UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Deep learning em dois estágios para detecção e classificação de doenças em folhas de plantas com aplicação em dispositivos móveis

Tiago de Miranda Leite

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP
SÃO CARLOS

Tiago de Miranda Leite

Deep learning em dois estágios para detecção e classificação de doenças em folhas de plantas com aplicação em dispositivos móveis

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Moacir Antonelli Ponti

USP – São Carlos
Janeiro de 2022

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

Leite, Tiago de Miranda
L533d Deep learning em dois estágios para detecção e classificação de doenças em folhas de plantas com aplicação em dispositivos móveis / Tiago de Miranda Leite; orientador Moacir Antonelli Ponti. -- São Carlos, 2021.
70 p.

Dissertação (Mestrado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2021.

1. Aprendizado profundo. 2. Redes neurais. 3. Doenças em plantas. 4. Detecção de objetos. 5. Aplicativo. I. Ponti, Moacir Antonelli, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Tiago de Miranda Leite

Deep learning in two stages for detection and classification of diseases in plant leaves with application in mobile devices

Dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP – in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Master in Science. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Moacir Antonelli Ponti

USP – São Carlos
January 2022

AGRADECIMENTOS

Meus agradecimentos aos meus pais Maria e Adélio, que não mediram esforços para fornecer uma boa educação a mim e a meus irmãos, João e Elis, a quem também sou grato.

Agradeço ao meu orientador, Professor Dr. Moacir Antonelli Ponti, por todo o apoio, oportunidade e conhecimento que me concedeu.

Ao Instituto de Ciências Matemáticas e de Computação (ICMC) da Universidade de São Paulo (USP), pelos recursos disponibilizados ao desenvolvimento deste trabalho.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) – código de financiamento 001 – pelo apoio financeiro, na modalidade bolsa de mestrado, concedido durante os primeiros meses.

A todos os amigos que conheci em São Carlos, que de alguma forma ou de outra me ajudaram em algum momento.

“Em algum lugar, algo incrível está
esperando para ser descoberto.”
(Carl Sagan)

RESUMO

LEITE, T. DE M. ***Deep learning*** **em dois estágios para detecção e classificação de doenças em folhas de plantas com aplicação em dispositivos móveis**. 2022. 70 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2022.

Deep learning é uma técnica do ramo de aprendizado de máquina que tem obtido grandes resultados em diversas tarefas, quando comparada às demais técnicas da área. Dentre tais tarefas, detecção e classificação de objetos em imagens destacam-se como exemplos notáveis de sucesso. Normalmente, nesse tipo de aplicação, uma única rede neural convolucional realiza o processo de detecção das regiões de interesse, delimitando assim a área que contém o objeto a ser identificado, bem como a classificação dessa região em uma classe. A fim de melhorar os resultados na detecção e classificação de áreas de doenças em imagens de folhas de plantas, este projeto tem como objetivo investigar uma abordagem que utiliza redes neurais convolucionais compostas por dois estágios independentes, um para realizar a detecção e outro a classificação das referidas regiões de doenças, por meio de aprendizado supervisionado. Para validar a abordagem, foram realizados experimentos com três diferentes conjuntos de dados compostos por imagens de folhas de macieira, afetadas por doenças como ferrugem e sarna, com a tarefa de detectar e classificar as regiões de doenças. Os resultados indicam que a abordagem de dois estágios tende a melhorar a precisão média da detecção em imagens de diferentes conjuntos de dados, além de permitir uma melhor transferência de aprendizado quando conjuntos de dados não vistos são usados para teste. Esta abordagem também permite maior flexibilidade na escolha de redes de detecção e classificação para adequar o modelo a cenários específicos. Além disso, as visualizações dos mapas de características dos modelos indicam que as redes de dois estágios apresentam mapas com regiões de ativação mais acentuadas, facilitando a interpretação dos resultados. Por fim, este trabalho também mostrou ser possível a utilização de tais redes neurais por meio de um protótipo de aplicativo para dispositivos móveis (como *smartphones* e *tablets*), permitindo um diagnóstico instantâneo das doenças e a criação de uma base colaborativa de novas imagens, bem como difundindo o uso da tecnologia pela população em geral.

Palavras-chave: *deep learning*, redes neurais, detecção de objetos, doenças em plantas, aplicativo.

ABSTRACT

LEITE, T. DE M. **Deep learning in two stages for detection and classification of diseases in plant leaves with application in mobile devices**. 2022. 70 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2022.

Deep learning is a technique in the machine learning branch that has achieved great results in several tasks when compared to other techniques in the area. Among such tasks, object detection and classification of images stand out as notable examples of success. Normally, in this type of application, a single convolutional neural network performs the process of detecting regions of interest, thus delimiting the area that contains the object to be identified, as well as the classification of that region. In order to improve results in the detection and classification of disease areas in plant leaf images, this project aims to investigate an approach that employs convolutional neural networks composed of two independent stages, one to perform the detection and the other to perform the classification of regions of diseases, by using supervised learning. In order to validate the approach, experiments were carried out using different datasets composed of images of apple leaves, affected by diseases such as rust and scab, with the task of detecting and classifying disease regions. Results indicate that the two-stage approach tends to improve the mean average precision (mAP) of detection on images from different datasets, in addition to allowing better transfer of learning when unseen datasets are used for testing. This approach also allows greater flexibility in choosing the detection and classification networks to tailor the model to specific scenarios. In addition, the visualizations of the feature maps yielded by the models indicate that the two-stage networks have maps with more accentuated activation regions, facilitating the interpretation of the results. Finally, this work also showed that it is possible to use such neural networks through an application prototype for mobile devices (such as smartphones and tablets), allowing an instant diagnosis of plant diseases as well as the creation of a collaborative dataset of new images, spreading the use of technology by the general population.

Keywords: deep learning, neural networks, object detection, plant diseases, mobile application.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

CAPÍTULO

1

INTRODUÇÃO

Doenças que afetam plantações são responsáveis pela perda de milhões de toneladas de alimentos anualmente, causando prejuízos econômicos, sociais e ecológicos. Calcula-se que uma fração de 10% de toda a produção mundial de alimentos é perdida devido às doenças que atacam as espécies vegetais (STRANGE; SCOTT, 2005). Assim, o monitoramento de plantações é fundamental para reduzir a disseminação de doenças e permitir um tratamento rápido e eficaz.

Sabe-se que a maioria das doenças em plantas gera alguma manifestação visível, particularmente em suas folhas (BARBEDO, 2013). Os métodos convencionais de identificação de doenças em folhas geralmente envolvem a atuação de especialistas humanos ou análises em laboratórios que geralmente estão localizados longe das regiões onde essas doenças ocorrem (BOCK *et al.*, 2010), processo que consome tempo e recursos. Sendo assim, em regiões com ausência de profissionais especialistas e de estrutura laboratorial, ou em casos em que uma avaliação rápida e automática seja necessária, a análise de imagem por métodos computacionais torna-se uma alternativa viável.

Nos últimos anos, modelos de aprendizado de máquina baseados em aprendizado profundo (do inglês *deep learning*), em particular as redes neurais convolucionais, ou CNN (sigla do inglês *convolutional neural network*), têm sido aplicadas com sucesso em tarefas de classificação de imagens (GOODFELLOW; BENGIO; COURVILLE, 2016), superando outras técnicas em competições de classificação de imagens (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), tendo assim mostrado ser a melhor técnica na atualidade para extrair atributos para esses tipos de dados (PONTI *et al.*, 2017).

Em geral, as doenças de plantas apresentam alguma manifestação visual em um órgão da planta – especialmente folhas (BARBEDO, 2013) –, o que tem despertado interesse em pesquisas utilizando CNNs para identificar tais doenças em imagens de folhas de diversas espécies de plantas, tais como tomate, maçã, uva e milho (NGUGI; ABELWAHAB; ABO-ZAHHAD, 2020; BOULENT *et al.*, 2019; BARBEDO *et al.*, 2018). Também neste âmbito, as

CNNs têm apresentado superioridade em relação às demais técnicas de extração de características (KAMILARIS; PRENAFETA-BOLDÚ, 2018).

Neste trabalho, investiga-se o uso de CNNs para a detecção e classificação de doenças em imagens de folhas de macieira, por meio de aprendizado supervisionado. A maçã é uma das frutas mais produzidas em todo o mundo e o diagnóstico rápido de doenças que podem atacá-las desempenha um papel vital para a realização de tratamentos adequados e prevenir a rápida propagação dessas doenças, melhorando a produtividade agrícola e reduzindo as perdas (JIANG *et al.*, 2019). Aborda-se essa identificação de doenças como uma tarefa de detecção e classificação de objetos em imagens, em que modelos de CNN são treinados de forma supervisionada para localizar e classificar áreas de doenças. Investigam-se duas estratégias: a primeira, em que o mesmo modelo de CNN realiza o processo de detecção e classificação, e uma segunda abordagem em que as tarefas de detecção e classificação são realizadas separadamente, cada uma pelo uso de uma CNNs independente em relação à outra. Abordagens como essa última têm sido empregadas em problemas de distinção de folhas doentes e saudáveis (SUWA *et al.*, 2019), também como estratégia de pós-processamento para diminuição de falsos positivos (FUENTES *et al.*, 2018), além de problemas de detecção de caracteres, a exemplo competição Kuzushiji Recognition, da plataforma de competições Kaggle[1]. Neste trabalho, pretende-se investigar mais a fundo tal estratégia, no contexto de identificação de regiões de doenças em imagens de folhas de macieira, e compará-las com as abordagens basadas em um único estágio – em que uma CNN realiza simultaneamente a detecção e classificação das referidas regiões de doença.

Além disso, com o objetivo de analisar a capacidade de generalização dos modelos treinados com essas duas abordagens, são utilizados três diferentes conjuntos de dados que contêm imagens de doenças em folhas da maçã, empregando-se métricas tradicionais para avaliação de modelos de detecção e classificação de objetos de imagem. Também foi explorada a visualização das características aprendidas por tais modelos, fornecendo indícios a respeito de quais são os principais atributos da imagem que tendem a ser considerados pelos modelos utilizados. A Figura 1 apresenta um esquema visual resumido a respeito das principais investigações mencionadas.

Por fim, foi criado um protótipo de aplicativo para *smartphone* com sistema Android, como uma maneira de fornecer uma prova de conceito; trata-se de uma aplicação que permitirá a identificação instantânea das classes de doenças em que as redes foram treinadas, através de imagens enviadas pelo usuário. Além de permitir a identificação de tais doenças, essa aplicativo também fornece as visualizações de características e, por fim, possibilita ao usuário fazer o *upload* das imagens a uma base de dados remota, permitindo um aspecto colaborativo por parte dos usuários em continuamente expandir a base de imagens de treinamento. O objetivo dessa etapa é mostrar a aplicabilidade dos métodos estudados no projeto, e sua capacidade de serem utilizados em uma aplicação móvel.

[1] Disponível em https://www.kaggle.com/c/kuzushiji-recognition

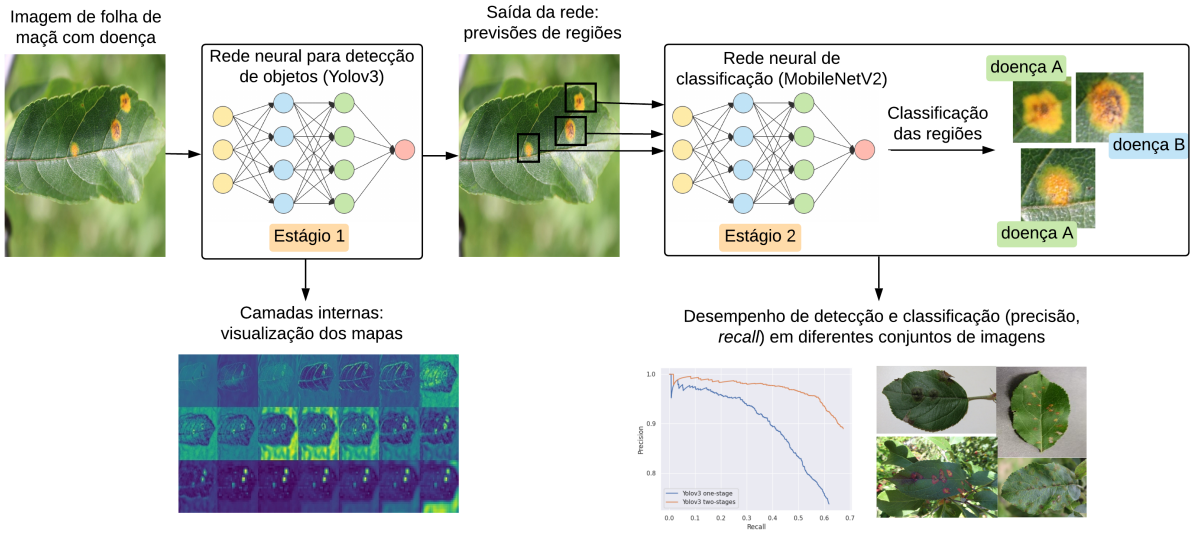


Figura 1 – Visão geral das investigações realizadas neste trabalho.

Fonte: Elaborada pelo autor.

1.1 Hipótese

A composição de dois estágios sequenciais de CNNs, um para realizar a detecção de regiões de interesse (pontos de doenças) e outro para efetuar a classificação de tais regiões, em imagens de folhas com doença, apresenta melhor capacidade de generalização quanto comparado a modelos de detecção baseados em um único estágio.

1.2 Objetivos

Geral: investigar modelos de CNNs que apresentam um único estágio para detecção e classificação de regiões de interesse em imagens, comparando-os com modelos compostos por dois estágios independentes de detecção e classificação, no contexto de identificação de doenças em folhas de plantas, particularmente maçãs.

Como objetivos específicos:

i) a obtenção de diferentes conjuntos de imagens para treinamento e avaliação dos modelos, e anotação de regiões de interesse (áreas das folhas afetadas por doenças) em imagens que não as possuam;

ii) com aprendizado supervisionado, treinar modelos e comparar seus desempenhos de detecção e classificação, utilizando validação interna (teste realizado no mesmo conjunto de dados) e externa (teste realizado em outros conjuntos de dados);

iii) visualização dos mapas de atributos gerados pelas CNNs ao processarem as imagens de teste, com o objetivo de obter uma interpretação visual a respeito dos principais atributos que os modelos aprenderam a distinguir durante seu processo de treinamento;

iv) elaboração de uma prova de conceito, na forma de um aplicativo para *smartphones* com sistema Android que, utilizando o melhor modelo obtido, permita ao usuário enviar uma imagem de uma folha de maçã e receber um diagnóstico a respeito das possíveis doenças presentes na imagem fornecida.

1.3 Organização da monografia

O conteúdo deste trabalho está organizado do seguinte modo:

- O Capítulo 2 apresenta os conceitos fundamentais relacionados ao desenvolvimento do trabalho, descrevendo especialmente as redes neurais e seus modelos para detecção e classificação de objetos em imagens.
- No Capítulo 3 são apresentados os principais trabalhos da literatura que se relacionam ao problema central abordado neste trabalho e utilizam conceitos destacados no Capítulo 2.
- O Capítulo 4 descreve a metodologia empregada, definindo os conjuntos de dados utilizados, os processos de treinamento e as métricas de avaliação dos modelos.
- Os resultados obtidos nos experimentos são apresentados e discutidos no Capítulo 5.
- Por fim, no Capítulo 6 apresentam-se as principais conclusões obtidas neste estudo, bem como sugestões de trabalhos futuros.

CAPÍTULO

2

CONCEITOS FUNDAMENTAIS

Este capítulo apresenta conceitos essenciais ao desenvolvimento deste projeto. Inicialmente, traça-se um retrospecto a respeito do surgimento e progresso das redes neurais, para então apresentar o modelo de rede convolucional e sua caracterização como técnica *deep learning*. Finalmente, realiza-se uma descrição dos principais modelos de redes neurais utilizados em tarefas de detecção e classificação objetos em imagens.

2.1 Evolução das redes neurais

Tomando como base o funcionamento dos neurônios presentes no sistema nervoso dos animais, McCulloch e Pitts (1943) propuseram o primeiro modelo matemático de um neurônio, que funcionava como portas lógicas e era capaz, em combinação, de representar funções *booleanas*. No entanto, tal modelo necessitava de ajuste manual de seus pesos para que pudesse fornecer os resultados esperados como saída. Sua evolução se deu com o trabalho de Rosenblatt (1958), que criou o Perceptron, um classificador binário capaz de aprender a configuração de seus pesos a partir de exemplos de objetos de cada classe fornecidos como amostra de treinamento e fazendo uso de algoritmo de atualização de tais pesos.

Entretanto, devido à sua incapacidade de classificar padrões não linearmente separáveis, tais como a saída de uma simples porta lógica XOR, haveria um período de desinteresse e desapontamento em relação às redes neurais durante a década de 1970. Embora já houvesse a ideia de arranjar neurônios em várias camadas, não era trivial a maneira de executar seu treinamento. O renascimento de interesses na área se deu com o trabalho de Rumelhart, Hinton e Williams (1986), em que se apresentou o algoritmo *backpropagation* como método de treinamento de redes neurais compostas por diversas camadas de neurônios, chamadas de redes *multi-layer Perceptron* (MLP), o que também permitiu que tais redes fossem capazes de resolver problemas de classificação em conjuntos não linearmente separáveis.

Uma das primeiras redes convolucionais a ganhar notoriedade foi a rede LeNet-5 (LECUN *et al.*, 1998), empregada com sucesso para reconhecimento de dígitos em imagens, utilizando o conjunto de dados MNIST. Contudo, foi somente a partir de meados da década de 2000 que o termo *deep learning* começou a ganhar popularidade, a partir da criação das chamadas *deep belief networks* (HINTON; OSINDERO; TEH, 2006), redes neurais profundas cujas várias camadas podiam ser treinadas uma de cada vez.

Em 2012, a rede convolucional AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) venceu o desafio ILSVRC (*ImageNet Large-Scale Visual Recognition Challenge*) por uma grande margem em relação ao segundo colocado, estabelecendo de vez a supremacia das redes convolucionais como modelo *deep learning* para tarefas de classificação de imagens (PONTI *et al.*, 2017). Além disso, a implementação da AlexNet utilizou ideias que atualmente são indispensáveis à criação de modelos *deep learning*, como a utilização de unidades de processamento gráfico (GPU) para o treinamento das redes. A partir de então, avanços na tecnologia de fabricação de GPUs cada vez mais rápidas e sofisticadas permitiram a criação de redes convolucionais cada vez mais profundas, como GoogLeNet (SZEGEDY *et al.*, 2015) e ResNet (HE *et al.*, 2016). Na Tabela 1 apresenta-se um resumo sobre as principais redes convolucionais atuais utilizadas para classificação.

2.2 Redes Neurais Convolucionais e *Deep Learning*

Inspiradas no funcionamento das células nervosas do córtex visual animal (HUBEL; WIESEL, 1962), as redes neurais convolucionais são aquelas que realizam operação de convolução em pelo menos uma de suas camadas; tal operação é capaz de extrair características locais dos dados sobre os quais ela é aplicada, tirando vantagem da estrutura espacial presente em tais dados. Nessas redes, cada camada de convolução apresenta vários filtros, cada um dos quais procura um padrão local específico nos dados.

Ao se tratar de imagens, por exemplo, cada filtro corresponde a um *kernel*, isto é, uma pequena matriz de números reais que realiza o produto escalar com a vizinhança de *pixels* que está abrangendo num dado momento. Cada *kernel* corresponde aos pesos de um nerônio de uma camada convolucional; além disso, todos os neurônios de uma mesma camada convolucional compartilham os mesmos pesos entre si e possuem função de ativação não linear. Por fim, cada neurônio liga-se a somente um pequeno grupo de neurônios adjacentes pertencentes à camada anterior. Essa forma de arranjar os neurônios provoca um efeito equivalente ao movimento de escaneamento do *kernel* ao longo das dimensões da imagem, para a direita e para baixo, realizando-se sucessivos produtos escalares conforme mencionado. Esse processo é chamado de convolução entre o volume de entrada e o *kernel*, fornecendo como resultado um mapa de atributos que corresponde a uma nova versão da imagem de entrada, porém apresentando algum aspecto realçado, ou seja, algum padrão que o filtro procura capturar.

Arquitetura	#Parâmetros	Características
LeNet (LECUN *et al.*, 1998)	60k	Uma das primeiras CNNs a obter sucesso. Menor quantidade de parâmetros em relação às demais redes. Originalmente empregada para reconhecimento de dígitos.
AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012)	60M	Popularizou o uso das CNNs em visão computacional, considerado o primeiro modelo moderno de CNN. Utiliza *dropout* para evitar sobreajuste e função de ativação ReLU. Vencedor do ILSVRC 2012.
ZFNet (ZEILER; FERGUS, 2014)	42,6M	Vencedor do ILSVRC 2013. Similar à rede AlexNet, mas com menos parâmetros. Introduziu o uso de rede deconvolucional para visualização.
VGGNet (SIMONYAN; ZISSERMAN, 2014)	133M a 144M	Segundo colocado no ILSVRC 2014. Variantes como VGG16 e VGG19 demonstraram que a profundidade da rede é um fator determinante de desempenho. Utiliza múltiplos filtros 3×3 em sequência em vez de filtros maiores (como 5×5 ou 7×7).
GoogLeNet (SZEGEDY *et al.*, 2015)	7M	Vencedor do ILSVRC 2014. Também conhecida como "Inception", trata-se de modelo bem profundo, com toda a arquitetura tendo cerca de 100 camadas no total.
ResNet (HE *et al.*, 2016)	25,6M a 60,2M	Utiliza o conceito de blocos residuais na construção de redes bem profundas, com variantes atingindo até cerca de 1200 camadas. Vencedor do ILSVRC em 2015.
MobileNet (HOWARD *et al.*, 2017)	$4,2$M	Considerável redução do número de parâmetros e de operações, ainda atingindo acurácia próxima aos demais modelos como VGG. Desenvolvida para funcionar em dispositivos móveis.
DenseNet (HUANG *et al.*, 2017)	$7,1$M	Conexões densas entre uma camada e todas as anteriores, evitando o problema de *vanishing gradient* e facilitando a propagação de atributos pela rede. Número reduzido de parâmetros.

Tabela 1 – Comparação das redes convolucionais profundas relacionadas principalmente ao desafio ImageNet.

A estrutura típica de uma rede convolucional, como ilustrada na Figura 2, inclui, além de camadas de convolução, camadas de agrupamento (*pooling*) e camadas totalmente conectadas (com neurônios conectados a todos os neurônios da camada anterior). As camadas de convolução,

conforme já mencionado, apresentam filtros que extraem características locais dos dados de entrada, agrupando-as em canais de saída denominados mapas de atributos. Para permitir a diminuição da dimensionalidade de tais mapas, normalmente são aplicadas camadas de *pooling* após cada convolução, o que também torna invariante a extração de atributos no espaço, ou seja, independente de sua localização espacial no volume de entrada. Uma camada de *pooling* toma um grupo de valores de entrada contidos numa janela e os resume a um único valor, normalmente o maior entre eles (neste caso denominado *maxpooling*). Finalmente, para fins de classificação, conecta-se uma ou mais camadas totalmente conectadas ao fim da rede, permitindo-lhe realizar a distinção entre as diferentes classes presentes.

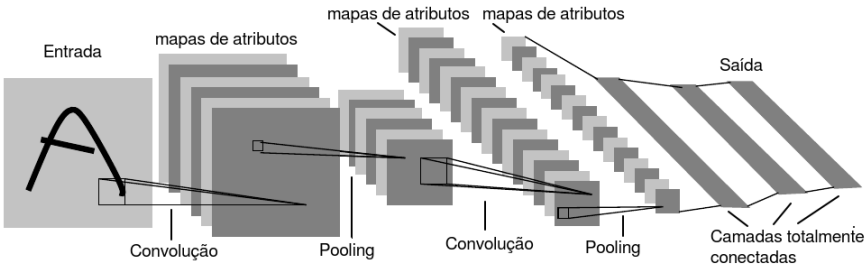


Figura 2 – Exemplo de rede convolucional e seus tipos de camadas.

Fonte: Adaptada de Lecun *et al.* (1998).

Várias camadas de convolução seguidas de *pooling* podem ser sequencialmente arranjadas, de modo que mapas de atributos de uma camada são fornecidos como entrada para a camada seguinte. Assim, o aprendizado ocorre segundo uma hierarquia de conceitos: à medida que aumenta sua profundidade na rede, uma camada aprende filtros que extraem atributos cada vez mais complexos e abstratos que a camada anterior. Esse processo automático, incremental e hierárquico de extração de características cada vez mais abstratas diretamente dos dados de entrada, eliminando o processo manual de *feature engineering*, confere às redes neurais convolucionais a principal qualidade que as caracteriza como modelos *deep learning* (PONTI *et al.*, 2017). De fato, os filtros aprendidos ao longo da rede extraem informações relevantes das imagens de entrada, as quais são úteis tanto para a base de dados com a qual a rede foi treinada, quanto para outras bases de dados, generalizando para outras aplicações e dados (SANTOS *et al.*, 2020).

O processo de aprendizado corresponde a minimizar o valor de uma função de custo diferenciável associada à rede; tal função fornece um valor de distância entre o valor final fornecido pela rede e o valor esperado fornecido pelos rótulos das amostras de treinamento, para o caso de aprendizado supervisionado. Utilizando ferramentas do cálculo como derivadas parciais e a regra da cadeia, é possível obter os valores dos gradientes da função de custo em relação a cada peso dos neurônios da rede e, através do método *backpropagation*, obter os valores de atualização de tais pesos: começando com o valor retornado pela função de custo a partir da última camada da rede, retrocede-se das camadas finais para as camadas iniciais, aplicando a

regra da cadeia para calcular a contribuição que cada peso teve no valor do custo. Logo, após cada iteração de treinamento, espera-se que a função de custo tenha seu valor gradualmente diminuído e, consequentemente, é natural que o erro cometido pela rede neural também decresça. Assim, aprende-se de modo automático, no caso das CNNs, quais os valores dos filtros necessários para extrair os atributos mais importantes da imagem, responsáveis por melhor distinguir as amostras de treinamento dentre as classes consideradas.

2.3 Redes neurais convolucionais para detecção de objetos

Detecção de objetos é um problema clássico em visão computacional: dada uma imagem, deve-se determinar uma ou mais sub-regiões retangulares que contêm objetos de interesse, bem como classificar tais objetos, conforme ilustrado na Figura 3. Para resolver tal tipo de problema, a rede deve fornecer, como saída, as coordenadas espaciais que definem a sub-região, bem como um vetor de probabilidades de classificação de cada objeto. Assim, pode-se interpretar essa tarefa como um problema misto de regressão dos valores das coordenadas das regiões e de classificação de tais regiões dentre as possíveis classes existentes (ZHAO *et al.*, 2019). Os exemplos mais bem sucedidos nesse contexto são as redes descritas nas seções a seguir.

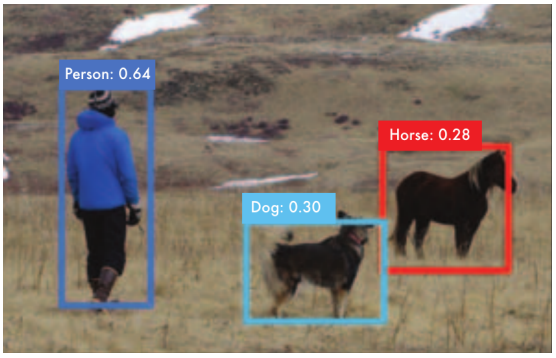


Figura 3 – Exemplo de imagem em que são realizadas detecção e classificação de objetos: deve-se fornecer a localização e a classe dos objetos de interesse.

Fonte: Redmon *et al.* (2016)

2.3.1 R-CNN, Fast R-CNN e Faster R-CNN

Uma das primeiras iniciativas em utilizar redes convolucionais para localização e detecção de objetos se deu com o trabalho de Girshick *et al.* (2014), criando as R-CNNs (*regions with CNN features*) – a Figura 4 fornece uma representação de sua arquitetura. Nessas redes, um processo de busca seletiva (UIJLINGS *et al.*, 2013) propõe um conjunto inicial de cerca de 2000 regiões de interesse, prováveis de conter um objeto a ser detectado. Tais regiões, após serem redimensionadas, são fornecidas como entrada para uma rede convolucional que então extrai um vetor de características a ser utilizado por uma Máquina de Vetor de Suporte (SVM – do inglês

Support Vector Machine) para realizar a classificação de cada região. Esse tipo de técnica era comumente utilizada, visto que a otimização de SVMs corresponde a um dos problemas mais bem estudados e com melhores garantias teóricas de convergência (MELLO; PONTI, 2018).

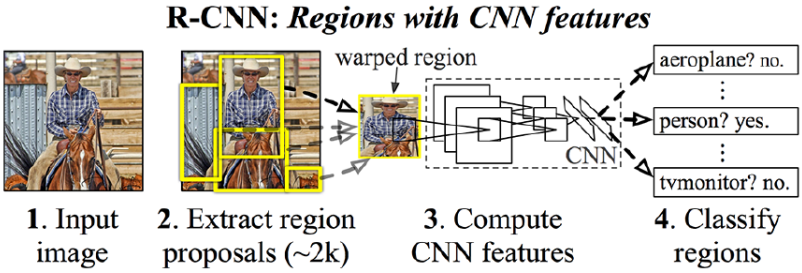


Figura 4 – Representação da arquitetura de uma R-CNN.

Fonte: Girshick *et al.* (2014).

Um dos principais problemas das R-CNNs é seu alto tempo de processamento; além disso, o mecanismo de extração de regiões de interesse – busca seletiva – é um processo independente da rede neural, não havendo aprendizado no que diz respeito à previsão de regiões com base nos exemplos de treinamento, já que somente o processo de classificação é treinado.

Para contornar tais problemas, pelo menos em parte, Girshick (2015) desenvolveu uma espécie de atualização da R-CNN, denominada Fast R-CNN (Figura 5). Nessa rede, as regiões de interesse são projetadas sobre uma mapa de atributos extraído da imagem original por camadas de convolução; então, cada projeção passa por uma camada de *pooling* customizada (denominada *RoI pooling*) para extração de atributos, reduzindo a região a um vetor de tamanho fixo, que é assim enviado a uma série de camadas totalmente conectadas, que por sua vez dividem-se em dois ramos: um para realizar a classificação do objeto e o outro para refinar as coordenadas da região de interesse que contém tal objeto, realizando regressão.

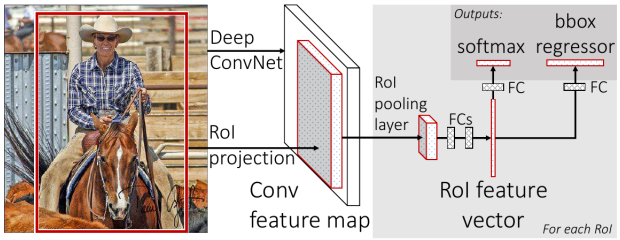


Figura 5 – Esquema de uma Fast R-CNN.

Fonte: Girshick (2015).

Apesar de serem mais rápidas que as R-CNNs, as Fast R-CNNs ainda estão limitadas pelo gargalo do processo independente de geração de regiões de candidatas, como a busca seletiva. Para resolver tal problema, Ren *et al.* (2015) apresentaram a rede **Faster R-CNN**, na qual a geração de regiões de interesse é realizada pela própria rede, para então ocorrer sua classificação e regressão, do mesmo modo que se dá na Fast R-CNN (Figura 6). Assim, a rede Faster R-CNN

é composta por dois módulos: o primeiro é uma rede completamente convolucional (FCN – do inglês *fully convolutional network*), responsável por propor as regiões de interesse, denominada *Region Proposal Network* (RPN); já o segundo módulo corresponde essencialmente a uma rede Fast R-CNN, responsável por classificar e refinar as localizações das regiões de interesse fornecidas.

A imagem de entrada inicialmente passa por uma rede de base para extração de atributos. Então, o módulo RPN utiliza um filtro de tamanho 3×3 que faz a convolução sobre o mapa de atributos de dimensões $W \times H$, obtido a partir da imagem original, gerando então outro mapa de atributos de profundidade 256 (ou 512, dependendo da implementação). Observe a Figura 6. Para cada posição do mapa de entrada, associam-se k regiões âncora (*anchor boxes*), responsáveis por detectar objetos em diversas proporções de tamanho. Após a primeira convolução, dois ramos paralelos se dividem: o primeiro corresponde a uma camada convolucional que realiza a classificação binária da âncora, indicando se esta contém um objeto *qualquer* de interesse ou simplesmente o fundo da imagem; o segundo ramo, também convolucional, faz a regressão entre as quatro coordenadas espaciais da âncora (x_{centro}, y_{centro}, *largura* e *altura*) para melhor ajustá-las à posição real do objeto. O primeiro e o segundo ramo apresentam filtros de tamanho 1×1 e fornecem como saída, respectivamente, mapas de dimensões $W \times H \times 2k$ e $W \times H \times 4k$. A obtenção do mapa de atributos inicial, a ser utilizado pelo módulo RPN, pode ser obtido utilizando alguma rede de base a depender da aplicação, como redes mais profundas tais como Resnet (HE *et al.*, 2016), ou mais compactas, como a MobileNet (HOWARD *et al.*, 2017).

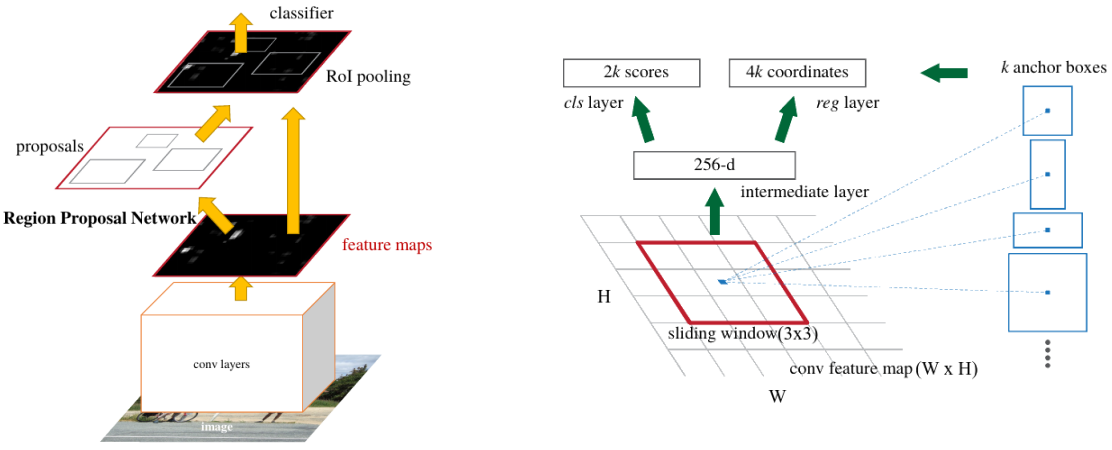


Figura 6 – Faster R-CNN.

Fonte: Adaptada de Ren *et al.* (2015).

Com as redes do tipo Faster R-CNN, tornou-se possível o treinamento de ponta-a-ponta de um modelo de rede neural para detecção de objetos. Além disso, tais redes reduziram drasticamente o tempo de processamento, chegando a operar a uma velocidade de detecção de 7 *frames* por segundo (FPS) em GPU, e atingiram resultados comparáveis ao estado da arte em detecção de objetos utilizando os conjuntos PASCAL VOC 2007 e 2012 (REN *et al.*, 2015).

Os modelos de detecção de objetos apresentados até aqui são constituídos de diferentes

módulos que realizam uma determinada parte da tarefa de localização e classificação. Ou seja, existe um componente responsável por extrair os mapas de atributos a partir da imagem original, outro por propor as regiões de interesse e finalmente um último para realizar a classificação e regressão das regiões. Existem, no entanto, arquiteturas de redes neurais que realizam o procedimento completo de classificação e regressão, através de um único fluxo ao longo da rede sem utilizar o mecanismo de geração de regiões de interesse. Nas seções a seguir são descritos dois exemplos bem-sucedidos de tais redes.

2.3.2 Yolo

Redmon *et al.* (2016) propuseram a rede **Yolo** (*you only look once*). Sua estratégia consiste em dividir a imagem em uma grade de tamanho $S \times S$ e, a cada célula da grade, associar um conjunto de B caixas âncora (como na Faster R-CNN). Cada âncora corresponde a um conjunto de 5 valores: x_{centro}, y_{centro}, *largura*, *altura*, p – este último indica um valor de probabilidade de que a âncora contenha um objeto. Uma célula é responsável por detectar um objeto se ela contiver as coordenadas do centro desse objeto, utilizando as caixas âncora para melhor ajustar a detecção daqueles objetos que apresentam diversas proporções de tamanho. Além disso, a cada célula atribui-se um vetor de probabilidades relativo às P classes de objetos presentes.

Assim, a saída da rede corresponde a um tensor de dimensões $S \times S \times (B * 5 + P)$. A implementação original utiliza 24 camadas convolucionais para extração de características, seguidas de 2 camadas totalmente conectadas. Para o conjunto de imagens PASCAL VOC, que apresenta $P = 20$ classes, utilizou-se $S = 7$, além de $B = 2$ âncoras por célula, resultando num tensor de saída de dimensões $7 \times 7 \times 30$, conforme a Figura 7.

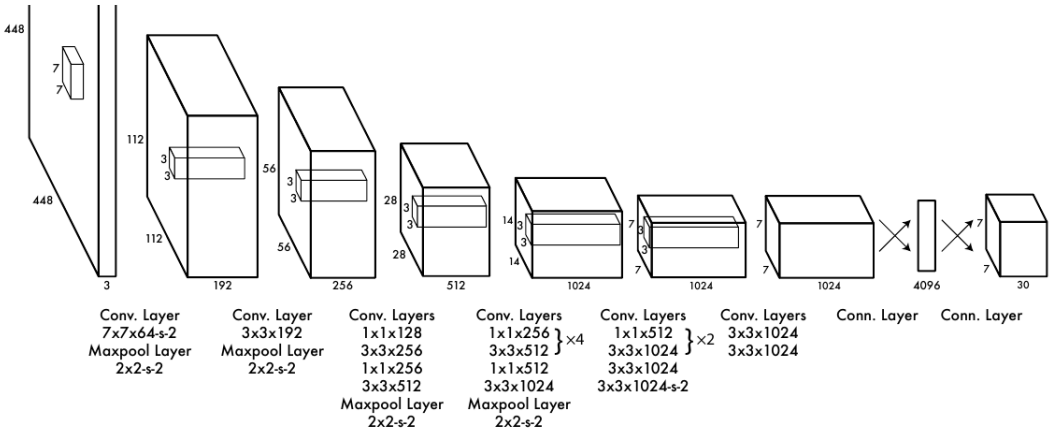


Figura 7 – Camadas da rede Yolo.

Fonte: Redmon *et al.* (2016).

Uma versão aprimorada da rede Yolo, chamada **Yolov2**, é proposta no trabalho seguinte (REDMON; FARHADI, 2017), melhorando os resultados de detecção ao adicionar normalização de lotes (*batch normalization*) (IOFFE; SZEGEDY, 2015), além de melhorar o mecanismo de

geração de caixas âncoras e de empregar uma nova rede de 19 camadas (Darknet-19) como extrator de características.

Em seguida, a rede **Yolov3** (REDMON; FARHADI, 2018) trouxe ainda mais aprimoramentos, ao realizar as detecções em três diferentes escalas (sendo capaz de melhor detectar objetos menores), além de fazer uso de blocos residuais (como na rede ResNet) e utilizar um número maior de camadas de convolução, com filtros de tamanho 3×3 e 1×1, para extração de características, totalizando o 53 camadas, assim denominada Darknet-53, cujas camadas e dimensões dos filtros são informados na Figura 8. A arquitetura completa da rede Yolov3, incluindo, além da Darknet-53, as camadas responsáveis pela detecção, é ilustrada na Figura 9.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figura 8 – Camadas da rede Darknet-53, utilizada pela rede Yolov3 como extrator base de características.

Fonte: Redmon e Farhadi (2018)

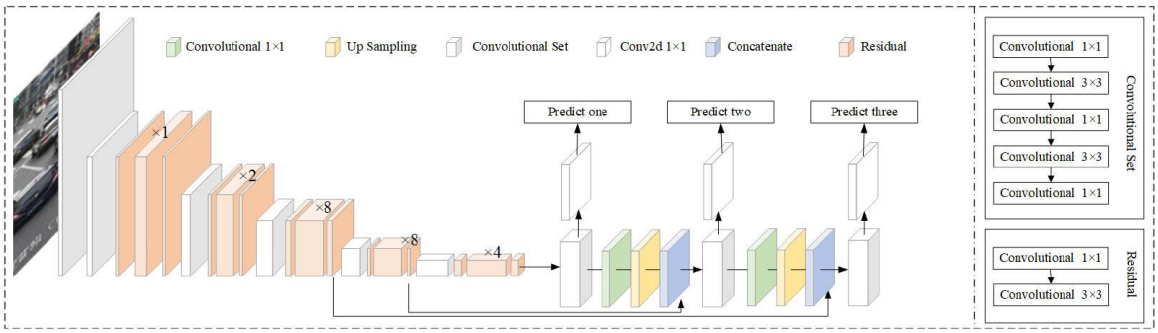


Figura 9 – Representação da arquitetura da rede Yolov3, contendo aprimoramentos como detecção em três escalas e uso de blocos residuais, melhorando o desempenho de detecção em relação a suas versões anteriores.

Fonte: Mao *et al.* (2019)

2.3.3 SSD: single shot detector

Outra arquitetura de rede neural para detecção de objetos em tempo real foi proposta por Liu *et al.* (2016), denominada **SSD** (do inglês *Single-Shot Detector*). Nessa rede, após extraírem-se os atributos da imagem a partir de uma rede base (como VGG16 ou MobileNet, por exemplo), adiciona-se uma sequência de diversas camadas convolucionais complementares que então extraem mapas de atributos de diferentes tamanhos (ver Figura 10). Cada mapa apresenta um conjunto de B caixas âncora, como na Yolo, e os diferentes tamanhos de mapa permitem detectar objetos em diferentes escalas; além disso, diversos tamanhos de âncoras permitem abranger objetos de várias proporções. Observe a Figura 11. Para cada posição espacial de um mapa de tamanho $M \times N$, associam-se B caixas âncora. Cada âncora calcula P valores de probabilidade para classificação e 4 valores de deslocamento (*offset*) em relação à localização das suas coordenadas originais, o que faz com que a rede forneça como saída um tensor de dimensões $M \times N \times (B \times (P+4))$ para um dado mapa de tamanho $M \times N$.

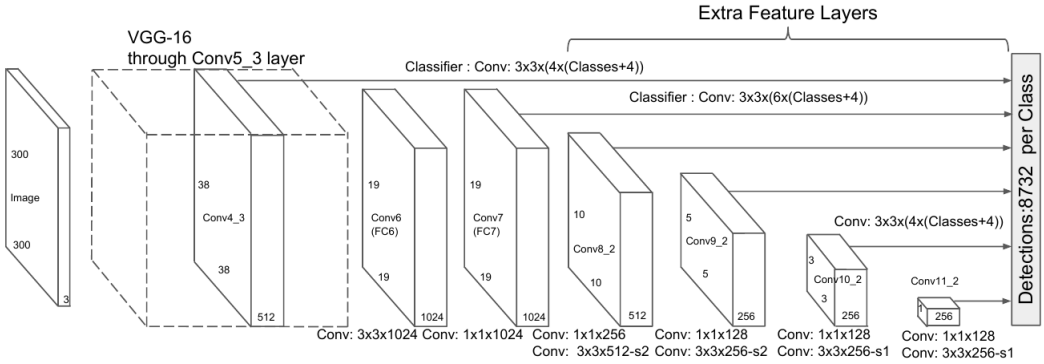


Figura 10 – Camadas da rede SSD.

Fonte: Liu *et al.* (2016)

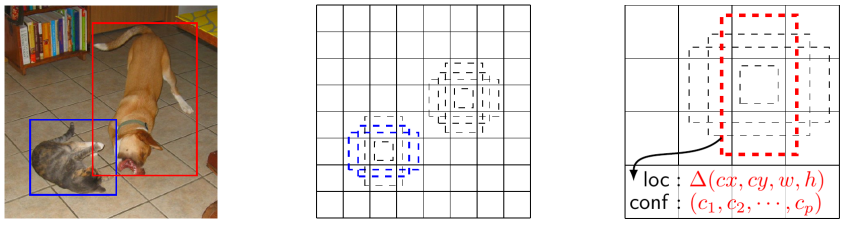


Figura 11 – Representação das caixas âncora da rede SSD.

Fonte: Liu *et al.* (2016)

A função de custo para treinamento da rede utiliza uma soma ponderada entre o custo de classificação e o de localização, para todos os mapas. Entropia cruzada é a função de custo utilizada para classificação, enquanto a de localização utiliza a função *smooth L1* (GIRSHICK, 2015) para fazer a regressão das coordenadas das regiões previstas. Como nas redes da família Yolo, todo o processamento ocorre ao longo de um único fluxo através da rede, o que a torna capaz de realizar detecção de objetos em tempo real.

CAPÍTULO

3

TRABALHOS RELACIONADOS

Segundo Sankaran *et al.* (2010), a identificação de doenças em plantas compreende métodos diretos, como técnicas moleculares para identificação dos agentes causadores da doença, e métodos indiretos, tais como espectroscopia e processamento de imagens. Os métodos diretos, por envolverem análise química, têm como principal limitação o fato de serem trabalhosos e consumirem tempo, requerendo procedimentos elaborados para sua execução, além de serem invasivos. Métodos indiretos que envolvem espectroscopia e processamento de imagens multi ou hiperespectrais, embora tenham mostrado avanços nos últimos anos, demandam o uso de sensores muitas vezes caros e volumosos, o que também torna o processo caro e pouco acessível (BARBEDO, 2013; NGUGI; ABELWAHAB; ABO-ZAHHAD, 2020). Entretanto, considerando que a maioria da doenças podem ser percebidas no espectro visível da luz (BARBEDO, 2013), e que a obtenção de imagens através de câmeras digitais presentes em dispositivos eletrônicos é cada vez mais frequente no cotidiano, o processamento desse tipo de imagem, captada a partir da luz visível, mostra-se uma alternativa satisfatória. Sendo assim, o escopo deste trabalho limita-se ao emprego desse último tipo de imagem.

Técnicas de aprendizado de características baseadas em *deep learning* têm se mostrado o estado da arte em diversos domínios, entre eles reconhecimento de padrões em imagens (LECUN; BENGIO; HINTON, 2015). Em um contexto mais específico, envolvendo a identificação de doenças em imagens de plantas, tais técnicas também têm se mostrado promissoras, superando métodos clássicos de processamento de imagem (KAMILARIS; PRENAFETA-BOLDÚ, 2018). Segundo Barbedo (2013), métodos tradicionais como análise de cores, realce, *thresholding* e análise discriminante apenas geraram pequenos avanços na área nas últimas décadas, principalmente devido ao fator de serem muito específicos e funcionarem bem somente sob certas condições. Assim, diversos trabalhos recentes têm se voltado ao uso de técnicas *deep learning*. Nesse contexto, as duas abordagens principais para identificação de doenças em plantas são classificação e detecção.

3.1 Métodos baseados em classificação

Trata-se de métodos que realizam a classificação da imagem como um todo, indicando sua categoria, sem informar a localização da região de interesse na imagem. Tendem a ter bom desempenho quanto o objeto de interesse ocupa a maior parte da imagem, mas geralmente tem seu desempenho afetado por condições como existência de outros objetos ou fundo ruidoso. São as técnicas mais frequentes de identificação de doenças em plantes via *deep learning* encontradas na literatura.

Sladojevic *et al.* (2016) utilizaram redes convolucionais para classificação de 13 tipos de doenças em folhas de 4 espécies de plantas (pera, pêssego, maçã e videira), além de incluir uma classe para folhas saudáveis, obtendo acurácias entre 91% e 98%, com média de 96,3%. Mohanty, Hughes e Salathé (2016) utilizaram os modelos GoogLeNet e AlexNet para identificar 26 doenças em 14 culturas, usando o conjunto de dados público Plant Village[1] (HUGHES; SALATHE, 2015), que contém 54309 imagens de folhas de plantas doentes e saudáveis. O melhor modelo obtido foi a rede GoogLeNet pré-treinada com o conjunto ImageNet, que alcançou uma acurácia de 99,35%. No entando, ao testar o modelo em um novo conjunto de imagens de teste coletado em condições diferentes, a acurácia caiu drasticamente para cerca de 31%. Ainda utilizando o mesmo conjunto de dados Plant Village, Ferentinos (2018) comparou o desempenho das redes AlexNet, GoogLeNet, Overfeat, AlexNetOWTBn e VGG16, alcançando com esta última 99,53% de taxa de sucesso na classificação das imagens; novamente, os modelos foram menos robustos quando treinados com imagens coletadas em campo e testados com imagens de laboratório, com taxa de acerto caindo para 65,69%, enfatizando a necessidade de estabelecer um número de imagens grande o suficiente, representando condições reais e diversas nos dados de treinamento, para gerar modelos mais robustos para seu uso em campo. De fato, Barbedo (2018b) aponta que ainda há ausência de uma base de imagens abrangente o suficiente para treinar modelos com maior generalização.

Em KC *et al.* (2019), variações da rede MobileNet foram analisadas, apresentando desempenho próximo à rede VGG ao classificar 55 classes compostas por pares espécie-doença do conjunto Plant Village: a versão reduzida de MobileNet chegou a 98,34% de acurácia de classificação, com 29 vezes menos parâmetros que a VGG e 6 vezes menos parâmetros que a versão original da MobileNet. Amara, Bouaziz e Algergawy (2017) treinaram a rede LeNet em um subconjunto de imagens do Plant Village contendo doenças em bananeira. A rede foi capaz de obter mais de 90% de acurácia de classificação, mesmo em imagens sob condições variadas e desafiadoras de iluminação, resolução, orientação e fundo ruidoso. Wang, Sun e Wang (2017) usaram as redes VGG16, VGG19, Inception-v3 e ResNet para diagnosticar 4 graus de severidade de uma doença comum em culturas de maçãs conhecida como podridão negra, também utilizando um subconjunto de imagens do Plant Village. As redes foram tanto treinadas do zero como pré-treinadas com a base ImageNet (*transfer learning*). A acurácia geral do melhor

[1] Disponível em: https://plantvillage.psu.edu/

modelo (VGG16 com *transfer learning*) para o conjunto de dados de teste foi de 90,4%. Em Liu *et al.* (2018) utiliza-se uma rede derivada da AlexNet para classificação de 4 tipos de doenças foliares em macieiras (mosaico, ferrugem, mancha marrom e mancha de Alternária), chegando a uma acurácia média de 97,6%.

Em outro estudo, DeChant *et al.* (2017) investigaram a identificação de uma doença popularmente conhecida como queima das folhas em culturas de milho, em condições de campo. Para tanto, foi desenvolvido um modelo composto de um *pipeline* de classificação composto por 3 etapas. Inicialmente, 5 CNNs independentes são treinadas em um conjunto de sub-imagens de mesmo tamanho, originado a partir de um recorte da imagem original. Na etapa 2, mapas de calor são gerados usando combinações das CNNs da etapa anterior. Na última etapa, uma última CNN recebe os mapas de calor da etapa anterior como entrada e retorna uma probabilidade de a imagem conter uma região com a doença; a acurácia reportada é de 96,7%.

Barbedo (2019) utiliza sub-regiões que contêm apenas as áreas da imagem que apresentam doenças, para treinamento do modelo de classificação. Gera-se assim um número maior de amostras de treinamento a partir de uma única imagem, eliminando as partes irrelevantes e assim permitindo que o aprendizado foque em aspectos mais importantes da imagem. A acurácia média de classificação através desse pré-processamento atinge 94%, em contraste com o valor de 82% obtido utilizando a imagem inteira. Além disso, o autor cria e disponibiliza a base de imagens utilizada[2].

3.2 Métodos baseados em detecção

Grande parte dos trabalhos recentes tratam o problema de identificação de doenças em plantas como classificação da imagem como um todo, utilizando modelos de redes que fornecem uma única saída para uma dada imagem de entrada, conforme os trabalhos mencionados na seção anterior. Um número menor de trabalhos, no entanto, abordam tal problema como uma tarefa de detecção de objetos, fornecendo, além da classificação, também a localização das regiões de doenças na imagem.

Em Ramcharan *et al.* (2019) utiliza-se a rede SSD, juntamente com a rede MobileNet como base, para detecção de 7 tipos de doenças, subdivididas em 2 graus de severidade cada, em folhas de mandioca (*Manihot esculenta* Crantz), por meio de um sistema que opera em *smartphones*. Os resultados obtidos apontaram que o sistema, apesar de ter bom desempenho de classificação em sintomas pronunciados, encontra dificuldade ao classificar doenças com sintomas pouco evidentes, sugerindo a existência de maior variabilidade de imagens com essa característica para o treinamento do modelo. Já Selvaraj *et al.* (2019) trataram da detecção doenças e pragas em bananeiras, utilizando três modelos de CNN – ResNet, Inception-V2 e

[2] Disponível em https://www.digipathos-rep.cnptia.embrapa.br/

MobileNet – com os detectores Faster R-CNN e SSD, obtendo diversos resultados cuja acurácia variou de 62% a 99%, a depender da parte de planta e do modelo utilizado.

No trabalho desenvolvido por Suwa *et al.* (2019) implementa-se um método de identificação de folhas doentes em pepino baseado em um sistema de dois estágios: no primeiro, realiza-se a detecção das folhas doentes, separando-as do fundo ou de outros elementos presentes na imagem; em seguida, as regiões detectadas são enviadas a um segundo módulo, o qual é composto por uma rede neural comum para classificação. Utilizando conjuntos obtidos de fazendas diferentes para treino e teste, os resultados mostraram que tal técnica apresentou maior capacidade de generalização e menor sobre-ajuste quando comparada aos detectores de um único estágio, em que a mesma rede detecta e classifica. O sistema não permite, no entanto, saber qual a doença em questão, apenas classifica as folhas em doentes ou saudáveis.

O experimento descrito por Fuentes *et al.* (2017) teve como objetivo superar algumas limitações existentes ao lidar com imagens em diferentes condições de iluminação, cores e tamanhos das regiões de doença, bem como existência de fundo de imagem ruidoso. A fim de criar um detector em tempo-real para reconhecimento 9 doenças e pragas em folhas de tomateiro, além de folhas saudáveis, os autores coletaram 5000 imagens. Após a anotação de *bounding boxes* ao redor das regiões de doença, o número de amostras de treinamento aumentou para 43398. Diversas redes profundas, como AlexNet, VGG-16, GoogLeNet e ResNet, foram utilizadas como extratores de características, juntamente com detectores como Faster R-CNN e SSD. A combinação de Faster R-CNN e VGG apresentou precisão média de 83,06%. No trabalho seguinte (FUENTES *et al.*, 2018), os autores utilizaram dois módulos de diagnóstico independentes: o primeiro corresponde a uma rede Faster R-CNN responsável por prever as prováveis regiões de interesse, enquanto a segunda unidade é um banco de filtros composto por várias CNNs, uma para cada possível classe, responsáveis por eliminar algumas detecções de falsos positivos, o que resultou em uma melhora relatada de 13% de precisão média, quando comparada com o trabalho anterior (FUENTES *et al.*, 2017).

Um resumo dos trabalhos mencionados é apresentado na Tabela 2.

3.3 Considerações finais

Neste capítulo foram apresentados e descritos os principais trabalhos encontrados na literatura sobre identificação de doenças em imagens de plantas que utilizam *deep learning*. Embora a maioria deles tratem desse problema como uma tarefa de classificação ou detecção em um único estágio, estratégias híbridas, envolvendo ambas abordagens, mostraram-se promissoras nesse tipo de problema.

Nota-se, ainda, que a maioria dos trabalhos encontrados na literatura treinam e testam seus modelos em conjuntos de dados similares, isto é, em subconjuntos de conjunto maior. Por exemplo, verifica-se que a maior parte dos trabalhos utilizam o conjunto de imagens Plant

Village, subdividindo-o em frações de treinamento e teste. Barbedo (2018a) considera que a realização de testes em conjuntos de dados distintos e, assim, mais desafiadores, devam ser realizados para a obtenção de avanços mais significativos na área.

Tendo em vista essa certa lacuna que existe quando se trata de aprender com um conjunto de dados de imagens e avaliar a capacidade de generalização do modelo resultante quando testado em dados coletados em condições diferentes, este trabalho investiga como melhorar detecção e classificação de doenças de plantas usando uma modelo de dois estágios. Em particular, mostra-se como tais modelos podem ser usados, após o treinamento, com novos dados para realizar as tarefas de detecção e classificação, avaliando-se seu desempenho. Além disso, investiga-se a extração de atributos da imagem pelos modelos por meio da visualização das ativações de suas camadas internas.

Trabalho	Abordagem	Modelo	Dataset	Descrição	Acurácia
Sladojevic *et al.* (2016)	Classificação	CaffeNet	Próprio (4483 imagens)	13 classes de doenças em 5 espécies	91% – 98%
Mohanty, Hughes e Salathé (2016)	Classificação	AlexNet e GoogleNet	Plant Village	26 doenças em 14 espécies	99,35%
Amara, Bouaziz e Algergawy (2017)	Classificação	LeNet	Plant Village (3700 imagens)	2 doenças em banana	92% – 99%
Barbedo (2019)	Classificação	GoogleNet	Próprio	79 doenças em 14 espécies	82% – 94%
Lu *et al.* (2017)	Classificação	Baseado em AlexNet	Próprio (500 imagens)	10 doenças em arroz	95,48%
Ferentinos (2018)	Classificação	Diversas CNNs	Plant Village	26 doenças em 14 espécies	95,48%
Liu *et al.* (2018)	Classificação	Baseado em AlexNet	Próprio (13689 imagens)	4 doenças em maçãs	97,62%
DeChant *et al.* (2017)	Classificação	*Pipeline* de CNNs	Próprio (1796 imagens)	1 doença em milho	96,7%
Wang, Sun e Wang (2017)	Classificação	Diversas CNNs	Plant Village (2086 imagens)	1 doença em maçã (4 graus de severidade)	90,4%
KC *et al.* (2019)	Classificação	MobileNet e variantes, VGG	Plant Village	55 classes (pares espécie-doença)	98,34%
Ramcharan *et al.* (2019)	Detecção	SSD + MobileNet	Próprio (2415 imagens)	7 doenças em mandioca (2 graus de severidade)	75% – 94% (mAP)
Selvaraj *et al.* (2019)	Detecção	(Faster R-CNN, SSD) + (ResNet, Inception-V2, MobileNet)	Próprio (18000 imagens)	8 doenças em bananeira	62% – 99%
Suwa *et al.* (2019)	Detecção e classificação	(Faster R-CNN, SSD) + VGG,	Próprio	1 doença em pepino	86% (F1-score médio)
Fuentes *et al.* (2017) / Fuentes *et al.* (2018)	Detecção / Detecção e classificação	(Faster R-CNN, SSD) + (AlexNet, VGG-16, GoogLeNet, ResNet)	Próprio (5000 imagens)	9 doenças em tomate	83% / 96% (mAP)

Tabela 2 – Resumo dos trabalhos de identificação de doenças em imagens de plantas baseadas em *deep learning*.

CAPÍTULO

4

MATERIAIS E MÉTODOS

4.1 Considerações iniciais

A ideia principal deste projeto consiste em analisar o desempenho de um modelo de detecção e classificação de objetos em imagens baseado em dois estágios e compará-lo com o modelo de um estágio, no contexto de reconhecimento de doenças em imagens de plantas. Acredita-se que, ao separar o processo de classificação do processo de detecção de regiões, pode-se obter um modelo mais robusto, capaz de obter melhor generalização, já que cada tarefa é dada a redes específicas. Esse tem sido o cenário típico de implementações vencedoras em uma competição de reconhecimento de caracteres, conforme mencionado no Capítulo 1, além de ter sido explorado em trabalhos recentes (SUWA *et al.*, 2019; FUENTES *et al.*, 2018) evidenciando o potencial de tal abordagem. Além disso, é possível associar diversas redes de classificação a uma única rede de detecção, esta última sendo agnóstica às classes, e assim fornecer maior flexibilidade quanto aos tipos de doenças ou espécies que são de interesse. Considerando um possível uso do modelo em um aplicativo de *smartphones*, tal flexibilidade é conveniente pois permite o lançamento incremental de diversas versões da aplicação, sem a necessidade de retreinar todo o modelo a cada nova atualização.

4.2 Proposta

Neste trabalho, investiga-se o uso de dois modelos de redes neurais para detectar regiões de doenças em imagens de folhas de macieira. O primeiro modelo consiste em usar uma rede neural convolucional para detecção de objetos, Yolov3 (REDMON; FARHADI, 2018), fornecendo simultaneamente como saída as caixas delimitadoras que demarcam as regiões da doença, bem como um valor de confiança (probabilidade) de classificação para cada classe (conforme a Figura 12). Já o segundo modelo consiste em nossa proposta de abordagem envolvendo duas etapas, cada uma realizando estas ações separadamente: inicialmente, a mesma rede Yolov3,

agnósticas às classes, desta vez detectam as regiões de interesse que contenham sintomas de uma doença, sem fornecer sua classe; posteriormente, outra rede convolucional classifica as caixas delimitadoras detectadas pela primeira rede, atribuindo uma classe a cada região que fora identificada – conforme ilustrado na Figura 13.

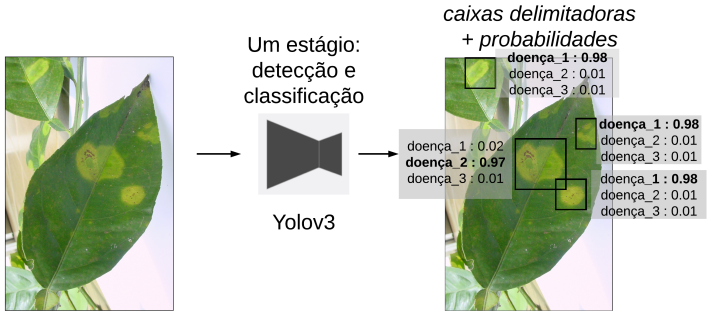


Figura 12 – Modelo típico de detecção de objetos, em que a detecção e a classificação são realizadas ao mesmo tempo, o que chamamos de abordagem de um estágio.

Fonte: Elaborada pelo autor.

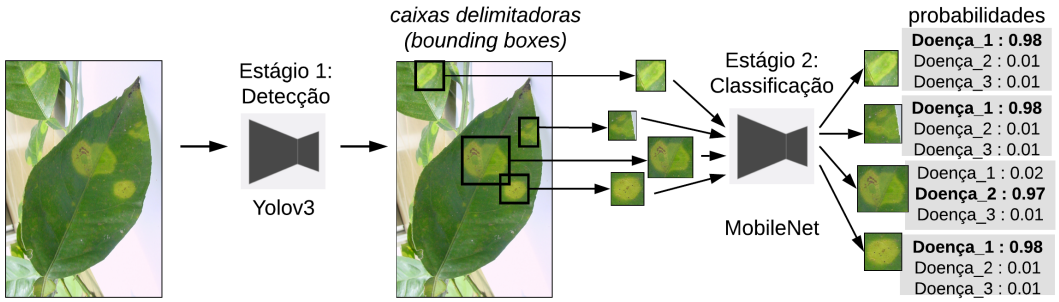


Figura 13 – Modelo de detecção e classificação baseado na abordagem proposta em dois estágios.

Fonte: Elaborada pelo autor.

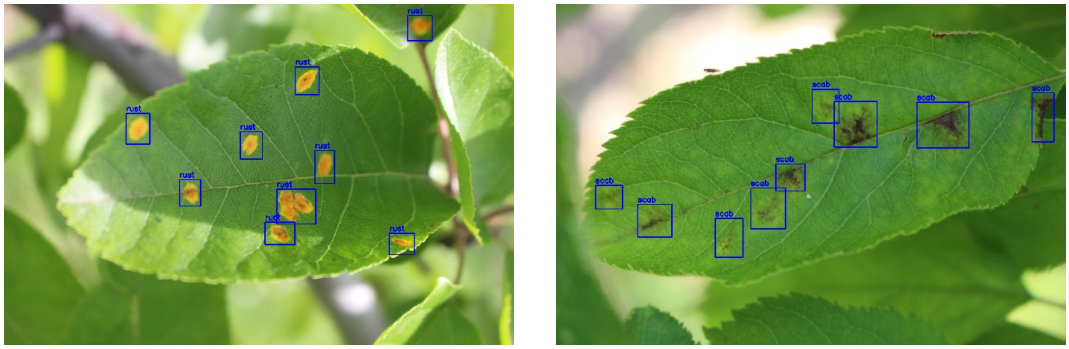
4.3 Conjuntos de dados e métricas de avaliação experimental

O conjunto de dados utilizado para o treinamento das redes foi obtido no site Kaggle[1], no contexto do desafio *Plant Pathology 2020* (THAPA *et al.*, 2020). Tal conjunto contém um total de 1820 imagens de folhas de macieira, distribuídas em 2 classes de doenças: **ferrugem** e **sarna**, que são duas doenças comuns que afetam as folhas de macieiras. Além disso, contém folhas saudáveis e algumas imagens que mostram as duas doenças na mesma folha. Neste trabalho,

[1] https://www.kaggle.com/

um subconjunto dessas imagens foi usado, contendo apenas as imagens em que apenas uma das doenças está presente, ao qual nos referimos como o conjunto de dados $\text{apple}_{pathology}$. As imagens foram coletadas em condições de campo e apresentam situações variadas de iluminação, fundo, sombras e reflexos.

Uma vez que se trata de um problema de detecção de objetos, anotamos as imagens do conjunto de dados, delimitando a região das doenças e fornecendo sua anotação de classe (ferrugem ou sarna), uma vez que não havia anotações de *bounding boxes* no conjunto de dados original. Para tanto, utilizou-se a ferramenta LabelImg (versão 1.8.1), disponível gratuitamente [2]. Embora as imagens pertencentes à classe "ferrugem" apresentassem sintomas bem definidos e facilmente delimitados, exemplos frequentes da classe "sarna" geralmente apresentavam sintomas sem uma delimitação óbvia – neste caso, forma anotadas regiões aproximadamente homogêneas, seguindo a mesma abordagem de anotação de sintomas empregada em (BARBEDO, 2019). Assim, foram anotados um número total de 1.048 arquivos de imagem e 11.328 caixas delimitadoras. A Figura 14 fornece exemplos de anotações para cada classe.



(a) (b)

Figura 14 – Exemplos de imagens anotadas para as doenças de maçã: ferrugem (a) e sarna (b).

Fonte: Elaborada pelo autor.

Além disso, para avaliar a robustez dos modelos ao lidar com imagens provenientes de outras fontes, um segundo conjunto de dados foi utilizado para a realização de novos testes de detecção e classificação. Empregou-se, assim, o conjunto de imagens descrito e disponibilizado por Nachtigall, Araujo e Nachtigall (2017), o qual contém 6 classes de doenças e deficiências nutricionais em folhas de maçã, dentre as quais apenas as imagens que pertencem à classe "sarna" foram usadas, pois é a única em comum com o conjunto de dados previamente utilizado para treinar as redes ($\text{apple}_{pathology}$), conforme descrito anteriormente. Para essas novas imagens, foram anotadas 2.727 caixas delimitadoras em 177 amostras de imagens de folhas apresentando a referida doença (exemplo na Figura 15), seguindo os mesmos critérios de anotação usados no primeiro conjunto de dados; referimo-nos a este conjunto de dados como apple_{unseen}.

[2] https://github.com/tzutalin/labelImg

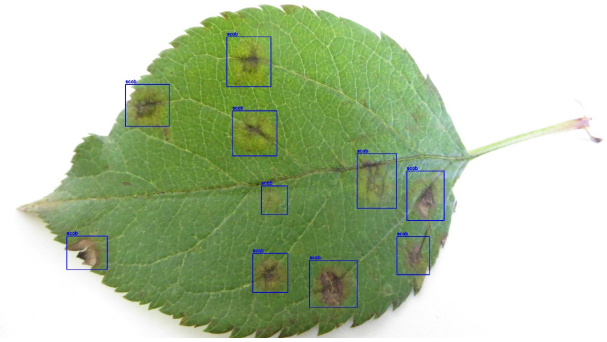


Figura 15 – Exemplo de imagem anotada do conjunto apple$_{unseen}$, acometida pela doença “sarna”.

Fonte: Elaborada pelo autor.

Com o objetivo de avaliar o desempenho dos modelos de redes utilizados neste trabalho, calculou-se a precisão média da média (mAP – *mean average precision*); trata-se de uma métrica comumente empregada em problemas de detecção de objetos em imagens (PADILLA; NETTO; SILVA, 2020). Para tanto, definimos o valor de intersecção sobre união (IoU): dadas duas caixas delimitadoras b_1 e b_2, seu valor IoU é dado pela Equação 4.1:

$$IoU(b_1, b_2) = \frac{área(b_1 \cap b_2)}{área(b_1 \cup b_2)} \tag{4.1}$$

Dados um conjunto de caixas delimitadoras previstas pela rede b_{preds} e uma determinada caixa delimitadora esperada b_{gt} (*ground truth*), associamos esta última à caixa prevista $b_p \in b_{preds}$ que tem o maior valor de IoU com b_{gt}. Além disso, pode-se definir um limite para IoU, o que significa que todas as previsões com valores IoU menores que um limiar não serão consideradas corretas. A precisão média para cada classe, AP, é dada pela área abaixo da curva de *precision-recall*. Seguindo a forma de calcular definida em Everingham *et al.* (2010), computa-se o valor de AP para cada classe ao fazer a média da precisão ao longo de um série de 11 valores de *recall* igualmente espaçados: $[0; 0,1; ...; 0,9; 1]$ (Equação 4.2).

$$AP = \frac{1}{11} \sum_{r \in [0;0,1;...;1]} p_{interp}(r) \tag{4.2}$$

O valor interpolado da precisão para o *recall* r, designado como $p_{interp}(r)$, é obtido tomando-se o valor máximo da precisão cujo correspondente valor de *recall* excede r, conforme a Equação 4.3.

$$p_{interp}(r) = \max_{\tilde{r}:\tilde{r} \geq r} p(\tilde{r}) \tag{4.3}$$

A Figura 16 fornece uma representação visual desse procedimento. Finalmente, o valor de mAP é dado pela média dos valores de AP por todas as classes presentes (Equação 4.4).

$$mAP = \frac{1}{N} \sum_{i=1}^{N} (AP)_i \tag{4.4}$$

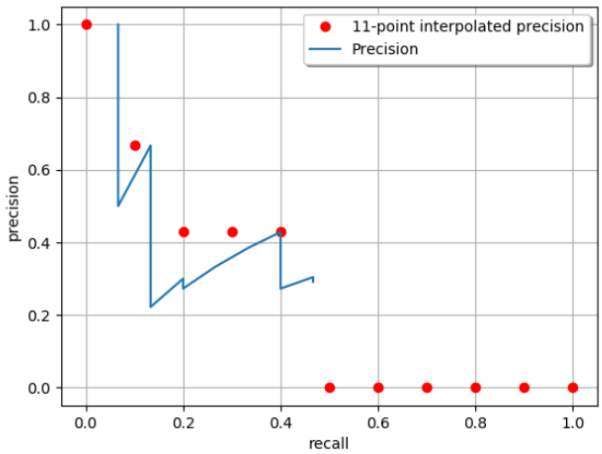


Figura 16 – Exemplo do procedimento para cálculo de AP.

Fonte: Padilla, Netto e Silva (2020).

Com o objetivo de avaliar as redes treinadas em uma tarefa puramente de classificação de imagens, ou seja, fornecendo uma única classificação geral a respeito da doença presente na imagem como um todo, utilizaram-se imagens do mencionado conjunto de dados Plant Village, o qual também contém amostras de folhas de macieira afetadas por ferrugem, sarna e folhas saudáveis, que foram então selecionadas do restante do conjunto. Essas 3 classes somam 2550 imagens no total: 1645 amostras de folhas saudáveis, 630 mostrando sarna e 275 atacadas pela ferrugem da maçã. O referido conjunto de dados já fornece as amostras de imagens separadas em diretórios, sendo um diretório por classe. Exemplos de imagens desse conjunto são informados na Figura 17.

(a) (b) (c)

Figura 17 – Imagens de amostra obtidas do conjunto de dados Plant Village, apresentando folhas afetadas por ferrugem (a) e sarna (b), além de exemplares saudáveis (c).

Fonte: Elaborada pelo autor.

Nesse caso do conjunto Plant Village, foi realizado um pós processamento das detecções fornecidas pelas redes neurais. Isso porque, embora as redes neurais utilizadas tenham retornado como saída as quatro coordenadas das caixas delimitadoras detectadas, bem como um valor de probabilidade para cada possível classe, estamos interessados aqui em obter uma única

classificação no que diz respeito à imagem como um todo. Para tanto, primeiramente foi calculada, para cada imagem, a probabilidade média das caixas delimitadoras, agrupadas por classe. Por exemplo, no caso de uma previsão contendo três caixas delimitadoras referentes à classe de ferrugem, com probabilidades de $0,5$, $0,6$ e $0,7$, e duas para a classe de sarna, com probabilidades de $0,2$ e $0,3$, temos valores médios de $0,6$ e $0,25$ para cada classe, respectivamente. Em seguida, pesam-se essas médias considerando o número de caixas delimitadoras previstas como peso; portanto, para o exemplo dado, a classe de ferrugem tem a pontuação final de $0,6 \times 3/(3+2) = 0,36$ e a classe de sarna, $0,25 \times 2/(3+2) = 0,1$. A classe com a maior pontuação (no exemplo, ferrugem) foi então a classe escolhida. Imagens sem previsão alguma de caixas delimitadoras foram consideradas como pertencentes à classe "saudável", uma vez que a rede não detectou nenhuma região de doença.

Com as pontuações de classificação calculadas, as métricas tradicionais para classificação de imagens – precisão (P), *recall* (R) e *F1-score* ($F1$) – foram computadas da seguinte forma:

$$P = \frac{TP}{TP+FP}, \quad R = \frac{TP}{TP+FN}, \quad F1 = \frac{2 \times P \times R}{P+R}, \tag{4.5}$$

em que TP corresponde ao número de verdadeiros positivos (classificações corretas), FP refere-se a falsos positivos (classificações erradas) e FN é o número de falsos negativos (classificações ausentes).

Em resumo, têm-se os seguintes conjuntos de dados e tarefas relacionadas:

- **apple**$_{pathology}$: usado para treinar redes de um e dois estágios para a tarefa de detecção e classificação, avaliada por meio de validação cruzada 5-folds;
- **apple**$_{unseen}$: todo o conjunto de dados foi usado apenas para fins de teste, para a tarefa de detecção e classificação da doença da classe "sarna".
- **Plant Village**: utilizado para a tarefa de classificação com transferência de aprendizado (*transfer learning*), avaliado usando uma divisão de tamanhos iguais de treinamento e teste em três situações – (a) redes treinadas do zero, (b) redes pré-treinadas com apple$_{patology}$, e (c) redes pré-treinados com apple$_{patology}$ e, em seguida, treinadas com os dados de treinamento do Plant Village, num processo conhecido com ajuste fino (*finetuning*). As redes utilizadas foram tanto a de um como a de dois estágios, com o processo de *finetuning* treinando todas as suas camadas.

4.4 Tecnologias utilizadas

Para a implementação dos modelos, foram utilizadas as bibliotecas de aprendizado de máquina Keras(CHOLLET; others, 2015) (versão 2.2.4), TensorFlow (versão 1.8.0) (ABADI *et al.*, 2016), MXNet (versão 1.6.0) e GluonCV (0.7.0) (GUO *et al.*, 2020), além de outras

bibliotecas auxiliares como Pandas (0.24.2) e Numpy (1.17.0), todas disponíveis para a linguagem de programação Python 3.5.2. Quanto ao desenvolvimento do protótipo do aplicativo, utilizou-se o ambiente integrado de desenvolvimento Android Studio 3.2, juntamente com a linguagem de programação Java 1.8.

4.5 Configuração e treinamento das redes

Conforme mencionado, a rede neural de detecção de objetos Yolov3 foi usada para a composição do modelo de um estágio e para a primeira parte do modelo de dois estágios. Em ambos os casos, foi utilizada a implementação da rede Yolov3 do *framework* de aprendizado profundo GluonCV, o qual fornece uma API de alto nível que permite carregar e treinar diversas redes neurais profundas. Carregou-se a rede Yolov3 com os seus pesos pré-treinados no conjunto de dados Pascal VOC (EVERINGHAM *et al.*, 2010) (apenas para evitar inicialização de pesos totalmente aleatória), a qual foi então treinada usando o conjunto de dados $\text{apple}_{pathology}$ com todas as imagens redimensionadas para 512×512; o otimizador utilizado foi o Adam (KINGMA; BA, 2014) com taxa de aprendizado inicial de 10^{-3}, com redução para a metade do valor a cada 20 épocas; todos os outros parâmetros foram mantidos como os valores padrão fornecidos pelo *framework* GluonCV. Todo o processo de treinamento durou 250 épocas e usou *minibatches* de tamanho 4. Tanto para o modelo de um e quanto de dois estágios, o experimento foi realizado usando uma estratégia de validação cruzada 5-folds com o conjunto de dados $\text{apple}_{pathology}$ mencionado.

Além disso, para o modelo de dois estágios, foram removidos os rótulos de classe do conjunto de dados, substituindo-os pelo rótulo genérico “doença”. Em seguida, na etapa de classificação, uma segunda rede foi usada – uma arquitetura MobileNetV2 (SANDLER *et al.*, 2018), usando como imagens de treinamento as caixas delimitadoras extraídas do conjunto de dados de treinamento (exemplos na Figura 18), todas redimensionadas para 64×64. Desta vez, os rótulos das classes das caixas delimitadoras foram fornecidas a esta segunda rede, permitindo assim o treinamento da etapa de classificação, também utilizando um procedimento de validação cruzada 5-folds.

4.6 Visualização dos mapas de atributos

Com o objetivo de interpretar as representações internas que as redes (tanto de um quanto de dois estágios) aprenderam a fim de distinguir as caixas delimitadoras e identificar as doenças, foram obtidos os valores dos mapas de atributos das camadas internas da rede Yolov3 para algumas imagens dos conjuntos de teste, tornando assim possível visualizar as maiores ativações de tais mapas, o que fornece indícios sobre o que está sendo considerado pelas redes ao fornecer suas as predições.

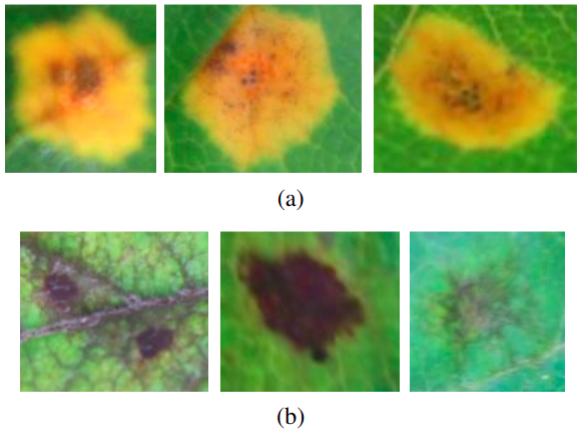


Figura 18 – Exemplos de imagens de treinamento para a rede de classificação, compostas de caixas delimitadores recortadas das imagens originais, pertencentes às classes ferrugem (a) e sarna (b).

Fonte: Elaborada pelo autor.

Conforme mencionado na Seção 2.3.2, a rede Yolov3 utiliza a rede Darknet-53 como extrator base de características das imagens e, como se pode observar na Figura 19, suas camadas convolucionais são agrupadas em blocos que abrangem duas camadas convolucionais (com filtros de tamanho 1×1 e 3×3) e uma camada residual, havendo também, entre as sequências de blocos, uma camada convolucional simples (com filtro 3×3 e *stride* 2), a qual também pode ser considerada um bloco simples. Assim, levando em conta esses agrupamentos das camadas em blocos, pode-se contabilizar um total de 28 blocos de camadas, excluindo-se a primeira camada e as duas camadas finais da rede (*avgpool* e *connected*) – conforme indicação da linha vertical tracejada na Figura 19.

Note que os filtros, dentro de cada bloco, apresentam quantidade variável, iniciando com 32 na primeira camada, depois 64, também 64 ao fim do primeiro bloco, e assim por diante, atingindo o valor 1024 ao fim do último bloco. Para gerar as visualizações, tomaram-se os filtros com os maiores valores (representando as maiores ativações dos neurônios da rede), dentro de cada um dos 28 blocos considerados, a fim de se obter uma ideia a respeito dos atributos extraídos pelos filtros presentes nesses blocos da rede. As imagens geradas nas visualizações correspondem a mapas de calor obtidos a partir dos valores dos atributos extraídos pelos filtros de maiores ativações, normalizados para o intervalo real [0, 1].

4.7 Arquitetura e funcionamento do aplicativo protótipo

Conforme mencionado, desenvolveu-se um protótipo funcional de um aplicativo para *smartphones* e *tablets* com sistema Android, que permite realizar o processo de detecção e classificação instantânea das doenças consideradas neste trabalho, por meio do envio de imagens a partir da câmera ou da galeria de fotos do dispositivo. Além disso, também há a opção para o

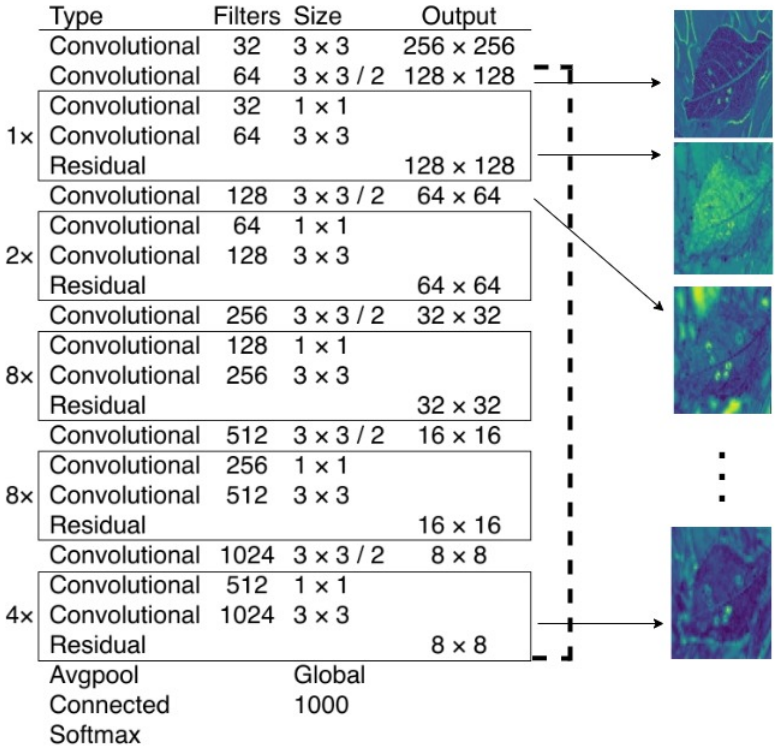


Figura 19 – Esquema de obtenção dos atributos extraídos pelos blocos de camadas da rede.

Fonte: Adaptada de Redmon e Farhadi (2018).

usuário fazer o *upload* da imagem, criando assim uma base colaborativa de imagens. Para tanto, utilizaram-se três tecnologias de computação em nuvem fornecidas pela Amazon Web Services (AWS), muito comuns no desenvolvimento de software atualmente, apresentadas a seguir:

- S3 (*Simple Storage Service*): trata-se de um serviço em nuvem para armazenamento de dados, com foco em escalabilidade e performance, podendo ser utilizado em uma ampla variedade de casos, tais como o armazenamento de cópias de bancos de dados, arquivos de *logs* e de aplicações *web* (tais como CSS e JavaScript), imagens, entre inúmeros outros. Para tanto, o serviço utiliza do conceito de *buckets*, que são uma maneira abstrata de definir contêineres onde os dados são armazenados. No protótipo desenvolvido neste trabalho, tal serviço foi empregado para armazenar as redes neurais treinadas e as imagens enviadas pelo usuário.

- API Gateway: é um serviço que permite a rápida criação, manutenção e monitoramento de APIs REST com escalabilidade; são baseadas em HTTP (implementando os métodos GET, POST, PUT, PATCH e DELETE) e habilitam a comunicação cliente-servidor, incluindo também o gerenciamento de tráfego e controles de acesso. Sua responsabilidade é expor um *endpoint* acessível para o aplicativo, permitindo que este envie a imagem na requisição HTTP; o serviço, então, redireciona as requisições que recebe para o serviço a seguir.

- Lambda: permite a execução de um código para uma dada aplicação, requerendo ao

programador apenas escrever a lógica desejada, sem a necessidade de definir e gerenciar a infra-estrutura necessária (como máquinas físicas e suas configurações de sistema operacional), que é alocada e escalada automaticamente. É possível, com esse serviço, construir pequenas funções que são disparadas a partir de eventos, podendo ser utilizadas, por exemplo, para responder a uma requisição HTTP, executando o código escrito pelo desenvolvedor da aplicação. No caso deste trabalho, tal serviço é responsável por carregar a rede neural treinada e alimentá-la com a imagem recebida na requisição, devolvendo então uma resposta ao API Gateway, que, por sua vez, envia-a de volta ao aplicativo.

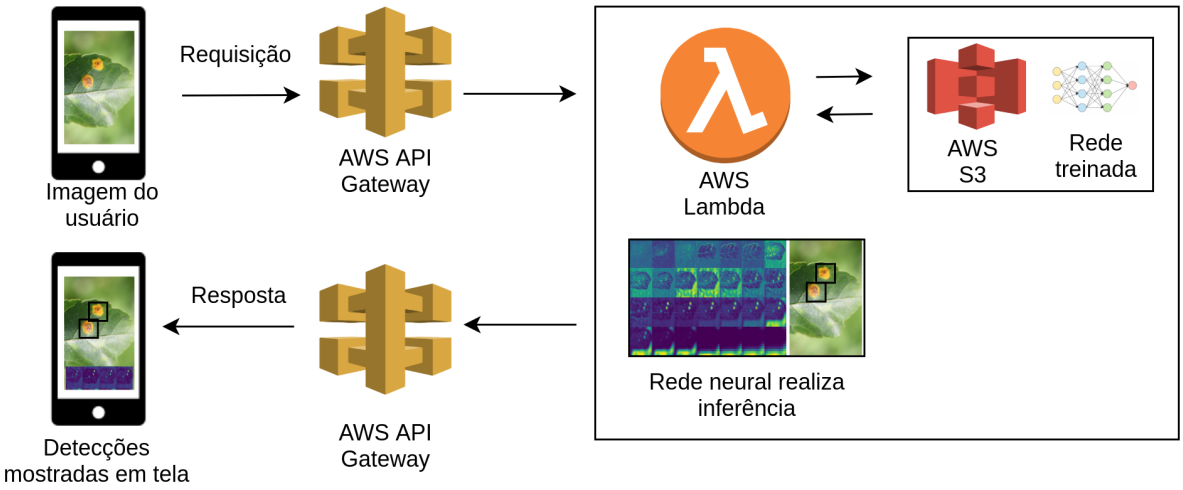


Figura 20 – Arquitetura do sistema. A camada da API Gateway atua como um intermediário entre o aplicativo e o serviço Lambda, o qual carrega a rede armazenada no AWS S3 para, então, realizar a inferência sobre a imagem, retornando as detecções e visualizações novamente ao aplicativo que enviou a requisição inicial.

Fonte: Elaborada pelo autor.

A Figura 20 apresenta a arquitetura final do sistema, indicando como esses serviços estão integrados. Através da interface do aplicativo, o usuário pode fornecer uma imagem da folha com doença, a qual é enviada via API, através da uma requisição HTTP (POST), ao serviço Lambda. Este, por sua vez, comunica-se ao S3 para recuperar a rede neural treinada e, então, processar a imagem recebida, obtendo as detecções e as visualizações das ativações para, finalmente, devolvê-las ao aplicativo através da mesma API. O aplicativo, então, renderiza em tela as detecções e as visualizações, exibindo os resultados ao usuário.

CAPÍTULO

5

RESULTADOS E DISCUSSÃO

5.1 Considerações iniciais

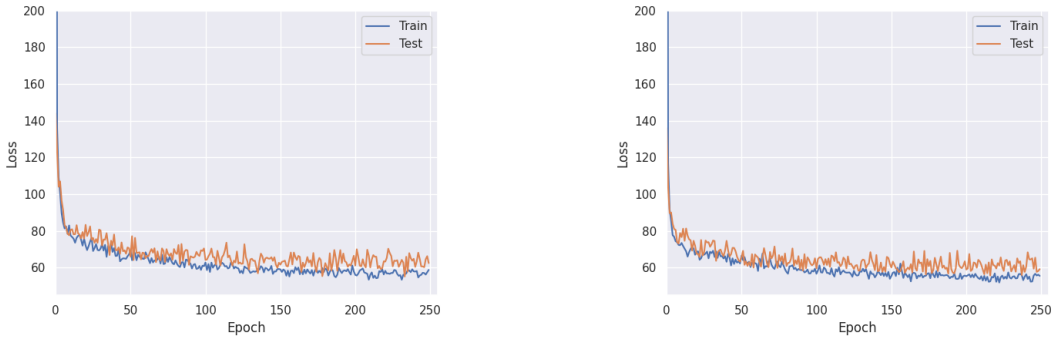
Neste capítulo são apresentados e discutidos os resultados obtidos neste projeto. Nas Seções 5.2 e 5.3 são apresentados e discutidos os desempenhos de detecção dos modelos, usando diferentes conjuntos de imagens de teste. Na Seção 5.4 são analisados os resultados de classificação, enquanto na Seção 5.5 são exibidas e discutidas as visualizações da ativações internas das redes. Por fim, o aplicativo desenvolvido é apresentado na Seção 5.6, onde são descritas suas funcionalidades e exibidas duas capturas de sua tela.

5.2 Comparação de resultados de modelos de detecção de objetos

O processo de treinamento e teste reportado nesta seção utilizou imagens do conjunto $\text{apple}_{pathology}$ e seguiu a estratégia 5-*fold* de validação cruzada, o que significa que foram realizadas 5 rodadas de treinamento, usando-se em cada uma delas, uma fração de 80% das imagens para treinamento e os 20% restantes para teste, de modo que em cada rodada o conjunto de teste continha amostras diferentes.

Os valores de função de custo (ou perda, *loss*) da rede, ao longo das 250 épocas de treinamento, para as redes Yolov3 de um e dois estágios, são apresentados na Figura 21. Como é possível verificar, ambos os modelos apresentam bons graus de ajuste, apresentando um rápido decaimento da função de custo nas primeiras épocas, havendo uma certa estabilização por volta da 200ª época, apesar de uma variação de baixa amplitude ocorrer ao longo de todo o treinamento. Embora os gráficos mostrem resultados para apenas um dos *folds*, os outros mostraram resultados semelhantes.

As curvas de precisão e *recall* e os valores de AP (conforme definido na Seção 4.3)

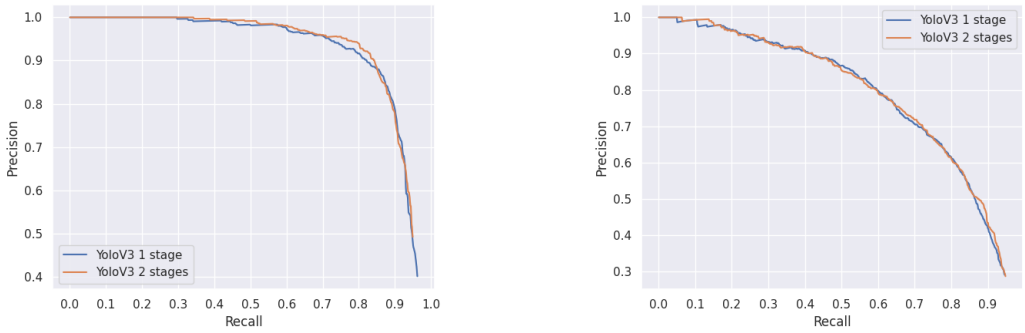


(a) (b)

Figura 21 – Curva da função de custo, para treinamento e teste, referindo-se 1 dos 5 *folds*, para as redes Yolov3 de um estágio (a) e dois estágios (b).

Fonte: Elaborada pelo autor.

foram calculados, para ambos os modelos e considerando as duas classes de doenças da maçã envolvidas (ferrugem e sarna). Conforme apresenta-se na Figura 22, bem como nas Tabelas 3 e 4, ambos os modelos apresentam desempenho semelhante, o que indica que, por enquanto, aparenta não haver diferença entre essas duas estratégias de detecção de objetos.



(a) (b)

Figura 22 – Curvas de precisão e *recall* em IoU=0.5, para as doenças ferrugem (a) e sarna (b), com os modelos de um e dois estágios.

Fonte: Elaborada pelo autor.

Além disso, verificou-se como os valores de mAP variam quando diferentes valores de limiar são definidos para IoU. Estabelecendo-se 10 valores para IoU no intervalo fechado de 0.5 a 0.95, com incrementos de 0.05, os correspondentes valores de mAP são exibidos na Figura 23. Conforme esperado, quanto mais altos são os valores desse limiar, isto é, quanto maior o rigor em relação à exata localização das caixas delimitadoras, menor é o valor de mAP.

Os resultados mostram que a classe da doença sarna da maçã é mais difícil de ser detectada do que a ferrugem; as detecções de caixas delimitadoras para duas imagens do conjunto de dados de teste são mostradas na Figura 24. Nota-se que as detecções da rede de

Fold	Precisão média (AP)		mAP
	Ferrugem	Sarna	
#0	0.8806	0.7318	0.8062
#1	0.9111	0.7673	0.8392
#2	0.9524	0.7515	0.8519
#3	0.9286	0.742	0.8353
#4	0.8978	0.6847	0.7913
Média ± d.p.	0.9141 ± 0.0277	0.7355 ± 0.0312	0.8248 ± 0.0251

Tabela 3 – Precisão média (AP) em IoU = 0.5, para o modelo Yolov3 de um estágio para as duas classes de doença.

Fold	Precisão média (AP)		mAP
	Ferrugem	Sarna	
#0	0.8875	0.7201	0.8037
#1	0.9093	0.7701	0.8397
#2	0.9435	0.7427	0.8431
#3	0.9216	0.7133	0.8174
#4	0.9093	0.6548	0.7820
Média ± d.p.	0.9142 ± 0.0204	0.7202 ± 0.0428	0.8172 ± 0.0255

Tabela 4 – Precisão média (AP) em IoU = 0.5, para o modelo Yolov3 de dois estágios para as duas doenças.

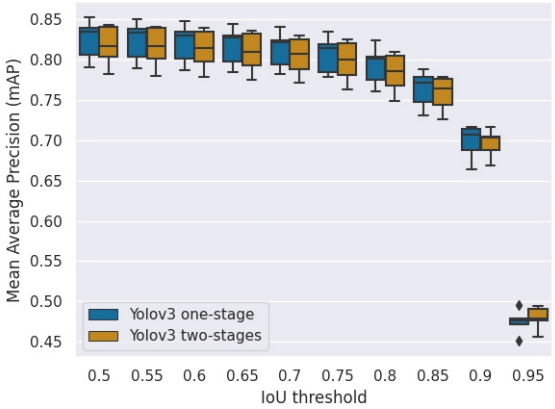
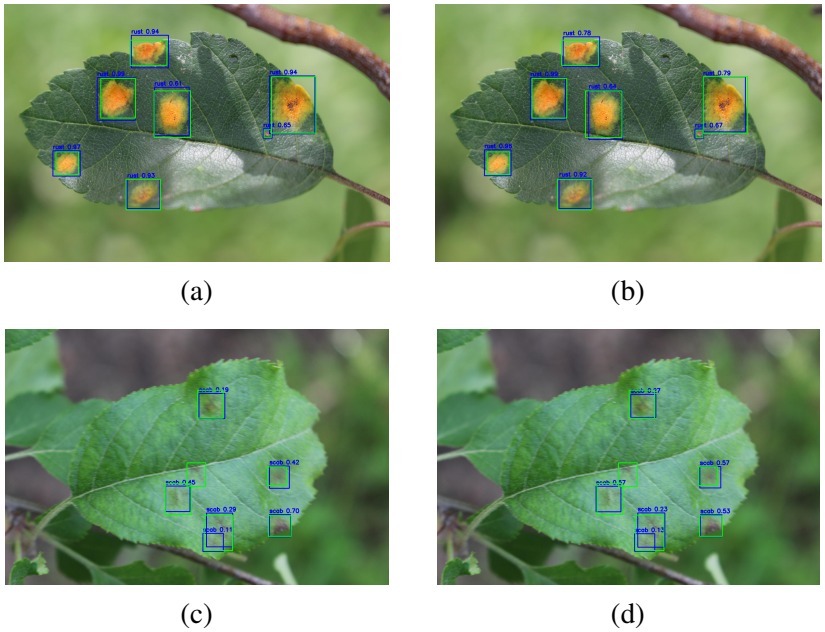


Figura 23 – Valores de mAP para diferentes limiares de IoU, variando de 0.5 a 0.95.

Fonte: Elaborada pelo autor.

um e dois estágios são semelhantes, com ambas detectando todas as regiões da imagem da classe ferrugem (Figuras 24a e 24b), porém não identificando uma região delimitadora no centro da imagem pertencente à classe sarna, como mostrado nas Figuras 24c e 24d. Como pode ser observado nessas imagens e, de modo geral, também nas demais imagens de teste do conjunto, sarna apresenta sintomas menos visualmente definidos em relação à ferrugem, daí sua detecção ter apresentado menor valor de precisão.



(a) (b)

(c) (d)

Figura 24 – Detecções das redes de um estágio (a, c) e dois estágios (b, d), para imagens de teste do conjunto apple$_{pathology}$, contendo as doenças ferrugem (a, b) e sarna (c, d). As detecções esperadas estão assinaladas em verde, enquanto as predições das redes estão em azul.

Fonte: Elaborada pelo autor.

5.3 Desempenho de detecção em um conjunto de imagens diferente

Os resultados reportados até agora exibem um desempenho de detecção semelhante para as redes de um e dois estágios quando utilizamos validação interna, i.e. as imagens oriundas de um mesmo conjunto de dados são divididas em treinamento e teste. Também é importante realizar validação externa, medindo o desempenho ao testar os modelos em imagens pertencentes a um conjunto de dados diferente. Como previamente mencionado na Seção 4.3, o conjunto de dados apple$_{unseen}$ foi selecionado e anotado, abrangendo 177 imagens afetadas pela doença da sarna.

Partindo das redes resultantes do treinamento apresentado na seção anterior, a etapa de teste foi realizada usando o conjunto de dados apple$_{unseen}$, a fim de analisar a capacidade de generalização das redes. Como cada rede fora treinada 5 vezes (uma vez para cada *fold*), temos 5

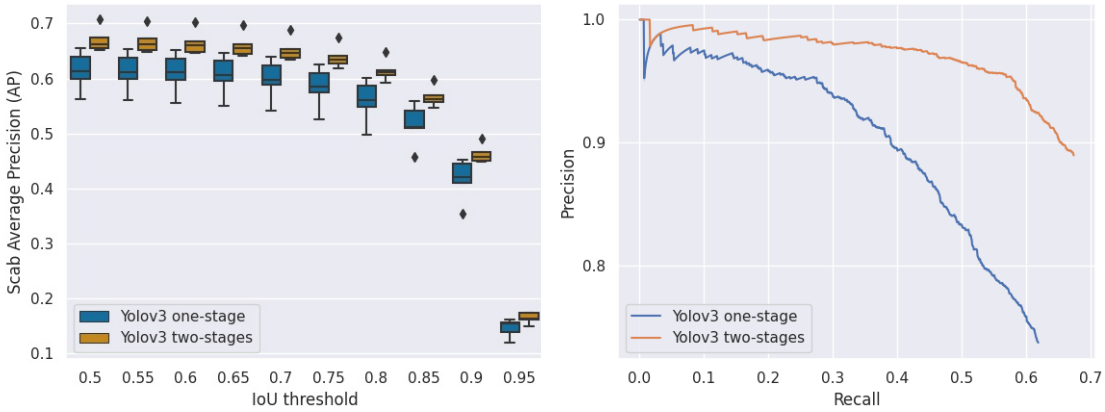
valores de AP para cada rede. A Tabela 5 apresenta a métrica AP para a classe sarna, usando 0.5 como limiar de IoU, para as redes de um e dois estágios. Desta vez, observa-se que esta última rede generalizou melhor, uma vez que foi capaz de alcançar AP significativamente maior do que a de um estágio. Exemplos de detecções, para ambas as redes, são mostrados na Figura 26.

Fold	**Precisão média (AP)**	
	Yolov3, um estágio	**Yolov3, dois estágios**
#0	0.6393	**0.7068**
#1	0.5617	**0.6546**
#2	0.6554	**0.6635**
#3	0.6135	**0.6508**
#4	0.5999	**0.6737**
Mean ± std	0.6140 ± 0.0363	**0.6700 ± 0.0224**

Tabela 5 – Precisão média (em IoU=0.5), para a doença sarna, no conjunto de dados apple$_{unseen}$, para os modelos de um e dois estágios da Yolov3. Desta vez, apenas a etapa de teste foi realizada, fazendo-se uso das redes anteriormente treinadas.

A diferença de desempenho de detecção dessas redes também foi observada ao considerar valores maiores para IoU, conforme ilustrado na Figura 25a. Nota-se, agora, que os *boxplots* das redes estão menos justapostos, com o da rede de dois estágios preservando-se acima do da rede de um estágio. Do mesmo modo, para o valor de IoU constante em 0.5, a curva de precisão e *recall* da rede de dois estágios mantém-se acima a curva da outra rede, conforme mostrado na Figura 25b, diferentemente da Figura 22 da seção anterior, quando ambas estavam praticamente sobrepostas. Além do mais, o menor desvio padrão da rede de dois estágios indica um desempenho de detecção mais regular.

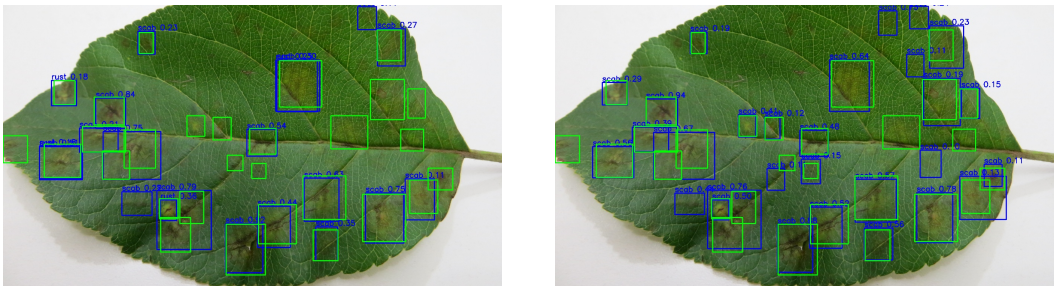
No entanto, ao comparar o desempenho médio de detecção das redes no conjunto de dados apple$_{unseen}$ e nas frações de teste do conjunto apple$_{pathology}$, constata-se que houve uma queda no desempenho em ambas as redes: o valor de AP para detecção de classe de sarna caiu de 0.7355 para 0.6140 (diferença de 0.1215) e de 0.7202 a 0.6700 (diferença de 0.0502), respectivamente, para redes de um e de dois estágios. Embora a diferença tenha sido menor no modelo de dois estágios, pode-se inferir que o conjunto de dados apple$_{pathology}$ não foi suficientemente abrangente para permitir a plena generalização por parte das redes com ele treinadas. De fato, trabalhos recentes apontam a falta de conjuntos de dados de imagens de plantas suficientemente extensos que permitam tal generalização (FERENTINOS, 2018; BARBEDO, 2019; BARBEDO, 2018a; LIU; WANG, 2021), em contraste, por exemplo, com o conjunto ImageNet (RUSSAKOVSKY *et al.*, 2015), o qual apresenta mais de 14 milhões de amostras. Ainda assim, o modelo de dois estágios apresentou maior capacidade de generalização, apesar dessa limitação do conjunto de dados.



(a) (b)

Figura 25 – Valores de AP para a classe sarna em função de diferentes valores de IoU (a), variando de 0.5 a 0.95, e curvas de precisão e *recall* para IoU=0.5 (b).

Fonte: Elaborada pelo autor.



(a) (b)

Figura 26 – Detecções dos modelos de um estágio (a) e dois estágios (b), para o conjunto apple_{unseen}. As detecções esperadas são mostradas em verde e as previstas, em azul.

Fonte: Elaborada pelo autor.

5.4 Resultados de classificação para o conjunto *Plant Village*

Investigou-se também a capacidade de generalização das redes, anteriormente treinadas, em uma tarefa puramente de classificação de imagens. Embora as redes utilizadas neste trabalho sejam voltadas à detecção de objetos, sua saída foi adaptada a fim de fornecer uma pontuação de classificação (tal como é feito em problemas de classificação de imagens), conforme definido na Seção 4.3.

Para tanto, foi utilizado um subconjunto das imagens do Plant Village (HUGHES; SALATHE, 2015), contendo apenas imagens de folhas de maçãs pertencentes a 3 classes: ferrugem, sarna e saudável. Tratam-se de imagens de menor resolução, com tamanho de 256×256 pixels (tal como originalmente presentes no conjunto de dados).

Para usar tais imagens sem dispor de anotações das caixas delimitadoras, realizaram-se as seguintes etapas: (1) metade das imagens (conjunto de treinamento) foram fornecidas às redes previamente treinadas, que executaram inferência e obtiveram suas previsões de caixas delimitadoras e classes; (2) essas redes foram retreinadas nessas mesmas imagens usando as próprias previsões geradas como rótulos de treinamento – uma técnica semelhante à pseudo-rotulagem (LEE, 2013). Então, testes foram realizados usando a outra metade das imagens, com 3 cenários sendo investigados:

I. Usando Plant Village, treinando as redes a partir do início: as redes de um e dois estágios foram treinadas do início (com a inicialização dos pesos a partir do *dataset* Pascal VOC) no conjunto de treinamento Plant Village (da mesma forma que foi feito na Seção 5.2), por 10 épocas, usando como rótulos as previsões obtidas após a execução de inferência nas imagens de treinamento, conforme o parágrafo anterior. Em seguida, o teste foi realizado a fim de produzir resultados de classificação de base.

II. Com a redes pré-treinadas no conjunto apple$_{pathology}$: foram usadas as redes treinadas da Seção 5.2, executando-se então a inferência nas imagens de teste do Plant Village, e, por fim, coletando-se os resultados da classificação.

III. Retreinadas a partir do conjunto apple$_{pathology}$ no Plant Village: as redes já treinadas da Seção 5.2 foram treinadas mais uma vez, executando-se um ajuste fino no conjunto de treinamento Plant Village; este processo de retreinamento também durou 10 épocas, para então serem obtidos os resultados da classificação nas imagens do teste do Plant Village.

Os resultados dos dois primeiros itens anteriores estão resumidos na Tabela 6. Como pode ser visto após o curto treinamento de 10 épocas a partir do início, ambas as redes tiveram dificuldades em classificar corretamente as imagens pertencentes à classe sarna – a rede de 1 estágio obteve *recall* 0, enquanto a rede de dois estágios apenas um valor um pouco maior, mas ambos ínfimos. Pode-se notar que, neste caso, ambas concentraram seus acertos na classe majoritária (saudável); apesar disso, em geral, a rede de dois estágios manteve valores mais elevados de precisão, *recall* e *F1-score*, quando comparada à rede de um estágio.

Também é possível observar melhores resultados, para ambos os modelos, ao fazer a previsão a partir do treinamento com o conjunto de dados apple$_{pathology}$, indicando que o aprendizado por transferência funcionou melhor do que um treinamento a partir do início; os resultados também mostram uma grande diferença de *F1-score* para classe sarna entre essas duas situações, em ambas as redes. Além disso, verifica-se um melhor desempenho geral de classificação da rede de dois estágios também nesta segunda situação.

Após o ajuste fino das redes, os resultados obtidos são mostrados na Tabela 7. Nota-se que foi obtida uma melhoria, para ambas as redes, uma vez que as métricas de classificação foram em geral superiores. A variação de cada métrica não foi tão grande quanto a variação

Métrica	Classe/ Média (avg)	Do início		Treinada com apple$_{pathology}$	
		Yolov3 um estágio	**Yolov3 dois estágios**	**Yolov3 um estágio**	**Yolov3 dois estágios**
P	saudável	0.6639	**0.7127**	0.7597	**0.8394**
	ferrugem	**0.8846**	0.8080	**0.9000**	0.7376
	sarna	0.0000	**0.6667**	0.7035	**0.7961**
	avg macro	0.5162	**0.7291**	0.7877	**0.7910**
	avg ponderada	0.5207	**0.7116**	0.7610	**0.8172**
R	saudável	**1.0000**	0.9988	0.9384	**0.9594**
	ferrugem	0.3262	**0.7163**	0.6383	**0.7376**
	sarna	0.0000	**0.0248**	0.3758	**0.5093**
	avg macro	0.4421	**0.5800**	0.6508	**0.7354**
	avg ponderada	0.6729	**0.7216**	0.7631	**0.8212**
F1	saudável	0.7980	**0.8318**	0.8397	**0.8954**
	ferrugem	0.4767	**0.7594**	**0.7469**	0.7376
	sarna	0.0000	**0.0479**	0.4899	**0.6212**
	avg macro	0.4249	**0.5464**	0.6921	**0.7514**
	avg ponderada	0.5610	**0.6258**	0.7411	**0.8087**

Tabela 6 – Desempenho de classificação de redes no conjunto de dados de teste do Plant Village (P: precisão, R: *recall*, F1: *f1-score*), para as redes treinadas a partir do início e a partir do um treinamento prévio com as imagens do conjunto apple$_{pathology}$.

observada na Tabela 6, o que é esperado, visto que a maior parte do conhecimento aprendido pelas redes fora obtido no treinamento inicial com conjunto apple$_{pathology}$, que durou mais épocas – o ajuste fino atuou como um último aprimoramento. Considerando a rede de um estágio, a média macro (não ponderada) do *F1-score* melhorou de 0.4249 para 0.6921 (Tabela 6) e, em seguida, para 0.7948 (Tabela 7). Correspondentemente, para a rede de dois estágios, a média macro do *F1-score* melhorou de 0.5464 para 0.7514 e, finalmente, para 0.8304 após o ajuste fino – superando a rede de um estágio não apenas nessa métrica, mas praticamente em todas as outras. Isso mostra o potencial dessa rede ao ser empregada para gerar rótulos para dados futuros, sem a necessidade de anotação de caixas delimitadoras, e alcançar melhorias significativas na tarefa de classificação a partir deles.

5.5 Visualização dos mapas de atributos

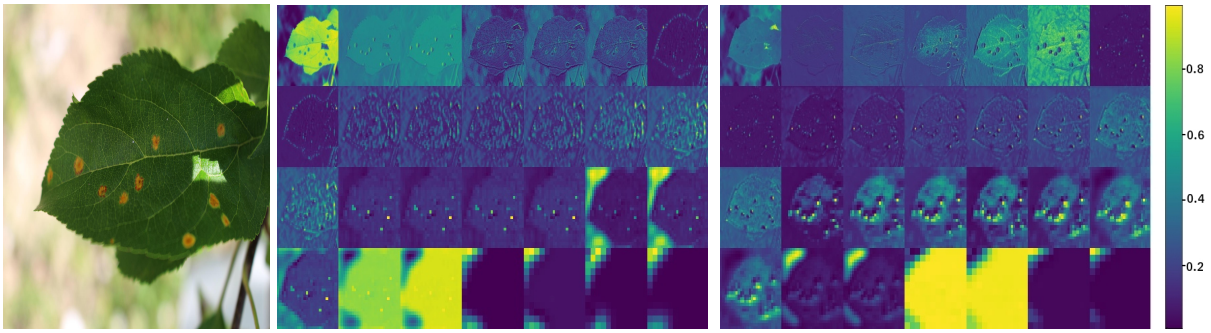
Visando interpretar os atributos que as redes aprenderam a fim de distinguir as áreas doentes nas imagens e classificar as doenças, foram obtidos e visualizados os valores dos mapas de atributos internos da redes para algumas imagens de teste, conforme descrito na Seção 4.6 e, especialmente, na Figura 19. A visualização das maiores ativações ao longo das camadas da rede Yolov3 dá dicas sobre o que está sendo utilizado para realizar o processo de detecção das regiões afetadas.

Métrica	Classe/ Média(avg)	Treinada com apple$_{pathology}$ e Plant Village (ajuste fino com pseudo-rotulagem)	
		Yolov3 um estágio	Yolov3 dois estágios
P	healthy	0.8575	**0.9211**
	rust	0.8170	**0.8579**
	scab	0.7297	**0.7863**
	macro avg	0.8014	**0.8551**
	weighted avg	0.8207	**0.8810**
R	healthy	0.9113	**0.9495**
	rust	**0.8865**	0.8864
	scab	0.5870	**0.6056**
	macro avg	0.7949	**0.8139**
	weighted avg	0.8267	**0.8557**
F1	healthy	0.8836	**0.9351**
	rust	0.8503	**0.8719**
	scab	0.6506	**0.6842**
	macro avg	0.7948	**0.8304**
	weighted avg	0.8211	**0.8647**

Tabela 7 – Desempenho de classificação de redes treinadas (ajuste fino) no conjunto de teste Plant Village.

A Figura 27 exibe essas visualizações para uma imagem de teste do conjunto de dados de apple$_{pathology}$(Figura 27a), contendo sintomas da doença ferrugem. Podem ser observados, nas Figuras 27b e 27c, os 28 mapas de atributos extraídos ao longo dos blocos de camadas convolucionais da rede, tomando, para cada bloco, o mapa com os maiores valores (maiores ativações da rede), conforme definido na Seção 4.6. As visualizações mostram que os sintomas visuais são nitidamente distinguíveis do resto da folha, mesmo nos primeiros mapas de atributos (no topo de cada figura); a folha também é bem diferenciada do fundo. Porém, os últimos mapas de atributos (na porção inferior de cada figura), por terem uma resolução baixa, tenderam a apresentar visualizações mais homogêneas, exibindo menos nitidamente as ativações nas áreas dos sintomas, o que sugere que se as redes dispusessem de algumas camadas a menos, provavelmente produziriam resultados semelhantes, uma vez que as ativações referentes às áreas da doença não estão claramente presentes nessas últimas camadas.

As visualizações mostradas na Figura 28 referem-se a uma imagem de teste contendo uma folha afetada pela sarna, também retirada do conjunto apple$_{pathology}$. É possível perceber que, ao contrário do exemplo anterior, neste caso as bordas dos sintomas não estão claramente definidas, sendo mais difíceis de distinguir. Os primeiros mapas de atributos não foram capazes de discriminar as regiões de doença; apenas mapas intermediários apresentam ativações que parecem referir-se às áreas da doença e, mesmo assim, as distinções não são tão claras como na visualização anterior. Isso pode explicar a razão pela qual a precisão média da classe sarna foi menor, conforme relatado na Seção 5.2: quanto menos nítidas as manifestações visuais da

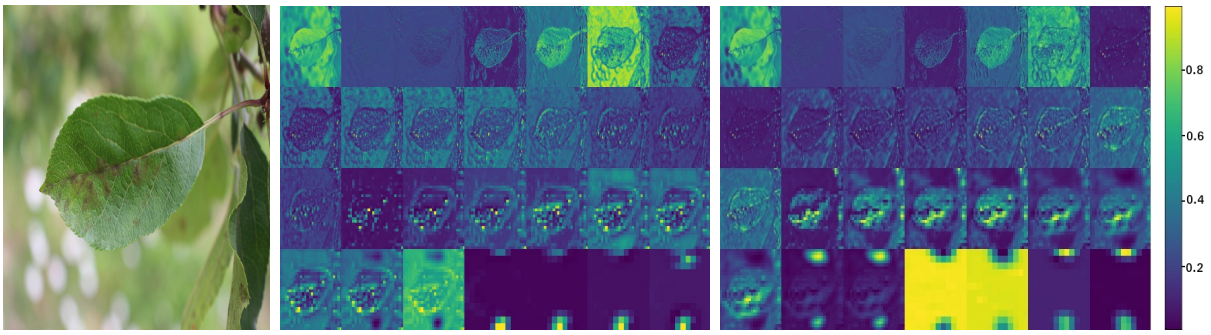


(a) (b) (c)

Figura 27 – Imagem de folha do conjunto apple$_{pathology}$(a) e mapas de atributos máximos para as redes de um estágio (b) e dois estágios (c), em sequência a partir do primeiro mapa (canto superior esquerdo) até o último (canto inferior direito), em que a cor amarela significa o maior valor de ativação.

Fonte: Elaborada pelo autor.

doença, mais difícil será sua distinção.



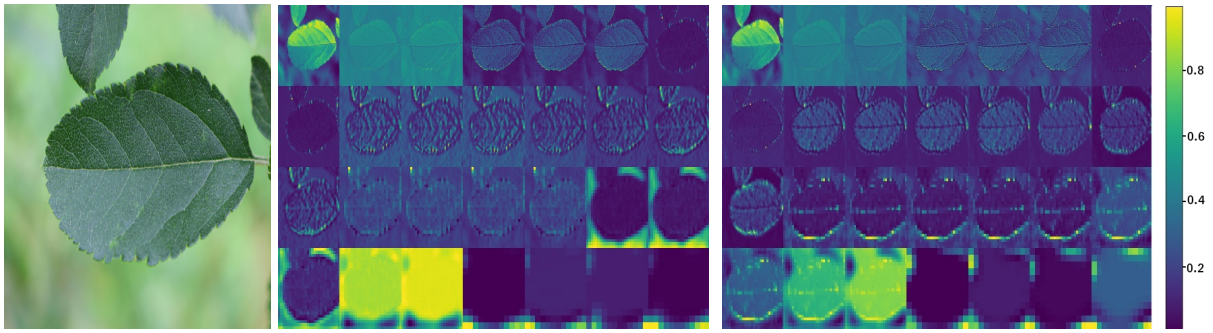
(a) (b) (c)

Figura 28 – Imagem de folha do conjunto apple$_{pathology}$(a) e mapas de atributos máximos para as redes de um estágio (b) e dois estágios (c).

Fonte: Elaborada pelo autor.

Uma amostra de folha saudável também teve suas visualizações obtidas e são mostradas na Figura 29. Como esperado para uma folha saudável, os mapas de atributos são mais homogêneos e não exibem regiões de ativação contrastantes. Como nos exemplos anteriores, quase todas as visualizações dos mapas mostram uma notável distinção entre a folha e o fundo da imagem, embora as redes não tenham sido explicitamente treinadas para tal propósito.

Por fim, a Figura 30 apresenta as visualizações para uma imagem do conjunto apple$_{unseen}$ afetada por sarna. Os mapas de atributos exibem mais ativações para a rede de dois estágios (note, por exemplo, as terceiras linhas dos mapas nas Figuras 30b e 30c), coincidindo com pontos de doença na imagem original (Figura 30a). Trata-se de uma explicação para o melhor desempenho observado pela rede de dois estágios nos testes que utilizaram o referido conjunto de imagens (Seção 5.3). Como nos exemplos anteriores, os mapas de resolução mais baixa fornecem menos

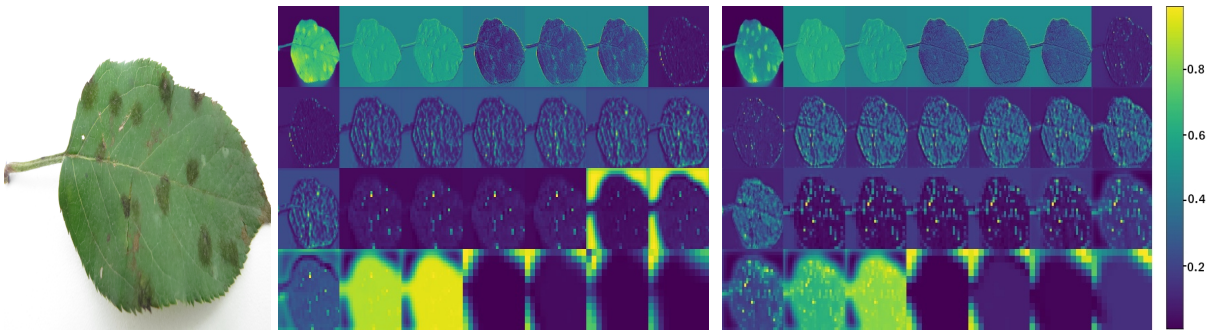


(a) (b) (c)

Figura 29 – Imagem de folha saudável do conjunto de apple$_{pathology}$ (a) e mapas de atributos máximos para redes de um estágio (b) e dois estágios (c).

Fonte: Elaborada pelo autor.

informações sobre a doença e estão mais relacionados à distinção entre a folha e o fundo.



(a) (b) (c)

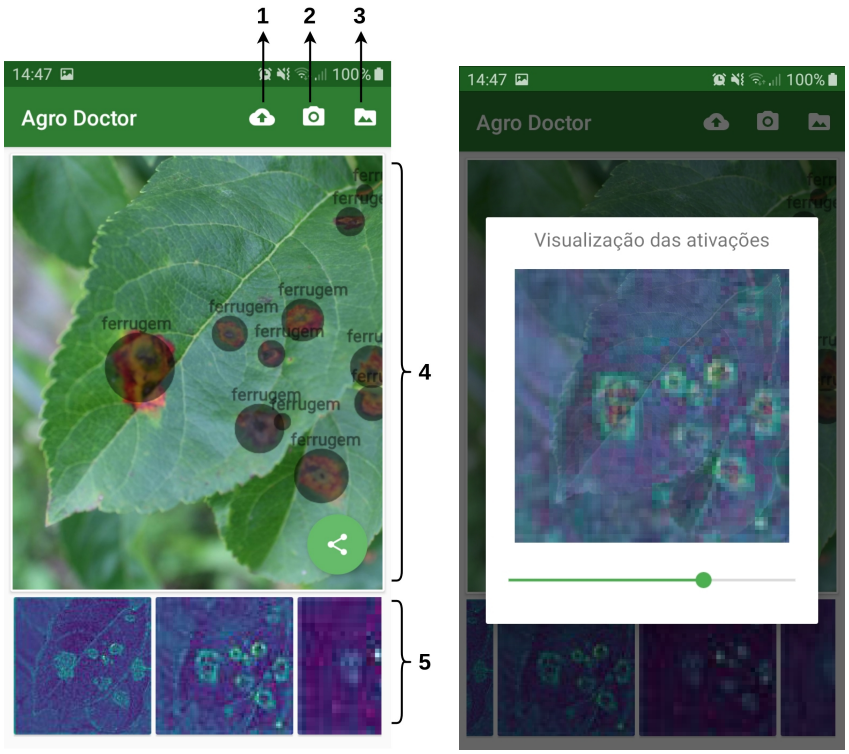
Figura 30 – Imagem de folha do conjunto apple$_{unseen}$, acometida por sarna(a) e mapas de atributos com os maiores valores, para redes de um (b) e dois estágios(c).

Fonte: Elaborada pelo autor.

5.6 Aplicativo

O aplicativo foi desenvolvido e duas capturas de tela são apresentadas na Figura 31. Por ser um protótipo, sua interface oferece simplicidade e consiste de uma única tela, cujos elementos presentes estão indicadas na Figura 31a. As finalidades das regiões indicadas pelos números de 1 a 5 são informadas a seguir:

1. Botão que permite ao usuário fazer o *upload* da imagem em um *bucket* do serviço S3 responsável por armazenas as imagens.
2. Botão que abre a câmera do dispositivo.
3. É possível acessar a galeria de fotos presentes no dispositivo através deste botão.



(a) (b)

Figura 31 – Capturas de tela do aplicativo, mostrando os elementos de sua composição (a) e uma janela flutuante (b), que exibe, em uma área maior, o mapa de atributos selecionado.

Fonte: Elaborada pelo autor.

4. Região que exibe a imagem enviada pelo usuário, sobre a qual se renderizam as detecções obtidas pela rede.

5. Região em que são apresentadas as visualizações dos mapas de atributos da rede, para a imagem em questão. Através de rolagem horizontal, é possível visualizar os demais mapas.

O usuário, após enviar a imagem, normalmente recebe a resposta da API em menos de 10 segundos. Nesse intervalo de tempo, o aplicativo faz a requisição à API, os dados trafegam através da internet, então a rede é carregada no serviço Lambda, processa a imagem e a API retorna a requisição ao aplicativo, contendo as detecções e visualizações. Nota-se que a maior parte desse período de tempo se deve ao processamento da imagem através da rede neural, uma vez que esta opera em ambiente de CPU. É possível diminuir esse intervalo de tempo ao se realizar o processamento em GPU, bem como utilizar tamanhos menores de imagem.

Ao clicar em uma das visualizações presentes na região 5, abre-se uma janela flutuante como se mostra na Figura 31b, que permite a visualização mais detalhada do mapa que escolhido, projetado sobre a imagem original. Assim, através de uma barra de ajuste (mostrada logo abaixo da visualização, na Figura 31b), é possível controlar a transparência do mapa sobre a

imagem: com a barra vazia, observa-se somente a imagem original; porém, com a barra completa, visualiza-se somente o mapa – e qualquer posição intermediária mostra uma fusão entre o mapa e a imagem original, gerando um efeito de transição.

A possibilidade oferecida pelo aplicativo de se fazer o *upload* da imagem a um repositório de armazenamento remoto permite que seja criada uma base colaborativa de dados, visando a criação, ao longo do tempo, de um conjunto mais abrangente de imagens de doenças de plantas, já que a ausência de tais dados ainda representa uma dificuldade em se conseguir um aprendizado mais generalizado por parte das redes neurais, conforme apontado na Seção 5.3.

5.7 Considerações Finais

Ao longo das seções deste capítulo, foram apresentados e discutidos os resultados alcançados neste trabalho, abrangendo os desempenho das redes neurais em cada conjunto de imagens utilizado, bem como as visualizações dos mapas de atributos internos e, por fim, as funcionalidades do aplicativo implementado.

CAPÍTULO

6

CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho, foi proposto um método para detecção e classificação de duas doenças em imagens de folhas de macieira – ferrugem e sarna –, baseado na combinação de duas redes convolucionais independentes, uma realizando a detecção de caixas delimitadoras (*bounding boxes*) que circunscrevem as áreas com sintomas de tais doenças nas folhas, e a outra realizando a classificação de cada uma dessas áreas delimitadas, fornecendo a classe da doença.

Ao comparar o método proposto com a típica detecção e classificação conjunta de objetos, realizada em um único estágio, os resultados mostraram que o desempenho de ambas as estratégias tende a ser equivalente se houver baixas variações intraclasse – por exemplo, quando os dados de treinamento e teste pertencem ao mesmo conjunto de dados, coletados no mesmo local e sujeitos às mesmas condições de captura. Contudo, ao introduzir dados externos para teste, obtidos em diferentes circunstâncias, adicionando assim mais variabilidade intraclasse, a abordagem de dois estágios apresentou melhores métricas de desempenho.

Além disso, foi apresentada uma estratégia para estender o uso de uma rede de detecção de objetos a uma tarefa de classificação, adaptando sua saída e, usando uma técnica de pseudo-rotulagem e executando um rápido treinamento de ajuste fino com novas amostras de imagens, provenientes de outros conjuntos de dados, pode-se notar um melhor desempenho de classificação dessa abordagem de dois estágios.

Também foi realizada a visualização de mapas de atributos gerados pelas redes, mostrando que as redes utilizadas foram capazes de disparar maiores ativações em pontos correspondentes às regiões de doenças nas imagens originais, além de proporcionar separação entre a folha e elementos irrelevantes da imagem. A visualização dos mapas também refletiu as diferenças de desempenho das redes ao serem testadas com um conjunto de imagens diferente do qual haviam sido treinadas.

Por fim, implementou-se um aplicativo para *smartphones* e *tablets* com sistema Android que, fazendo uso de serviços em nuvem para o processamento da rede neural, torna possível

um usuário enviar uma imagem de folha de maçã a tal rede, para que seja feito o processo de detecção e classificação das regiões de doença, exibindo os resultados e as visualizações dos mapas de atributos em tela, de maneira rápida e, neste último caso, também interativa. Além disso, o aplicativo fornece a possibilidade de formação de uma base colaborativa de imagens, com o objetivo de permitir a criação e manutenção de um conjunto de imagens mais abrangente do que os existentes.

Em conclusão, a abordagem de dois estágios para detecção e classificação de doenças de plantas tendeu a melhorar precisão média, sendo também possível realizar uma interpretação visual das características aprendidas ao gerar visualizações dos mapas de atributos internos à rede; além disso, o modelo transferiu melhor o conhecimento aprendido a novos conjuntos de dados, quando comparado com a rede de um estágio ou com o treinamento realizado a partir do zero, em tarefas tanto de detecção de objetos como de simples classificação de imagens.

Trabalhos futuros podem se concentrar no uso de um número maior de doenças e de espécies de plantas consideradas, avaliando-se o desempenho da rede por meio do treinamento e teste em diferentes espécies. Além disso, maneiras distintas de classificar as caixas delimitadoras podem ser usadas, por exemplo, permitindo imagens de tamanhos diferentes, e não pré-estabelecidos. Possíveis informações não visuais, como aquelas coletadas por sensores de umidade e temperatura, bem como as características do solo e dados históricos e de sazonalidade, por exemplo, podem ser agregados aos atributos da imagem apreendidos pelas redes, a fim de melhorar seu desempenho de teste em novos dados.

REFERÊNCIAS

ABADI, M.; BARHAM, P.; CHEN, J.; CHEN, Z.; DAVIS, A.; DEAN, J.; DEVIN, M.; GHEMAWAT, S.; IRVING, G.; ISARD, M.; others. Tensorflow: A system for large-scale machine learning. In: **12th Symposium on Operating Systems Design and Implementation (OSDI 16)**. [S.l.: s.n.], 2016. p. 265–283. Citado na página 44.

AMARA, J.; BOUAZIZ, B.; ALGERGAWY, A. A deep learning-based approach for banana leaf diseases classification. In: **Lecture Notes in Informatics (LNI), Proceedings - Series of the Gesellschaft fur Informatik (GI)**. [S.l.: s.n.], 2017. v. 266, p. 79–88. ISBN 9783885796602. ISSN 16175468. Citado nas páginas 34 e 38.

BARBEDO, J. G. A. Digital image processing techniques for detecting, quantifying and classifying plant diseases. **SpringerPlus**, v. 2, n. 1, p. 1–12, 2013. ISSN 21931801. Citado nas páginas 19 e 33.

____. Factors influencing the use of deep learning for plant disease recognition. **Biosystems Engineering**, Academic Press, v. 172, p. 84–91, 8 2018. ISSN 15375110. Citado nas páginas 37 e 53.

____. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. **Computers and Electronics in Agriculture**, Elsevier B.V., v. 153, p. 46–53, 10 2018. ISSN 01681699. Citado na página 34.

____. Plant disease identification from individual lesions and spots using deep learning. **Biosystems Engineering**, v. 180, p. 96–107, 2019. ISSN 15375110. Citado nas páginas 35, 38, 41 e 53.

BARBEDO, J. G. A.; KOENIGKAN, L. V.; HALFELD-VIEIRA, B. A.; COSTA, R. V.; NECHET, K. L.; GODOY, C. V.; JUNIOR, M. L.; PATRICIO, F. R. A.; TALAMINI, V.; CHITARRA, L. G.; OLIVEIRA, S. A. S.; ISHIDA, A. K. N.; FERNANDES, J. M. C.; SANTOS, T. T.; CAVALCANTI, F. R.; TERAO, D.; ANGELOTTI, F. Annotated plant pathology databases for image-based detection and recognition of diseases. **IEEE Latin America Transactions**, IEEE Computer Society, v. 16, n. 6, p. 1749–1757, 6 2018. ISSN 15480992. Citado na página 19.

BOCK, C. H.; POOLE, G. H.; PARKER, P. E.; GOTTWALD, T. R. Plant Disease Severity Estimated Visually, by Digital Photography and Image Analysis, and by Hyperspectral Imaging. **Critical Reviews in Plant Sciences**, v. 29, n. 2, p. 59–107, 2010. ISSN 07352689. Citado na página 19.

BOULENT, J.; FOUCHER, S.; THÉAU, J.; ST-CHARLES, P. L. **Convolutional Neural Networks for the Automatic Identification of Plant Diseases**. [S.l.]: Frontiers Media S.A., 2019. Citado na página 19.

CHOLLET, F.; others. **Keras**. 2015. Https://keras.io. Disponível em: <https://keras.io>. Citado na página 44.

DECHANT, C.; WIESNER-HANKS, T.; CHEN, S.; STEWART, E. L.; YOSINSKI, J.; GORE, M. A.; NELSON, R. J.; LIPSON, H. Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning. **Phytopathology**, v. 107, n. 11, p. 1426–1432, 2017. ISSN 0031949X. Disponível em: <https://doi.org/10.1094/PHYTO-11-16-0417-R>. Citado nas páginas 35 e 38.

EVERINGHAM, M.; GOOL, L. V.; WILLIAMS, C. K.; WINN, J.; ZISSERMAN, A. The pascal visual object classes (VOC) challenge. **International Journal of Computer Vision**, v. 88, n. 2, p. 303–338, 2010. ISSN 09205691. Citado nas páginas 42 e 45.

FERENTINOS, K. P. Deep learning models for plant disease detection and diagnosis. **Computers and Electronics in Agriculture**, Elsevier B.V., v. 145, p. 311–318, 2 2018. ISSN 01681699. Citado nas páginas 34, 38 e 53.

FUENTES, A.; YOON, S.; KIM, S. C.; PARK, D. S. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. **Sensors (Switzerland)**, MDPI AG, v. 17, n. 9, 9 2017. ISSN 14248220. Citado nas páginas 36 e 38.

FUENTES, A. F.; YOON, S.; LEE, J.; PARK, D. S. High-performance deep neural network-based tomato plant diseases and pests diagnosis system with refinement filter bank. **Frontiers in Plant Science**, Frontiers Media S.A., v. 9, 8 2018. ISSN 1664462X. Citado nas páginas 20, 36, 38 e 39.

GIRSHICK, R. Fast r-cnn. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2015. p. 1440–1448. Citado nas páginas 28 e 32.

GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; MALIK, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2014. p. 580–587. Citado nas páginas 27 e 28.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. Citado na página 19.

GUO, J.; HE, T.; LAUSEN, L.; LI, M.; LIN, H.; SHI, X.; WANG, C.; XIE, J.; ZHA, S.; ZHANG, A.; ZHANG, H.; ZHANG, Z. Z.; ZHANG, Z. Z.; ZHENG, S.; ZHU, Y.; HE, H.; HE, T.; LAUSEN, L.; LI, M.; LIN, H.; SHI, X.; WANG, C.; XIE, J.; ZHA, S.; ZHANG, A.; ZHANG, H.; ZHANG, Z. Z.; ZHANG, Z. Z.; ZHENG, S.; ZHU, Y. **GluonCV and GluonNLP: Deep Learning in Computer Vision and Natural Language Processing**. [S.l.], 2020. v. 21, 1–7 p. Disponível em: <http://arxiv.org/abs/1907.04433http://jmlr.org/papers/v21/19-429.html.> Citado na página 44.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep Residual Learning for Image Recognition. **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, IEEE, 6 2016. Disponível em: <http://dx.doi.org/10.1109/CVPR.2016.90>. Citado nas páginas 24, 25 e 29.

HINTON, G. E.; OSINDERO, S.; TEH, Y.-W. A fast learning algorithm for deep belief nets. **Neural computation**, MIT Press, v. 18, n. 7, p. 1527–1554, 2006. Citado na página 24.

HOWARD, A. G.; ZHU, M.; CHEN, B.; KALENICHENKO, D.; WANG, W.; WEYAND, T.; ANDREETTO, M.; ADAM, H. **MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications**. 2017. Citado nas páginas 25 e 29.

HUANG, G.; LIU, Z.; MAATEN, L. V. D.; WEINBERGER, K. Q. Densely connected convolutional networks. In: **Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017**. [s.n.], 2017. v. 2017-Janua, p. 2261–2269. ISBN 9781538604571. Disponível em: <http://arxiv.org/abs/1608.06993https://github.com/liuzhuang13/DenseNet.> Citado na página 25.

HUBEL, D. H.; WIESEL, T. N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. **The Journal of Physiology**, J Physiol, v. 160, p. 106–154, 1 1962. ISSN 14697793. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/14449617/>. Citado na página 24.

HUGHES, D. P.; SALATHE, M. An open access repository of images on plant health to enable the development of mobile disease diagnostics. 11 2015. Disponível em: <http://arxiv.org/abs/1511.08060>. Citado nas páginas 34 e 54.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: **32nd International Conference on Machine Learning, ICML 2015**. International Machine Learning Society (IMLS), 2015. v. 1, p. 448–456. ISBN 9781510810587. Disponível em: <https://arxiv.org/abs/1502.03167v3>. Citado na página 30.

JIANG, P.; CHEN, Y.; LIU, B.; HE, D.; LIANG, C. Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks. **IEEE Access**, Institute of Electrical and Electronics Engineers Inc., v. 7, p. 59069–59080, 2019. ISSN 21693536. Citado na página 20.

KAMILARIS, A.; PRENAFETA-BOLDÚ, F. X. **Deep learning in agriculture: A survey**. [S.l.]: Elsevier B.V., 2018. 70–90 p. Citado nas páginas 20 e 33.

KC, K.; YIN, Z.; WU, M.; WU, Z. Depthwise separable convolution architectures for plant disease classification. **Computers and Electronics in Agriculture**, Elsevier BV, v. 165, p. 104948, 10 2019. ISSN 01681699. Citado nas páginas 34 e 38.

KINGMA, D. P.; BA, J. Adam: A Method for Stochastic Optimization. 2014. Disponível em: <http://arxiv.org/abs/1412.6980>. Citado na página 45.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet classification with deep convolutional neural networks. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2012. v. 2, p. 1097–1105. ISBN 9781627480031. ISSN 10495258. Citado nas páginas 19, 24 e 25.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, Nature Publishing Group, v. 521, n. 7553, p. 436, 2015. Citado na página 33.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P.; BOTTOU, L. E.; BENGIO, Y.; ABSTRACT|, P. H. Gradient-Based Learning Applied to Document Recognition. **Proceedings of the IEEE**, v. 86, p. 2278–2324, 1998. Citado nas páginas 24, 25 e 26.

LEE, D.-H. Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks. In: **ICML 2013 Workshop : Challenges in Representation Learning (WREPL)**. [S.l.: s.n.], 2013. Citado na página 55.

LIU, B.; ZHANG, Y.; HE, D. J.; LI, Y. Identification of apple leaf diseases based on deep convolutional neural networks. **Symmetry**, v. 10, n. 1, p. 11, 12 2018. ISSN 20738994. Disponível em: <http://www.mdpi.com/2073-8994/10/1/11>. Citado nas páginas 35 e 38.

LIU, J.; WANG, X. Plant diseases and pests detection based on deep learning: a review. **Plant Methods**, BioMed Central, v. 17, n. 1, p. 22, 12 2021. ISSN 1746-4811. Citado na página 53.

LIU, W.; ANGUELOV, D.; ERHAN, D.; SZEGEDY, C.; REED, S.; FU, C.-Y.; BERG, A. C. SSD: Single shot multibox detector. In: SPRINGER. **European conference on computer vision**. 2016. p. 21–37. Disponível em: <http://arxiv.org/abs/1512.02325http://dx.doi.org/10.1007/978-3-319-46448-0_2>. Citado na página 32.

LU, Y.; YI, S.; ZENG, N.; LIU, Y.; ZHANG, Y. Identification of rice diseases using deep convolutional neural networks. **Neurocomputing**, Elsevier B.V., v. 267, p. 378–384, 12 2017. ISSN 18728286. Citado na página 38.

MAO, Q. C.; SUN, H. M.; LIU, Y. B.; JIA, R. S. Mini-YOLOv3: Real-Time Object Detector for Embedded Applications. **IEEE Access**, v. 7, p. 133529–133538, 2019. ISSN 21693536. Citado na página 31.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, v. 5, n. 4, p. 115–133, 1943. Citado na página 23.

MELLO, R. F.; PONTI, M. A. **Machine learning: a practical approach on the statistical learning theory**. [S.l.]: Springer, 2018. Citado na página 28.

MOHANTY, S. P.; HUGHES, D. P.; SALATHÉ, M. Using deep learning for image-based plant disease detection. **Frontiers in Plant Science**, Frontiers Research Foundation, v. 7, n. September, 9 2016. ISSN 1664462X. Citado nas páginas 34 e 38.

NACHTIGALL, L. G.; ARAUJO, R. M.; NACHTIGALL, G. R. Classification of Apple Tree Disorders Using Convolutional Neural Networks. In: . [S.l.]: Institute of Electrical and Electronics Engineers (IEEE), 2017. p. 472–476. Citado na página 41.

NGUGI, L. C.; ABELWAHAB, M.; ABO-ZAHHAD, M. **Recent advances in image processing techniques for automated leaf pest and disease recognition – A review**. [S.l.]: China Agricultural University, 2020. Citado nas páginas 19 e 33.

PADILLA, R.; NETTO, S. L.; SILVA, E. A. D. A Survey on Performance Metrics for Object-Detection Algorithms. In: **International Conference on Systems, Signals, and Image Processing**. [S.l.]: IEEE Computer Society, 2020. v. 2020-July, p. 237–242. ISBN 9781728175393. ISSN 21578702. Citado nas páginas 42 e 43.

PONTI, M. A.; RIBEIRO, L. S. F.; NAZARE, T. S.; BUI, T.; COLLOMOSSE, J. Everything you wanted to know about deep learning for computer vision but were afraid to ask. In: IEEE. **2017 30th SIBGRAPI conference on graphics, patterns and images tutorials (SIBGRAPI-T)**. [S.l.], 2017. p. 17–41. Citado nas páginas 19, 24 e 26.

RAMCHARAN, A.; MCCLOSKEY, P.; BARANOWSKI, K.; MBILINYI, N.; MRISHO, L.; NDALAHWA, M.; LEGG, J.; HUGHES, D. P. A mobile-based deep learning model for cassava disease diagnosis. **Frontiers in Plant Science**, Frontiers Media S.A., v. 10, 3 2019. ISSN 1664462X. Citado nas páginas 35 e 38.

REDMON, J.; DIVVALA, S.; GIRSHICK, R.; FARHADI, A. You only look once: Unified, real-time object detection. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.]: IEEE Computer Society, 2016. v. 2016-Decem, p. 779–788. ISBN 9781467388504. ISSN 10636919. Citado nas páginas 27 e 30.

REDMON, J.; FARHADI, A. YOLO9000: Better, faster, stronger. In: **Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017**. Institute of Electrical and Electronics Engineers Inc., 2017. v. 2017-Janua, p. 6517–6525. ISBN 9781538604571. Disponível em: <http://arxiv.org/abs/1612.08242>. Citado na página 30.

____. YOLOv3: An Incremental Improvement. 4 2018. Disponível em: <http://arxiv.org/abs/1804.02767>. Citado nas páginas 31, 39 e 47.

REN, S.; HE, K.; GIRSHICK, R. B.; SUN, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. **CoRR**, abs/1506.0, 2015. Disponível em: <http://arxiv.org/abs/1506.01497>. Citado nas páginas 28 e 29.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, v. 65, n. 6, p. 386–408, 1958. Citado na página 23.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **Nature**, v. 323, p. 533–536, 1986. Citado na página 23.

RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATHY, A.; KHOSLA, A.; BERNSTEIN, M.; BERG, A. C.; FEI-FEI, L. ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision**, Springer New York LLC, v. 115, n. 3, p. 211–252, 12 2015. ISSN 15731405. Citado na página 53.

SANDLER, M.; HOWARD, A.; ZHU, M.; ZHMOGINOV, A.; CHEN, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. **Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, IEEE Computer Society, p. 4510–4520, 1 2018. Disponível em: <http://arxiv.org/abs/1801.04381>. Citado na página 45.

SANKARAN, S.; MISHRA, A.; EHSANI, R.; DAVIS, C. A review of advanced techniques for detecting plant diseases. **Computers and Electronics in Agriculture**, Elsevier, v. 72, n. 1, p. 1–13, 6 2010. ISSN 01681699. Citado na página 33.

SANTOS, F. P. D.; ZOR, C.; KITTLER, J.; PONTI, M. A. Learning image features with fewer labels using a semi-supervised deep convolutional network. **Neural Networks**, Elsevier, v. 132, p. 131–143, 2020. Citado na página 26.

SELVARAJ, M. G.; VERGARA, A.; RUIZ, H.; SAFARI, N.; ELAYABALAN, S.; OCIMATI, W.; BLOMME, G. AI-powered banana diseases and pest detection. **Plant Methods**, v. 15, n. 1, 2019. ISSN 1746-4811. Disponível em: <https://doi.org/10.1186/s13007-019-0475-z>. Citado nas páginas 35 e 38.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014. Citado na página 25.

SLADOJEVIC, S.; ARSENOVIC, M.; ANDERLA, A.; CULIBRK, D.; STEFANOVIC, D. Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. **Computational Intelligence and Neuroscience**, Hindawi Publishing Corporation, v. 2016, 2016. ISSN 16875273. Citado nas páginas 34 e 38.

STRANGE, R. N.; SCOTT, P. R. Plant Disease: A Threat to Global Food Security. **Annual Review of Phytopathology**, v. 43, n. 1, p. 83–116, 2005. ISSN 0066-4286. Disponível em: <www.annualreviews.org>. Citado na página 19.

SUWA, K.; CAP, Q. H.; KOTANI, R.; UGA, H.; KAGIWADA, S.; IYATOMI, H. A comparable study: Intrinsic difficulties of practical plant diagnosis from wide-angle images. **Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019**, Institute of Electrical and Electronics Engineers Inc., p. 5195–5201, 10 2019. Disponível em: <http://arxiv.org/abs/1910.11506>. Citado nas páginas 20, 36, 38 e 39.

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. Going deeper with convolutions. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 1–9. Citado nas páginas 24 e 25.

THAPA, R.; SNAVELY, N.; BELONGIE, S.; KHAN, A. The Plant Pathology 2020 challenge dataset to classify foliar disease of apples. 4 2020. Disponível em: <http://arxiv.org/abs/2004.11958>. Citado na página 40.

UIJLINGS, J. R. R.; SANDE, K. E. A. van de; GEVERS, T.; SMEULDERS, A. W. M. Selective Search for Object Recognition. **International Journal of Computer Vision**, v. 104, n. 2, p. 154–171, 9 2013. ISSN 1573-1405. Disponível em: <https://doi.org/10.1007/s11263-013-0620-5>. Citado na página 27.

WANG, G.; SUN, Y.; WANG, J. Automatic Image-Based Plant Disease Severity Estimation Using Deep Learning. **Computational Intelligence and Neuroscience**, Hindawi Limited, v. 2017, 2017. ISSN 16875273. Citado nas páginas 34 e 38.

ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. In: **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**. [S.l.]: Springer Verlag, 2014. v. 8689 LNCS, n. PART 1, p. 818–833. ISBN 9783319105895. ISSN 16113349. Citado na página 25.

ZHAO, Z.-Q. Q.; ZHENG, P.; XU, S.-t. T.; WU, X. Object detection with deep learning: A review. **IEEE transactions on neural networks and learning systems**, IEEE, v. 30, n. 11, p. 3212–3232, 2019. ISSN 21622388. Disponível em: <https://arxiv.org/pdf/1807.05511.pdf>. Citado na página 27.


ICMC
USP
SÃO CARLOS