

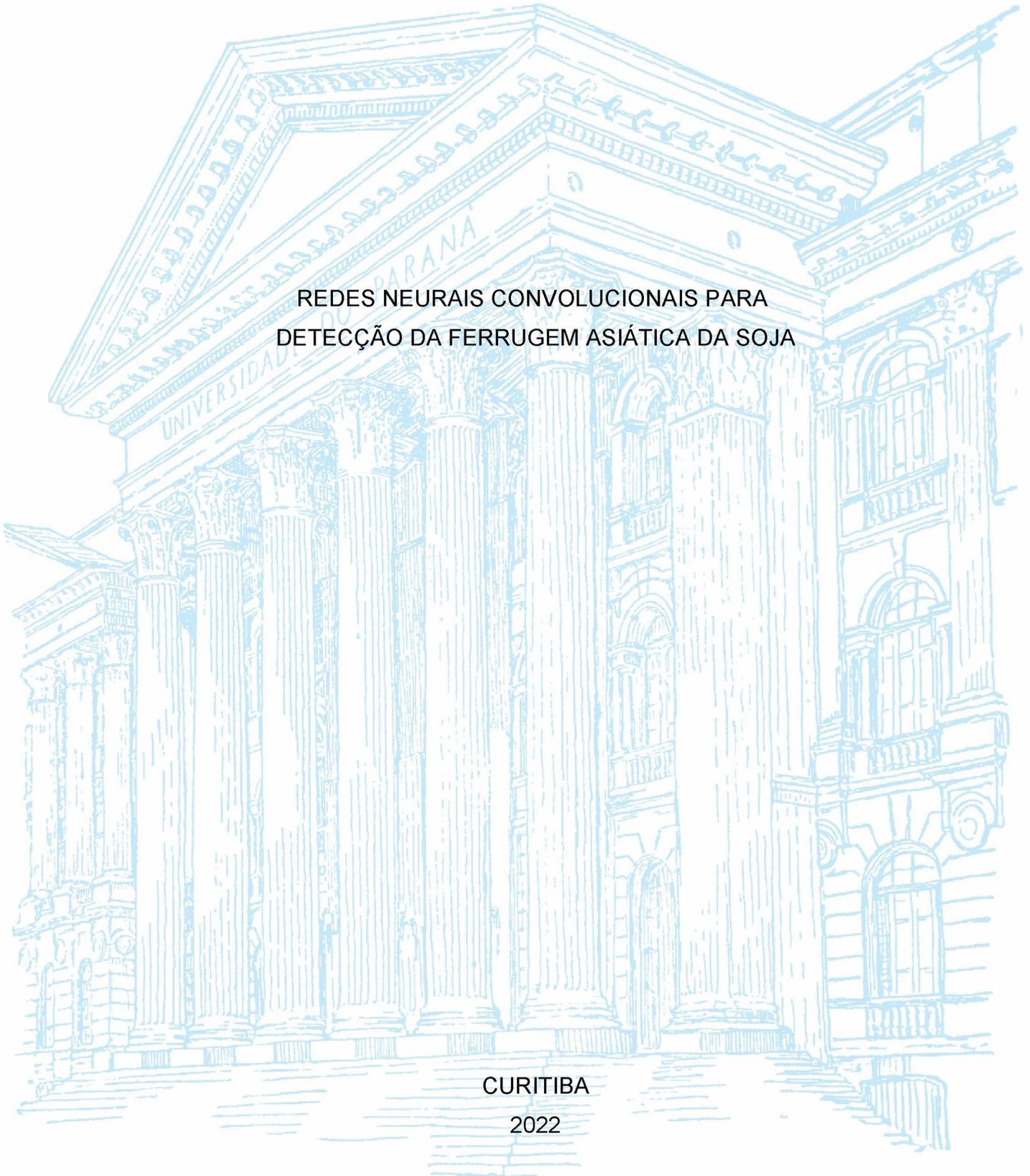
UNIVERSIDADE FEDERAL DO PARANÁ

MARCOS LUIZ CLITON BEZERRA

REDES NEURAIS CONVOLUCIONAIS PARA
DETECÇÃO DA FERRUGEM ASIÁTICA DA SOJA

CURITIBA

2022



Marcos Luiz Cliton Bezerra

REDES NEURAIAS CONVOLUCIONAIS PARA DETECÇÃO DA FERRUGEM
ASIÁTICA DA SOJA

Trabalho de Conclusão de Curso apresentado ao curso de Especialização em Inteligência Artificial Aplicada, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Especialista em Inteligência Artificial Aplicada.

Orientador: Prof. Dr. Razer Anthom Nizer Rojas Montaña

CURITIBA

2022

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INTELIGÊNCIA ARTIFICIAL APLICADA da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **MARCOS LUIZ CLITON BEZERRA** intitulada: **Redes Neurais Convolucionais para Detecção da Ferrugem Asiática da Soja**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa. A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 24 de Novembro de 2022.



RAZER ANTHOM NIZER ROJAS MONTAÑO
Presidente da Banca Examinadora



LUCAS FERRARI DE OLIVEIRA
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Redes Neurais Convolucionais para Detecção da Ferrugem Asiática da Soja

Marcos Luiz Cliton Bezerra
Setor de Educação Profissional e Tecnológica - SEPT
Universidade Federal do Paraná - UFPR
Curitiba, Brasil
mlcbezerra@hotmail.com

Prof. Dr^o Razer A. N. Rojas Montano
Setor de Educação Profissional e Tecnológica - SEPT
Universidade Federal do Paraná - UFPR
Curitiba, Brasil
razer@ufpr.br

Resumo—O Brasil é o maior produtor de soja do mundo com produção recorde de 135,4 milhões de toneladas em 2021, total de 8,5% ou 10,6 milhões de toneladas superior à safra 2019/2020. A ferrugem asiática da soja é causada pelo fungo *Phakopsora pachyrhizi* H. Sydow & P. Sydow, sendo a doença mais severa da cultura causando perdas de até 90%. As práticas de Manejo Integrado de Pragas (MIP) para o cultivo da soja são definidas com ações de controle, isoladas ou associadas e análises do custo e benefício dependendo do nível de integração do programa. Para um correto programa de controle de doenças em plantas, o primeiro passo é a correta identificação do agente causal. O objetivo deste trabalho é verificar a aplicabilidade da rede Mask R-CNN na identificação dos fungos utilizando uma base de imagens sintéticas. Os resultados finais demonstram condições favoráveis para o uso das redes neurais convolucionais na identificação de classes em imagens sintéticas. A precisão alcançada de 97,42647%, demonstra alta capacidade de evitar falsos positivos. O recall de 87,84530% especifica o acerto das amostras positivas. A média F ou F1 foi de 92,38815 podendo ser interpretada como uma medida de confiança da acurácia. A acurácia ficou com um desempenho de 88,13406% demonstrando o quão bom o modelo está prevendo corretamente.

Palavra-chave—*Phakopsora pachyrhizi*, Manejo Integrado de Pragas (MIP), soja, ferrugem da soja, ferrugem asiática, ferrugem asiática da soja.

Abstract—Brazil is the world's largest producer of soybeans with a record production of 135.4 million tons in 2021, a total of 8.5% or 10.6 million tons more than the 2019/2020 crop. Asian soybean rust is caused by the fungus *Phakopsora pachyrhizi* H. Sydow & P. Sydow, being the most severe disease of the crop causing losses of up to 90%. The practices of Integrated Pest Management (IPM) for soybean cultivation are defined with control actions, isolated or associated and analyzes the cost and benefit depending on the level of integration of the program. For a correct plant disease control program, the first step is the correct identification of the causal agent. The objective of this work is to verify the applicability of the Mask R-CNN network in the identification of the fungi using a database of synthetic images. The final results demonstrate favorable conditions for the use of convolutional neural networks in the identification of classes in synthetic images. The accuracy reached of 97.42647%, demonstrates a high capacity to avoid false positives. The 87.84530% recall specifies the hit of the positive samples. The F or F1 mean was 92.38815% and can be interpreted as a measure of confidence of the accuracy. The accuracy had a performance of 88.13406% demonstrating how well the model is predicting correctly.

Index Terms—*Phakopsora pachyrhizi*, Integrated Pest Management (IPM), soybean, soybean rust, Asian soybean rust, Asian soybean rust.

I. INTRODUÇÃO

Os agentes causais das doenças na cultura da soja são: fungos, oomicetos, bactérias, vírus e nematoides. A ferrugem asiática da soja é causada pelo fungo *Phakopsora pachyrhizi* H. Sydow & P. Sydow, é a doença mais grave na cultura. A ferrugem asiática da soja geralmente aparece como pequenas lesões circulares nas folhas das plantas de soja. Essas lesões são geralmente de cor vermelha ou marrom e podem ter uma aparência difusa ou pulverulenta. As bordas das lesões podem ser levemente elevadas. A ferrugem asiática da soja também pode fazer com que as folhas das plantas afetadas fiquem amarelas ou marrons e, eventualmente, caiam. É disseminada pelo vento e pela chuva e pode infectar plantas de soja em qualquer estágio de crescimento. Os estádios fenológicos de uma planta são as etapas de desenvolvimento na qual esta se encontra. São as fases evolutivas, como estando emergida (tem os cotilédones acima da superfície do solo e formando um ângulo de 90° ou maior), o crescimento de ramos primários ou o aparecimento de flores e frutos. A doença é mais prejudicial para as variedades de soja de maturação precoce. Os grupos de maturação estão dispostos de acordo com a latitude da localização geográfica cultivo e são classificados por uma numeração que varia de 0 (mais próximo dos pólos) a 10 (mais próximo do Equador, norte do Brasil) (Figura 1). As faixas de maturação classificam as áreas por grupos de acordo com a melhor adaptação por região. O ciclo de infecção e incubação na cultura está entre 6 e 9 dias, dependendo das condições climáticas (temperatura, molhamento folia e radiação solar) [13][27][32]. As variedades são classificadas numericamente da seguinte forma (Tabela D).

A ferrugem asiática da soja é uma doença fúngica grave e que pode causar altas perdas de produtividade. É difícil de controlar porque pode se espalhar rápida e facilmente pelo ar tornando-se uma séria ameaça à produção. O fungo também pode sobreviver em restos de plantas, dificultando sua erradicação. Pode causar perda significativa de rendimento em soja, sendo difícil de controlar uma vez que se estabeleça em

Tabela I
GRUPOS DE MATURAÇÃO

Faixas	Grupos de Maturação
abaixo de 6.0	super-precoces
entre 6.0 a 6.5	precoces
acima de 6.5 a 7.0	ciclo normal
acima de 7.0	tardias

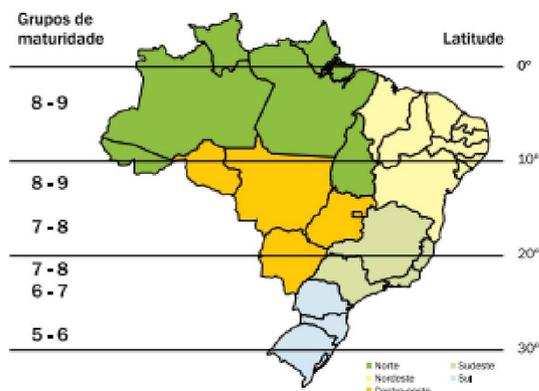


Figura 1. Distribuição dos grupos de maturidade relativa de cultivares de soja no Brasil, em função da latitude. Fonte: [1].

campo. A detecção precoce ao tratamento da ferrugem da soja são essenciais para evitar perdas de produtividade. Podendo ajudar os agricultores a ter ações que evitem a propagação da praga.

As principais ações realizadas para o controle da ferrugem asiática são o uso de fungicidas e medidas quanto ao plantio, sendo:

- Calendário de semeadura. Cultivar a soja precocemente, no início do período recomendado, é uma estratégia para evitar a doença;
- Não cultivar durante o vazio sanitário;
- Correto uso no controle químico;
- Inspeção constante para detecção de focos da ferrugem asiática;
- Organização no manejo da soja (apoio técnico, custos de plantio e manutenção, planejamento agrícola e inspeção no plantio).

O objetivo deste trabalho é desenvolver uma solução para identificação do patógeno *Phakopsora pachyrhizi* em campo utilizando visão computacional com o aplicação de redes neurais convolucionais. Permitindo uma tomada de decisão mais rápida e assertiva na detecção correta do fungo e no alerta correto da ferrugem asiática da soja. O menor tempo de resposta permitirá uma intervenção e tratamento precoce, reduzindo significativamente o impacto da doença e não permitindo o rápido avanço no plantio.

A hipótese de analisar as imagens das lâminas de microscopia com uma rede neural convolucional (*Convolution Neural Network* - CNN) ainda em campo torna possível a identificação, classificação de diferentes tipos de plantas e

pragas, a detecção de sinais precoces de doenças e possibilita ações que podem prevenir surtos generalizados. Esta ação será benéfica para o gerenciamento da cultura e análise de rendimento. Inspeccionar visualmente a lavoura é uma prática aplicada e eficaz de manter o controle da ferrugem asiática. A incidência de ferrugem asiática nas lavouras da mesma região é um risco direto a lavoura. Um dos fatores que viabiliza este estudo é o pouco tempo hábil para a identificação do fungo. Havendo a necessidade de uma confirmação em laboratório o avanço da doença impõe ao agricultor o controle químico com aplicações de fungicidas.

II. FUNDAMENTAÇÃO TEÓRICA

As melhores estratégias de manejo para o controle da ferrugem da soja no Brasil são o uso de cultivares de ciclo precoce e semeaduras no início da safra, o uso de cultivares com genes de resistência, o monitoramento da cultura desde o início de seu desenvolvimento, o uso de fungicidas preventivamente ou no início dos sintomas e a redução das janelas de semeadura para reduzir o número de aplicações de fungicidas ao longo da safra. O correto manejo da ferrugem asiática da soja requer a integração de medidas culturais, de plantas com resistência genética e uso preventivo de fungicidas [2].

A doença ocorre em todas as áreas produtoras, podendo aparecer em qualquer estágio de desenvolvimento da planta e ter perdas de produtividade entre 10% e 90% [14]. Em março de 2001 a ferrugem asiática da soja foi detectada pela primeira na região de Pitapó no Paraguai. Em maio deste mesmo ano já estava disseminada em grande parte do Paraguai e no oeste e norte do Paraná, Brasil. Em 2003 a doença foi diagnosticada em 90% da área de cultivo da soja em solo brasileiro [15][29].

As condições favoráveis para a ocorrência da doença são: temperatura entre 15°C e 25°C e molhamento mínimo de seis horas e sendo considerado o ideal de 12h [14]. O fungo inicia o desenvolvimento nas primeiras semeaduras e é disseminado facilmente pelo vento para áreas vizinhas a plantação. O plantio na época recomendada é uma estratégia de controle eficiente da doença. As datas limites para o plantio tem como objetivo a redução do número de aplicações de fungicidas ao longo da safra, reduzindo a seleção de resistência aos fungicidas.

A perda de eficiência dos fungicidas devido ao elevado número de aplicações fica evidente, com uma série de mutações sendo acumuladas no genoma do fungo [26][7][8]. Diante da situação de controle reduzido da ferrugem asiática fornecida por fungicidas, foi sugerido a implementação de diversas medidas, incluindo o plantio da soja safrinha [25]. Essa medida tem o objetivo de reduzir a janela de semeadura e evitar que fungos com resistência continuem recebendo novas aplicações de fungicidas, aumentando a seleção de resistentes. Devido ao menor período de precipitação nas condições climáticas menos favoráveis, o desenvolvimento da ferrugem asiática é menor do plantio tardio. O aumento do ponto de cultivo e o aparecimento da doença, requer o controle químico e aumenta a pressão de resistência aos fungicidas. O

escape é uma medida de controle conhecido como evasão. O objetivo é prevenir o patógeno realizando o plantio sem a presença da doença ou com condições climáticas favoráveis ao desenvolvimento da praga. Para o controle da doença, é realizado o plantio de cultivares precoces de soja precoces no início da época recomendada. Após as primeiras sementeiras realizadas ao vazio sanitário, os sintomas da doença tendem aparecer depois da fase de enchimento dos grãos.

A contaminação do fungo *Phakopsora pachyrhizi* ocorre diretamente na folha através da epiderme. Para a contaminação ocorrer tem que haver disponibilidade de água livre na superfície da folha, sendo necessário no mínimo 6 horas de molhamento. Para que ocorra a infecção, a temperatura deve estar na faixa entre 15°C e 25°C, considerada ótima e mais de 8 horas, para valores extremos, como 10°C ou 27°C [27]. A precipitação de chuva favorece o desenvolvimento das epidemias [28]. As altas temperaturas, acima de 27°, tem efeito negativo em experimentos com condições controladas [27]. Há relatos de epidemias severas no Brasil onde as temperaturas médias nesta faixa ocorrem na safra, com precipitação bem distribuída [30][28].

Para uma temperatura de 26°C o período latente, que é o tempo entre o início da infecção e a esporulação é de 6 dias. O período aumenta quando a temperatura varia para 15°C, em ambas as direções, podendo chegar a 12 ou 16 dias sob temperatura de 15°C [33]. A germinação dos esporos e a taxa de desenvolvimento são afetadas quando as temperaturas estão acima da faixa favorável [10].

Na face inferior da folha, os sintomas iniciais da presença do fungo são pequenos pontos entre 1 e 2 mm, com coloração entre esverdeado e cinza esverdeado. As lesões são a formação de protuberâncias que são as urédias, estruturas reprodutivas do fungo. Com o avanço da doença, na parte superior da planta forma-se uma cavidade por onde são expelidos os esporos denominados uredósporos. Estes são produzidos nos urédios, durante a fase propícia ao desenvolvimento da doença. Os urédios são formados na face inferior da folha, tem distribuição irregular e possuem 1 mm de diâmetro. Estes possuem uma coloração inicial hialina mudando com o passar do tempo para o bege, acumulando-se em volta dos poros ou sendo transportados pelo vento. As plantas infectadas apresentam queda foliar precoce, comprometendo a formação total da planta (Figura 2).



Figura 2. Primeiros sintomas caracterizados por minúsculos pontos identificando os sintomas da infecção. Fonte: [12].

O fungo da ferrugem asiática da soja é facilmente disseminado pelo vento. O monitoramento com um coletor de esporos

é uma ferramenta importante para a detecção precoce da doença e para a tomada de decisões adequadas sobre o uso de fungicidas. O desenvolvimento e uso deste equipamento data desde a metade do século 19 [17][31]. Em 1985 o pesquisador engenheiro agrônomo Seiji Igarashi do antigo IAPAR-Emater (atualmente IDR-Paraná), utilizou o coletor de esporos para a detecção de fungos que ocorrem nas espigas do trigo. Este equipamento faz a coleta de esporos que estão sendo transportados pelo vento, permitindo a detecção antecipada da doença (Figura 3).



Figura 3. Coletor de Esporos. Fonte: [18].

A construção do equipamento é feita com tubos de PVC que são apoiados em uma haste metálica que é fixada no solo. O sistema possui um rolamento na haste permitindo que este gire livremente. O vento passa pelo interior do tubo, onde há duas lâminas de microscopia com fita dupla face, aderindo os esporos que forem capturados [18].

Os esporos que estão sendo carregados pelo vento e passam pelo interior do coletor, são captados por uma fita adesiva de dupla face que está fixada em uma lâmina de microscopia na parte interna do tupo principal. As lâminas são etiquetadas com informações ao monitoramento, sendo:

- identificação do coletor;
- data de instalação / troca da lâmina;
- estágio fenológico do plantio.

As lâminas são trocadas semanalmente e enviadas para análise laboratorial (Figura 4).

O coletor de esporos deve ser instalado após a emergência das plantas, podendo ser mantido durante todo o ciclo da cultura. Se houver a detecção da presença do fungo, deverá ser retirado de campo. A inspeção da lâmina deve ser realizada pelo menos uma vez por semana. Durante o estágio do plantio reprodutivo ou fechamento das entrelinhas, deve se aumentar a inspeção da lâmina para duas vezes por semana.

É utilizado um microscópio óptico para fazer a leitura da lâmina, para identificar a presença de uredósporos é feita uma comparação com uma lâmina montada com es-



Figura 4. Lâmina de microscopia com fita adesiva dupla face e etiqueta de identificação (A). Visão interna do coletor de esporos (B). Fonte: [18].

poros dos sintomas relacionados. São considerados aspectos relacionados ao comprimento, a largura, a relação comprimento/largura e a coloração se mostraram parâmetros importantes na identificação de ferrugens [19].

O principal fator para que o uso do coletor de esporos fosse tão difundido foi a condição de detecção da presença de esporos sem ter ocorrido a manifestação da doença na plantação. A necessidade humana para a troca das lâminas e da análise laboratorial, faz com que o processo seja ineficiente pelo tempo do ciclo de infecção e incubação de no máximo 9 dias. Foi realizado um estudo com diferentes ampliações no reconhecimento de esporos da ferrugem asiática da soja. Foi descoberto que estes podem ser analisados com precisão em ampliações de 100x, 400x e 1000x com e sem lamínula [19] (Figura 5).

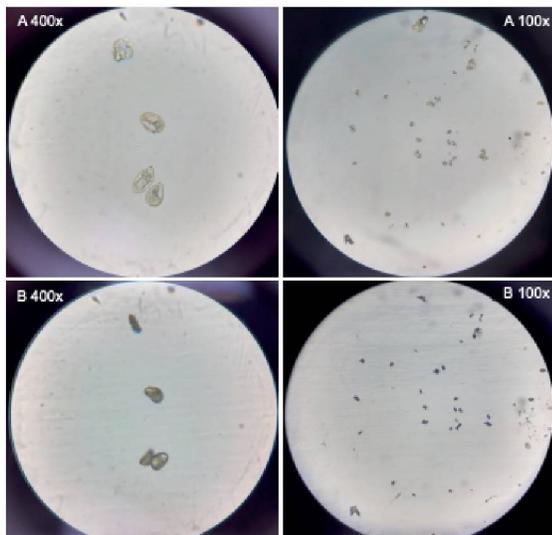


Figura 5. Esporos da *Phakopsora pachyrhizi* em microscópio óptico com diferentes aumentos. Fonte: [18].

No Brasil as principais ações de controle da ferrugem asiática da soja é o vazio sanitário (período no qual tem a ausência total de plantas vivas de soja, prevenindo a proliferação da doença), o uso de sementes com ciclo precoce e genes resistentes, semeaduras no início da época, monitoramento da lavoura, redução das janelas de semeaduras e aplicações corretas de fungicidas [16] (Figura 6). O uso do vazio sanitário permite um atraso da incidência do fungo

na lavoura. A aplicação correta nos meses de novembro, dezembro ou janeiro em alguns estados, possibilita um alto controle da doença no início da semeadura [2].

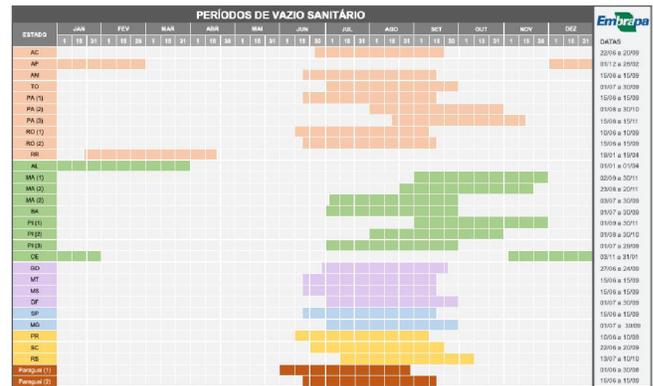


Figura 6. Vazio Sanitário. Fonte: [2].

O Manejo Integrado de Pragas (MIP) é um conjunto de técnicas econômicas e de sustentabilidade ambiental. Os benefícios são a redução dos custos de produção, a diminuição do impacto ambiental, redução das perdas de produção e redução do desenvolvimento da praga agrícola. O processo consiste de inspeções frequentes verificando o número e tamanho dos insetos-pragas que estão presentes na lavoura e os danos já causados. As principais práticas do MIP adotadas na agricultura são definidas com o controle de ações isoladas ou não, analisando o custo e benefícios de inserção ao programa [11]. O objetivo da prática é de reduzir a população de pragas abaixo do nível de controle, que é a menor densidade populacional causadora de danos a lavoura.

Os três principais grupos de sítio-específicos fungicidas utilizados para o controle da doença que possui resistência e menor sensibilidade ao fungo são [6][7][8]:

- inibidores da desmetilação (IDM - triazóis)
- inibidores da quinona externa (IQe - estrobilurinas)
- inibidores da succinato desidrogenase (ISDH- carboxamidas).

O uso de fungicidas para controle da doença deve ocorrer no início dos primeiros sintomas ou de forma preventiva. Para o controle preventivo são considerados as condições de aparecimento do fungo. É importante fazer a rotação de fungicidas e evitar aplicações sequenciais do mesmo produto para reduzir a pressão de seleção de resistência do fungo. A rotação é importante no plantio porque ajuda a prevenir o acúmulo de cepas de fungos resistentes. Ao alternar entre diferentes fungicidas, você pode garantir que os fungos não se tornem resistentes a nenhum fungicida específico. Isso ajudará a manter suas plantas de soja saudáveis e livres de doenças.

Bio inseticidas são um tipo de pesticida que usa organismos vivos para matar pragas. Esses pesticidas podem ser feitos de bactérias, fungos, vírus ou outros materiais naturais. Eles são frequentemente usados como uma alternativa aos pesticidas químicos, pois são considerados mais ecológicos. Os bioinseticidas podem ser eficazes contra uma ampla variedade

de pragas, incluindo insetos, aranhas, ácaros e nematóides. Existem várias razões para usar bioinseticidas:

- Eles costumam ser mais eficazes do que os pesticidas químicos no controle de pragas e também são mais específicos, o que significa que têm menos probabilidade de prejudicar insetos benéficos;
- Geralmente são mais baratos que os pesticidas químicos;
- Eles são mais ecológicos, pois são feitos de ingredientes naturais e se decompõem mais rapidamente no meio ambiente.

Como o número de uredósporos aumenta nas sementeiras tardias, devido à multiplicação nas primeiras sementeiras, a definição de prazos de sementeira é importante para evitar excesso de aplicações em uma safra, reduzindo a pressão de seleção de fungos resistentes a fungicidas. A resistência aos fungicidas é natural e tendem a eliminar as populações mais sensíveis aumentando as populações menos sensíveis. A resistência pode ser cruzada a fungicidas, fungos resistentes a um determinado tipo pode ser resistente a outro com o mesmo modo de ação, mesmo que não tenha sido ainda exposto [5].

A rápida disseminação da doença e a dificuldade de identificação correta dos primeiros sintomas, faz com que o monitoramento com um coletor de esporos seja utilizado na detecção preventiva do fungo *Phakopsora pachyrhizi* antes da infecção na lavoura.

A. Neurônio McCulloch-Pitts (MCP)

O neurônio McCulloch-Pitts (MCP) é um modelo matemático simples de neurônio biológico que foi proposto pela primeira vez pelo neurofisiologista Walter S. McCulloch e pelo lógico Walter Pitts em 1943 [23]. Os neurônios são as células-chave que compõem o cérebro. Eles são eletricamente ativos e se comunicam com outras células por meio de conexões especiais. Vários modelos biológicos são usados para descrever suas propriedades e comportamentos. O neurônio biológico recebe uma entrada de sinal dos seus dendritos (ou seja, de outros neurônios), que é então passado para a colina axônica. Se a entrada for maior que o limite, o neurônio é acionado; caso contrário, está em repouso. O estado do nosso neurônio (ligado ou desligado) então se propaga através de seu axônio e é passado para outros neurônios conectados através de suas sinapses. A rede neural foi proposta pela primeira vez por Walter Pitts em 1943 como um modelo matemático de como os neurônios no cérebro podem funcionar. Apesar de muito simples, seu modelo se mostrou extremamente versátil e fácil de modificar. Hoje, variações de seu modelo original tornaram-se os blocos de construção elementares da maioria das redes neurais (Figura 7).

$$f(x) = \begin{cases} 1, & \text{if } v(i) \geq 0 \\ 0, & \text{if } v(i) < 0 \end{cases} \quad (1)$$

B. Perceptron

O algoritmo Perceptron foi desenvolvido na década de 1950 por Frank Rosenblatt e é considerado um dos primeiros

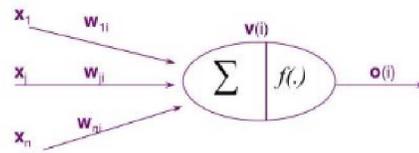


Figura 7. Modelo Neurônio MCP. Fonte: [23].

algoritmos de aprendizado de máquina [21]. Em 1957, Rosenblatt propôs um modelo de aprendizado automático baseado em um único neurônio artificial com um conjunto de pesos e uma função de ativação. É usado para aprender funções de classificação que podem ser usadas para separar dados em dois conjuntos linearmente separáveis. Se os dados não forem separáveis linearmente, o algoritmo não convergirá para uma solução e os dados não serão classificados corretamente. No entanto, se os dados forem linearmente separáveis, o algoritmo Perceptron pode encontrar um conjunto de pesos que classificará corretamente os dados.

O algoritmo Perceptron é um algoritmo de aprendizado supervisionado, o que significa que requer um conjunto de dados de treinamento rotulados para aprender a função de classificação. O algoritmo é treinado usando o método gradiente descendente para minimizar um erro quadrático médio. Uma vez que o algoritmo é treinado, ele pode ser usado para classificar novos dados usando a função de classificação aprendida. O perceptron é uma rede neural muito simples, mas ainda assim poderosa. Ele foi usado para reconhecimento de padrões, aprendizado de máquina e até mesmo processamento de linguagem natural. É um algoritmo de aprendizagem baseado em um modelo de rede neural artificial (RNA) de uma única camada, com um limiar de ativação. Seus limites são determinados pelo número de neurônio na camada de saída, que trabalham ponderando os dados de entrada e alimentando-os por meio de uma função de limite para produzir uma saída. Este modelo matemático permite que de uma forma simples possam compreender o funcionamento das redes neurais. O modelo proposto recebe várias entradas e produz somente uma saída binária. A regra para o cálculo é a atribuição de pesos (w_1, w_2, w_3, \dots) que são números reais que expressam a importância dos valores de entrada (x_1, x_2, x_3, \dots) em relação a saída (0 ou 1) (Figura 8).

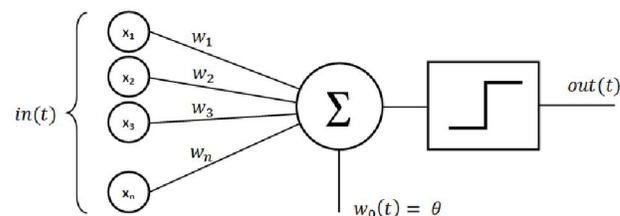


Figura 8. Rede neural perceptron. Fonte: [21].

O neurônio proposto por Rosenblatt é semelhante ao McCulloch & Pitts de 1943, mas possui algumas diferenças:

- O modelo de neurônio proposto por Rosenblatt é um modelo linear enquanto o modelo de neurônio proposto por McCulloch & Pitts é um modelo não linear.
- É baseado na ideia de um perceptron, que é uma rede neural de camada única. O modelo de neurônio proposto por McCulloch & Pitts é baseado na ideia de uma unidade lógica de limiar (TLU), que é um tipo de neurônio artificial.
- Pode ser usado para tarefas de aprendizado supervisionado, como classificação. O modelo de neurônio proposto por McCulloch & Pitts pode ser usado tanto para tarefas de aprendizado supervisionadas quanto não supervisionadas.

A saída do neurônio é determinada pela soma ponderada, menor ou maior do que algum valor limiar (*threshold*). Os pesos e o *threshold* são parâmetros do neurônio. O algoritmo perceptron é capaz de aprender a função de limiar usando apenas o *feedback* binário para as suas respostas. O algoritmo funciona atualizando os pesos dos atributos de entrada na função de limiar para maximizar as respostas corretas. Nas redes neurais *feedforward* o sinal de entrada propaga somente em uma única direção, não retornando para o sinal para a entrada. A função de ativação é definida como:

$$f(x) = \begin{cases} 1, & \text{if } w * x + b > 0 \\ 0, & \text{caso contrário} \end{cases} \quad (2)$$

A regra pode ser mudada trocando o *threshold* pelo viés ou também chamado de bias. A mudança é alterar a posição do *threshold* na equação. O viés é uma medida que proporciona com que a rede tenha o valor 1 na saída. O treinamento é para que o modelo aprenda os valores ideais para os pesos e viés.

$$f(x) = \begin{cases} 0, & \text{if } w * x + b \leq 0 \\ 1, & \text{if } w * x + b > 0 \end{cases} \quad (3)$$

Normalmente as redes neurais artificiais são compostas pela camada de entrada, a camada oculta e a camada de saída. A rede *feedforward* pode possuir múltiplas camadas, sendo que a saída de uma camada é a entrada da outra (Figura 9).

C. Backpropagation

As redes neurais artificiais são compostas por um grande número de nós de processamento interconectado. Cada nó em uma RNA está conectado a vários outros nós e tem um peso associado. Os pesos das conexões são ajustados durante o processo de aprendizado para que a saída da RNA fique próxima da saída desejada. Para este ajuste é utilizada uma função de erro, que calcula o quanto a rede está imprecisa.

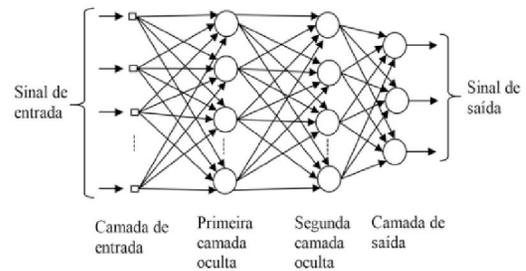


Figura 9. Exemplo rede *feedforward*. Fonte: [3].

O *backpropagation* é um método usado para treinar redes neurais. É uma forma de gradiente descendente, onde o gradiente é calculado usando o algoritmo de retropropagação. O algoritmo é usado para calcular o erro em cada camada da rede e, em seguida, atualizar os pesos. É o processo de treinamento da rede neural que ajusta os pesos das conexões entre seus neurônios. Primeiro é calculado o gradiente de erro para cada neurônio e, em seguida, os pesos são ajustados na direção oposta do gradiente. O gradiente de erro para um neurônio é calculado multiplicando o erro do neurônio com a saída do neurônio. Este valor é a quantidade que a saída do neurônio precisa ser alterada para reduzir o erro. Os pesos são ajustados adicionando uma fração do gradiente de erro ao peso atual. O tamanho da fração é determinado pela taxa de aprendizado. *Backpropagation* é um método eficiente de treinar uma rede neural e pode ser usado tanto para aprendizado supervisionado quanto não supervisionado. É um tipo de aprendizado supervisionado, onde o erro é propagado de volta pela rede para atualizar os pesos. Existem algumas razões para usar a retropropagação.

- É um algoritmo relativamente simples que pode ser usado para treinar uma rede neural;
- A retropropagação é muito eficaz no treinamento de redes neurais para realizar tarefas complexas;
- A retropropagação é frequentemente usada em conjunto com outros algoritmos de treinamento de rede neural, como gradiente descendente, para melhorar ainda mais o desempenho da rede neural;
- A retropropagação é um método amplamente utilizado de treinamento de redes neurais devido à sua eficácia.

D. Gradiente descendente

O gradiente descendente é um algoritmo de otimização usado para encontrar os valores dos parâmetros (coeficientes) de uma função (f) que minimiza uma função de custo. A função de custo é uma medida de quão longe a hipótese atual (h) está do valor real (y). A função de custo também é conhecida como função de erro ou função objetivo. O algoritmo funciona movendo-se iterativamente na direção da descida mais íngreme (a direção do gradiente) até atingir um mínimo local. O tamanho do passo é determinado pela taxa de aprendizado. Uma desvantagem do algoritmo de descida

de gradiente padrão é que ele requer muitas iterações para convergir para o mínimo global. Uma maneira de superar essa desvantagem é usar uma técnica chamada momentum. É um método que ajuda a acelerar a descida do gradiente levando em consideração os gradientes anteriores para suavizar a atualização. O Momentum é uma técnica de descida do gradiente que combina a atualização de parâmetros com a atualização de velocidade. É a média ponderada entre a atualização de parâmetros atual e a atualização de parâmetros anterior. Essa técnica aproveita o momento adicionando uma fração do gradiente anterior ao gradiente atual, aumentando a velocidade de aprendizado e potencialmente reduzindo a quantidade de tempo para encontrar a solução.

E. Função de custo

Uma função de custo de rede neural é uma função matemática que determina a saída ideal de uma rede neural dado um conjunto de entradas. Determinando a melhor forma de ajustar os pesos e vieses da rede de forma a minimizar a função de erro. A função de custo é usada para minimizar o erro entre a saída real da rede neural e a saída desejada. A função de erro é uma medida de quão bem a rede neural é capaz de aprender com os dados de treinamento. A função de custo é usada para otimizar os pesos e vieses da rede neural para que a função de erro seja minimizada.

Existem algumas maneiras de ajustar os pesos e vieses da função de custo de uma rede neural. Uma maneira é usar uma técnica chamada *Backpropagation*. É um método de treinamento de redes neurais que envolve o ajuste dos pesos e bias da rede para que o erro entre a saída prevista e a saída real seja minimizado. Outra maneira de ajustar os pesos e vieses da função de custo é usar uma técnica chamada gradiente descendente. A descida de gradiente é um método de treinamento de redes neurais que envolve o ajuste dos pesos e desvios da rede para que o erro entre a saída prevista e a saída real seja minimizado.

F. Taxa de aprendizagem

A taxa de aprendizado de uma rede neural é a velocidade com que a rede aprende com os dados de treinamento. Uma taxa de aprendizado mais alta significa que a rede aprende mais rápido, enquanto uma taxa de aprendizado mais baixa significa que a rede aprende mais devagar. A taxa de aprendizado ideal para uma determinada rede neural depende da estrutura da rede e do tipo de dados que está sendo usado para treinamento. É um hiperparâmetro que controla o quanto os pesos da rede são atualizados em resposta à função de perda. Uma taxa de aprendizado mais alta significa que a rede aprende mais rápido, mas também pode ser mais provável que os dados sejam superajustados. Uma taxa mais baixa significa que a rede aprende mais lentamente, mas pode ser mais provável que generalize para novos dados.

G. Épocas de treinamento

Uma época é uma única passagem por todo o conjunto de dados de treinamento. Durante cada época, o modelo itera sobre o conjunto de dados de treinamento e atualiza os pesos. O treinamento é realizado a cada volta ou época, minimizando a taxa de erro.

H. Overfitting e Underfitting

No aprendizado da rede pode ocorrer o *overfitting* que é quando um modelo captura muitos detalhes dos dados de treinamento, a ponto de começar a aprender com o ruído e não com o sinal. Isso pode levar a um baixo desempenho de generalização em novos dados. Um modelo superajustado é um modelo estatístico que foi excessivamente otimizado para se ajustar a um determinado conjunto de dados. O *overfitting* pode levar a um desempenho de generalização ruim, o que significa que o modelo não funciona bem em dados não vistos. Geralmente é o resultado do uso de muitos dados de treinamento ou de um modelo complexo. Pode ser evitado usando um modelo mais simples ou usando menos dados de treinamento.

O *underfitting* ocorre quando um modelo é muito simples e não captura detalhes suficientes dos dados de treinamento. Isso pode levar a um desempenho ruim nos dados de treinamento e nos novos dados. Ocorre quando um modelo não consegue capturar com precisão as tendências subjacentes nos dados. Também pode ocorrer se os dados não forem pré-processados adequadamente (por exemplo, se recursos importantes não forem incluídos). As consequências do *underfitting* são que o modelo terá um desempenho preditivo ruim em novos dados. No aprendizado de máquina, isso geralmente se reflete em uma pontuação de baixa precisão.

Existem algumas maneiras de combater o *underfitting*. Uma delas é usar um modelo mais complexo. Isso geralmente envolve adicionar mais recursos ou aumentar a flexibilidade do modelo (por exemplo, usando um modelo não linear em vez de um modelo linear). Outra maneira de combater o *underfitting* é pré-processar os dados com mais cuidado. Isso pode envolver adicionar recursos, remover recursos ou transformar os dados de alguma forma. Finalmente, às vezes é possível usar técnicas de regularização para reduzir o *overfitting* de um modelo.

Uma maneira de corrigir o *overfitting* é usar a validação cruzada. Esta é uma técnica que ajuda a avaliar quão bem um modelo será generalizado para novos dados. Outra maneira de corrigir o *overfitting* é usar a regularização. Esta é uma técnica que ajuda a penalizar modelos complexos e encorajar modelos mais simples. Finalmente, outra maneira de corrigir o *overfitting* é usar o aumento de dados. Esta é uma técnica que ajuda a criar novos pontos de dados a partir de pontos de dados existentes.

I. Função de ativação

A função de ativação é uma função matemática que é usada para determinar a saída de uma rede neural. A função

é usada para mapear os valores de entrada (x) para os valores de saída (y). Há uma variedade de funções de ativação que podem ser usadas, e a escolha da função depende do tipo de rede neural que está sendo usada. A função de ativação mais comum é a função sigmóide, que é usada na regressão logística. A função sigmóide recebe um valor de entrada (x) e produz um valor entre 0 e 1. Essa saída pode ser interpretada como a probabilidade de a entrada pertencer a uma determinada classe. Outras funções de ativação que são comumente usadas incluem a função tanh, que emite valores entre -1 e 1, e a função ReLU, que emite valores entre 0 e 1. A escolha da função de ativação pode ter um impacto significativo no desempenho de um sistema neural. É importante experimentar diferentes funções de ativação para encontrar aquela que funciona melhor para seus dados e sua rede neural. Alguns tipos de funções utilizadas:

1) *Função Sigmóide*: Também chamada de função logística, é uma função matemática que pode ser usada para modelar uma ampla variedade de situações (Figura 10).

A função sigmóide ou logística e sua derivada, respectivamente:

$$\text{sigmóide}(x) = \frac{1}{1 + e^{-x}} \quad (4)$$

$$\sigma'(x) = \sigma(x)(1 - \sigma(x)) \quad (5)$$

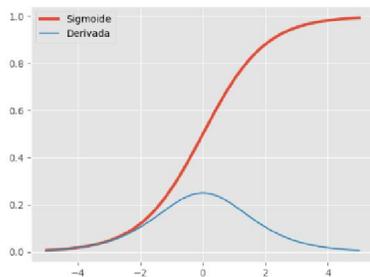


Figura 10. Gráfico da função sigmóide. Fonte: [4].

A função sigmóide tem várias propriedades úteis. Primeiro, é uma função suave que é fácil de trabalhar matematicamente. Em segundo lugar, é limitado entre 0 e 1, o que o torna útil para modelar probabilidades. Finalmente, a função é simétrica em torno de 0, o que significa que valores negativos e positivos de x produzirão valores que são imagens espelhadas um do outro. A função sigmóide é usada em uma variedade de aplicações, incluindo:

- Redes neurais: a função sigmóide é frequentemente usada como função de ativação em redes neurais artificiais;
- Regressão logística: a função sigmóide pode ser usada para modelar a probabilidade de um determinado evento ocorrer;
- Compactação de dados: a função sigmoid pode ser usada para compactar dados mapeando-os em um intervalo limitado de valores;

- Processamento de imagem: A função sigmóide pode ser usada para processar imagens mapeando intensidades de pixel em uma faixa limitada de valores.

2) *Função ReLU*: A ReLU (unidade linear retificada) é uma função de ativação popular para redes neurais. É uma função não linear que pode ser usada para modelar relacionamentos complexos entre variáveis de entrada e saída. ReLU é fácil de calcular e tem um gradiente que é fácil de calcular. O ReLU também é robusto ao ruído e pode lidar com uma grande variedade de valores de entrada. Uma desvantagem da ativação ReLU é que as unidades passam a resultar somente valores zeros. Isso acontece quando a soma ponderada antes de aplicar a função ReLU se torna negativa, fazendo com que a unidade produza uma saída igual a zero. Nesta região, a derivada também é zero, o que significa que os pesos da unidade não serão mais atualizados com a descida do gradiente (Figura 11).

A função linear retificada (ReLU) e sua derivada, respectivamente:

$$\text{ReLU}(x) = \max(0, x) \quad (6)$$

$$\text{ReLU}'(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{caso contrário} \end{cases} \quad (7)$$

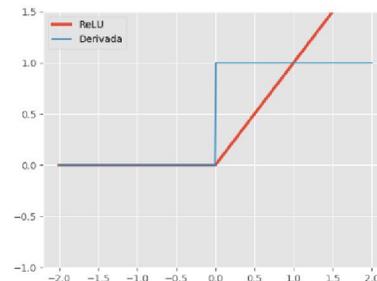


Figura 11. Gráfico da função ReLU. Fonte: [4].

3) *Função Leaky ReLU*: A solução para o problema de zerar a função foi proposta por [22], foi aplicada uma inclinação para parte negativa do seu domínio (Figura 12).

A função Leaky ReLU e sua derivada, respectivamente:

$$\text{LeakyReLU}(x, \sigma) = \max(\sigma x, x) \quad (8)$$

$$\text{LeakyReLU}'(x, \sigma) = \begin{cases} 1, & \text{if } x \geq 0 \\ \sigma, & \text{caso contrário} \end{cases} \quad (9)$$

4) *Função TanH*: A função TanH é usada para ativar neurônios em uma rede neural. Essa função também é conhecida como função tangente hiperbólica. TanH retorna um valor entre -1 e 1. Esta função é usada para esmagar a saída de um neurônio para que fique entre esses dois valores. Esta função é usada em muitas redes neurais porque é diferenciável. Isso significa que o erro pode ser retropropagado pela rede para atualizar os pesos (Figura 13).

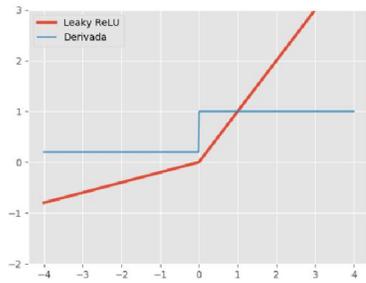


Figura 12. Gráfico da função Leaky ReLU. Fonte: [4].

A função TanH e sua derivada, respectivamente:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (10)$$

$$\tanh'(x) = 1 - \tanh^2(x) \quad (11)$$

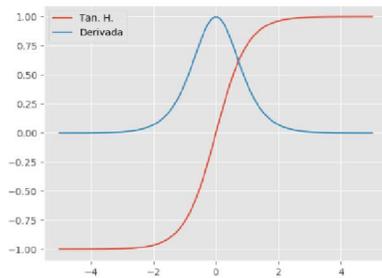


Figura 13. Gráfico da função TanH. Fonte: [4].

J. Função Convolação

A convolução de ativação é um processo que ajuda a melhorar o desempenho de uma rede neural. É uma técnica usada para criar modelos mais eficientes, reduzindo o número de parâmetros que precisam ser aprendidos. Esse processo pode ser aplicado a qualquer tipo de rede neural, incluindo redes neurais convolucionais (CNNs). A convolução é o processo de aplicar um filtro a uma entrada para produzir uma saída. Em uma rede neural convolucional, os filtros são aprendidos automaticamente a partir de um conjunto de dados de treinamento. A convolução pode acontecer diversas vezes, resultando em um mapa de ativações.

As formas de realizar a convolução de ativação, envolvem o uso de uma camada convolucional para aplicar uma função não linear aos dados de entrada. A mais comum é usando uma unidade linear retificada (ReLU), que simplesmente substitui todos os valores negativos na entrada por zeros. Outras funções não lineares também podem ser usadas, como a função sigmóide ou a função tanh.

K. Transfer Learning

O uso do aprendizado de transferência (*Transfer Learning*) é o processo de usar um modelo de rede neural pré treinado em um novo conjunto de dados. Isso pode ser feito ajustando os pesos do modelo no novo conjunto de dados ou usando o modelo como um extrator de recursos e treinando um novo classificador nos recursos extraídos. É frequentemente usado quando o conjunto de dados para a nova tarefa é pequeno ou quando a nova tarefa é semelhante à tarefa em que o modelo foi originalmente treinado. Isso ocorre porque geralmente é mais fácil e rápido treinar um novo modelo em um pequeno conjunto de dados quando o modelo pode começar com os pesos de um modelo pré-treinado. Características do aprendizado de transferência:

- O modelo pré-treinado deve ser adequado para o novo conjunto de dados. Se o modelo não foi treinado em um conjunto de dados semelhante, é provável que o modelo não funcione bem no novo conjunto de dados.
- Sendo o novo conjunto de dados muito diferente do conjunto original, é melhor não ajustar os pesos do modelo pré-treinado. Isso ocorre porque o modelo provavelmente se ajustará ao novo conjunto de dados se os pesos forem muito ajustados.

L. Rede Neural Convolutiva (CNN)

Uma rede neural convolutiva (Convolutional Neural Networks - CNN) é um tipo de rede neural artificial usada no reconhecimento e classificação de imagens [9]. Estas são semelhantes às redes neurais tradicionais, mas possuem uma camada adicional de camadas convolucionais que ajudam a aprender recursos nas imagens. Podem ser usadas para tarefas de aprendizado supervisionadas e não supervisionadas.

Uma CNN é uma rede neural que foi projetada para trabalhar com dados bidimensionais, como imagens. São compostas por uma série de camadas, cada uma das quais executa uma operação de convolução nos dados para extrair recursos. A última camada desta rede é tipicamente uma camada totalmente conectada, que é usada para fazer previsões sobre os dados. São muito eficazes na extração de recursos de imagens e têm sido usadas para uma variedade de tarefas, como classificação de imagens, detecção de objetos e segmentação de imagens.

Existem várias razões pelas quais alguém pode optar por usar uma CNN em uma rede neural tradicional. As CNNs são melhores em lidar com dados organizados espacialmente, como imagens. Isso ocorre porque a camada convolucional é capaz de extrair características da imagem de entrada. As CNN usam muitas das mesmas ideias de outras redes neurais, como retropropagação, gradiente descendente, regularização e funções de ativação não linear. As redes neurais convolucionais (ConvNets ou CNNs) são um tipo de rede neural que se mostrou muito eficaz em áreas como reconhecimento e classificação de imagens. As CNNs são semelhantes a outros tipos de redes neurais, mas são compostas por uma série de

camadas, cada uma delas realizando uma tarefa específica, organizadas em uma ordem específica (Figura 14).

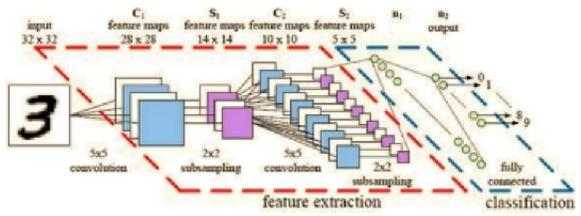


Figura 14. Arquitetura de rede convolucional. Fonte: [20].

Na rede neural convolucional, uma das características mais importantes e inovadoras é a capacidade de aprender automaticamente um grande número de filtros, em paralelo a um conjunto de dados de treinamento para uma funcionalidade específica, como a classificação de imagens, por exemplo.

Os pesos e vieses de uma rede determinam a força das conexões entre os neurônios na rede e, como tal, podem ter um impacto significativo na precisão da rede. No entanto, é importante notar que muito ajuste pode levar ao overfitting, o que pode diminuir a precisão da rede. Como tal, é importante encontrar um equilíbrio ao ajustar os pesos e vieses de uma rede neural convolucional.

Uma camada de pooling é uma camada em uma rede neural que reduz a dimensionalidade da entrada executando uma operação de resumo. O max pooling é uma operação comum em redes neurais convolucionais, normalmente executada após uma camada convolucional para reduzir a dimensionalidade dos dados. O objetivo é reduzir a amostragem dos dados, normalmente obtendo o valor máximo em cada região de pool. O max pooling tem a vantagem de poder reduzir a amostragem de dados sem perder informações, além de ser simples de implementar. No entanto, este também pode levar ao overfitting, pois pode remover completamente as informações de uma região se todos os valores nessa região forem os mesmos (Figura 15).

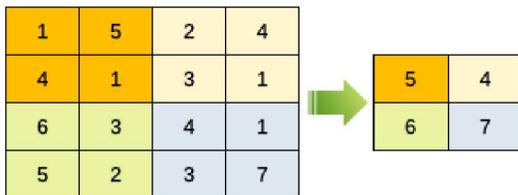


Figura 15. Funcionamento max pooling com filtro 2x2 e uma imagem 4x4. Fonte: O Autor.

Em uma camada totalmente conectada de uma rede neural convolucional, cada neurônio está conectado a todos os outros neurônios da próxima camada. Isso permite um fluxo mais direto de informações entre as camadas e permite que a rede aprenda padrões complexos. Camadas totalmente conectadas são normalmente usadas nas camadas finais de uma rede neural convolucional, onde os padrões aprendidos podem ser

usados para fazer previsões. Devido ao aumento do número de conexões, camadas totalmente conectadas podem ser mais difíceis de treinar do que outros tipos de camadas. No entanto, eles podem aprender padrões complexos que outros tipos de camadas podem não ser capazes de aprender.

O overfitting é um fenômeno que ocorre quando um modelo de aprendizado de máquina se ajusta muito aos dados de treinamento. Isso pode levar a um desempenho de generalização ruim em dados não vistos. Isso pode ser causado por umavariabilidade de fatores, incluindo o uso de muitos recursos, poucos exemplos ou muita flexibilidade no modelo.

O hiperparâmetro é um parâmetro que determina como o algoritmo de aprendizado funciona. Os mais comuns são a taxa de aprendizado e o parâmetro de regularização. A taxa de aprendizado determina a rapidez com que o algoritmo de aprendizado aprende. O parâmetro de regularização controla a quantidade de overfitting que é permitida. Outros hiperparâmetros comuns incluem o número de unidades ocultas em uma rede neural, o número de clusters em um algoritmo k-means e a largura do kernel gaussiano em uma máquina de vetor de suporte.

M. Matriz de confusão

Uma matriz de confusão é uma tabela usada para avaliar o desempenho de um algoritmo de aprendizado de máquina. É usada para comparar os valores previstos com os valores reais. A matriz de confusão representa o resultado de desempenho do modelo preditivo. Possui quatro quadrantes, sendo: o primeiro quadrante é o verdadeiro positivo, o segundo quadrante é o falso positivo, o terceiro quadrante é o falso negativo e o quarto quadrante é o verdadeiro negativo (Tabela II).

Tabela II
ESTRUTURA DA MATRIZ DE CONFUSÃO

Classe Predita	Classe Real	
	Positivo	Negativo
Positivo	Verdadeiro Positivo (VP)	Falso Positivo (FP)
Negativo	Falso Negativo (FN)	Verdadeiro Negativo (VN)

Descrição campos matriz de confusão:

- Verdadeiro Positivo (VP): quando a classe foi prevista corretamente;
- Falso Positivo (FP): quando a classe foi prevista incorretamente;
- Verdadeiro Negativo (VN): ocorre quando a classe que não estamos buscando prever foi prevista corretamente;
- Falso Negativo (FN): ocorre quando a classe que não estamos buscando prever foi prevista incorretamente.

Cálculo da métricas de avaliação:

- Precisão (*precision*): é a capacidade de evitar falsos positivos.

$$precision = \frac{VP}{VP + FP} \quad (12)$$

- Revocação (*recall*): qual foi o acerto das amostras positivas.

$$recall = \frac{VP}{VP + FN} \quad (13)$$

- Média F (*F-Measure* ou *FI*): Um valor alto significa que a acurácia é relevante. Pode ser interpretada como uma medida de confiança da acurácia.

$$F - Measure = \frac{2 * precision * recall}{precision + recall} \quad (14)$$

- Acurácia (*accuracy*): são as segmentadas corretamente, o quão bom está o modelo.

$$accuracy = \frac{VP + VN}{VP + FP + VN + FN} \quad (15)$$

- Especificidade (*Specificity*): são as previsões da classe negativa para todos os casos negativos.

$$specificity = \frac{VN}{VN + FP} \quad (16)$$

N. Arquitetura Mask R-CNN

A arquitetura Mask R-CNN é baseada no Faster R-CNN que utiliza redes neurais convolucionais para áreas em imagens, retornando uma caixa delimitadora contendo o objeto de uma determinada classe. A estrutura desta arquitetura é composta por duas etapas. Na primeira há uma série de camadas convolucionais com um regressor e um classificador, que tem como saída um mapa de features que são passados para a Region Proposal Network (RPN). Este classificador determina a probabilidade de um dos pixels do mapa ser da área de fundo ou do objeto. Já o regressor determina as coordenadas da caixa de segmentação do objeto da classe determinada. Após é utilizado uma camada de pooling na saída da RPN, garantindo que todas as imagens terão o mesmo tamanho. Para a classificação final que reconhece o objeto, é utilizado uma ativação. O que difere na arquitetura Mask R-CNN é que após a camada de pooling existe duas sub-redes. Sendo que uma contém as camadas densamente conectadas e a outra são as camadas convolucionais que farão a segmentação dos pixels. O Mask R-CNN é uma rede neural convolucional de detecção de objetos que pode identificar objetos em uma imagem e, em seguida, gerar uma máscara de pixel para cada um. Ele é baseado na estrutura de rede neural convolucional baseada em região (R-CNN) e foi projetado para ser rápido e preciso (Figura 16).

O sistema Mask R-CNN é composto por dois componentes principais:

- A primeira é uma rede neural convolucional profunda que é usada para gerar propostas (regiões de interesse) para objetos potenciais em uma imagem.
- O segundo componente é um modelo Mask R-CNN que pega as propostas geradas pelo primeiro componente e gera máscaras para cada objeto.

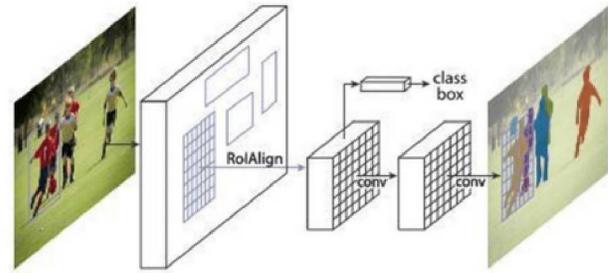


Figura 16. Arquitetura Mask R-CNN. Fonte: [34].

O. Arquitetura Faster R-CNN

A arquitetura Faster R-CNN é um modelo de dois estágios que primeiro extrai recursos de uma imagem usando uma rede neural convolucional (CNN) e, em seguida, usa uma rede de proposta de região (RPN) para gerar propostas para objetos na imagem. Essas propostas são então alimentadas em uma rede Fast R-CNN, que gera as detecções finais de objetos. O Faster R-CNN é mais rápido do que o R-CNN porque compartilha camadas convolucionais entre o RPN e o Fast R-CNN e usa uma camada de pool de região de interesse em vez de uma camada totalmente conectada (Figura 17).

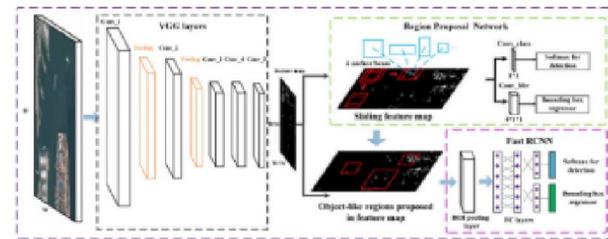


Figura 17. Arquitetura Faster R-CNN. Fonte: [35].

A Faster R-CNN é uma rede neural profunda desenvolvida para detectar objetos. Esta rede possui dois módulos:

- Uma rede de proposta de região (Region Proposal Network (RPN));
- E a Fast R-CNN [Gir15].

Um RPN é uma rede neural que recebe uma imagem como entrada e gera um conjunto de propostas, ou regiões potenciais de interesse, para processamento posterior (Figura 18). Os RPNs são normalmente usados em tarefas de detecção de objetos, onde o objetivo é identificar e localizar objetos em uma imagem. RPNs são um tipo de rede neural convolucional (CNN) e são frequentemente usados em conjunto com outras CNNs em uma arquitetura conhecida como Faster R-CNN. Os RPNs funcionam tomando uma imagem como entrada e, em seguida, deslizando uma pequena janela, ou kernel, pela imagem. Em cada local, o RPN gera um conjunto de pontuações que indicam a probabilidade de que esse local contenha um objeto.

O RPN também gera um conjunto de coordenadas da caixa delimitadora para cada local. Essas caixas delimitadoras são

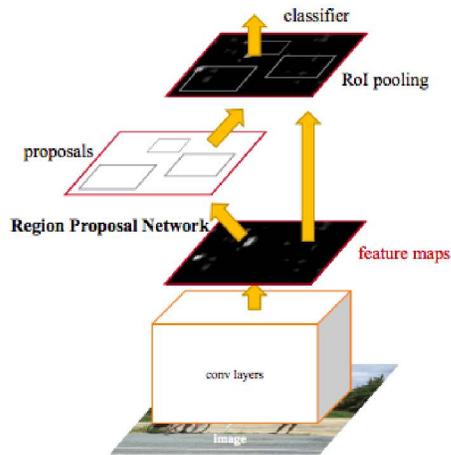


Figura 18. Arquitetura da rede de propostas de regiões (RPN). Fonte: [37].

usadas para recortar a imagem em cada local e as imagens recortadas são então passadas para uma segunda CNN para processamento adicional. O RPN foi projetado para ser computacionalmente eficiente, de modo que possa ser executado em tempo real em uma CPU padrão. Os RPNs provaram ser eficazes para uma variedade de tarefas de detecção de objetos, incluindo a detecção de pessoas em imagens e vídeos.

O segundo módulo da arquitetura Faster R-CNN é o Fast R-CNN [Gir15] (Figura 19) que utiliza regiões propostas que contém objetos. Esta camada abstrai as regiões propostas projetadas no mapa de características da camada de convolução e redimensiona para um tamanho fixo. As regiões são agrupadas e repassadas para as camadas totalmente conectadas para serem classificadas conforme a quantidade de classes.

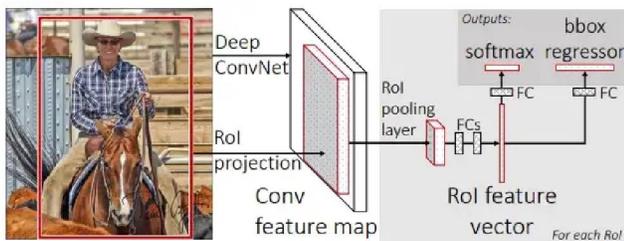


Figura 19. Fast R-CNN Network Architecture. Fonte: [24].

P. Vantagem do uso de uma Base de Imagem Sintética

A vantagem de usar uma base de imagens sintética é por ser muito versátil e poder ser usada para criar uma grande variedade de imagens. Pode ser usado para criar dados de treinamento representativos do conjunto de dados real, mas sem o ruído e a variabilidade que geralmente estão presentes em conjuntos de dados reais. Isso pode ser útil no treinamento de modelos de aprendizado de máquina mais precisos. Além disso, bancos de dados sintéticos podem ser usados para gerar conjuntos de dados maiores e mais diversos do que os disponíveis no mundo real. Existem várias vantagens de criar

um banco de imagens sintético:

- Maior precisão: Construindo cuidadosamente imagens com propriedades conhecidas, é possível obter um grau de precisão muito maior do que com bancos de imagens tradicionais;
- Maior variedade: É possível criar uma maior variedade de imagens do que com os bancos de imagens tradicionais;
- Custo-benefício: O custo de criação e manutenção de um banco de imagens sintético costuma ser menor do que o custo dos bancos de imagens tradicionais.

Q. Base de Treinamento e Teste

No aprendizado de máquina, é comum dividir os dados em um conjunto de treinamento e um conjunto de teste. O conjunto de treinamento é usado para treinar o modelo, enquanto o conjunto de teste é usado para avaliar o modelo.

Há vários benefícios em dividir os dados em um conjunto de treinamento e um conjunto de teste. Permite avaliações mais precisas do modelo. Evita o *overfitting*, que é quando o modelo tem um bom desempenho nos dados de treinamento, mas não nos dados de teste. Permite comparações mais confiáveis entre diferentes modelos. Se dois modelos forem treinados nos mesmos dados, eles provavelmente terão um desempenho semelhante. No entanto, se esses modelos forem avaliados em diferentes conjuntos de dados, o modelo com melhor desempenho no conjunto de teste provavelmente será um modelo melhor. Permite uma experimentação mais flexível. Se o conjunto de treinamento for pequeno, você poderá experimentar modelos diferentes sem se preocupar com o *overfitting*.

A separação dos dados em treino e teste é fundamental, se for realizada de maneira errada poderá resultar em uma modelagem deficiente. Uma das técnicas utilizadas para avaliar esta separação de base é o *Cross-Validation*. Este divide a base de dados em treino e teste, tendo condições de detectar se um modelo está com *overfitting*. A validação cruzada é um método estatístico usado para estimar a precisão dos modelos de aprendizado de máquina [36]. É frequentemente usado em conjunto com outros métodos, como pré-processamento de dados, seleção de recursos e ajuste de parâmetros. Funciona dividindo os dados em um conjunto de treinamento e um conjunto de teste. O modelo é então ajustado no conjunto de treinamento e avaliado no conjunto de teste. Este processo é repetido várias vezes, com diferentes divisões dos dados, até que uma estimativa robusta da precisão do modelo seja obtida. validação cruzada.

A validação cruzada K-fold é um método de validação de um modelo particionando o conjunto de dados em k subconjuntos, treinando o modelo em k-1 subconjuntos e testando o modelo no subconjunto restante. Este processo é repetido k vezes, com cada subconjunto servindo como o conjunto de teste uma vez. A pontuação média de validação em todas as k execuções é então usada para avaliar o modelo. Esse método é benéfico em relação aos métodos tradicionais de

divisão de teste de trem, pois permite melhores estimativas de precisão fora da amostra, pois os dados são usados com mais eficiência. Além disso, pode ajudar a reduzir a variação nas estimativas, pois diferentes subconjuntos de dados são usados para treinamento e teste. No entanto, a validação cruzada k-fold pode ser computacionalmente cara, especialmente para grandes conjuntos de dados. Além disso, pode não ser apropriado para todos os tipos de dados ou modelos.

A escolha do método dependerá dos dados e do tipo de modelo utilizado. É uma ferramenta poderosa que pode ser usada para melhorar a precisão dos modelos de aprendizado de máquina. É importante entender as limitações do método, no entanto, e usá-lo em conjunto com outros métodos, como pré-processamento de dados, seleção de recursos e ajuste de parâmetros.

III. MATERIAIS E MÉTODOS

Após as fotografias iniciais das lâminas de microscopia foi selecionado material a ser utilizado para a preparação do banco sintético de imagens. Para a imagem de fundo foram considerados as lâminas que não tinham a presença de fungos e para as imagens de primeiro plano foi realizado a segmentação somente dos fungos.

A. Aquisição de dados

A base de imagens foi criada a partir de lâminas para microscopia medindo 26 x 76cm. Estas foram preparadas com o fungo *Phakopsora pachyrhizi* da ferrugem asiática da soja no Laboratório de Fitopatologia da UFPR. O material utilizado para aquisição das imagens foram as lâminas preparadas, microscópio óptico modelo Olympus XC40RF 10 e máquina fotográfica Sony DSC-W210.

B. Metodologia para montagem das lâminas

Ao total foram preparadas vinte lâminas para a coleta das imagens. A montagem foi realizada com folhas de soja que apresentavam sinais da ferrugem asiática cultivadas *ex vivo*. Estas foram vigorosamente agitadas sobre as lâminas de vidro contendo uma gota de lactofenol. A finalidade de usar o Lactofenol de Amann é para preparar a lâmina dando condições de ser realizado exame microscópico. Proporcionando uma coloração para os esporos permitindo a fácil identificação conforme Figura 20.

Para proteger a montagem da lâmina é depositado cuidadosamente, com o auxílio de uma agulha histológica, a lamínula em cima da lâmina. A lamínula foi fechada em toda a sua borda com verniz transparente, sendo posteriormente armazenada em estojo próprio (Figura 21) a temperatura ambiente.

C. Coletas das imagens

As coletas foram realizadas no Laboratório de Epidemiologia para Manejo Integrado de Doenças de Plantas (Lemid)



Figura 20. Ação do lactofenol sobre um esporo. Fonte: O Autor.



Figura 21. Estojo para armazenamento de lâminas para microscopia. Fonte: O Autor.

na Universidade Federal do Paraná (UFPR) em setembro de 2021. Para obter as fotografias foi utilizado o procedimento de varredura nas lâminas. É realizado um percurso nas direções horizontal e vertical com o microscópio realizando uma leitura sem sobrepor as áreas já fotografadas. Assim foi possível ter uma ampla variedade de imagens dos esporos da ferrugem asiática da soja. A máquina fotográfica utilizada foi o modelo digital Sony DSC-W210 e para poder ter o encaixe no microscópio foi utilizado um acoplador que permite a fixação da máquina fotográfica ao microscópio (Figura 22).



Figura 22. Máquina fotográfica Sony DSC-W210 com acoplador. Fonte: O Autor.

A máquina fotográfica foi acoplada em microscópio óptico modelo Olympus XC40RF 100, a seleção para ampliação foi de 400x, a lente ocular regulada em 10x e a lente objetiva em 40x (Figura 23).

D. Preparação das imagens

As imagens foram preparadas sendo realizado um redimensionamento para 512 x 512px com um tamanho médio final variando entre 41KB e 68KB. O programa foi desenvolvido em linguagem Python versão 3.7.10 e executado em um Jupyter Notebook. Neste processo a imagem é renomeada a fim de ter



Figura 23. Microscópio óptico modelo Olympus XC40RF 100. Fonte: O Autor.

uma organização, a regra criada especifica características de construção e origem de cada imagem.

Nomenclatura utilizada para organização e identificação das imagens:

- 0001 sequência numérica;
- 0 imagem sem processamento;
- 1 imagem com processamento;
- 400 ampliação utilizada no microscópio;
- 512-512 dimensão da imagem;
- DSC07888.jpg registro na foto na máquina fotográfica.

E. Criação do banco de imagens sintéticas

O banco de imagens sintéticas foi criado no formato COCO com auxílio da biblioteca cocosynth no ambiente GPU do Google colab utilizando a linguagem de programação Python 3.7. A criação é realizada com uma imagem de fundo de uma lâmina para microscopia sem o objeto de interesse (Figura 24) e imagens segmentadas do fungo (Figura 25) que são sobrepostas aleatoriamente em diversos pontos. Para a criação das imagens de primeiro plano foi utilizado o software GIMP – GNU, realizando uma segmentação do objeto sem fundo e recorte da imagem. Para a criação do banco de imagens sintéticas foram utilizadas 30 imagens de fundo (lâmina de microscopia) e 9 imagens de primeiro plano (fungo *Phakopsora pachyrhizi*).



Figura 24. Imagem de fundo: lâmina de microscopia sem a presença de fungo. Fonte: O Autor.

Foi utilizado um programa em linguagem Python versão 3.7.10 sendo executado em um *Google colab* que



Figura 25. Imagens de fungos da Ferrugem Asiática. Fonte: O Autor.

realiza a sobreposição das imagens, rotacionando, alterando o posicionamento, brilho, escala e número máximo de imagens de primeiro plano. O resultado da sobreposição é a criação sintética de uma lâmina com a presença de fungos aleatoriamente dispersos (Figura 26).



Figura 26. Resultado Final da sobreposição. Fonte: O Autor.

Após a criação das imagens sintéticas foi criada uma segunda imagem que representa a máscara referente aos fungos presentes na lâmina (Figura 27).

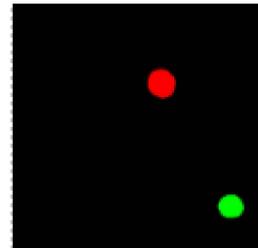


Figura 27. Máscara da imagem criada sinteticamente. Fonte: O Autor.

Para o conjunto de treinamento foi de criado uma base com 4000 imagens sintéticas. E para a base de validação foi utilizado o conjunto de 720 imagens reais que foram realizadas na etapa da microscopia. O uso das bases de treinamento e teste proporciona uma melhor avaliação do modelo.

F. Configuração da rede neural convolucional

Foi utilizado o algoritmo Mask R-CNN no ambiente do *Google colab* com GPU (Figura 28).

Para a transferência de aprendizado foi utilizado a rede ResNet101. Esta uma rede neural convolucional profunda que foi treinada no conjunto de dados ImageNet. Possui uma rede residual de 101 camadas e é capaz de obter um desempenho muito bom em uma ampla gama de tarefas de reconhecimento de imagem. A rede ResNet101 também é relativamente fácil de treinar e não requer uma grande quantidade de dados para obter um bom desempenho. Para o uso do algoritmo Mask R-CNN foram considerados os parâmetros da Tabela III.

GPU Name			Persistence-M		Bus-Id	Disp.A	Volatile Uncorr. ECC	
Fan	Temp	Perf	Pwr:Usage/Cap		Memory-Usage	GPU-Util	Compute M.	MIG M.
0	36C	P8	9W / 70W	Off	00000000:00:04:0	0%	Default	0
N/A					0MiB / 15109MiB			N/A

Figura 28. GPU *Google colab*atory. Fonte: O Autor.

Tabela III
CONFIGURAÇÕES DA REDE MASK-RCNN

Parâmetros	Valores
BACKBONE	resnet101
BACKBONE_STRIDES	[4, 8, 16, 32, 64]
BATCH_SIZE	20
BBOX_STD_DEV	[0.1 0.1 0.2 0.2]
COMPUTE_BACKBONE_SHAPE	None
DETECTION_MAX_INSTANCES	100
DETECTION_MIN_CONFIDENCE	0.7
DETECTION_NMS_THRESHOLD	0.3
FPN_CLASSIF_FC_LAYERS_SIZE	1024
GPU_COUNT	2
GRADIENT_CLIP_NORM	5.0
IMAGES_PER_GPU	10
IMAGE_CHANNEL_COUNT	3
IMAGE_MAX_DIM	320
IMAGE_META_SIZE	14
IMAGE_MIN_DIM	320
IMAGE_MIN_SCALE	0
IMAGE_RESIZE_MODE	square
IMAGE_SHAPE	[320 320 3]
LEARNING_MOMENTUM	0.9
LEARNING_RATE	0.001
MASK_POOL_SIZE	14
MASK_SHAPE	[28, 28]
MAX_GT_INSTANCES	50
MEAN_PIXEL	[123.7 116.8 103.9]
MINI_MASK_SHAPE	(56, 56)
NAME	Rede
NUM_CLASSES	2
POOL_SIZE	7
POST_NMS_ROIS_INFERENCE	500
POST_NMS_ROIS_TRAINING	1000
PRE_NMS_LIMIT	6000
ROI_POSITIVE_RATIO	0.33
RPN_ANCHOR RATIOS	[0.5, 1, 2]
RPN_ANCHOR_SCALES	(8, 16, 32, 64, 128)
RPN_ANCHOR_STRIDE	1
RPN_BBOX_STD_DEV	[0.1 0.1 0.2 0.2]
RPN_NMS_THRESHOLD	0.7
RPN_TRAIN_ANCHORS_PER_IMAGE	256
STEPS_PER_EPOCH	50
TOP_DOWN_PYRAMID_SIZE	256
TRAIN_ROIS_PER_IMAGE	32
USE_MINI_MASK	True
USE_RPN_ROIS	True
VALIDATION_STEPS	5

Para o treinamento foi utilizado o algoritmo Mask R-CNN, sendo executado no Google Colab, Python 3.7.14 e a rede ResNet101 para extrair as características. Para o treinamento foi utilizada a técnica de aprendizado por transferência do modelo pré-treinado MS COCO. Foi utilizada a técnica de aprendizagem por transferência, que inicia a treinamento a partir de pesos sinápticos já treinados no formato cocodata. Para treinamento foram consideradas 100 épocas e para o ajuste dos pesos sinápticos da última camada da rede.

IV. RESULTADOS E DISCUSSÕES

Com a realização do treinamento foi gerado o relatório com as épocas que possuem os valores dos resultados obtidos no modelo. A Tabela IV representa os resultados da época 36 que possui o melhor valor de perda para a validação (loss=0,2153 e val-loss = 0,2203).

Os resultado apresentados do treinamento são demonstrados especificando a perda em: separar o plano de fundo, localizar os objetos, reconhecer e segmentar para treino e validação (val).

Tabela IV
RESULTADO DA ÉPOCA 36 QUE APRESENTOU MENOR PERDA

Perdas	Valores
loss	0,2153
rpn-class-loss	0,0024
rpn-bbox-loss	0,1087
mrcnn-class-loss	0,0131
mrcnn-bbox-loss	0,0272
mrcnn-mask-loss	0,0638
val-loss	0,2203
val-rpn-class-loss	0,0023
val-rpn-bbox-loss	0,1237
val-mrcnn-class-loss	0,0090
val-mrcnn-bbox-loss	0,0271
val-mrcnn-mask-loss	0,0582

Definição das perdas calculadas para Treino e Validação:

- rpn-class-loss: quão bem a RPN separa o plano de fundo com objetos;
- rpn-bbox-loss: quão bem o RPN localiza objetos;
- mrcnn-bbox-loss: quão bem o Mask RCNN localiza os objetos
- mrcnn-class-loss: quão bem o Mask RCNN reconhece a classe de objeto;
- mrcnn-mask-loss: quão bem o Mask RCNN segmenta os objetos;
- loss: somatório de todas as perdas.

O Figura 29 representa os resultados de treinamento do modelo. Observa-se que após a época 36 há uma estabilidade nos valores de saída da função de perda. Isto demonstra que o treinamento alcançou um ponto de estabilidade, e que a continuidade de mais passos não resultam em um modelo melhor. Dar continuidade ao treino do modelo poderá ocasionar a um overfitting, fazendo com que a rede neural perca a capacidade de generalização e identificação da classe treinada.

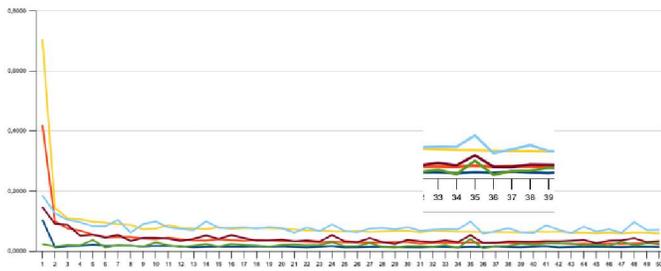


Figura 29. Resultados de treinamento do modelo. Fonte: O Autor.

Os pesos sinápticos da época 36 foram utilizados para a identificação dos fungos na base de imagens de teste. A Figura 30 é de uma identificação da classe phakopsora e respectivas acurácias. Para validar os pesos calculados, foi desenvolvido um programa em linguagem Python 3.7 para identificar os fungos. Ao realizar a identificação utiliza uma caixa delimitadora e calcula a respectiva acurácia.

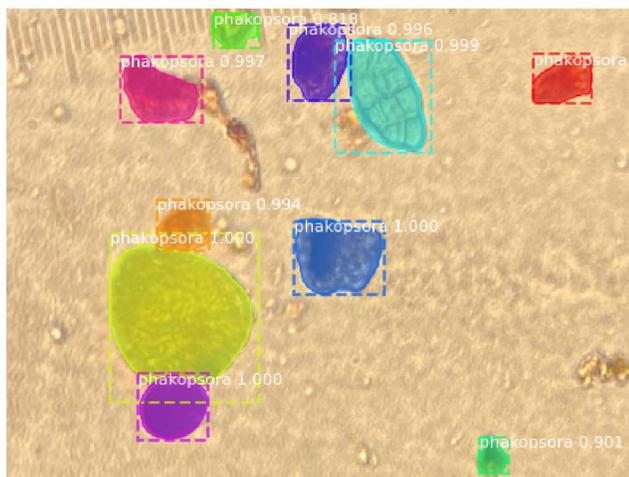


Figura 30. Identificação da classe com os pesos sinápticos calculados. Fonte: O Autor.

No Tabela V temos a matriz de confusão com os resultados da identificação da classe Phakopsora realizada com os novos pesos calculados.

Tabela V
DESEMPENHO DO MODELO

	Classe	Phakopsora	
		Sim	Não
Realidade	Classificação		
	Sim	795	21
	Não	110	178
	Precision	97,42647%	
	Recall	87,84530%	
	F-Measure	92,38815%	
	Accuracy	88,13406%	
	Specificity	89,44724%	

A F-Measure possui um valor alta de confiança na acuracidade do modelo. A Acuracidade na base de imagens sintéticas demonstra um desempenho de capacidade para

segmentação excelente. Demonstrando que o estudo e trabalho proposto resultam em uma prática muito satisfatória. O valor de precisão foi de 97,42647%, demonstrando uma alta capacidade de evitar falsos positivos. O *recall* foi 87,84530% que especifica o acerto das amostras positivas. A média F ou F1 pode ser interpretada como uma medida de confiança da acurácia, ficou com um valor alto e relevante de 92,38815%. A acurácia é o quão bom o modelo está prevendo corretamente, o modelo apresentou um desempenho de 88,13406%. A especificidade representa o acerto da classe negativa para todos os casos negativos 89,44724%.

V. CONSIDERAÇÕES FINAIS

Neste trabalho foi apresentado um estudo para identificação do fungo Phakopsora pachyrhizi, utilizando redes neurais convolucionais e o uso de uma base de imagens sintéticas. A arquitetura utilizada Mask R-CNN para detecção de regiões demonstrou resultados satisfatórios. O método de estudo proposto é promissor, as taxas de acerto ficaram iguais ou superiores a 88%. O estudo demonstrou que as redes neurais conseguem altas taxas de precisão, sem a necessidade de uma grande quantidade de imagens. Este forma de uso é uma alternativa a ser avaliada e aprofundada em novas classes para poder verificar o comportamento da rede.

Para os trabalhos futuros será considerado validar novos resultados sendo:

- Utilizar lâminas de microscopia retiradas de campo como imagens de fundos, tornando mais próximo do real. Estas são lâminas que possuem uma base com muitas possibilidades de estudo, a presença em campo permite extrair muitos elementos de pesquisa;
- Aumentar a quantidade de imagens de fundo e de primeiro plano para a construção do banco de imagens sintéticas;
- Testar outras arquiteturas de CNN;
- Desenvolver o software para contagem de fungos presentes em uma lâmina. Atualmente este procedimento é realizado manualmente, sendo a contagem visual ou com auxílio de algum equipamento.

REFERÊNCIAS

- [1] ALLIPRANDINI, L. F.; ABATTI, C.; BERTAGNOLLI, P. F.; CAVASSIM, J. E.; GABE, H. L.; KUREK, A.; MATSUMOTO, M. N.; OLIVEIRA, M. A. R.; PITOL, C.; PRADO, L. C.; STECKLING, C. 2009. Understanding soybean maturity groups in Brazil: environment, cultivar classification, and stability. *Crop Science*, v. 49, p.801-808.
- [2] GODOY, C. V.; SEIXAS, C. D. S.; MEYER, M. C.; SOARES, R. M. Ferrugem-asiática da soja: bases para o manejo da doença e estratégias antirresistência. Londrina: Embrapa Soja, 2020. 39 p. (Embrapa Soja. Documentos, 428)
- [3] NASCIMENTO, E. O.; OLIVEIRA, L. N. Sensitivity Analysis of Cutting Force on Milling Process using Factorial Experimental Planning and Neural Networks. *IEEE Latin America Transactions*, vol. 14, n. 12, p. 4811-4820, 2016.
- [4] FACURE, Matheus. Funções de Ativação - Entendendo a importância da ativação correta nas redes neurais, 2017.
- [5] BRENT, K.J.; HOLLOMON, D.W. Fungicide resistance: the assessment of risk. Brussels: GCPF, 2007. 52p.

- [6] SCHMITZ, H. K.; MEDEIROS, C. A.; CRAIG, I. R.; STAMMLER, G. Sensitivity of *Phakopsora pachyrhizi* towards quinone outside-inhibitors and demethylation-inhibitors, and corresponding resistance mechanisms. *Pest Management Science*, v. 70, p. 378-388, 2014.
- [7] KLOSOWSKI, A. C.; MAY DE MIO, L. L.; MIESSNER, S.; RODRIGUES, R.; STAMMLER, G. Detection of the F129L mutation in the cytochrome b gene in *Phakopsora pachyrhizi*. *Pest Management Science*, v. 72, p. 1211-1215, 2016.
- [8] SIMÕES, K.; HAWLIK, A.; REHFUS, A.; GAVA, F.; STAMMLER, G. First detection of a SDH variant with reduced SDHI sensitivity in *Phakopsora pachyrhizi*. *Journal of Plant Diseases and Protection*, v. 125, p. 21-26, 2018.
- [9] LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFNER, P. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, nov., 1998.
- [10] KOCHMAN, J.K. The effect of temperature on development of soybean rust (*Phakopsora pachyrhizi*). *Australian Journal of Agricultural Research*, Victoria, n. 30, p. 273-277, 1979.
- [11] KOGAN, M. Integrated Pest Management: Historical perspectives and contemporary developments. *Annual Review of Entomology*, v.43, p.243-270, 1998.
- [12] EMBRAPA SOJA, 2014. Manual de identificação de doenças de soja / Ademir Assis Henning ... [et al.] - 5.ed. Londrina: 76p. : il. color. ; 18cm. - (Documentos / Embrapa Soja, ISSN 1516-781X; n. 256).
- [13] MARCHETTI, M. A.; MELCHING, J. S.; BROMFIELD, K. R. The effects of temperature and dew period on germination and infection by uredospores of *Phakopsora pachyrhizi*. *Phytopathology*, v. 66, p. 461-463, 1976.
- [14] HARTMAN, G. L.; SIKORA, E. J.; RUPE, J. C. Rust. In: HARTMAN, G. L.; RUPE, J. C.; SIKORA, E. J.; DOMIER, L. L.; DAVIS, J. A.; STEFFEY, K. L. (Ed.). *Compendium of soybean diseases and pests*. 5 ed. St. Paul, Minnesota: APS Press, 2015. p. 56-58.
- [15] JACCOUD FILHO, D. S.; HIAR, C. P.; BONA, P. F.; GASPERINI, L. Ocorrência da ferrugem de Soja na Região dos Campos Gerais do Paraná. In: REUNIÃO DE PESQUISA DE SOJA DA REGIÃO CENTRAL DO BRASIL, 23., 2001, Londrina.
- [16] EMBRAPA SOJA, 2013. Tecnologias de produção de soja – Região Central do Brasil 2014. Londrina: Embrapa Soja, 2013. 265 p. 21cm. - (Sistemas de Produção, ISSN 2176-2902; n. 16) https://ainfo.cnptia.embrapa.br/digital/bitstream/item/95_489/1/SP-16_online.pdf
- [17] HIRST, J. M. An automatic volumetric spore trap. *Annals of applied biology*. v. 39, n. 2, p. 257-265, 1952.
- [18] EMBRAPA SOJA, 2020. Coletor de esporos: descrição, uso e resultados no manejo da ferrugem asiática da soja / Circular Técnica 167. (Documentos / Embrapa Soja, ISSN 2176-2864). https://ainfo.cnptia.embrapa.br/digital/bitstream/item/22_0504/1/Circ-Tec-167.pdf
- [19] IGARASHI, W. T.; ISHIKARA, M. S.; OLIVEIRA, G. M.; ABI SAAB, O. J. G.; IGARASHI, S. Comparative morphology between uredospores of species of agricultural importance. *Pesquisa Aplicada & Agrotecnologia* v. 5, n. 3, p. 89-94, 2012.
- [20] VARGAS, A. C. G., PAES, A., & VASCONCELOS, C. N. (2016, July). Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. In *Proceedings of the XXIX Conference on Graphics, Patterns and Images* (pp. 1-4).
- [21] ROSENBLATT, FRANK. "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review* 65.6 (1958): 386.
- [22] MAAS, R., et al. (2013). The reverb challenge: A common evaluation framework for dereverberation and recognition of reverberant speech. In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 1–4. IEEE.
- [23] MCCULLOCH, W. & PITTS, W. (1943). A Logical Calculus of Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, 5, 127–147.
- [24] GIRSHICK, R. "Fast r-cnn". In: *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.
- [25] REIS, E. M.; REIS, A. C.; ZANATTA, M.; SILVA, L. H. C. P.; SIQUERI, F. V.; SILVA, J. R. Evolução da redução de sensibilidade de *Phakopsora pachyrhizi* a fungicidas e estratégias para recuperar a eficiência de controle. *Passo Fundo: Berthier*, 2017. 104 p.
- [26] SCHMITZ, H. K.; MEDEIROS, A. C.; CRAIG, I. R.; STAMMLER, G. Sensitivity of *Phakopsora pachyrhizi* towards quinone-outside-inhibitors and demethylation-inhibitors, and corresponding resistance mechanisms. *Pest Management Science*, v. 7, p. 378-388, 2014.
- [27] MELCHING, J. S.; DOWLER, W. M.; KOOGLER, D. L.; ROYER, M. H. Effects of duration, frequency, and temperature of leaf wetness periods on soybean rust. *Plant Disease*, v. 73, p. 117-122, 1989.
- [28] DEL PONTE, E. M.; GODOY, C. V.; LI, X.; YANG, X. B. Predicting severity of Asian soybean rust epidemics with empirical rainfall models. *Phytopathology*, v. 96, p. 797-803, 2006.
- [29] YORINORI, J. T.; MOREL, W.; FERNANDEZ, F. T. P. Epidemia de ferrugem de soja no Paraguai e na Costa Oeste do Paraná, em 2001. In: REUNIÃO DE PESQUISA DE SOJA DA REGIÃO CENTRAL DO BRASIL, 23., 2001, Londrina. Resumos... Londrina: Embrapa Soja, 2001. p.117-118. (Embrapa Soja. Documentos, 157).
- [30] YORINORI, J. T.; PAIVA, W. M.; FREDERICK, R. D.; COSTAMILAN, L. M.; BERTAGNOLLI, P. F.; GODOY, C. V.; NUNES JUNIOR, J. Epidemics of soybean rust (*Phakopsora pachyrhizi*) in Brazil and Paraguay from 2001 to 2003. *Plant Disease*, v. 89, p. 675-677, 2005.
- [31] JACKSON, S. L.; BAYLISS, K. L. Spore traps need improvement to fulfil plant biosecurity requirements. *Plant Pathology*, v. 60, p. 801-810, 2011.
- [32] ISARD, S. A.; DUFAULT, N. S.; MILES, M. R.; HARTMAN, G. L.; RUSSO, J. M.; WOLF, E. D.; MOREL, W. The Effect of solar irradiance on the mortality of *Phakopsora pachyrhizi* urediniospores. *Plant Disease*, v. 90, n. 7, p. 941-945, 2006.
- [33] ALVES, S. A. M.; FURTADO, G. Q.; GODOY, C. V.; BERGAMIN FILHO, A. Efeito da temperatura e do período de molhamento na germinação de *Phakopsora pachyrhizi*. *Fitopatologia Brasileira*, v. 29, p. S207, 2004.
- [34] XIN WU et al, 2019 "Improvement of Mask-RCNN Object Segmentation Algorithm".
- [35] ZHIPENG DENG et al, 2018. "Multi-scale object detection in remote sensing imagery with convolutional neural networks".
- [36] KOVAKI R. "A study of cross validation and bootstrap for accuracy estimation and model selection." In *Proc 14th Int Joint Conf on AI*, 1995, pp. 1137–1145.
- [37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks"