

CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA “PAULA SOUZA”

FACULDADE DE TECNOLOGIA DE BEBEDOURO

TECNOLOGIA EM BIG DATA NO AGRONEGÓCIO

**IDENTIFICAÇÃO DE DOENÇAS NA CULTURA DA
SOJA UTILIZANDO APRENDIZADO DE MÁQUINA**

AUTOR: RICARDO FARIA FIRMINO

ORIENTADOR: prof. Me. RENAN GUILERME NESPOLO

BEBEDOURO

2024

RICARDO FARIA FIRMINO

IDENTIFICAÇÃO DE DOENÇAS NA CULTURA DA SOJA UTILIZANDO APRENDIZADO DE MÁQUINA

Monografia apresentada à Faculdade de Tecnologia
de Bebedouro, como parte dos requisitos para a
obtenção do título de Tecnólogo em Big Data no
Agronegócio

Orientador: **prof. Me. Renan Guilherme Nespolo**

BEBEDOURO

2024

Agradecimentos

Agradeço primeiramente a Deus por conceder as condições necessárias para que eu pudesse me dedicar aos estudos no decorrer desses 3 anos de faculdade.

Agradeço a minha família, nas pessoas de minha esposa Tatiane, meu filho Caio e minha filha Rafaela pela paciência e compreensão durante os momentos de minha ausência neste período de estudos.

E por último, mas não menos importante, agradeço ao meu orientador Renan Guilherme Nespolo por todo apoio, ensino e incentivo na construção deste trabalho.

FIRMINO, R. F. **Identificação de doenças na cultura da soja utilizando aprendizado de máquina**. Trabalho de Graduação (Monografia). Centro Estadual de Educação Tecnológica “Paula Souza”. Faculdade de Tecnologia de Bebedouro. 50 p. 2024.

RESUMO

As doenças são um dos principais fatores de impacto na produtividade agrícola, afetando diretamente seu rendimento e qualidade.

Este estudo apresenta uma proposta para identificação de doenças na cultura da soja por meio do Aprendizado de Máquina e técnicas de classificação de imagens, baseado em um modelo de Redes Neurais Convolucionais (CNNs) que é capaz de classificar imagens de plantas de soja de acordo com características e padrões que determinem a que classe ou categoria ela se enquadre, comparando dois frameworks (TensorFlow e Pytorch) de implementação de CNN diferentes a fim de validar qual a melhor abordagem para este tipo de problema. Utilizando um conjunto de dados de aproximadamente 7 mil imagens de folhas de soja, previamente rotuladas de acordo com seu estado (saudável ou doente), o presente estudo busca implementar e treinar um modelo de CNN que possa classificar tais imagens com eficiência e agilidade, provendo uma forma automatizada de identificação de doenças nesta cultura. Dois frameworks amplamente utilizados pela comunidade científica para a implantação do modelo de CNN foram considerados e comparados o desempenho dado em acurácia apresentada e tempo de execução de inferência. Como inferência foi adotado o método *k-fold Cross-Validation* com 20% do *Dataset* nos *folds* de validação. Os resultados apresentados se mostram promissores, sobressaindo um dos frameworks nas duas métricas avaliadas.

Palavras-chave: Soja, Redes Neurais Convolucionais, Classificação, Identificação Doenças.

FIRMINO, R. F. **Identificação de doenças na cultura da soja utilizando aprendizado de máquina**. Trabalho de Graduação (Monografia). Centro Estadual de Educação Tecnológica “Paula Souza”. Faculdade de Tecnologia de Bebedouro. 50 p. 2024.

ABSTRACT

Diseases are one of the main factors that impact agriculture productivity, which directly affects yield and quality.

This study presents a solution to identify soybean diseases using Machine Learning and image classification techniques, based on a Convolutional Neural Network (CNN) model capable of classifying soybean plant according to the features and patterns that determine which class its belong to, by comparing two different frameworks (TensorFlow and Pytorch) with the proposal of evaluate which one is the best approach for this kind of problem. Using a dataset with about 7.000 soybean leaf images, previously labeled according to their condition (healthy or sick), this study aims to implement and train a Convolutional Neural Network that can classify these imagens efficiently and quickly, providing an automated method for disease identification in this agricultural culture. Two widely used frameworks by the scientific community for CNN implementations were evaluated and compared based on their performance in terms of accuracy and inference execution time. The k-fold Cross-Validation was adopted, with 20% of the dataset to each fold for validation. The presented results are promisors, with one of the frameworks achieving better results in the two evaluate metrics.

Keywords: Soybean, Convolutional Neural Network, Image classification, Detection, .

SUMÁRIO

1	INTRODUÇÃO	13
2	FUNDAMENTAÇÃO TEÓRICA.....	15
2.1	AGRICULTURA.....	15
2.2	CULTURA DA SOJA	16
2.3	APRENDIZADO DE MÁQUINA.....	20
2.4	REDES NEURAIS ARTIFICIAIS.....	24
2.5	REDES NEURAIS CONVOLUCIONAIS PARA CLASSIFICAÇÃO DE IMAGENS ...	28
2.6	INFERÊNCIA E AVALIAÇÃO.....	32
3	MATERIAIS E MÉTODOS.....	34
3.1	CONJUNTO DE DADOS	34
3.2	MODELO PROPOSTO.....	38
3.3	PROPOSTA DE EXPERIMENTO.....	41
3.4	INFERÊNCIA E AVALIAÇÃO.....	41
3.5	MATERIAIS UTILIZADOS	42
4	RESULTADOS E DISCUSSÕES.....	44
5	CONCLUSÃO	48
	REFERÊNCIAS.....	49

LISTA DE FIGURAS

Figura 1:	Folha de soja contaminada com Ferrugem Asiática	18
Figura 2:	Folha de soja contaminada com mancha alvo.	18
Figura 3:	Folha da soja contaminada com mancha parda.	19
Figura 4:	Folha da soja infectada com a doença Míldio	19
Figura 5:	Folhas da soja contaminadas com crestamento bacteriano	20
Figura 6:	Ilustração de um processo de treinamento de um sistema <i>Anti-Spam</i>	21
Figura 7:	Processo de agrupamento (clusterização) não-supervisionado.....	22
		22
Figura 8:	Ilustração aprendizado de máquina Semissupervisionado	23
Figura 9:	Modelo geral de aprendizado de máquina supervisionado e técnicas de classificação	24
Figura 10:	Estrutura de um neurônio biológico	25
Figura 11:	Ilustração da lógica por trás de um neurônio artificial	26
Figura 12:	Diagrama de uma rede neural artificial não profunda	27
Figura 13:	Ilustração do mapeamento de características da CNN	28
Figura 14:	Ilustração completa das múltiplas camadas	29
Figura 15:	Ilustração da visão humana	30
Figura 16:	Ilustração das camadas de convolução.....	30
		30
Figura 17:	Diagrama do processo de treinamento do modelo CNN	31
Figura 18:	Ilustração funcionamento do <i>K-fold cross-validation</i>	32
Figura 19:	Exemplar de imagem que compõe o <i>Dataset</i> , contendo uma planta saudável.....	35
Figura 20:	Exemplar de imagem que compõe o <i>Dataset</i> , contendo uma planta doente com crestamento bacteriano	35

Figura 21:	Exemplar de imagem que compõe o <i>Dataset</i>, contendo uma planta doente com ferrugem asiática.....	36
Figura 22:	Exemplar de imagem que compõe o <i>Dataset</i>, contendo uma planta doente com míldio	36
Figura 23:	Exemplar de imagem que compõe o <i>Dataset</i>, contendo uma planta doente com mancha alvo	37
		37
Figura 24:	Exemplar de imagem que compõe o <i>Dataset</i>, contendo uma planta doente com mancha parda	38
		38
Figura 25:	Arquitetura CNN.....	40
Figura 26:	Exemplar de imagens após etapa de pré-processamento e segmentação.....	43
		43

Lista de Tabelas

Tabela 1:	Resultados de classificação possíveis	33
Tabela 2:	Arquitetura da CNN proposta no presente estudo.....	39
Tabela 3:	Comparativo acurácia média e tempo de treinamento entre os frameworks.....	44
Tabela 4:	Comparativo tempo de processamento e treinamento	45
Tabela 5:	Descritivo <i>Tensorflow</i> com 400 épocas e 5 <i>folds</i>	46

Lista de Gráficos

Gráfico 1:	Comparativo acurácia média entre <i>frameworks</i>	45
Gráfico 2:	Comparativo tempo de processamento e treinamento	46
Gráfico 3:	Descritivo Tensorflow com 400 épocas e 5 <i>folds</i>	47

LISTA DE EQUAÇÕES

Acurácia (1).....	33
Eval_{acc} (2).....	42

Lista de Siglas

CNN – *Convolutional Neural Network*

RNA – Rede Neural Artificial

ANN – *Artificial Neural Network*

DNN – *Deep Neural Network*

MLP – *Multi Layer Perceptron*

1 INTRODUÇÃO

Em um mundo em que se busca altos níveis de produção para atender à crescente demanda por alimentos, é de extrema importância o desenvolvimento de soluções que minimizem os impactos das doenças nas plantações e maximizem a produção agrícola. De acordo com a FAO - *Food and Agriculture Organization* (2017), agência especializada das Nações Unidas que lidera os esforços para erradicar a fome no planeta, será necessário produzir quase 50% mais alimentos até 2050 em comparação a produção de 2012 a fim de atender a demanda global. A estimativa é de que a população mundial atinja 9,73 bilhões de pessoas, o que aumentará significativamente a necessidade por alimentos, consequentemente demandando maior necessidade de produção. Considerando que as doenças são um dos principais desafios que afetam a qualidade e produtividade de qualquer cultivo, este trabalho busca desenvolver soluções tecnológicas para melhorar a detecção de doenças na cultura da soja, para que as ações de combate a essas doenças possam ser aplicadas o mais precoce possível, facilitando assim a rápida recuperação das plantas infectadas e impedindo a disseminação da doença na lavoura, resultando em menores perdas de produção. Qualquer anomalia que comprometa o crescimento de uma planta ou cultura é considerada uma doença. Essas doenças são responsáveis por grandes perdas na produção agrícola (Henning, 2022). A ferrugem asiática, uma das doenças que buscamos identificar neste trabalho, pode causar perdas de até 90% da produção se não for controlada de maneira adequada e precoce (Conab, 2024). O manual de identificação de doenças da soja da Embrapa (2014) explica que a ferrugem asiática provoca a desfolha das plantas, afetando o desenvolvimento dos grãos e, consequentemente, a produção.

O uso de técnicas de Aprendizado de Máquina tem se intensificado a cada dia em diversas áreas, como por exemplo: Financeira, usada a fim de contribuir na detecção de fraudes, ou então na predição de valores de ativos; Internet: utilizada em sistemas de recomendação de conteúdo, seja em redes sociais ou streaming; Segurança: muito utilizada em sistemas de vigilância para reconhecimento facial, identificação de atitudes suspeitas entre outros; Segurança da Informação: muito aplicada em sistemas de AntiSpam, Antivírus, detecção de intrusão; Esses são apenas alguns exemplos da aplicação de técnicas de Aprendizado de Máquina na realidade atual, e assim como esses exemplos, essas técnicas são muito bem-vindas na agricultura, pois podem contribuir em diversos aspectos dentro de um fluxo de produção agrícola.

Por meio dessas técnicas, buscamos aplicar mais especificamente a técnica de Aprendizado de Máquina voltada para a classificação de imagens, e com isso, desenvolver uma solução capaz de identificar diversos tipos de doenças na cultura da soja.

Deste modo, os objetivos deste trabalho são:

- Propor uma solução capaz de identificar se uma planta de soja está doente ou saudável.
- No caso de uma planta doente, desenvolver uma solução para identificar qual doença está infectando a planta.
- Aplicar técnicas de Aprendizado de Máquina para processamento e análise de imagens de plantas de soja, classificando-as de acordo com a doença identificada.
- Comparar dois frameworks de implementação de Redes Neurais Convolucionais profundas.

O presente trabalho está segmentado em: 1 Introdução; 2 Fundamentação Teórica; 3 Materiais e Métodos; 4 Resultado e Discussões; 5 Conclusão; ao final as Referências.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção são apresentados todos os fundamentos do presente estudo. A seção está separada nos seguintes tópicos: 2.1 Agricultura; 2.2 Cultura da Soja; 2.3 Aprendizado de Máquinas; 2.4 Redes Neurais Artificiais; 2.5 Redes Neurais Convolucionais para a classificação de imagens.

2.1 AGRICULTURA

A agricultura, em sua essência, é a arte de cultivar o solo com o objetivo de produzir uma ampla variedade de produtos agrícolas, que incluem alimentos destinados ao consumo humano e animal, como também matérias-primas para a indústria. (Castanho *et al*, 2017).

O surgimento dessas técnicas e práticas de cultivo, que são definidas como agricultura, remontam os anos 10.000 antes da nossa era, foi no período conhecido como Neolítico que se tem registros do início dessas práticas de semeadura e cultivo de plantas assim como o trato de animais em cativeiro, transformando de forma paulatinamente essa sociedade em cultivadores. Até então, os seres humanos eram considerados nômades caçadores, que se moviam entre as regiões em busca de alimentos. (Mazoyer *et al*, 2010).

Ao longo do tempo, a agricultura evoluiu junto com o desenvolvimento humano. À medida que as pessoas começaram a se estabelecer em determinadas regiões, deixaram de depender exclusivamente dos alimentos encontrados na natureza e passaram a cultivar e produzir seus próprios alimentos.

A agricultura primitiva, que visava exclusivamente a produção de meios de subsistência e que dependia basicamente da força humana, ficou conhecida como Agricultura 1.0. Esse modelo predominou desde o surgimento da agricultura até meados da década de 1920. Com o início da industrialização, surgiu a era da Agricultura 2.0, que incorporou técnicas voltadas para o aumento da produtividade por meio do uso de máquinas, fertilizantes e variedades melhoradas de plantas. O foco desse novo modelo era aumentar a produção por área de plantio. Para efeito de comparação, na Agricultura 1.0, no início do século passado, um agricultor produzia alimentos suficientes para apenas 4 pessoas. Já no modelo subsequente, isto é, Agricultura 2.0, esse número subiu para 26 pessoas.

Com a chegada da Agricultura 3.0, que se caracteriza no início dos anos 1990, um agricultor passou a ser capaz de produzir para mais de 150 pessoas. Este salto foi impulsionado pelo uso de tecnologias de automação no campo, práticas de sustentabilidade e coleta de dados de produção, que facilitaram a tomada de decisões e aumentam os índices de produtividade. Foi nessa fase que a chamada agricultura de precisão começou a ganhar destaque, utilizando sensores e automação para alcançar maior eficiência e economia no uso de fertilizantes e controle de pragas. A partir de 2010, temos o início da fase da Agricultura 4.0, caracterizada por avanços significativos na genética de sementes, que agora produzem mais em menor tempo e são resistentes a diversas pragas. Além disso, essa fase integra soluções tecnológicas como monitoramento avançado, ferramentas de automação, conectividade, big data e processamento de dados, aproveitando de processos digitais que trazem resultados significantes na produção agrícola. (QUEIROZ et al, 2022).

Atualmente vivemos o início da era da chamada Agricultura 5.0, a qual tem como base a inteligência artificial, aprendizado de máquinas, computação em nuvem, sensoriamento remoto entre outras tecnologias. A figura a seguir ilustra um pouco dos eventos e tecnologias de cada era dessa jornada revolucionária. (Martos, 2021).

2.2 CULTURA DA SOJA

Dentre as inúmeras culturas cultivadas no Brasil e mundo a fora, temos a cultura da soja, a qual está entre as atividades econômicas que apresentaram crescimentos mais expressivos nas últimas décadas. (Hirakuri *et al*, 2014).

De acordo com o Departamento de Agricultura dos Estados Unidos (USDA), o Brasil produziu cerca de 169 milhões de toneladas de soja na safra de 2024 ainda em andamento, liderando o ranking mundial de produção de soja com 39% da produção, seguido por Estados Unidos 29% com 124 milhões de toneladas e Argentina com 12% e 51 milhões de toneladas.

Segundo a Frente Parlamentar da Agropecuária (FPA), desde a década de 1970, a soja se consolidou como a principal cultura agrícola do Brasil. Inicialmente cultivada no Sul do país devido às condições climáticas, seu cultivo foi impulsionado pela indústria de óleo aliada ao desenvolvimento de novas tecnologias, que permitiram a adaptação da planta a outras regiões. Com o surgimento de cultivares resistentes e adaptadas, a soja passou a ser cultivada em

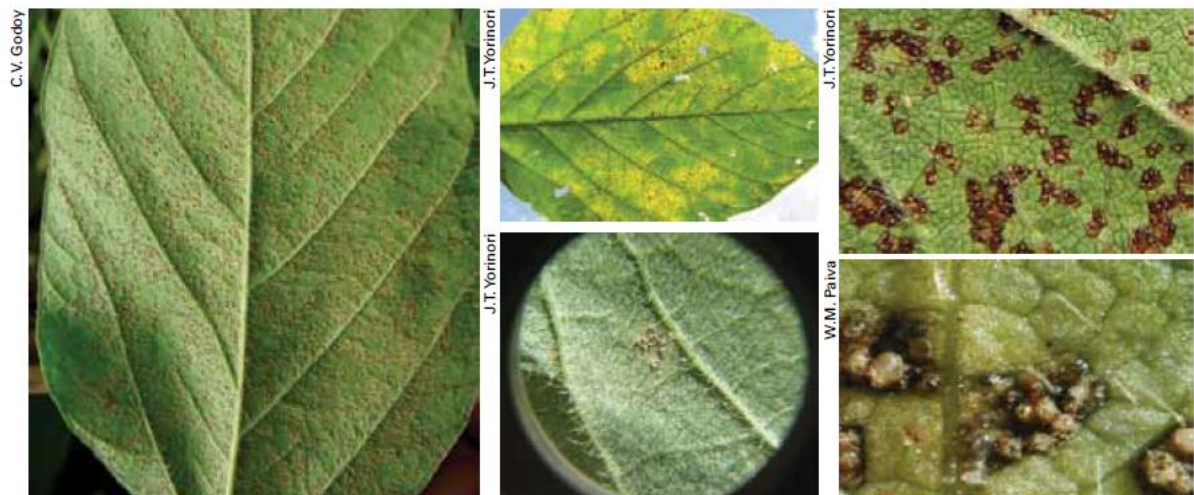
diversas áreas do território nacional, o que resultou em um aumento significativo da área plantada e da produção. Em apenas 40 anos, a produção saltou de 26 milhões para mais de 160 milhões de toneladas. Atualmente, o cultivo de soja movimenta cerca de 100 bilhões de dólares por ano, tornando-se uma cultura de extrema importância para a economia brasileira e mundial.

Quando falamos em capacidade de produção, devemos destacar que existem alguns fatores que podem impactar a produtividade da soja, entre eles estão: questões climáticas, nutrição do solo, pragas e doenças. As doenças têm papel de destaque devido ao seu potencial de perdas de produção quando não manejadas adequadamente. Este fator requer atenção especial dos produtores para que possam ser identificadas e rapidamente tratadas a fim de minimizar as perdas causadas por essas doenças. (Castro-Júnior, 2020).

Neste trabalho iremos abordar algumas das principais doenças que afetam a produtividade da soja. Dentre as principais doenças que incidem nesta cultura, podemos destacar a ferrugem asiática (*Phakopsora pachyrhizi*) que quando não identificada e remediada precocemente pode causar perdas de até 90% da produção, mancha alvo (*Corynespora cassiicola*), mancha parda (*Septoria glycines*), míldio (*Peronospora manshurica*) e Crestamento bacteriano (*Pseudomonas savastanoi*), além é claro, das plantas saudáveis que serão da mesma forma identificadas pelo algoritmo proposto no trabalho. (Castro-Júnior, 2020).

De acordo com o Manual de Identificação de doenças de Soja da Embrapa, os sintomas da ferrugem asiática são identificados de forma visual, que se inicia com minúsculos pontos mais escuros no tecido da folha, de colocação esverdeada a cinza-esverdeada, que se espalham rapidamente. (Henning, 2014). Na figura 1 observa-se uma folha de soja contaminada com ferrugem asiática.

Figura 1: Folha de soja contaminada com Ferrugem Asiática



Fonte: Henning (2014).

A mancha alvo, caracteriza-se por lesões que se iniciam por pontuações pardas com halos amarelados, evoluindo para grandes manchas circulares de coloração castanho-claro. Normalmente as manchas apresentam uma pontuação escura no centro, semelhante a um alvo. (Henning, 2014). Na figura 2 a seguir temos imagens de plantas de soja infectada com mancha alvo.

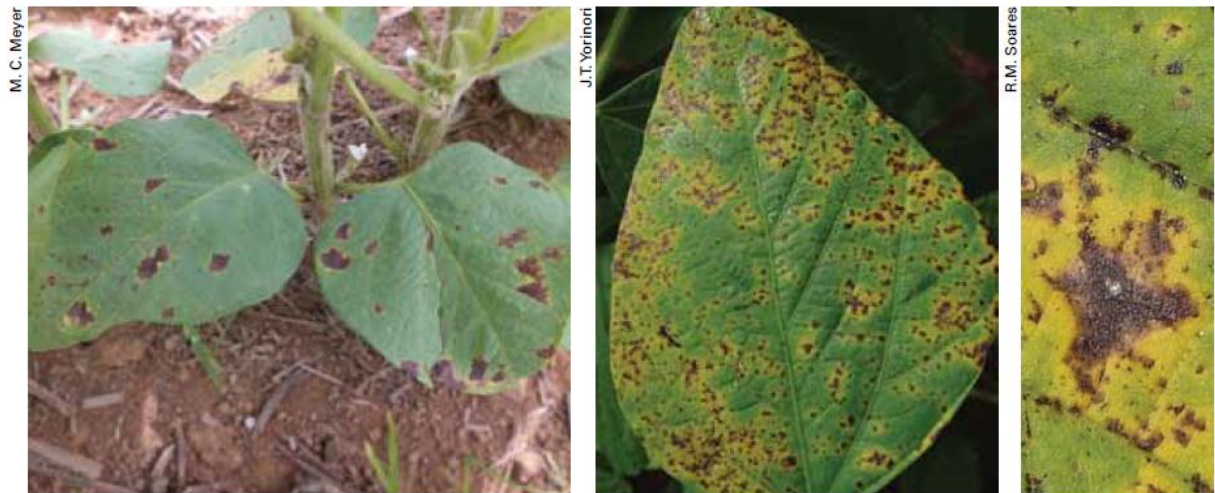
Figura 2: Folha de soja contaminada com mancha alvo.



Fonte: Henning (2014).

Já a mancha parda atinge a planta duas semanas após a emergência, e os sintomas são pequenos pontos ou manchas com contornos angulares avermelhados. Já com a evolução das doenças, as folhas apresentam manchas pardas de até 4mm de diâmetro, conforme vê-se na figura 3 abaixo.

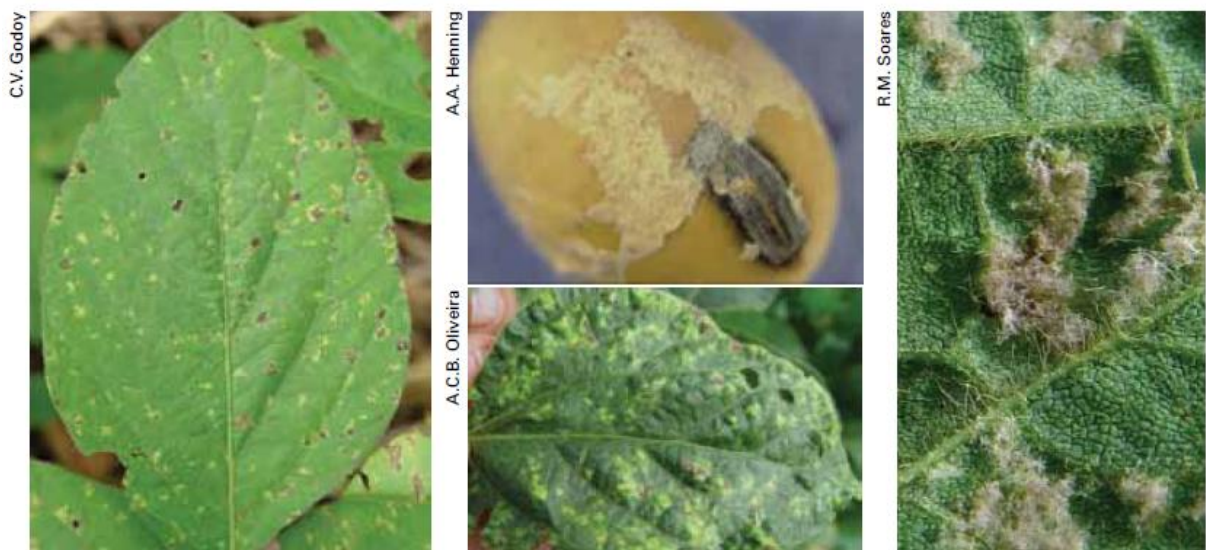
Figura 3: Folha da soja contaminada com mancha parda.



Fonte: Henning (2014).

Outra doença abordada para aplicar a solução de identificação por meio do processamento de imagens com aprendizado de máquina é o míldio. Os sintomas dessa doença se caracterizam pelo aparecimento de manchas de cor verde clara que evoluem para a amarela na parte superior da folha. Estas manchas podem medir aproximadamente 5mm, e podem causar a deterioração da semente ou infecção parcial. A figura 4 mostra as características de uma planta com míldio. (Henning, 2014).

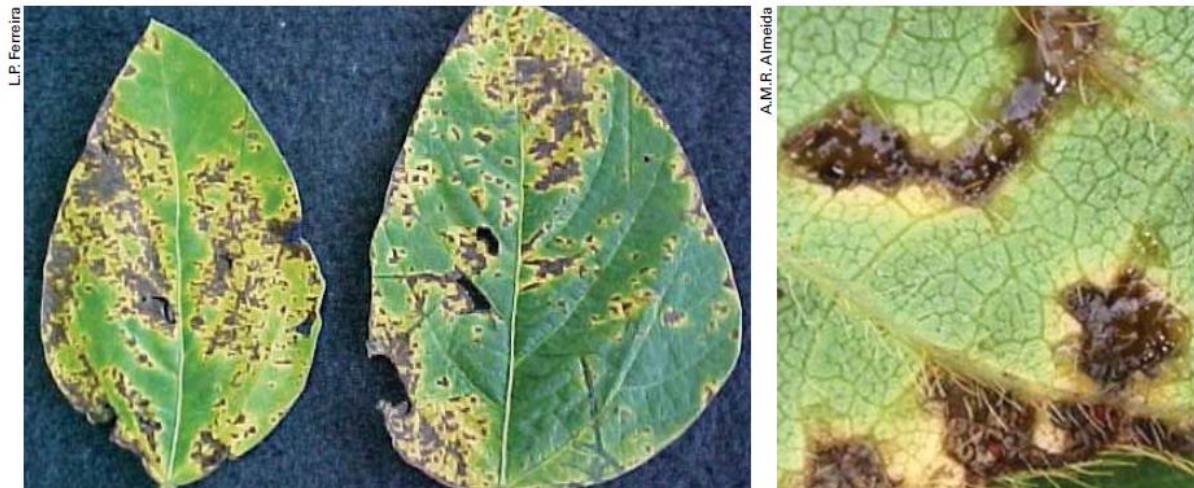
Figura 4: Folha da soja infectada com a doença Míldio



Fonte: Henning (2014).

A quinta e última doença é a cretamento bacteriano. É uma doença que normalmente atinge as folhas, mas pode atacar a haste, pecíolo e vagem. Seus sintomas são manchas aquosas, semitransparentes, que necrosam e aglutinam, formando áreas de tecido morto, como observado na figura 5 logo abaixo.

Figura 5: Folhas da soja contaminadas com cretamento bacteriano



Fonte: Embrapa (2014).

Como podemos observar, todas essas doenças permitem identificação visual a partir das folhas da soja, pois expressam sintomas com características específicas para cada tipo de doença.

2.3 APRENDIZADO DE MÁQUINA

Uma vez que essas doenças exigem uma análise visual de suas características para que possam ser identificadas, podemos tomar proveito da tecnologia para tornar esse processo mais assertivo, rápido, eficiente e automatizado, trazendo assim ganhos significativos no combate de tais fatores impactantes.

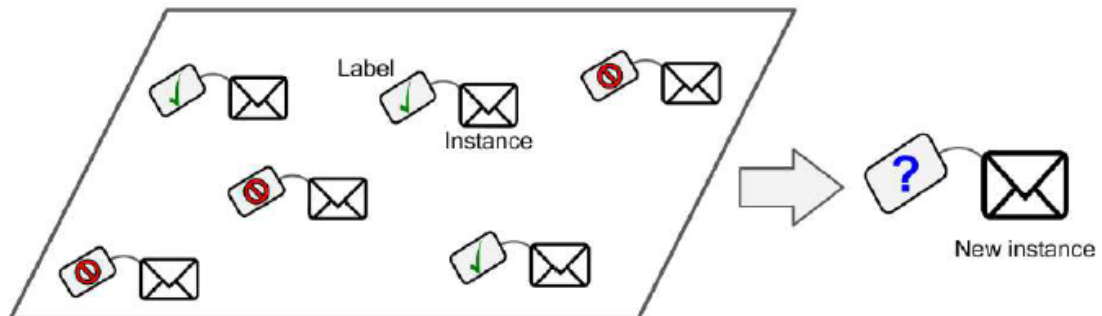
Uma das tecnologias que nos permite fazer isso é o Aprendizado de Máquina.

Podemos definir o Aprendizado de Máquina como a ciência de programar um computador para que ele possa aprender algo a partir de um ou mais conjuntos de dados.

Um ótimo exemplo de um programa baseado no método de Machine Learning é o filtro de *Spam* de e-mails. A partir de alguns exemplos de e-mails previamente marcados como *Spam*, o programa é capaz de processar novos e-mails e, ao identificar características suficientemente

semelhantes, classificá-los automaticamente como spam ou não, sem que seja necessário intervenção humana direta, apresentado na figura 6 (Géron, 2022).

Figura 6: Ilustração de um processo de treinamento de um sistema *Anti-Spam*.



Fonte: Géron (2022).

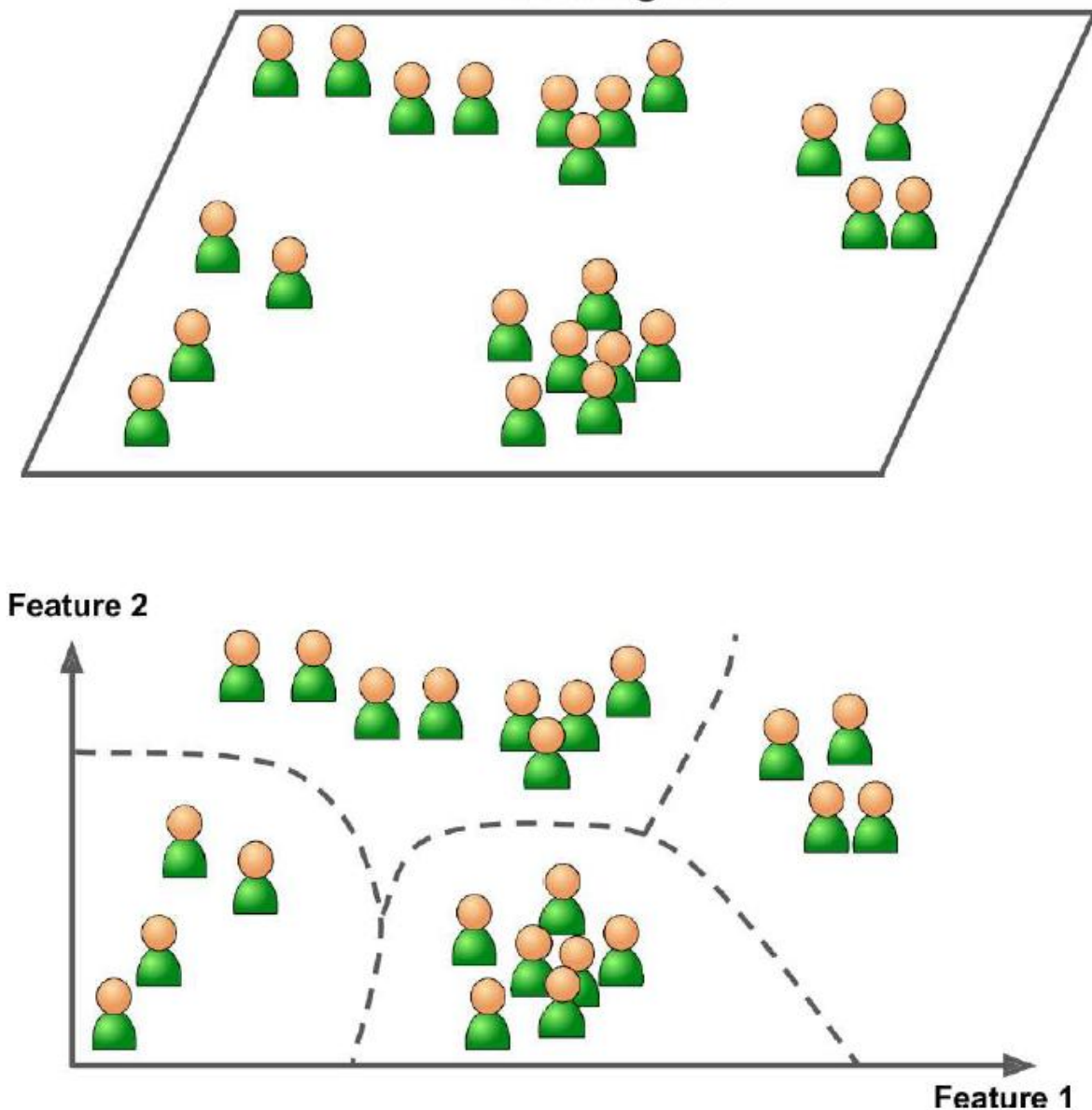
Uma vez que temos a definição do que é Aprendizado de Máquina, podemos então compreender como e de que formas a máquina ou o algoritmo de fato aprende algo.

Os três principais tipos de aprendizado de máquina são: Aprendizado supervisionado, Aprendizado semissupervisionado e Aprendizado não-supervisionado.

No aprendizado não-supervisionado, os dados fornecidos ao algoritmo não possuem rótulos ou categorização alguma, de início aparenta ser o mais difícil entre todos, uma vez que o objetivo do algoritmo é aprender algo a partir de um conjunto de dados, porém, sem ser informado os meios de como fazer isso. Nesta abordagem, usa-se métodos de agrupamentos baseados em similaridade por características, a fim de identificar padrões entre os dados fornecidos.

Na figura 7 abaixo, temos uma simples demonstração visual, onde dado um conjunto de dados não rotulados, o algoritmo trabalha a fim de identificar padrões a fim de agrupar os elementos por proximidade ou similaridade.

Figura 7: Processo de agrupamento (clusterização) não-supervisionado



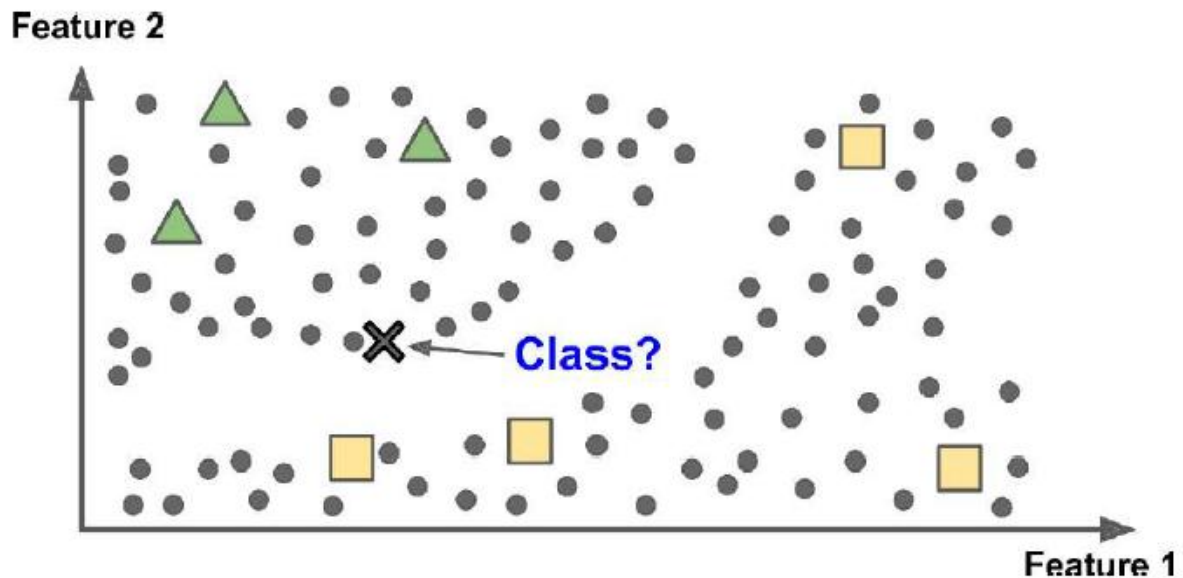
Fonte: Géron (2022).

Como podemos observar, Feature 1 e Feature 2 representam atributos ou características que são usadas no processo de agrupamento dos dados.

Já no aprendizado Semissupervisionado, temos uma combinação de dados com e sem rótulos.

Conforme a demonstração na figura 8 abaixo, temos uma ilustração de conjunto de dados parcialmente rotulados, representados pelas classes triângulo e quadrado, e os dados não rotulados representados pelos círculos. A partir dessas informações os algoritmos realizam um processo de treinamento a fim de rotular os dados que não foram previamente categorizados.

Figura 8: Ilustração aprendizado de máquina Semissupervisionado



Fonte: Géron (2022).

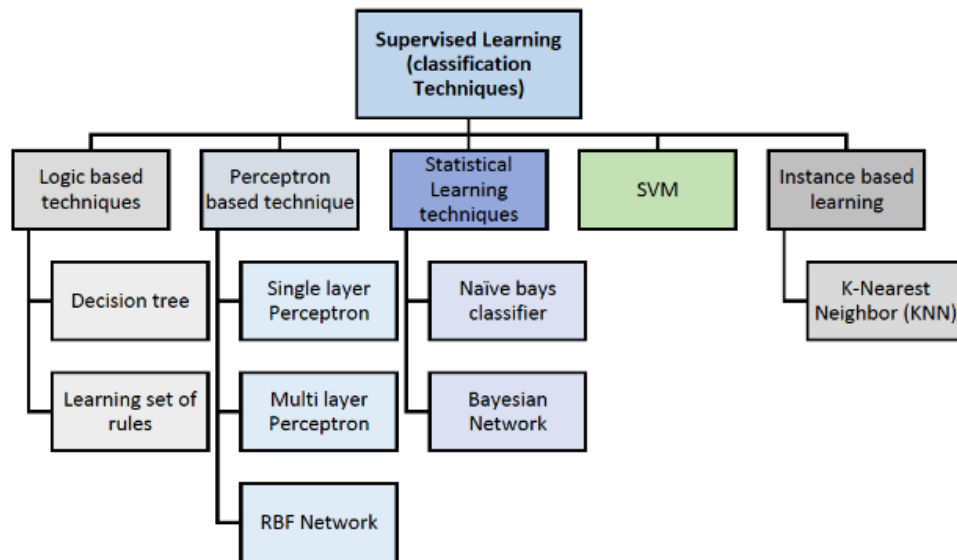
Por último temos o aprendizado supervisionado, onde o conjunto de dados fornecidos são previamente rotulados. Com base nesses rótulos, o algoritmo aprende a identificar padrões e características de cada categoria, tornando-o capaz de classificar novas entradas de dados de acordo com a similaridade das características apresentadas, conforme apresentado na figura 7. (Zhang, 2010).

Dentro da técnica de aprendizado de máquina supervisionado, temos 2 categorias, são elas:

- Regressão: modelo treinado para prever um valor contínuo e numérico a partir de um conjunto de dados rotulados.
- Classificação: modelo treinado para prever e agrupar instancias de dados por categorias discretas, a partir de conjuntos de dados também rotulados.

A classificação é uma abordagem de mineração de dados que é utilizada para prever a categoria ou grupo no qual um determinado dado pertence. Existem diversas técnicas de classificação, cada uma com suas características e aplicações específicas. Abaixo, temos a figura 9 que ilustra um modelo geral das técnicas de classificação disponíveis. (Soofi, 2017).

Figura 9: Modelo geral de aprendizado de máquina supervisionado e técnicas de classificação



Fonte: Soofi (2017)

2.4 REDES NEURAIS ARTIFICIAIS

O termo Redes Neurais Artificiais (RNA) ou Artificial Neural Networks (ANN), tem suas origens na busca para encontrar representações matemáticas do processamento de informações baseados em sistemas biológicos, ou seja, o conceito por trás disso é entender como o próprio cérebro e demais sistemas biológicos processam as informações e a partir daí, criar hipóteses e modelos matemáticos para replicar esse comportamento. (Bishop, 2006).

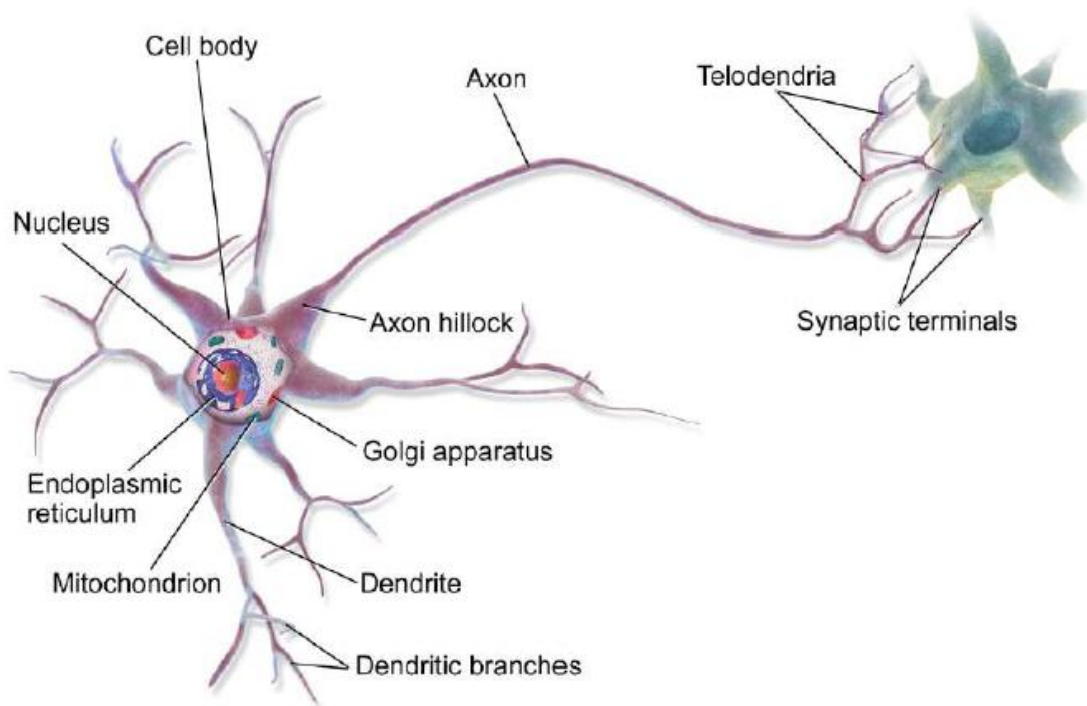
Olhando para a história, vemos vários exemplos de criações inspiradas na natureza e/ou nos sistemas biológicos. Os pássaros inspiraram o homem a voar, uma pequena planta chamada *Arctium* inspirou a criação do velcro, e da mesma forma, a arquitetura do cérebro humano é a inspiração que nos leva a tentar desenvolver algo que seja capaz de construir uma espécie de máquina inteligente.

Para que possamos compreender o funcionamento das ANNs, primeiro precisamos revisitar a estrutura de um neurônio biológico.

De uma forma bem simplificada, o neurônio é composto por um corpo celular que possui o núcleo e a maioria dos componentes celulares complexos. Ele contém muitas extensões chamadas dendritos, e uma outra extensão bem mais longa chamada axônio. Próximo a sua

extremidade, o axônio se divide em muitas ramificações que se chamam telodendros, e na ponta dessas ramificações estão as estruturas chamadas terminais sinápticos, que se conectam aos dendritos ou corpos celulares de outros neurônios. O neurônio produz impulsos elétricos bem curtos, que são chamados de sinais, os quais viajam pelos axônios e fazem com que as sinapses liberem sinais químicos chamados neurotransmissores, e são esses neurotransmissores que ativam ou inibem a ativação de sinais nos neurônios que o recebem. (Raschka, 2015). Na figura 10 é apresentado a formação de neurônios biológicos.

Figura 10: Estrutura de um neurônio biológico

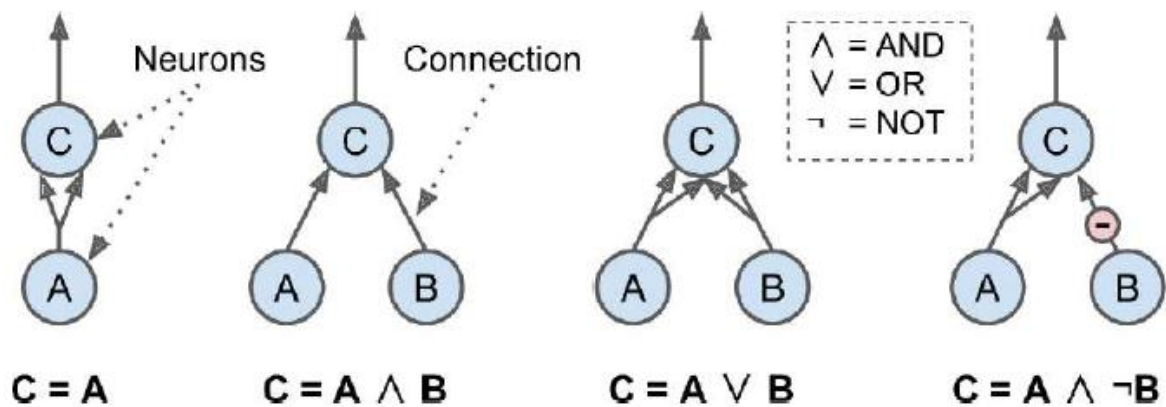


Fonte: RASCHKA (2015)

Embora possa parecer uma representação bem simples de um neurônio, sua estrutura é muito mais complexa do que aparenta ser, pois cada neurônio está integrado a uma imensa rede formada por milhares de outros neurônios interconectados, formando assim uma estrutura altamente sofisticada.

No âmbito dos neurônios artificiais, temos uma proposta bem simples do seu funcionamento, no qual há um ou mais entradas binárias (*ON/OFF*) e uma saída binária. Este neurônio artificial faz a ativação de sua saída (*ON*) quando um certo número de entradas é ativado, conforme apresentados na figura 11.

Figura 11: Ilustração da lógica por trás de um neurônio artificial



Fonte: Raschka (2015)

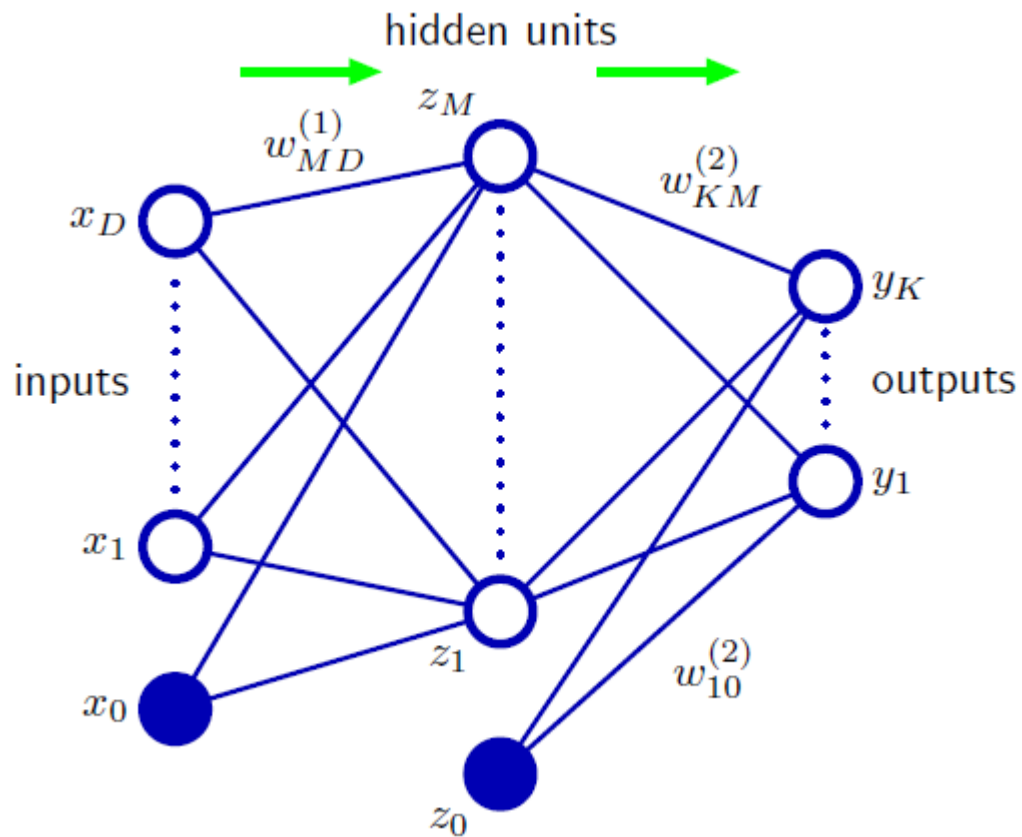
Descrevendo o que temos na imagem:

- No primeiro desenho temos somente 2 neurônios (A e C). Se o neurônio A está *ON*, então o neurônio C será *ON*, mas se o A for *OFF*, então o C também será *OFF*.
- Na segunda ilustração temos uma rede que possui uma lógica condicional “E”, logo, o neurônio C será *ON* somente se os neurônios A e B estiverem ativos (*ON*).
- No terceiro modelo, temos a condicional “OU”, que significa, se somente um dos neurônios de entrada forem *ON*, já é o suficiente para a ativação do neurônio de saída C.
- No quarto e último desenho, temos a junção de duas condicionais, são elas: OU e NÃO. Neste modelo temos uma ação de inibição de ativação, o que acontece com a estrutura biológica também, onde um impulso elétrico tem a ação de desativação de um sinal, e não ativação. Neste exemplo, o neurônio C será ativado somente se o neurônio A for *ON* e o neurônio B for *OFF*. (Raschka, 2015).

As ANNs podem ter uma ou múltiplas camadas ocultas. Essas camadas ocultas são as camadas responsáveis por processar e transformar as informações obtidas das camadas de entrada, e como resultado extrair padrões e características dos dados, conforme diagrama representado na figura 12 abaixo. É nesta etapa do processo que ocorre o que chamamos de aprendizado, onde o algoritmo deve ser capaz de resolver o problema que foi proposto, no caso

deste trabalho, a classificação de doenças na soja.

Figura 12: Diagrama de uma rede neural artificial não profunda



Fonte: BISHOP (2006)

A denominação rede neural artificial não profunda ou profunda se baseia na quantidade de camadas ocultas de uma arquitetura. Considera-se não profunda quando se tem até 2 camadas ocultas, e profunda quando contém 3 ou mais. Redes neurais profundas (*deep neural network – DNN*) podem ter milhares de camadas ocultas.

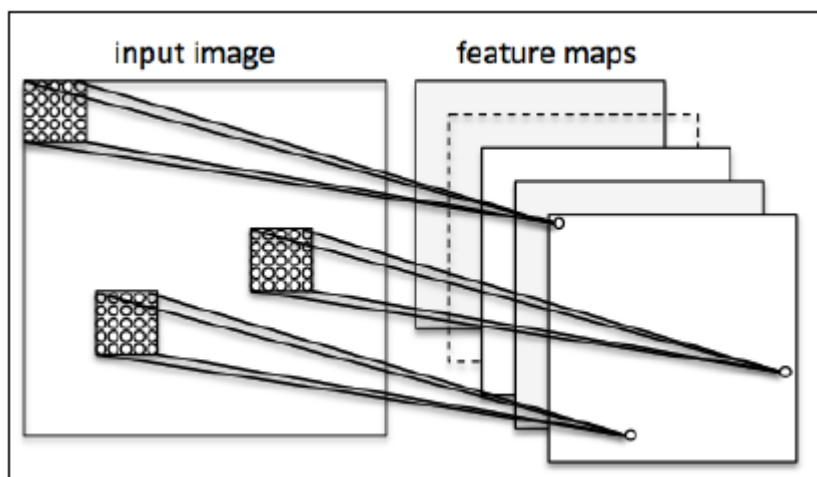
2.5 REDES NEURAIS CONVOLUCIONAIS PARA CLASSIFICAÇÃO DE IMAGENS

Um dos métodos mais indicados para classificação de imagens são os algoritmos baseados em Redes Neurais Convolucionais ou CNN (*Convolutional Neural Networks*), pois possuem ótimos resultados e performance neste tipo de tarefas. As CNNs são redes neurais muito populares nas arquiteturas de aprendizado profundo (*deep learning*), onde a ideia principal por trás das CNNs é construir várias camadas de detecção de características onde será considerado a disposição espacial dos pixels em uma imagem.

Diferente de algumas técnicas de classificação de imagens, como a *multi layer perceptron* (MLP), a qual transforma as imagens em vetores lineares de características, as CNNs não transformam as imagens em vetores, mas ao invés disso, utiliza o que chamamos de campos receptivos para analisar pequenos blocos ou janelas da imagem de cada vez. Essas janelas se comportam como filtros que se movem pela imagem a fim de detectar bordas, texturas e demais padrões que possam ser identificados. Esses campos receptivos podem ser entendidos como blocos sobrepostos que deslizam sobre os pixels da imagem de entrada com o objetivo de criar um mapa de características, onde o processo de criação desse mapa de características é chamado de convolução. (Raschka, 2019).

A figura 13 abaixo ilustra uma camada de convolução, onde os pixels de entradas são conectados ou mapeados a cada unidade do mapa de característica extraída do processamento da imagem.

Figura 13: Ilustração do mapeamento de características da CNN

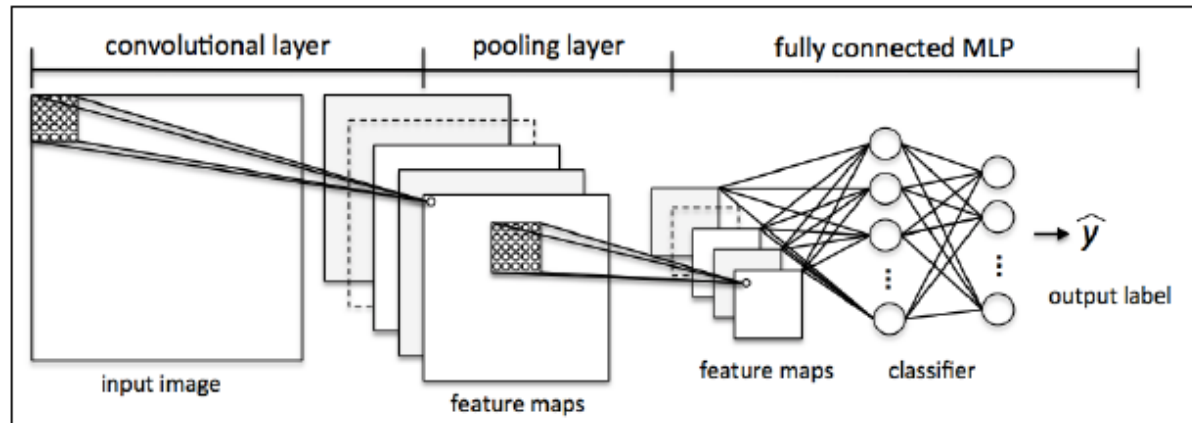


Fonte: Raschka (2015)

Um ponto importante a se destacar dentro deste processo é que os detectores de características são replicados, ou seja, os campos receptivos que mapeiam as características a cada unidade da próxima camada compartilham os mesmos pesos. O objetivo principal disso é que se um detector de característica foi útil em alguma parte da imagem, ele pode ser útil em outros blocos da imagem também.

Em CNNs, uma camada convolucional é seguida por uma camada de *Pooling*. Essa camada pode ser definida como um método de extração de características, no qual utilizamos o valor médio ou máximo de uma região de características vizinhas o qual é passado adiante para a próxima camada. A fim de se criar uma CNN profunda, basicamente é necessário empilhar múltiplas camadas de convolução e camadas de *Pooling*, antes de conectá-la a um MLP para a classificação final, onde, baseada nas características obtidas, essa última camada terá condições de classificar a que classe ou categoria a imagem pertence, como pode-se ver na ilustração da figura 14 a seguir.

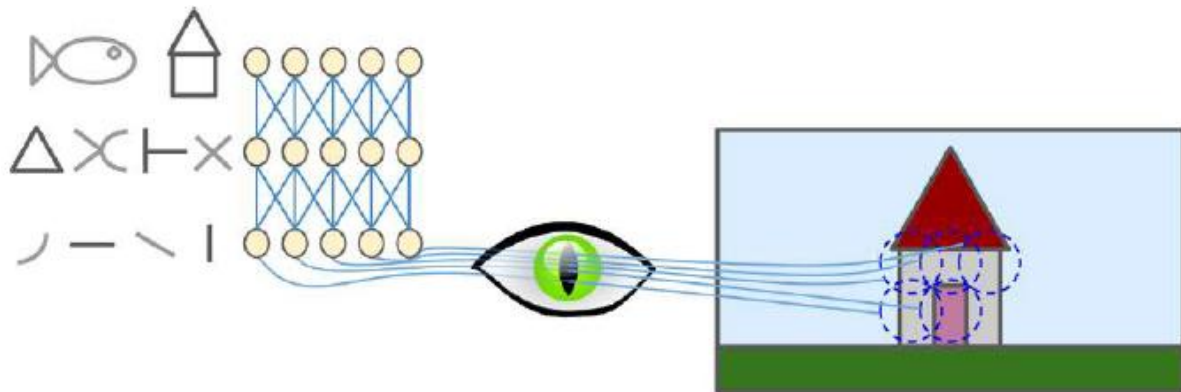
Figura 14: Ilustração completa das múltiplas camadas



Fonte: Raschka (2015)

Uma forma de exemplificar as camadas de convolução de uma perspectiva mais humana, vamos observar a imagem da figura 15 abaixo, na qual ilustra uma imagem sendo visualizada e processada por um neurônio biológico e identifica padrões em pequenas regiões da imagem.

Figura 15: Ilustração da visão humana

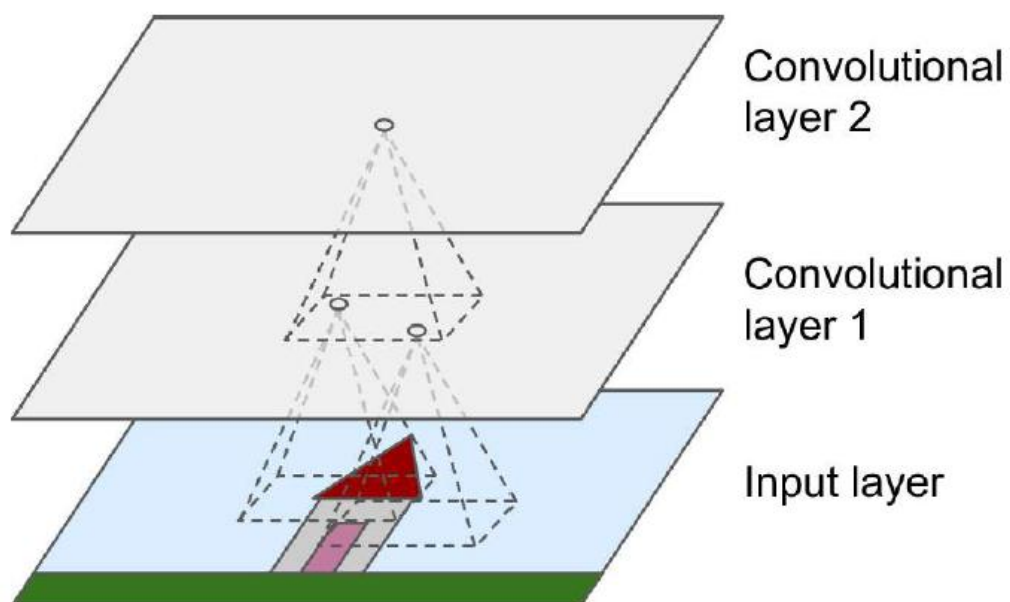


Fonte: Géron (2022)

Como podemos ver, diversos padrões são identificados quando olhamos para uma imagem, e mais características podem ser extraídas quando focamos em partes menores dessa mesma imagem.

Nas CNNs o processo é semelhante a isso, os neurônios das camadas de convolução focam em pequenas partes das imagens, como explicamos acima, e identificam os padrões daquele bloco de pixels. É importante salientar que os neurônios das camadas seguintes não processam o mesmo bloco que foi processado anteriormente, mas focam em blocos localizados em seu respectivo campo de detecção, conforme apresentado na Figura 16.

Figura 16: Ilustração das camadas de convolução

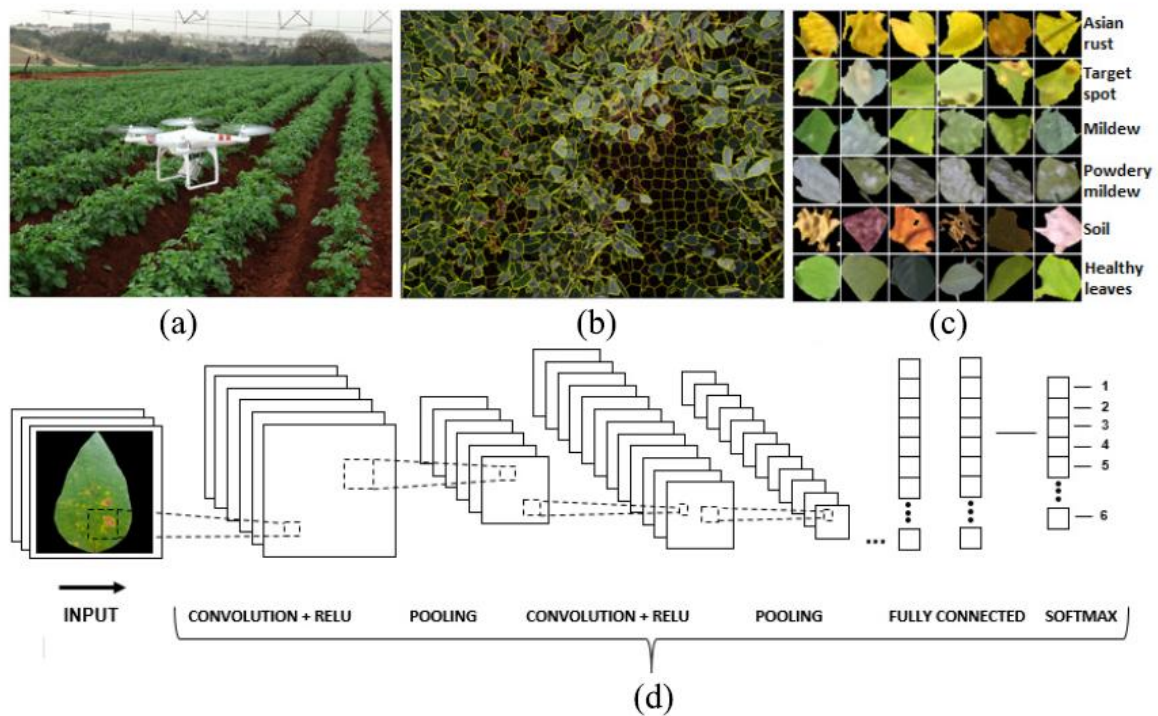


Fonte: Géron (2022)

Esta arquitetura permite que a rede se concentre em um bloco de características menor na primeira camada oculta, e na sequência, reúna essas características em um bloco maior, e assim sucessivamente para as demais camadas convolucionais ocultas.

Por todas essas características, usaremos um modelo de rede convolucional para realizar a classificação e identificação de doenças na cultura da soja. Uma vez que as imagens das plantas são obtidas, realizamos o treinamento supervisionado do modelo CNN para que o mesmo possa extrair os padrões e características de cada classe de imagem, seja de uma planta saudável ou doente. Na figura 17 a seguir mostramos um diagrama do processo de treinamento do modelo, que inicia com a obtenção das imagens (a), segmentação em classes (b), criação do *Dataset* (c) e classificação (d).

Figura 17: Diagrama do processo de treinamento do modelo CNN



Fonte: Tetila (2019)

2.6 INFERÊNCIA E AVALIAÇÃO

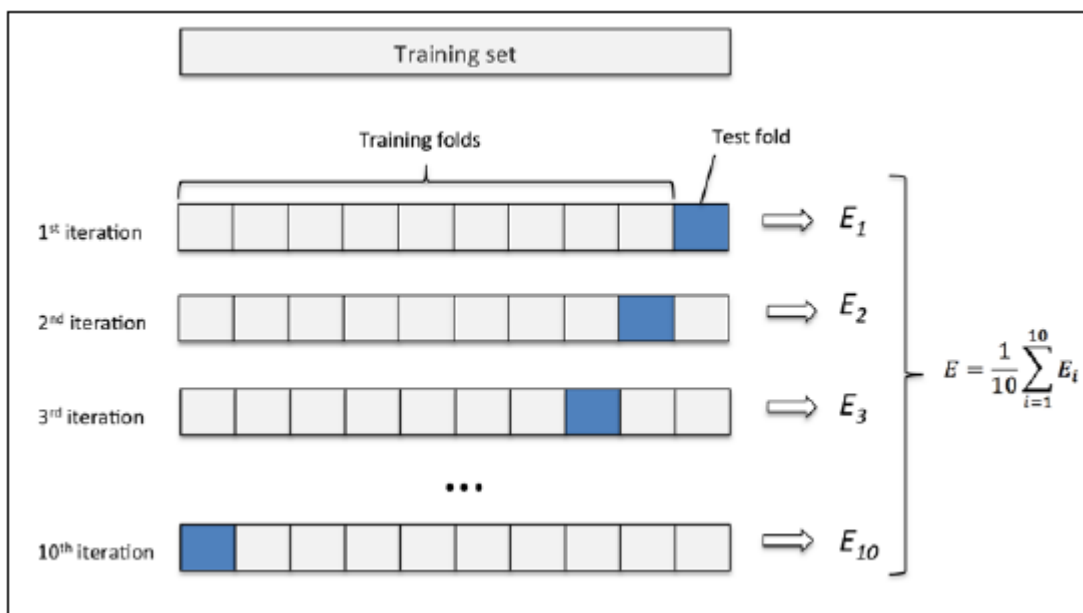
O processo de inferência e avaliação de um modelo de aprendizado de máquina é utilizado para aferir o quanto este modelo de fato aprendeu na etapa de treinamento, ou seja, medir o desempenho e capacidade de generalização deste modelo.

Uma das técnicas mais utilizadas para este tipo de validação é o *k-fold Cross-Validation*. Esta técnica divide o conjunto de dados em k partes iguais, e repete o processo de treinamento e validação k vezes. Em cada repetição, o algoritmo utiliza 1 parte (*fold*) do conjunto de dados para validação e o restante das partes (*folds*) são utilizados para treinamento.

Para exemplificar, digamos que definimos o número de $k=10$, ou seja, o conjunto de dados será dividido em 10 partes iguais, de modo que, 9 partes serão utilizados no processo de treinamento do modelo, e 1 parte será utilizada para validação. Este processo se repetirá por 10 vezes, em cada uma das repetições uma nova parte dos dados será utilizada para teste, sendo uma parte diferente das anteriores, para que os dados de teste sejam sempre dados não utilizados anteriormente para validação, e as outras 9 partes são utilizadas no treinamento. (Raschka, 2015).

A figura 18 a seguir ilustra este processo.

Figura 18: Ilustração funcionamento do *K-fold cross-validation*



Raschka (2015)

Em cada repetição é gerado uma avaliação, que corresponde a acurácia ou percentual de acertos em relação a quantidade de previsões feitas pelo modelo. A acurácia é calculada a partir da porcentagem de amostras positivas e negativas classificadas corretamente dividido pela soma total de amostras, apresentado na equação 1, dada como:

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

Abaixo tabela (tabela 1) de detalhamento:

Tabela 1: Resultados de classificação possíveis

Valor	Descrição
<i>Verdadeiro Positivo (VP)</i>	O algoritmo classificou como verdadeiro, e de fato o resultado é verdadeiro.
<i>Verdadeiro Negativo (VN)</i>	O algoritmo classificou como falso, e de fato o resultado é falso.
<i>Falso Positivo (FP)</i>	O algoritmo classificou como verdadeiro, e o resultado é falso.
<i>Falso Negativo (FN)</i>	O algoritmo classificou como falso, e o resultado é verdadeiro.

Fonte: Elaborado pelo autor

3 MATERIAIS E MÉTODOS

O presente estudo tem por objetivo classificar imagens de plantas de soja em 6 diferentes classes: planta saudável, planta com ferrugem asiática, planta com cretamento bacteriano, planta com mancha alvo, planta com mancha parda e planta com míldio. As imagens para treinamento do algoritmo foram obtidas a partir do repositório de conjunto de dados (*Datasets*) aberto (*open source*) para utilização em projetos de visão computacional *Universe Roboflow*.

3.1 CONJUNTO DE DADOS

O conjunto de dados nomeado como *Soyscan_Dataset* foi produzido por Rayhan Adi Wicaksono e disponibilizado no repositório *Universe Roboflow* em 31/03/2024. (Wicaksono, 2024). Este conjunto de dados possui 6891 imagens, dividido nas seguintes classes: saudável com 1610 imagens; cretamento bacteriano 464 imagens; ferrugem asiática, com 1605 imagens; macha alvo, com 1520 imagens; mancha parda, compondo 1060 imagens; e míldio, com 632 imagens.

As imagens foram pré-processadas, onde aplicou-se os seguintes ajustes:

- 1- Auto orientação de dados de pixels com remoção de *EXIF-orientation*: Significa remover do metadados da imagem os dados referentes a orientação em que a foto foi tirada (retrato ou paisagem);
- 2- Redimensionamento das imagens para 640x640: Realizado redimensionamento para manter um padrão de entrada de imagens.

Abaixo alguns exemplos de imagens do conjunto de dados:

Na figura 19, uma folha de soja saudável foi apresentada, a qual não possui nenhuma característica de infecção por doenças.

Figura 19: Exemplo de imagem que compõe o *Dataset*, contendo uma planta saudável



Fonte: Wicaksono (2024)

Figura 20: Exemplo de imagem que compõe o *Dataset*, contendo uma planta doente com cretamento bacteriano



Fonte: Wicaksono (2024)

Na figura 20 foi apresentado uma folha de soja contaminada por cretamento bacteriano, a qual apresenta manchas e necroses dos tecidos da planta, e causam rasgamento dos espaços inter-nervais da folha e até queda. (Henning, 2014).

Figura 21: Exemplar de imagem que compõe o *Dataset*, contendo uma planta doente com ferrugem asiática



Fonte: Wicaksono (2024)

A figura 21 apresenta um exemplo com a doença ferrugem asiática, pode-se observar os pequenos pontos mais escuros que se diferem do tecido sadio da folha, os quais causam a desfolha da planta, resultando na diminuição severa dos grãos de soja. (Henning, 2014).

Figura 22: Exemplar de imagem que compõe o *Dataset*, contendo uma planta doente com míldio



Fonte: Wicaksono (2024)

A figura 22 (acima) apresenta uma folha contaminada por míldio, doença que apresenta como sintoma pequenas manchas verde-claras que podem evoluir para amarela. E como

impacto da doença pode ocorrer a deterioração da semente ou infecção parcial. (Henning, 2014).

Figura 23: Exemplar de imagem que compõe o *Dataset*, contendo uma planta doente com mancha alvo



Fonte: Wicaksono (2024)

Na figura 23 foi apresentada uma cultura contaminada com mancha alvo. Essa doença é caracterizada por manchas de cor castanho causada por fungos, os quais contaminam não só as folhas, mas também as raízes. (Henning, 2014).

Figura 24: Exemplar de imagem que compõe o *Dataset*, contendo uma planta doente com mancha parda



Fonte: Wicaksono (2024)

Neste último exemplar (figura 25) pode-se ver uma folha contaminada pela doença mancha parda, a qual apresenta pequenas pontuações de avermelhadas e pode causar severa desfolha da planta, afetando o desenvolvimento dos grãos. (Henning, 2014).

3.2 MODELO PROPOSTO

A arquitetura da rede neural foi definida de acordo com a natureza dos dados e objetivo almejado. O diagrama a seguir ilustra o modelo utilizado neste trabalho, conforme segue figura 25.

Na tabela 2, foi detalhado as camadas do modelo proposto.

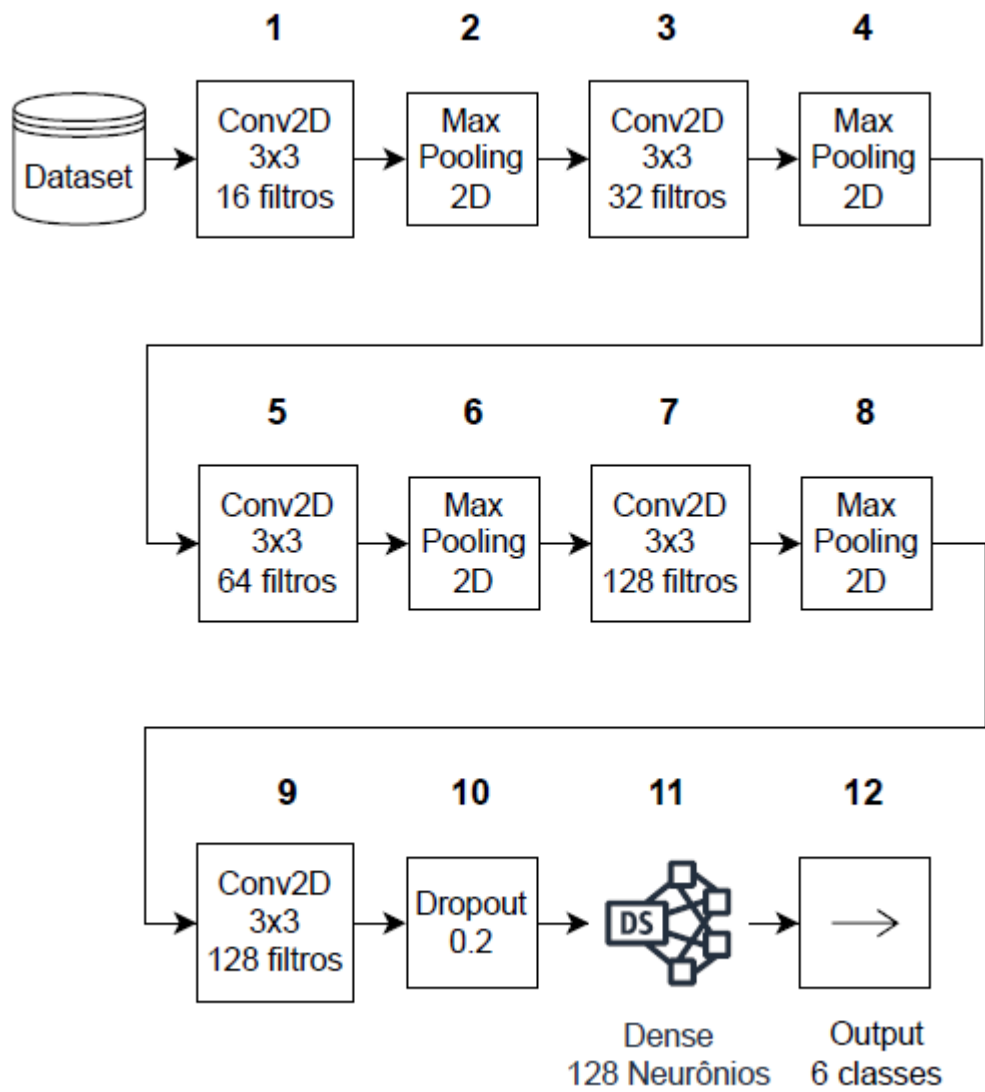
Tabela 2: Arquitetura da CNN proposta no presente estudo.

CAMADA	DESCRIÇÃO
1	<i>Conv2D</i> : Camada de convolução com 16 filtros de características de tamanho 3x3, os quais são utilizados no processo de <i>feature map</i> , ou mapa de características. Definiu-se o parâmetro <i>padding="same"</i> a fim de manter as bordas de entrada e de saída sem alteração. A função de ativação ReLU (<i>Rectified Linear Unit</i>) foi utilizada a fim de ajudar o modelo no processo de aprendizado de relações não lineares.
2	<i>MaxPooling2D</i> : Camada de <i>Pooling</i> que tem por função reduzir o tamanho das imagens após o processamento da camada de convolução com o objetivo de diminuir a carga computacional, uso de memória e o número de parâmetros, resultando em um risco menor de <i>Overfitting</i> .
3	<i>Conv2D</i> : Segunda camada de convolução, agora com 32 filtros de características de tamanho 3x3, definido com o objetivo de aumentar a extração de características das imagens.
4	<i>MaxPooling2D</i> : Camada de <i>Pooling</i> intercalada entre cada camada de convolução, com as mesmas funções detalhadas na camada 2 desta tabela.
5	<i>Conv2D</i> : Terceira camada de convolução, agora com 64 filtros de características de tamanho 3x3. Seguiu-se aumentado a quantidade de filtros a fim de extrair o máximo de características das imagens processadas, mantendo-se os mesmos parâmetros mencionados nas camadas anteriores
6	<i>MaxPooling2D</i> : Camada de <i>Pooling</i> intercalada entre cada camada de convolução, com as mesmas funções detalhadas na camada 2 desta tabela
7	<i>Conv2D</i> : Quarta e última camada de convolução, com 128 filtros de características de tamanho 3x3. Seguiu-se aumentado a quantidade de filtros a fim de extrair o máximo de características das imagens processadas, mantendo-se os mesmos parâmetros mencionados nas camadas anteriores
8	<i>MaxPooling2D</i> : Última camada de <i>Pooling</i> intercalada entre cada camada de convolução, com as mesmas funções detalhadas na camada 2 desta tabela
9	<i>Dropout</i> : Ao final do empilhamento de camadas de convolução e <i>Pooling</i> adicionou-se uma camada de <i>dropout</i> definido em 0.2 (20%). Este parâmetro fez com que 20% dos neurônios fossem desativados durante o processo de treinamento com o objetivo de evitar <i>Overfitting</i> .

CAMADA	DESCRIÇÃO
10	<i>Flatten</i> : Camada responsável por transformar a saída em um vetor unidimensional a fim de adequar o formato da imagem para que pudesse ser processada pela camada densa.
11	<i>Dense</i> : Camada com 128 neurônios e função de ativação ReLU, responsável por analisar as características que foram extraídas e identificar padrões complexos.
12	<i>Output</i> : Camada de saída com 6 neurônios, que representam cada uma das 6 classes de saída do modelo.

Fonte: Elaborado pelo autor.

Figura 25: Arquitetura CNN



Fonte: Elaborado pelo autor

Antes de iniciar o treinamento do modelo, empregou-se a função *model.compile* para especificar o otimizador (*ADAM- Adaptive Moment Estimation*), a função de perda e as métricas de avaliação do modelo. Neste trabalho, utilizou-se a métrica de acurácia, ou seja, durante o processo de validação, o algoritmo calculou a proporção de previsões corretas em relação ao total de previsões feitas, conforme descrito na (Seção 2.6) deste estudo.

3.3 PROPOSTA DE EXPERIMENTO

Neste experimento buscou-se avaliar qual o melhor *framework* para implementação de uma rede neural convolucional para a classificação de imagens de folhas de soja, baseado em acurácia e tempo de treinamento, onde os *frameworks* experimentados foram: *TensorFlow* e *Pytorch*. Ambos se tratam de *frameworks open-source* para redes profundas, os quais disponibilizam APIs que facilitam a construção e interação de redes neurais profundas. O *Tensorflow* é um *framework* desenvolvido pelo *Google*, enquanto o *Pytorch* foi desenvolvido pelo *Facebook*.

Eles foram submetidos aos mesmos parâmetros de treinamento, avaliação e inferência, com os seguintes parâmetros: (1) Épocas: 100, 200 e 400; (2) Avaliação: Acurácia; e (3) *k-fold cross-validation*: 5.

3.4 INFERÊNCIA E AVALIAÇÃO

Para o presente estudo foi definido para o hiper parâmetro $k = 5$, referente a inferência *k-fold Cross-Validation*, apresentada na seção 2.6. Deste modo, o *Dataset* foi dividido em 5 *folds* iguais, sendo 4 *folds*, ou 80% do *Dataset* (5514 imagens) separado para realização do treinamento, e 1 *fold*, equivalente a cerca de 20% (1377 imagens), separado para validação. Este processo foi repetido 5 vezes, alternando os *folds* de treinamento e validação, conforme apresentado na seção 2.6.

Ao final de cada iteração, o qual foi processado de acordo com o número de épocas, foi extraído o valor de acurácia (Seção 2.6) do *fold* separado para validação. Após o processamento dos 5 *folds*, foi calculado a média da acurácia dos *folds* de validação, apresentado na equação

2.

$$eval_{acc} = \frac{\sum_{i=1}^k acc_i}{k}, \quad (2)$$

sendo acc o valor percentual da medida de acurácia do *fold* de validação da iteração i , e k o hiper parâmetro definido como 5 que é o número de *folds*.

3.5 MATERIAIS UTILIZADOS

Apesar de as imagens já terem passado por um processo de ajuste de dimensão e auto orientação de dados dos pixels, foi necessária uma etapa adicional de processamento como parte da preparação das imagens para serem submetidas a CNN. Esta etapa foi de suma importância, pois definiu-se o formato adequado de entrada dos dados que foram processados.

Nesta etapa de preparação, aplicou-se um *re-shape* da imagem para o tamanho de 180x180.

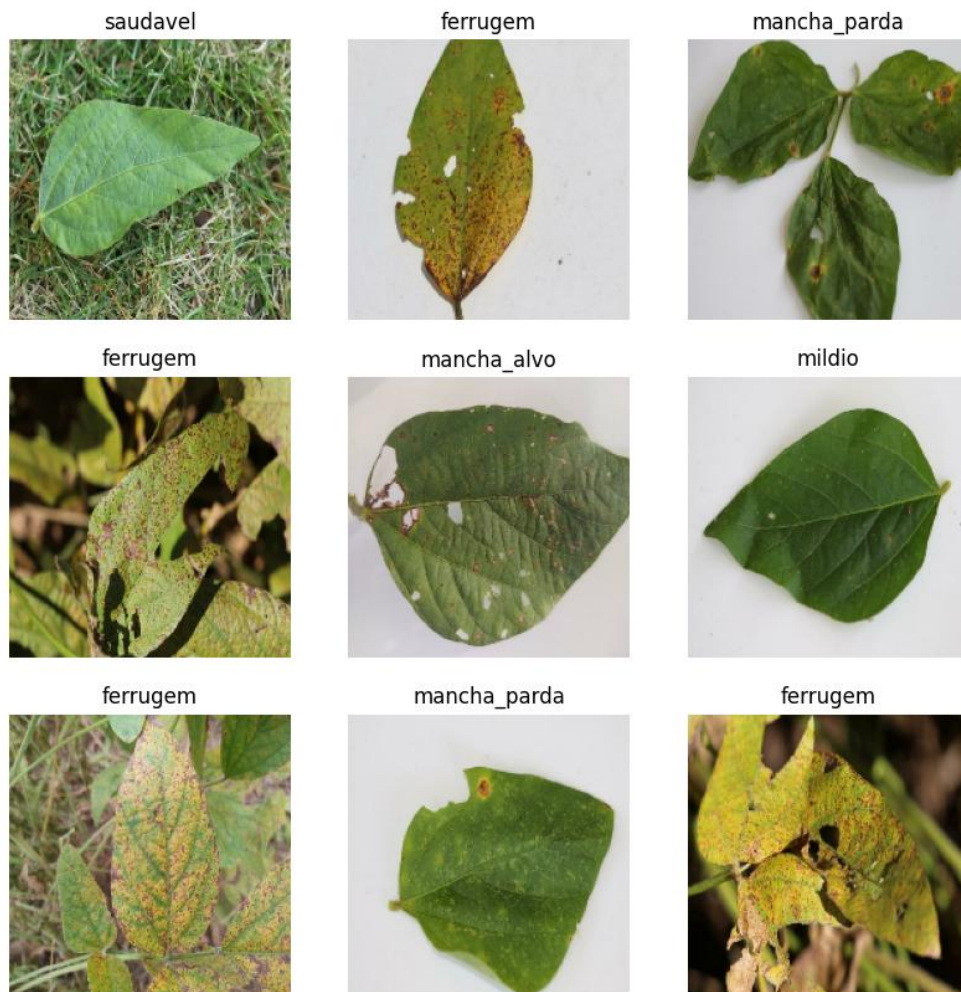
Todo o processamento para treinamento do modelo foi realizado utilizando um servidor virtual alocado na nuvem da Microsoft Azure, com a seguinte especificação: Tamanho da Instância: Standard B4as v2; vCPUs: 4; Memória Ram: 16GB; Espaço em Disco: 127GB; e Sistema Operacional: *Windows Server 2022 Datacenter Azure Edition*.

Detalhamento de materiais e métodos para o desenvolvimento do modelo:

- 1- Linguagem de programação: Python 3.10.11
- 2- Bibliotecas utilizadas: *Numpy*; *PIL (Python Imaging Library)*; *matplotlib.pyplot*; *tensorflow*; *tensorflow.keras*; *pathlib*; *time*; *os*; *torch*; *torchvision*.
- 3- O *Dataset* contendo as imagens foi disponibilizado no disco local do servidor virtual, categorizado de acordo com cada classe através da nomenclatura do diretório;
- 4- O tamanho do lote (*batch size*) é definido em 32;
- 5- Após o carregamento do *Dataset* e aplicação das etapas de redimensionamento e segmentação do conjunto de dados para treinamento e validação, foi obtido um exemplar do formato das imagens e suas respectivas classes, apresentados na figura

26:

Figura 26: Exemplos de imagens após etapa de pré-processamento e segmentação



Fonte: Imagens processadas pelo algoritmo do trabalho (2024)

- 6- Os valores de pixel RGB vão de 0-255, porém, estes valores não são ideais para algoritmos de redes neurais, portanto, foi necessário ajustar a escala desses valores para que ficasse entre 0-1, e para isso, dividiu-se os valores por 255.

4 RESULTADOS E DISCUSSÕES

Conforme descritos na Seção 3.3, foram realizadas 3 etapas de treinamento com cada um dos 2 frameworks, inicialmente com 100 épocas, em seguida 200 e finalmente com 400 épocas, a fim de avaliar o desempenho e tempo de treinamento entre ambos.

Para fins de avaliação e inferência, conforme detalhado nas seções 2.6 e 3.4, foi utilizado a métrica de acurácia para mensurar o desempenho obtido, assim como a técnica *k-fold*, com o hiper parâmetro 5, ou seja, para cada interação (*fold*), o algoritmo realizou o treinamento com a respectiva quantidade de épocas (100, 200 e 400). Ao final do processamento dos 5 *folds*, foi calculado a média da acurácia entre os *folds* para extrair os resultados mostrados a seguir.

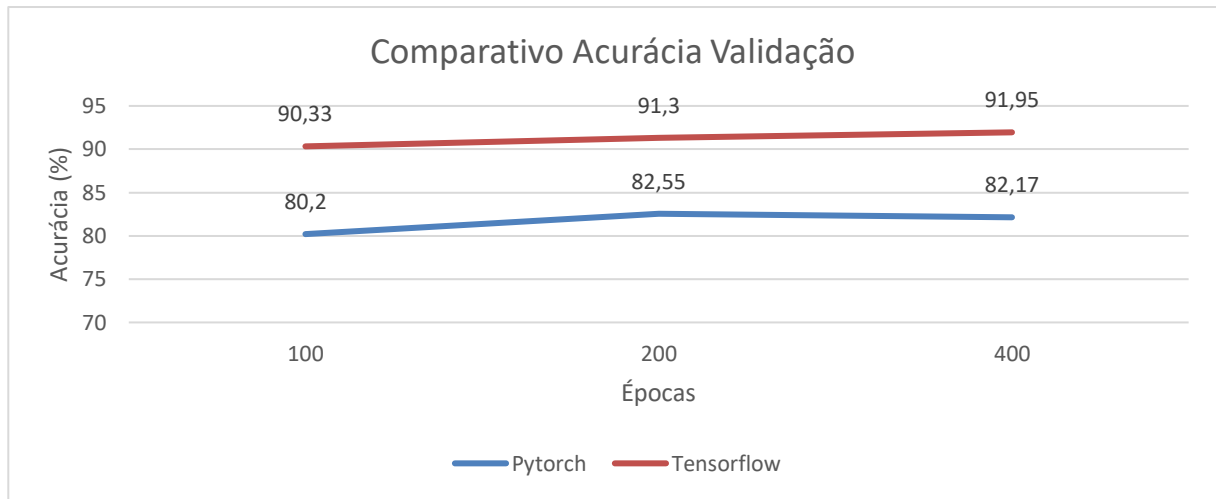
Abaixo temos alguns gráficos e tabelas que registram os resultados obtidos, iniciando pela tabela 3, a qual são apresentados os resultados de acurácia obtidos a partir do treinamento do modelo com 100, 200 e 400 épocas, comparando as implementações *Tensorflow* e *Pytorch*:

Tabela 3: Comparativo acurácia média entre os frameworks.

<i>Framework</i>	Épocas	Acurácia
<i>TensorFlow</i>	100	90,33 %
	200	91,3 %
	400	91,95 %
<i>Pytorch</i>	100	80,2 %
	200	82,55 %
	400	82,17 %

Fonte: Elaborado pelo autor.

Em seguida, pode-se observar a representação gráfica (gráfico 1) desses resultados:

Gráfico 1: Comparativo acurácia média entre *frameworks*

Conforme observado no Gráfico 1 e na Tabela 3, o *Tensorflow* apresentou resultados de desempenho de aprendizado superiores ao *Pytorch*, atingindo 91,95% de acurácia média sobre 82,17% do *Pytorch* no mesmo parâmetro de épocas.

Enquanto o *Tensorflow* apresentou melhores taxas de acurácia conforme se aumentou a quantidade de épocas, o *Pytorch* obteve melhor *performance* com 200 épocas, frente aos ciclos de 100 e 400 épocas.

Com relação ao tempo de processamento e treinamento, o *Tensorflow* alcançou melhores resultados, exigindo significativamente menos tempo para realização do processo de aprendizado e avaliação, como podemos observar na tabela 4 a seguir:

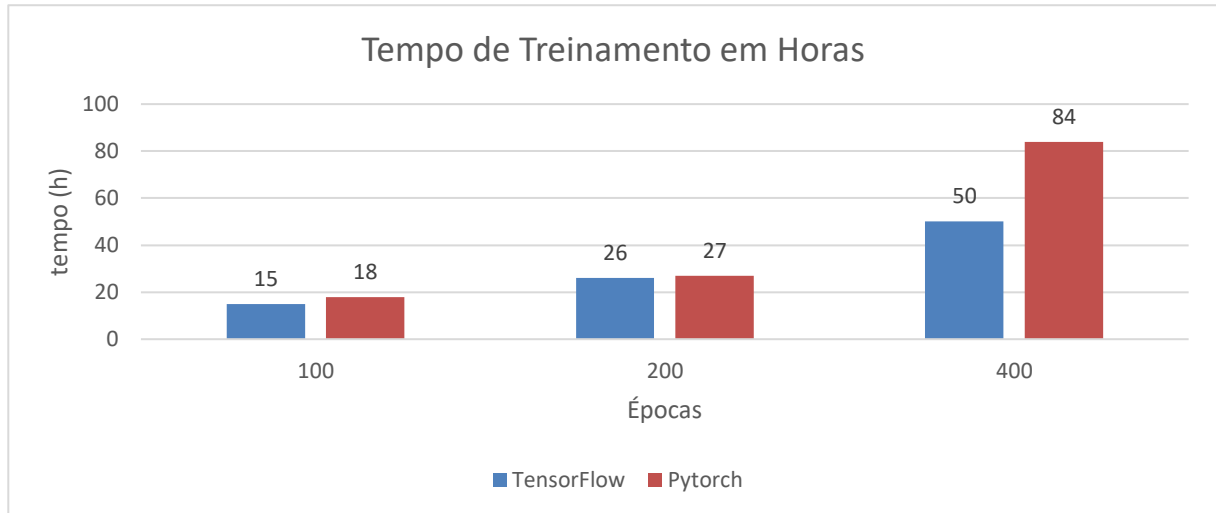
Tabela 4: Comparativo tempo de processamento e treinamento

<i>Framework</i>	Épocas	Tempo Treinamento (Horas)
<i>Tensorflow</i>	100	15
	200	26
	400	50
<i>Pytorch</i>	100	18
	200	27
	400	84

Fonte: Elaborado pelo autor.

Abaixo temos a representação gráfica (gráfico 2) da tabela 4, com os resultados de tempo de treinamento obtidos.

Gráfico 2: Comparativo tempo de processamento e treinamento



Fonte: Elaborado pelo autor.

Para os dois primeiros experimentos, com 100 e 200 épocas respectivamente, ambos frameworks apresentaram tempos de treinamento similares. No entanto, no último experimento com 400 épocas, o *Pytorch* exigiu 40% mais tempo de treinamento em relação ao *Tensorflow*.

Levando em consideração que os melhores resultados foram obtidos com o *Tensorflow* com 400 épocas, demonstra-se abaixo na tabela 5 um detalhamento deste experimento com seus respectivos resultados, *fold a fold*, onde observa-se a evolução das métricas de acurácia e tempo de treinamento/validação.

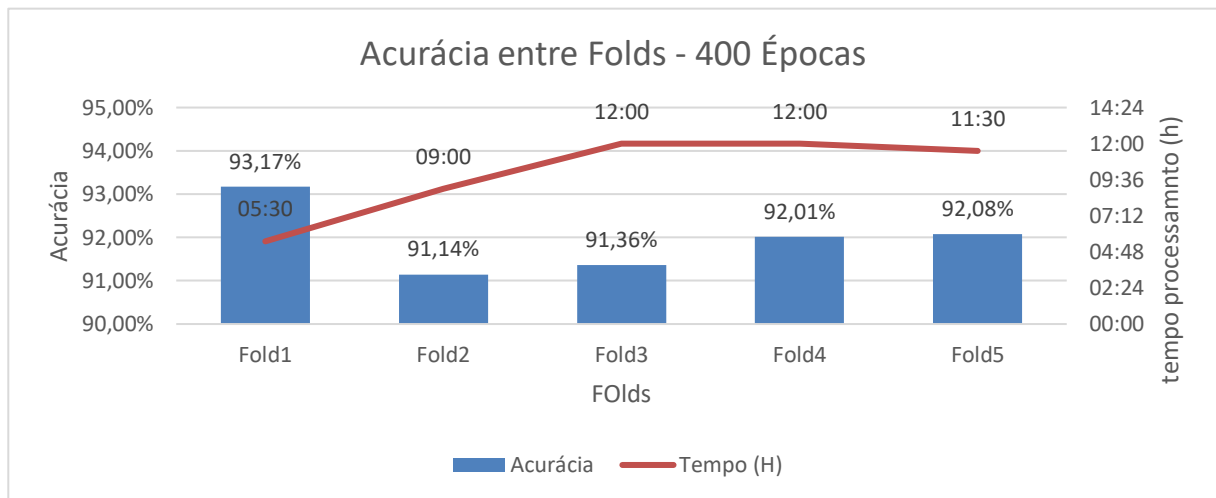
Tabela 5: Descritivo *Tensorflow* com 400 épocas e 5 folds

Evolução de acurácia e tempo de processamento por fold - 400 Épocas					
	<i>Fold1</i>	<i>Fold2</i>	<i>Fold3</i>	<i>Fold4</i>	<i>Fold5</i>
Acurácia	93,17%	91,14%	91,36%	92,01%	92,08%
Tempo (h)	05:30	09:00	12:00	12:00	11:30

Fonte: Elaborado pelo autor.

Em seguida, temos a representação gráfica (gráfico 3) do resultado apresentados na tabela anterior.

Gráfico 3: Descritivo Tensorflow com 400 épocas e 5 folds



Fonte: Elaborado pelo autor

Para este experimento foi necessárias aproximadamente 220 horas de processamento de imagens para extração dos resultados de acurácia entre os frameworks abordados, a um custo aproximado de R\$ 400,00 com computação em nuvem. Deste modo o modelo de CNN implementado utilizando o *framework TensorFlow* apresentou resultados melhores, uma acurácia cerca de 10% em média, comparados com o *PyTorch* (Tabela 3). O *TensorFlow* também apresentou menos horas de processamento (média 12,5 horas a menos) comparados ao *PyTorch*, garantindo um custo menor se a aplicação estiver sendo executada em serviços na Nuvem, como foi realizado este estudo.

5 CONCLUSÃO

A identificação de doenças na soja é de suma importância no combate aos fatores de impacto a produtividade e qualidade desta cultura. Os desafios em desenvolver soluções capazes de automatizar e agilizar este processo são enormes, onde destaco como o principal deles a obtenção e categorização de imagens suficientes para criação de um *Dataset* completo e confiável para treinamento de modelos de aprendizado de máquina.

Os experimentos apresentados levaram a constatação que o *framework Tensorflow* obteve um desempenho melhor para a solução de identificação e classificação de doenças na soja por meio do aprendizado de máquina e redes neurais artificiais profundas. Atingindo uma taxa de 91,95%, e exigindo até 40% menos tempo para treinamento do modelo CNN em relação ao segundo *framework Pytorch* (no experimento com 400 épocas). Ambos os experimentos foram validados utilizando a inferência *k-fold Cross-Validation* e a métrica de acurácia.

Os resultados são promissores com o uso de CNN implementados via *Tensorflow* para a solução do problema de identificação de doenças na cultura da soja, destaca-se a importância na continuidade deste tipo de experimento. Como trabalhos futuros espera-se aplicar o modelo proposto com maiores quantidade de épocas para treinamento, em conjuntos de dados com maior volume e qualidade de dados, com o potencial de alcançar resultados ainda mais expressivos no treinamento de modelos para aplicação deste tipo de solução em campo, com o objetivo de ganhos em produtividade e qualidade no setor.

REFERÊNCIAS

- BISHOP, Christopher M. **Pattern recognition and machine learning by Christopher M. Bishop**. Springer Science+ Business Media, LLC, 2006.
- CASTANHO, ROBERTO BARBOZA; TEIXEIRA, MATHEUS EDUARDO SOUZA. **A evolução da agricultura no mundo: da gênese até os dias atuais**. Brazilian Geographical Journal: Geosciences and Humanities research medium, Ituiutaba, v. 8, n. 1, p. 136-146, 2017.
- CASTRO-JÚNIOR, Roberto Pereira et al. **COMBINAÇÕES QUÍMICAS E INTERVALOS DE APLICAÇÃO: IMPACTOS NA FERRUGEM-ASIÁTICA DA SOJA**. 2020.
- DE QUEIROZ, Daniel Marçal et al. (Ed.). **Agricultura digital**. Oficina de Textos, 2022.
- FPA. **Importância da soja para o Brasil**. FDA Assuntos Temáticos, Resumos Executivos, 5 novembro 2021. Disponível em: <<https://fpagropecuaria.org.br/2021/10/18/importancia-da-soja-para-o-brasil>>. Acesso em: 13 de Setembro de 2024.
- GÉRON, Aurélien. **Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow**. " O'Reilly Media, Inc.", 2022.
- HENNING, Ademir Assis et al. **Manual de identificação de doenças de soja Embrapa**. 2014.
- HIRAKURI, Marcelo Hiroshi; LAZZAROTTO, Joelsio José. **O agronegócio da soja nos contextos mundial e brasileiro**. 2014.
- MARTOS, Vanesa et al. **Ensuring agricultural sustainability through remote sensing in the era of agriculture 5.0**. *Applied Sciences*, v. 11, n. 13, p. 5911, 2021.
- MAZOYER, Marcel; ROUDART, Laurence. **Histórias das agriculturas no mundo. Do neolítico à crise contemporânea**. Universidade Estatal Paulista (UNESP), 2010.
- NAIKWADE, Rohan R. et al. **Agriculture 5.0: Future of Smart Farming**. In: National Conference on Innovative Global Technology Trends in Art, Design, Technology, Management, Vedic Science, Education and Architecture, Film & Media. 2021. p. 1-6.
- RASCHKA, Sebastian. **Python Machine Learning: unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analytics**, 2015.
- SOOFI, A. A.; AWAN, A. **Classification Techniques in Machine Learning: Applications and Issues**. *Journal of Basic & Applied Sciences*, [s. l.], v. 13, p. 459–465, 2017.
- TETILA, Everton Castelhão. **Deteção e classificação de doenças e pragas da soja usando imagens de veículos aéreos não tripulados e técnicas de visão computacional**. 2019.
- USDA. United State Department of Agriculture, 2024. Disponível em: <<https://fas.usda.gov/data/production/commodity/2222000>>. Acesso em: 13 setembro 2024.

WICAKSONO, Rayhan Adi. SOYSCAN_DATASET, 2024. Disponível em: <<https://universe.roboflow.com/rayhan-adi-wicaksono-fcqu9/soyscan>>. Acesso em: 24 de Setembro de 2024.

ZHANG, Y. **New Advances in Machine Learning**. [S. l.]: BoD – Books on Demand, 2010.