UNIVERSIDADE FEDERAL RURAL DO SEMIÁRIDO - UFERSA
CURSO DE BACHARELADO EM CIÊNCIA E TECNOLOGIA
Trabalho de Conclusão de Curso (2023).

REDES NEURAIS CONVOLUCIONAIS NA AGRICULTURA: DETECÇÃO DE DOENÇAS FÚNGICAS EM PLANTAS DE UVA.

Anderson Matheus Melo da Silva[1], Matheus da Silva Menezes[2]

Resumo: Este artigo apresenta um estudo sobre a utilização de redes neurais convolucionais (CNN) para identificar e classificar doenças fúngicas em plantas, com foco específico nas doenças foliares da planta da uva. O modelo de CNN foi treinado e avaliado utilizando um conjunto de dados das doenças *Black Rot*, *ESCA*, *Leaf Blight* e folhas saudáveis. Os resultados mostraram que o modelo foi capaz de aprender de forma eficiente as características distintivas associadas a cada classe durante o treinamento, resultando em uma alta acurácia na classificação das doenças foliares da uva. Essa abordagem promissora tem implicações significativas na agricultura, oferecendo uma ferramenta automatizada para a detecção precoce de doenças, possibilitando a implementação de medidas de controle adequadas e reduzindo as potenciais perdas de colheita.

Palavras-chave: Redes neurais artificiais, Redes neurais convolucionais; Doenças fúngicas em plantas; Identificação de características.

1. INTRODUÇÃO

Com o aumento da demanda global por alimentos e a importância da sustentabilidade na produção agrícola, torna-se essencial a utilização de tecnologias avançadas para detectar e prevenir doenças em plantas [1]. A família *Vitaceae* é composta por 15 gêneros e cerca de 900 espécies, dentre elas, a *Vitis vinifera*, é a que possui a maior variedade, sendo assim cultivada por seus frutos de alta qualidade tendo um papel importante na produção de vinhos [2].

No entanto, essas variedades são mais suscetíveis a estresses bióticos (danos causados por fungos, insetos, nematóides, bactérias, etc.) e abióticos (por exemplo, temperatura, metais pesados, salinidade, etc.), o que pode resultar em rendimentos significativamente menores [2]. Um exemplo disso é a luta enfrentada pelos vinhedos na Europa contra a podridão negra das uvas, causada pelo fungo *Guignardia bidwellii*, que apesar de ser conhecido há muito tempo na literatura científica, é relativamente nova na prática [2]. O fungo foi identificado na América do Norte em 1853 e se espalhou para a Europa no final do século XIX. Vinhedos que forem fortemente infectados podem perder um valor de até 100% da produção [2].

Identificar precocemente doenças fúngicas pode ser crucial para prevenir a disseminação dessas doenças para outras áreas da plantação, além de ajudar a determinar o tipo de ação preventiva que deve ser tomada. No caso de cultivos convencionais, o uso de fungicidas para doenças comuns pode ser eficaz no controle da podridão negra. Entretanto, em cultivos orgânicos, é comum evitar o uso de produtos químicos na agricultura, o que torna a abordagem baseada em produtos químicos menos utilizada e incentiva o uso de práticas agrícolas alternativas para prevenção e controle de doenças [2].

Algoritmos são amplamente utilizados para identificar e prever doenças em plantas, e podem ser divididos em três categorias: a primeira é baseada em condições climáticas e de temperatura, a segunda em processamento de imagem, que é a mais comumente utilizada; e a terceira é baseada em diferentes tipos de dados provenientes de fontes heterogêneas [2].

A inteligência artificial é uma área da computação que se dedica ao desenvolvimento de sistemas que se assemelham ao comportamento humano. Os sistemas de inteligência artificial são capazes de realizar tarefas que exigem a percepção, raciocínio e aprendizado, tornando-se cada vez mais importantes em diversas áreas, incluindo a agricultura. Em relação à identificação e prevenção de doenças em plantas, a inteligência artificial pode ser utilizada para analisar grandes quantidades de dados e identificar padrões que possam indicar a presença de uma doença. Além disso, a inteligência artificial pode ser utilizada para prever o surgimento de doenças com base em fatores como clima, temperatura e outras variáveis ambientais [3]

Uma técnica de inteligência artificial amplamente utilizada que foi inspirado no córtex visual de animais é a Rede Neural Convolucional. Atualmente, seu uso se expandiu para outras áreas além do reconhecimento de objetos, como rastreamento de objetos, estimativa de pose, detecção e reconhecimento de texto, detecção de saliência visual, reconhecimento de ação, rotulagem de cena e muitos outros [4].

Neste estudo, é avaliada a possibilidade de utilizar um modelo de rede neural convolucional para classificar diferentes tipos de doenças fúngicas e auxiliar na escolha de uma ação mais eficaz contra essas doenças. O artigo é dividido em mais quatro seções, onde na seção 2 tem-se a fundamentação teórica a respeito de redes neurais

artificiais e redes neurais convolucionais, na seção 3 está a metodologia do trabalho apresentando um descritivo do ambiente computacional, do banco de dados e da rede desenvolvida neste artigo, na seção 4 está o teste e a avaliação da rede neural convolucional criada, na seção 5 estão as considerações finais a respeito do tema e do objetivo do trabalho e na seção 6 as referências utilizadas.

2. FUNDAMENTAÇÃO TEÓRICA

Para uma melhor compreensão do conteúdo, é recomendado que o leitor esteja familiarizado com o funcionamento básico de redes neurais. Fontes como [1], [2] e [4] são sugeridas para consulta. Este trabalho pode interessar a estudantes, professores e pesquisadores, tanto da área da computação e tecnologia, quanto da área da agronomia e biologia, já que tratará de uma aplicação que correlaciona ambas as áreas e conhecimentos prévios sobre o assunto pode levar a melhor compreensão do conteúdo apresentado.

Uma Rede Neural Artificial é uma forma de aprendizagem de máquina que busca simular, através de cálculos matemáticos, o comportamento de neurônios biológicos. As redes neurais são métodos auto adaptativos baseados em dados, ajustando-se aos dados sem qualquer especificação funcional ou forma distribucional sendo dita ao modelo de forma explícita [5].

De forma análoga a um neurônio, uma RNA é um modelo matemático complexo que realiza cálculos a fim de resolver um problema. Há aplicações de Redes Neurais Artificiais em reconhecimento de fala, através do reconhecimento da frequência e intensidade na pronúncia de determinada frase, detecção de fraudes em sistemas financeiros ao se identificar compras ou transações incomuns, e, classificação e reconhecimento em imagens, como identificação de objetos específicos em uma foto ou vídeo. As redes neurais se tornaram uma ferramenta importante para a classificação. A vasta pesquisa recente na área de classificação neural mostrou que as redes neurais artificiais são uma alternativa promissora a vários métodos convencionais de classificação [5]. Formado por pesos, chamados de neurônios em referência à contraparte biológica em que se inspira, este tipo de modelo utiliza algoritmos de otimização que buscam diminuir o erro entre o valor calculado e o valor esperado. É através desta diminuição do erro que este modelo simula a aprendizagem e atualiza os valores de seus pesos, a rede então é um modelo matemático que se adapta aos dados [5]. Existem diferentes modelos de redes neurais artificiais que são classificados de acordo com sua arquitetura, uma destas arquiteturas é chamada de Rede Neural Convolucional, que é especialmente adequada para processar dados de imagem.

2.1 Redes Neurais Convolucionais

Uma CNN, *Convolutional Neural Network*, ou, rede neural convolucional é capaz de extrair características de uma imagem através do processo de aprendizagem e diminuição do erro que uma rede neural artificial efetua. Este tipo de rede teve como inspiração o campo receptivo visual. A matriz de imagem de entrada flui para cada camada de neurônios na CNN e, após sair, torna-se a matriz de mapa de características. Cada valor de pixel no mapa de características é mapeado por uma área de tamanho correspondente na imagem original, que é chamada de campo receptivo. Em outras palavras, os valores de pixel dentro de uma determinada área da matriz de entrada determinam em conjunto o valor de pixel de um determinado ponto da matriz de saída; o valor de pixel de um determinado ponto da matriz de saída é influenciado por todos os pixels na área do campo receptivo da imagem de entrada [6].

Antes de treinar um modelo de rede neural convolucional, é realizada a etapa de pré-processamento das imagens, com o objetivo de melhorar a consistência das imagens no conjunto de dados, preparando-as para serem processadas pelo classificador CNN de forma mais eficiente. Essa etapa de pré-processamento é fundamental para garantir a qualidade e a padronização das imagens, o que contribui para o desempenho geral do modelo durante o treinamento e a classificação posterior. Um pré-processamento bastante utilizado é normalização dos valores da imagem, para que se encontrem em valores entre 0 e 1, tornando mais eficiente de se realizar cálculos computacionais. Outro pré-processamento que pode ser utilizado para diminuir a complexidade dos dados e consequentemente do modelo é a transformação das imagens para uma escala de cinza [7].

A rede neural convolucional é um modelo especial de rede neural profunda. Suas conexões não são completamente conectadas, já que o valor de um pixel na imagem de saída não é calculado a partir de todos os pixels da imagem de entrada, mas sim, de um subgrupo de pixels determinados pelo campo receptivo. A influência dos valores de cada pixel dentro do campo receptivo não é necessariamente uniforme, pois é adicionado um peso à influência de cada pixel no campo receptivo, aumentando ou diminuindo a influência de um pixel. Adicionar peso de influência para cada pixel do campo receptivo é um processo matemático realizado pelo *kernel* de convolução, núcleo do processo de convolução deste tipo de rede. O *kernel* e o peso que ele atribui para cada pixel dentro do campo receptivo é compartilhado para a geração de mapas de características em uma mesma camada e os valores do *kernel* convolucional são obtidos através de um processo de aprendizagem com os dados, ao se computar o erro ao final de uma etapa do treinamento atualiza-se os valores dos pesos de cada *kernel* através de funções de minimização de erro [6].

2.2 Camadas de uma Rede Neural Convolucional

Um modelo de rede neural é definido levando em conta o objetivo da aplicação, o tipo de dados utilizado e a disponibilidade de recursos computacionais. Redes Neurais Convolucionais apresentam uma arquitetura geral formada por uma camada de entrada, camada convolucional, camada de *pooling*, camada de achatamento, camada totalmente conectada e a camada de saída.

2.2.1 Camada de Entrada

A camada de entrada é definida de acordo com o formato dos dados de entrada. É preciso que a camada de entrada esteja preparada para receber os dados da rede de forma correta, pois a incompatibilidade entre o formato da imagem de entrada e o formato definido pela camada de entrada irá definir se será possível ou não realizar a inferência com a rede para esta determinada imagem [8]. As demais camadas irão realizar as operações de forma a relacionar a informação de entrada com um resultado de saída que será comparado à saída esperada.

2.2.2 Camada Convolucional

A camada de convolução é composta por uma quantidade de *kernels*, que são os filtros convolucionais que irão realizar um produto, elemento a elemento, com uma área da imagem de entrada. Esta área é definida pelas dimensões do campo perceptível analisado na imagem para a geração de um pixel de saída que vai compor a imagem de saída chamada de mapa de características. Para realizar o produto elemento a elemento, é necessário que este campo perceptível analisado tenha dimensões iguais ao do *kernel* utilizado. A operação de convolução segue com o campo perceptível percorrendo áreas da imagem, gerando os demais pixels do mapa de característica [9].

2.2.3 Camada de *Pooling*

A camada de *pooling* é uma camada usada para reduzir a dimensionalidade dos dados, tornando-os mais fáceis de serem processados. Geralmente esta camada é colocada após a camada de convolução, reduzindo então o tamanho dos mapas de características gerados pela convolução, o que resulta em um menor número de parâmetros a serem passados adiante na rede. A redução de dimensões pode ser feita a partir de diferentes métodos, sendo mais frequentes a operação de *max-pooling*, que atribui ao pixel de saída o valor mais alto da região analisada pelo campo perceptivo analisado a partir do mapa de características de entrada, e, a operação de *average-pooling*, que atribui ao pixel de saída a média dos valores encontrados na janela de entrada analisada.

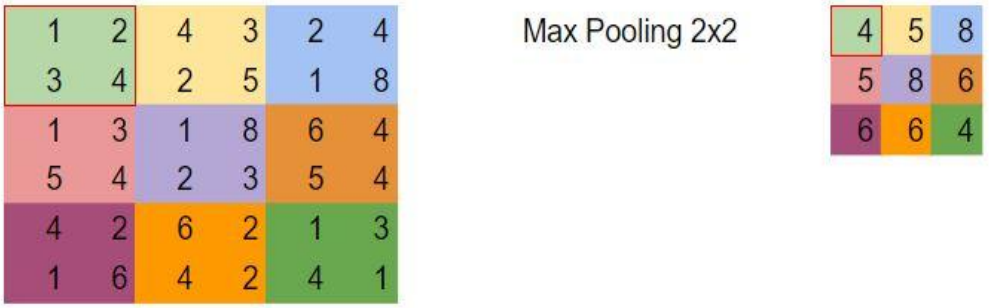

Figura 2. *Max Pooling* 2x2 em uma matriz 6x6. (Autoria própria)

A diminuição de dimensões feita pela camada de *pooling* causa uma diminuição da complexidade computacional. Este fator permite que camadas subsequentes utilizem valores superiores de parâmetros (*kernels* ou neurônios, dependendo do tipo de camada subsequente) possibilitando mais características aprendidas pela rede sem que a complexidade computacional sofra um aumento proporcional à quantidade de parâmetros, pois os dados foram reduzidos [9].

2.2.4 Camada de Achatamento

Após as camadas convolucionais e de *pooling*, geralmente há uma camada de "*flatten*" (achatamento), que transforma a saída da camada anterior em um vetor unidimensional. Essa camada é importante para preparar os dados para as camadas totalmente conectadas, que realizam o raciocínio de alto nível na rede neural convolucional. Os operadores de convolução e *pooling* extraem as características da imagem de entrada, que são então achatadas em um vetor e alimentadas para a primeira camada totalmente conectada. Cada neurônio da camada totalmente conectada está conectado a todos os neurônios da camada anterior, permitindo que a rede

neural convolucional aprenda as relações entre as características extraídas das camadas anteriores. O tamanho da camada de *flatten* é determinado pelo tamanho da saída da última camada de *pooling*. Depois da camada totalmente conectada, a saída é alimentada para a camada de saída para o reconhecimento e classificação de imagens. O número de neurônios na camada de saída é igual ao número de classes a serem classificadas. [9]

2.3 Função de Ativação

Para que modelos de redes neurais sejam capazes de correlacionar dados de forma não linear, são utilizadas as funções de ativação. A função de ativação é uma operação não linear que é aplicada às saídas das camadas convolucionais e totalmente conectadas a fim de permitir que a rede neural solucione problemas não lineares.

2.3.1 Função *ReLU*

Em redes neurais modernas, a função de ativação preferível é a unidade linear retificada (ou *rectified linear unit*, *ReLU*). A função *ReLU* é utilizada pois apresenta um tempo de treinamento baixo em comparação a outras funções. Sua aplicação transforma todos os valores menores que zero em zero, fazendo com que a informação de entrada não seja utilizada no modelo naquela execução. E para os demais valores acima de zero, ela retorna o próprio valor, de forma linear [8,9]. Veja a Equação 1 que apresenta o cálculo da função *ReLU:*

$$f(u) = \{u,\ u > 0\ ;\ 0,\ u \leq 0\} \tag{1}$$

2.3.2 Função *Sigmoid*

Já na camada de saída, que geralmente é uma camada totalmente conectada, pode ser aplicada uma função *sigmoid* ou *softmax*. Quando a função *sigmoid* é aplicada, é produzido como saída um valor entre 0 e 1, muito útil para aplicações de classificação binária [8,9]. A Equação 2 mostra como ocorre o cálculo desta função:

$$f(u) = \frac{1}{1 + exp(-a\,u)} \tag{2}$$

Sendo a o valor que define a inclinação da Sigmoid

2.3.3 Função *Softmax*

A função *softmax,* quando aplicada, normaliza o vetor de entrada para valores no intervalo de 0 a 1, de forma que a soma dos valores de saída quando somados resultam em 1 e cada valor que sai da função pode ser interpretada como a probabilidade da entrada da rede ser classificada como cada uma das classes representadas pelos neurônios da camada de saída [8,9]. Veja a função *softmax* definida na Equação 3 a seguir:

$$P_j^i = exp(W_j^T x^i + a_j) \,/\, \sum_{n=1}^{C} exp(W_n^T x^i + a_n) \tag{3}$$

Onde P_j^i é a probabilidade do dado x^i pertencer à classe de imagens j, W_j e a_j são os valores calculados dos parâmetros a partir do dado j, e C é o número de classes de imagens.

A seguir, na Figura 3, é possível observar o gráfico demonstrando o comportamento das três funções de ativação citadas.

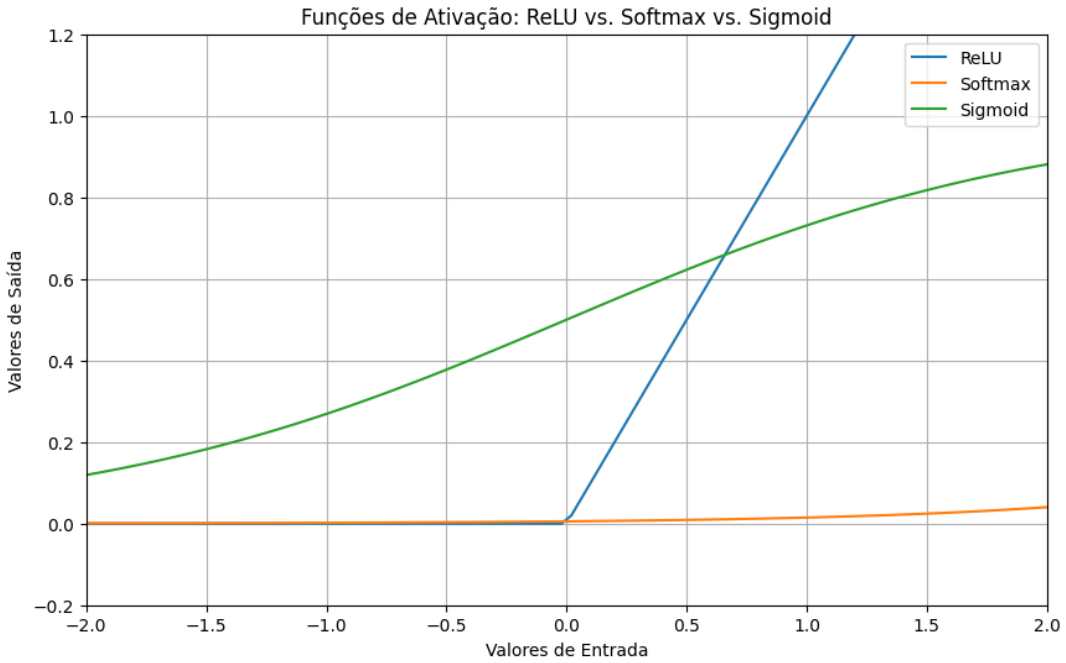

Figura 3. Gráficos das funções de ativação *ReLU, Sigmoid e Softmax*. (Autoria própria)

2.4 Otimizadores

Para melhorar o desempenho do modelo no processo de classificação, é necessário ajustar as variáveis em todas as camadas da rede. Para avaliar o quão bem o modelo está se saindo, é importante comparar a saída prevista do modelo com a saída desejada. Uma medida crucial nesse sentido é a entropia cruzada. A entropia cruzada é uma função contínua que sempre retorna valores positivos e se aproxima de zero quando a saída prevista coincide com a saída desejada. Portanto, o objetivo da otimização é minimizar a entropia cruzada, buscando valores próximos a zero [10].

2.4.1 Adagrad

No Adagrad, abreviação para *Adaptative Gradient*, ou gradiente adaptativo, a taxa de aprendizado é modificada individualmente para cada parâmetro em cada passo de tempo, com base nos gradientes históricos calculados para aquele parâmetro. Especificamente, parâmetros que têm gradientes mais frequentes e maiores têm suas taxas de aprendizado reduzidas, enquanto parâmetros com gradientes mais raros e menores têm suas taxas de aprendizado aumentadas [10].

2.4.2 Adam

No método Adam, são calculadas duas estimativas para cada parâmetro: o primeiro momento (média) e o segundo momento (variação não centralizada) dos gradientes. Essas estimativas são atualizadas a cada iteração do algoritmo de otimização [11]. O primeiro momento é uma média exponencialmente ponderada dos gradientes anteriores, enquanto o segundo momento é uma média exponencialmente ponderada das variações dos gradientes anteriores. Essa combinação de médias exponenciais decrescentes permite que o Adam ajuste as taxas de aprendizado de forma adaptativa, levando em consideração tanto a direção (primeiro momento) quanto a variabilidade (segundo momento) dos gradientes. Isso ajuda a otimizar o processo de treinamento do modelo, permitindo uma adaptação eficiente das taxas de aprendizado para cada parâmetro [10, 11].

2.5 Métricas de avaliação do modelo

Em um estudo de redes neurais de classificação, cada exemplo pode ser classificado como positivo ou negativo. No entanto, pode haver casos em que a classificação não corresponda ao rótulo real do exemplo. Para avaliar o desempenho do modelo, usamos as métricas TP - *True Positive* (verdadeiro positivo), TN - *True Negative* (verdadeiro negativo), FP - *False Positive* (falso positivo) e FN - *False Negative* (falso negativo). Para uma classificação multiclasse, a maior quantidade de classes do que a simples classificação binária permite que haja valores de falsos positivos e de falsos negativos maiores [10].

A acurácia mede a proporção de dados classificados corretamente com relação à quantidade total de classificações realizadas. Assim, indica a porcentagem de previsões corretas realizadas pelo modelo [12]. A

acurácia pode ser calculada pela Equação 4 abaixo:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

Outra métrica utilizada para avaliar modelos de classificação é o *F1-Score, que* calcula uma média harmônica entre duas métricas, a precisão e a sensibilidade, o que significa que dá mais peso à menor delas. O F1-score é usado para avaliar o desempenho do modelo de aprendizado de máquina em tarefas de classificação e fornece uma medida geral do equilíbrio entre a precisão e a sensibilidade do modelo. A precisão irá medir a razão entre TP e valores classificados como positivos (TP + FP), dando ênfase em avaliar se houve um valor alto de classificações de falsos positivos. Já a sensibilidade trata da porcentagem de TP com relação a todos os valores que foram previstos de forma correta (TP+FN) [12]. O cálculo do F1-Score pode ser visto na Equação 5:

$$F1 = 2 * (Precisão * Sensibilidade) / (Precisão + Sensibilidade) \quad (5)$$

3. METODOLOGIA

3.1 Ambiente Computacional

O treinamento do modelo de rede neural convolucional deste trabalho e a avaliação e testes de inferência feitos com ele foram realizados na plataforma de cloud computing do Google Colab©. O ambiente de execução foi criado utilizando os recursos gratuitos da plataforma e consistia em uma máquina com: GPU Tesla T4 com CUDA 12.0, possuindo 15.0 GB de memória RAM; 12.7 GB de RAM e 78.2 GB de disco na máquina.

O *framework* utilizado foi o *TensorFlow*, que é um *framework* de código aberto para computação numérica e aprendizado de máquina, na versão 2.12.0 e com a utilização também do *framework* Keras, que é uma biblioteca de *deep learning* conhecida por sua simplicidade. Algumas outras dependências são as bibliotecas pandas versão 1.5.3, sklearn versão 1.2.2, seaborn versão 0.12.2, numpy versão 1.22.4.

3.2 Banco de Imagem e *Pipeline* de Dados

O banco de imagens utilizado para a realização deste trabalho foi o *Grape Disease Dataset (Original)*, um banco de imagens públicas disponível na plataforma Kaggle que conta com 9.027 imagens, com 256x256 pixels, de folhas da planta da uva, separadas em 4 categorias: *Black Rot, ESCA, Leaf Blight* e *Healthy* [13].

Podridão-negra, também conhecida como *Black Rot*. Causada pelo fungo *Guignardia bidwellii*, essa doença provoca o apodrecimento das plantas, especialmente durante estações úmidas ou chuvosas. É possível identificar visualmente a presença da doença por meio de manchas negras nas folhas. *ESCA* é uma doença que resulta de um complexo de fungos de diferentes espécies. A planta afetada pelo *ESCA* apresenta manchas de coloração amarela nas folhas, no tronco e nas raízes. Esse problema pode reduzir o crescimento e a produção das uvas, além de causar a morte de tecidos da planta. A *Leaf Blight* é mais uma doença fúngica que ataca as plantas de uva. O fungo *Phomopsis viticola* infecta folhas, frutos e ramos, manifestando-se por meio de manchas circulares necróticas. Essas manchas possuem bordas escuras e centro claro. A categoria "*Healthy*" refere-se às folhas saudáveis da planta de uva, que não apresentam manchas ou sinais de podridão.

Há um total de 9.028 imagens divididas nestas categorias. Para preparar o *dataset* para ser utilizado na aplicação, foi feita a divisão das imagens em uma arquitetura de pastas, sendo uma pasta de teste e uma pasta de treinamento. Cada uma destas pastas possui 4 pastas em seu interior, uma para cada categoria. Na pasta de teste, tem-se 486 imagens de Black Rot, 503 imagens de *ESCA*, 421 imagens de *Leaf Blight* e 396 imagens de folhas saudáveis, totalizando 1806 imagens para teste, o equivalente a aproximadamente 20% do tamanho total do *dataset*. Para a pasta de treinos, a divisão de imagens ocorre nas seguintes quantidades: 1888 imagens de *Black Rot*, 1920 imagens de *ESCA*, 1722 imagens de *Leaf Blight* e 1692 imagens da categoria *Healthy* (folhas saudáveis), totalizando o restante das 7.222 imagens do *dataset*. Uma quantidade equilibrada destas imagens entre as categorias é de fundamental importância para que a rede não beneficie a inferência de uma categoria sobre outra.

Esta forma de encadeamento de páginas irá auxiliar no carregamento dos dados para o algoritmo, pois o *TensorFlow* apresenta uma função que é capaz de reconhecer esta divisão e de selecionar as categorias automaticamente, se orientando pelo nome das pastas. Veja abaixo uma amostra de imagens do banco de dados utilizado:

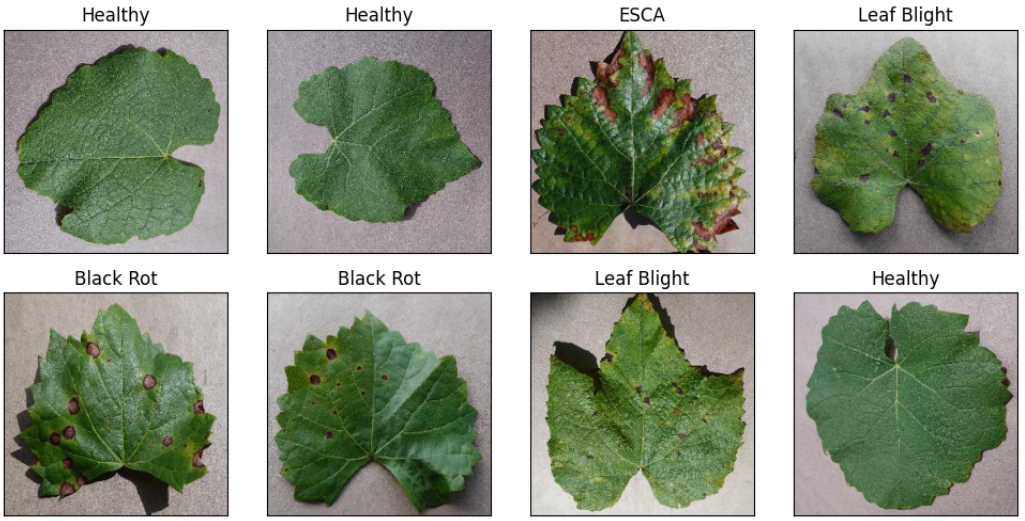

Figura 4. Amostra de imagens do banco de imagens utilizado. (Autoria própria)

3.3 Arquitetura da Rede

A rede neural criada consiste em duas etapas de pré-processamento das imagens para reduzir as dimensões das imagens para 128x128 pixels e realizar a normalização dos valores. A normalização consiste em padronizar os valores de entrada da rede para que seja possível aprender de forma mais eficiente. No caso desta rede convolucional, foi realizada a divisão do valor de cada pixel por 255, que é o valor máximo que um pixel pode ter em uma imagem de 8 bits, isto resulta entre valores normalizados entre 0 e 1 que estão em uma escala mais fácil de ser trabalhada. Estas duas etapas fazem o processamento da imagem assim que ela for passar na rede, conhecido como "pré-processamento *on the fly*", apesar de trazer vantagens como a economia de espaço de memória, já que não é preciso armazenar todos os dados processados, eles são processados já na hora da utilização, há também desvantagens neste método, como o custo computacional maior, já que todas as imagens serão processadas ao entrar na rede, diferente da opção de utilizar imagens já pré-processadas.

Seguido desta etapa de pré-processamento, há a primeira camada convolucional da rede composta por 16 *kernels* de tamanho 5x5 e utilizando a função de ativação *ReLU*. Logo após há uma camada de *MaxPooling* utilizando uma janela 2x2, reduzindo cada sub-região dos mapas de características gerados pela etapa convolucional em um único pixel de valor igual ao maior valor desta sub-região. Assim, a característica mais forte da sub-região é preservada e as dimensões dos mapas de características são reduzidos pela metade para que possam ser utilizados pelas camadas seguintes. As próximas camadas são camadas de convolução e de *pooling* intercaladas, uma camada de convolução com 32 *kernels* de tamanho 3x3 e ativação *ReLU* seguido de uma camada de *pooling* 2x2, após isso mais uma cada de convolução com 64 *kernels* de tamanho 3x3 e também com ativação *ReLU*. E então a última camada de *pooling* com janela 2x2.

Todos os valores dos mapas de características de saída dessa sequência são vetorizados pela camada de achatamento (*Flatten*), que deixa seus valores em um formato ideal para passarem por uma rede totalmente conectada. A etapa totalmente conectada é formada por 3 camadas densas, a primeira com 100 neurônios e ativação *ReLU*, a segunda com 20 neurônios e com a mesma função de ativação e a última camada, a camada de saída, sendo uma camada densa com 4 neurônios e ativação *Softmax*. É importante ressaltar que o número de neurônios na camada de saída precisa ser equivalente ao número de classes que se pretende classificar. Esta rede resultou em um total de 1.280.956 parâmetros treináveis. Um resumo deste modelo e das saídas de cada camada podem ser vistas na imagem do sumário criado por uma função do *TensorFlow* a seguir:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resizing (Resizing)         (32, 128, 128, 3)         0

 rescaling (Rescaling)       (32, 128, 128, 3)         0

 conv2d (Conv2D)             (32, 124, 124, 16)        1216

 max_pooling2d (MaxPooling2D  (32, 62, 62, 16)         0
 )

 conv2d_1 (Conv2D)           (32, 60, 60, 32)          4640

 max_pooling2d_1 (MaxPooling  (32, 30, 30, 32)         0
 2D)

 conv2d_2 (Conv2D)           (32, 28, 28, 64)          18496

 max_pooling2d_2 (MaxPooling  (32, 14, 14, 64)         0
 2D)

 flatten (Flatten)           (32, 12544)               0

 dense (Dense)               (32, 100)                 1254500

 dense_1 (Dense)             (32, 20)                  2020

 dense_2 (Dense)             (32, 4)                   84

=================================================================
Total params: 1,280,956
Trainable params: 1,280,956
Non-trainable params: 0
_________________________________________________________________
```

Figura 4. Sumário das camadas e parâmetros do modelo. (Autoria própria)

4. RESULTADOS E DISCUSSÕES

Nesta seção, apresentamos os resultados e a discussão da precisão do modelo proposto para a identificação de doenças em plantas utilizando redes convolucionais. O objetivo principal deste estudo foi investigar a eficácia do modelo na detecção e classificação de diferentes doenças foliares na planta da uva, classificando entre as classes de plantas saudáveis, *black rot, ESCA e leaf blight*. A seguir, são apresentados os resultados obtidos a partir da avaliação do modelo utilizando o conjunto de dados de teste, contendo imagens de plantas com diferentes doenças e plantas saudáveis.

A precisão do modelo foi avaliada utilizando a métrica de acurácia, que representa a proporção de classificações corretas em relação ao total de amostras avaliadas, nessa métrica, o modelo apresentou uma avaliação de 0,9795.

Durante a revisão de estudos relacionados à classificação de doenças em plantas de uva, encontramos um trabalho intitulado "*Identification of Grape Leaf Diseases Using Convolutional Neural Network*", no qual os autores Hasan et al. investigaram uma amostra de 1000 imagens do mesmo conjunto de dados do Kaggle utilizado neste estudo. Em sua pesquisa, eles aplicaram pré-processamento utilizando normalização e conversão para escala de cinza, exploraram diferentes combinações de hiperparâmetros e analisaram a influência desses parâmetros na precisão do modelo. Após o treinamento com 100 épocas e uma taxa de aprendizagem de 0,001, eles alcançaram uma acurácia de 91,37%. Embora não tenhamos informações específicas sobre a arquitetura do modelo utilizado por Hasan et al., é evidente que existe uma diferença na avaliação em relação ao modelo desenvolvido neste estudo. Isso pode ser atribuído ao fato de que nosso trabalho considerou os três canais de cores em vez de converter os dados para escala de cinza. Além disso, é importante ressaltar que nosso modelo foi treinado por apenas 15 épocas, um número menor em comparação às 100 épocas do estudo de Hasan. No entanto, é válido destacar que treinamos nosso modelo com um conjunto de imagens significativamente maior, o que pode ter contribuído para um aprendizado mais eficiente e rápido.

Essa taxa de acerto indica que o modelo aprendeu de forma eficiente as características distintivas associadas a cada classe durante o treinamento, permitindo uma classificação precisa no conjunto de dados de teste. A acurácia é uma métrica fundamental na avaliação de modelos de classificação, pois reflete a capacidade do modelo em fazer previsões corretas em relação ao número total de amostras avaliadas.

A Figura 5 ilustra a matriz de confusão, que mostra a distribuição das classificações corretas e incorretas para cada classe.

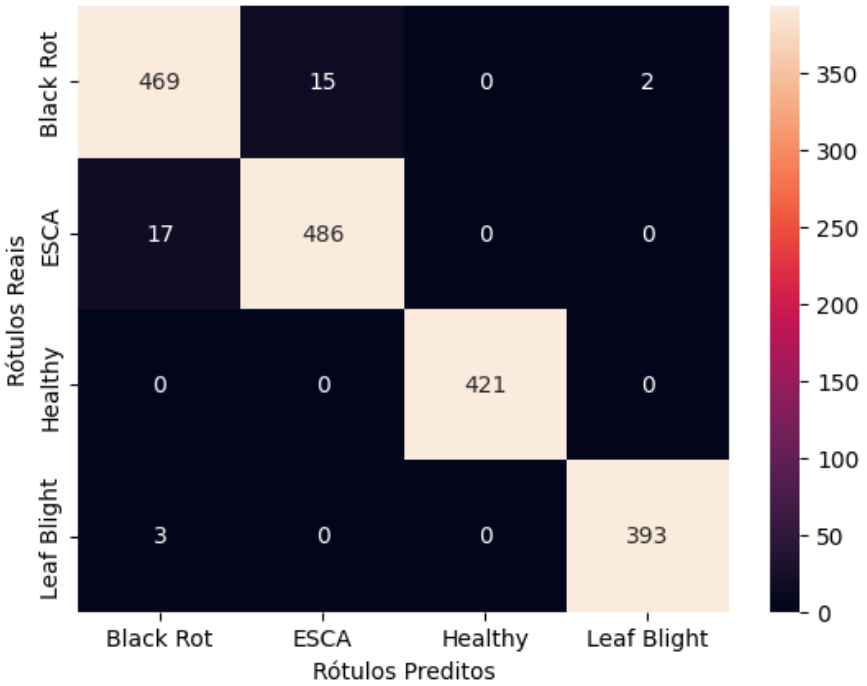

Figura 5. Mapa de calor da matriz de confusão dos resultados do modelo aplicado no dataset de teste. (Autoria própria)

Observa-se que houve um total de 37 predições incorretas, sendo que 32 delas ocorreram entre as classes *Black Rot* e *ESCA*. Essa confusão entre as classes pode ser atribuída às características semelhantes das doenças, que apresentam manchas negras nas plantas. Embora essa ocorrência represente mais de 86% dos erros, ao analisar a quantidade de acertos, fica evidente que o modelo é capaz de distinguir entre essas duas classes com um alto grau de confiabilidade. Abaixo, na Figura 6, é possível visualizar 18 imagens do banco de dados de teste, com os respectivos rótulos verdadeiros e a predição realizada pelo modelo para aquela imagem, para facilitar a visualização, predições corretas foram exibidas na cor verde e predições incorretas na cor vermelha.

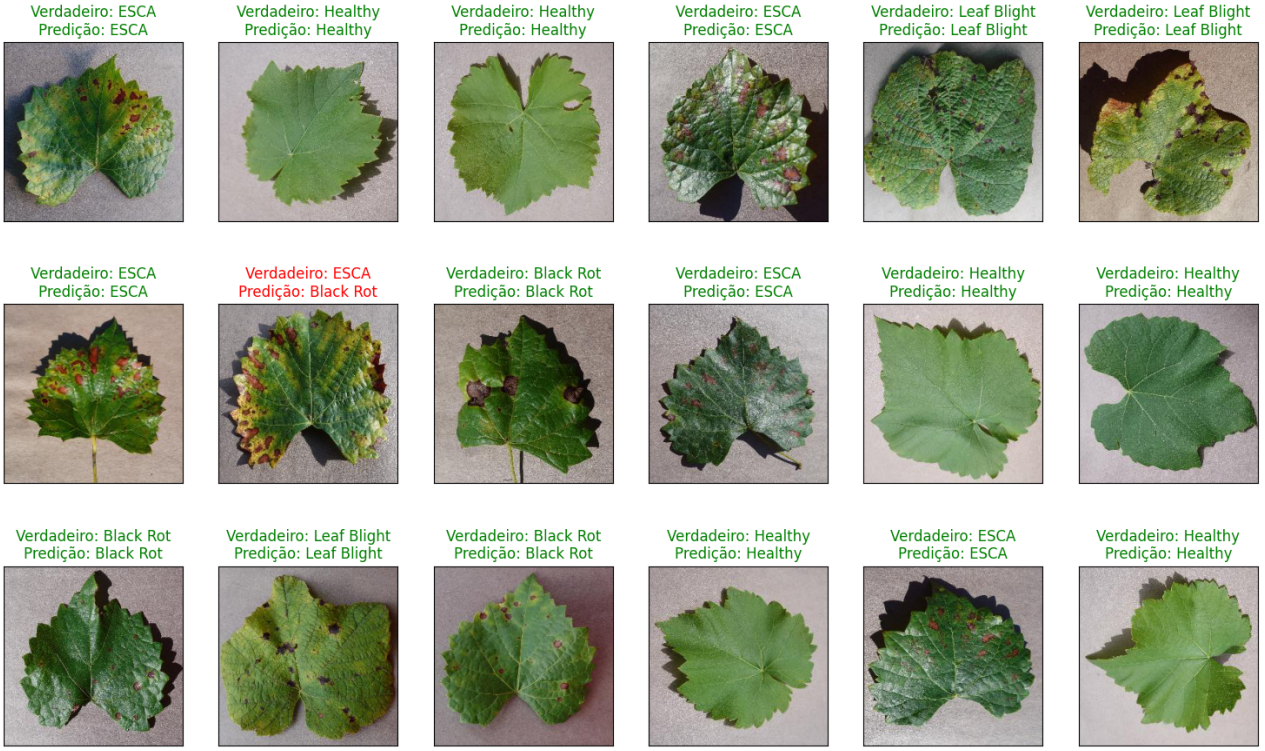

Figura 6. Comparativo entre classes verdadeiras e classificações realizadas pelo modelo. (Autoria própria)

Utilizando a matriz de confusão, temos a oportunidade de calcular diversas métricas adicionais, permitindo uma compreensão mais aprofundada do desempenho do modelo ao classificar cada classe. Essa análise é crucial para determinar se o modelo aprendeu de maneira adequada e equilibrada a realizar a classificação em todas as classes, ou se há algum desequilíbrio que precisa ser corrigido. Isso pode envolver a coleta de um maior número de imagens representativas dessa classe durante o treinamento ou a realização de alterações na arquitetura do modelo para melhor acomodar as características específicas dessa classe. Além disso, a análise das métricas individuais também pode fornecer *insights* sobre o desempenho relativo entre as classes. Por exemplo, se observarmos que uma classe específica apresenta uma acurácia significativamente menor do que as outras, isso pode indicar a necessidade de investigar e entender melhor os desafios associados à classificação desta classe em particular.

Tabela 1. Acurácia e *F1-Score* do modelo por classe. (Autoria própria)

Classes	Nº Predições corretas	Nº de predições total	Acurácia por classe	*F1 - Score*
Black Rot	469	486	0,96502	0,96205
ESCA	486	503	0,96620	0,96812
Healthy	421	421	1,00000	1,00000
Leaf Blight	393	396	0,99242	0,99367

Ao se considerar o valor do *F1-Score*, que é uma medida que combina precisão e sensibilidade, também podemos avaliar a qualidade do modelo. Nesse caso, observamos um *F1-Score* de 0,96205, 0,96812, 1,00000 e 0,99367 para cada classe, respectivamente. Esses valores elevados indicam um desempenho consistente e confiável do modelo na classificação de cada classe.

Esses resultados evidenciam o potencial do modelo proposto para a identificação precisa das diferentes doenças em plantas da uva, contribuindo para o avanço da área de diagnóstico de doenças foliares. No entanto, é importante ressaltar que existem desafios e limitações a serem abordados para melhorar ainda mais a precisão do modelo. Discutiremos esses aspectos e possíveis direções futuras na próxima seção.

5. CONSIDERAÇÕES FINAIS

Este estudo investigou a eficácia de um modelo baseado em redes convolucionais para identificação de doenças em plantas, com foco na classificação de diferentes doenças foliares na planta da uva. Os resultados obtidos demonstraram que o modelo foi capaz de aprender de forma eficiente as características distintivas associadas a cada classe durante o treinamento.

Além disso, a alta acurácia também indica a eficácia das redes convolucionais como uma abordagem promissora para a identificação de doenças em plantas. Esses resultados podem ter implicações significativas no campo da agricultura e manejo de cultivos, oferecendo uma ferramenta eficiente e automatizada para detectar precocemente doenças nas plantas. Isso permitiria uma resposta rápida e precisa, possibilitando a implementação de medidas de controle adequadas e reduzindo potenciais perdas de colheita.

No entanto, é importante reconhecer que, mesmo com alta acurácia e *F1-Score*, o modelo pode apresentar limitações em cenários específicos. Todo o *dataset* de treino e imagens utilizados vieram de um banco de imagens público do *kaggle*. Embora a quantidade e variedade de folhas de cada classe seja suficiente para atingir boas métricas com o modelo criado, para um caso prático, se faz necessário a criação de um banco de imagens com características mais próximas às do campo de sua aplicação. O cenário analisado contava com um caso onde todas as imagens se encontravam categorizadas e em um ambiente de iluminação controlada, além de contar com imagens com apenas uma classe por imagem e com fundo neutro à cor da folha. O que pode variar de acordo com onde o modelo seria aplicado.

No contexto do diagnóstico de doenças em plantas, os resultados deste estudo estabelecem uma base sólida para pesquisas futuras e o desenvolvimento de sistemas mais avançados de detecção e classificação. Essas descobertas destacam o potencial das redes convolucionais como uma ferramenta eficaz para auxiliar agricultores e profissionais do setor agrícola no monitoramento e controle de doenças em plantas, contribuindo para a melhoria sustentável da produção agrícola.

Futuros trabalhos poderiam dar continuidade utilizando um banco de imagens com características do ponto de implementação do modelo ou desenvolver um banco de imagens com tais características, explicitando que resultados se deseja alcançar com tal banco de imagens, “identificar a doença no meio ao cultivo” ou outro recorte semelhante.

6. REFERÊNCIAS BIBLIOGRÁFICAS

[1] KAMILARIS, ANDREAS; PRENAFETA-BOLDÚ, FRANCESC X. Deep learning in agriculture: A survey. Computers and Electronics in Agriculture, v. 147, p. 70-90, 2018. ISSN 0168-1699. Disponível em: <https://doi.org/10.1016/j.compag.2018.02.016>. Acesso em: 12 abr. 2023.

[2] SZABÓ, M. et al, Black Rot of Grapes (Guignardia bidwellii)—A Comprehensive Overview. Horticulturae, 9, 130, 2023. Disponível em: <https://doi.org/10.3390/horticulturae9020130>. Acesso em: 14 Abr. 2023.

[3] RUSSELL, S. J., & NORVIG, P. (2010). Artificial Intelligence: A Modern Approach. Pearson Education.

[4] ALOYSIUS, NEENA; GEETHA, M. A review on deep convolutional neural networks. In: Communication and Signal Processing (ICCSP), 2017 International Conference on. IEEE, 2017. p. 0588-0592.

[5] ZHANG, GUOQIANG PETER. Neural networks for classification: a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), v. 30, n. 4, p. 451-462, 2000.

[6] PENG WU AND ZHOU QIAN 2021 J. Phys.: Conf. Ser. 1820 012161.

[7] HASAN, M. A., RIANA, D., SWASONO, S., PRIYATNA, A., PUDJIARTI, E., & PRAHARTIWI, L. I. (2020). Identification of Grape Leaf Diseases Using Convolutional Neural Network. Journal of Physics: Conference Series, 1641(1), 012007. doi:10.1088/1742-6596/1641/1/012007

[8] HU, JING; CHEN, ZHIBO; YANG, MENG; ZHANG, RONGGUO; CUI, YAJI. A Multiscale Fusion Convolutional Neural Network for Plant Leaf Recognition. IEEE SIGNAL PROCESSING LETTERS, v. 25, n. 6, p. 782-786, Jun. 2018.

[9] ZHANG, S., HUANG, W., & ZHANG, C. (2018). Three-channel convolutional neural networks for vegetable leaf disease recognition. College of Information Engineering, Xijing University, Xi'an 710123, China & Tianjin University of Science and Technology, Tianjin 300222, China: Elsevier.

[10] TAQI, A. M.; AWAD, A.; AL AZZO, F. et al. The Impact of Multi-Optimizers and Data Augmentation on TensorFlow Convolutional Neural Network Performance. In: 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR). April 2018. DOI: 10.1109/MIPR.2018.00032.

[11] RIBEIRO, A. M.; ARAUJO JUNIOR, F. P. S. Um Estudo Comparativo Entre Cinco Métodos de Otimização Aplicados Em Uma RNC Voltada ao Diagnóstico do Glaucoma. Universidade Estadual do Piauí (UESPI).

[12] VILELA JUNIOR, G. B. et al. Métricas Utilizadas para Avaliar a Eficiência de Classificadores em Algoritmos Inteligentes. ISSN: 2178-7514. Vol. 14, Nº. 2, Ano 2022.

[13] RM1000. *Grape Disease Dataset (Original)* Disponível em: <https://www.kaggle.com/datasets/rm1000/grape-disease-dataset-original>