta características mais consistentes ou que foram melhor representadas no treinamento. No geral, a validação do FOLD-1 evidencia que o modelo ainda encontra dificuldades ao lidar com a diversidade visual de imagens reais, mesmo após o uso do PCA e do XGBoost, reforçando a necessidade de ampliar o dataset pessoal ou aplicar técnicas adicionais de robustez.

4.5.2 Redução de dimensionalidade via MINIROCKET

O MiniRocket é um método eficiente para a transformação de séries temporais em vetores de características altamente discriminativos por meio de convoluções unidimensionais. No contexto deste trabalho, o método é empregado sobre vetores de embeddings extraídos previamente por uma rede convolucional profunda do EfficientNet, os quais são reinterpretados como séries unidimensionais artificiais, permitindo sua adaptação ao formato de entrada exigido pelo MiniRocket.

O objetivo dessa etapa é realizar uma projeção convolucional não linear das representações latentes, explorando padrões estruturais ao longo da dimensão do embedding, sem a necessidade de aprendizado de parâmetros adicionais. O MiniRocket utiliza um conjunto fixo e determinístico de kernels convolucionais, definidos a partir de combinações específicas de pesos binários e diferentes dilatações, o que possibilita a captura de padrões em múltiplas escalas com elevado desempenho computacional.

As etapas dentro deste processo foram:

- **Aplicação dos kernels convolucionais do MiniRocket:** Os kernels determinísticos são aplicados de forma eficiente sobre as séries unidimensionais derivadas dos embeddings, utilizando diferentes dilatações e vieses, gerando mapas de ativação representativos.

- **Extração das feições transformadas (features MiniRocket):** Para cada convolução, o MiniRocket calcula estatísticas simples, principalmente a Proportion of Positive Values (PPV), que quantifica a fração de ativações superiores a um limiar. Essas estatísticas compõem um vetor de características de dimensão fixa, altamente discriminativo e estável.

- **Composição do conjunto final de características:** Os vetores gerados pelo MiniRocket são utilizados como entrada para classificadores supervisionados, dispensando etapas adicionais de normalização ou ajuste de parâmetros.

- **reinamento dos classificadores:** Com base nas features transformadas, são empregados classificadores eficientes, incluindo Logistic Regression, Support Vector Machines, Random Forest, Gradient Boosting, XGBoost e redes neurais rasas (MLP), permitindo avaliar o impacto da transformação MiniRocket em diferentes paradigmas de aprendizado supervisionado.

A imagem 4.10 mostra os resultados dos treinamentos com detalhes das métricas *accuracy*, *precision*, *recall* e *f1-score* de cada modelo. As matriz de confusão da figuras 4.11 indicaram que, de modo geral, os modelos apresentaram acurácia alta nos dois modelos (acima de 83%).

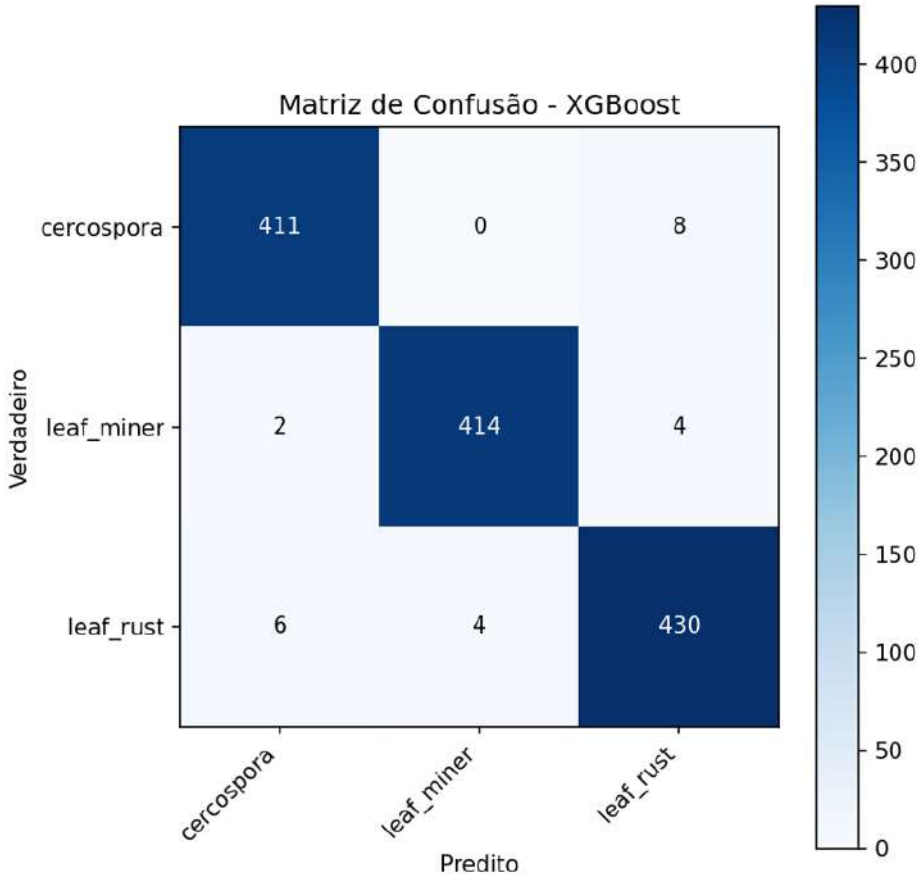
Figura 4.10 – Métricas alcançadas do modelo XGboost utilizando a técnica de redução de dimensionalidade (via MINIROCKET)

Resultados comparativos:

	accuracy	precision	recall	f1
Logistic Regression	0.937451	0.939085	0.937216	0.937648
SVM (RBF)	0.830336	0.835252	0.832375	0.829081
Random Forest	0.965598	0.965885	0.965706	0.965778
Gradient Boosting	0.960907	0.961354	0.960938	0.961057
MLP	0.877248	0.877813	0.878027	0.877291
XGBoost	0.981235	0.981396	0.981298	0.981343

Fonte: do Autor

Figura 4.11 – Matriz do modelo melhor modelo no treinamento (XGboost) utilizando a técnica de redução de dimensionalidade via MINIROCKET



Fonte: do Autor

4.5.2.1 K-fold do modelo XGboost usando redução de dimensionalidade via MINIROCKET

A tabela 4.11 mostram os resultados dos treinamento com detalhes das métricas *accuracy*, *precision*, *recall* e *f1-score* de cada modelo. As matrizes de confusão da figuras 4.8 indicaram que há uma queda significativa nas métricas, porem ainda tem um resultado satisfatório.

Tabela 4.11 – Melhores acurácias obtidas para cada fold de validação cruzada modelo XGboost via MINIROCKET

K Folds	**Accuracy**	**Precision**	**Recall**	**f1-score**
Fold 1	0.754545	0.747905	0.744511	0.745851
Fold 2	0.719318	0.710618	0.710840	0.710618
Fold 3	0.746591	0.739361	0.738213	0.738742
Fold 4	0.722570	0.715250	0.711602	0.713016
Fold 5	0.736782	0.727579	0.727690	0.727628

Fonte: Autor (2025).

4.5.2.2 Validação do modelo XGboost (Fold-1) usando redução de dimensionalidade via MINIROCKET

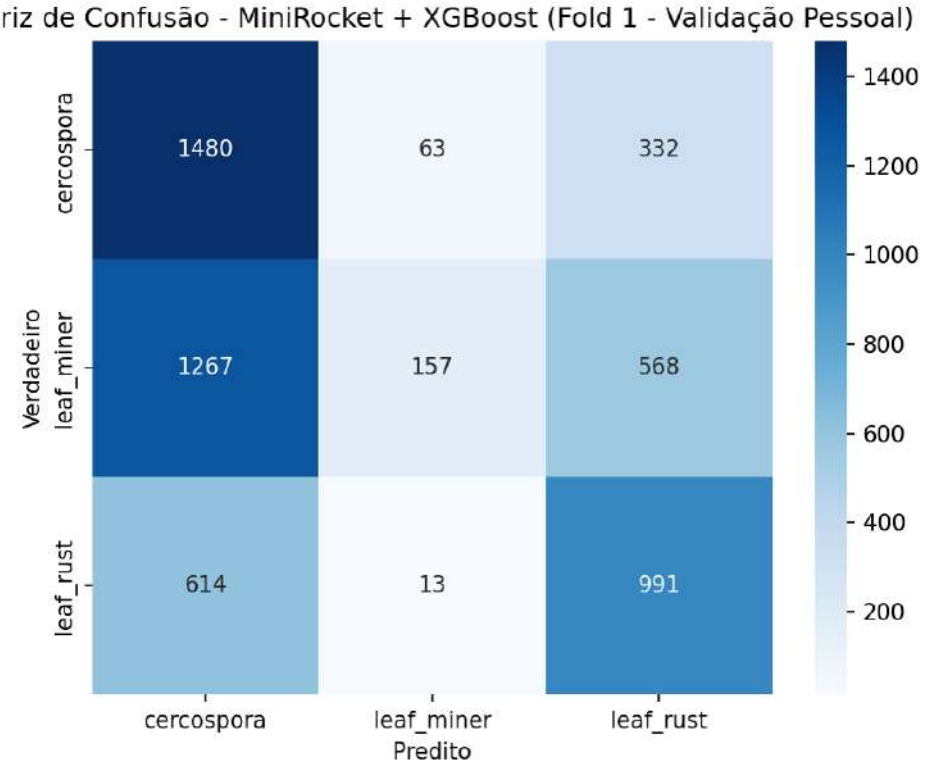
A figura 4.12 mostram os resultados dos treinamentos com detalhes das métricas *accuracy*, *precision*, *recall* e *f1-score* de cada k-fold. A matriz de confusão da figuras 4.12 indicaram uma queda brusca nas métricas e uma forte confusão do modelo de muitas predições incorretas como cercospora e leaf-miner, semelhante ao já constatado em modelos anteriores.

Tabela 4.12 – Métricas alcançadas do K-fold do modelo XGboost utilizando a técnica de redução de dimensionalidade (via MiniRocket)

Metrica	**Valor Aproximado**
accuracy	0.479125
precision	0.546075
recall	0.493544
f1-score	0.423758

Fonte: Autor (2025).

Figura 4.12 – Matriz de validação do modelo XGboost(fold-1) utilizando a técnica de redução de dimensionalidade via MINIROCKET)



Fonte: do Autor

4.5.2.3 Comparações dos resultados

A tabela 4.13 mostra a valores obtidos de todos treinamento e a tabela 4.14 referese a comparação entre os modelos teste K-FOLD e validação externa.

Tabela 4.13 – Comparação entre os modelos de treinamento e teste

Modelo	**Treinamento**
EfficienteNet (backbone Densenet)	0.9750
EfficienteNet (backbone Resnet)	0.9773
XGBoost (Via PCA)	0.9719
XGBoost (Via MINIROCKET)	0.9812

Fonte: Autor (2025).

Tabela 4.14 – Comparação entre os modelos teste K-FOLD e validação

Modelo	**K-fold**	**Validação**
EfficienteNet (backbone Densenet)	0.841500	0.7311
EfficienteNet (backbone Resnet)	0.845500	0.435552
XGBoost (Via PCA)	0.781250	0.479125
XGBoost (Via MINI-ROCKET)	0.754545	0.479125

Fonte: Autor (2025).

5 CONCLUSÃO

Os experimentos conduzidos permitiram avaliar não apenas o desempenho preditivo das arquiteturas analisadas, mas também seu impacto computacional e ambiental, aspecto cada vez mais relevante em aplicações de inteligência artificial. Observou-se que as arquiteturas base, como DenseNet e ResNet, apresentaram baixo consumo energético e reduzida emissão de CO_2, com tempos de execução de 00:13:09 e 00:10:46, resultando em 0,006389 kWh (0,000628 kg de CO_2) e 0,004969 kWh (0,000489 kg de CO_2), respectivamente. Em contrapartida, os modelos híbridos baseados em EfficientNet mostraram custo computacional significativamente superior, sobretudo na configuração com backbone ResNet, que demandou 00:38:59 de processamento, consumindo 0,139309 kWh e emitindo 0,013701 kg de CO_2. Mesmo a versão EfficientNet com backbone DenseNet apresentou aumento relevante no consumo (0,029507 kWh e 0,002902 kg de CO_2). Já as etapas de validação (fold-3 e fold-5) demonstraram impacto ambiental praticamente desprezível, com consumo inferior a 0,001 kWh e emissões inferiores a 0,0001 kg de CO_2, evidenciando que o custo energético concentra-se majoritariamente na fase de treinamento.

Os resultados demonstraram que, em validação interna, as arquiteturas convolucionais atingiram elevada acurácia, confirmando sua capacidade de aprendizado sob condições controladas, semelhante aos resultados de Abuhayi e Mossa (2023), Paulos e Woldeyohannis (2022) e Novtahaning et al. (2022). Entretanto, nas validações externas com o banco de dados pessoal, houve degradação significativa do desempenho, evidenciando limitações de generalização. Entre os modelos híbridos, a combinação EfficientNet + PCA + XGBoost apresentou equilíbrio entre desempenho e eficiência computacional, mantendo elevada estabilidade no K-Fold. Contudo, sob a perspectiva energética e ambiental, observa-se que arquiteturas mais complexas implicam aumento considerável no consumo de energia e na emissão de CO_2, o que deve ser ponderado em aplicações práticas e escaláveis no contexto agrícola. Assim, conclui-se que futuras estratégias devem buscar não apenas maximizar a acurácia e a robustez, mas também otimizar a eficiência energética, ampliando a diversidade do banco de dados e incorporando técnicas de adaptação de domínio para melhorar a generalização em cenários reais.

REFERÊNCIAS

ABDI, H.; WILLIAMS, L. J. Principal component analysis. **Wiley Interdisciplinary Reviews: Computational Statistics**, v. 2, n. 4, p. 433–459, 2010.

ABUHAYI, B. M.; MOSSA, A. A. Coffee disease classification using convolutional neural network based on feature concatenation. **Informatics in Medicine Unlocked**, Elsevier, v. 39, p. 101245, 2023.

AGRAWAL, R.; SRIKANT, R. et al. Fast algorithms for mining association rules. In: SANTIAGO. **Proc. 20th int. conf. very large data bases, VLDB**. [S.l.], 1994. v. 1215, p. 487–499.

ALBUQUERQUE, L. D.; GUEDES, E. B. Coffee plant leaf disease detection for digital agriculture. **SBC Journal on Interactive Systems**, 2024.

AUFAR, Y.; ABDILLAH, M. H.; ROMADONI, J. et al. Web-based cnn application for arabica coffee leaf disease prediction in smart agriculture. **Jurnal Resti (Rekayasa Sistem Dan Teknologi Informasi)**, v. 7, n. 1, p. 71–79, 2023.

BENGIO, Y.; COURVILLE, A.; VINCENT, P. Representation learning: A review and new perspectives. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 35, n. 8, p. 1798–1828, 2013.

CHEN, J.; LITTLE, J. J. Where should cameras look at soccer games: Improving smoothness using the overlapped hidden markov model. **Computer Vision and Image Understanding**, v. 159, p. 59–73, 2017. ISSN 1077-3142. Computer Vision in Sports. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1077314216301709>.

CHU, J. et al. Pay more attention to discontinuity for medical image segmentation. In: SPRINGER. **Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part IV 23**. [S.l.], 2020. p. 166–175.

CUI, Y.; ZHOU, F.; LIN, Y.; BELONGIE, S. Large scale fine-grained categorization and domain-specific transfer learning. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2018. p. 4109–4118.

DEMPSTER, A.; PETITJEAN, F.; WEBB, G. I. Rocket: Exceptionally fast and accurate time series classification using random convolutional kernels. **Data Mining and Knowledge Discovery**, Springer, v. 34, n. 5, p. 1454–1495, 2020.

DEMPSTER, A.; SCHMIDT, D. F.; WEBB, G. I. Minirocket: A very fast (almost) deterministic transform for time series classification. **Data Mining and Knowledge Discovery**, Springer, v. 35, n. 6, p. 2403–2424, 2021.

DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: IEEE. **2009 IEEE conference on computer vision and pattern recognition**. [S.l.], 2009. p. 248–255.

DWYER, B.; AL. et. **coffee-leaf-disease**. 2025. Disponível em: <https://app.roboflow.com/epamig-vnijh/barakobama-coffee-leaf-disease-wvno7/1>.

FERENTINOS, K. P. Deep learning models for plant disease detection and diagnosis. **Computers and Electronics in Agriculture**, v. 145, p. 311–318, 2018. ISSN 0168-1699. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168169917311742>.

GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; MALIK, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2014. p. 580–587.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. [S.l.]: MIT press, 2016.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Regularization for deep learning. **Deep learning**, MIT press Cambridge, MA, USA, p. 216–261, 2016.

GOODFELLOW, I. et al. Generative adversarial nets. **Advances in neural information processing systems**, v. 27, 2014.

GOOGLE. **Goggle Maps**. 2024. Disponível em: <https://www.google.com.br/maps/>.

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**. [S.l.: s.n.], 2016. p. 770–778.

HOCHREITER, S. Long short-term memory. **Neural Computation MIT-Press**, 1997.

HOTELLING, H. Analysis of a complex of statistical variables into principal components. **Journal of Educational Psychology**, v. 24, n. 6, p. 417–441, 1933.

HU, R.; DOLLÁR, P.; HE, K.; DARRELL, T.; GIRSHICK, R. Learning to segment every thing. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2018. p. 4233–4241.

HUANG, G.; LIU, Z.; MAATEN, L. V. D.; WEINBERGER, K. Q. Densely connected convolutional networks. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [s.n.], 2017. p. 4700–4708. Disponível em: <https://arxiv.org/abs/1608.06993>.

HWANG, D.; KIM, J.-J.; MOON, S.; WANG, S. Image augmentation approaches for building dimension estimation in street view images using object detection and instance segmentation based on deep learning. **Applied Sciences**, MDPI, v. 15, n. 5, p. 2525, 2025.

IŞIK, A. H.; ESKICIOGLU, Ö. C. Classification of coffee leaf diseases through image processing techniques. In: **Artificial Intelligence and Smart Agriculture Applications**. [S.l.]: Auerbach Publications, 2022. p. 233–252.

JAMES; WITTEN; HASTIE; TIBSHIRANI. An introduction to statistical learning. springer, 2013.

JOLLIFFE, I. T.; CADIMA, J. Principal component analysis: A review and recent developments. **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**, v. 374, n. 2065, p. 20150202, 2016.

JURAFSKY, D.; MARTIN, J. H. Speech and language processing, prentice hall. **Upper Saddle River, NJ**, 2008.

Kaggle. **Kaggle: Your Machine Learning and Data Science Community**. 2025. <https://www.kaggle.com>. Accessed: 10 Jan. 2025.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. **Communications of the ACM**, AcM New York, NY, USA, v. 60, n. 6, p. 84–90, 2017.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **nature**, Nature Publishing Group UK London, v. 521, n. 7553, p. 436–444, 2015.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, Ieee, v. 86, n. 11, p. 2278–2324, 1998.

LEGG, S.; HUTTER, M. Universal intelligence: A definition of machine intelligence. **Minds and machines**, Springer, v. 17, p. 391–444, 2007.

LEITE, D.; BRITO, A.; FACCIOLI, G. Advancements and outlooks in utilizing convolutional neural networks for plant disease severity assessment: A comprehensive review. **Smart Agricultural Technology**, Elsevier, v. 9, p. 100573, 2024.

LIAO, M.; WAN, F.; GUO, Z.; YE, Q. Hierarchical attentionshift for pointly supervised instance segmentation. **IEEE Transactions on Neural Networks and Learning Systems**, IEEE, 2025.

LIU, W. et al. Ssd: Single shot multibox detector. In: SPRINGER. **Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14**. [S.l.], 2016. p. 21–37.

MACQUEEN, J. **Some methods for classification and analysis of multivariate observations**. [S.l.]: University of California Press, 1967.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. **arXiv preprint arXiv:1301.3781**, 2013.

NILSSON, N. J. **The quest for artificial intelligence**. [S.l.]: Cambridge University Press, 2009.

NOVTAHANING, D.; SHAH, H. A.; KANG, J.-M. Deep learning ensemble-based automated and high-performing recognition of coffee leaf disease. **Agriculture**, MDPI, v. 12, n. 11, p. 1909, 2022.

PAULOS, E. B.; WOLDEYOHANNIS, M. M. Detection and classification of coffee leaf disease using deep learning. In: IEEE. **2022 International Conference on Information and Communication Technology for Development for Africa (ICT4DA)**. [S.l.], 2022. p. 1–6.

PAVLIDIS, T.; LIOW, Y.-T. Integrating region growing and edge detection. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 12, n. 3, p. 225–233, 1990.

PEARSON, K. Liii. on lines and planes of closest fit to systems of points in space. **The London, Edinburgh, and Dublin philosophical magazine and journal of science**, Taylor & Francis, v. 2, n. 11, p. 559–572, 1901.

PECUáRIA, M. da Agricultura e. **Brasil é o maior produtor mundial e o segundo maior consumidor de café**. 2023. Disponível em: <https://www.gov.br/agricultura/pt-br/assuntos/noticias/brasil-e-o-maior-produtor-mundial-e-o-segundo-maior-consumidor-de-cafe>.

PINTO, M. F.; MELO, A. G.; MARCATO, A. L. M.; URDIALES, C. Case-based reasoning approach applied to surveillance system using an autonomous unmanned aerial vehicle. In: **2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)**. [S.l.: s.n.], 2017. p. 1324–1329.

QIU, J.; LIU, J.; SHEN, Y. Computer vision technology based on deep learning. In: **2021 IEEE 2nd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)**. [S.l.: s.n.], 2021. v. 2, p. 1126–1130.

QUINLAN, J. R. Induction of decision trees. **Machine learning**, Springer, v. 1, p. 81–106, 1986.

REIS, P. R.; CUNHA, R. L. da. **Café Arábica, do Plantio à Colheita**. [S.l.]: Epamig, Consórcio Pesquisa Café, 2010. 261 - 267 p.

RONCERO, V. G. **Um estudo de segmentação de imagens baseado em um método de computação evolucionária**. Tese (Doutorado) — Dissertação de mestrado em engenharia elétrica. Rio de Janeiro. 70p, 2005.

RUIZ, A. P.; FLYNN, M.; LARGE, J.; MIDDLEHURST, M.; BAGNALL, A. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. **Data Mining and Knowledge Discovery**, Springer, v. 35, n. 2, p. 401–449, 2021.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **nature**, Nature Publishing Group UK London, v. 323, n. 6088, p. 533–536, 1986.

RUSSELL, S. J.; NORVIG, P. **Artificial intelligence: a modern approach**. [S.l.]: Pearson, 2016.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. **Neural networks**, Elsevier, v. 61, p. 85–117, 2015.

SHIM, D.; LEE, C. Small-object semantic segmentation of satellite ship images using modified u-net with morphological loss. **IEEE Access**, v. 13, p. 27700–27713, 2025.

SIGNO, S. D. R.; TUQUERO, C. L. G.; ARBOLEDA, E. R. Coffee disease detection and classification using image processing: A literature. 2024.

SILVA, J. H. d. N. Segmentação de imagens de tomografia computadorizada utilizando o algoritmo u-net. estudo de caso: imagem da bexiga. 2025.

SILVER, D. et al. Mastering the game of go with deep neural networks and tree search. **nature**, Nature Publishing Group, v. 529, n. 7587, p. 484–489, 2016.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2015.

SOUZA, F. d. F.; SANTOS, J. C. F.; COSTA, J. N. M.; SANTOS, M. M. d. Características das principais variedades de café cultivadas em rondônia. Embrapa Rondônia, 2004.

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. **The journal of machine learning research**, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014.

SUTTON, R. S. Reinforcement learning: an introduction. **A Bradford Book**, 2018.

SZELISKI, R. **Computer vision: algorithms and applications**. [S.l.]: Springer Nature, 2022.

TAIZ, L.; ZEIGER, E.; MØLLER, I. M.; MURPHY, A. **Fisiologia Vegetal**. 6. ed. Porto Alegre: Artmed, 2017.

TAJANE, S. et al. Detection of mineral deficiencies and pests symptoms in coffee crop using computer vision. In: . [S.l.: s.n.], 2024. p. 835–842.

TAN, M.; LE, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks. In: **Proceedings of the 36th International Conference on Machine Learning (ICML)**. [s.n.], 2019. p. 6105–6114. Disponível em: <https://arxiv.org/abs/1905.11946>.

TUIA, D.; VOLPI, M.; COPA, L.; KANEVSKI, M.; MUNOZ-MARI, J. A survey of active learning algorithms for supervised remote sensing image classification. **IEEE Journal of Selected Topics in Signal Processing**, IEEE, v. 5, n. 3, p. 606–617, 2011.

TÜRKOĞLU, M.; HANBAY, D. Plant disease and pest detection using deep learning-based features. **Turkish Journal of Electrical Engineering and Computer Sciences**, v. 27, n. 3, p. 1636–1651, 2019.

VAPNIK, V. Support-vector networks. **Machine learning**, v. 20, p. 273–297, 1995.

VASWANI, A. Attention is all you need. **Advances in Neural Information Processing Systems**, v. 86, n. 11, p. 5998–6008, 2017.

WATKINS, C. J.; DAYAN, P. Q-learning. **Machine learning**, Springer, v. 8, p. 279–292, 1992.

WU, D. et al. Semantic segmentation via pixel-to-center similarity calculation. **CAAI Transactions on Intelligence Technology**, Wiley Online Library, v. 9, n. 1, p. 87–100, 2024.

WU, M. et al. Object detection based on rgc mask r-cnn. **IET Image Processing**, Wiley Online Library, v. 14, n. 8, p. 1502–1508, 2020.

YAMASHITA, J. V. Y. B.; LEITE, J. P. R. Coffee disease classification at the edge using deep learning. **Smart Agricultural Technology**, Elsevier, v. 4, p. 100183, 2023.

ZHENG, S.; WANG, J. High definition map-based vehicle localization for highly automated driving: Geometric analysis. In: IEEE. **2017 International Conference on Localization and GNSS (ICL-GNSS)**. [S.l.], 2017. p. 1–8.