

ARTIGO ORIGINAL

## Aprendizado profundo Aplicado à classificação de doenças foliares do milho

## Deep learning applied to the classification of corn foliar diseases

Vinicius Matheus Pimentel Ariza <sup>1</sup> and Odemir Martinez Bruno <sup>1</sup>

<sup>1</sup>Universidade de São Paulo (USP)

\*[viniciusariza@usp.br](mailto:viniciusariza@usp.br); [bruno@ifsc.usp.br](mailto:bruno@ifsc.usp.br)

Recebido: 25/11/2023. Revisado: 06/07/2024. Aceito: 31/07/2024.

### Resumo

A busca pelo desenvolvimento de modelos inteligentes capazes de resolver problemas complexos é cada vez mais recorrente em diversas áreas. Uma delas é a agricultura, onde as doenças são uma grande preocupação, principalmente pelas perdas de produtividade, impacto social e ecológico que podem causar. Assim, o emprego da tecnologia no auxílio à tomada de decisão pode ser um aliado no acompanhamento do cultivo, e, consequentemente, no sucesso da colheita. O Aprendizado Profundo é uma subárea do Aprendizado de Máquina que tem obtido casos de sucesso no desenvolvimento de modelos inteligentes, se destacando na detecção e classificação de imagens, por meio da utilização de Redes Neurais Convolucionais. Neste contexto, o presente trabalho tem como objetivo avaliar modelos baseados em Aprendizado Profundo para classificar doenças foliares do milho, por meio da análise de regiões em imagens da folha. A metodologia envolveu a utilização de Transferência de Aprendizagem, com a aplicação das Redes Neurais ResNet50 e VGG19 em um subconjunto de dados público com 3.838 imagens de folhas, divididas em quatro classes. Os resultados indicaram acurácia máxima de 98,31% utilizando a Rede Neural VGG19 e técnicas de aumento de dados.

**Palavras-Chave:** Agricultura; Aprendizado Profundo; Doenças Foliares; Redes Neurais Convolucionais; Transferência de Aprendizagem.

### Abstract

The search for the development of intelligent models capable of solving complex problems is increasingly common in various fields. One of them is agriculture, where diseases are a major concern, mainly due to the potential loss of productivity and their social and ecological impact. Thus, the use of technology to aid decision-making can be an ally in monitoring crops and, consequently, ensuring successful harvests. Deep Learning is a subfield of Machine Learning that has achieved successful cases in developing intelligent models, particularly in image detection and classification, through the use of Convolutional Neural Networks. In this context, the present work aimed to evaluate Deep Learning-based models for classifying corn leaf diseases by analyzing regions in leaf images. The methodology involved the use of transfer learning, applying the ResNet50 and VGG19 Neural Networks to a subset of publicly available data with 3,838 leaf images, divided into four classes. The results indicated a maximum accuracy of 98,31% using the VGG19 Neural Network and data augmentation techniques.

**Keywords:** Agriculture; Convolutional Neural Networks; Deep Learning; Foliar Diseases; Transfer Learning.

## 1 Introdução

A agricultura é um setor fundamental na economia global, e a produção saudável de culturas é essencial para garantir a segurança alimentar. No entanto, a saúde das plantas pode ser afetada por diversas doenças, que podem ser causadas por patógenos como vírus, bactérias e fungos. A identificação precoce dessas doenças é fundamental para prevenir sua disseminação e garantir uma produção saudável. Além disso, a detecção correta e o tratamento adequado ajudam a evitar o uso excessivo de pesticidas e o desperdício de recursos.

É conhecido que a maioria das doenças que afetam as plantações gera alguma manifestação visível em suas folhas (Arnal Barbedo, 2013). Porém, a identificação incorreta e o consequente uso equivocado de pesticidas podem causar o desenvolvimento de resistência a longo prazo dos patógenos, reduzindo severamente a capacidade de reversão. Assim, é importante o desenvolvimento de métodos que ajudem na identificação da doença e em um tratamento rápido e eficaz, evitando o desperdício de recursos e alcançando uma produção saudável (Sladojevic et al., 2016).

Atualmente, os métodos de detecção de doenças em plantas são, muitas vezes, baseados em observação visual e análise laboratorial. No entanto, esses métodos podem ser caros, demorados e, em muitos casos, exigem a presença de especialistas ou infraestrutura laboratorial que não estão disponíveis em todas as regiões (Bock et al., 2010). O diagnóstico de doenças em plantas com base na observação das folhas pode ser uma tarefa desafiadora mesmo para especialistas experientes, devido ao grande número de espécies de plantas e problemas fitopatológicos. Isso pode levar a conclusões equivocadas e tratamentos ineficazes, resultando em danos às plantações (Ferentinos, 2018).

A análise de imagem por meio de métodos computacionais pode ser uma alternativa viável para a detecção rápida e precisa de doenças foliares no campo, o que pode contribuir para a eficiência no manejo e prevenção de doenças nas plantações (Ashwini and Sellam, 2024). Assim, o desenvolvimento de sistemas automatizados de diagnóstico de doenças de plantas, que utilizam tecnologias como Inteligência Artificial (IA) e Aprendizado de Máquina (AM), pode ser útil para os agricultores. Esses sistemas têm o potencial de identificar com precisão as doenças presentes nas plantas, permitindo a detecção precoce e a prevenção da disseminação, além de economizar tempo e recursos (Sladojevic et al., 2016). No entanto, há desafios a serem superados para melhorar esses sistemas como, por exemplo, a necessidade de um banco de dados de imagens abrangente que cubra uma variedade maior de doenças, bem como a avaliação dos modelos em culturas específicas (Mohanty et al., 2016; Sladojevic et al., 2016; Ferentinos, 2018). Isso evidencia a necessidade de pesquisas na área, a fim de aprimorar a eficiência e a precisão do diagnóstico de doenças de plantas, o que pode beneficiar os agricultores e contribuir para a sustentabilidade da agricultura (Yang and Guo, 2017).

Neste contexto, o presente trabalho tem como objetivo avaliar modelos baseados em Aprendizado Profundo para classificar doenças foliares da cultura do milho, utilizando regiões de imagens de folhas doentes e saudáveis para o

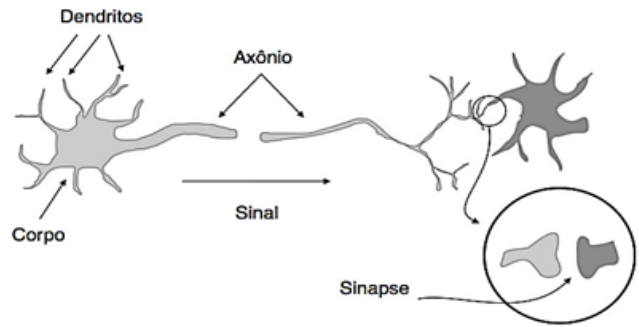


Figura 1: Representação do neurônio biológico.

Fonte: Faceli et al. (2011)

treinamento.

O conteúdo deste trabalho está dividido em mais cinco capítulos. Na Seção 2 são abordados os conceitos fundamentais utilizados no desenvolvimento do estudo, como Redes Neurais Artificiais (RNA), Redes Neurais Convolucionais (CNN), Aprendizado Profundo, Transferência de Aprendizagem e uma breve descrição de algumas doenças foliares do milho.

Em seguida, na Seção 3, são apresentados alguns trabalhos da literatura relacionados a aplicação de Aprendizado Profundo na classificação de doenças foliares em plantas.

Na Seção 4 sobre a metodologia, são descritos o conjunto de dados, as técnicas, as métricas de avaliação e as ferramentas utilizadas no desenvolvimento dos modelos.

Os resultados obtidos foram apresentados na Seção 6. Nesta sessão foi feita a avaliação dos modelos treinados.

Por fim, na Seção 7, são apresentadas as conclusões e considerações finais sobre o estudo.

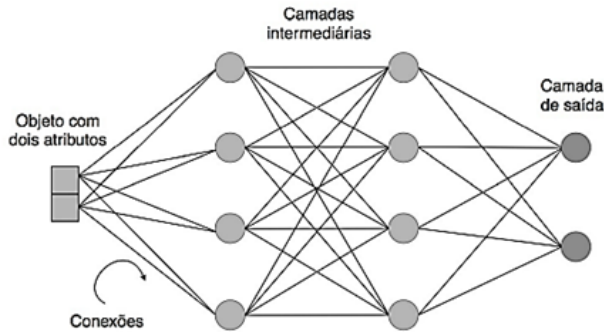
## 2 Fundamentação Teórica

### 2.1 Redes Neurais Artificiais

As RNAs são modelos computacionais de IA inspirados na estrutura neural do cérebro humano que adquirem conhecimento por meio da experiência. Neste contexto, é importante entender o funcionamento básico dessa estrutura a fim de facilitar a compreensão do funcionamento de uma RNA.

O sistema nervoso é composto por uma rede de neurônios que formam o cérebro, responsável por receber informações e tomar decisões. Essa rede é formada por neurônios interconectados, o que permite a comunicação entre eles. Essa comunicação é realizada por meio de impulsos elétricos que produzem uma substância neurotransmissora, transmitida do corpo celular para o axônio. Em síntese, segundo Cintra (2018), os principais componentes de um neurônio são os dendritos, o corpo celular, o axônio e as sinapses, conforme representado na Fig. 1.

Os dendritos são responsáveis por receber estímulos vindos de outros neurônios. O corpo celular, por sua vez, coleta e combina informações vindas de vários dendritos. O axônio é responsável por transmitir estímulos entre as células, enquanto as sinapses permitem a transmissão das informações entre os neurônios (Cintra, 2018).



**Figura 2:** MLP com duas camadas ocultas.  
Fonte: Faceli et al. (2011)

As RNAs, baseadas no modelo biológico, são compostas por unidades de processamento organizadas em uma ou mais camadas, conectadas por diversas conexões que possuem pesos para armazenar o conhecimento adquirido e medir a entrada recebida por cada neurônio na rede (Braga et al., 2000).

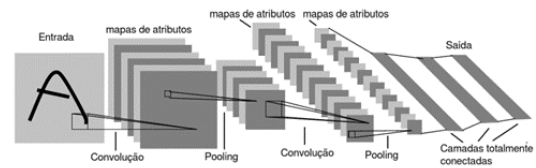
Assim, além das camadas de entrada e saída, as RNAs podem apresentar camadas intermediárias, conhecidas como camadas ocultas. Nelas, os neurônios recebem valores de saída de neurônios da camada anterior e enviam seus valores de saída para os terminais de entrada dos neurônios da camada posterior, formando um modelo de processamento de informação complexo e adaptativo. O objetivo do treinamento da RNA é encontrar os pesos ideais para que a rede possa realizar uma determinada tarefa de forma eficiente (Faceli et al., 2011). A Fig. 2 representa uma RNA *Multilayer Perceptron* (MLP), ilustrando a utilização de duas camadas ocultas.

Conforme Faceli et al. (2011), as RNAs são amplamente utilizadas para resolver problemas complexos devido às suas vantagens, como a capacidade de tolerar dados com ruído e a natural capacidade de serem paralelizáveis, o que pode acelerar o processo computacional. Elas ainda são capazes de aprender e se adaptar a partir dos dados de entrada, o que as tornam adequadas para tarefas de reconhecimento de padrões, classificação e previsão.

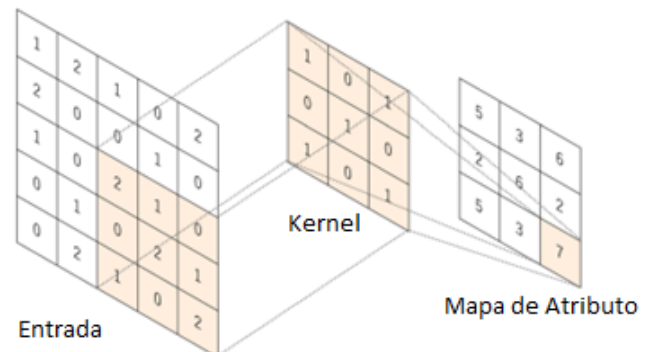
## 2.2 Redes Neurais Convolucionais e Aprendizado Profundo

O Aprendizado Profundo é uma técnica de AM baseada em RNAs com múltiplas camadas, conhecidas como Redes Neurais Profundas. Cada camada dessa rede é responsável por extrair características dos dados de entrada e transformá-los em representações abstratas. Essas camadas permitem que a rede aprenda características hierárquicas complexas a partir de dados brutos, permitindo que as RNAs realizem tarefas cada vez mais sofisticadas, como a classificação de imagens (Goodfellow et al., 2016).

As Redes Neurais Convolucionais – do inglês, *Convolutional Neural Networks* (CNN) – usadas na extração de característica de imagens, possuem uma arquitetura base que as diferem de outros modelos de RNAs, como por exemplo a Rede Neural MLP. A MLP é comumente usada para



**Figura 3:** Camadas de uma CNN.  
Fonte: LeCun et al. (1995)



**Figura 4:** Aplicação de filtro 3x3 em uma camada de convolução.

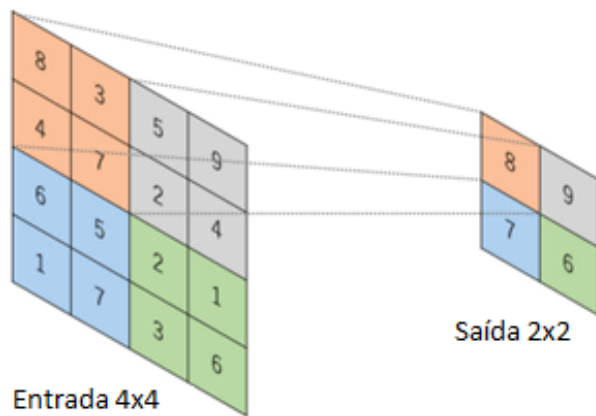
Fonte: Yamashita et al. (2018)

recursos de aprendizado simultâneo e classificação de dados. A principal limitação com o seu uso em imagens é que o número de neurônios pode ser muito alto, mesmo para arquiteturas rasas. A ideia básica por trás das CNNs é criar uma solução para reduzir o número de parâmetros, permitindo que uma rede seja mais profunda com menos parâmetros (Yamashita et al., 2018).

As CNNs, propostas pelos pesquisadores LeCun et al. (1995), são redes que realizam operação de convolução em pelo menos uma de suas camadas, aplicando filtros para extrair características locais dos dados ao qual são aplicadas. Uma CNN possui pelo menos três componentes básicos: camada de convolução, *pooling* e camada totalmente conectada. A Fig. 3 ilustra a arquitetura de uma CNN (LeCun et al., 1995).

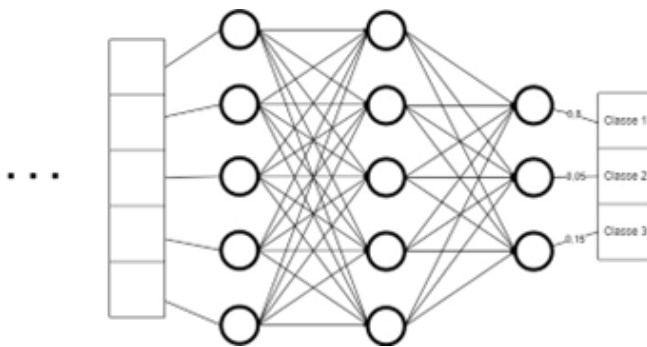
A camada de convolução é responsável por extrair características relevantes das imagens por meio da aplicação de filtros, ou *kernels*. Esses filtros são projetados para detectar padrões específicos nas imagens, como bordas, texturas e formas, e a aplicação repetida desses filtros em diferentes partes da imagem resulta em uma representação rica e hierárquica das características da imagem (Yamashita et al., 2018). A Fig. 4 ilustra uma camada de convolução com a aplicação de um filtro 3x3.

Já a camada de *pooling* tem como função reduzir a dimensionalidade das características extraídas pela camada de convolução, o que torna o processo de treinamento da rede mais eficiente. Isso é feito por meio da aplicação de uma operação de agregação em uma região local da imagem, que resulta em uma única representação daquela região. Esse processo reduz a quantidade de informação na rede, tornando-a mais eficiente e ajudando a evitar



**Figura 5:** Operação de *max pooling* em uma camada de pooling.

Fonte: Yamashita et al. (2018)



**Figura 6:** Rede totalmente conectada.

Fonte: Autor, 2023

problemas de *overfitting* (Yamashita et al., 2018). A Fig. 5 ilustra uma operação de *max pooling*.

Por fim, a rede totalmente conectada é a última camada das CNNs e é responsável por realizar a classificação dos dados de entrada após a extração das características relevantes pelas camadas anteriores. Nessa camada, as características são transformadas em uma representação vetorial unidimensional e passadas por uma ou mais camadas densas, que realizam uma combinação linear das características para gerar as saídas da rede (Yamashita et al., 2018). A Fig. 6 ilustra a camada totalmente conectada, última etapa para a classificação de imagens.

O processo de aprendizagem ocorre por meio do ajuste dos pesos das conexões entre as camadas. Os pesos são inicializados aleatoriamente e a rede é alimentada com um conjunto de dados de treinamento pré processado, envolvendo geralmente o redimensionamento e normalização das matrizes de imagens. Durante o treinamento, a CNN realiza a propagação dados de entrada para frente, passando-os pelas camadas convolucionais e de *pooling* para extrair características das imagens. As características são então alimentadas nas camadas totalmente conectadas para produzir a saída final. Ao calcular a diferença entre

as saídas previstas e as saídas esperadas, a CNN usa uma função de perda que mede a diferença entre as previsões e os rótulos reais. Em seguida, a retropropagação é usada para calcular o gradiente da função de perda em relação aos pesos da rede. Com o gradiente calculado, um algoritmo de otimização, como o *Stochastic Gradient Descent* (SGD), ajusta gradualmente os pesos da rede para minimizar a função de perda. Esse processo é repetido muitas vezes, usando um conjunto de treinamento rotativo para evitar *overfitting*. O objetivo da aprendizagem é minimizar a diferença entre as saídas previstas pela rede e as saídas esperadas. Obtendo um desempenho satisfatório para o cenário aplicado, a CNN pode ser usada para fazer previsões em novas imagens.

O processo de treinamento de uma CNN pode ser computacionalmente intensivo para conjuntos grandes e complexos, no entanto, uma vez treinada, a rede é capaz de classificar novas imagens com rapidez e precisão, tornando-a uma ferramenta útil em diversas aplicações. Além disso, é possível a utilização de técnicas como a Transferência de Aprendizagem, que permitem reutilizar parte do conhecimento adquirido durante o treinamento em outras tarefas, o que pode melhorar o desempenho da rede.

### 2.3 Transferência de Aprendizagem

A Transferência de Aprendizagem é uma estratégia utilizada no Aprendizado Profundo na qual modelos pré-treinados são empregados como ponto de partida para novos modelos. O aprendizado de uma RNA aplicada em tarefas de visão computacional geralmente requer uma grande quantidade de dados, mas nem sempre é possível obter acesso a conjuntos de dados extensos o suficiente para treinar os modelos. Nessa abordagem, os modelos pré-treinados são utilizados como extratores de características de imagens, eles são submetidos a grandes conjuntos de imagens pertencentes a diferentes classes. A Transferência de Aprendizagem permite aproveitar esse conhecimento prévio em tarefas relacionadas, reduzindo a quantidade de parâmetros e acelerando o tempo de treinamento (Yosinski et al., 2014).

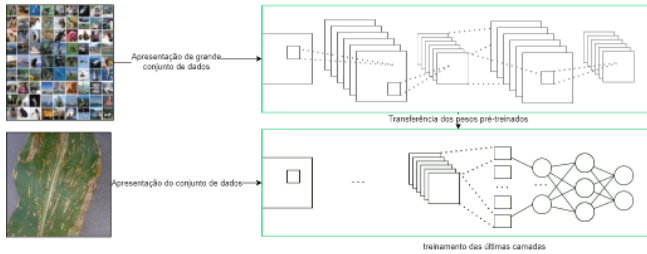
Neste estudo essa estratégia será aplicada ao copiar as primeiras "n" camadas da rede de origem para as primeiras "n" camadas da rede de destino. As camadas restantes da rede de destino terão neurônios com pesos inicializados aleatoriamente e serão treinadas para a tarefa de classificação, conforme ilustrado na Fig. 7.

Quando o conjunto de dados de destino é significativamente menor que o conjunto de dados base, como é o caso desse estudo, a Transferência de Aprendizagem pode ser uma ferramenta poderosa para permitir o treinamento, evitando o *overfitting* (Yosinski et al., 2014).

### 2.4 Doenças foliares do milho

Nos últimos anos as doenças têm se tornado uma grande preocupação por parte de técnicos e produtores envolvidos no agronegócio do milho. A evolução das doenças do milho está relacionada, entre outros fatores, à evolução do sistema de produção desta cultura, bem como modifica-





**Figura 7:** Exemplo de aplicação da Transferência de Aprendizagem.

Fonte: Autor, 2023



**Figura 8:** Ferrugem Comum.

Fonte: Hughes and Salathé (2014)

ções ocorridas no sistema de produção, que resultaram no aumento da produtividade da cultura (Costa et al., 2021). Dentre as doenças que atacam a cultura do milho estão a Ferrugem Comum, a Cercosporiose e a Helmintosporiose ou Mancha de Turcicum, que serão utilizadas no escopo deste estudo.

A Ferrugem Comum é caracterizada pela formação de pústulas. As pústulas da Ferrugem Comum apresentam formato circular a alongado e coloração castanho clara a escuro, que se acentua à medida em que as pústulas amadurecem e se rompem, liberando os uredósporos, que são os esporos típicos do patógeno (Costa et al., 2021). A Fig. 8 ilustra uma folha com a doença Ferrugem Comum.

A Cercosporiose é caracterizada por manchas de coloração cinza, predominantemente retangulares, com as lesões paralelas às nervuras. A Fig. 9 ilustra uma folha com a doença Cercosporiose (Costa et al., 2021).

Por fim, a Mancha de Turcicum se caracteriza por lesões necróticas, elípticas. A coloração do tecido necrosado varia de cinza a marrom e, no interior das lesões, observa-se intensa esporulação do patógeno, onde as primeiras lesões aparecem geralmente nas folhas mais velhas (Costa et al., 2021). A Fig. 10 ilustra uma folha com a doença Mancha de Turcicum.

Neste estudo serão avaliados modelos computacionais



**Figura 9:** Cercosporiose.

Fonte: Hughes and Salathé (2014)



**Figura 10:** Mancha de Turcicum.

Fonte: Hughes and Salathé (2014)

para classificar as plantas saudáveis e doentes, indicando a classe da doença pertencente.

### 3 Trabalhos Relacionados

Devido a característica da manifestação visual de doenças em plantas, a utilização de Aprendizado Profundo por meio de CNNs vem obtendo resultados promissores na detecção e classificação de imagens para a identificação de doenças foliares.

Mohanty et al. (2016), treinaram modelos baseados em CNN (GoogLeNet e AlexNet) para identificar 14 espécies de culturas e 26 doenças utilizando um conjunto de dados público de imagens de folhas de plantas, atingindo uma acurácia de 99,35% ao aplicar a Transferência de Aprendizagem com a rede GoogLeNet. No entanto, ao testar o modelo em imagens capturadas em condições diferentes da base de teste, a acurácia foi de 31%.

Sladojevic et al. (2016), utilizaram CNNs para reconhecer e classificar 13 tipos diferentes de doenças em 4 espécies de plantas a partir da imagem de suas folhas, alcançando acurácia média de 96,3%.

Liu et al. (2017), utilizaram CNNs derivadas de AlexNet para detectar doenças de folhas de macieira, usando um conjunto de dados de 13.689 imagens, atingindo uma acurácia média de 97,62%.

Ferentinos (2018), comparou modelos de CNNs para a classificação de doenças de plantas usando imagens de folhas saudáveis e doentes, aplicando as redes AlexNet, GoogLeNet, VGG16, Overfeat e AlexNetOWTBn. O melhor modelo (VGG16) obteve acurácia de 99,53%. Porém, a acurácia foi reduzida para 65,69% quando treinado com imagens coletadas em campo e testado com imagens de laboratório.

Zhang et al. (2018) propôs modelos derivados do GoogLeNet e Cifar10 para classificar oito tipos de doenças foliares do milho. Foram utilizadas 500 imagens coletadas de diferentes fontes. Com a aplicação de técnicas de aumento de dados e o ajuste de hiperparâmetros, o melhor modelo (GoogLeNet) atingiu uma acurácia máxima de 98,9%.

Waheed et al. (2020) propôs uma arquitetura otimizada de CNN densa (DenseNet) para a classificação de doenças foliares do milho. Utilizou-se no estudo um conjunto de 12.332 imagens, coletadas de diversas fontes, referentes a 4 classes. Após a aplicação de técnicas de aumento de dados, o modelo proposto atingiu uma acurácia de 98,06%.

Mishra et al. (2020) desenvolveram um modelo de CNN implantado em um Raspberry Pi 3 para a detecção em tempo real de doenças em plantas de milho, alcançando uma acurácia de 88,46%.

Jasrotia et al. (2023) desenvolveram um modelo de CNN para classificar doenças em plantas de milho, usando técnicas de pré-processamento como, por exemplo a transformação logarítmica e a conversão de RGB para HSV. O modelo treinado alcançou acurácia máxima de 99,76%.

Ashwini and Sellam (2024) propôs um modelo híbrido 3D-CNN-RNN para a classificação de doenças nas folhas do milho em dois conjuntos de dados distintos, alcançando uma acurácia superior a 94% em ambos os conjuntos de dados e desempenho acima de 90% em todas as métricas avaliadas.

Nos estudos relacionados, a Transferência de Aprendizagem tem sido amplamente utilizada. No entanto, o teste de alguns modelos em culturas específicas foi pouco explorado. Além disso, a falta de conjuntos de imagens diver-

sificados, limita a classificação quando aplicado a imagens distintas do conjunto de treinamento.

Este estudo propõe-se a avaliar modelos alternativos de CNNs utilizando Transferência de Aprendizagem, a fim de identificar aquele que apresenta melhor capacidade de generalização em doenças que afetam a cultura do milho.

### 4 Materiais e Métodos

#### 4.1 Proposta

O estudo consiste em avaliar o desempenho na classificação de doenças em imagens de folhas da cultura do milho, utilizando duas arquiteturas de CNNs: VGG19 e ResNet50. Para isso, serão realizados experimentos com um conjunto de dados composto por imagens de folhas de plantas com diferentes doenças foliares.

Para treinar os modelos, será aplicada a técnica de Transferência de Aprendizagem, que consiste em utilizar arquiteturas de RNAs pré-treinadas em bases de dados com grandes conjuntos de imagens e aproveitar os pesos aprendidos durante esse treinamento, conforme explicado na Seção 2.3.

O primeiro modelo será construído com base na arquitetura VGGNet, originalmente proposta por (Simonyan and Zisserman, 2014). A VGG19, uma das variações da VGGNet, é uma rede neural profunda composta por 19 camadas, organizadas em uma sequência de camadas convolucionais com filtros de tamanho 3x3 e por operações de *maxpooling*. Após cada camada convolucional, é aplicada a função de ativação ReLU. A cada duas camadas convolucionais ocorre uma operação de *maxpooling*, e o número de filtros é duplicado após cada uma dessas operações. A rede VGG foi pioneira ao utilizar filtros menores, de dimensão 3x3, em cada camada convolucional.

O segundo modelo terá como base a rede ResNet. A ResNet é uma arquitetura que tem como principal característica a incorporação de blocos residuais, que são módulos especiais adicionados às camadas convolucionais padrão. Esses blocos residuais permitem que a ResNet aprenda resíduos ou diferenças entre os recursos extraídos em cada camada. A ResNet utiliza conexões de atalho, onde o gradiente é diretamente retropropagado para camadas anteriores (He et al., 2016). A variação utilizada será a ResNet50, que possui 50 camadas.

#### 4.2 Conjuntos de Dados

O subconjunto utilizado foi proveniente do conjunto de dados *PlantVillage Dataset* (Hughes and Salathé, 2014), que contém aproximadamente 54 mil imagens rotuladas de folhas de plantas saudáveis e infectadas. Para o estudo, foram selecionadas imagens relacionadas a doenças do milho, divididas em quatro classes (Saudável, Ferrugem Comum, Cercosporiose e Mancha Turcicum), totalizando 3.838 imagens. As imagens foram separadas na proporção de 80% para treinamento e 20% para validação e teste, conforme mostrado na Tabela 1.

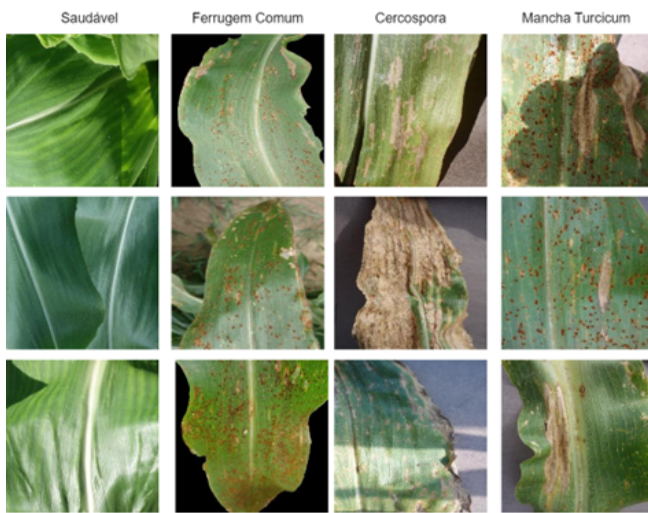
As imagens foram capturadas em condições diferentes de luminosidade, contraste e base de fundo. A Fig. 11 ilustra o conteúdo do conjunto de dados.



**Tabela 1:** Número de imagens do subconjunto de dados.

Categoria	Imagens	Treinamento	Validação
Saudável	1162	930	232
Ferrugem Comum	1192	953	239
Cercosporiose	509	406	103
Mancha Turcicum	975	778	197
<b>Total</b>	<b>3838</b>	<b>3067</b>	<b>771</b>

Fonte: Hughes and Salathé (2014)

**Figura 11:** Amostra de imagens do conjunto de dados.

Fonte: Hughes and Salathé (2014)

Por fim, foram aplicadas estratégias de aumento de dados, por meio da rotação e espelhamento do conjunto de imagens de treinamento.

### 4.3 Métricas de Avaliação

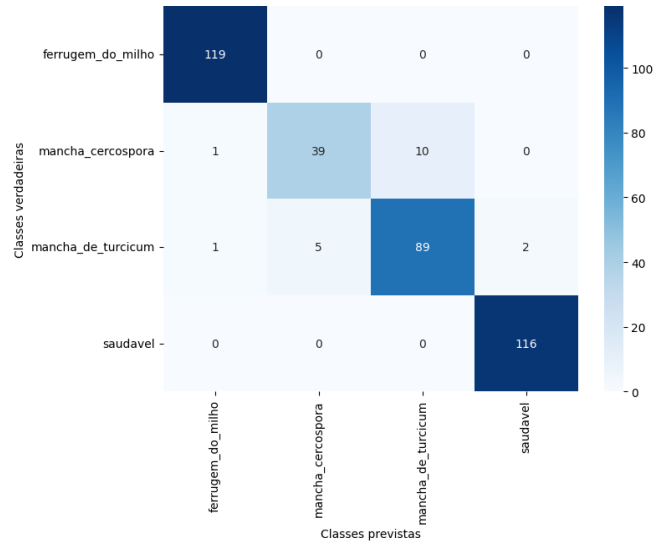
Para avaliar os modelos de classificação, foram utilizadas as métricas de acurácia, precisão, revocação e perda.

A acurácia mede a capacidade do modelo de classificar corretamente as instâncias em todas as classes, enquanto a precisão mede a proporção de classificações corretas para a classe positiva. Por sua vez, a revocação mede a proporção de instâncias da classe de interesse corretamente classificadas pelo modelo. As Eqs. (1) a (3) mostram as definições matemáticas para cada uma das métricas utilizadas neste estudo.

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN} \quad (1)$$

$$\text{Precisão} = \frac{VP}{VP + FP} \quad (2)$$

$$\text{Revocao} = \frac{VP}{VP + FN} \quad (3)$$

**Figura 12:** Exemplo de uma matriz de confusão.

Fonte: Autor, 2023

Nas equações acima, VP, VN, FP e FN correspondem aos valores verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos, respectivamente.

A perda representa a diferença entre as saídas do modelo e as saídas esperadas. O objetivo é minimizar o valor da perda, o que significa que o modelo está fazendo previsões mais precisas. Neste estudo a perda foi calculada por meio da métrica *Root Mean Squared Error* (RMSE), definido na Eq. (4).

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (4)$$

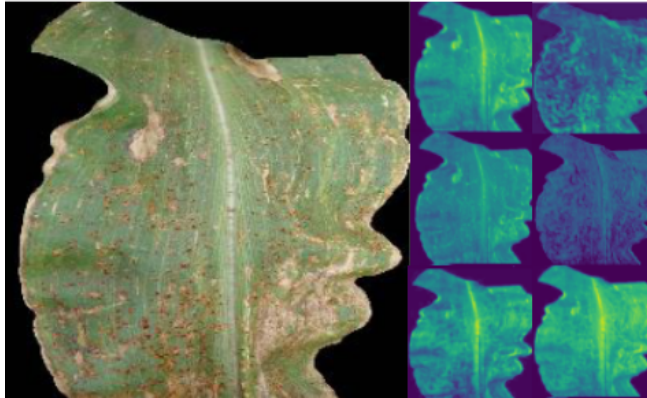
Ainda, a fim de analisar a assertividade de classificação dos modelos para cada uma das classes, foram geradas matrizes de confusão.

A matriz de confusão é uma tabela que apresenta as classificações corretas e incorretas feitas pelo modelo para cada classe. Ela nos permite visualizar de forma mais detalhada o desempenho do modelo em cada classe de interesse, ajudando a identificar possíveis padrões de erros de classificação. As células da matriz de confusão são preenchidas com os valores de VP, VN, FP e FN, conforme ilustrado na Fig. 12.

Também foram gerados mapas de atributos a fim de avaliar visualmente as principais regiões da imagem que os modelos aprenderam a distinguir durante o treinamento. A Fig. 13 ilustra os mapas de atributos de uma camada convolucional em uma imagem de folha doente.

Os mapas de atributos podem ser usados para interpretar o funcionamento de uma RNA e ajudar a entender como ela está tomando suas decisões.

Como cada camada de convolução apresenta diferentes quantidades de filtros, selecionou-se os aqueles com os maiores valores dentro de cada camada convolucional, representando as maiores ativações. As imagens geradas



**Figura 13:** Exemplo de geração de mapas de atributos.  
Fonte: Autor, 2023

foram normalizadas para o intervalo entre 0 e 1, onde 1 representa o maior valor possível.

#### 4.4 Tecnologias

Para o desenvolvimento dos modelos, foram utilizadas as bibliotecas Tensorflow, Keras e bibliotecas auxiliares, como Sklearn, OpenCV, Pandas, Numpy, Matplotlib, Seaborn e PIL. O ambiente de desenvolvimento utilizado para o treinamento dos modelos foi o *Google Colaboratory*. O *Google Colaboratory* é um ambiente de *Jupyter Notebook* que opera totalmente na nuvem, ele é pré empacotado com algumas das bibliotecas utilizadas no estudo, como o Tensorflow e o Keras (*Google Colab*, n.d.). As especificações do ambiente são mostradas na *Tabela 2*.

**Tabela 2:** Especificações do ambiente de desenvolvimento.

Nome	Especificação
CPU	Intel(R) Xeon(R) CPU @ 2.20GHz
GPU	Tesla T4 16GB
SO	Linux Ubuntu 22.04.2
Linguagem	Python 3.10.6

Fonte: *Google Colab* (n.d.)

A escolha do ambiente foi motivada pela disponibilidade de recursos de *hardware* especializado para a aceleração do treinamento dos modelos.

## 5 Configuração e Treinamento

Os modelos foram desenvolvidos com base nas arquiteturas VGG19 e ResNet50, conforme mencionado na *Seção 4.1*.

O conjunto de 3.838 imagens foi redimensionado para 256x256 pixels, normalizado e submetido a técnicas de aumento de dados, por meio da rotação (em até 40°) e espelhamento horizontal. Os pesos pré-treinados no conjunto de dados ImageNet foram empregados, e duas camadas totalmente conectadas foram adicionadas, cada uma com 64 neurônios. Além disso, entre as duas cama-

das totalmente conectadas, foi adicionada uma camada de *dropout* com uma taxa de 0,25. Por fim foi adicionada uma camada de saída com a aplicação da função *softmax* para 4 classes.

Ao longo de 50 épocas de treinamento, ajustou-se somente os pesos das camadas totalmente conectadas. A taxa de aprendizado de 0,01 foi utilizada nas primeiras 10 épocas, posteriormente sendo reduzida a uma taxa de aproximadamente 20% do valor a cada época, até que atingisse o valor mínimo de 0,00001, com o otimizador SGD. Os parâmetros utilizados são mostrados na *Tabela 3*. Os demais parâmetros mantiveram-se nos valores padrão fornecidos pela biblioteca Keras.

**Tabela 3:** Parâmetros utilizados no treinamento dos modelos.

Parâmetros	VGG19	Resnet50
Épocas de treinamento	50	50
Tamanho do lote	32	32
Taxa de aprendizagem	0,01, com decaimento	0,01, com decaimento
Otimizador	SGD	SGD
Tamanho da imagem	256x256x3	256x256x3
Aumento de dados	Sim	Sim

Fonte: Autor, 2023.

A seleção dos parâmetros foi fundamentada em um procedimento sistemático de experimentação empírica, onde diversas combinações foram avaliadas a fim de determinar a configuração que proporcionasse o melhor desempenho.

## 6 Resultados e Discussões

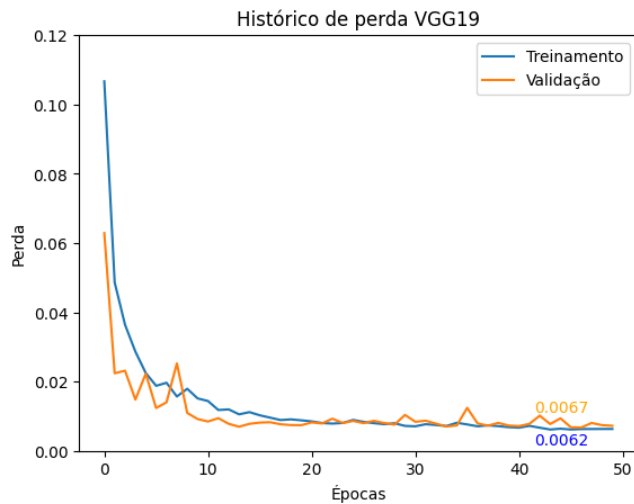
### 6.1 Comparação do desempenho dos modelos de classificação

Utilizou-se para o treinamento um subconjunto de imagens da base de dados *Plant Village*, especificamente relacionadas à cultura do milho. Para essa etapa, foi adotada uma divisão de 80% das imagens para treinamento e 20% para validação e teste, conforme mencionado na *Seção 4.2*.

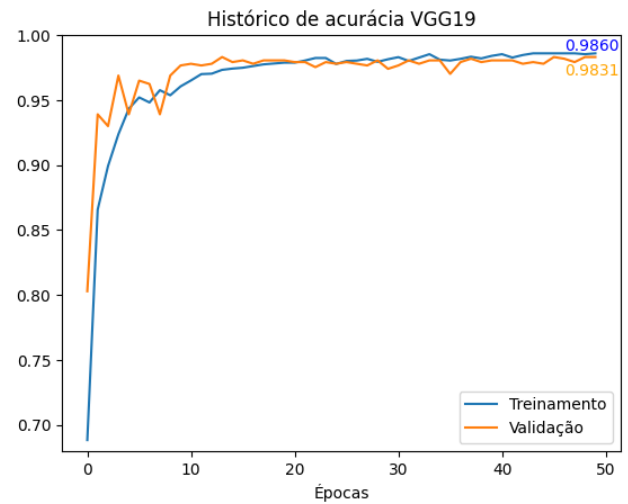
As *Figs. 14* e *15* ilustram a evolução da taxa de perda ao longo das 50 épocas de treinamento para os modelos VGG19 e ResNet50 respectivamente. Em ambas as arquiteturas, observa-se uma queda acentuada de perda nas etapas iniciais do treinamento, tornando-se mais gradual por volta da 20ª época. Observa-se ainda que, o modelo baseado na arquitetura ResNet50, atingiu um valor de perda menor durante o treinamento em comparação com o modelo baseado na arquitetura VGG19. No entanto, ao avaliar o desempenho na base de teste, o modelo ResNet50, exibiu um valor de perda maior, o que elevou a amplitude entre o valor de perda no treinamento e o valor de perda no teste. Isso pode indicar que, o modelo VGG19 obteve uma melhor capacidade de generalização.

A curva de acurácia dos modelos, ilustrada nas *Figs. 16* e *17*, apresenta comportamento semelhante, aumentando até a vigésima época e mantendo-se estável a partir desse ponto. Nota-se ainda que, a rede VGG19 demonstra um de-

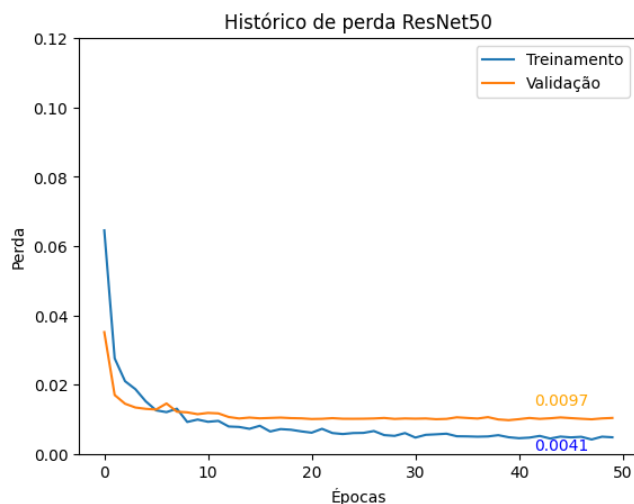




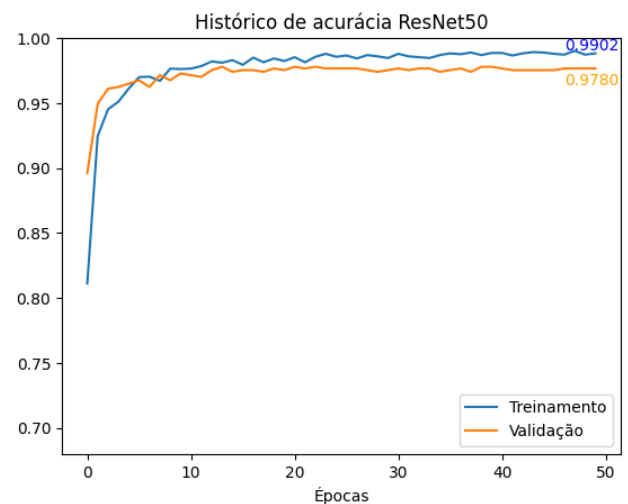
**Figura 14:** Histórico de perda por época VGG19.  
Fonte: Autor, 2023



**Figura 16:** Histórico de acurácia por época ResNet50.  
Fonte: Autor, 2023



**Figura 15:** Histórico de perda por época ResNet50.  
Fonte: Autor, 2023



**Figura 17:** Histórico de acurácia por época ResNet50.  
Fonte: Autor, 2023

sempenho superior na base de teste, quando comparada ao modelo ResNet50. A maior acurácia obtida na base de teste foi por meio da rede VGG19, chegando a 98,31%, frente a 97,80% da ResNet50.

A fim de analisar a assertividade de classificação dos modelos para cada uma das classes, foram geradas matrizes de confusão.

Observa-se nas matrizes de confusão (Figs. 18 e 19) que os modelos classificaram corretamente todas as amostras para as classes Ferrugem do Milho e Saudável. Para as demais classes observa-se que houve falsos positivos. Esses falsos positivos, caracterizados por erros de classificação entre essas duas categorias, sugerem a que o modelo considerou que há características visuais semelhantes entre as doenças Mancha Cercospora e Mancha de Turcicum. Observa-se também que a classe Mancha Cercospora apre-

senta uma quantidade substancialmente menor de amostras de dados, o que pode ter contribuído para o menor desempenho na classificação.

Alguns estudos destacam a escassez de conjuntos de dados de imagens de plantas suficientemente abrangentes que possibilitem uma boa generalização. Essa constatação ressalta a necessidade de aprimorar a diversidade e a representatividade do conjunto de dados para melhorar a capacidade de generalização dos modelos de classificação. A inclusão de mais exemplos e variações nas condições de captura das imagens pode fornecer uma base mais sólida para o treinamento dos modelos e melhorar seu desempenho em cenários reais.

Ao analisar as demais métricas de avaliação, evidencia-se que ambos os modelos não apenas exibem uma acurácia elevada, mas também demonstram um padrão semelhante

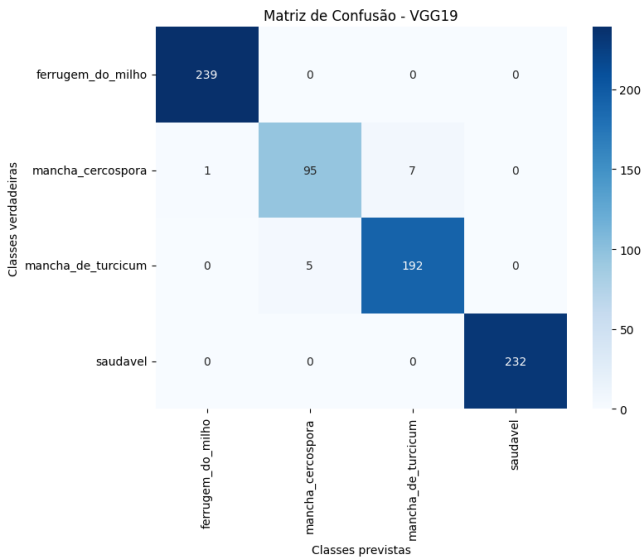


Figura 18: Matriz de confusão com a assertividade de classificação de cada classe pelo modelo VGG19.  
Fonte: Autor, 2023

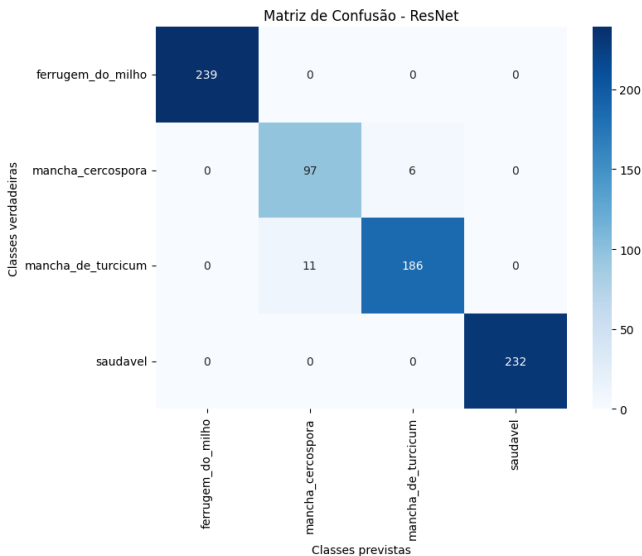


Figura 19: Matriz de confusão com a assertividade de classificação de cada classe pelo modelo ResNet50.  
Fonte: Autor, 2023

de precisão e revocação. Isso indica que ambos modelos são capazes de classificar as instâncias na base de dados utilizada. A VGG19 apresenta pequena vantagem em termos de acurácia, registrando 98,31%, em comparação com os 97,80% da ResNet50. O mesmo se repete para as métricas de precisão e revocação média. A ResNet50 apresentou uma precisão de 96,75% e revocação de 97,00%, enquanto a VGG19 apresentou valores de precisão e revocação de 97,75% e 97,25%, respectivamente, conforme mostrado na Tabela 4.

A análise das matrizes de confusão reforçou os desafios

Tabela 4: Comparativo de acurácia, precisão e revocação dos modelos VGG19 e ResNet50.

Métrica / Modelo	VGG19	ResNet50
Acurácia	98,31%	97,80%
Precisão	97,75%	96,75%
Revocação	97,25%	97,00%

Fonte: Autor, 2023.

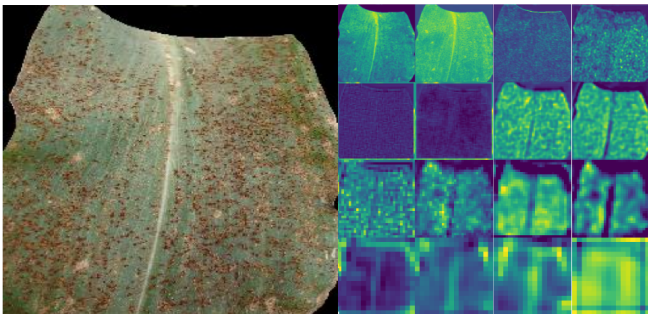


Figura 20: Mapas de atributos VGG19 para a doença Ferrugem Comum.  
Fonte: Autor, 2023

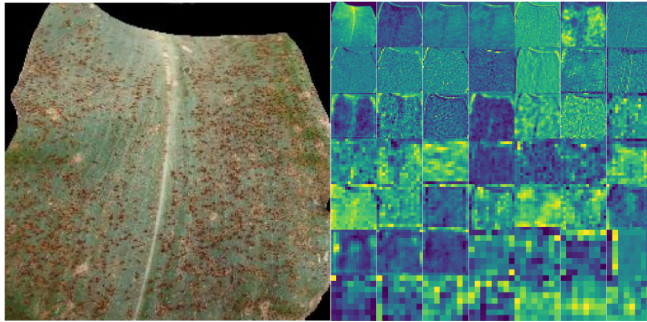
em distinções entre doenças visualmente similares, sugerindo oportunidades de aprimoramento do conjunto de dados.

6.2 Visualização dos mapas de atributos

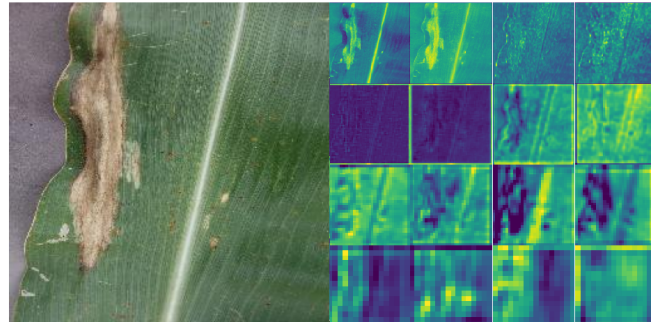
Para visualizar as regiões utilizadas pelas redes para distinguir áreas doentes em imagens e classificar as doenças, realizou-se a obtenção dos mapas de atributos, como descrito na Seção 4.3.

As Figs. 20 e 21 ilustram as visualizações dos mapas de atributos da rede VGG19 e ResNet50, respectivamente, para uma imagem que contém sintomas da doença Ferrugem Comum. Em cada camada convolucional, selecionou-se o mapa com os maiores valores de ativação da rede. As visualizações dos mapas de calor revelaram que os sintomas visuais são identificados e destacados frente o restante da folha, principalmente nas primeiras camadas convolucionais da rede. No entanto, observa-se que, devido a diminuição da resolução, os últimos mapas de atributos tendem a apresentar visualizações com menos nitidez nas ativações das áreas dos sintomas. Isso indica que, como esperado, as primeiras camadas capturaram características mais perceptíveis ao humano, como bordas e texturas, enquanto as camadas posteriores desenvolveram uma representação mais abstrata e complexa das características da imagem.

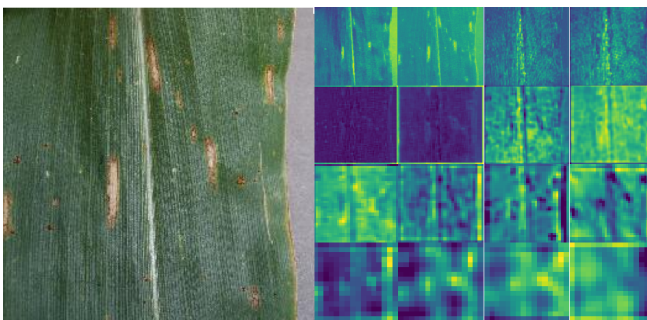
O mesmo é observado nas Figs. 22 e 23, que ilustram as visualizações dos mapas de atributos para uma imagem que contém sintomas da doença Cercospora, e nas Figs. 24 e 25, que ilustram as visualizações dos mapas de atributos para uma imagem que contém sintomas da doença Mancha Turcicum, onde as regiões doentes são destacadas das demais regiões da folha. Observa-se ainda que a nervura



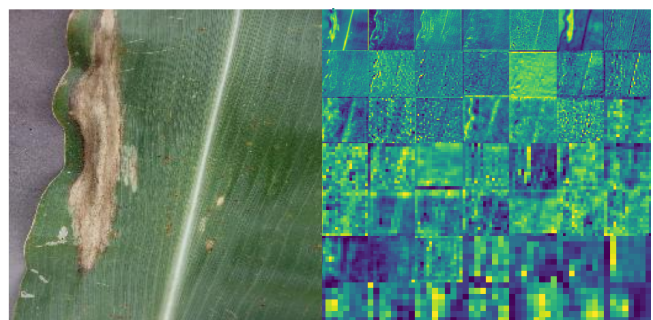
**Figura 21:** Mapas de atributos ResNet50 para a doença Ferrugem Comum.  
Fonte: Autor, 2023



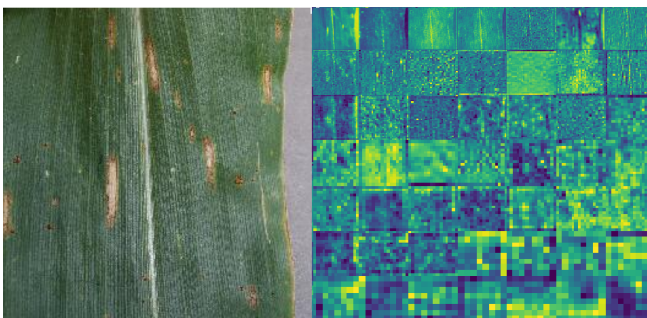
**Figura 24:** Mapas de atributos VGG19 para a doença Mancha Turcicum.  
Fonte: Autor, 2023



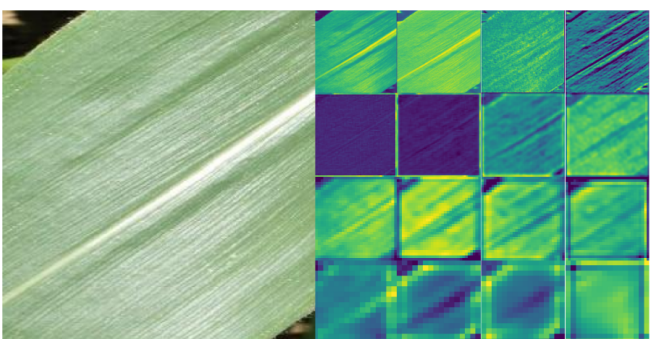
**Figura 22:** Mapas de atributos VGG19 para a doença Cercosporiose.  
Fonte: Autor, 2023



**Figura 25:** Mapas de atributos ResNet50 para a doença Mancha Turcicum.  
Fonte: Autor, 2023



**Figura 23:** Mapas de atributos ResNet50 para a doença Cercosporiose.  
Fonte: Autor, 2023



**Figura 26:** Mapas de atributos VGG19 para uma folha saudável.  
Fonte: Autor, 2023

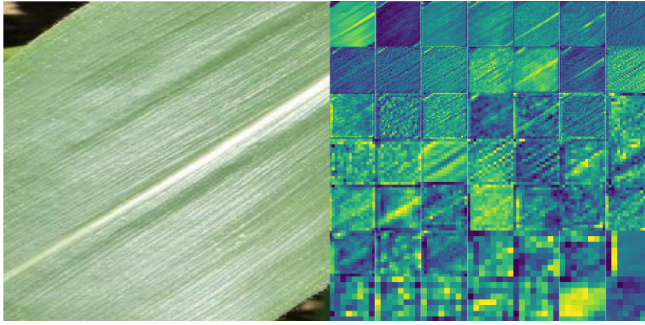
principal da folha também recebe destaque nos mapas.

Para uma imagem da classe Saudável, os mapas de atributos exibem maior homogeneidade e não apresentam regiões de ativação contrastantes, exceto pela ativação na nervura principal da folha, como nas demais classes. A [Fig. 26](#) e [Fig. 27](#) ilustram os mapas de atributos para uma folha saudável.

Os mapas de atributos mostraram que ambas as redes treinadas foram capazes de identificar as características discriminativas em pontos correspondentes as doenças nas folhas do milho. Observou-se convergência das características identificadas pelas redes com as regiões afetadas

pelas doenças foliares do milho. Observou-se ainda que a nervura da folha foi considerada importante para o modelo, trabalhos futuros podem concentrar-se em treinar os modelos com imagens que representam apenas a região doente da folha a fim de avaliar os mapas de atributos gerados pela rede.





**Figura 27:** Mapas de atributos ResNet50 para uma folha saudável.

Fonte: Autor, 2023

## 7 Considerações Finais

Com o objetivo de classificar doenças foliares da cultura do milho, foram desenvolvidos dois modelos, usando como base as arquiteturas VGG19 e ResNet50. Os resultados indicaram que ambas as redes obtiveram desempenhos similares, com a VGG19 alcançando 98,31% de acurácia em comparação com 97,80% da ResNet50. Os resultados mostraram que a rede baseada na arquitetura VGG19 apresentou melhor capacidade de generalização.

A visualização dos mapas de atributos das redes, reforçou a capacidade dos modelos em discernir os pontos de manifestação das doenças em contraste com o restante da folha, onde as ativações relacionadas as áreas afetadas pela doença foram mais evidentes nas primeiras camadas de convolução.

Assim, conclui-se que a utilização das arquiteturas VGG19 e ResNet50 por meio da Transferência de Aprendizagem podem ser uma alternativa viável para a classificação de doenças foliares na cultura do milho. No entanto, é importante ressaltar a necessidade da construção de conjuntos de dados abrangentes e representativos para aprimorar ainda a capacidade de generalização desses modelos, visando uma detecção precisa e uma aplicabilidade mais abrangente na detecção e classificação de doenças nas plantações de milho.

Trabalhos futuros poderão se concentrar na obtenção bases de dados mais abrangentes, com diferentes formas de manifestação das doenças, possibilitando o desenvolvimento de modelos mais robustos para detectar e classificar as doenças do milho em campo. Além disso, poderão focar na aplicação do modelo para a identificação de folhas saudáveis e doentes e na classificação de doenças em culturas não abordadas nesse trabalho.

## Referências

- Arnal Barbedo, J. G. (2013). Digital image processing techniques for detecting, quantifying and classifying plant diseases, *SpringerPlus* 2(1). <https://doi.org/10.1186/2193-1801-2-660>.
- Ashwini, C. and Sellam, V. (2024). An optimal model for identification and classification of corn leaf disease using hybrid 3d-cnn and lstm, *Biomedical Signal Processing and Control* 92: 106089. <https://doi.org/10.1016/j.bspc.2024.106089>.
- Bock, C., Poole, G., Parker, P. and Gottwald, T. (2010). Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging, *Critical reviews in plant sciences* 29(2). <https://doi.org/10.1080/07352681003617285>.
- Braga, A. d. P., Ludermir, T. B. and Carvalho, A. C. P. d. L. F. (2000). *Redes neurais artificiais: teoria e aplicações*, LTC.
- Cintra, R. (2018). *Introdução a neurocomputação*, INPE, São Paulo. Disponível em [http://www.inpe.br/elac2018/arquivos/ELAC2018\\_MC3\\_apostila.pdf](http://www.inpe.br/elac2018/arquivos/ELAC2018_MC3_apostila.pdf).
- Costa, R. V., Casela, C. R. and Cota, L. V. (2021). Milho. doenças foliares., *Agência de Informação Tecnológica*. Disponível em <https://www.embrapa.br/en/agencia-de-informacao-tecnologica/cultivos/milho/producao/pragas-e-doencas/doencas/doencas-foliares>.
- Faceli, K., Lorena, A. C., Gama, J. and Carvalho, A. C. P. d. L. F. d. (2011). Inteligência artificial: uma abordagem de aprendizado de máquina.
- Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis, *Computers and Electronics in Agriculture* 145. <https://doi.org/10.1016/j.compag.2018.01.009>.
- Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep learning*, MIT press.
- Google Colab (n.d.). Disponível em <https://colab.research.google.com>.
- He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*. 10.1109/CVPR.2016.90.
- Hughes, D. P. and Salathé, M. (2014). An open access repository of images on plant health to enable the development of mobile disease diagnostics, *PLoS ONE* 9(6). Disponível em <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0105600>.
- Jasrotia, S., Yadav, J., Rajpal, N., Arora, M. and Chaudhary, J. (2023). Convolutional neural network based maize plant disease identification, *Procedia Computer Science* 218: 1712–1721. <https://doi.org/10.1016/j.procs.2023.01.149>.
- LeCun, Y., Bengio, Y. et al. (1995). Convolutional networks for images, speech, and time series, *The handbook of brain theory and neural networks* 3361(10).
- Liu, B., Zhang, Y., He, D. and Li, Y. (2017). Identification of apple leaf diseases based on deep convolutional neural networks, *Symmetry* 10(1). <https://doi.org/10.3390/sym10010011>.
- Mishra, S., Sachan, R. and Rajpal, D. (2020). Deep convolutional neural network based detection system for real-time corn plant disease recognition, *Procedia Computer Science* 167: 2003–2010. <https://doi.org/10.1016/j.procs.2020.03.236>.

- Mohanty, S. P., Hughes, D. P. and Salathé, M. (2016). Using deep learning for image-based plant disease detection, *Frontiers in plant science* 7. <https://doi.org/10.3389/fpls.2016.01419>.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556*. <https://doi.org/10.48550/arXiv.1409.1556>.
- Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D. and Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification, *Computational intelligence and neuroscience* 2016. <https://doi.org/10.1155/2016/3289801>.
- Waheed, A., Goyal, M., Gupta, D., Khanna, A., Hassanien, A. E. and Pandey, H. M. (2020). An optimized dense convolutional neural network model for disease recognition and classification in corn leaf, *Computers and Electronics in Agriculture* 175. <https://doi.org/10.1016/j.compag.2020.105456>.
- Yamashita, R., Nishio, M., Do, R. K. G. and Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology, *Insights into imaging* 9. <https://doi.org/10.1007/s13244-018-0639-9>.
- Yang, X. and Guo, T. (2017). Machine learning in plant disease research, *March* 31. <http://dx.doi.org/10.18088/ejbmr.3.1.2017.pp6-9>.
- Yosinski, J., Clune, J., Bengio, Y. and Lipson, H. (2014). How transferable are features in deep neural networks?, *Advances in neural information processing systems* 27. <https://doi.org/10.48550/arXiv.1411.1792>.
- Zhang, X., Qiao, Y., Meng, F., Fan, C. and Zhang, M. (2018). Identification of maize leaf diseases using improved deep convolutional neural networks, *Ieee Access* 6. <https://doi.org/10.1109/ACCESS.2018.2844405>.