

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**METODOLOGIA PARA A CLASSIFICAÇÃO
AUTOMÁTICA DE DOENÇAS EM PLANTAS
UTILIZANDO REDES NEURAIS
CONVOLUCIONAIS**

VANESSA CASTRO REZENDE

DM 38/2019

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil

2019

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

VANESSA CASTRO REZENDE

**METODOLOGIA PARA A CLASSIFICAÇÃO
AUTOMÁTICA DE DOENÇAS EM PLANTAS
UTILIZANDO REDÉS NEURAIIS
CONVOLUCIONAIS**

DM 38/2019

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil

2019

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

VANESSA CASTRO REZENDE

**METODOLOGIA PARA A CLASSIFICAÇÃO
AUTOMÁTICA DE DOENÇAS EM PLANTAS
UTILIZANDO REDES NEURAIIS CONVOLUCIONAIS**

Dissertação submetida à Banca Examinadora do Programa de Pós-Graduação em Engenharia Elétrica da UFPA para obtenção do Grau de Mestre em Engenharia Elétrica na área de Computação Aplicada.

Orientador: Roberto Célio Limão de Oliveira
Coorientador: Adam Dreyton Ferreira dos Santos

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil

2019

**Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)**

R467m Rezende, Vanessa Castro
Metodologia para a classificação automática de doenças em plantas utilizando redes neurais convolucionais / Vanessa Castro Rezende. — 2019.
81 f. : il. color.

Orientador(a): Prof. Dr. Roberto Célio Limão de Oliveira
Coorientador(a): Prof. Dr. Adam Dreyton Ferreira dos Santos
Dissertação (Mestrado) - Programa de Pós-Graduação em Engenharia Elétrica, Instituto de Tecnologia, Universidade Federal do Pará, Belém, 2019.

1. Redes neurais convolucionais. 2. Doenças em Plantas. 3. Base de Dados . 4. Aprendizado Profundo. 5. Classificação. I. Título.

CDD 621.31098115

Dedico este trabalho aos meus pais, à minha irmã e a minha avó Aldaires Castro Clementino (in memoriam), que infelizmente não pode estar presente neste momento tão importante da minha vida.

Agradecimentos

Primeiramente a Deus, reconhecendo em todo o meu percurso a sua mão grandiosa. Agradeço ainda pela saúde, sabedoria e forças para desenvolver este trabalho mediante todas as dificuldades.

Ao meu orientador, Prof. Dr. Roberto Célio Limão de Oliveira, pela orientação, pelo apoio e pelo aprendizado.

Ao meu co-orientador, Prof. Dr. Adam Dreyton Ferreira dos Santos, pela orientação, pela inspiração, pela paciência e por me ajudar tantas vezes que não consigo contar.

Ao bolsista Michel Victor Carvalho Costa que participou diretamente da execução desse projeto.

À Universidade Federal do Sul e Sudeste do Pará e a Universidade Federal do Pará, pela minha formação profissional.

Aos meus pais, Alessandra Castro Rezende e Euvaldo Sobreira Rezende, pela força e incentivo que sempre prestaram em todos os projetos que iniciei em minha vida.

À minha irmã Maressa Castro Rezende, que sempre mostrou seu sorriso doce e seu olhar de criança feliz, me dando forças quando estava desanimada.

À minhas vós Aldaires Castro Clementino e Edna Rezende de Souza que se orgulharam muito pelo meu ingresso no programa.

À minha melhor amiga Aryane Pinheiro Vilhena pela amizade e prestatividade desde quando eu tenho memória de existir.

Ao meu noivo Jairo Vinícius de Oliveira, pelo apoio e pela imensa paciência quando foi necessário. Agradeço por sua força e as milhares de vezes que me encorajou quando estava desanimada.

À todos que participaram direta ou indiretamente desse trabalho. Obrigada!

*“Só se vê bem com o coração, o essencial é invisível aos olhos.”
(Antoine de Saint-Exupéry - O Pequeno Príncipe)*

Sumário

1	Introdução	1
1.1	Trabalhos Relacionados	2
1.2	Motivações	5
1.3	Objetivos e contribuições	7
1.4	Organização da Dissertação	8
2	Fundamentação Teórica	10
2.1	Aprendizagem Profunda	10
2.1.1	Redes Neurais Artificiais	11
2.1.1.1	Estrutura de um Neurônio Artificial	12
2.1.1.2	Arquitetura Geral de uma RNA	13
2.1.2	Processo de Aprendizagem em Redes Neurais Artificiais	14
2.1.3	Vantagens e Desafios na Utilização de Redes Neurais Artificiais	17
2.2	Redes Neurais Convolucionais	18
2.2.1	Arquitetura de uma Rede Neural Convolucional	19
2.2.1.1	VGG	23
2.2.1.2	RESNET	24
2.2.1.3	RESNETXT	27
2.2.1.4	DENSENET	29
2.2.2	Treinamento em Redes Neurais Convolucionais	32
3	Materiais e Métodos	34
3.1	Base de Dados de Doenças em Plantas XDB	34
3.1.1	Digiphatos Embrapa	34
3.1.2	Pré-processamento da base de dados XDB	36
3.2	Treinamento das Redes Convolucionais Utilizadas	39
3.2.1	Divisão do Conjunto de Dados	39
3.2.2	Técnicas de Regularização Utilizadas	39
3.2.3	Configurações dos Modelos	41
3.3	Métricas utilizadas na avaliação de desempenho dos resultados	42
4	Resultados	46
4.1	Avaliação de desempenho através de acurácia, <i>f1-score</i> e AUC	47
4.2	Teste estatístico não-paramétrico	52
5	Considerações finais e trabalhos futuros	55

Referências	58
APÊNDICE A Quantidade de imagens após o pré-processamento da base 32hard.	65
APÊNDICE B Quantidade de imagens após o pré-processamento da base 64hard.	66
APÊNDICE C Imagens da Base de Dados de Doenças de Plantas XDB.	67

Lista de ilustrações

Figura 1	Neurônio Artificial.	13
Figura 2	Modelo de uma rede neural <i>feedforward</i> MLP.	14
Figura 3	Funcionamento geral de uma CNN aplicada a classificação de doenças em plantas.	19
Figura 4	Operação de Convolução.	20
Figura 5	Operação de agrupamento (<i>pooling</i>).	21
Figura 6	Operação em uma camada convolucional com ReLU e <i>Pooling</i>	23
Figura 7	Arquitetura VGG16 aplicada ao problema de classificação de doenças em plantas.	24
Figura 8	Arquitetura de um bloco residual.	26
Figura 9	Arquitetura ResNet aplicada ao problema de classificação de doenças em plantas.	26
Figura 10	Arquitetura DenseNet aplicada ao problema de classificação de doenças em plantas.	30
Figura 11	Um bloco denso de 4 camadas com uma taxa de crescimento de $t = 3$	31
Figura 12	Metodologia para a classificação de imagens de doenças em plantas.	34
Figura 13	Produção Agrícola Anual Brasileira em 2018 (Estimada em Milhões de Toneladas).	35
Figura 14	Exemplo das etapas de pré-processamento.	38
Figura 15	Exemplo de AUC.	43
Figura 16	Desempenhos das CNNs ResNetXt50, DenseNet169, VGG16 e ResNet101v2 em termos de acurácia durante a etapa de treinamento na base de dados 32hard.	50
Figura 17	Desempenhos das CNNs ResNetXt50, DenseNet169, VGG16 e ResNet101v2 em termos de acurácia durante a etapa de teste na base de dados 32hard.	51
Figura 18	Desempenhos das CNNs ResNetXt50, DenseNet169, VGG16 e ResNet101v2 em termos de acurácia durante a etapa de treinamento na base de dados 64hard.	51
Figura 19	Desempenhos das CNNs ResNetXt50, DenseNet169, VGG16 e ResNet101v2 em termos de acurácia durante a etapa de teste na base de dados 64hard.	52

Lista de tabelas

Tabela 1	Arquitetura da VGG16.	25
Tabela 2	Configuração das arquiteturas de ResNet com 50, 101 e 152 camadas. .	27
Tabela 3	Comparação das arquiteturas ResNet e ResNetXt com 50 camadas. . .	29
Tabela 4	Configuração das arquiteturas de DenseNet com 121, 169 e 201 camadas.	32
Tabela 5	Taxa de diferença aplicada de acordo com cada dimensão.	37
Tabela 6	Configurações das bases de dados.	38
Tabela 7	Média da métrica de acurácia no teste para as bases de dados.	48
Tabela 8	Média da métrica de <i>f1-score</i> para as bases de dados.	48
Tabela 9	Média da métrica AUC para as bases de dados.	49
Tabela 10	Ranking médio computado pelo teste de Friedman.	53
Tabela 11	Valores <i>p</i> não-ajustado (Friedman) e ajustado (Shaffer e Bergmann) para múltiplas comparações.	53
Tabela 12	Teste Estatístico não-paramétrico de Estimativa por Contraste.	54

Lista de abreviaturas e siglas

AP	Aprendizado Profundo
AUC	Área Sob a Curva
BN	<i>Batch Normalization</i>
CNN	<i>Convolutional Neural Network</i>
EMBRAPA	Empresa Brasileira de Pesquisa Agropecuária
FC	<i>Fully-connected</i>
GPU	<i>Graphics Processing Units</i>
IA	Inteligência Artificial
IBGE	Instituto Brasileiro de Geografia Estatística
IC	Inteligência Computacional
MLP	<i>Multilayer Perceptron</i>
PDDB	Base de Imagens de Sintomas de Doenças de Plantas
PIB	Produto Interno Bruto
SGD	<i>Stochastic Gradient Descent</i>
SVM	Máquinas de Vetores de Suporte
RNA	Rede Neural Artificial
RNC	Rede Neural Convolucional
ROC	<i>Receiver Operating Characteristic</i>
UFPA	Universidade Federal do Pará
UNIFESSPA	Universidade Federal do Sul e Sudeste do Pará
XDB	Base de Dados PDDB Extendida

Resumo

As redes neurais convolucionais (CNNs) são uma das técnicas de aprendizado profundo que, devido ao avanço computacional dos últimos anos, alavancaram a área de visão computacional ao possibilitar ganhos substanciais nos mais variados problemas de classificação, principalmente aqueles que envolvem imagens digitais. Tendo em vista as vantagens na utilização dessas redes, diversas aplicações para a identificação automática de doenças de plantas foram desenvolvidas para assistência especializada ou ferramentas de triagem automática, contribuindo para práticas agrícolas mais sustentáveis e maior segurança na produção de alimentos. Nesse contexto, este trabalho tem como objetivo propor uma metodologia para a classificação de múltiplas patologias referentes a diversas espécies de plantas tendo como insumo uma base de dados composta de imagens digitais de doenças em plantas. Inicialmente, esta metodologia envolveu etapas de tratamento das imagens da base de dados de doenças em plantas para possibilitar que estivessem aptas a serem entradas dos modelos de CNNs selecionados (VGG16, ResNet101v1, ResNet101v2, ResNetXt50 e DenseNet169), assim como para geração de dez novas bases, variando entre 50 e 66 classes com maior representatividade, com o intuito de submeter os modelos a situações diversas. Após o treinamento dos modelos, um estudo comparativo foi conduzido com base em métricas de classificação amplamente utilizadas, como acurácia no teste, *f1-score* e área sob a curva. A fim de atestar a significância dos resultados obtidos, foi realizado o teste estatístico não-paramétrico de Friedman e dois procedimentos *post-hoc*, que demonstraram que ResNetXt50 e DenseNet169 obtiveram resultados superiores quando comparadas com VGG16 e ResNets.

Palavras-chave: Redes neurais convolucionais, Doenças em Plantas, Base de dados, Classificação, Aprendizado Profundo.

Abstract

Convolutional neural networks (CNNs) are one of the deep learning techniques that, due to the computational advancement of recent years, have enhanced the field of computer vision by enabling substantial gains in various classification problems, especially those involving digital images. Given the advantages of using these networks, a variety of applications for automatic plant diseases identification have been developed for specialized assistance or automated screening tools, contributing to more sustainable farming practices and improved food production security. In this context, this work aims to propose a methodology for the classification of multiple pathologies from distinct plant species, having as input a database composed of digital images of plant diseases. Initially, this methodology involved image preprocessing activities on the plant disease database to provide the appropriate input for selected CNN models (VGG16, ResNet101v1, ResNet101v2, ResNetXt50 and DenseNet169), as well as to generate ten new bases, ranging from 50 to 66 classes with greater representativeness, to submit the models to different situations. After model training, a comparative study was conducted based on widely used classification metrics such as test accuracy, *f1-score*, and area under the curve. To attest the significance of obtained results, the Friedman nonparametric statistical test and two post-hoc procedures were performed, which showed that ResNetXt50 and DenseNet169 obtained superior results when compared with VGG16 and ResNets.

Keywords: Convolutional Neural Networks, Plant Diseases, Database, Classification, Deep Learning.

1 Introdução

Nas últimas décadas, os investimentos maciços na mecanização do campo, no melhoramento genético de sementes e nos avanços tecnológicos aplicados a fertilização do solo, ocasionaram a elevação da produtividade mundial do setor agrícola (THORSTENSEN; GABRIEL; MESQUITA, 2019).

Seguindo a tendência mundial, nos últimos 50 anos, a agricultura brasileira enfrentou uma série de mudanças significativas com relação a suas estruturas de produção, suprimento de insumos, comercialização e distribuição interna e externa, desempenhando um papel fundamental na economia brasileira (MESSA, 2015). Tais mudanças foram possibilitadas principalmente devido a incorporação da tecnologia no setor, além da expansão da fronteira agrícola que justifica o aumento expressivo da participação do Brasil no agronegócio mundial.

Contudo, um dos maiores desafios desse setor está relacionado as incidências de doenças em plantas que representam uma ameaça significativa à segurança alimentar global, comprometendo as lavouras nacionais e mundiais, além de representar grandes perdas de lucro para o setor agrícola (STRANGE; SCOTT, 2005). Os efeitos destas desordens variam de sintomas leves até catástrofes nas quais grandes áreas plantadas para a produção de alimentos são destruídas. Em termos biológicos, essas doenças são difíceis de controlar porque suas populações são variáveis no tempo, espaço e genótipo.

Nesse contexto, diversos estudos têm sido realizados para tentar prever os incidentes e minimizar as perdas através de técnicas de inteligência computacional (IC) baseadas no aprendizado profundo (AP) e na visão computacional para o reconhecimento de padrões em bases de dados de imagens de doenças em plantas (GANDHI et al., 2018; LIU et al., 2017; MOHANTY; HUGHES; SALATHÉ, 2016).

Entre várias arquiteturas de redes neurais utilizadas na área de AP, as redes neurais convolucionais (*convolutional neural networks* - CNN) têm se consolidado como uma abordagem dominante para o reconhecimento de objetos visuais devido seus resultados satisfatórios em diferentes problemas que envolvam visão computacional (OLIVEIRA; ALVES; MALQUI, 2018; PENHA, 2018; TODA; OKURA, 2019). O uso de tais modelos vêm sendo impulsionado pelos atuais avanços na capacidade de processamento dos computadores, principalmente com relação ao uso de unidades gráficas de processamento (*graphics processing units* - GPUs).

As CNNs consistem em camadas convolucionais, que são conjuntos de filtros de

imagens que geram mapas de características, juntamente com outras camadas (por exemplo, *pooling*). Na classificação de imagens, esses mapas de características são extraídos repetitivamente através de convolução e outras camadas de processamento, e a rede eventualmente exibe um rótulo indicando uma classe estimada.

Dado um conjunto de dados de treinamento, a CNN otimiza os pesos e os parâmetros de filtro nas camadas ocultas para gerar características adequadas para resolver o problema de classificação, ao contrário das técnicas tradicionais de aprendizado de máquina que geralmente utilizam características extraídas ou selecionadas por técnicas externas (TODA; OKURA, 2019). Em princípio, os parâmetros na rede são otimizados por retropropagação e abordagens de gradiente descendente para minimizar o erro nas tarefas de classificação.

Contudo, embora as CNNs tenham sido originalmente introduzidas há mais de 20 anos, melhorias no *hardware* de computadores e na estrutura de rede permitiram o treinamento de CNNs realmente profundas apenas nos últimos anos. A LeNet (LECUN et al., 1990) original consistia em 5 camadas, a VGG (LIU; DENG, 2015) apresentava 16 camadas, e somente recentemente as *Residual Networks* (ResNets)(HE et al., 2016) e a DenseNet (HUANG et al., 2017) ultrapassaram a barreira das 100 camadas.

Nesse contexto geral, o presente trabalho tem como foco o desenvolvimento de uma metodologia para a classificação de doenças em plantas que envolve pré-processamento de imagens e aplicação de diferentes arquiteturas de redes convolucionais com o objetivo de classificar entre 50 e 66 doenças de múltiplas espécies de plantas presentes na base de dados de doenças de plantas XDB, disponibilizada pela Embrapa (BARBEDO et al., 2018). Foram criadas dez bases de dados com diferentes números de classes e distintas resoluções de imagens. A avaliação dos resultados foi baseada primariamente em métricas amplamente difundidas, como acurácia na etapa de teste, *f1-score* e área sob a curva (AUC). Por completude, considerou-se um estudo estatístico não-paramétrico a fim de verificar a significância dos resultados obtidos, tendo como entrada a primeira métrica supracitada.

1.1 Trabalhos Relacionados

As CNNs constituem uma das arquiteturas mais utilizadas na área de AP supervisionado, atuando de maneira que as características dos dados possam ser exploradas de forma automática para obtenção dos melhores resultados. Esse tipo de rede pode ser resumida basicamente em um modelo que toma como base a integração da visão biológica com o sistema neural (GOLHANI et al., 2018). Pesquisadores descrevem a CNN como uma arquitetura complexa que leva consideravelmente mais tempo para treinar os neurônios. Apesar disso, essas redes possuem um alto poder de precisão nas tarefas de classificação,

além de uma alta taxa de reconhecimento de objetos, o que justifica o seu uso (TODA; OKURA, 2019).

As CNNs tiveram destaque a partir do ano de 2012, quando foram aplicadas na redução da taxa de erro na competição de reconhecimento de imagens ImageNet12. A partir da ascensão dessas redes, a área de Visão Computacional tem sido invadida por abordagens que utilizam as CNNs para tarefas de classificação e detecção de objetos em imagens, em substituição às abordagens anteriores que envolviam a extração manual das características das imagens para posteriormente aplicar algum método de classificação, como as máquinas de vetores de suporte (BEZERRA, 2016).

Aplicações recentes de CNNs envolvem a detecção de pedestres e placas de sinais de trânsito (REDMON et al., 2016), reconhecimento de imagens (YU et al., 2018), detecção de tumores (ÖZYURT et al., 2019; SEETHA; RAJA, 2018), identificação do gênero de pequenos insetos (MARQUES et al., 2018) e detecção de doenças em plantas (LIU et al., 2017; SLADOJEVIC et al., 2016; FERREIRA, 2017).

Em particular, a CNN provou ser uma ferramenta poderosa para reconhecimento e classificação de imagens, bem como para extrair suas características não-lineares, discriminantes e invariantes. No quesito extração de características, o trabalho de Lu et al. (2017) propôs um novo método de diagnóstico automático de doenças do arroz, baseado na utilização de AP das CNNs que foram treinadas para identificar dez doenças comuns do arroz em um conjunto de 500 imagens naturais referentes às folhas e hastes doentes e saudáveis do arroz. Sob a estratégia de validação cruzada com dez repetições, o modelo proposto atingiu uma precisão de 95,48%. Comprovando a superioridade da precisão das CNNs em relação a aplicação de modelos convencionais de aprendizado de máquina para esse tipo de problema.

A pesquisa de Liu et al. (2017) aponta a importância do diagnóstico precoce e a identificação precisa das doenças da folha da macieira, que pode auxiliar no controle da propagação das patologias e garantir o desenvolvimento saudável da indústria da maçã. Esse trabalho propôs uma abordagem de identificação precisa para doenças da folha de maçã com base em uma CNN baseada no modelo de rede AlexNet, utilizando um conjunto de dados de 13.689 imagens de folhas de macieiras doentes. Os resultados mostraram que a variante AlexNet obteve uma precisão geral de 97,62% com uma redução significativa na quantidade de parâmetros do modelo, obtendo uma melhora de cerca de 10,83% em sua taxa de precisão com relação a AlexNet original.

O trabalho de Sladojevic et al. (2016) envolveu o desenvolvimento de um modelo para o reconhecimento de doenças de plantas baseado na classificação de imagens foliares utilizando uma CNN tradicional pré-treinada na base de dados ImageNet. Todas as imagens coletadas para o conjunto de dados foram baixadas da Internet e pesquisadas por doenças e nome da planta em diversas fontes e em diferentes idiomas. As imagens do

conjunto de dados foram agrupadas em 15 classes diferentes, sendo que 13 delas representavam doenças de plantas que podiam ser visualmente determinadas a partir de folhas. Para o teste de precisão, a técnica de validação cruzada com dez repetições também foi utilizada para avaliar um modelo preditivo. Os resultados experimentais alcançados por esse modelo obtiveram precisão entre 91% e 98%, resultando em uma média de 96,3%.

No ramo das aplicações móveis inteligentes, Mohanty, Hughes e Salathé (2016) implementaram um sistema automatizado de reconhecimento de imagem através do uso de smartphones e processadores de alto desempenho para a detecção de doenças em plantas. Para essa tarefa, utilizaram CNNs, tais com as arquiteturas AlexNet e GoogleNet para fins comparativos. Como esperado, a GoogLeNet apresentou desempenho consistentemente melhor que a AlexNet. Em um dos casos de teste, essa arquitetura alcançou uma precisão geral de 99,35% na detecção de 26 doenças (ou sua ausência) em 14 espécies de plantas em um total de 54.306 imagens coloridas das patologias.

Como observado, as CNNs vêm comprovando seu potencial quando aplicadas a tarefas gerais e altamente variáveis em muitas categorias de objetos. Além disso, já foi demonstrado que seu poder pode exceder o desempenho humano em algumas tarefas de reconhecimento de objetos (YU et al., 2018). Entretanto, para o sucesso desses modelos, é necessária uma grande quantidade de dados de treinamento rotulados com suas respectivas classes para produzir uma rica hierarquia de características, já que são redes difíceis (mas não impossíveis) de treinar utilizando pequenos conjuntos de dados.

Nesse cenário, Gandhi et al. (2018) propôs um sistema para detecção de doenças em plantas no qual o agricultor realizaria a captura da imagem da patologia com seu dispositivo móvel que enviaria essa imagem ao servidor, através do qual o modelo é executado, e responderia a essa requisição com a classificação da doença. O sistema proposto foi baseado na aplicação de CNNs para a classificação das doenças em plantas indianas, fazendo uso de Redes Contraditórias Generativas para aumentar o limitado conjunto de dados de plantas e doenças indianas. Além disso, a base de dados *Plant Village* da *Pennsylvania State University* foi utilizada para aumentar a representatividade da base utilizada e gerar os dados de treinamento. Para a tarefa de classificação, os modelos de redes convolucionais Inceptionv3 e MobileNet foram testados para comparação tanto em termos de precisão, como velocidade de treinamento. Os resultados dos testes mostraram uma vantagem significativa na acurácia do modelo MobileNet que alcançou uma precisão final de teste de 92%, enquanto a Inceptionv3 obteve uma precisão de 88,6%.

No Brasil, o trabalho de Ferreira (2017) teve como objetivo utilizar CNNs para realizar a detecção de ervas daninhas em imagens de lavouras de soja no Mato Grosso do Sul e classificar essas ervas entre gramíneas e folhas largas, com o objetivo de auxiliar no diagnóstico do tipo de herbicida a partir do tipo de erva daninha detectado. O início desse trabalho se deu pela captura das imagens que iriam compor a base de dados de

treinamento da plantação de soja através de um drone. Para a tarefa de classificação, o modelo de CNN AlexNet foi aplicado na base de dados gerada. Esse trabalho obteve como resultado um precisão de cerca de 98% na tarefa de detecção.

Logo, o presente trabalho justifica a aplicação das CNNs baseado nos diversos estudos da literatura que comprovam que elas são um dos métodos da área de AP mais utilizado não só na classificação de doenças em plantas, mas também em diversas outras áreas de pesquisa. A alta no uso dessas redes se deve ao fato dela comprovadamente possuir resultados mais precisos que os métodos tradicionais de reconhecimento de objetos. Além de apresentar resultados satisfatórios, a utilização de CNNs reduz a complexidade no que tange a definição do tipo de arquitetura dos filtros convolucionais que, nos métodos tradicionais de Visão Computacional são definidos manualmente. Adicionalmente, a metodologia deste trabalho se diferencia das outras propostas na literatura, entre outros aspectos, no que tange o uso de algumas CNNs mais atuais, aplicação na base de dados XDB, e estudo estatístico não-paramétrico nos resultados obtidos.

1.2 Motivações

As patologias que acometem as plantas têm sido uma das maiores ameaças à segurança alimentar, porque reduzem drasticamente o rendimento das colheitas, comprometendo a qualidade de frutas, legumes, grãos e leguminosas. Normalmente, essas patologias danificam o aparelho fotossintético e afetam o crescimento da planta. Cerca de 85% das doenças de plantas é causada por organismos semelhantes a fungos. Outras doenças graves das plantas são causadas por bactérias, vírus e viróides, e poucas doenças são causadas por certos nematóides (TODA; OKURA, 2019; GOLHANI et al., 2018).

Segundo a *Society for General Microbiology* (S.G.M., 2011), as doenças das plantas contribuem com cerca de 10% a 16% das perdas na safra global anual, custando aproximadamente US \$220 bilhões. De acordo com um relatório da *Food and Agriculture Organization* (F.A.O., 2009), a produção de alimentos deve crescer 70% no mundo e duplicar nos países em desenvolvimento para atender às necessidades alimentares da população mundial que deve chegar a 9,1 bilhões em 2050, levando em consideração as mudanças climáticas e a rápida urbanização.

Além disso, vale ressaltar que essas patologias também representam consequências desastrosas para os pequenos agricultores cujos meios de subsistência dependem de culturas saudáveis. Na maioria dos países em desenvolvimento, mais de 80% da produção agrícola é gerada por pequenos agricultores e estima-se que ocorra uma perda de rendimento de mais de 50% devido a pragas e doenças, tornando os pequenos agricultores um grupo particularmente vulnerável a perturbações derivadas de patógenos no suprimento de alimentos (MOHANTY; HUGHES; SALATHÉ, 2016).

Dessa forma, vários esforços foram desenvolvidos para evitar a perda de colheitas devido a doenças. As abordagens históricas da aplicação generalizada de pesticidas foram, na última década, cada vez mais complementadas por abordagens de gestão integrada de pragas (MOHANTY; HUGHES; SALATHÉ, 2016).

Contudo, o uso abundante de produtos químicos como bactericidas, fungicidas e nematicidas para controlar essas patologias têm causado efeitos adversos no agroecossistema, tornando necessária a utilização de técnicas efetivas de detecção precoce do agente causador para o controle das doenças e a sustentabilidade do agroecossistema, evitando o desperdício desnecessário de recursos financeiros e, conseqüentemente, obtendo uma produção mais saudável (GOLHANI et al., 2018; BARBEDO et al., 2018).

No nível biológico, a identificação rápida e precisa do organismo causal, assim como estimativas precisas da gravidade da doença, seu efeito e a identificação de seus mecanismos de virulência, fazem com que seja possível tomar decisões que possam minimizar a doença a partir da redução do inóculo do agente patogênico, inibindo seus mecanismos de virulência, além de promover a diversidade genética na cultura (STRANGE; SCOTT, 2005).

Diagnósticos incorretos, causados muitas vezes pela dificuldade de identificar o patógeno a campo, podem acarretar utilização excessiva de agrotóxicos, elevando os malefícios causados por estes produtos, reforçando o fato de que o diagnóstico preciso de doenças ainda representa um desafio significativo para esse setor (GONÇALVES et al., 2010). Tradicionalmente, a identificação de doenças de plantas dependeu da anotação humana por inspeção visual.

Atualmente, os métodos de diagnóstico precoce envolvem a combinação dos métodos humanos com os métodos tecnológicos ou somente a utilização do último (TODA; OKURA, 2019). Essa mudança se deu pelos avanços tecnológicos e a redução de custos no campo da aquisição de imagens digitais que possibilitaram a introdução de uma série de métodos de diagnóstico de doenças em plantas baseados na classificação de imagens em um nível prático.

Embora a identificação por especialistas ainda seja uma das formas mais utilizada na identificação de patologias, sistemas de IC podem fornecer ferramentas alternativas e complementares confiáveis para identificação de padrões, enquanto reduz o número de identificações de rotina realizadas por especialistas. Uma técnica de IC que tem se mostrado promissora na resolução desse tipo de problema é a AP aplicada a classificação de imagens (MARQUES et al., 2018).

O objetivo da classificação de imagens através da utilização de técnicas de AP é classificar uma imagem específica de acordo com um conjunto de categorias possíveis. Os classificadores de imagens de última geração geralmente resultam de abordagens de

aprendizado de transferência baseadas em redes neurais pré-treinadas.

Uma tentativa inicial de utilizar o aprendizado profundo para diagnósticos de doenças de plantas baseadas em imagens foi relatada em 2016, através do trabalho de Mohanty, Hughes e Salathé (2016), onde o modelo treinado foi capaz de classificar 14 culturas e 26 doenças com uma precisão de 99,35% contra imagens ópticas. Desde então, foram relatadas sucessivas gerações de diagnósticos de doenças baseadas em AP em diversos trabalhos da literatura.

No que tange doenças em plantas, as CNNs são as técnicas de AP que têm sido mais utilizadas para análise de imagens de plantas desde os primeiros dias de sua evolução devido comprovadamente já possuírem resultados confiáveis em diversos trabalhos da literatura que envolveram a identificação de espécies, plantas, macroinvertebrados aquáticos, mamíferos, insetos, entre outros casos (MARQUES et al., 2018).

Graças ao rápido desenvolvimento do *hardware* e à melhoria dos métodos de aprendizagem, as CNNs de larga escala tornaram-se treináveis a partir de 2010 (TODA; OKURA, 2019). Um ponto de virada importante para as CNNs foi a introdução da Alex-Net, que superou significativamente a precisão de classificação de imagens obtida por abordagens tradicionais de aprendizado de máquina no Desafio de Reconhecimento Visual de Grande Escala do ImageNet em 2012.

A grande vantagem da CNN, quando comparada com os métodos tradicionais de AP e Visão Computacional, para problemas de classificação diz respeito à capacidade que a primeira possui de aprender características de forma intrínseca, eliminando assim a necessidade de uma fase anterior de extração (PENHA, 2018). Além disso, a possibilidade das CNNs conseguirem misturar múltiplos mapas de características ao mesmo tempo as permite extrair características cada vez mais complexas, quando comparadas com técnicas clássicas, sendo capazes de sozinhas criar filtros extremamente complexos aproveitando ao máximo as informações provenientes dos dados de treinamento. Essas vantagens motivaram a utilização dessas redes para a criação da metodologia proposta neste trabalho, a fim de classificar imagens digitais de doenças em plantas.

1.3 Objetivos e contribuições

O trabalho em questão tem como objetivo central desenvolver uma metodologia que se baseia no pré-processamento de imagens e utilização de CNNs para realizar a tarefa de classificação de doenças em plantas em bases de dados com diversas culturas e patologias. Esta pesquisa se baseou no fato de que a AP, especialmente relacionada aos modelos de redes convolucionais, atualmente representa o estado da arte para as tarefas de classificação e reconhecimento de imagens.

A definição do objetivo geral resultou no entendimento acerca da problemática retratada, fazendo com que fosse possível realizar a delimitação do escopo deste estudo que se materializa através dos seguintes objetivos específicos:

- Criar um método para a realização do pré-processamento da base de dados baseado na taxa de diferença e no redimensionamento das imagens, visando torná-las aptas aos modelos de CNN selecionados;
- Implementar as CNNs VGG16, ResNet101v1, ResNet101v2, ResNetXt50 e DenseNet169, selecionadas para treinamento e teste, tendo como entrada uma base de dados de doenças em plantas e;
- Realizar um estudo estatístico não-paramétrico a fim de atestar a significância dos resultados obtidos através das métricas acurácia no teste, *f1-score* e área sob a curva (AUC), gerando um *ranking* de desempenho das CNNs.

A partir da definição dos objetivos específicos deste trabalho pode-se definir como principais contribuições:

- Disponibilização de uma metodologia útil aos especialistas da área de doenças em plantas de forma a possibilitar o diagnóstico incipiente, identificando patologias em estágios prematuros em uma determinada espécie ou evitando a disseminação de uma doença em uma plantação;
- Auxílio no controle da propagação das patologias e suporte na garantia de desenvolvimento saudável da indústria agrícola; e
- Estudo comparativo entre as CNNs deste trabalho, atestando a significância dos resultados obtidos para diferentes bases de dados com múltiplas doenças de distintas espécies de plantas oriundas das regiões brasileiras.

1.4 Organização da Dissertação

O presente trabalho está organizado em cinco capítulos dispostos da seguinte forma:

- O Capítulo 1 delimita o contexto no qual este trabalho está inserido através de uma breve introdução, trabalhos relacionados, motivações, objetivos e contribuições;
- O Capítulo 2 introduz os principais conceitos acerca da AP, das redes neurais tradicionais e os principais aspectos do processo de funcionamento das CNNs;

-
- O Capítulo 3 aborda a metodologia proposta empregada na tarefa de classificação de imagens de bases de dados de doenças em plantas, assim como as ferramentas computacionais utilizadas;
 - O Capítulo 4 apresenta e discute os resultados obtidos com a aplicação das CNNs nas bases de dados geradas a partir da XDB;
 - O Capítulo 5 discute as considerações finais e os trabalhos futuros que podem ter origem a partir deste trabalho.

2 Fundamentação Teórica

2.1 Aprendizagem Profunda

Em geral, o aprendizado é o conceito que denomina as ações que impulsionam determinado comportamento a partir de um estímulo externo recebido, representando o processo de aquisição de conhecimento. Ao longo do processo de aprendizagem biológico, os animais são previamente estimulados a interpretar e compreender conceitos mais simples para posteriormente se tornarem capazes de aprender conceitos mais complexos baseados em acontecimentos previamente observados (FERNANDES, 2013 apud SILVA, 2017).

Baseado neste mecanismo, o aprendizado de máquina surgiu como um ramo da IC que se traduz em um método de análise de dados que objetiva automatizar o desenvolvimento de modelos analíticos e tem se consolidado rapidamente como um importante instrumento para resolução de tarefas computacionais complexas (OEI et al., 2019).

Nesse contexto, a AP surge como uma subárea do aprendizado de máquina que busca estudar a aplicação de técnicas que objetivem simular o comportamento do cérebro humano quando exposto a tarefas, como reconhecimento visual e processamento da linguagem natural (BEZERRA, 2016). As técnicas de AP envolvem múltiplas camadas de processamento não-linear que buscam realizar o reconhecimento de padrões, permitindo a extração automática de características a partir de um dado conjunto de treinamento, sem a necessidade de uma série de técnicas de processamento de imagens e reconhecimento de padrões. Dessa forma, extração, seleção e classificação de características são abstraídas no próprio modelo.

Nos últimos anos, as redes neurais artificiais (RNA) se tornaram amplamente bem sucedidas, especialmente no campo da análise de imagens, devido, dentre outros fatores, o avanço computacional que possibilitou o aumento significativo da quantidade camadas, que gera a AP. Segundo Hubel e Wiesel (1998), os algoritmos de AP produzem representações hierárquicas de alto nível a partir dos dados de entrada e ao longo das camadas de processamento sequenciais em uma RNA. Logo, quanto maior a quantidade de camadas ocultas a RNA possuir, mais profunda é a rede e, dependendo do problema, maior a sua capacidade de precisão.

Segundo Moen et al. (2018), a característica mais notável das redes neurais de AP é a sua generalidade. Com dados de treinamento suficientes elas podem ser utilizadas

para classificar os dados brutos diretamente (por exemplo, uma matriz de *pixels*). Funcionando de forma que as camadas inferiores da rede aprendem a distinguir características primitivas automaticamente, identificando arestas ou transições de cores. As camadas subsequentes aprendem a reconhecer características mais abstratas como combinações de características da camada inferior e, finalmente, mesclam essas informações para fornecer uma classificação de alto nível nas camadas mais próximas a saída da rede.

2.1.1 Redes Neurais Artificiais

RNA é um conceito que teve sua origem através do crescente interesse científico em IC durante a primeira metade dos anos 50 e se caracteriza por um modelo matemático ou computacional, bem documentado e desenvolvido, que busca simular os principais aspectos funcionais de uma rede neural biológica através de cálculos de troca de sinais entre neurônios artificiais (XIE et al., 2016), podendo ser utilizada para produzir um sistema automático de identificação através do aprendizado e da sumarização de dados conhecidos.

As RNAs também podem ser classificadas como técnicas de modelagem que possibilitam a realização de tarefas como aprendizado, lembrança, decisão e inferência, além de serem aplicadas de forma eficiente na resolução de problemas que possuem relações matemáticas complexas em decorrência de sua natureza não linear.

Em geral, a estrutura de uma rede neural é composta por sistemas paralelamente distribuídos formados por unidades de processamento maciçamente interligadas, conhecidas como “neurônios artificiais” ou “nodos”, que calculam determinadas funções matemáticas objetivando armazenar conhecimento experimental e o disponibilizar para os mais variados fins (HAYKIN, 2001). Esses neurônios são organizados em uma ou várias camadas e interligados por um número considerável de conexões, frequentemente unidirecionais, que, em grande parte dos modelos, estão associados a pesos que armazenam o conhecimento representado no modelo e são úteis para estimar a entrada recebida por cada neurônio da rede e responder com uma saída.

O processamento das RNAs envolve a informação de modelos configurados para uma aplicação específica através de um processo de formação, conhecido por “topologia” ou “arquitetura”, que é representado pelo conjunto arbitrário de neurônios disposto nas camadas de entrada, oculta e saída, fornecendo um mapeamento aleatório entre um vetor de entrada e um vetor de saída (POURZANGBAR et al., 2017). Para produzir o vetor de saída, cada neurônio combina as entradas e estima seus pesos. Em seguida, eles produzem uma saída passando os valores somados de todos os neurônios através de funções de transferência. O número de neurônios da camada de entrada é igual aos parâmetros da entrada e, o número de neurônios na camada de saída é igual ao número de parâmetros de saída.

A arquitetura de uma RNA oferece a atividade sináptica através de uma matriz de peso (valores numéricos), que é tipicamente atualizada por um processo de aprendizagem que busca ser semelhante ao humano (ÇELEBI et al., 2017; CASCARDI; MICELLI; AIELLO, 2017). Conseqüentemente, as redes aprendem a mapear valores no espaço de entrada para valores no espaço de saída através do treinamento que é fornecido por algum algoritmo de aprendizado. Durante o período de treinamento, os pesos da rede podem ser alterados dependendo da informação que passa por eles, através do fornecimento de certa quantidade de dados de entrada-saída.

As RNAs treinadas fornecem um mapeamento rápido das quantidades de entrada para as quantidades de saída desejadas. Contudo, para garantir a precisão do mapeamento, a rede deve ser adequadamente treinada. RNAs bem treinadas podem ser utilizadas como modelos substitutos, aumentando a eficiência computacional de um processo de análise numérico.

Tradicionalmente, a neuro-evolução é utilizada como um método para evoluir a arquitetura das RNAs e matrizes de peso sináptico de tal forma que as operações tradicionais de aprendizagem em rede sejam ignoradas e o objetivo seja encontrar uma arquitetura ótima com seus pesos sinápticos associados para alguma instância de problema (THAMES; ABLER; SCHAEFER, 2011). Contudo, devido à falta de um método versátil que determine a melhor topologia para qualquer problema, os modelos de RNAs são frequentemente projetados pela intuição humana. Tentativas anteriores de criar um sistema para automatizar a descoberta de topologias de rede têm utilizado a computação evolutiva, como em Campos, Oliveira e Roisenberg (2016).

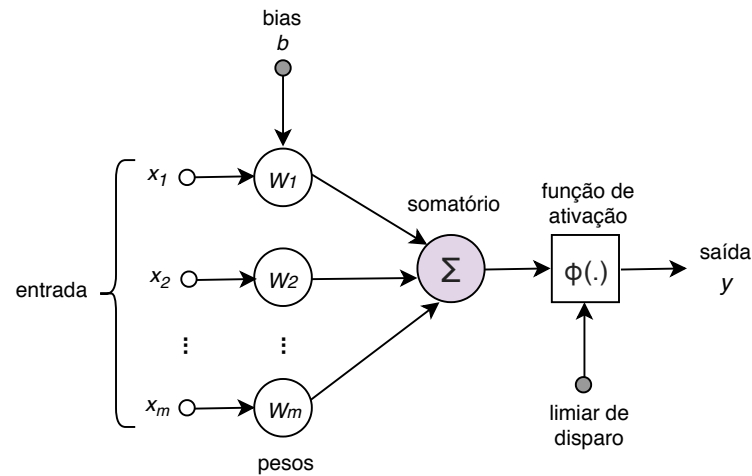
Na última década, as redes neurais têm atraído a atenção de pesquisadores em várias áreas, por serem métodos que podem ser aplicados de maneira eficaz na resolução de problemas em diversos domínios de aplicação. Tais domínios incluem a aproximação de sinal (ou função), classificação de dados, padrão ou reconhecimento de sequências (imagem, fala, texto, etc.), agrupamento de dados, sistemas de jogos, controle (robótico, de processo, veicular, etc.), entre outros (THAMES; ABLER; SCHAEFER, 2011; GANJEFAR; TOFIGHI; KARAMI, 2015).

2.1.1.1 Estrutura de um Neurônio Artificial

A estrutura de um neurônio artificial (Figura 1) consiste basicamente em duas partes: a função da rede e a função de ativação. A função da rede determina como os dados de entrada (sinais de entrada) são combinados dentro do neurônio, enquanto a função de ativação define o valor de saída do neurônio que será propagado aos neurônios seguintes (BRAGA; LUDERMIR; CARVALHO, 2007).

Para cada conexão que liga um neurônio a outro é atribuído um peso, chamado de peso sináptico, que representa o meio pelo qual uma rede armazena o conhecimento

Figura 1: Neurônio Artificial.



Fonte: A autora (2019).

adquirido por um processo de aprendizagem (HAYKIN, 2001). Além disso, o modelo matemático do neurônio pode incluir a variável *bias* ao somatório da função de ativação com o objetivo de aumentar a capacidade de aproximação da rede, sendo este valor ajustado assim como os pesos sinápticos.

O somatório é o elemento responsável por realizar a soma dos valores de entrada já pesados de acordo com os respectivos pesos atribuídos a cada conjunto de sinapses. Já a variável denominada como limiar de disparo ou *threshold* representa um valor de controle que determina a ação posterior da rede.

2.1.1.2 Arquitetura Geral de uma RNA

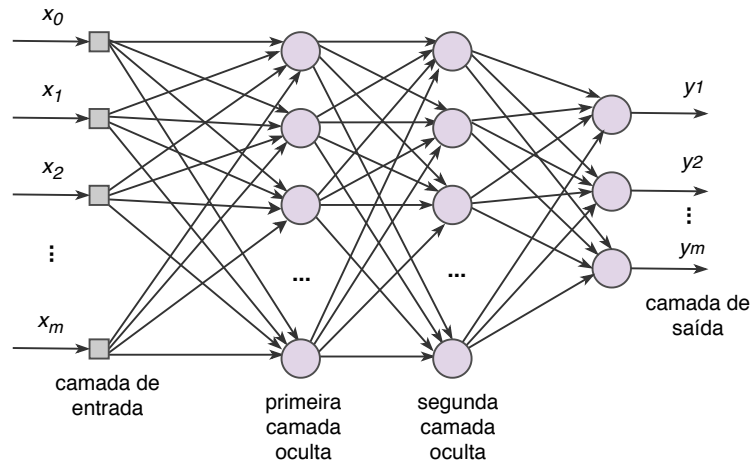
O estabelecimento da arquitetura de uma RNA é considerado um dos fatores mais importantes na sua criação, visto que ela delimita o escopo do problema que pode ser tratado pela rede. Por exemplo, redes recorrentes são mais indicadas para solucionar problemas que envolvam processamento temporal, já as redes que possuem uma única camada de neurônios só solucionam problemas linearmente isolados (BRAGA; LUDERMIR; CARVALHO, 2007). A arquitetura de uma rede é definida pelas seguintes propriedades: número de camadas da rede, número de neurônios em cada camada, tipo de conexão entre os neurônios e a topologia da rede.

Segundo Giovanis et al. (2017), a arquitetura da rede descreve as ligações entre os neurônios artificiais, podendo ser organizados em camadas que estão conectadas entre si com padrões diferentes. Não há limitação com relação ao número de camadas ou de neurônios em cada camada que uma RNA pode ter. Os tipos de redes neurais mais amplamente utilizados são:

1. A rede de alimentação em cadeia única, que consiste numa camada de entrada e uma camada de saída;
2. A rede de alimentação em cadeia múltipla, que consiste numa camada de entrada, camadas ocultas e uma camada de saída. Estes tipos de redes têm apenas uma restrição; as entradas devem ser propagadas para frente, isto é, os dados e cálculos devem fluir numa única direção, a partir da camada de entrada para a camada de saída.

A Figura 2 representa a estrutura mais comum de RNA, *feedforward* ou perceptron multicamada (MLP), na qual o lado esquerdo contém os neurônios que representam a camada de entrada. Esses neurônios geralmente transmitem os dados que lhes foram dados mais adiante na rede. As próximas camadas na rede são conhecidas como “camadas ocultas”, onde cada neurônio nesta camada toma a soma de toda a saída da camada anterior e aplica uma função de ativação aos dados. Se o resultado for maior que um limiar (*threshold*) específico, o neurônio disparará e enviará um valor numérico para a próxima camada.

Figura 2: Modelo de uma rede neural *feedforward* MLP.



Fonte: A autora (2019).

Uma RNA pode ter várias camadas ocultas e qualquer número de neurônios por camada. Este tipo de rede é uma aproximação muito grosseira da complexidade de uma verdadeira rede neural biológica.

2.1.2 Processo de Aprendizagem em Redes Neurais Artificiais

Para que as RNAs sejam capazes de fornecer soluções a um determinado problema é indispensável que passem inicialmente por um processo de aprendizagem que tem por

objetivo extrair características úteis a partir da informação fornecida na entrada. Ao longo do processo de aprendizagem, os pesos da rede são ajustados automaticamente de acordo com os valores fornecidos pelas camadas anteriores (FREITAS, 2001). Conseqüentemente, o conhecimento adquirido pelas RNAs é representado pelo conjunto de valores assumidos por seus pesos ao final do processo. A precisão da saída para um determinado domínio do problema depende da seleção apropriada dos dados utilizados no treinamento da rede, bem como de vários parâmetros de projeto que definem a arquitetura da rede antes de ela ser treinada.

O procedimento utilizado para executar o processo de aprendizagem é chamado de algoritmo de aprendizado que é a função capaz de modificar o peso sináptico da rede. Quando o cérebro recebe um estímulo, ele entra na rede neural para interpretação através do impulso elétrico. Os efetores ¹ convertem o sinal elétrico gerado pela rede neural biológica em uma resposta (CASCARDI; MICELLI; AIELLO, 2017). Da mesma forma, a RNA propõe uma matriz de peso de valor numérico e um viés para cada neurônio, onde, diferentes fórmulas analíticas são resolvidas para criar ajustes na própria matriz, sendo que as quantidades de entrada são processadas através de sucessivas camadas de neurônios.

Desse modo, camadas de entrada e saída são sempre consideradas no processo, além de um número definido de camadas ocultas. Frequentemente, apenas uma camada oculta é considerada, contudo, o uso de duas ou mais camadas ocultas pode, por vezes, melhorar significativamente o desempenho da RNA (HAYKIN, 2001).

O processo de adaptação dos parâmetros e o procedimento de ajuste dos pesos da rede são fundamentais para que as RNAs sejam capazes de fornecer soluções a um determinado problema. As mudanças sinápticas ocorridas durante esse processo podem ser comparadas às mudanças específicas necessárias para, por exemplo, memorizar um padrão visual ou aprender uma tarefa motora (BRAGA; LUDERMIR; CARVALHO, 2007).

Diversas pesquisas envolvendo as atividades cerebrais comprovaram que o cérebro emprega uma variedade de estratégias de aprendizado diretamente relacionadas ao tipo de problema que se deseja resolver. Várias partes do cérebro foram associadas a diferentes categorias de aprendizado, como os gânglios da base com o aprendizado por reforço, o córtex cerebral com aprendizado não-supervisionado e o cerebelo com o aprendizado supervisionado (POTJANS; MORRISON; DIESMANN, 2009).

As técnicas de aprendizado por reforço se referem a uma classe de algoritmos baseados em sistemas de adaptação onde o agente deve aprender através de processos de tentativas e erros, possibilitando um aprendizado através da experiência, sem a necessidade de um modelo do seu ambiente. Nesse tipo de aprendizagem, as ações desejadas são

¹São órgãos responsáveis pelas reações do indivíduo aos estímulos recebidos.

reforçadas com recompensas positivas, enquanto ações indesejadas são penalizadas por sinais de recompensa negativos (NAKANO et al., 2015). Os algoritmos de aprendizado por reforço necessitam de um intervalo de tempo para verificar o fruto de suas decisões, operando em um ambiente de retorno atrasado, onde pode ser difícil entender qual ação leva a qual resultado com o passar do tempo.

A estrutura dos sistemas de aprendizado não-supervisionado pode consistir, por exemplo, em uma camada de entrada, uma camada de saída, conexões *feedforward* da entrada para a saída e conexões laterais entre os neurônios da camada de saída. Outro exemplo é uma rede *feedforward* com múltiplas camadas, em que a livre organização procede na base de camada por camada (BRAGA; LUDERMIR; CARVALHO, 2007). Nesses exemplos, o processo de aprendizado consiste em modificar repetidamente o peso sináptico de todas as conexões do sistema em resposta às entradas. Ao contrário do aprendizado supervisionado, o aprendizado não-supervisionado não possui um “treinador” e o treinamento é realizado baseado em um conjunto de respostas pré-estabelecidas, fazendo com que a rede mensure os valores dos parâmetros que fornecerão a melhor resposta de maneira independente.

A aprendizagem supervisionada, por sua vez, envolve um mecanismo de fornecimento das saídas desejadas com as entradas correspondentes. A rede então processa as entradas e compara suas saídas resultantes com as saídas desejadas. Os erros são calculados para controlar o ajuste de peso sináptico (LIN; WANG; HAO, 2017). Esse processo ocorre uma e outra vez até que os pesos sinápticos confluem para certos valores. O conjunto de dados que habilita o treinamento é chamado de conjunto de treinamento.

Na aprendizagem supervisionada, as informações de uma rede de neurônios agem como um sinal instrutivo para influenciar o padrão de conectividade em outra rede. Como resultado, a rede instruída aprende a processar informações para alcançar determinado objetivo, contribuindo para o desenvolvimento e manutenção de uma variedade de funções cerebrais, por exemplo, os sistemas sensoriais que controlam movimentos dirigidos por objetivos que são calibrados através do *feedback* sensorial, indicando a precisão com que os movimentos são realizados. Além disso, esse tipo de aprendizado contribui para o estabelecimento de redes que suportam certas habilidades cognitivas, tais como reconhecimento de padrões, entre outros (KNUDSEN, 1994). Um dos tipos de algoritmos de aprendizado supervisionado mais utilizados na literatura são os algoritmos de aprendizagem supervisionada baseados em gradiente descendente.

Em suma, a precisão da saída da rede para um determinado domínio do problema depende da seleção apropriada de dados de treinamento, bem como de vários parâmetros de projeto que definem a estrutura da rede antes de ela ser treinada (HAYKIN, 2001).

2.1.3 Vantagens e Desafios na Utilização de Redes Neurais Artificiais

De acordo com Braga, Ludermir e Carvalho (2007), um dos maiores atrativos da solução de problemas por meio de RNAs está no potencial da rede em aprender através de exemplos, assim como generalizar a informação aprendida. Essa generalização está relacionada à habilidade da rede aprender através de conjuntos de dados de entrada, além de dar respostas coerentes a dados não conhecidos. Logo, a capacidade de auto-organização e de processamento temporal faz das RNAs uma ferramenta atrativa para solução de problemas complexos.

Além disso, outra vantagem direta desta abordagem é que os dispositivos de computação neural não precisam ser programados, ao contrário, a escolha aleatória dos pesos iniciais induz ao aprendizado do processo de ajustar os pesos por conta própria, buscando atingir o mínimo de erros na predição.

Entretanto, projetar uma RNA para uma tarefa específica não é um esforço trivial e normalmente é realizada por tentativa e erro (TRUJILLO; MARTÍNEZ; MELIN, 2011). Por exemplo, um projetista de sistemas deve escolher o tipo de função de ativação, a forma como os neurônios estão conectados, a regra de aprendizado, o número de camadas ocultas e número de neurônios em cada camada. Todavia, dado um problema, é difícil inferir como a RNA deveria ser construída para solucioná-lo.

Devido a esses fatores, é necessário possuir um conhecimento do domínio do espaço de entrada-saída que está sendo apresentado à rede, a fim de determinar parâmetros de configuração apropriados à rede antes de seu treinamento. A seleção incorreta dos parâmetros de configuração pode resultar, em muitos casos, em uma rede que não funciona bem após o treinamento. Nesses casos, a má classificação é resultado do mapeamento de características de entrada para classes de saída incorretas, ocasionando em redes com altas taxas de erro.

Ademais, apesar das redes neurais serem eficazes na determinação de um mapeamento entre as entradas e as saídas de um sistema ainda possuem uma natureza de “caixa preta” (ONIZ; KAYNAK, 2015). Isso dá origem à falta de conhecimento sobre as relações causais entre essas entradas e saídas. Os parâmetros que têm maior impacto sobre uma saída particular não podem ser distinguidos entre outros, e assim as redes neurais não conseguem fornecer uma visão da dinâmica do sistema.

Apesar de muitas melhorias na formação de RNAs, menos progressos foram realizados acerca da melhor forma de escolher uma topologia de rede eficaz (PANTRIDGE; SPECTOR, 2016). Uma prática corrente é tentar sub-redes menores que são treinadas para executar partes específicas de uma computação global. Tais sub-redes treinadas são então incluídas em uma rede neural muito maior, e a rede maior é treinada como um todo. A questão de como melhor projetar uma topologia de uma rede neural para solucionar

determinado problema ainda se encontra amplamente sem resposta.

Em síntese, assim como as redes neurais possuem vantagens computacionais consideráveis, também possuem alguns problemas na prática. Por exemplo, elas não são adequadas para executar simulações em um tempo razoável em computadores clássicos, devido suas características massivamente paralelas (GANJEFAR; TOFIGHI; KARAMI, 2015).

2.2 Redes Neurais Convolucionais

Uma CNN é uma arquitetura de AP inspirada em regiões específicas do mecanismo de percepção visual dos animais, denominadas campos receptores, que são responsáveis pela ativação de diferentes neurônios durante a captação visual de uma imagem em um determinado ambiente real, apresentando um nível de sobreposição entre os campos receptivos de neurônios vizinhos (HUBEL; WIESEL, 1968; OLIVEIRA; ALVES; MALQUI, 2018). Baseado nesse comportamento cerebral, diversos sistemas foram desenvolvidos para a extração de características a partir de dados de entrada de um determinado problema.

Os primeiros modelos computacionais baseados nas conexões locais entre neurônios e nas transformações de imagens organizadas de forma hierárquica foram desenvolvidos por Fukushima (1980) em seu modelo denominado *neocognitron*. Posteriormente, LeCun et al. (1990) apresentou uma das primeiras estruturas modernas de uma CNN que consistia basicamente em uma RNA multi-camada denominada LeNet-5 que podia classificar dígitos manuscritos. Como outras redes neurais, a LeNet-5 possuía múltiplas camadas e podia ser treinada com um algoritmo de retropropagação de erro, obtendo representações eficazes da imagem original, além de possibilitar o reconhecimento de padrões visuais diretamente de *pixels* brutos com um pré-processamento baixo ou inexistente.

Em paralelo ao estudo de LeCun et al. (1990), o trabalho de Zhang et al. (1990) utilizou uma RNA invariante ao deslocamento para reconhecer caracteres de uma imagem. No entanto, devido à falta de uma grande quantidade de dados de treinamento e das limitações computacionais da época, essas redes não conseguiram ter um bom desempenho em problemas mais complexos, como classificação de imagem e vídeo em larga escala.

Contudo, o avanço da computação permitiu que esse tipo de rede seja aplicado atualmente com sucesso em diversas tarefas que envolvam processamento, classificação, detecção e reconhecimento em vídeos e imagens digitais, sendo um dos modelos de AP mais conhecidos e utilizados pelos pesquisadores (PENHA, 2018; PONTI; COSTA, 2017).

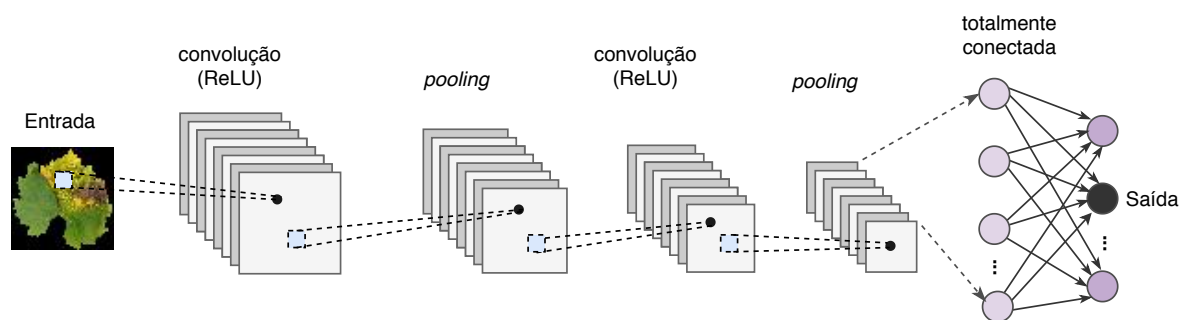
Em geral, o funcionamento das CNNs envolve diversas etapas de processamento da informação, realizadas principalmente pelas camadas convolucionais, que utilizam a operação de convolução para realizar a extração de características a partir dos dados da camada

de entrada (que corresponde aos *pixels* da imagem). Tal operação utiliza uma janela de dados deslizante, também conhecida como filtro convolutivo ou kernel, que percorre a entrada da rede e busca atuar de forma similar à sobreposição dos campos receptores biológicos (OLIVEIRA; ALVES; MALQUI, 2018). Durante o processo de treinamento da rede, os filtros convolutivos têm seus valores ajustados para que sejam capazes de extrair diferentes características que, posteriormente, se convertem em entradas para um determinado algoritmo de aprendizado aplicado em tarefas de classificação ou regressão.

Entre as camadas de convolução é comum a realização de diversas operações adicionais, como *pooling*, normalização e *padding*, que são responsáveis por reduzir a dimensionalidade espacial das representações, gerando mapas de características com diversos atributos específicos, tais como bordas, intensidade de cor, contornos e formas. Esses mapas possuem um conjunto de pesos compartilhados e podem reduzir relativamente à complexidade computacional da rede. Ao final da parte convolutiva da rede, geralmente é utilizada uma sequência de camadas de neurônios conectados a todas as ativações das camadas anteriores de forma análoga às camadas das redes neurais tradicionais.

O esquema geral de funcionamento de uma CNN aplicada ao reconhecimento de uma imagem referente a patologias de plantas em uma base de dados de imagens é ilustrado pela Figura 3. Nota-se que esse modelo de rede neural combina num algoritmo a extração de características e a classificação/regressão.

Figura 3: Funcionamento geral de uma CNN aplicada a classificação de doenças em plantas.



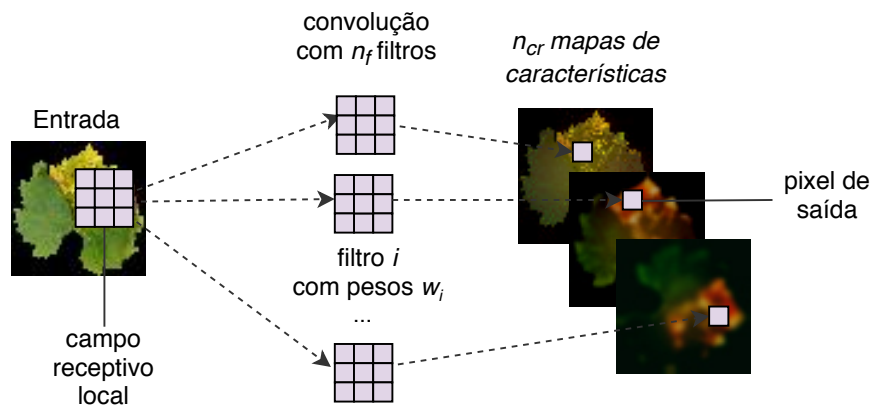
Fonte: A autora (2019).

2.2.1 Arquitetura de uma Rede Neural Convolutacional

A arquitetura típica de uma CNN é dividida em uma série de etapas de processamento da informação que consistem basicamente em uma camada i que transforma a matriz de características anterior, representada por $n_{ca_{i-1}}$ canais e $n_{ce_{i-1}} \times n_{ce_{i-1}}$ células, em outra matriz de características com n_{ca_i} canais e $n_{ce_i} \times n_{ce_i}$ células.

A primeira etapa do processamento de uma CNN consiste na aplicação das camadas convolucionais nos dados de entrada para extrair características de alto nível. (VARGAS; PAES; VASCONCELOS, 2016; FERREIRA, 2017). Essas camadas são compostas por diversos neurônios que são responsáveis por aplicar filtros em uma parte específica da imagem, como demonstrado na Figura 4. Além disso, cada neurônio é conectado a um conjunto de *pixels* da camada anterior e, a cada uma dessas conexões, é atribuído um peso que é definido por uma matriz que representa o filtro de convolução de imagens no domínio espacial.

Figura 4: Operação de Convolução.



Fonte: A autora (2019).

A quantidade de filtros n_f é definida como janelas com $n_{ca_{i-1}}$ canais e $n_w \times n_w$ pesos, onde cada janela irá deslocar toda a saída bidimensional da camada anterior com o mesmo conjunto de filtros. Essa operação é controlada pelas variáveis de *padding* e *stride* e podem ser executadas de diferentes maneiras.

A variável de *padding* ou *zero-padding*, p_a , corresponde à inserção de valores extras em torno da entrada bidimensional (MARQUES et al., 2018). Os valores extras geralmente são definidos como zeros para não alterarem os valores de entrada. Essa informação pode ajudar o campo receptivo a funcionar melhor nas bordas da matriz ou a controlar a dimensão da saída.

Por sua vez, a variável *stride*, s , representa o passo de mudança dos campos receptivos através da matriz de *pixels*, ou seja, determina quantos *pixels* serão “saltados” entre cada janela (PONTI; COSTA, 2017). Com *stride* igual a 1, o deslocamento é de um *pixel*. Logo, para uma imagem de entrada com tamanho de 64×64 é gerado um mapa de características com tamanho 64×64 . A utilização de *strides* maiores que 1 é aplicada em casos nos quais se deseja diminuir o tempo de execução e o tamanho das imagens. Sendo assim, o valor de *stride* igual a 2 para uma imagem de entrada de 64×64 gera como saída um mapa de características de tamanho 32×32 .

Portanto, de forma geral, a operação de convolução sobre a saída da camada anterior resulta em n_{cr} mapas de características bidimensionais, cada um com dimensões $d_{cr} \times d_{cr}$, tal que:

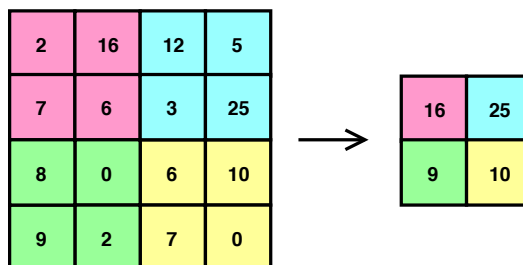
$$d_{cr} = \frac{n_{ce} - n_f + 2p_a}{s} + 1. \quad (2.1)$$

Na saída da camada de convolução é aplicada uma função de ativação nos neurônios que objetiva realizar uma transformação nos dados recebidos (FERREIRA, 2017). Normalmente, são utilizadas funções não lineares e parcialmente não lineares, como a função ReLu (*Rectified Linear Unit*) que é definida por $a(x) = \max(0, x)$ e tem como vantagem a redução no tempo de ajuste dos pesos quando comparada com funções sigmoidais tradicionais.

Em geral, uma das vantagens da utilização das camadas convolucionais é o fato de ser necessário definir apenas a arquitetura dos filtros, ou seja, a quantidade, o tamanho e o *stride* por camada, não havendo a necessidade de definir qual filtro deve ser utilizado, diferentemente do processo tradicional de Visão Computacional, onde a arquitetura parte da definição dos filtros que serão aplicados.

Outra camada muito importante, comumente utilizada após as camadas de convolução e ativação, é a camada de agrupamento (*pooling*). De forma geral, uma operação de *pooling* calcula uma norma específica dos valores de ativação produzidos por uma pequena região de cada canal. O objetivo desse cálculo é diminuir os custos computacionais através da redução do tamanho de cada canal da camada anterior, computando o máximo local de uma determinada região do mapa de características (MARQUES et al., 2018). A Figura 5 apresenta um mapa de características resultante da operação de *max pooling* com filtros de tamanho 2×2 e *stride* igual a 2.

Figura 5: Operação de agrupamento (*pooling*).



Fonte: A autora (2019).

A redução da dimensionalidade de um mapa de características feita pela aplicação do *pooling* traz, dentre outras vantagens, agilidade no treinamento e invariância espacial.

Além disso, vale ressaltar que a camada de *pooling* não reduz a profundidade da entrada, ela apenas reduz a altura e a largura de um mapa.

Em geral, em uma CNN, vários estágios de convolução, não-linearidade e *pooling* são empilhados, seguidos por mais camadas de convolução, e camadas completamente conectadas ao final da arquitetura para realizar tarefas de classificação ou regressão.

Existem outras variáveis que são utilizadas para auxiliar o processamento geral de uma CNN, como o processo de normalização (*normalization* ou *batch normalization*), que é utilizado para acelerar o processo de convergência e reduzir as chances de ficar preso no mínimo local (PENHA, 2018). Esse procedimento normaliza os sinais de saída da camada utilizando operações lineares e não lineares, aumentando a eficiência durante a aplicação da transformação.

Para minimizar o efeito de *overfitting*, a variável de *dropout* (ou eliminação) pode ser aplicada, já que ela realiza uma transformação semelhante a uma máscara fixa que torna uma parte das células como zero (salvando a célula para uso futuro), fazendo com que o cálculo do gradiente seja ignorado por essas células durante algumas épocas (MARQUES et al., 2018). Após esse período, outra máscara aleatória é aplicada, eliminando células de forma aleatória.

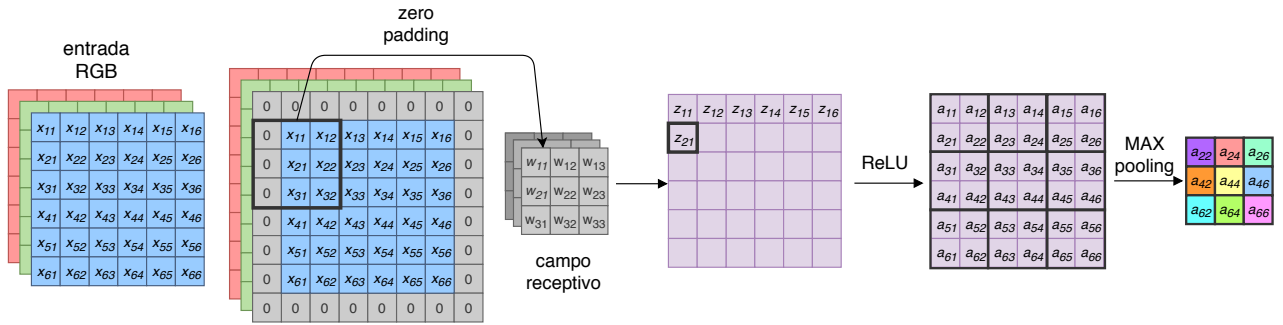
Por fim, uma camada totalmente conectada (*fully-connected* - FC) é adicionada para aprender as combinações não lineares das características de alto nível representados pela saída da camada convolucional (SAHA, 2018). Nessa camada, cada célula da saída é calculada através da soma ponderada de todas as células de todos os canais da camada anterior.

Normalmente, a camada FC objetiva nivelar a imagem processada em um vetor que servirá de entrada para uma rede neural profunda. Durante uma série de épocas, o modelo é capaz de distinguir entre características dominantes e certas características de baixo nível em imagens e classificá-las utilizando a função de classificação *softmax* que é aplicada na saída da rede neural. Na função *softmax*, a saída para cada rótulo é exponenciada e, em seguida, dividida pela soma da saída de todos os rótulos para fazer com que essa saída varie entre zero e um, expressando o grau de participação da amostra analisada em cada classe para problemas de classificação.

Para demonstrar o funcionamento da arquitetura geral de CNN baseada nos conceitos apresentados até então, é exibido um esquema na Figura 6 que exemplifica a ligação entre as camadas consecutivas de uma CNN, utilizando a função ReLu e a operação de *pooling*.

A primeira coluna da Figura 6 mostra um exemplo de uma entrada (6×6) com 3 canais. A partir da segunda coluna é demonstrado a interação dessa entrada com um campo receptivo (3×3), representado na terceira coluna, gerando uma saída (6×6) com 1

Figura 6: Operação em uma camada convolucional com ReLU e *Pooling*.



Fonte: A autora (2019).

canal na quarta coluna. A segunda coluna mostra *padding* e *stride* com valor 1. A quinta coluna representa a saída de uma operação ReLU. A última coluna apresenta a saída 3×3 para uma operação max pooling de janela 2×2 (representada na sexta coluna).

Baseados nos princípios das CNNs foram desenvolvidas diversas arquiteturas que têm sido fundamentais para a construção de algoritmos que sejam eficazes na área de IC (PONTI; COSTA, 2017). Entre os algoritmos mais conhecidos e utilizados vale ressaltar a arquitetura VGG, as Redes Residuais (ResNet), as Redes Neurais Profundas com Transformações Residuais Agregadas (ResNetXt) e as Redes Neurais Densas (DenseNets).

2.2.1.1 VGG

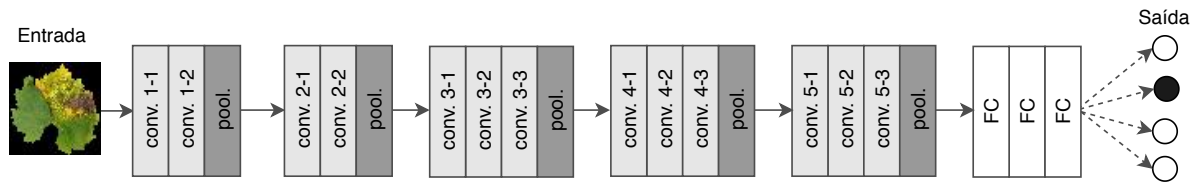
As CNNs se tornaram extremamente úteis em diversas tarefas como a categorização de imagens e frames de vídeos e, devido ao aumento da acurácia desses algoritmos, diversos pesquisadores e empresas estavam interessados em reduzir a complexidade e melhorar a eficiência das primeiras arquiteturas existentes.

Nesse cenário, a arquitetura de CNN denominada VGG, desenvolvida por Liu e Deng (2015), foi criada com o objetivo de investigar o efeito da adição de profundidade na rede convolucional no que tange sua precisão quando aplicada ao reconhecimento de imagens em larga escala, demonstrando uma melhoria significativa nas configurações cuja profundidade das camadas correspondia a 16 e 19, gerando as variantes VGG16 e VGG19 que atualmente são amplamente utilizadas na literatura para os mais variados fins na área de Visão Computacional.

Originalmente, a imagem de entrada possui um tamanho pré-definido de 224×224 , sendo em seguida passada através de uma pilha de camadas convolucionais com filtros cujo tamanho do campo receptivo corresponde a 3×3 , todas seguidas de camadas convolucionais de tamanho 2×2 que são responsáveis por realizar um agrupamento espacial, além de três camadas FC, que representam cerca de 89% de todos os parâmetros do modelo, além da camada de saída com função *softmax* (GOPALAKRISHNAN et al., 2017).

A função de ativação ReLu é aplicada em todas as camadas ocultas e o modelo utiliza a regularização *dropout* nas camadas FC. Um esquema da arquitetura VGG16 é mostrado na Figura 7.

Figura 7: Arquitetura VGG16 aplicada ao problema de classificação de doenças em plantas.



Fonte: A autora (2019).

Em geral, um conjunto de convoluções é aplicado nessa rede seguido das não linearidades e de uma camada de *pooling*. Na saída da última camada de *pooling* os mapas de características são redimensionados na forma de um vetor e utilizados para classificação com as últimas camadas tradicionais seguidas da função *softmax*. A largura da rede começa com um valor pequeno de 64 e aumenta em um fator de 2 após cada camada de subamostragem/*pooling* (LIU; DENG, 2015). A Tabela 1 apresenta as configurações detalhadas da VGG de 16 camadas.

A VGG foi a primeira CNN a utilizar filtros pequenos (3×3) em cada camada convolucional (ARAÚJO et al., 2017). Isso era contrário aos princípios das suas antecessoras, LeNet e AlexNet, onde filtros grandes (9×9 e 11×11) eram utilizados para capturar características similares na imagem. A grande contribuição da VGG foi a ideia de que múltiplas convoluções (3×3) em sequência poderiam substituir os efeitos de filtros de máscaras maiores (5×5 e 7×7) que resultavam em maior custo computacional.

Outro ponto a ressaltar é o fato desse modelo possuir uma arquitetura muito uniforme, o que a torna uma rede de destaque na literatura. Atualmente, é uma das escolhas preferidas na comunidade para extrair características das imagens devido, entre outros fatores, ter seus pesos pré-treinados em diversas bases de dados disponíveis publicamente, o que facilita seu treinamento para uma ampla gama de aplicações. Contudo, o fato dessa rede possuir cerca de 138 milhões de parâmetros na versão de 16 camadas convolucionais faz com que haja um aumento no custo computacional quando comparada a suas sucessoras ResNet e DenseNet.

2.2.1.2 RESNET

As redes neurais profundas são capazes de aprender funções e representações mais complexas da entrada com um melhor desempenho que as redes tradicionais, desde que hajam meios eficazes para o treinamento (HUANG et al., 2017). No entanto, diversas

Tabela 1: Arquitetura da VGG16.

Camadas	Dim. Saída	VGG16
Bloco 1: conv1	224×224	3×3, 64, <i>stride</i> 1
Bloco 1: conv2	224×224	3×3, 64, <i>stride</i> 1
Bloco 1: <i>pooling</i>	112×112	3×3 <i>max. pool.</i>, <i>stride</i> 2
Bloco 2: conv1	112×112	3×3, 128, <i>stride</i> 1
Bloco 2: conv2	112×112	3×3, 128, <i>stride</i> 1
Bloco 2: <i>pooling</i>	56×56	3×3 <i>max. pool.</i>, <i>stride</i> 2
Bloco 3: conv1	56×56	3×3, 256, <i>stride</i> 1
Bloco 3: conv2	56×56	3×3, 256, <i>stride</i> 1
Bloco 3: conv3	56×56	3×3, 256, <i>stride</i> 1
Bloco 3: <i>pooling</i>	28×28	3×3 <i>max. pool.</i>, <i>stride</i> 2
Bloco 4: conv1	28×28	3×3, 512, <i>stride</i> 1
Bloco 4: conv2	28×28	3×3, 512, <i>stride</i> 1
Bloco 4: conv3	28×28	3×3, 512, <i>stride</i> 1
Bloco 4: <i>pooling</i>	14×14	3×3 <i>max. pool.</i>, <i>stride</i> 2
Bloco 5: conv1	14×14	3×3, 512, <i>stride</i> 1
Bloco 5: conv2	14×14	3×3, 512, <i>stride</i> 1
Bloco 5: conv3	14×14	3×3, 512, <i>stride</i> 1
Bloco 5: <i>pooling</i>	7×7	3×3 <i>max. pool.</i>, <i>stride</i> 2
<i>Fully-connected</i> 1	1×1	4096D
<i>Fully-connected</i> 2	1×1	4096D
		<i>max. pool.</i>, 1000D <i>fully-connected</i>, <i>softmax</i>

pesquisas apontaram que o aumento na profundidade das CNNs resultaram em um novo problema, as informações sobre a entrada ou gradiente poderiam “desaparecer” conforme passavam pelas inúmeras camadas no momento em que chegavam ao início ou fim da rede, resultando no problema conhecido como gradiente de desaparecimento. O referido problema dificulta a convergência desse tipo de rede, tornando a precisão saturada e ocasionando o efeito de degradação.

Uma das soluções encontradas para o efeito de degradação foi a adição de blocos residuais nas camadas de uma arquitetura tradicional de CNN para aprender funções residuais com referência às entradas da camada, ao invés de aprender funções não referenciadas (JORDAN, 2018), facilitando o treinamento de redes que são substancialmente mais profundas do que as outras arquiteturas.

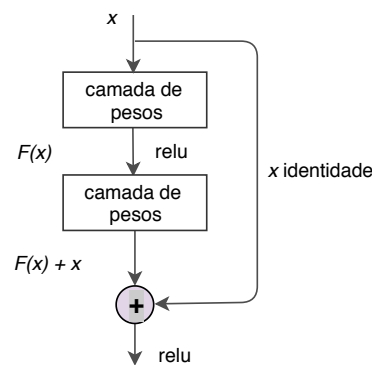
O trabalho de He et al. (2016) foi o primeiro a propor uma estrutura de aprendizado residual profundo para minimizar o problema da degradação de dados e o desaparecimento do gradiente durante a retropropagação (*backpropagation*), gerando as redes neurais residuais, também conhecidas como ResNets.

Com a introdução das conexões residuais, o modelo não mapeia as entradas das camadas anteriores diretamente para camadas de convolução, mas sim para as camadas residuais que funcionam como funções não-lineares das camadas convolutivas. Segundo

Fonseca (2017), é mais simples otimizar o mapeamento residual do que otimizar o mapeamento original não referenciado.

A arquitetura de uma ResNet é formada por camadas tradicionais e camadas residuais que são compostas basicamente de duas camadas de convolução com uma função de ativação ReLU entre elas. A saída da camada residual é a função ReLU da sua própria entrada acrescentada da saída da segunda camada intermediária (ARAÚJO et al., 2017). A ideia por trás das camadas residuais é que uma entrada x passa por uma série de operações de convolução-relu-convolução, como exemplifica a Figura 8. O resultado da operação $F(x)$ é adicionado à entrada original x , tal que $H(x) = F(x) + x$.

Figura 8: Arquitetura de um bloco residual.

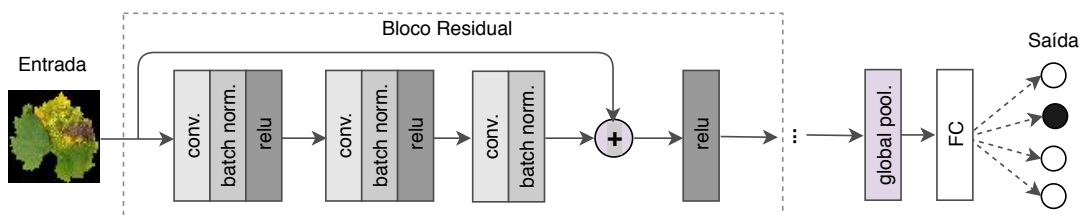


Fonte: A autora (2019).

Nas CNNs tradicionais, o resultado de processamento $H(x)$ é igual $F(x)$, alterando completamente o espaço de saída em relação ao espaço de entrada x . Já na arquitetura ResNet, a função $F(x)$ funciona somente como um termo de regularização (ARAÚJO et al., 2017). Portanto, o espaço de saída é somente uma alteração do espaço de entrada.

A Figura 9 apresenta um esquema genérico do modelo ResNet aplicado no problema de classificação de doenças de plantas. É importante notar a presença de um mecanismo de *pooling* global antes das camadas FC.

Figura 9: Arquitetura ResNet aplicada ao problema de classificação de doenças em plantas.



Fonte: A autora (2019).

De uma maneira geral, as ResNets são inspiradas principalmente pela filosofia das redes VGG, pois suas camadas convolucionais são formadas principalmente de filtros 3×3 e seguem duas regras de projeto simples: (i) para o mesmo tamanho do mapa de características da saída, as camadas têm o mesmo número de filtros; e (ii) se o tamanho do mapa de características for reduzido pela metade, o número de filtros é duplicado para preservar a complexidade de tempo por camada.

A Tabela 2 apresenta a configuração das principais variações desse modelo de CNN, considerando uma entrada de dimensão 112×112 e variando em quantidades de 50, 101 e 152 camadas.

Tabela 2: Configuração das arquiteturas de ResNet com 50, 101 e 152 camadas.

Camadas	Dim. Saída	50	101	152
conv1	112×112	$7 \times 7, 64, \text{stride } 2$		
pooling	56×56	$3 \times 3 \text{ max. pool., stride } 2$		
conv2	56×56	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3	28×28	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4	14×14	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5	7×7	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	<i>avg. pool., 1000D fully-connected, softmax</i>		

2.2.1.3 RESNETXT

A ResNetXt foi criada com o objetivo de ser uma arquitetura convolucional simples e altamente modularizada para a classificação de imagens. Essa CNN opera repetindo um bloco residual composto por um conjunto de transformações. Seu projeto simples resulta em um modelo homogêneo, com várias ramificações, que necessita da definição de uma quantidade significativamente menor de hiperparâmetros quando comparado a outros modelos de CNN (XIE et al., 2017). Em geral, as pilhas de blocos residuais que compõem esse modelo possuem a mesma topologia e estão sujeitos à duas regras simples que foram inspiradas principalmente em preceitos das arquiteturas VGG e ResNet:

1. Ao produzir mapas espaciais do mesmo tamanho, os blocos compartilham os mesmos hiperparâmetros (largura e tamanho do filtro);

2. Toda vez que o mapa espacial é reduzido por um fator de 2, a largura dos blocos é multiplicada por um fator de 2.

A segunda regra garante que a complexidade computacional, em termos de operações convolucionais seja aproximadamente a mesma para todos os blocos. A principal vantagem dessa abordagem envolve a redução na complexidade da arquitetura já que se faz necessária a criação de apenas um módulo, enquanto todos os outros podem ser criados baseados nele.

Tradicionalmente, os neurônios mais simples das redes neurais realizam produto interno (soma ponderada), que é a transformação elementar feita pelas camadas FC e as camadas convolucionais. No modelo ResNetXt, essa operação é substituída por uma função mais genérica que, por si só, também pode ser uma rede (XIE et al., 2017). Ao contrário dos modelos tradicionais cuja transformação aumenta a dimensão da profundidade, essa rede expande ao longo de uma nova dimensão, denominada de “cardinalidade”, gerando um paradigma orientado ao neurônio (rede-em-neurônio). Formalmente, as transformações agregadas são dadas por:

$$f(x) = \sum_{i=1}^C \tau_i(x), \quad (2.2)$$

onde τ_i representa uma função arbitrária que, de forma análoga a um neurônio simples, deve projetar x em um novo espaço de características (opcionalmente de baixa dimensão) e depois transformá-lo. C é o tamanho do conjunto de transformações a serem agregadas, ou seja, a cardinalidade.

De acordo com Xie et al. (2017), enquanto a dimensão da largura está relacionada ao número de transformações simples (produto interno), a dimensão da cardinalidade controla o número de transformações mais complexas, sendo uma dimensão que demonstrou ser mais eficaz do que as dimensões de largura e profundidade em casos práticos.

Em geral, a fim de simplificar a projeção das funções de transformação, o valor de todos os τ_i possuem a mesma topologia. Isso estende a estratégia de repetir camadas da mesma forma que o modelo VGG, sendo útil para isolar alguns fatores e aumentar a quantidade de transformações (XIE et al., 2017). Além disso, a transformação também serve como uma função residual, dada por:

$$H(x) = \sum_{i=1}^C \tau_i(x) + x. \quad (2.3)$$

A Tabela 3 apresenta uma comparação resumida entre a configuração de um modelo de ResNetXt, com cardinalidade igual a 32, com uma arquitetura de ResNet tradicional, ambas com 50 camadas, considerando uma entrada de dimensão 112×112 . Observa-se

que a largura de entrada/saída do modelo ResNetXt é fixo como 256 e todas as larguras são dobradas cada vez que o mapa de características é subamostrado.

Tabela 3: Comparação das arquiteturas ResNet e ResNetXt com 50 camadas.

Camadas	Dim. Saída	ResNet50	ResNetXt50
conv1	112×112	7×7, 64, stride 2	
pooling	56×56	3×3 max. pool., stride 2	
conv2	56×56	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128, C=32 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3	28×28	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256, C=32 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
conv4	14×14	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512, C=32 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$
conv5	7×7	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 1024 \\ 3 \times 3, 1024, C=32 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	avg. pool., 1000D fully-connected, softmax	

Essa arquitetura adota a estratégia de repetir camadas, como os modelos VGG e ResNet, enquanto explora a estratégia de divisão, transformação e mesclagem de uma maneira simples e extensível. Um módulo dessa rede realiza um conjunto de transformações, cada uma num espaço de características de baixa dimensão, cujas saídas são agregadas por soma.

A cardinalidade é uma dimensão concreta e mensurável de suma importância, além das dimensões de largura e profundidade, que possibilita aumentar consideravelmente a quantidade de transformações sem projetos especializados. As experiências apresentadas na pesquisa de Xie et al. (2017) demonstraram que o aumento da cardinalidade é uma maneira mais eficaz de obter precisão do que aprofundar ou ampliar a rede, especialmente quando profundidade e largura começam a fornecer retornos decrescentes para os modelos de CNN existentes.

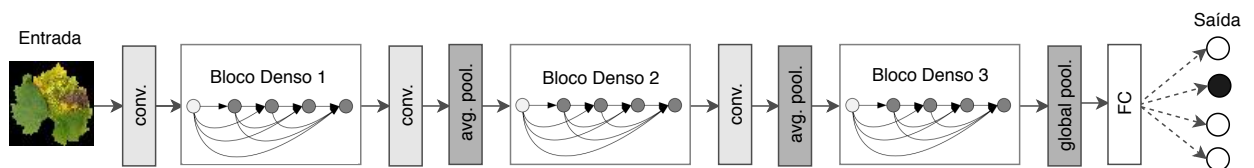
2.2.1.4 DENSENET

Estudos recentes mostraram que as redes convolucionais podem ser substancialmente mais profundas, mais precisas e eficientes de treinar se possuírem conexões mais curtas entre as camadas próximas à entrada e as camadas próximas à saída. A DenseNet ou CNN densa utiliza um padrão de conectividade simples que conecta cada camada a todas as outras camadas da rede com o objetivo de garantir o máximo fluxo de informações (HUANG et al., 2017). Logo, cada camada obtém entradas adicionais de todas as

camadas anteriores e passa seus próprios mapas de características para todas as camadas subsequentes, preservando a natureza *feedforward*.

A Figura 10 representa de forma simplificada o funcionamento na DenseNet aplicada a um problema de classificação. As camadas entre dois blocos adjacentes são referidas como camadas de transição e alteram os tamanhos dos mapas de características por meio de convolução e *pooling*.

Figura 10: Arquitetura DenseNet aplicada ao problema de classificação de doenças em plantas.



Fonte: A autora (2019).

A partir da literatura, as principais vantagens percebidas através da utilização das DenseNets aplicadas nas tarefas de reconhecimento de padrões incluem: redução do problema do gradiente de desaparecimento, aumento na propagação e reutilização de características, e redução significativa da quantidade de parâmetros. Nessas CNNs não há necessidade de reaprender mapas de características redundantes, exigindo menos computação para obter um alto desempenho.

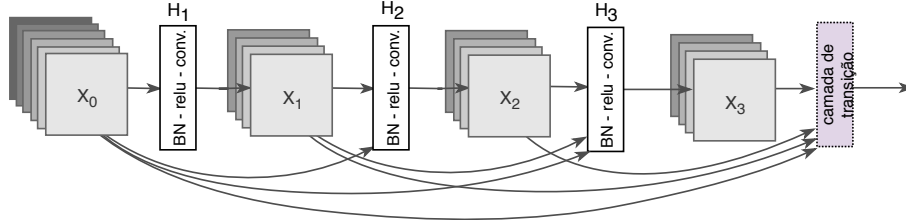
Além de melhorias na eficiência de compartilhamento de parâmetros, uma grande vantagem desse modelo é o fluxo aprimorado de informações e gradientes em toda a rede, o que facilita seu treinamento. Segundo Lee et al. (2015), arquiteturas nas quais cada camada tem acesso direto aos gradientes da função de perda e do sinal de entrada original levam a uma supervisão profunda implícita. Isso ajuda no treinamento de arquiteturas de rede mais profundas. Além disso, os criadores da DenseNet observaram que as conexões densas tinham um efeito de regularização, o qual reduzia o ajuste excessivo em tarefas com conjuntos de treinamento de tamanhos menores.

As camadas da DenseNet são muito estreitas (por exemplo, 12 filtros por camada), adicionando apenas um pequeno conjunto de mapas de características ao “conhecimento coletivo” da rede e mantendo os demais mapas de características inalterados (HUANG et al., 2017). O classificador final toma uma decisão baseada em todos os mapas de características da rede.

A Figura 11 ilustra o funcionamento da DenseNet de forma esquemática. X_0 representa a imagem de entrada passada por uma rede convolucional de L camadas, sendo que cada camada l implementa uma transformação não-linear $H_l(\cdot)$. $H_l(\cdot)$ é uma função composta de três operações consecutivas: Normalização de lote (BN), ativação ReLU e

operação de *Pooling* ou Convolução (Conv). A saída da camada final (transição) l^{final} é representada por X_l que consequentemente recebe os mapas de características de todas as camadas anteriores, X_0, \dots, X_{l-1} , como entrada.

Figura 11: Um bloco denso de 4 camadas com uma taxa de crescimento de $t = 3$.



Fonte: A autora (2019).

$$X_l = H_l([X_0, X_1, \dots, X_{l-1}]), \quad (2.4)$$

onde, $[X_0, X_1, \dots, X_{l-1}]$ refere-se à concatenação dos mapas de características produzidos nas camadas $0, \dots, l - 1$. Se cada função H_l produz n_{cr} mapas de características, logo a camada l^{final} tem uma camada de entrada de $n_{ca} + t \times (l - 1)$ mapas de características, onde n_{ca} é o número de canais na camada de entrada. O hiperparâmetro t representa a taxa de crescimento da rede que é responsável por regular quanta informação nova cada camada contribui para o estado global. De acordo com Huang et al. (2017), o estado global, uma vez escrito, pode ser acessado de qualquer lugar dentro da rede e, ao contrário das arquiteturas de rede tradicionais, não há necessidade de replicá-lo de uma camada para outra.

Além disso, a quantidade de mapas de características nas camadas de transição foi reduzida com o objetivo de melhorar a compactação desse modelo. A Tabela 4 apresenta a configuração das principais variações da DenseNet, variando em quantidades de 121, 169 e 201 camadas. A taxa de crescimento para todas as redes é $t = 32$ e cada camada convolucional corresponde à sequência BN-ReLU-Conv.

Ao contrário das ResNets, as DenseNets não combinam características por meio do somatório antes de serem passados para uma camada, ao invés disso, combinam características através da concatenação (HUANG et al., 2017). Portanto, a camada l^{final} consiste nos mapas de características de todos os blocos convolucionais anteriores. Seus próprios mapas de características são passados para todas as $L - l$ camadas subsequentes. Isso introduz $L(l + 1)/2$ conexões em uma CNN de L -camadas, diferente de apenas L , como nas arquiteturas tradicionais.

Tabela 4: Configuração das arquiteturas de DenseNet com 121, 169 e 201 camadas.

Camadas	Dim. Saída	121	169	201
Convolução	112×112	7×7 conv, stride 2		
Pooling	56×56	3×3 max pool, stride 2		
Bloco Denso 1	56×56	$\begin{matrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{matrix} \times 6$	$\begin{matrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{matrix} \times 6$	$\begin{matrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{matrix} \times 6$
Camada de Transição 1	56×56	1×1 conv		
	28×28	2×2 avg. pool., stride 2		
Bloco Denso 2	28×28	$\begin{matrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{matrix} \times 12$	$\begin{matrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{matrix} \times 12$	$\begin{matrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{matrix} \times 12$
Camada de Transição 2	28×28	1×1 conv		
	14×14	2×2 avg. pool., stride 2		
Bloco Denso 3	14×14	$\begin{matrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{matrix} \times 24$	$\begin{matrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{matrix} \times 32$	$\begin{matrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{matrix} \times 48$
Camada de Transição 3	14×14	1×1 conv		
	7×7	2×2 avg. pool., stride 2		
Bloco Denso 4	7×7	$\begin{matrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{matrix} \times 16$	$\begin{matrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{matrix} \times 32$	$\begin{matrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{matrix} \times 32$
Camada de Classificação	1×1	7×7 global average pool.		
		1000D fully-connected, softmax		

2.2.2 Treinamento em Redes Neurais Convolucionais

Em uma CNN, os parâmetros dos neurônios (pesos) são estimados através do treinamento do modelo utilizando o aprendizado supervisionado em uma quantidade significativa de dados de entrada para que sua capacidade de generalização seja eficaz quando aplicada a entradas as quais nunca foi exposta. Em geral, o processo de treinamento consiste basicamente nas etapas de *forwardpass* e *backpropagation*, respectivamente.

Na etapa de *forwardpass*, a rede faz uma previsão com base nos dados fornecidos na entrada. Em outras palavras, consiste na passagem do volume da entrada pelas camadas da rede. Nesse processo, diversas operações típicas são realizadas levando em consideração os pesos já aprendidos ou inicializados aleatoriamente. Os resultados obtidos pelo *forwardpass* de cada camada são então utilizados na etapa de *backpropagation* para a efetivação da atualização dos parâmetros da rede.

No *backpropagation*, a rede aprende com seu erro calculando o gradiente de uma função de perda e, em seguida, utiliza o gradiente para atualizar os pesos dos neurônios. A otimização da função de perda determina a qualidade da classificação dos dados de entrada. Atualmente, esse tipo de otimização é geralmente realizado utilizando um método chamado Gradiente Descendente Estocástico (*Stochastic Gradient Descent* - SGD) que busca a minimização da função de perda ao alterar os pesos na direção de maior declive do gradiente dessa função (PENHA, 2018). Existem ainda vários outros algoritmos de otimização que podem ser utilizados no treinamento da rede, como RMSProp

(SCHMIDHUBER, 2015) e Adam (KINGMA; BA, 2015).

Em geral, o processo de aprendizado da CNN altera os pesos ao longo do treinamento até encontrar os melhores valores, de acordo com os filtros, para o conjunto de dados utilizado. Uma das formas de se avaliar o aprendizado é verificar as ativações dos filtros obtidos no final do treinamento. Em CNNs aplicadas à detecção de objetos, pode-se perceber que as primeiras camadas costumam aprender filtros de arestas, bordas e cores (VARGAS; PAES; VASCONCELOS, 2016).

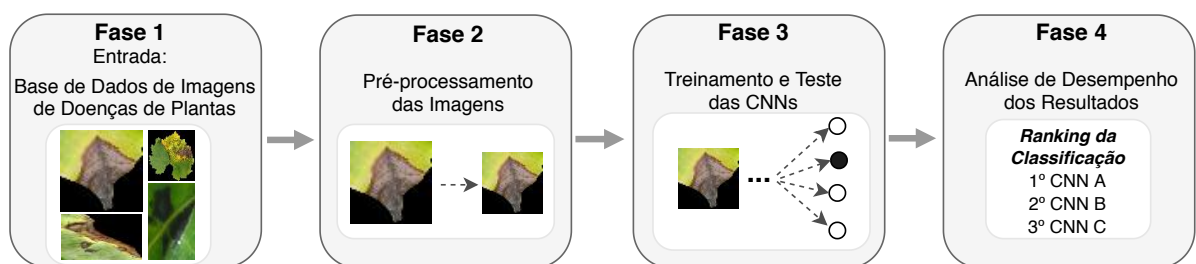
Nos níveis mais profundos de uma CNN, partes de interesse e detalhes cada vez mais complexos são detectados, fazendo com que a rede produza uma hierarquia de características (VARGAS; PAES; VASCONCELOS, 2016). Outra característica notável nas CNNs é a capacidade de criar filtros. Ao longo do treinamento, é possível definir filtros com largura, altura e um parâmetro de profundidade atravessando múltiplos canais. Filtros clássicos de processamento de imagens dificilmente possuem tal característica, pois a maior parte deles é 2D.

Além disso, para que um conjunto de neurônios de um determinado mapa aplique o mesmo filtro em diferentes posições da imagem, é feito um compartilhamento dos pesos durante o processo de treinamento, o que reduz significativamente o número de parâmetros a serem aprendidos, assim como o tempo de treinamento da rede.

3 Materiais e Métodos

Visando alcançar o objetivo de desenvolver uma metodologia para a classificação de variadas doenças de espécies de plantas distintas, foram desenvolvidas subatividades categorizadas em quatro fases discriminadas através do fluxo geral descrito na Figura 12.

Figura 12: Metodologia para a classificação de imagens de doenças em plantas.



Fonte: A autora (2019).

A primeira fase envolveu a seleção da base de dados de doenças que acometem plantas como entrada para as CNN que foram utilizadas. Contudo, para tornar o treinamento das CNNs possível, se fez necessária a inclusão de uma fase intermediária, denominada como pré-processamento das imagens, que foi responsável por padronizar a dimensão de todas as imagens da base de dados selecionada, além de eliminar imagens com dimensões extremamente pequenas que porventura pudessem compor a base de dados em questão. Após essa padronização, deu-se início a fase de treinamento e teste das CNNs que de fato realizou a extração de características relevantes que tornaram possível a classificação das patologias. Por fim, a fase de análise de desempenho dos resultados objetivou avaliar a saída das CNNs utilizadas e classificar as melhores redes através de um *ranking* baseado em estudos estatísticos não-paramétricos.

Nas próximas seções serão apresentadas as formas como as fases representadas na Figura 12 foram de fato implementadas com maior especificidade.

3.1 Base de Dados de Doenças em Plantas XDB

3.1.1 Digiphatos Embrapa

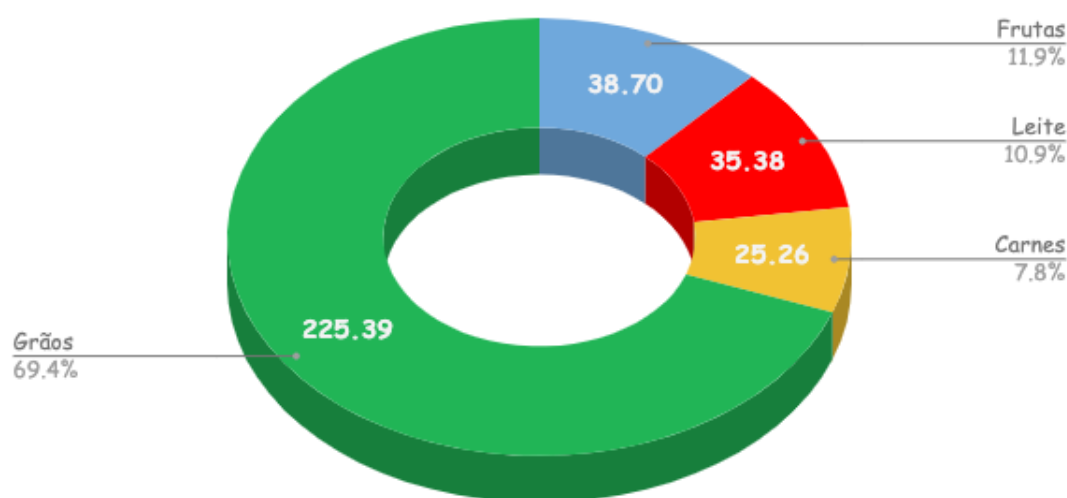
Com a abertura para o mercado externo e o constante crescimento da renda per capita e da população brasileira, surgiu a necessidade em aumentar significativamente

os investimentos em ciência e inovação agrícola para reduzir o abismo entre a oferta e a demanda de alimentos. Nesse cenário, a partir da década de 1970, o Ministério da Agricultura propôs a criação de métodos de pesquisa agropecuária com o objetivo de fortalecer a geração de conhecimentos técnico-científicos do setor. Logo, para suprir essa necessidade, em 26 de abril de 1973 foi criada a Empresa Brasileira de Pesquisa Agropecuária, mais conhecida como Embrapa (BOLFE, 2018).

As pesquisas feitas pela Embrapa, em parceria com instituições nacionais e internacionais, por meio de cooperações técnicas e científicas, possibilitaram uma evolução exponencial da agricultura, assim como as inovações e tecnologias que têm sido responsáveis por cerca de 60% do aumento da produção agrícola no Brasil. Bolfe (2018) aponta que, o crescimento da produção de grãos aumentou em cinco vezes, passando de 38 milhões de toneladas em 1975 para 236 milhões em 2017, com apenas o dobro da área plantada, representando um aumento na produção de alimentos com uma menor quantidade de recursos naturais gastos.

Nos últimos anos, o agronegócio tem representado cerca de 20% do PIB brasileiro. O gráfico da Figura 13 apresenta a produção anual referente ao ano de 2018 desse setor, levando em consideração os grãos, frutas, leite e carnes, no qual se pode observar a expressividade agrícola que corresponde a 81,3% da produção total. Em geral, a agricultura brasileira é baseada em mais de 300 espécies de cultivos e envia ao mundo 350 tipos de produtos que chegam a cerca de 200 mercados do planeta. O Brasil é o maior exportador mundial de soja, café, açúcar, suco de laranja, etanol de cana-de-açúcar, carne bovina e de frango, o segundo maior produtor mundial de soja e o terceiro maior de frutas (DUARTE; ESCOBAR; SILVA, 2019).

Figura 13: Produção Agrícola Anual Brasileira em 2018 (Estimada em Milhões de Toneladas).



Fonte: Adaptado de IBGE - Ceplac, Conab, CNA Brasil apud Duarte, Escobar e Silva (2019).

Nesse cenário, devido a importância da saúde alimentar brasileira, assim como a renda gerada pela exportação de alimentos e a agricultura familiar, surgiu a necessidade em diagnosticar doenças em plantas para reduzir perdas e manter os lucros do setor que é comprovadamente essencial para a economia. Uma das soluções desenvolvidas pela Embrapa para facilitar o diagnóstico precoce das patologias que acometem essas culturas foi a criação e desenvolvimento de uma base de dados de imagens digitais dessas doenças disponível publicamente através do endereço eletrônico <https://www.digipathos-rep.cnptia.embrapa.br/>.

Contudo, os desafios envolvidos nessa empreitada são significantes, pois existe uma quantidade expressiva de desordens que podem acometer cada espécie de planta, e cada uma dessas desordens produz uma gama de sintomas que dependem de fatores ambientais e contextuais.

Apesar dos desafios citados, a base de dados de imagens denominada de PDDB (Base de Imagens de Sintomas de Doenças de Plantas) foi construída ao longo de quatro anos e envolveu diversos pesquisadores, além de estudantes e estagiários que trabalharam na coleta e organização da base de dados. Como resultado, foram coletadas 2.326 imagens de 171 doenças e outras desordens acometendo 21 espécies de plantas (BARBEDO et al., 2018).

Entretanto, como mencionado pelos pesquisadores, a base de dados PDDB não possui uma quantidade de imagens que viabilize a aplicação de técnicas de AP para a extração de características das doenças, uma vez que boa parte das classes não possui representatividade (BARBEDO et al., 2018). Baseado nesse problema, a fim de aumentar o tamanho da base de dados, as imagens foram divididas em imagens menores contendo lesões individuais ou regiões de sintomas, de acordo com alguns critérios, como a extração de folhas de plantas doentes. Como resultado, esta base de dados de imagens expandida (XDB) atualmente contém 46.513 imagens. O Apêndice C apresenta algumas imagens que fazem parte da base de dados XDB.

3.1.2 Pré-processamento da base de dados XDB

Antes de utilizar as imagens da base de dados XDB como entrada para as CNNs deste estudo, se fez necessária a realização de etapas de pré-processamento, pois para executar modelos de AP, é crucial que as imagens tenham um padrão e que possuam representatividade suficiente em cada classe de forma que as patologias possam ser devidamente classificadas, facilitando treinamento e, subsequente, teste das CNNs.

A princípio, a associação de um rótulo a imagem é um passo inevitável no aprendizado supervisionado. O método de anotação dos dados depende da abordagem geral escolhida para análise de imagens. Como este estudo objetiva classificar doenças, essa

etapa consistiu na associação de um rótulo a cada imagem e na posterior organização das imagens rotuladas em diretórios referentes às diferentes classes, respeitando a classificação realizada pelos especialistas que geraram a base de dados XDB. Após a anotação dos dados, o pré-processamento envolveu duas etapas principais: i) seleção de imagens e (ii) redimensionamento de imagens.

A etapa de seleção das imagens consistiu basicamente de uma validação que é justificada pelo fato da base de dados XDB ser composta por imagens de tamanhos diversos, sendo algumas destas extremamente pequenas, o que inviabiliza sua utilização em técnicas de AP. Nesse sentido, foi implementado um método para realizar a filtragem das imagens desejadas para a criação das dez bases de dados que foram utilizadas neste trabalho.

Essa validação foi baseada em um fator de proximidade da dimensão alvo, denominada taxa de diferença, que considerou a dimensão de entrada, comparando o tamanho da imagem atual com o tamanho alvo. Caso a imagem atual possuísse uma dimensão inferior ao limite mínimo definido pela taxa de diferença, a mesma seria eliminada do grupo de imagens que iria compor a base de dados com a dimensão alvo. A Tabela 5 apresenta o valor correspondente as taxas aplicadas a cada uma das dimensões estabelecidas para as bases de dados do presente trabalho.

Tabela 5: Taxa de diferença aplicada de acordo com cada dimensão.

Dimensão	Taxa de Diferença
32×32	0
40×40	5%
48×48	10%
56×56	15%
64×64	20%

Ao desconsiderar a taxa de diferença na montagem das bases de dados de dimensão 32×32, todas as imagens iguais ou maiores que essa dimensão foram consideradas. Essa metodologia foi adotada devido os modelos de CNNs aplicados utilizarem 32×32 como a dimensão mínima para suas entradas. Apesar de ser possível definir uma taxa de diferença e posteriormente redimensionar as imagens para a dimensão alvo desejada, nesse caso, foi observado que as imagens menores que 32×32 seriam extremamente pequenas e, como dito anteriormente, poderiam dificultar a precisão dos algoritmos aplicados.

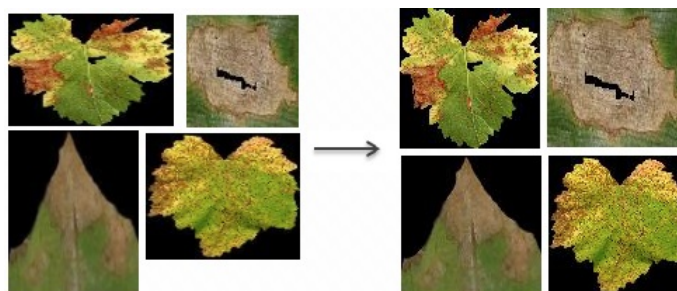
A partir da dimensão 40×40 até 64×64, a taxa de diferença variou com acréscimo de 5% a medida que as dimensões alvo foram aumentadas, significando que todas as imagens analisadas que possuíssem dimensão a partir de $d_{px \times px} - (d_{px \times px} \times t)$ seriam consideradas na montagem da base de dados em questão, onde $d_{px \times px}$ refere-se a dimensão alvo e t à taxa de diferença aplicada.

Tabela 6: Configurações das bases de dados.

Denominação	Dim. alvo	Qtd. Classes	Qtd. Imagens
bd32final	32×32	50	35.430
bd32hard	32×32	66	36.558
bd40final	40×40	50	32.043
bd40hard	40×40	65	33.049
bd48final	48×48	50	29.020
bd48hard	48×48	64	29.878
bd56final	56×56	50	27.079
bd56hard	56×56	62	27.786
bd64final	64×64	50	25.211
bd64hard	64×64	60	25.778

Após a execução dessa etapa, as imagens selecionadas prosseguiram para a etapa de redimensionamento que consistiu na padronização das dimensões aceitas na etapa de seleção para as dimensões alvo. A Figura 14 apresenta o funcionamento das etapas de pré-processamento de forma simplificada.

Figura 14: Exemplo das etapas de pré-processamento.



Fonte: A autora (2019).

Após a execução das etapas de pré-processamento foram geradas as dez bases de dados com configurações que podem ser observadas através da Tabela 6. Inicialmente, foram criadas bases de dados com dimensões que variavam entre si em 8 *pixels* com dimensões partindo de 32×32 até 64×64, todas com as 50 classes mais representativas, denominadas neste estudo com o sufixo “final”. Posteriormente, foram criadas bases de dados com quantidade de classes variando de acordo com as classes de maior representatividade em cada dimensão alvo, partindo de 60 classes na base de dados de 64×64 e chegando até 66 classes na base de dados de 32×32. Essas bases de dados seguiram o mesmo padrão anterior de dimensões e foram classificadas através do sufixo “hard”.

Devido ao fato das imagens da base de dados XDB serem imagens menores que representam de forma mais eficaz a doença, as bases de dados com imagens de dimensão 32×32 possuem um número maior de imagens, como esperado. Por outro lado, as base de dados com imagens de dimensão 64×64 são aquelas que agrupam o menor número de

imagens, apesar de possuírem a maior resolução. O Apêndice A apresenta a relação da quantidade de imagens de doenças em plantas por classe para a base de dados 64hard, enquanto o Apêndice B exibe a quantidade de imagens por classe referente a base de dados 32hard, ambos após a aplicação da etapa de pré-processamento.

3.2 Treinamento das Redes Convolucionais Utilizadas

3.2.1 Divisão do Conjunto de Dados

Ao utilizar CNNs, geralmente três conjuntos de dados são necessários para desenvolver um modelo: treinamento, validação e teste. O conjunto de treinamento é a coleção de imagens a serem utilizadas pela rede para aprender automaticamente seus parâmetros intrínsecos, e.g. os pesos. O conjunto de validação é utilizado para ajustar manualmente os hiperparâmetros que são essenciais às configurações que não podem ser aprendidas automaticamente durante o treinamento. Estes incluem, entre outros, taxa de aprendizagem, tamanho do lote (*batch size*), quantidade de épocas e arquitetura da rede. Os valores desses hiperparâmetros costumam ser definidos empiricamente, pois estão vinculados ao problema, ao conjunto de dados e à arquitetura do modelo.

Sendo assim, o modelo que está sendo treinado pode ser avaliado no conjunto de validação no final de cada época, permitindo que o processo de treinamento seja monitorado e possíveis *underfitting* e *overfitting* possam ser detectados. Geralmente, os conjuntos de treinamento e validação são provenientes da mesma fonte de dados que é subdividida.

Por sua vez, o conjunto de dados de teste é utilizado somente quando a fase de treinamento é concluída, com o objetivo de avaliar a capacidade de generalização final do modelo. A precisão no conjunto de teste é, portanto, uma das métricas mais importante a ser computada, pois fornece uma visão geral do desempenho do modelo além do processo de exploração do hiperparâmetro.

Especificamente para este estudo, não haverá distinção entre os conjuntos de dados de validação e teste, pois não há pretensão de otimizar os hiperparâmetros das CNNs através do conjunto de validação, já que a maioria foi fixada ou variada empiricamente a partir de trabalhos da literatura. Sendo assim, para cada base de dados (ver Tabela 6), 70% das imagens foram utilizadas na etapa de treinamento dos modelos, enquanto os 30% restantes foram aplicados para avaliação (validação/teste) dos modelos treinados em todas as arquiteturas de CNNs utilizadas.

3.2.2 Técnicas de Regularização Utilizadas

O principal desafio no aprendizado de máquina é obter um modelo treinado capaz de analisar dados novos e não vistos pela rede. Este aspecto está longe de ser garantido pela

alta precisão do treinamento. De fato, a principal armadilha na aprendizagem profunda é o excesso de preparo. Isso ocorre quando o número de amostras de entrada é muito pequeno em comparação com a capacidade de aprendizado da rede. O *overfitting* não permite aprender as características gerais das classes e, em vez disso, capta o ruído do conjunto de treinamento, levando a um modelo que, embora possua alta precisão durante o treinamento, não é capaz de generalizar, não alcançando alta precisão na etapa de teste da rede (BOULENT et al., 2019).

Para contornar esse problema, esse estudo aplicou técnicas de regularização que são amplamente utilizadas para melhorar a capacidade de generalização dos modelos de redes convolucionais utilizados, reduzindo o *overfitting*. Geralmente, o primeiro passo de regularização envolve reunir mais dados, ou seja, aumentar a quantidade de imagens de patologias em plantas. Contudo, obter muitas imagens para uma determinada classe pode ser complicado em um contexto de doenças em plantas, especialmente devido a base de dados XDB ainda estar em processo de construção.

Em virtude disso, este estudo utilizou técnicas de aumento de dados baseados em imagens já existentes somente no conjunto destinado ao treinamento dos modelos. Essas técnicas consistem na transformação da geometria ou intensidade das imagens originais para fazê-las parecerem novas imagens, sendo compostas de operações como rotação, espelhamento, corte aleatório, zoom ou até mesmo adição de ruído, mudança no contraste e nos valores de brilho, além de simular novos planos de fundo, aumentando o tamanho do conjunto de dados artificialmente. As operações de aumento podem ser executadas de diferentes maneiras, utilizando uma ou várias transformações por imagem. Além disso, as transformações também podem ser aplicadas antes do início de todo o processo de treinamento ou quando cada lote de imagens é carregado. Todos os efeitos de aumento de dados foram realizados utilizando *framework* Keras.

No presente trabalho, foi desenvolvido um método automático para aumentar a quantidade de imagens no conjunto de treinamento que utilizou a aplicação de efeitos como giro horizontal, deslocamento horizontal, deslocamento vertical e *ZCA whitening*, que foram realizados a medida que cada lote de imagens era carregado durante cada época do treinamento.

Entre as técnicas de regularização aplicadas diretamente aos modelos pode-se citar o *dropout* que consiste de um mecanismo que objetiva eliminar um conjunto aleatório de ativações dos neurônios para evitar co-adaptações complexas associadas que são responsáveis por gerar um ajuste excessivo durante a etapa de treinamento dos modelos, fazendo com que cada neurônio possa aprender a detectar um recurso útil para produzir a resposta correta. No presente trabalho, em todas as CNNs utilizadas, seguindo cada camada totalmente conectada (exceto a camada de saída) foi aplicado *dropout* equivalente a 0.5.

3.2.3 Configurações dos Modelos

Neste estudo, as arquiteturas de redes convolucionais VGG, DenseNet, ResNet e ResNetXt foram utilizadas para classificar variadas doenças em plantas. O modelo VGG foi selecionado devido ser uma CNN que é referência na área de visão computacional, sendo uma das primeiras a obter resultados satisfatórios em um grande volume de dados. Já os modelos DenseNet e ResNet foram selecionados por possuírem resultados superiores na literatura atual. A ResNetXt foi escolhida por concentrar características dos modelos VGG e ResNet, tornando interessante a comparação de desempenho entre essas CNNs. Especificamente, as redes VGG16, DenseNet169, ResNet101 v1, ResNet101 v2 e ResNetXt foram adaptadas e então aplicadas nas dez bases de dados geradas a partir da XDB (Tabela 5).

Em termos de adaptações nas arquiteturas originais, a VGG16 foi configurada com 200 neurônios nas duas primeiras camadas totalmente conectadas, ambas seguidas por uma regularização *dropout* de 0.5, antes da camada de saída. Os modelos de DenseNet tiveram a adição de uma camada totalmente conectada composta por 200 neurônios, seguida por uma regularização *dropout* de 0.5, antes da camada de saída. A mesma configuração da DenseNet foi adotada nos modelos ResNet101 v1, ResNet101 v2 e ResNetXt. Todas as CNNs foram pré-treinadas na base de dados ImageNet (Deng et al., 2009) e suas implementações foram baseadas no *framework* Keras (KERAS. . . , 2019).

O treinamento de todos os modelos foi realizado através do uso do algoritmo *stochastic gradient descent* (SGD) (POLYAK; JUDITSKY, 1992) com momento igual a 0,9 e taxa de aprendizado variável, iniciando em 10^{-3} e decaindo ao longo das épocas até no máximo $0,5 \times 10^{-6}$. Foram consideradas 200 épocas e *batch* de tamanho 64 para todas as CNNs, com redução na taxa de aprendizado pelos fatores 10^{-1} , 10^{-2} , 10^{-3} , e $0,5 \times 10^{-1}$ quando o número de épocas era maior que 80, 120, 160, e 180, respectivamente, considerando que o resultado nos dados de teste não melhorou após 5 épocas consecutivas.

O SGD é um dos algoritmos mais populares para executar a otimização de pesos de redes neurais, demonstrando bom desempenho para problemas de larga escala. Além disso, a maioria das bibliotecas de AP contêm implementações de vários algoritmos para otimizar o gradiente descendente, como Keras, Theano, Caffe e Lasagne (RUDER, 2016).

O funcionamento do SGD envolve a execução de uma atualização por vez, o que geralmente torna o processo de treinamento muito mais rápido, facilitando a aprendizagem. Em geral, o SGD possibilita o salto para mínimos locais novos e potencialmente melhores (POLYAK; JUDITSKY, 1992). Por outro lado, isso acaba complicando a convergência ao mínimo exato, pois o SGD continuará se superando. No entanto, foi demonstrado que quando a taxa de aprendizado é diminuída lentamente, o algoritmo demonstra o mesmo comportamento de convergência que o gradiente descendente em lote, aproximadamente

convergindo para um mínimo local ou global para otimizações convexas e não-convexas, respectivamente (RUDER, 2016). Por esses motivos, a redução na taxa de aprendizagem ao longo das épocas é justificada neste estudo.

O momento é uma técnica utilizada para acelerar a descida do gradiente do algoritmo SGD para a direção de maior relevância, além de reduzir oscilações. Polyak (1964) mostrou que o momento pode acelerar consideravelmente a convergência para um mínimo local, trazendo a vantagem de obter o mesmo nível de precisão com uma quantidade menor de iterações.

3.3 Métricas utilizadas na avaliação de desempenho dos resultados

A área de aprendizagem de máquina envolve diversas maneiras de avaliar o desempenho dos algoritmos de aprendizado e os classificadores que eles produzem. As métricas de qualidade da classificação são construídas a partir de uma matriz de confusão que registra exemplos reconhecidos corretos e incorretos para cada classe (SOKOLOVA; JAPKOWICZ; SZPAKOWICZ, 2006). No presente trabalho, visando avaliar de forma concisa os resultados, todos os modelos de CNNs utilizados foram executados dez vezes e então médias foram calculadas para obter valores das seguintes métricas: acurácia na etapa de teste, *f1-score*, e AUC. Tais métricas foram escolhidas por serem amplamente utilizadas na literatura de AP (BRADLEY, 1997).

Em geral, o desempenho da classificação sem focar em uma classe específica é a maneira mais geral de comparar algoritmos, não favorecendo nenhuma aplicação específica. A introdução de um novo problema de aprendizagem inevitavelmente se concentra em seu domínio, mas omite uma análise detalhada. Assim, a medida empírica mais utilizada, a acurácia, se refere a taxa de acerto global do experimento, ou seja, proporção de predições corretas em relação ao tamanho do conjunto de dados, não distinguindo entre o número de rótulos corretos de diferentes classes. Essa métrica é dada por:

$$\text{acurácia} = \frac{v_p + v_n}{v_p + f_p + f_n + v_n}, \quad (3.1)$$

onde v_p representa casos verdadeiros positivos, v_n verdadeiros negativos, f_p falsos positivos e f_n falsos negativos.

A métrica *f1-score* contempla a medida de acurácia no teste de um determinado experimento, se limitando a um cálculo baseado nas métricas de precisão e *recall* (PENHA, 2018). A precisão se refere a taxa de acertos dos exemplos classificados como positivos que realmente são positivos, dada por $p_e = \frac{v_p}{v_p + f_p}$, enquanto o *recall* verifica a frequência com que o classificador encontra os exemplos de uma classe, ou seja, indica a relação entre o número de exemplos corretamente classificados e o número total de exemplos que

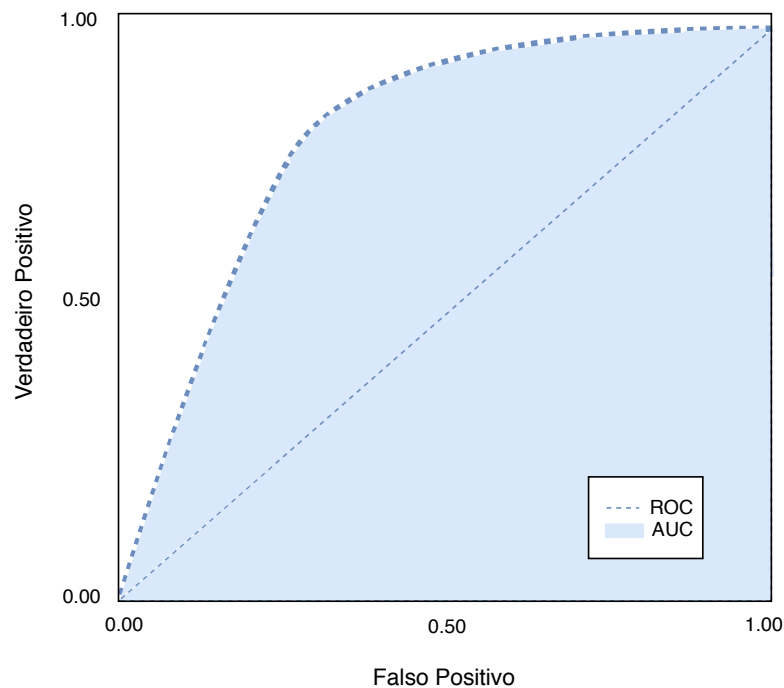
deveriam ter sido corretamente classificados, dado por $r_e = \frac{v_p}{v_p + f_n}$. Sendo assim, o *f1-score* é calculado tal que:

$$f1\text{-score} = \frac{2 \times (p_e \times r_e)}{p_e + r_e}. \quad (3.2)$$

A curva ROC (*Receiver Operating Characteristic*) é utilizada para visualizar o desempenho de um classificador ao selecionar um ponto operacional adequado ou um limiar de decisão. Ao comparar vários esquemas de classificação diferentes, geralmente é desejável obter uma única figura como uma medida de desempenho do classificador. Nesse sentido, a métrica AUC foi desenvolvida para fornecer uma forma abrangente e gráfica de resumir o desempenho de diferentes algoritmos de classificação (BRADLEY, 1997).

Teoricamente, a métrica AUC pode ser definida como uma medida de desempenho para o problema de classificação que se baseia nos conceitos de especificidade e sensibilidade, representando o grau ou medida de separabilidade, explicitando o quanto um determinado modelo é capaz de distinguir entre classes. Logo, quanto maior é o valor da métrica AUC, melhor é o desempenho do modelo. Essa medida varia no intervalo de 0 a 1. Um modelo cujas previsões são 100% incorretas tem uma AUC de 0, enquanto aquele modelo cujas previsões são 100% corretas tem uma AUC de 1,0. A Figura 15 apresenta um exemplo gráfico dessa métrica.

Figura 15: Exemplo de AUC.



Fonte: A autora (2019).

Com o objetivo de atestar a significância dos resultados obtidos na etapa de teste das CNNs neste trabalho, a avaliação de desempenho também considerou a aplicação do procedimento estatístico não-paramétrico denominado teste de Friedman (FRIEDMAN, 1937), considerando os procedimentos *post-hoc* associados Shaffer (SHAFFER, 1986) e Bergmann (BERGMANN; HOMMEL, 1988). As médias da métrica de acurácia na etapa de teste a partir de dez execuções, para cada CNN aplicada em cada base de dados, serviram como entrada para esse estudo estatístico.

Testes não paramétricos, além de sua definição original para lidar com dados nominais/ordinais, também podem ser aplicados a dados contínuos, realizando transformações baseadas em classificação, ajustando os dados de entrada aos requisitos de cada teste. Esses testes podem executar duas classes de análises: comparações pareadas e comparações múltiplas. Os procedimentos estatísticos em pares realizam comparações individuais entre dois algoritmos, obtendo em cada aplicação um valor p (*p-value*) independente de outro. Já os testes de comparações múltiplas devem ser utilizados para realizar uma comparação que envolva mais de dois algoritmos (HOLLANDER; WOLFE, 1999).

Particularmente, considerando testes de comparações múltiplas, vislumbram-se duas possibilidades de comparações, entre N_a modelos/algoritmos. Nas comparações $1 \times N_a$, um método de controle é destacado (normalmente, um algoritmo proposto) através da aplicação do teste. Então, todas as hipóteses de igualdade entre o método de controle e o restante podem ser testadas pela aplicação de um conjunto de procedimentos *post-hoc*. Também são possíveis comparações $N_a \times N_a$, considerando as hipóteses de igualdade entre todos os pares de algoritmos existentes, com a inclusão de procedimentos *post-hoc* específicos para essa tarefa (DERRAC et al., 2011).

O teste de Friedman é um teste de comparações múltiplas utilizado para detectar diferenças de significância entre dois ou mais algoritmos (FRIEDMAN, 1937). Na primeira etapa no cálculo da estatística no teste de Friedman são consideradas as $k_a(k_a - 1)/2$ possíveis comparações entre os k_a algoritmos. Os resultados dessas comparações podem ser ordenados pelo seu valor p de forma crescente em uma espécie de *ranking* calculado por cada teste utilizando uma aproximação normal. Esse procedimento considera os seguintes passos:

1. Reunir os resultados observados para cada par algoritmo/problema.
2. Classificar os valores de 1 (melhor resultado) a k_a (pior resultado) para cada problema i . Esse *ranking* é denotado por:

$$r_{ij}(1 \leq j \leq k_a). \quad (3.3)$$

3. Para cada algoritmo j , calcular a média das classificações obtidas em todos os n_p

problemas para obter a classificação final, definida por:

$$R_j = \frac{1}{n_p} \sum_i r_{ij} \quad (3.4)$$

No contexto do desempenho das CNNs, se a significância estatística apontada pelo teste de Friedman for válida para pelo menos duas CNNs consideradas, um procedimento *post-hoc* precisará ser executado para encontrar as CNNs específicas onde a significância existe (SANTOS et al., 2019). Para as comparações $N_a \times N_a$, os procedimentos *post-hoc* com resultados mais confiáveis são Shaffer e Bergmann (DERRAC et al., 2011) e, por esse motivo, foram os métodos selecionados para esse estudo.

4 Resultados

Nos capítulos anteriores foram apresentadas a fundamentação teórica e a proposta metodológica que pautaram este estudo no que tange a classificação de doenças em plantas a partir de imagens digitais. O presente capítulo apresentará os resultados obtidos a partir da aplicação da metodologia proposta neste trabalho.

Ressalta-se que todos os experimentos foram executados em uma GPU NVIDIA GeForce GTX 1050 TI com 4 GB de memória dedicada, juntamente com um processador Intel Core i7-8750H de 2,2 GHz (4,1 GHz) e 16 GB de memória principal. A média no tempo computacional das execuções mais custosas considerando a base de dados 32hard, que possui a maior quantidade de imagens, envolveu as CNNs ResNet101v1 e a ResNet101v2, com 10h e 10min e 11h e 30min, respectivamente, seguida pela DenseNet com aproximadamente 9h e 30min e a ResNetXt com 6h. A CNN com menor tempo médio de execução foi a VGG16 que obteve 3h e 30min para essa mesma base. Note que apesar das CNNs terem sido executadas em GPU, realizar mais que dez execuções para cada modelo mostrou-se custoso em termos temporais, mas as médias obtidas considerando todas as execuções apresentaram desvios-padrões relativamente baixos.

A estratégia utilizada na aplicação dos modelos de CNN citados no problema de classificação de doenças de plantas envolveu iniciar o treinamento desses algoritmos a partir de um ponto intermediário, ao invés de realizar o treinamento do princípio, isso porque a base de dados ImageNet possui imagens de plantas (Deng et al., 2009), mesmo que saudáveis, logo o pré-treinamento nesse conjunto de dados contribuiu para uma maior eficácia no treinamento refinado realizado após essa etapa. Esse fato também justifica a utilização do algoritmo de treinamento SGD e do decaimento na taxa de aprendizado ao longo das épocas, uma vez que o objetivo é de otimização lenta de modelos pré-treinados, um refinamento, sem abruptas explorações do espaço de soluções/características.

A seção seguinte apresentará os resultados das execuções nas CNNs supracitadas, baseado nas métricas de desempenho acurácia na etapa de teste, *f1-score* e AUC. Posteriormente, a seção Teste Estatístico Não-paramétrico abrangerá os resultados da aplicação dessa técnica utilizando a medida de desempenho acurácia na etapa de teste, além de uma teste estatístico de Estimativa por Contraste afim de avaliar a diferença entre o desempenho das CNNs utilizadas.

4.1 Avaliação de desempenho através de acurácia, *f1-score* e AUC

A partir de dez execuções de cada CNN em cada base dados, as médias de desempenho para as métricas de acurácia, *f1-score* e AUC nos dados de teste são apresentadas através das Tabelas 7, 8 e 9, respectivamente. Em uma análise global, é possível verificar que na maioria dos casos, a ResNetXt50 manteve melhor desempenho nas bases de dimensão maior que 40×40 , entre todas as CNNs e métricas consideradas, enquanto a DenseNet169 obteve resultados melhores nas bases de dimensão menor que 48×48 e resultados minimamente inferiores aos da ResNetXt50 para as bases de dimensão maior que 40×40 . Em contrapartida, a ResNet101v1 obteve os piores resultados nas bases de dados de dimensão 32×32 , embora tenha superado a ResNet101v2 em todas as bases de dados de dimensão superiores a 32×32 e tido resultados superiores aos da VGG16 nas bases de dados de dimensão superiores a 48×48 , exceto por um empate de desempenho com a mesma na base de dados 56final. As métricas relacionadas com acurácia no teste e AUC ressaltaram melhor os desempenhos das CNNs, apresentando diferenças relevantes entre os desempenhos alcançados.

Analisando os resultados baseados na métrica de acurácia no teste (Tabela 7) nas bases de dados com dimensão 40×40 e 64×64 , é possível observar que os desempenhos das CNNs ResNetXt50 e DenseNet169 se mantiveram similares, com pouca vantagem para a primeira. Uma diferença um pouco maior entre os resultados dessas duas CNNs pode ser observada nas bases de 48×48 e 56×56 , com destaque positivo para a ResNetXt50, enquanto nas bases de 32×32 a DenseNet169 obteve os melhores resultados entre todas as CNNs estudadas. Em contrapartida, as ResNets com 101 camadas apresentaram desempenho inferior, sendo a ResNet101v2 o destaque negativo para todas as bases de dados utilizadas nesse estudo, enquanto a ResNet101v1 ainda conseguiu obter resultados minimamente superiores aos da VGG16 nas bases de dados com dimensão 56×56 e 64×64 .

É importante ressaltar que as bases de dados com as menores dimensões e com a maior quantidade de classes, como a base de dados de 32×32 e 40×40 , são aquelas que possuem mais imagens, o que consequentemente exigiu mais das CNNs, isto é, onde elas apresentaram seus menores desempenhos em termos de performance de classificação.

Tendência semelhante nos resultados foram observadas ao se considerar a métrica *f1-score* (Tabela 8) nas bases de dados com dimensão maiores que 40×40 , nas quais a ResNetXt50 obteve um desempenho levemente superior que a DenseNet169, embora a última tenha alcançado os melhores resultados nas bases de dimensão 32×32 entre todas as CNNs testadas. Por outro lado, embora a VGG16 tenha obtido resultados relativamente superiores aos das ResNets nas bases de dados de dimensão 32×32 e na base de dados 40final, em todas as outras bases de dados subsequentes o desempenho dessa CNN foi menor quando comparado a ResNet101v1, embora tenha sido consideravelmente melhor quando comparada a ResNet101v2.

Tabela 7: Média da métrica de acurácia no teste para as bases de dados.

	VGG16	ResNet101v1	ResNet101v2	ResNetXt50	DenseNet169
32final	0,936	0,876	0,916	0,940	0,942
32hard	0,929	0,878	0,911	0,936	0,940
40final	0,947	0,943	0,933	0,955	0,954
40hard	0,941	0,939	0,930	0,950	0,949
48final	0,955	0,953	0,943	0,965	0,962
48hard	0,950	0,946	0,938	0,961	0,957
56final	0,959	0,959	0,951	0,966	0,962
56hard	0,954	0,958	0,949	0,965	0,961
64final	0,963	0,964	0,953	0,970	0,971
64hard	0,960	0,966	0,949	0,968	0,967

Tabela 8: Média da métrica de *f1-score* para as bases de dados.

	VGG16	ResNet101v1	ResNet101v2	ResNetXt50	DenseNet169
32final	0,917	0,852	0,896	0,925	0,926
32hard	0,881	0,828	0,864	0,904	0,910
40final	0,930	0,928	0,913	0,940	0,940
40hard	0,896	0,901	0,885	0,915	0,916
48final	0,934	0,937	0,920	0,950	0,946
48hard	0,907	0,910	0,889	0,931	0,922
56final	0,940	0,940	0,931	0,950	0,948
56hard	0,911	0,926	0,907	0,938	0,930
64final	0,944	0,948	0,930	0,958	0,956
64hard	0,929	0,936	0,913	0,946	0,946

Os resultados apontados pela métrica AUC (Tabela 9) reforçam que, de um panorama geral, os melhores resultados são da ResNetXt50 seguida da DenseNet169 por uma diferença mínima, onde se pode ressaltar um empate nos resultados para a base 40hard e uma vantagem da DenseNet169 nas bases de dados de dimensão 32×32 , reforçando o resultado das métricas anteriores. No que tange a VGG16, foi demonstrado que seus resultados foram superiores as ResNets de 101 camadas no que tange as bases de dados de dimensão inferior a 56×56 , exceto pelo resultado da base de dados 40hard onde o desempenho da ResNet101v1 foi minimamente superior ao da VGG16. Na base de dados 56final, houve um empate entre o desempenho da ResNet101v1 e a VGG16 e, na base de dados 56hard e nas duas bases de dados de dimensão 64×64 , a ResNet101v1 superou o desempenho da VGG16 e da ResNet101v2. Ainda, vale ressaltar que, para as bases de dimensão 32×32 , a métrica AUC apontou que a ResNet101v2 obteve resultados melhores que a ResNet101v1, embora ainda não tenham sido suficientes para ultrapassar

os resultados das melhores CNNs, ResNetXt e DenseNet169.

Tabela 9: Média da métrica AUC para as bases de dados.

	VGG16	ResNet101v1	ResNet101v2	ResNetXt50	DenseNet169
32final	0,957	0,922	0,945	0,960	0,962
32hard	0,937	0,907	0,925	0,947	0,951
40final	0,963	0,962	0,954	0,969	0,968
40hard	0,945	0,946	0,937	0,953	0,953
48final	0,966	0,965	0,958	0,972	0,970
48hard	0,950	0,948	0,938	0,959	0,955
56final	0,969	0,969	0,964	0,974	0,971
56hard	0,953	0,958	0,949	0,965	0,960
64final	0,969	0,970	0,961	0,975	0,975
64hard	0,963	0,965	0,951	0,969	0,970

Através dos resultados apresentados pelas métricas de acurácia no teste, *f1-score* AUC é verificado que os modelos ResNetXt50 e DenseNet169 obtiveram um melhor desempenho no que tange todas as bases de dados utilizadas nesse estudo, com uma leve vantagem para a ResNetXt50.

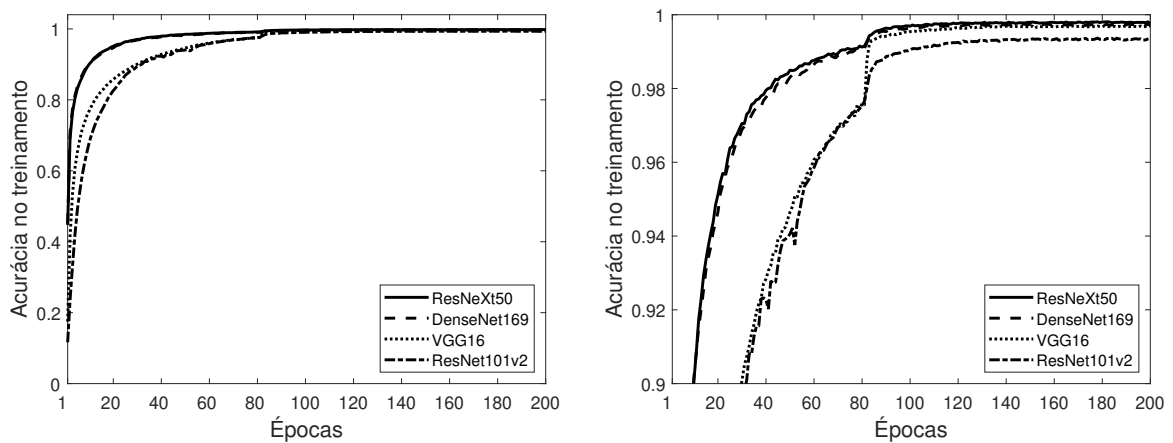
É possível inferir que um dos motivos que fizeram com que a CNN ResNetXt50 obtivesse um resultado superior aos outros modelos se deve a simplicidade de sua arquitetura, já que ela é altamente modularizada e trabalha com reutilização de blocos residuais, reduzindo o custo computacional nas execuções, além do acréscimo da dimensão de cardinalidade que, comprovadamente possui resultados satisfatórios na literatura (XIE et al., 2017), reunindo os pontos fortes presentes nas CNNs VGG e ResNet. Quanto aos resultados da DenseNet169, pode-se deduzir que foram satisfatórios devido ao fluxo avançado de informações que percorre a rede durante a etapa de treinamento, fazendo com que todas as camadas do modelo conheçam as informações inerentes as camadas anteriores. Outro ponto a ressaltar, é o fato de que tanto a ResNetXt50 como a DenseNet169 possuem mecanismos de regularização que reduzem significativamente problemas como o gradiente de desaparecimento, que ainda se configura como um desafio na arquitetura VGG16. Quanto aos resultados obtidos pelas ResNets, supõe-se que o pré-treinamento na base de dados Imagenet não favoreceu ou não foi suficiente para uma melhor performance de classificação dado o fato do pré-treinamento incluir apenas imagens de plantas saudáveis, ou ainda pela arquitetura das ResNets terem grande profundidade necessitando de mais épocas para atingir resultados mais aceitáveis.

Para efeitos de visualização dos desempenhos das CNNs nas etapas de treinamento e teste, a seguir são apresentados resultados para as bases de dados 32hard e 64hard, pois elas denotam extremos em termos de quantidade de imagens. Note que a ResNet101v1

foi excluída dessa análise apenas para facilitar a visualização dos resultados.

Na Figura 16 são apresentados os desempenhos das CNNs em termos de acurácia ao longo das épocas da etapa de treinamento na base de dados 32hard. Observa-se que ResNetXt50 e DenseNet169 já iniciam o treinamento com resultados superiores as outras CNNs, embora a primeira possua uma leve vantagem até por volta da época 80. As CNNs VGG16 e ResNet101v2, inicialmente, mantiveram desempenho com algumas diferenças até por volta da época 60, onde até aproximadamente a época 80 os desempenhos foram significativamente similares. Nas épocas seguintes, a VGG16 conseguiu ultrapassar o desempenho da ResNet101v2 até o final do treino.

Figura 16: Desempenhos das CNNs ResNetXt50, DenseNet169, VGG16 e ResNet101v2 em termos de acurácia durante a etapa de treinamento na base de dados 32hard.

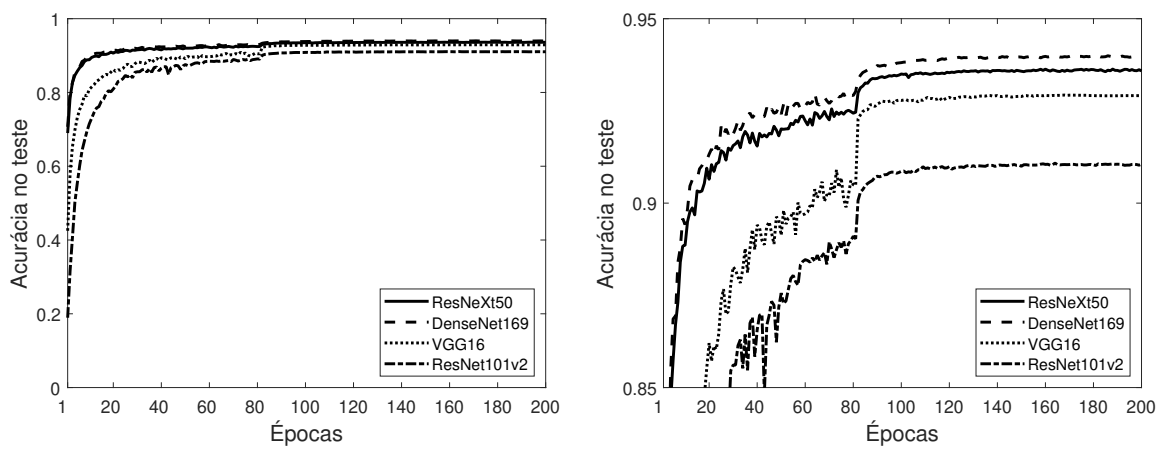


Fonte: A autora (2019).

No que diz respeito aos desempenhos das CNNs na etapa de teste na base de dados 32hard, verifica-se através da Figura 17 que os melhores resultados no treinamento das CNNs ResNetXt50 e DenseNet169 contribuíram positivamente para uma melhor generalização na classificação de novas imagens das variadas classes de doenças de plantas, enquanto um desempenho relativamente insatisfatório ficou perceptível em relação a ResNet101v2, mas que não caracteriza um efeito de *overfitting*. A VGG16 obteve um desempenho intermediário, em certo grau aproximando-se dos melhores resultados a partir da época 80, onde se pode perceber o efeito positivo da aplicação da redução da taxa de aprendizado para esse modelo, e ao mesmo tempo distanciando-se ainda mais do resultado obtido pela ResNet101v2.

A Figura 18 apresenta os desempenhos das CNNs em termos de acurácia ao longo das épocas da etapa de treinamento na base de dados 64hard. Infere-se novamente que apenas as CNNs ResNetXt e DenseNet169 iniciaram o treinamento com resultados superiores, enquanto após a época 100 o desempenho entre todas as CNNs se manteve similar, demonstrando que a estratégia de geração de novas imagens durante o treino teve efeito

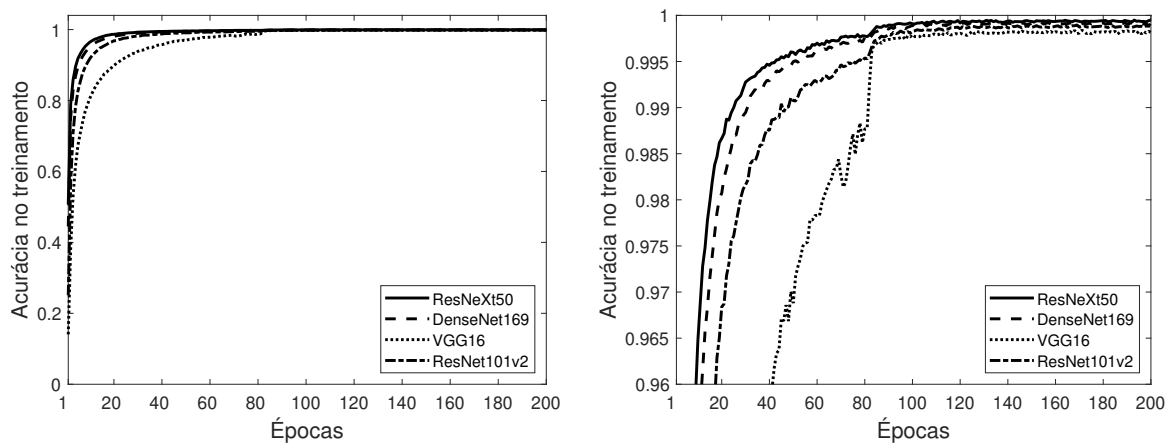
Figura 17: Desempenhos das CNNs ResNetXt50, DenseNet169, VGG16 e ResNet101v2 em termos de acurácia durante a etapa de teste na base de dados 32hard.



Fonte: A autora (2019).

positivo ao longo das épocas para todas as CNNs aplicadas nesta base de dados. Especificamente, apesar de diferenças imperceptíveis a partir da época 100, a ResNetXt50 permaneceu com desempenho geral superior, enquanto a ResNet101v2 se manteve em último lugar.

Figura 18: Desempenhos das CNNs ResNetXt50, DenseNet169, VGG16 e ResNet101v2 em termos de acurácia durante a etapa de treinamento na base de dados 64hard.

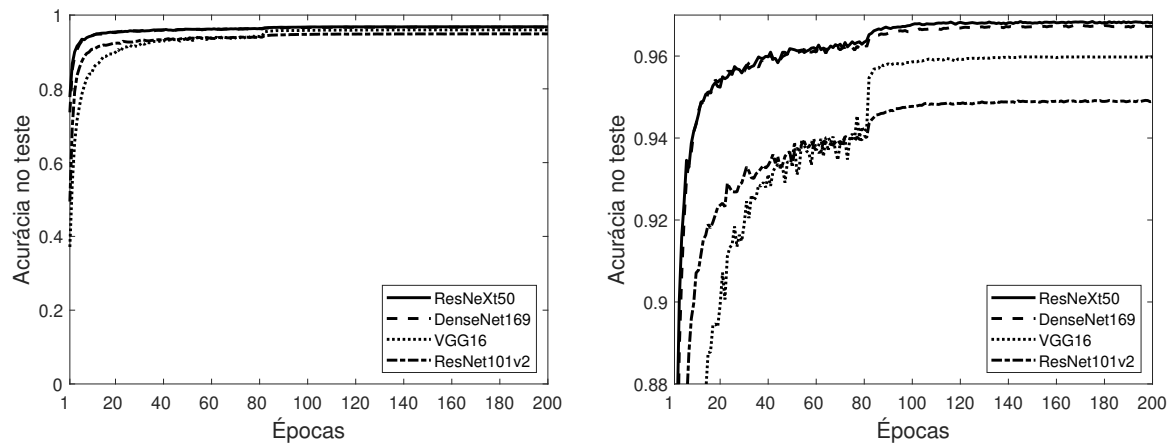


Fonte: A autora (2019).

Na etapa de teste das CNNs na base de dados 64hard certas discrepâncias ficaram mais evidentes nos resultados, quando comparados aos resultados do treinamento, como pode ser observado na Figura 19. Uma diferença quase imperceptível entre os desempenho das CNNs ResNetXt50 e DenseNet169 é notada, ao contrário do desempenho da VGG16 onde é clara a diferença entre ela em relação as demais CNNs. Apesar dos resultados do treinamento demonstrarem similaridades, a capacidade de generalização das CNNs para

novas imagens de teste demonstrou que o modelo treinado da ResNet101v2 teve certa especialização nos dados de treinamento, quando comparado com as outras CNNs.

Figura 19: Desempenhos das CNNs ResNetXt50, DenseNet169, VGG16 e ResNet101v2 em termos de acurácia durante a etapa de teste na base de dados 64hard.



Fonte: A autora (2019).

4.2 Teste estatístico não-paramétrico

Com a finalidade de verificar significância em todas as possíveis comparações de performance par-a-par entre as CNNs, o teste estatístico não-paramétrico de Friedman foi realizado, sendo complementado pelos procedimentos *post-hoc* Shaffer e Bergmann de forma a ajustar o valor p computado inicialmente. Esses experimentos assumiram as seguintes hipóteses nula e alternativa, respectivamente: H_0 que considera que os resultados dos algoritmos comparados não apresentam diferenças; e H_1 que considera que os resultados dos algoritmos comparados apresentam diferenças.

Para aplicar o teste de Friedman, os resultados originais em termos de acurácia no teste (Tabela 7) foram convertidos por meio de uma transformação baseada em *ranking* (DERRAC et al., 2011). A Tabela 10 sintetiza o *ranking* computado a partir do teste de Friedman, ressaltando as CNNs ResNeXt50 e DenseNet169 como as melhores, com uma leve vantagem para a primeira, enquanto a ResNet101v2 foi considerada a pior CNN levando em consideração todas as bases de dados deste estudo. Logo, é possível inferir que existe significância nas múltiplas comparações possíveis entre as CNNs, e que em alguns casos a hipótese H_0 pode ser refutada e a hipótese H_1 pode ser aceita.

Na Tabela 11 são apresentadas todas as hipóteses de comparações e os valores p não-ajustados e ajustados pelos procedimentos *post-hoc* de Shaffer e Bergmann. É importante notar que quanto menor o valor de p , maior a significância entre os resultados das CNNs comparadas, culminando na rejeição da hipótese nula H_0 e, conseqüente, aceitação da hipótese alternativa H_1 com um grau de confiança elevado.

Tabela 10: Ranking médio computado pelo teste de Friedman.

CNN	Ranking
ResNetXt	1,3
DenseNet169	1,7
VGG16	3,4
ResNet101v1	3,8
ResNet101v2	4,8

Dessa forma, conforme atesta a Tabela 11, a CNN ResNetXt50 possui uma performance significativamente superior quando comparada a ResNet101v1, ResNet101v2 e a VGG16, possuindo os melhores resultados. Por outro lado, o comparativo entre ResNetXt50 e DenseNet169 não demonstrou significância adequada para garantir qualquer inferência de performance melhor ou pior com grau de confiança satisfatório. Ao comparar o desempenho da VGG16 com a ResNet101v1, percebe-se que não há significância, enquanto a comparação entre VGG16 e ResNet101v2 sugere alguma significância, apontando certa superioridade nos resultados apresentados pela VGG16. Por fim, as hipóteses que fizeram a comparação entre o desempenho das ResNets de 101 camadas não apresentou significância.

Tabela 11: Valores p não-ajustado (Friedman) e ajustado (Shaffer e Bergmann) para múltiplas comparações.

Hipótese	p não-ajustado	p Shaffer	p Bergmann
ResNet101v2 vs. ResNetXt50	7,430983723703081E-7	7,430983723703081E-6	7,430983723703081E-6
ResNet101v2 vs. DenseNet169	1,1648657367238044E-5	6,989194420342827E-5	6,989194420342827E-5
ResNet101v1 vs. ResNetXt50	4,0695201744492504E-4	0,0024417121046695503	0,0024417121046695503
VGG16 vs. ResNetXt50	0,0029794666563330327	0,017876799937998195	0,01191786662533213
ResNet101v1 vs. DenseNet169	0,0029794666563330444	0,017876799937998268	0,01191786662533213
VGG16 vs. DenseNet169	0,01620954140922542	0,06483816563690167	0,03241908281845084
VGG16 vs. ResNet101v2	0,04771488023735128	0,19085952094940511	0,19085952094940511
ResNet101v1 vs. ResNet101v2	0,1572992070502848	0,47189762115085443	0,3145984141005696
ResNetXt50 vs. DenseNet169	0,5716076449533316	1,1432152899066632	1,1432152899066632
VGG16 vs. ResNet101v1	0,5716076449533326	1,1432152899066632	1,1432152899066632

Adicionalmente, foi realizado um teste estatístico não-paramétrico de comparações pareadas ($1 \times N$) denominado estimativa por contraste baseada em medianas (DOKSUM, 1967), com o objetivo de estimar o contraste entre as medianas das amostras de resultados, considerando todas as comparações par-a-par entre as CNNs, obtendo como resultado uma diferença quantitativa que é especialmente útil para estimar até que ponto uma CNN supera a outra. O desempenho das CNNs é refletido pelas magnitudes das diferenças entre elas em cada domínio.

A Tabela 12 apresenta o resultado da aplicação do teste de estimativa por contraste nos resultados referentes a métrica de acurácia na etapa de teste (Tabela 7). Note que as comparações entre os mesmos modelos de CNNs não apresentam valor justamente

por não serem comparações válidas. Nas comparações entre CNNs diferentes, esse teste demonstrou que as ResNets de 101 camadas não superam nenhuma das outras CNNs apresentadas e, entre si, a ResNet101v1 supera o desempenho da ResNet101v2 por uma margem pequena.

Por sua vez, as comparações envolvendo o desempenho da VGG16 demonstraram que ela superou somente as ResNets, com uma diferença de desempenho maior e menor para ResNet101v2 e ResNet101v1, respectivamente. Entre os melhores resultados, vale ressaltar a ResNetXt50 que ultrapassou o desempenho de todas as outras CNNs, embora, entre essas comparações, a diferença no desempenho dessa CNN com a DenseNet169 tenha sido mínima.

Tabela 12: Teste Estatístico não-paramétrico de Estimativa por Contraste.

	VGG16	ResNet101v1	ResNet101v2	ResNetXt	DenseNet169
VGG16	0,000	0,002410	0,01146	-0,008925	-0,007413
ResNet101v1	-0,002410	0,000	0,009048	-0,01134	-0,009823
ResNet101v2	-0,01146	-0,009048	0,000	-0,02038	-0,01887
ResNetXt50	0,008925	0,01134	0,02038	0,000	0,001512
DenseNet169	0,007413	0,009823	0,01887	-0,001512	0,000

Em suma, os testes estatísticos validaram muitas das conclusões obtidas nos experimentos, como o fato das CNNs ResNetXt50 e DenseNet169 terem obtido os melhores resultados quando comparadas as demais, e o baixo desempenho de generalização das ResNets. Além disso, esses testes também auxiliaram na refutação de significância principalmente nos seguintes comparativos: ResNet101v1 vs. ResNet101v2; ResNetXt50 vs. DenseNet169; e VGG16 vs. ResNet101v1.

5 Considerações finais e trabalhos futuros

O presente trabalho apresentou como proposta uma metodologia para a classificação de doenças em plantas baseada na aplicação de técnicas de AP em uma base de dados de imagens digitais de múltiplas patologias referentes a diversas culturas. Visando alcançar os objetivos propostos, as imagens da base de dados XDB foram submetidas às técnicas de pré-processamento baseadas na taxa de diferença e no redimensionamento, objetivando padronizar as imagens que serviram de entrada para o treinamento das CNNs. Ressalta-se ainda que foram criadas dez bases de dados de diferentes dimensões, a partir da base de dados XDB, a fim de avaliar o desempenho das CNNs quando submetidos a diferentes situações. As bases criadas envolveram quantidades de classes diferentes, levando em consideração as classes de maior representatividade.

As duas bases de dados de dimensão 32×32 dispunham de 50 e 66 classes de doenças, enquanto as duas bases de dimensão 40×40 referiam-se a 50 e 65 classes de patologias. As duas bases de dimensão 48×48 possuíam 50 e 64 classes de desordens, enquanto as duas bases de dimensão 56×56 referem-se a 50 e 62 classes. Por fim, as bases de dimensão 64×64 com 50 e 60 classes. Todas as bases de dados estavam associadas a 16 tipos de espécies vegetais distintas.

Posteriormente, foram desenvolvidas as implementações das CNNs VGG16, ResNet101v1, ResNet101v2, ResNetXt50 e DenseNet169 utilizando o *framework* Keras, a fim de efetivamente realizar o treinamento e possibilitar a classificação das doenças em plantas nas dez bases de dados de imagens digitais criadas.

Após o treinamento dos modelos, deu-se início a etapa de avaliação dos resultados obtidos a partir das métricas de avaliação: acurácia na etapa de teste, *f1-score* e AUC. Em seguida, com o objetivo de avaliar o desempenho das CNNs utilizadas na tarefa de classificação, foi realizado o estudo estatístico não-paramétrico de Friedman considerando os procedimentos *post-hoc* associados Shaffer e Bergmann, com intuito de atestar a significância dos resultados obtidos através da métrica de acurácia no teste, pois dentre as métricas citadas anteriormente foi a que garantiu resultados onde se pôde observar melhor as diferenças entre as redes, gerando um *ranking* de desempenho das CNNs.

A partir dos resultados obtidos, a CNN ResNetXt50 alcançou resultados de classificação promissores quando comparados àqueles obtidos pelas CNNs VGG16, ResNet101v1 e ResNet101v2, tanto em termos de acurácia no teste e outras métricas relacionadas, quanto através do teste estatístico não-paramétrico de Friedman, e pelo teste adicional de

estimativa por contraste. Por outro lado, levando em consideração o teste de Friedman, a comparação entre o desempenho da ResNetXt50 e da DenseNet169 não obteve significância que possibilitasse inferir qual CNN possuiu o melhor desempenho, ao contrário do teste de estimativa por contraste que apontou pequena diferença entre essas duas CNNs, revelando o melhor desempenho da ResNetXt50.

A utilização de CNNs se mostrou eficiente no que diz respeito a tarefa de classificação de imagens de plantas doentes, assim como a configuração selecionada para o treinamento, exceto pelas ResNets de 101 camadas cujo efeito do treinamento na base de dados ImageNet não obteve resultados tão eficientes quanto os outros modelos testados sob as mesmas condições. Além disso, acredita-se que o aumento na quantidade de épocas poderia melhorar a precisão da classificação para essas CNNs, embora ainda seja necessário avaliar o custo benefício desse aumento com relação ao custo computacional, já que as ResNets foram os modelos que necessitaram de maior tempo de execução, especificamente cerca de quatro vezes mais tempo que a VGG16, CNN com tempo menor de execução.

Em geral, a metodologia proposta neste estudo se difere de outros trabalhos da literatura que, em sua maioria, descrevem apenas a aplicação de CNNs na classificação de doenças em plantas, não abordando os detalhes relacionados ao pré-processamento das imagens. Além disso, este trabalho acrescenta a aplicação de estudos estatísticos não-paramétricos, que ajudam a assegurar a significância dos desempenhos das CNNs.

Ademais, os experimentos desta pesquisa apontaram que as CNNs selecionadas conseguem aprender os padrões relacionados as patologias presentes nas imagens digitais de plantas doentes, sendo úteis para classificar problemas com uma variada quantidade de dados e múltiplas classes de forma satisfatória, embora seja necessário avaliar se a alteração da configuração geral das CNNs aumentaria o desempenho da rede.

A partir das análises realizadas, pode-se concluir que a proposta metodológica associada a essa pesquisa tem validade para realizar a classificação de bases de dados de imagens digitais de doenças em plantas, trazendo consigo as seguintes contribuições:

- Apresentação de uma proposta metodológica para a classificação de doenças em plantas baseada no processamento de imagens digitais de culturas doentes, que pode ser útil aos especialistas da área de doenças em plantas, possibilitando uma rapidez no diagnóstico precoce que contribui para a redução de perdas de lucro no setor agrícola;
- Fornecimento de uma ferramenta que auxilia no controle do alastramento das epidemias, contribuindo para a segurança alimentar na produção de alimentos;
- Demonstração da viabilidade na utilização da base de dados de doenças em plantas XDB que, até o momento, ainda não foi utilizada publicamente por outros pesqui-

sadores no que tange a aplicação de técnicas de AP; e

- Divulgação de trabalho apresentado à outros pesquisadores por meio de conferência internacional e nacional:
 - REZENDE, V. et al. Image processing with convolutional neural networks for classification of plant diseases. In: . [S.l.]: 2019 BRACIS 8th Brazilian Conference on Intelligent Systems, 2019.
 - REZENDE, V. et al. Aplicação de redes neurais convolucionais na classificação de doenças de plantas. In: . [S.l.]: 2019 CBIC XIV Congresso Brasileiro de Inteligência Computacional, 2019.

Com o objetivo de dar continuidade a pesquisa desenvolvida, a seguir são elencados possíveis trabalhos futuros:

- Criação de bases de dados de doenças em plantas baseada na base de dados XDB com dimensões maiores que 64×64 ;
- Alteração da configuração das CNNs, aumentando a quantidade de épocas, a fim de avaliar o impacto no desempenho das CNNs, especialmente as ResNets;
- Realizar a execução dos modelos de ResNet com a configuração sem pré-treinamento, além da execução de variantes pré-treinadas e não pré-treinadas dessa CNN com maior profundidade;
- Executar a ResNetXt utilizando profundidades maiores que 50;
- Estudar os processos inerentes a utilização de outros modelos de CNN;
- Analisar de forma mais robusta os resultados de classificação obtidos, levando em consideração as classes com maior taxa de erro e buscando investigar a razão para tal comportamento;
- Combinar dois ou mais critérios para a tomada de decisão acerca do melhor modelo de CNN; e
- Adição de outros testes estatísticos não-paramétricos na etapa de análise de desempenho das CNNs, com o objetivo de reforçar a confiabilidade dos resultados, além de acrescentar outras métricas de desempenho como insumo para esses testes.

Referências

- ARAÚJO, F. H. D. et al. Redes neurais convolucionais com tensorflow: Teoria e prática. In: *2017 SBC III Escola Regional de Informática do Piauí*. [S.l.]: SBC, 2017. v. 1, n. 1, p. 382–406. ISSN 978-85-7669-395-6. Citado 2 vezes nas páginas 24 e 26.
- BARBEDO, J. G. A. et al. Annotated plant pathology databases for image-based detection and recognition of diseases. *IEEE Latin America Transactions*, 2018. Institute of Electrical and Electronics Engineers (IEEE), v. 16, n. 6, p. 1749–1757, jun. 2018. Citado 3 vezes nas páginas 2, 6 e 36.
- BERGMANN, B.; HOMMEL, G. Improvements of general multiple test procedures for redundant systems of hypotheses. In: *Multiple Hypothesenprüfung / Multiple Hypotheses Testing*. [S.l.]: Springer Berlin Heidelberg, 1988. p. 100–115. Citado na página 44.
- BEZERRA, E. Introdução à aprendizagem profunda. In: OGASAWARA, V. (Ed.). *Tópicos em Gerenciamento de Dados e Informações*. 1rd. ed. [S.l.]: SBC, 2016. v. 3, p. 57–86. Citado 2 vezes nas páginas 3 e 10.
- BOLFE Édson. *Embrapa: 45 anos dedicados à ciência e à inovação agrícola*. 2018. Disponível em: <<https://www.embrapa.br/busca-de-noticias/-/noticia/33608570/artigo—embrapa-45-anos-dedicados-a-ciencia-e-a-inovacao-agricola>>. Acesso em: 20.08.2019. Citado na página 35.
- BOULENT, J. et al. Convolutional neural networks for the automatic identification of plant diseases. *Frontiers in Plant Science*, 2019. Frontiers Media SA, v. 10, jul. 2019. Citado na página 40.
- BRADLEY, A. P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 1997. Elsevier BV, v. 30, n. 7, p. 1145–1159, jul. 1997. Citado 2 vezes nas páginas 42 e 43.
- BRAGA, A. de P.; LUDERMIR, T. B.; CARVALHO, A. C. P. de L. F. *Redes Neurais Artificiais: Teoria e Aplicações*. 2. ed. [S.l.]: LTC, 2007. Citado 5 vezes nas páginas 12, 13, 15, 16 e 17.
- CAMPOS, L. M. L. de; OLIVEIRA, R. C. L. de; ROISENBERG, M. Optimization of neural networks through grammatical evolution and a genetic algorithm. *Expert Systems with Applications*, 2016. Elsevier BV, v. 56, p. 368–384, set. 2016. Citado na página 12.
- CASCARDI, A.; MICELLI, F.; AIELLO, M. A. An artificial neural networks model for the prediction of the compressive strength of FRP-confined concrete circular columns. *Engineering Structures*, 2017. Elsevier BV, v. 140, p. 199–208, jun. 2017. Citado 2 vezes nas páginas 12 e 15.
- ÇELEBI, K. et al. Experimental and artificial neural network approach of noise and vibration characteristic of an unmodified diesel engine fuelled with conventional diesel, and biodiesel blends with natural gas addition. *Fuel*, 2017. Elsevier BV, v. 197, p. 159–173, jun. 2017. Citado na página 12.

- Deng, J. et al. Imagenet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2009. p. 248–255. ISSN 1063-6919. Citado 2 vezes nas páginas 41 e 46.
- DERRAC, J. et al. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 2011. Elsevier BV, v. 1, n. 1, p. 3–18, mar. 2011. Citado 3 vezes nas páginas 44, 45 e 52.
- DOKSUM, K. Robust procedures for some linear models with one observation per cell. *The Annals of Mathematical Statistics*, 1967. Institute of Mathematical Statistics, v. 38, n. 3, p. 878–883, jun. 1967. Citado na página 53.
- DUARTE, J.; ESCOBAR, J.; SILVA, H. D. da. *Agricultura movida a Ciência*. Embrapa, 2019. Disponível em: <<https://www.embrapa.br/documents/10180/1600893-/Agricultura+movida+a+ciencia/b419fc42-93db-ab17-2e7b-48c0e5b0baf2>>. Acesso em: 21.05.2019. Citado na página 35.
- F.A.O. Declaration of the world summit on food security. In: . [S.l.]: FAO, 2009. Citado na página 5.
- FERNANDES, B. J. T. *Redes Neurais com Extração Implícita de Características para Reconhecimento de Padrões Visuais*. Tese (Doutorado) — Universidade Federal de Pernambuco, Recife, 2013. Citado na página 10.
- FERREIRA, A. dos S. *Redes Neurais Convolucionais Profundas na Detecção de Plantas Daninhas em Lavoura de Soja*. Dissertação (Mestrado) — Universidade Federal de Mato Grosso do Sul, Campo Grande, 2017. Citado 4 vezes nas páginas 3, 4, 20 e 21.
- FONSECA, L. M. V. A. *Classificação de Objetos em Contexto Real por Redes Neurais Convolutivas*. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio de Janeiro – PUC Rio, Rio de Janeiro, 2017. Citado na página 26.
- FREITAS, S. O. de. *Utilização de um Modelo Baseado em Redes Neurais para a Precificação de Opções*. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, Belo Horizonte, 2001. Disponível em: <https://repositorio.ufmg.br/bitstream-/1843/BUBD-9BHGGH/1/disserta_o_sander_oliveira_de_freitas.pdf>. Citado na página 15.
- FRIEDMAN, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 1937. Informa UK Limited, v. 32, n. 200, p. 675–701, dez. 1937. Citado na página 44.
- FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 1980. Springer Nature, v. 36, n. 4, p. 193–202, abr. 1980. Citado na página 18.
- GANDHI, R. et al. Plant disease detection using CNNs and GANs as an augmentative approach. In: *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*. [S.l.]: IEEE, 2018. Citado 2 vezes nas páginas 1 e 4.
- GANJEFAR, S.; TOFIGHI, M.; KARAMI, H. Fuzzy wavelet plus a quantum neural network as a design base for power system stability enhancement. *Neural Networks*, 2015. Elsevier BV, v. 71, p. 172–181, nov. 2015. Citado 2 vezes nas páginas 12 e 18.

- GIOVANIS, D. G. et al. Bayesian updating with subset simulation using artificial neural networks. *Computer Methods in Applied Mechanics and Engineering*, 2017. Elsevier BV, v. 319, p. 124–145, jun. 2017. Citado na página 13.
- GOLHANI, K. et al. A review of neural networks in plant disease detection using hyperspectral data. *Information Processing in Agriculture*, 2018. Elsevier BV, v. 5, n. 3, p. 354–371, set. 2018. Citado 3 vezes nas páginas 2, 5 e 6.
- GONÇALVES, A. et al. Importância da clínica de doenças de plantas na implementação da agricultura sustentável. 2010. Extensio: Revista Eletrônica de Extensão, v. 7, n. 9, p. 24–32, 2010. ISSN 1807-0221. Disponível em: <<https://periodicos.ufsc.br/index.php/extensio/article/view/1807-0221.2010v7n9p24>>. Citado na página 6.
- GOPALAKRISHNAN, K. et al. Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. *Construction and Building Materials*, 2017. Elsevier BV, v. 157, p. 322–330, dez. 2017. Citado na página 23.
- HAYKIN, S. *Redes Neurais: Princípios e Prática*. 2. ed. [S.l.]: Bookman, 2001. Citado 4 vezes nas páginas 11, 13, 15 e 16.
- HE, K. et al. Deep residual learning for image recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.]: IEEE, 2016. Citado 2 vezes nas páginas 2 e 25.
- HOLLANDER, M.; WOLFE, D. A. *Nonparametric Statistical Methods*. 2. ed. [S.l.]: WileyInterscience, 1999. Citado na página 44.
- HUANG, G. et al. Densely connected convolutional networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.]: IEEE, 2017. Citado 5 vezes nas páginas 2, 24, 29, 30 e 31.
- HUBEL, D. H.; WIESEL, T. N. Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, 1968. Wiley, v. 195, n. 1, p. 215–243, mar. 1968. Citado na página 18.
- HUBEL, D. H.; WIESEL, T. N. Early exploration of the visual cortex. *Neuron*, 1998. Elsevier BV, v. 20, n. 3, p. 401–412, mar. 1998. Citado na página 10.
- JORDAN, J. *Common architectures in convolutional neural networks*. 2018. Disponível em: <<https://www.jeremyjordan.me/convnet-architectures/inception>>. Acesso em: 21.06.2019. Citado na página 25.
- KERAS: The Python Deep Learning library. 2019. Disponível em: <<https://keras.io/>>. Citado na página 41.
- KINGMA, D. P.; BA, J. L. Adam: A method for stochastic optimization. In: . [S.l.]: 2015 ICLR International Conference on Learning Representations, 2015. Citado na página 33.
- KNUDSEN, E. Supervised learning in the brain. *The Journal of Neuroscience*, 1994. Society for Neuroscience, v. 14, n. 7, p. 3985–3997, jul. 1994. Citado na página 16.

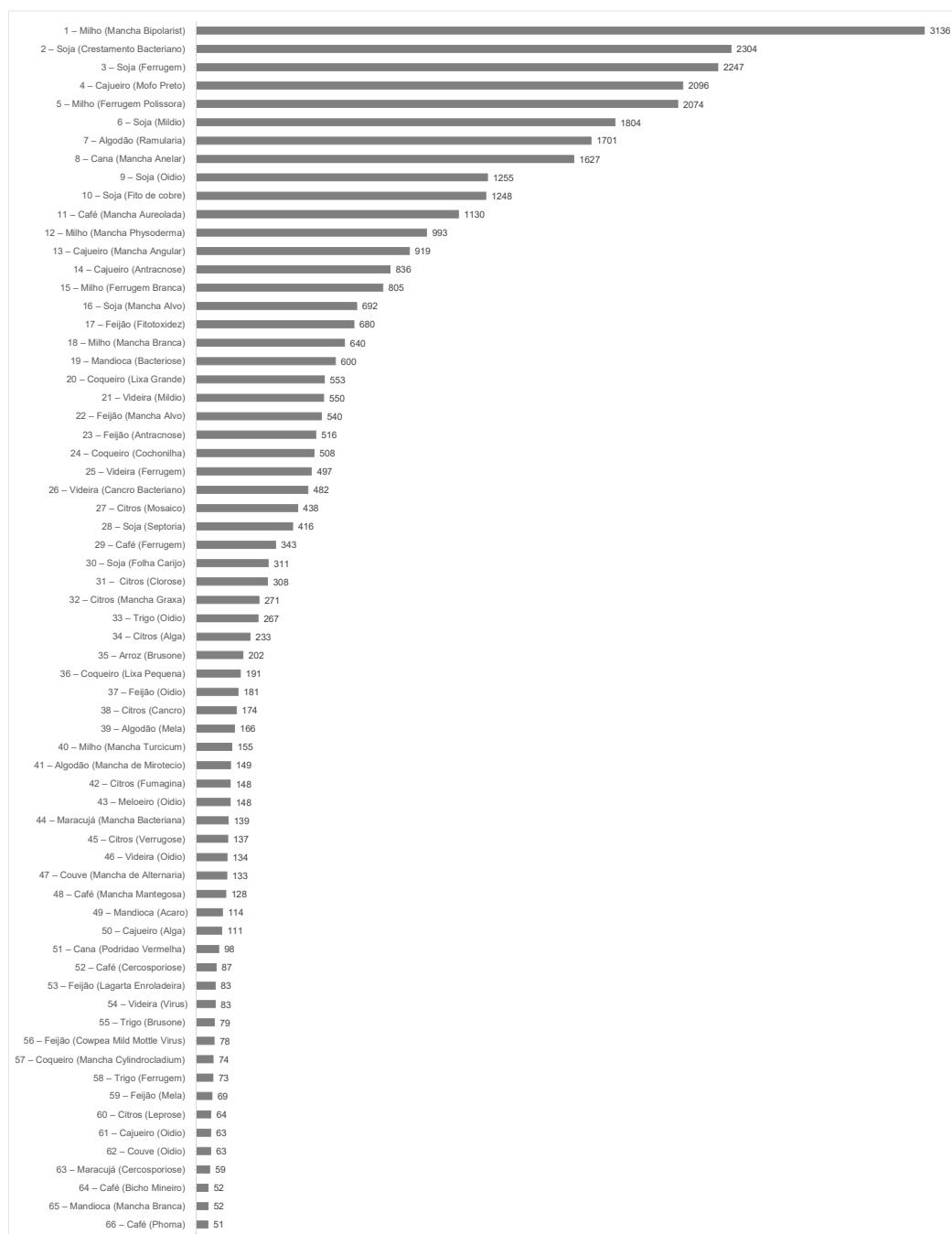
- LECUN, Y. et al. Handwritten digit recognition with a back-propagation network. In: TOURETZKY, D. S. (Ed.). *Advances in Neural Information Processing Systems 2*. Morgan-Kaufmann, 1990. p. 396–404. Disponível em: <<http://papers.nips.cc/paper/293-handwritten-digit-recognition-with-a-back-propagation-network.pdf>>. Citado 2 vezes nas páginas 2 e 18.
- LEE, C.-Y. et al. Deeply-Supervised Nets. In: LEBANON, G.; VISHWANATHAN, S. V. N. (Ed.). *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*. San Diego, California, USA: PMLR, 2015. (Proceedings of Machine Learning Research, v. 38), p. 562–570. Disponível em: <<http://proceedings.mlr.press/v38/lee15a.html>>. Citado na página 30.
- LIN, X.; WANG, X.; HAO, Z. Supervised learning in multilayer spiking neural networks with inner products of spike trains. *Neurocomputing*, 2017. Elsevier BV, v. 237, p. 59–70, maio 2017. Citado na página 16.
- LIU, B. et al. Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry*, 2017. MDPI AG, v. 10, n. 1, p. 11, dez. 2017. Citado 2 vezes nas páginas 1 e 3.
- LIU, S.; DENG, W. Very deep convolutional neural network based image classification using small training sample size. In: *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*. [S.l.]: IEEE, 2015. Citado 3 vezes nas páginas 2, 23 e 24.
- LU, Y. et al. Identification of rice diseases using deep convolutional neural networks. *Neurocomputing*, 2017. Elsevier BV, v. 267, p. 378–384, dez. 2017. Citado na página 3.
- MARQUES, A. C. R. et al. Ant genera identification using an ensemble of convolutional neural networks. *PLOS ONE*, 2018. Public Library of Science (PLOS), v. 13, n. 1, p. e0192011, jan. 2018. Citado 6 vezes nas páginas 3, 6, 7, 20, 21 e 22.
- MESSA, A. Determinantes da produtividade na indústria brasileira. In: *Produtividade no Brasil: desempenho e determinantes*. [S.l.]: IPEA, 2015. p. 23–42. Citado na página 1.
- MOEN, E. et al. Automatic interpretation of otoliths using deep learning. *PLOS ONE*, 2018. Public Library of Science (PLOS), v. 13, n. 12, p. e0204713, dez. 2018. Citado na página 10.
- MOHANTY, S. P.; HUGHES, D. P.; SALATHÉ, M. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 2016. Frontiers Media SA, v. 7, set. 2016. Citado 5 vezes nas páginas 1, 4, 5, 6 e 7.
- NAKANO, T. et al. A spiking neural network model of model-free reinforcement learning with high-dimensional sensory input and perceptual ambiguity. *PLOS ONE*, 2015. Public Library of Science (PLOS), v. 10, n. 3, p. e0115620, mar. 2015. Citado na página 16.
- OEI, R. W. et al. Convolutional neural network for cell classification using microscope images of intracellular actin networks. *PLOS ONE*, 2019. Public Library of Science (PLOS), v. 14, n. 3, p. e0213626, mar. 2019. Citado na página 10.
- OLIVEIRA, R.; ALVES, E.; MALQUI, C. Redes neurais convolucionais aplicadas à preensão robótica. In: *Anais do 13 Congresso Brasileiro de Inteligência Computacional*. Curitiba, PR: ABRICOM, 2018. Citado 3 vezes nas páginas 1, 18 e 19.

- ONIZ, Y.; KAYNAK, O. Control of a direct drive robot using fuzzy spiking neural networks with variable structure systems-based learning algorithm. *Neurocomputing*, 2015. Elsevier BV, v. 149, p. 690–699, fev. 2015. Citado na página 17.
- ÖZYURT, F. et al. Brain tumor detection based on convolutional neural network with neutrosophic expert maximum fuzzy sure entropy. *Measurement*, 2019. Elsevier BV, v. 147, p. 106830, dez. 2019. Citado na página 3.
- PANTRIDGE, E. R.; SPECTOR, L. Evolution of layer based neural networks. In: *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion - GECCO 16 Companion*. [S.l.]: ACM Press, 2016. Citado na página 17.
- PENHA, D. de P. *Rede Neural Convolutacional Aplicada à Identificação de Equipamentos Residenciais para Sistemas de Monitoramento não Intrusivo de Carga*. Dissertação (Mestrado) — Universidade Federal do Pará, Belém, 2018. Citado 6 vezes nas páginas 1, 7, 18, 22, 32 e 42.
- POLYAK, B. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 1964. Elsevier BV, v. 4, n. 5, p. 1–17, jan. 1964. Citado na página 42.
- POLYAK, B. T.; JUDITSKY, A. B. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 1992. Society for Industrial & Applied Mathematics (SIAM), v. 30, n. 4, p. 838–855, jul. 1992. Citado na página 41.
- PONTI, M. A.; COSTA, G. B. P. da. Como funciona o deep learning. In: *Tópicos em Gerenciamento de Dados e Informações*. [S.l.]: SBC, 2017. p. 63–93. Citado 3 vezes nas páginas 18, 20 e 23.
- POTJANS, W.; MORRISON, A.; DIESMANN, M. A spiking neural network model of an actor-critic learning agent. *Neural Computation*, 2009. MIT Press - Journals, v. 21, n. 2, p. 301–339, fev. 2009. Citado na página 15.
- POURZANGBAR, A. et al. Prediction of non-breaking wave induced scour depth at the trunk section of breakwaters using genetic programming and artificial neural networks. *Coastal Engineering*, 2017. Elsevier BV, v. 121, p. 107–118, mar. 2017. Citado na página 11.
- REDMON, J. et al. You only look once: Unified, real-time object detection. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.]: IEEE, 2016. Citado na página 3.
- RUDER, S. An overview of gradient descent optimization algorithms. 2016. 09 2016. Citado 2 vezes nas páginas 41 e 42.
- SAHA, S. *A Comprehensive Guide to Convolutional Neural Networks*. 2018. Disponível em: <<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>>. Acesso em: 20.06.2019. Citado na página 22.
- SANTOS, R. et al. Empirical study on rotation and information exchange in particle swarm optimization. *Swarm and Evolutionary Computation*, 2019. Elsevier BV, v. 48, p. 312–328, ago. 2019. Citado na página 45.

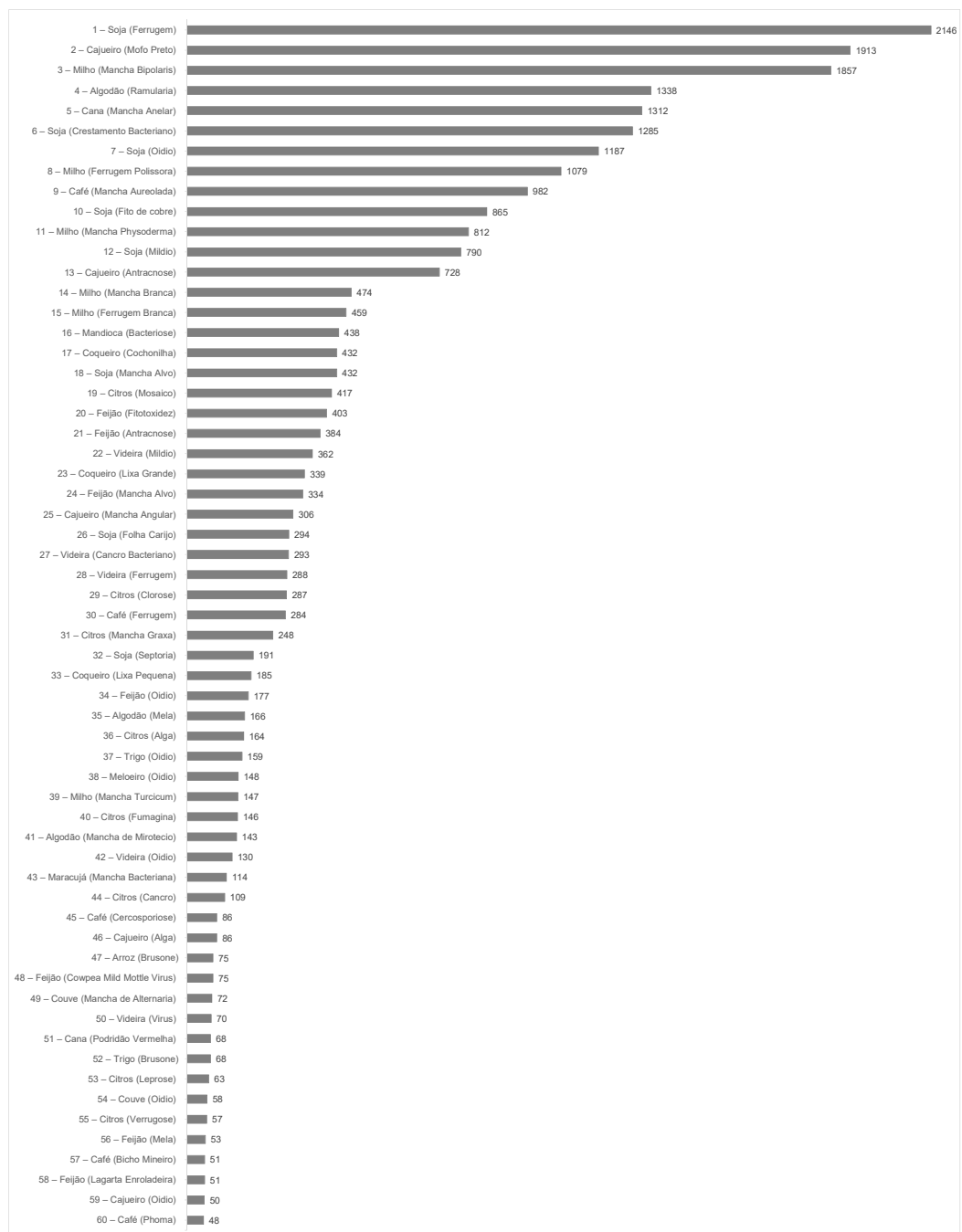
- SCHMIDHUBER, J. Deep learning in neural networks: An overview. *Neural Networks*, 2015. Elsevier BV, v. 61, p. 85–117, jan. 2015. Citado na página 33.
- SEETHA, J.; RAJA, S. S. Brain tumor classification using convolutional neural networks. *Biomedical and Pharmacology Journal*, 2018. Oriental Scientific Publishing Company, v. 11, n. 3, p. 1457–1461, set. 2018. Citado na página 3.
- S.G.M. *Combating plant diseases is key for sustainable crops*. 2011. Disponível em: <<https://www.sciencedaily.com/releases/2011/04/110411194819.htm>>. Citado na página 5.
- SHAFFER, J. P. Modified sequentially rejective multiple test procedures. *Journal of the American Statistical Association*, 1986. Informa UK Limited, v. 81, n. 395, p. 826–831, set. 1986. Citado na página 44.
- SILVA, G. L. F. da. *Diagnóstico de nódulos pulmonares em imagens de tomografia computadorizada usando redes neurais convolucionais evolutivas*. Tese (Doutorado) — Universidade Federal do Maranhão, São Luís, 2017. Citado na página 10.
- SLADOJEVIC, S. et al. Deep neural networks based recognition of plant diseases by leaf image classification. *Computational Intelligence and Neuroscience*, 2016. Hindawi Limited, v. 2016, p. 1–11, 2016. Citado na página 3.
- SOKOLOVA, M.; JAPKOWICZ, N.; SZPAKOWICZ, S. Beyond accuracy, f-score and roc: A family of discriminant measures for performance evaluation. In: *2006 AI Australasian Joint Conference on Artificial Intelligence (AI)*. [S.l.]: Springer, 2006. Citado na página 42.
- STRANGE, R. N.; SCOTT, P. R. Plant disease: A threat to global food security. *Annual Review of Phytopathology*, 2005. Annual Reviews, v. 43, n. 1, p. 83–116, set. 2005. Citado 2 vezes nas páginas 1 e 6.
- THAMES, J. L.; ABLER, R.; SCHAEFER, D. Parametric optimization of artificial neural networks for signal approximation applications. In: *Proceedings of the 49th Annual Southeast Regional Conference on - ACM-SE 11*. [S.l.]: ACM Press, 2011. Citado na página 12.
- THORSTENSEN, V.; GABRIEL, V. D. R.; MESQUITA, A. L. A participação de brasil e estados unidos na formulação das regras multilaterais do comércio agrícola. *Revista de Direito Internacional*, 2019. Centro de Ensino Unificado de Brasília, v. 15, n. 3, mar. 2019. Citado na página 1.
- TODA, Y.; OKURA, F. How convolutional neural networks diagnose plant disease. *Plant Phenomics*, 2019. American Association for the Advancement of Science (AAAS), v. 2019, p. 1–14, mar. 2019. Citado 6 vezes nas páginas 1, 2, 3, 5, 6 e 7.
- TRUJILLO, L.; MARTÍNEZ, Y.; MELIN, P. How many neurons? In: *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation - GECCO 11*. [S.l.]: ACM Press, 2011. Citado na página 17.
- VARGAS, A. C. G.; PAES, A.; VASCONCELOS, C. N. Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. In: *2015 29rd Conference on Graphics, Patterns and Images (SIBGRAPI)*. [S.l.]: SIBGRAPI, 2016. Citado 2 vezes nas páginas 20 e 33.

- XIE, S. et al. Aggregated residual transformations for deep neural networks. 2017. 11 2017. Citado 4 vezes nas páginas 27, 28, 29 e 49.
- XIE, Y. xin et al. AUV dead-reckoning navigation based on neural network using a single accelerometer. In: *Proceedings of the 11th ACM International Conference on Underwater Networks & Systems - WUWNet 16*. [S.l.]: ACM Press, 2016. Citado na página 11.
- YU, C. et al. Acral melanoma detection using a convolutional neural network for dermoscopy images. *PLOS ONE*, 2018. Public Library of Science (PLoS), v. 13, n. 3, p. e0193321, mar. 2018. Citado 2 vezes nas páginas 3 e 4.
- ZHANG, W. et al. Parallel distributed processing model with local space-invariant interconnections and its optical architecture. *Applied Optics*, 1990. The Optical Society, v. 29, n. 32, p. 4790, nov. 1990. Citado na página 18.

APÊNDICE A – Quantidade de imagens após o pré-processamento da base 32hard.



APÊNDICE B – Quantidade de imagens após o pré-processamento da base 64hard.



APÊNDICE C – Imagens da Base de Dados de Doenças de Plantas XDB.

