



INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES
Autarquia associada à Universidade de São Paulo

**Desenvolvimento de modelos de redes neurais convolucionais para avaliação
nutricional da *Brachiaria brizantha***

BRUNO ROVER DAL PRÁ

Tese apresentada como parte dos
requisitos para obtenção do Grau de
Doutor em Ciências na Área de Tecnologia
Nuclear - Reatores

Orientador: Prof. Dr. Delvonei Alves de
Andrade

São Paulo
2020

INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES
Autarquia associada à Universidade de São Paulo

**Desenvolvimento de modelos de redes neurais convolucionais para avaliação
nutricional da *Brachiaria brizantha***

Versão Corrigida
Versão Original disponível no IPEN

BRUNO ROVER DAL PRÁ

Tese apresentada como parte dos
requisitos para obtenção do Grau de
Doutor em Ciências na Área de
Tecnologia Nuclear - Reatores

Orientador: Prof. Dr. Delvonei Alves de
Andrade

São Paulo
2020

FOLHA DE APROVAÇÃO

Autor: **Bruno Rover Dal Prá**

Título: **Desenvolvimento de modelos de redes neurais convolucionais para avaliação nutricional da *Brachiaria brizantha*.**

Tese apresentada ao Programa de Pós-Graduação em Tecnologia Nuclear da Universidade de São Paulo para obtenção do título de Doutor em Ciências.

Data: 16/11/2020

Banca Examinadora

Prof. Dr.: Marcelo da Silva Rocha

Instituição: Instituto de Pesquisas Energéticas e Nucleares (IPEN)

Julgamento: Aprovado

Prof. Dr.: José Gomes Neto

Instituição: Inter Solution

Julgamento: Aprovado

Prof. Dr.: Vóldi Costa Zambenedetti

Instituição: Pontifícia Universidade Católica do Paraná (PUC-PR)

Julgamento: Aprovado

Autorizo a reprodução e divulgação total ou parcial deste trabalho, para fins de estudo e pesquisa, desde que citada a fonte.

Como citar:

DAL PRÁ, B. R. *Desenvolvimento de modelos de redes neurais convolucionais para avaliação nutricional da Brachiaria brizantha*. 2020. 120 p. Tese (Doutorado em Tecnologia Nuclear), Instituto de Pesquisas Energéticas e Nucleares, IPEN-CNEN/SP, São Paulo. Disponível em: (data de consulta no formato: dd/mm/aaaa)

Ficha catalográfica elaborada pelo Sistema de geração automática da Biblioteca IPEN/USP, com os dados fornecidos pelo(a) autor(a)

Dal Prá, Bruno Rover

Desenvolvimento de modelos de redes neurais convolucionais para avaliação nutricional da Brachiaria brizantha / Bruno Rover Dal Prá; orientador Delvonei Alves de Andrade. -- São Paulo, 2020.

120 p.

Tese (Doutorado) - Programa de Pós-Graduação em Tecnologia Nuclear (Reatores) -- Instituto de Pesquisas Energéticas e Nucleares, São Paulo, 2020.

1. Diagnose foliar. 2. Redes neurais profundas. 3. Modelos pré-treinados. 4. Transferência de aprendizado. I. Andrade, Delvonei Alves de, orient. II. Título.

DEDICATÓRIA

*Dedico este trabalho a toda minha família e em especial, a minha esposa Roseli, as minhas filhas Ana
Lívia e Júlia.*

AGRADECIMENTOS

A Deus por todas as bênçãos recebidas.

Em especial ao meu orientador Dr. Delvonei Alves de Andrade, por todo o tempo dedicado a mim, pelos conhecimentos compartilhados e amizade.

Aos pesquisadores Dr. Ernando Balbinot e Dr. Fabiano Gama de Souza, pelo apoio no desenvolvimento das amostras utilizadas nesta pesquisa.

Ao Instituto de Pesquisas Energéticas e Nucleares – IPEN, pelas instalações, conhecimento científica e grande apoio no desenvolvimento.

Ao instituto Federal de Rondônia – IFRO, pelo convênio firmado com o IPEN e pelo incentivo a capacitação.

Ao instituto Federal de Rondônia – Campus Colorado do Oeste, por disponibilizar as instalações, de fundamental importância para o desenvolvimento desta pesquisa.

A Nvidia, pelo apoio e doação da placa GPU Titan XP utilizada no desenvolvimento e testes dos modelos pesquisados.

A CAPES, por dinamizar o conhecimento técnico por meio dos programas de Pós-Graduação que administra.

Aos amigos e professores, que de alguma forma contribuíram para a realização deste trabalho.

RESUMO

DAL PRÁ, B. R. *Desenvolvimento de modelos de redes neurais convolucionais para avaliação nutricional da Brachiaria brizantha*. 2020. 120 p. Tese (Doutorado em Tecnologia Nuclear) – Instituto de Pesquisas Energéticas e Nucleares – IPEN-CNEN/SP. São Paulo.

Uma das formas de realizar avaliação nutricional em plantas é pela análise dos sintomas apresentados visualmente em suas folhas. Esta avaliação é realizada pela técnica da diagnose foliar por pessoas especializadas sendo, em sua grande maioria de forma manual, assim, necessitando de mão de obra especializada o que acaba dificultando sua utilização, principalmente na região amazônica onde esta mão de obra ainda é escassa. O Objetivo deste estudo foi desenvolver e aplicar modelos de Redes Neurais Convolucionais (CNN), que realizam a classificação do estado nutricional da pastagem *Brachiaria brizantha* cv. Marandu utilizando a imagem de suas folhas. Para isso, foram implementados e avaliados seis modelos de CNNs: o baseado na AlexNet e os pré-treinados VGG-16, VGG-19, *Inception-V3*, ResNet-50 e MobileNetV2. Todos os modelos pré-treinados utilizaram transferência de aprendizado que permite economizar tempo e obter um melhor resultado na identificação das deficiências. Para classificá-las foi criado um conjunto de dados de imagens, tanto de folhas deficientes como saudáveis, cultivadas em casa de vegetação para que sirvam como as informações a serem aprendidas pelos modelos, durante o treinamento. Estes modelos classificam as deficiências de Potássio, Nitrogênio e Fósforo, além de identificar se a planta é saudável. Esta tecnologia pode melhorar a produção das pastagens brasileiras e, conseqüentemente, melhorar o índice de animais por área de pastagem, desta forma, contribuindo para uma produção mais sustentável, principalmente na região amazônica. De todos os modelos testados o que apresentou a melhor acurácia foi o VGG-16 com 96,93%, em dados de teste.

Palavras-chave:

Diagnose foliar, redes neurais profundas, modelos pré-treinados, transferência de aprendizado.

ABSTRACT

DAL PRÁ, B. R. **Development of models of convolutional neural networks for nutritional assessment of *Brachiaria brizantha***. 2020. 120 p. Tese (Doutorado em Tecnologia Nuclear) – Instituto de Pesquisas Energéticas e Nucleares – IPEN-CNEN/SP. São Paulo.

One possibility to perform nutritional assessment on plants is by analyzing the symptoms visually presented on their leaves. This evaluation is carried out by the technique of leaf diagnosis by specialized people, most of which is done manually, thus requiring specialized labor, which makes it difficult to use, especially in the Amazon region where this labor is still scarce. The objective of this study was the development and apply of Convolutional Neural Networks (CNN) models, which perform the classification of the *Brachiaria brizantha* cv. Marandu nutritional status using the image of its leaves. Six CNN models were implemented and evaluated: one based on AlexNet and the pre-trained VGG-16, VGG-19, Inception-V3, ResNet-50 and MobileNetV2. All pre-trained models used transfer of learning that saves time and obtains a better result in the identification of deficiencies. To classify them, a set of image data was created, both for deficient and healthy leaves, grown in a greenhouse to serve as the information to be learned by the models during training. These models classify the deficiencies of Potassium, Nitrogen and Phosphorus, in addition to identifying whether the plant is healthy. This technology can improve the production of Brazilian pastures and, consequently, improve the number of animals per pasture area, thus contributing to a more sustainable production, especially in the Amazon region. Of all the models tested, the best accuracy was VGG-16 with 96.93%, in test data.

Keywords:

Leaf diagnosis, deep neural networks, pre-trained models, transfer of learning.

LISTA DE TABELAS

Tabela 1 – Alterações provocadas nas plantas por deficiência de Nitrogênio.....	27
Tabela 2 – Alterações provocadas nas plantas por deficiência de Fósforo.	28
Tabela 3 – Alterações provocadas nas plantas por deficiência de Potássio.....	29
Tabela 4 – Período de plantio e registro das imagens.....	72
Tabela 5 – Lista de nutrientes e o número de amostras após o Data Augmentation.	78
Tabela 6 – Resultados utilizando as métricas <i>Precision</i> , <i>Recall</i> e <i>F1_Score</i> para amostras de teste.	87
Tabela 7 – Resultados utilizando as métricas <i>Precision</i> , <i>Recall</i> e <i>F1_Score</i> para amostras de teste (Modelo VGG-16).	92
Tabela 8 – Resultados utilizando as métricas <i>Precision</i> , <i>Recall</i> e <i>F1_Score</i> para amostras de teste (Modelo VGG-19).	95
Tabela 9 – Resultados utilizando as métricas <i>Precision</i> , <i>Recall</i> e <i>F1_Score</i> para amostras de teste (modelo Inception-V3).	98
Tabela 10 – Resultados utilizando as métricas <i>Precision</i> , <i>Recall</i> e <i>F1_Score</i> para amostras de teste (modelo ResNet50).	101
Tabela 11 – Resultados utilizando as métricas <i>Precision</i> , <i>Recall</i> e <i>F1_Score</i> para amostras de teste (Modelo MobileNetV2).	104
Tabela 12 – Resultados de acurácia para validação e teste para todos os modelos avaliados.	106

LISTA DE ILUSTRAÇÕES

Figura 1 – Divisão do dataset para aprendizado de máquina.	33
Figura 2 – Neurônio Artificial.	36
Figura 3 – Rede Neural Multicamada.	37
Figura 4 – Representação da relação entre o campo da Inteligência Artificial e seus sistemas especialistas Machine Learning e Deep Learning.	38
Figura 5 – Diferença entre aprendizado de máquina tradicional e aprendizado profundo. .	39
Figura 6 – Diferença entre uma camada tradicional totalmente conectada e uma camada convolucional.	41
Figura 7 – Composição básica de um modelo de rede neural convolucional.	41
Figura 8 – Representação de uma imagen RGB para uma CNN.	42
Figura 9 – Campo receptivo local neurônio de entrada.	43
Figura 10 – Deslizando a janela sobre a imagem	43
Figura 11 – Exemplo de operação de convolução.	45
Figura 12 – Filtro se movendo através da imagem calculando novos valores de ativação para os neurônios de saída.	45
Figura 13 – Exemplo de utilização de <i>padding</i>	47
Figura 14 – <i>Max pooling</i> com tamanho 2x2.	48
Figura 15 – Função de ativação RELU.	49
Figura 16 – Funcionamento da função Softmax.	50
Figura 17 – Modelos VGG-16 E VGG-19	53
Figura 18 – Módulo Inception, versão naïve.	54
Figura 19 – Módulo Inception com dimensões reduzidas.	55
Figura 20 – Modelo GoogLeNet.	56
Figura 21 – Fatoração da convolução 5x5 agora representada por duas de 3x3.	57
Figura 22 – Bancos de filtros expandidos.	57
Figura 23 – Diagrama esquemático do Inception V3.	58
Figura 24 – Erro de treinamento (esquerda) e erro de teste (direita) no CIFAR–10 com redes convencionais de aprendizado profundo de 20 camadas e 56 camadas.	59
Figura 25 – Elemento básico da rede residual	60
Figura 26 – Modelo ResNet 34 camadas(HE et al., 2016)	60

Figura 27 – Bloco básico imagem (A) e bloco com design de gargalo (Bottleneck Design) (B).....	61
Figura 28 – Convolução padrão (A), convolução separável em profundidade (B) com BN e ReLU(HOWARD et al., 2017).	62
Figura 29 – O processo geral de convolução separável em profundidade.....	62
Figura 30 – Arquitetura geral do MobileNetV2.	63
Figura 31 – Blocos convolucionais em rede MobileNetV1 e MobileNetV2.....	64
Figura 32 – Rede neural normal e rede neural depois do dropout.....	64
Figura 33 – Matriz de confusão.	67
Figura 34 – Visão geral do sistema proposto de aprendizagem profunda.	70
Figura 35 – Unidade experimental composta por tubos de PVC.....	73
Figura 36 – Semeadura e emergência do capim.	73
Figura 37 – Adubação dos devidos tratamentos.	74
Figura 38 – Imagem folha com deficiência de Nitrogênio (A), Fósforo (B), Potássio (C) a nível do substrato e planta saudável (D).....	74
Figura 39 – Matriz das imagens segmentadas.	76
Figura 40 – Divisão do conjunto de dados.	77
Figura 41 – Exemplo de aumento de dados usados nas imagens.	78
Figura 42 – VGG-16 acurácia do modelo em relação a taxa de aprendizado.	80
Figura 43 – Arquitetura em camadas do modelo proposto da CNN.....	83
Figura 44 – Gráficos de acurácia(A) e gráfico de perda(B).	86
Figura 45 – Matriz de confusão de modelo utilizando dados de teste.....	86
Figura 46 – Arquitetura VGG-16 modificada.....	89
Figura 47 – Gráficos de acurácia e gráfico de perda.	91
Figura 48 – Matriz de confusão de modelo utilizando dados de teste.....	91
Figura 49 – Arquitetura VGG-19 modificada.....	93
Figura 50 – Gráficos de acurácia e gráfico de perda.	94
Figura 51 – Matriz de confusão de modelo utilizando dados de teste.....	95
Figura 52 – Arquitetura Resnet-50 modificada.	96
Figura 53 – Gráficos de acurácia e gráfico de perda.	97
Figura 54 – Matriz de confusão de modelo utilizando dados de teste.....	98
Figura 55 – Arquitetura Resnet-50 modificada.	99
Figura 56 – Gráfico de acurácia e gráfico de perda.....	100

Figura 57 – Matriz de confusão de modelo utilizando dados de teste.....	100
Figura 58 – Arquitetura MobileNetV2 modificada.	102
Figura 59 – Gráficos de acurácia e gráfico de perda.	103
Figura 60 – Matriz de confusão de modelo utilizando dados de teste.....	104

LISTA DE ABREVIATURAS E/OU SIGLAS

Adam – Método para Otimização Estocástica

AlexNet – Rede neural convolucional, projetada por Alex Krizhevsky em colaboração com Ilya Sutskever e Geoffrey Hinton, que era Ph.D. de Krizhevsky

B – Boro

Ca – Cálcio

CIFAR-10 – Coleção de imagens comumente usadas para treinar algoritmos de aprendizado de máquina e visão computacional.

Cl – Cloro

CNN – Rede Neural Convolucional (*Convolutional Neural Network*)

CPU – Unidade Central de Processamento

Cu – Cobre

CUDA – Plataforma de computação paralela

DAE – Dias após a emergência

DBC – Delineamento em blocos casualizado

FC – Totalmente conectado (Full Connected)

FCNN – Rede Neural Totalmente Conectada (Fully Connected Neural Network)

Fe – Ferro

FN – Falso negativo (False negative)

FP – Falso positivo (False positive)

GoogLeNet – Rede neural convolucional desenvolvida pela Google

GPU – Unidades de processamento gráficos

IA – Inteligência artificial (Artificial intelligence)

IFRO – Instituto Federal de Educação, Ciência e Tecnologia de Rondônia

ILSVRC – Desafio de Reconhecimento Visual em Grande Escala – ImageNet (ImageNet Large Scale Visual Recognition Challenge)

ImageNet – Grande banco de dados visual projetado para uso em pesquisa de software de reconhecimento de objetos visuais.

Inception-ResNet – Rede Neural Residual profunda

Inception-V1 – Primeira versão da Rede Neural Convolucional Inception do Google

Inception-V2 – Segunda versão da Rede Neural Convolucional Inception do Google

Inception-V3 – Terceira versão da Rede Neural Convolutacional Inception do Google

IPEN – Instituto de Pesquisas Energéticas e Nucleares

K – Potássio

Kappa – Método estatístico para avaliar o nível de concordância ou reprodutibilidade entre dois conjuntos de dados

Mg – Magnésio

Min-Max – Mínimo e Máximo

MLP – Perceptron multicamadas (Multi Layer Perceptron)

Mn – Manganês

Mo – Molibdênio

MobileNets – Arquitetura simplificada que usa convoluções separáveis em profundidade para construir redes neurais profundas e leves

MobileNetV2 – Segunda versão da MobileNet

MS COCO – Conjunto de dados de detecção, segmentação e legendagem de objetos em larga escala

N – Nitrogênio

NoSql – Not Only SQL, um termo genérico que representa os bancos de dados não relacionais

NPK – Adubação de Nitrogênio, Fósforo e Potássio

OPENCV – Biblioteca de Código aberto para visão computacional (Open Source Computer Vision Library)

P – Fósforo

PVC – Polímero de adição policloreto de vinila

RAM – Memória de acesso randômico

ReLU – Função de ativação linear retificada (Rectified Linear Activation Function)

ResNet-50 – Rede Neural Residual profunda com 50 camadas

ResNet-34 – Rede Neural Residual profunda com 34 camadas

RGB – Abreviatura de um sistema de cores aditivas em que o Vermelho, o Verde e o Azul

ROC – Curva Característica de Operação do Receptor

S – Enxofre

SGD – Descida de gradiente estocástico (Stochastic Gradient Descent)

Softmax – Função exponencial normalizada

TFLOPS – Teraflops

TN – Verdadeiro negativo (True negative)

TP – Verdadeiro positivo (True positive)

VGG16 – Grupo de Geometria Visual com 16 camadas (Visual Geometry Group with 19 layers)

VGG19 – Grupo de Geometria Visual com 19 camadas (Visual Geometry Group with 19 layers)

Zn – Zinco

SUMÁRIO

1 INTRODUÇÃO.....	19
2 OBJETIVO DO TRABALHO.....	24
2.1 Objetivo geral.....	24
2.2 Objetivos específicos.....	24
3 REVISÃO DA LITERATURA.....	25
3.1 Nutrição Mineral das Plantas	25
3.2 Micronutrientes.....	25
3.3 Macronutrientes.....	25
3.3.1 Nitrogênio	26
3.3.2 Fósforo	27
3.3.3 Potássio	28
3.4 Diagnose Foliar	29
3.5 Processamento de Imagem.....	30
3.6 Aprendizado de Máquina.....	31
3.6.1 Componentes do Aprendizado de Máquina.....	32
3.7 Redes Neurais Artificiais.....	34
3.8 Redes Neurais Artificiais Multicamadas	36
3.9 <i>Deep Learning</i>	37
3.10 Redes Neurais Convolucionais.....	39
3.10.1 Campos Receptivos Locais.....	42
3.10.2 Camadas Convolucionais.....	44
3.10.3 <i>Padding</i>	46
3.10.4 <i>Pooling</i>	47
3.10.5 Função de ativação ReLU(<i>Rectified Linear Activation Function</i>)	48
3.10.6 Função Softmax	49

3.10.7 Camadas Totalmente Conectadas (Densas).....	50
3.10.8 Treinamento (Backpropagation).....	50
3.11 Redes VGGNet.....	52
3.12 Redes <i>Inception V3</i>	54
3.13 ResNet	58
3.14 MobileNetV2	61
3.15 Técnicas de Regularização	64
3.15.1 <i>Dropout</i>	64
3.15.2 <i>Data Augmentation</i>	65
3.15.3 <i>Transfer Learning</i>	65
3.16 Métricas de Avaliação	66
3.16.1 Matriz de Confusão.....	66
3.16.2 Acurácia (<i>Accuracy</i>)	67
3.16.3 Precisão (<i>Precision</i>).....	68
3.16.4 Revocação/Sensibilidade (Recall)	68
3.16.5 <i>F1_Score</i>	69
4 METODOLOGIA.....	70
4.1 Cultivo e Aquisição das Imagens.....	71
4.2 Processamento das Imagens.....	75
4.2.1 Aplicação do <i>Data Augmentation</i>	77
4.2.2 Normalização dos Dados com Min-Max.....	78
4.3 Modelos Desenvolvidos.....	79
4.3.1 Configurações dos parâmetros das Redes Neurais Convolucionais aplicadas ...	79
4.3.2 Tecnologia Utilizada.....	81
4.3.3 Treinamento das redes neurais convolucionais	81
4.3.4 Modelo baseado na AlexNet.....	82
5 RESULTADOS	85
5.1 Resultados do modelo baseado na AlexNet	85

5.2 Modelo VGG-16 modificado	88
5.2.1 Resultados do modelo VGG-16	90
5.3 Modelo VGG-19 modificado	92
5.3.1 Resultados do modelo VGG-19	93
5.4 Modelo <i>InceptionV3</i> modificada	95
5.4.1 Resultados do modelo Inception-V3	96
5.5 Modelo ResNet-50 modificada	98
5.5.1 Resultados do modelo ResNet-50	99
5.6 Modelo MobileNetV2 modificada	101
5.6.1 Resultados do modelo MobileNetV2	102
5.7 Desempenho dos diversos modelos testados	105
6 CONCLUSÃO	107
7 TRABALHOS FUTUROS	109
REFERÊNCIAS BIBLIOGRÁFICAS	110
GLOSSÁRIO	119

1 INTRODUÇÃO

O Brasil é atualmente o segundo maior produtor e o maior exportador mundial de carne bovina. Praticamente toda a produção brasileira tem como base as pastagens (produção extensiva), considerada a forma mais econômica e prática de produzir e oferecer alimentos para os rebanhos. As pastagens, portanto, desempenham papel fundamental na pecuária brasileira, garantindo baixos custos de produção (DIAS-FILHO, 2015). Sendo assim, a degradação delas é um dos maiores problemas enfrentados, hoje, pelos bovinocultores brasileiros (BRASÍLIA et al., 2012).

Nessa perspectiva, o estado de Rondônia exportou em 2019, 185,3 mil toneladas de carnes e derivados de bovinos, totalizando em US\$ 631,2 milhões. Sua participação está com uma fatia de 10% de toda a carne bovina exportada pelo Brasil (ABRAFRIGO, 2019). Existe uma tendência de crescimento das exportações e, conseqüentemente, um aumento no número de animais e de pastagens cultivadas, necessitando de um melhor controle para a obtenção de números de produtividade mais satisfatórios (TABORDA, 2015). Isso consiste em melhorar a qualidade da alimentação dos próprios bovinos, pois um animal com boa genética só expressará seu potencial de produção se estiver corretamente alimentado (ALVIM et al., 2002).

Em se tratando da alimentação do rebanho brasileiro, a *Brachiaria brizantha* é um cultivar introduzido, no país, em 1952, mas é originária da África do sul. A opção pelo seu cultivo justificou-se pelo fato de ser considerada uma excelente forrageira, perene e com grande produção de massa foliar de boa qualidade, resistente ao pastejo e ao pisoteio, além de proteger o solo contra a erosão (ALVIM et al., 2002; BIANCO et al., 2005). Os capins do gênero *Brachiaria* ocupam espaços cada vez maiores na pecuária brasileira. Em menos de vinte anos, após sua implantação, por ser uma planta pouco exigente de solos ricos, ela se configurou como suporte alimentar essencial na criação de gado, tanto de corte quanto de leite. Como as pastagens continuam sendo tratadas como culturas extrativistas e de utilização extensiva, não é oferecida a atenção necessária ao manejo e correção da fertilidade do solo (COSTA et al., 2006). Além da inexistência dessa correção, por se tratar de uma cultura sem a realização de replantios anualmente, é necessária a manutenção da fertilidade do solo e, conseqüentemente, de sua produtividade.

O Brasil possui uma área de pastagem de aproximadamente 180 milhões de hectares, dessas 70% são do tipo *Brachiaria brizantha* e, desse montante, apenas 20% encontra-se em condições ótimas ou adequadas (PASTAGEM.ORG, 2018).

Segundo Herling (2007), as pastagens não eram consideradas culturas e não recebiam os devidos cuidados com relação ao estado nutricional e ao seu manejo. Isso resultou em imensas áreas de pastagens degradadas e originou um grande problema nacional, pois uma parte dessas pastagens ocupam solos agricultáveis (MARCHESIN et al., 2007).

O solo é o meio pelo qual a planta, por intermédio da absorção radicular, obtém os elementos minerais essenciais. Quando o meio não tem e/ou não fornece as quantidades adequadas dos nutrientes, passa a ser avaliado pela análise química do solo. Nesse caso, as plantas não terão as suas exigências nutricionais atendidas. Haverá, portanto, redução do crescimento e produção das culturas devido à eficiência nutricional (FAQUIN, 2002).

Pode-se realizar diagnóstico do estado nutricional das plantas utilizando a técnica da diagnose visual, pois a falta ou o excesso de um dado elemento provoca sempre a mesma manifestação de anormalidade visível em qualquer que seja a espécie (MALAVOLTA, 1980).

As folhas refletem bem o nível de elementos do solo. Uma análise química pode ser substituída por uma análise da folha, sendo assim uma amostra do solo pode ser substituída por uma amostra da folha (MALAVOLTA; ROMERO, 1975).

Uma forma de se avaliar a fertilidade do solo é realizar uma análise foliar, utilizando as características de suas folhas. Dessa maneira, consegue-se verificar as deficiências nutricionais. Essa análise, no Brasil, começou a ser realizada, no fim do século XIX, no Instituto Agrônomo de Campinas (SP) (MALAVOLTA, 2006).

A diagnose foliar pode ser utilizada para avaliação do estado nutricional das plantas e a determinação ou ajustes das doses de adubo. Com isso, utilizando a avaliação nutricional, pode-se realizar a identificação da fome escondida, que seria a ocorrência de menor crescimento e menor produção sem, entretanto, aparecerem os sintomas típicos. Também é possível realizar o levantamento do estado nutricional, indicações de interações entre elementos, indicação de exigências nutricionais diferentes entre espécies e variedades, indicação da relação entre nutrição, pragas e moléstias e identificação dos fatores não nutricionais (MALAVOLTA, 2006).

Com essa técnica aplicada em culturas temporárias, as análises podem ser feitas com antecedência e servir como um guia para correções, principalmente, em anos seguintes. Diferente de culturas perenes, como o caso de pastagens, podem ser imediatamente úteis para correção das deficiências encontradas (MALAVOLTA, 2006).

Com a determinação de doses de adubo, pode ser realizado o estabelecimento, tanto da quantidade a ser produzida quanto da sua qualidade, bem como ajustar as doses dentro do ano ou no seguinte. Existe, ainda, a possibilidade de identificar a eficiência da época e da localização do adubo, com respeito à colheita, qualidade e meio, para assim realizar a comparação de fontes de nutrientes e identificar os elementos responsáveis pelo efeito do produto (MALAVOLTA, 2006).

Os nutrientes minerais das plantas estão divididos em macronutrientes e micronutrientes. De acordo com as quantidades encontradas no solo (MALAVOLTA, 1976). Os macronutrientes são aqueles que possuem uma alta concentração no solo e, com isso, tendo uma grande influência na produtividade.

Após a formação da pastagem, essa apresenta bons níveis de produtividade e terá sua fertilidade incrementada pela incorporação das cinzas, a qual possui um tempo médio de produtividade entre 3 a 5 anos de uso. A partir desse momento, tem-se o início da queda de produtividade, com incremento de plantas invasoras pelo motivo da cultura não conseguir sustentar seu nível de produtividade com baixos níveis de fertilidade. Dos minerais mais limitantes estão o Nitrogênio (N), Potássio (K) e o Fósforo (P), o qual é identificado em pastagens com alto grau de degradação. Esses três elementos passam a ser grandes limitadores de produção, em decorrência dos baixos teores de matéria orgânica no solo (TOWNSEND, C.R.; COSTA, N. DE L.; MENDES, A.M.; PEREIRA, R. DE G.A.; MAGALHÃES, 2001).

Os métodos atualmente utilizados para identificar possíveis deficiências nutricionais das plantas consistem, basicamente, em diagnose visual e diagnose foliar, sendo os métodos passíveis de erros de interpretação (MARIN, 2012). Para minimizar as possíveis falhas e criar uma ferramenta que realize a apreciação desses dados com eficiência, existem métodos computacionais capazes de realizar o reconhecimento de padrões e análise de imagens, podendo, dessa forma, serem aplicados nesse problema. Um deles é a metodologia das redes neurais artificiais e aprendizagem profunda, tornando-se uma das mais importantes tecnologias para classificação de imagens (BARBEDO, 2018; LU et al., 2017b).

Vários estudos foram realizados, utilizando aprendizagem profunda, aplicados a reconhecimento de imagens de patologias em plantas. As mais promissoras utilizam a arquitetura de redes neurais convolucionais (CNN) e modelos pré-treinados (HU; WU; et al., 2019; SUH et al., 2018; SUN et al., 2018; TRAN et al., 2019). Para a identificação de doenças em folhas, alguns autores propuseram, recentemente, o uso de CNN para realizar a

identificação de tais doenças com o auxílio de imagens digitais extraídas das folhas (AMARA et al., 2017; ARNAL BARBEDO, 2019; CHEN et al., 2019; GEETHARAMANI; J., 2019; HU; YANG; et al., 2019; JI et al., 2020; LU et al., 2017a; MA et al., 2018a; TM et al., 2018; ZHANG, S. et al., 2019; ZHANG, X. et al., 2018a).

As CNNs também foram utilizadas para identificação de formas em imagens, sendo aplicadas na identificação do tamanho e qualidade de grãos de cevada (KOZŁOWSKI et al., 2019), tal como na detecção e contagem de órgãos vegetais, no cultivo da uva (GRIMM et al., 2019).

TRAN *et al.* (2019) implementaram modelos de CNNs para realizar a identificação de deficiência dos macronutrientes Cálcio, Nitrogênio e Potássio. Em seu estudo, obteve resultados com acurácia de 87,27% com uma rede neural convolucional conhecida como *Inception-ResNet* (Rede Neural Residual profunda) (TRAN et al., 2019). GHOSAL *et al.* (2018), além de realizar análises de duas deficiências nutricionais por meio de uma CNN, também conseguiu diagnosticar, na mesma rede, cinco patologias presentes nas folhas. Com isso, obteve-se um excelente desempenho, atingindo uma acurácia total de 94.13% (GHOSAL et al., 2018).

Neste estudo, propõem-se a utilização de 6 modelos para avaliação do estado nutricional de Nitrogênio, Fósforo, Potássio e planta saudável para pastagens do tipo *Brachiaria brizantha* cv. Marandu (1984), com o uso da tecnologia baseada em redes neurais convolucionais profundas (CNN). Para a implementação do sistema foi necessária a criação de uma base de dados de imagens das folhas do cultivar, induzindo as deficiências, totalizando 284 imagens entre saudáveis e deficientes. Dos modelos implementados, o que apresentou melhor resposta de classificação em teste foi o VGG-16, atingindo uma acurácia de 96,93%.

Esta tese está organizada em seis capítulos. Neste capítulo é realizada a introdução e a contextualização da pesquisa, apresentando a relevância e a contribuição do trabalho. Já os objetivos são descritos no capítulo 2.

No capítulo 3 é realizada uma revisão de literatura relacionada ao tema da tese. Inicialmente, são abordados os temas como nutrição de plantas e diagnósticos, tópicos de aprendizado de máquina e *Deep Learning*, tais como modelos de redes neurais convolucionais, finalizando com métricas de avaliação.

A metodologia aplicada nesta pesquisa é apresentada no capítulo 4 e demonstra como realizar o cultivo e criação das amostras para preparação do conjunto de dados. Em

seguida é apresentada a implementação de todos os modelos de CNNs utilizados, bem como, o processo de configuração e treinamento.

Já os resultados são apresentados no capítulo 5, sendo descritos para todos os modelos. Este capítulo é finalizado com um comparativo entre estes modelos e o de melhor resultado é apresentado mais detalhadamente.

Por fim, no capítulo 6 são apresentadas as conclusões finais do trabalho e as recomendações de projetos futuros.

2 OBJETIVO DO TRABALHO

2.1 Objetivo geral

O objetivo deste trabalho foi desenvolver e aplicar modelos de redes neurais convolucionais capazes de realizar avaliação nutricional do solo por meio de diagnose foliar da pastagem *Brachiaria brizantha* cv. Marandu cultivada em casa de vegetação.

2.2 Objetivos específicos

- Cultivar a pastagem em casa de vegetação, induzindo as deficiências que serão classificadas;
- Catalogar as imagens das plantas com informações do tipo da deficiência e sua quantidade de nutrientes;
- Criar conjunto de dados (*Dataset*) de imagens para ser utilizado no aprendizado e testes do sistema;
- Realizar o pré-processamento das imagens para que seja possível alimentar o sistema inteligente;
- Implementar e aplicar modelos de redes neurais convolucionais para a análise e comparação dos dados;
- Testar e avaliar os modelos computacionais.

3 REVISÃO DA LITERATURA

Este capítulo apresenta as premissas conceituais necessárias para a compreensão da proposta. São discutidos temas como nutrição de plantas e diagnósticos, redes neurais do tipo *Fully Connected Neural Network* (FCNN), *Convolutional Neural Network* (CNN), *Transfer Learning*, redes pré-treinadas (VGG-16, VGG-19, *Inception-V3*, Resnet-50 e MoblieNetV2).

3.1 Nutrição Mineral das Plantas

Todas as espécies de vegetais necessitam de elementos minerais para se desenvolverem, sendo eles absorvidos do meio em que vivem (MALAVOLTA, 1980). Dessa forma, todas as plantas necessitam ser alimentadas para conseguirem se desenvolver. Para isso é necessária a realização da reposição de minerais essenciais para atenderem a esse crescimento. Esses elementos minerais absorvidos pela planta são classificados da seguinte maneira: Essenciais, benéfico e tóxicos (MALAVOLTA, 2006).

São minerais Essenciais aqueles indispensáveis às plantas. São classificados dependendo da quantidade presentes no solo e se classificam em macronutrientes e micronutrientes.

3.2 Micronutrientes

Os micronutrientes se apresentam em pequenas quantidades no solo, mas são essenciais à obtenção de uma boa produção, são eles: zinco (Zn), cobre (Cu), ferro (Fe), manganês (Mn), molibdênio (Mo), boro (B) e cloro (Cl).

3.3 Macronutrientes

Os macronutrientes são os de maior concentração no solo e os mais exigidos pelos cultivos, são os seguintes elementos, N, P, K, Ca, Mg e S. Eles estão divididos em dois grandes grupos, isso dependendo da quantidade exigida pela planta:

Macronutrientes primários – N, P e K

Macronutrientes secundários – Ca, Mg e S.

Para o estudo foram avaliados os macronutrientes primários, por se apresentarem em maior quantidade no solo e serem, pelas plantas, a maior exigência (IPNI, 1998; MALAVOLTA, 1980).

3.3.1 Nitrogênio

O Nitrogênio é o elemento mais exigido nas plantas e o que se encontra em maior concentração (FAQUIN, 2005), é responsável por 5% da matéria orgânica do solo. O elemento é absorvido pelas raízes e transportado para a parte aérea através do xilema (tecido das plantas vasculares por onde circula a água com sais minerais dissolvidos) e facilmente distribuído pela planta via floema (tecido das plantas vasculares responsável pela condução de seiva elaborada em seu corpo). Quando os níveis de Nitrogênio estão em níveis adequados as colorações das folhas tendem a verde escuro, caso contrário causa clorose (amarelecimento) das folhas (IPNI, 1998).

A deficiência de Nitrogênio é inicialmente visualizada pelas folhas mais velhas, pois quando existe tal deficiência, o Nitrogênio dessas folhas é transportado para as mais jovens e órgãos. Por esse motivo, as folhas mais velhas apresentam sintomas de deficiência primeiro do que as folhas mais jovens (FAQUIN, 2005; MALAVOLTA, 2006). Essas redistribuições resultam em um colapso de cloroplastos acontecendo uma diminuição da clorofila nas folhas, gerando um amarelecimento das folhas mais velhas demonstrando a deficiência de tal nutriente (FAQUIN, 2002, 2005).

O Nitrogênio é o grande responsável pela melhoria nas características nutricionais das gramíneas forrageiras (BONFIM-SILVA; MONTEIRO, 2010).

A deficiência desse nutriente é o grande responsável pela degradação das pastagens brasileiras, principalmente as da variedade *Brachiaria*. Ele se destaca por ser um dos principais responsáveis pela recuperação de áreas degradadas desse capim, sendo assim, é considerado de grande importância no estudo em questão (COSTA et al., 2008).

Uma forma de se determinar a deficiência de Nitrogênio em uma planta é o uso da diagnose visual, pois, por meio da coloração da planta, é verificado se existe ou não tal deficiência. No entanto, essa técnica possui sua precisão limitada à experiência do técnico e, por sua vez, é muito trabalhosa (BAESSO, 2005).

Muitas pesquisas utilizam a análise de clorofila para realizar a investigação do nível de Nitrogênio da folha, verificando se existe ou não deficiência (BAESSO, 2005; BONFIM-SILVA; MONTEIRO, 2010). Esses níveis de clorofila indicam o teor de pigmento

presente na folha, ou seja, os níveis de verde. Na tabela 1, está descrito as alterações provocadas pela deficiência de tal elemento (COSTA et al., 2006).

Devido à relação existente entre concentração de N total e concentração de clorofila nas folhas (GIRARDIN et al., 1985), esse atributo tem sido utilizado para avaliar o estado nutricional das plantas com relação ao Nitrogênio. Verifica-se, assim, a possibilidade do uso dos níveis de pigmentação verde ou clorofila para a realização da análise do *status* de Nitrogênio no solo através da folha.

Mesmo sabendo que, com a adubação nitrogenada, é obtida melhoria na qualidade e produção de forrageiras, muitos pecuaristas ainda não têm a visão da necessidade desse tipo de correção. Com isso, não conseguem atingir valores ideais de produção.

Tabela 1 – Alterações provocadas nas plantas por deficiência de Nitrogênio

Visíveis	Folhas amareladas, inicialmente nas mais velhas, como resultado da proteólise. Ângulo agudo entre caule e folhas. Dormência de gemas laterais. Redução do perfilhamento. Senescência precoce. Folhas menores devido ao menor número de células. Coloração Amarelada em forma de “V” na extremidade da folha.
Químicos	Baixo teor de clorofila. Produção de outros pigmentos algumas vezes.
Crescimento	Em geral diminuído, com possíveis aumento das raízes em alguns casos.
Citológicos	Pequenos núcleos. Cloroplastos pequenos.
Metabólicos	Redução na síntese de proteínas. Alto conteúdo de açúcar e alta pressão osmótica.

Fonte: (MALAVOLTA, 2006; MARCHESIN et al., 2007).

3.3.2 Fósforo

O Fósforo (P) é essencial para o crescimento das plantas e nenhum outro nutriente pode substituí-lo. A maioria das culturas tem dificuldade na obtenção de Fósforo, com isso ele pode ser um limitante para essa produção, bem mais do que qualquer outra deficiência (IPNI, 1998).

Inicialmente os sintomas aparecem nas folhas mais velhas, seguindo para as mais novas. A cor das folhas mais velhas tende ao verde escuro até tons de roxo nas pontas e margens, coloração pardo-avermelhada das bordas das folhas como demonstrado na tabela 2 (MARCHESIN et al., 2007).

Tabela 2 – Alterações provocadas nas plantas por deficiência de Fósforo.

Visíveis	Cor amarelada das folhas, a princípio as mais velhas, pouco brilho, cor verde-azulada ou manchas pardas. Ângulos foliares mais estreitos. Menor perfilhamento. Gemas laterais dormentes. Número reduzido de frutos e sementes. Atraso no florescimento.
Químicos	Aumento de pigmentos vermelhos ou roxos em algumas espécies. Aumento no conteúdo de carboidratos. Aumento na relação P orgânico / P inorgânico.
Anatômicos	Restrição na diferenciação dos caules.

Fonte: (MALAVOLTA, 2006; MARCHESIN et al., 2007).

3.3.3 Potássio

O Potássio, ao contrário do Nitrogênio e Fósforo, não está ligado a formação de compostos orgânicos, mas ao metabolismo da planta como, por exemplo, na fotossíntese e outros processos (IPNI, 1998; MATTOS; MONTEIRO, 2018).

Quando seu teor é deficiente afeta a fotossíntese e a respiração da planta aumenta. Sendo assim, diminuem o suprimento de carboidratos. A deficiência de Potássio afeta o crescimento, pois, ele é responsável pela decomposição dos carboidratos que fornecem energia para a planta (IPNI, 1998).

A deficiência de Potássio se apresenta de várias formas como, para exemplificar, os sintomas visuais. Esses sintomas apresentam, normalmente, murcha ou queima ao longo da borda ou margem das folhas. Tais características são visíveis primeiro nas folhas mais velhas, principalmente nas gramíneas (IPNI, 1998). Além disso, as plantas apresentam um crescimento mais lento e pouca resistência a doenças, descrito na tabela 3.

Tabela 3 – Alterações provocadas nas plantas por deficiência de Potássio.

Visíveis	Clorose e depois necrose das margens e pontas das folhas, inicialmente das mais velhas. Internódios mais curtos em plantas anuais. Diminuição da dominância apical. Menor tamanho de frutos. Deficiência de ferro induzida (acúmulo de ferro nos nós inferiores).
Químicos	Aumento nas frações de Nitrogênio alpha amínico e amídico. Alto conteúdo de ácidos orgânicos. Menor teor de açúcar e amido em órgãos de reserva.
Anatômicos	Diferenciação prejudicada dos tecidos condutores. Perda de atividade cambial.

Fonte: (MALAVOLTA, 2006).

3.4 Diagnose Foliar

O estado nutricional das plantas pode ser determinado por meio de procedimentos indiretos e diretos. Os indiretos são aqueles em que a concentração de certo nutriente na planta é estimada por características correlacionadas com as concentrações do nutriente na planta e os diretos são realizados determinando as concentrações dos nutrientes na matéria seca do tecido vegetal. A análise da massa seca das folhas por meio de procedimentos químicos é conhecida como análise foliar (FONTES, 2001; MALAVOLTA, 2006).

A folha é o órgão no qual os distúrbios fisiológicos se apresentam mais evidentes. Por isso, a folha é o local em que se realizam quase todos os diagnósticos nutricionais das plantas (MOTTA, 2016). “Folha” substitui “planta” por ser o órgão, pelo qual reflete melhor o estado nutricional da cultura (MALAVOLTA, 2006).

A premissa da diagnose foliar é a relação entre o suprimento de nutrientes e os índices dos elementos nas folhas. Sendo assim, conforme os níveis presentes, as características se alteram.

Existem fatores que podem afetar os níveis dos nutrientes nas folhas, como a idade e a área a ser analisada. Os tecidos mais novos têm a tendência de uma maior concentração de nutrientes. Os macronutrientes estudados são móveis, isso significa que há uma movimentação desses nutrientes das folhas mais velhas para as mais jovens. Com isso há maior concentração de nutrientes nas mais jovens e as características de deficiência nutricional se apresentam primeiro nas folhas mais velhas (MALAVOLTA, 2006; MOTTA,

2016). Com isso, pode-se observar que uma diagnose foliar seria mais recomendada a sua realização nas folhas mais velhas, no caso dos nutrientes N, P e K.

Essa diagnose foliar é feita através da experiência de quem avalia seu estado. Logo, será realizada por um modelo de rede neural convolucional (CNN). Dessa forma, essa rede realizará a avaliação de qual deficiência existe.

De acordo com Martines (1999), para a aplicação da diagnose foliar deve-se cumprir três etapas sequenciais (MARTINEZ et al., 1999):

1. normatização da amostragem, preparo das amostras e análise química do material vegetal;
2. obtenção dos padrões de referência comparativos; e
3. interpretação dos resultados obtidos.

Para a criação das amostras e padrões de referência foi realizada a produção de mudas de *Brachiaria brizantha* em casa de vegetação. Essa produção passou por processos de indução de deficiência de nutrientes para realizar a coleta das características dos sintomas, as quais são extraídas por meio de imagens da cultura realizada por câmera digital.

3.5 Processamento de Imagem

O processamento de imagens digitais possui duas áreas principais de aplicação, melhoria de informação visual e processamento de dados de cenas para utilização em sistemas inteligentes. Hoje, essas técnicas de processamento de imagens estão sendo utilizadas em inúmeras áreas como, por exemplo, na área médica, na análise de imagens de satélite, na agricultura de precisão etc.

A habilitação de uma imagem para ser analisada em um sistema de inteligência artificial tem a necessidade da realização de algum tipo de pré-processamento, assim, possibilitando o destaque ou a extração das características importantes para a análise desejada. Para isso, essa imagem necessita de um processamento capaz de evidenciar suas características e, do mesmo modo, ser extraída mais facilmente e com uma qualidade aceitável.

Na área de processamento de imagens existe uma grande quantidade de técnicas e algoritmos utilizados para esse fim, dependendo, apenas, do tipo de imagem, sua qualidade e quais informações pretendem-se extrair delas (GONZALEZ; WOODS, 2010).

As redes neurais convolucionais (CNNs), são projetadas para reconhecimento de padrões diretamente de imagens com um pré-processamento mínimo. Nelas não são necessários realizar grandes processamentos nas imagens, como é o caso de outras tecnologias aplicadas a este fim.

3.6 Aprendizado de Máquina

Aprendizado de máquina é o ramo da inteligência artificial responsável por fazer máquinas executarem funções a partir de experiências. Utilizar o aprendizado envolve a generalização das informações coletadas e o seu desempenho não deve ser apenas na repetição de problemas idênticos, mas na resolução de problemas semelhantes no domínio (LUGER, 2013).

A ideia central, por trás do aprendizado de máquina, é poder representar o mundo real usando uma função matemática, desconhecida, antecipadamente, pelo algoritmo antecipadamente. Todavia, prevendo, depois de ver alguns dados (MUELLER; MASSARON, 2019). Essas funções podem ser modificadas utilizando parâmetros e pesos extraídos dos dados apresentados a esses algoritmos.

Ao mesmo tempo, podem ser amplamente definidos como métodos computacionais que usam o conhecimento para melhorar o desempenho e fazer previsões mais assertivas (MOHRI et al., 2018). O conhecimento refere-se a dados ou informações eletrônicas coletadas, catalogadas e disponibilizadas para análise. A qualidade e quantidade dessas informações são de suma importância para o sucesso de sistemas baseados em aprendizado de máquina. As técnicas desse aprendizado são métodos orientados a dados que combinam conceitos de computação, probabilidade e estatística e otimização (MOHRI et al., 2018).

O aprendizado de máquina não se limita apenas a prever o rótulo de saída de algum documento, mas sim, em muitas outras aplicações como, por exemplo, classificação de texto ou documento, processamento de fala, biologia computacional, visão computacional, entre muitos outros problemas.

Dentre esses exemplos, existe a visão computacional, considerada como uma das áreas mais promissoras em aprendizado de máquina, podendo resolver inúmeros problemas como reconhecimento de objetos, identificação de objetos, detecção de rosto, reconhecimento de caracteres, recuperação de imagens, detecção de câncer, detecção de doenças em plantas e animais.

3.6.1 Componentes do Aprendizado de Máquina

Para realizar o aprendizado de máquina são necessários alguns componentes, para assim chegar ao sucesso dos algoritmos de aprendizado.

- Exemplos: itens ou instâncias de dados a serem usados para o aprendizado ou avaliação do algoritmo. Um exemplo é a imagem de uma planta doente utilizada para o aprendizado e testes do sistema.
- *Features* (Características): conjunto de atributos representados por um vetor e associada a um exemplo. No caso de folhas de plantas, podem ser caracterizados pelas cores das folhas, tamanho, formato, pixels etc.
- *Labels*: são os valores ou categorias atribuídas aos exemplos. No caso de deficiências nutricionais de folhas, suas deficiências podem ser descritas como categorias.
- *Hyperparameters* (Hiperparâmetros): são parâmetros não determinados pelo algoritmo de aprendizado, mas especificados como entradas desse algoritmo.
- Exemplos de treinamento: exemplos utilizados para o treinamento do modelo de aprendizado. Um conjunto de imagens de folhas deficientes e saudáveis podem ser usadas como exemplos de treinamento.
- Exemplos de validação: exemplos utilizados para uma avaliação imparcial do modelo de aprendizado enquanto ajusta os hiperparâmetros. Embora o modelo não aprenda com esses dados, é indiretamente afetado por eles, devido à sua participação no processo de decisão das configurações dos hiperparâmetros. Após executar diferentes configurações dos hiperparâmetros, com base no desempenho do modelo testado no conjunto de dados de validação, um modelo vencedor é selecionado. Um conjunto de imagens de folhas deficientes e saudáveis podem ser usadas como exemplo de validação, desde que não estejam contidas nos dados de treinamento (SALEH, 2018) (MOHRI et al., 2018).
- Exemplos de teste: são usados para realizar a avaliação final do treinamento. Dessa forma, com eles, pode-se avaliar o desempenho do modelo de aprendizado. Isso permite validar o desempenho do modelo, na realidade, para futuras previsões. Um conjunto de imagens de folhas deficientes e saudáveis pode ser usado como exemplo de validação, desde que não esteja contido nos dados de treinamento e validação.
- *Loss function*: é a função usada para calcular a diferença, ou perda, entre o valor do rótulo previsto com o rótulo verdadeiro.

Para realizar o aprendizado de máquina necessita-se de uma determinada coleção de dados rotulados. Esses dados devem ser divididos aleatoriamente em uma amostra de treinamento, validação e teste. Essa divisão vai depender de cada aplicação ou problema a resolver, por exemplo, o número de dados de validação pode depender do número de hiperparâmetros que é necessário aprender (MOHRI et al., 2018). Um exemplo desse tipo de divisão pode ser visualizado na figura 1. Os dados de treinamento devem representar bem o domínio do problema, pois, se isso não se concretizar pode-se ter resultados que não serão úteis.

Durante o processo de treinamento, analisa-se o desempenho do modelo e faz-se alterações nos parâmetros, conforme necessário, para que se obtenha um melhor resultado. Esse processo utiliza os dados de entrada para criar uma função matemática, que nada mais é, que um algoritmo que descobre como adaptar a função aos dados a ela apresentados (MUELLER; MASSARON, 2019).

Juntamente com o treinamento é realizada a validação que seria uma divisão dos dados para realizar testes no sistema. Após cada fase do treinamento é realizada a validação para verificar o desempenho do modelo. Após encontrar qual alcançou o melhor resultado, utilizando os dados de validação é realizado o teste com dados nunca vistos por ele, para verificar se realmente funcionará com um conjunto de dados novos.

Figura 1 – Divisão do dataset para aprendizado de máquina.



Fonte: autor da tese.

A quantidade de dados no dataset também pode influenciar nessa divisão, quando a amostra rotulada é muito pequena, geralmente a quantidade de dados de treinamento é maior do que os dados de validação e teste, pois o aprendizado da rede depende diretamente desses dados para ser bem-sucedido.

Para funcionar de forma satisfatória um modelo de aprendizado de máquina deve representar uma visão geral dos dados fornecidos. Se ele não conseguir extrair características suficientes, para poder identificar os dados de forma eficiente, seu treinamento será ruim e terá o *underfitting*, ou seja, aprendizado insuficiente.

Caso o modelo extraia características em excesso, nesse caso, ele terá o *overfitting*, ou seja, aprendeu em excesso. Logo, não terá um resultado satisfatório.

É possível verificar a necessidade de treinar ou ensinar o modelo de maneira equilibrada, durante a etapa de aprendizagem. Em suma, para que nem aprenda pouco os dados e nem aprenda demais, pois os dois extremos levam a resultados ruins. Tudo isso ocorre, pois, assim ele não se torna generalizável o suficiente para produzir dados úteis para a classificação do sistema.

O desempenho da rede é verificado utilizando a função de perda associada aos dados de erro de teste e não no erro nos dados de treinamento.

A aprendizagem de máquina se dividiu em três grandes áreas: o aprendizado supervisionado, o aprendizado não supervisionado e o aprendizado por reforço (RUSSELL; NORVIG, 2013). Segue uma breve descrição de cada uma delas:

- **Aprendizado supervisionado:** possui um conjunto de dados com inúmeras amostras de entrada com suas respectivas saídas desejadas, ou seja, cada amostra de treinamento possui suas entradas com suas respectivas saídas (Ex.: imagem folha de pastagem saudável rotulada como “saudável”). Essa prática permite generalizar o conceito aprendido, utilizando o conjunto de dados para realizar novas previsões em novos dados não etiquetados.
- **Aprendizado não supervisionado:** o seu conjunto de dados não possui etiquetas. Dessa forma a máquina deve descobrir sozinha os padrões dos dados. As saídas desejadas não existem.
- **Aprendizado por reforço:** ele avalia constantemente a defasagem de valor entre a resposta produzida em relação a saída desejada, utilizando um mecanismo de acúmulo de recompensa. As tomadas de decisões feitas pela máquina são recompensadas por decisões corretas e penalizadas por decisões erradas.

Dentre os tipos de aprendizado de máquina citados o supervisionado é o mais popular entre os três, bem como será utilizado, neste estudo.

3.7 Redes Neurais Artificiais

As redes neurais artificiais são modelos computacionais baseados na estrutura do cérebro biológico, que apesar de não possuir programa, controla todas as ações do indivíduo (RUSSELL; NORVIG, 2013).

O modelo de neurônio artificial mais simples, e utilizado como ponto de partida para os estudos de redes neurais, foi proposto por McCulloch e Pitts, sendo ainda o mais utilizado nas arquiteturas atuais (MCCULLOCH; PITTS, 1943; SILVA et al., 2010).

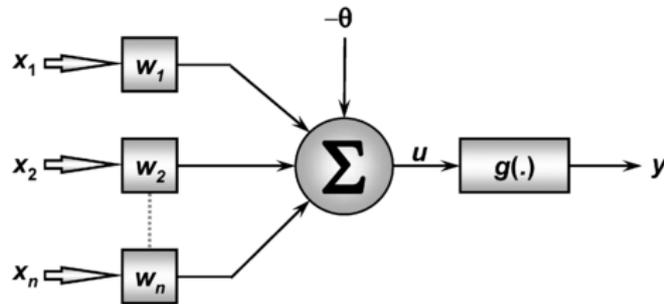
Um dos primeiros modelos de redes neurais artificiais foi proposto por (ROSENBLATT, 1958), que recebeu o nome de *perceptron* (figura 2) e teve como base o neurônio de McCulloch.

As redes neurais são modelos matemáticos que recebem informações de entrada e geram valores de saída (SILVA et al., 2010). Elas não são programas fixos, mas modelos a processarem informações de entrada, resultando dados de saída.

A base das Redes neurais é o neurônio artificial e consiste em:

- Sinais de Entrada $\{X_1, X_2, \dots, X_n\}$: são sinais ou informações vindas de dados externos e representam valores de alguma aplicação específica (Ex.: pixels das imagens de folhas).
- Conjunto de pesos sinápticos $\{W_1, W_2, \dots, W_n\}$: descrevem as forças das conexões e servem para ponderar os sinais de entrada da rede.
- Combinador Linear $\{\odot\}$: é o nível de ativação agregador de todos os sinais de entrada ponderados com todos os pesos sinápticos. Dessa maneira, produzindo um valor do potencial de ativação.
- Limiar de Ativação $\{\cup\}$: especifica o valor de um patamar apropriado para que o resultado produzido pelo combinador linear esteja em um valor abaixo ou acima de um valor de limiar.
- Potencial de Ativação $\{u\}$: é o valor produzido entre o combinador linear e o limiar de ativação. Se o valor for $u \geq 0$, o neurônio produz potencial excitatório, caso contrário produz potencial inibitório.
- Função de Ativação $\{g\}$: é utilizado para limitar a saída de um neurônio entre um intervalo de valores.
- Sinal de Saída $\{y\}$: é o valor final produzido por um neurônio, podendo ser utilizado como entrada de outros neurônios que estão sequencialmente interligados (SILVA et al., 2010).

Figura 2 – Neurônio Artificial.



Fonte: (SILVA et al., 2010)

O funcionamento de um neurônio se resume em apresentar um conjunto de dados à camada de entrada do neurônio, multiplicar essas entradas pelo seu respectivo peso sináptico e realizar a soma ponderada dos sinais de entrada, subtraindo do limiar de ativação e obtendo o potencial de ativação. Em seguida, aplica-se a função de ativação, assim, limitando a saída do neurônio. Por fim, é aplicada a função de ativação em semelhança ao seu potencial de ativação e realizada a compilação da saída (ROSENBLATT, 1958; SILVA et al., 2010).

Uma variação do modelo *perceptron* foi a criação da *Multi Layer Perceptron* (MLP), sendo um dos modelos mais utilizados até hoje para classificação.

3.8 Redes Neurais Artificiais Multicamadas

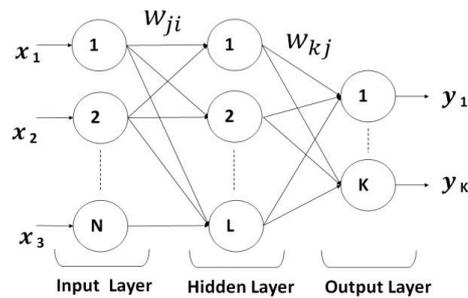
Um sistema de visão artificial ou visão computacional pode ser definido como um conjunto de métodos e técnicas, utilizando sistemas computacionais capazes de interpretar uma imagem de forma automática ou semiautomática (SAHA; UDUPA, 2001).

Redes neurais artificiais do tipo Perceptron Multicamadas são modelos computacionais inspirados no modelo biológico de interligação entre neurônios. Além disso, procuram realizar o aprendizado de máquina. As redes neurais multicamadas (MLPs) são redes que utilizam o aprendizado supervisionado. Para que isso seja possível, para qualquer classificação, é necessário um processo de treinamento no qual vários exemplos de dados pré-classificados são apresentados (SILVA et al., 2010) (MIRANDA, 2011).

As MLPs são caracterizadas por existir pelo menos 1 camada escondida (oculta) (SILVA et al., 2010), encontrada entre a camada de entrada e de saída como demonstrado na figura 3. Elas são utilizadas para muitas aplicações como, por exemplo, classificação, reconhecimento de imagens e padrões. O grande diferencial delas é a capacidade de resolver

problemas não-lineares. Com isso, atendendo a um número muito grande de aplicações encontradas na atualidade, pois a maioria dos problemas não podem ser solucionados por uma fronteira linear simples entre duas classes.

Figura 3 – Rede Neural Multicamada.



Fonte: (MEHMET et al., 2016).

Como dito anteriormente, para a MLP funcionar é preciso treiná-la. O treinamento consiste em modificar ou ajustar os dados dos pesos dos neurônios para que o aprendizado seja realizado. Esse ajuste de pesos é realizado por algoritmos de aprendizado. Vale ressaltar a utilização da função de gradiente descendente como um dos principais.

Para realizar esse treinamento é necessário um conjunto de amostras que possua dados de entrada com suas respectivas informações de saída desejada. A saída é comparada com a entrada e o erro gerado é propagado camada a camada, ajustando os pesos sinápticos a fim de minimizar os erros. Essa regra de aprendizado chamada como “regra da retropropagação” foi proposta por Rumelhart (RUMELHART et al., 1986).

Os principais parâmetros a serem especificados para o treinamento são o critério de parada, a taxa de aprendizagem e o número de épocas (LUGER, 2013).

Existe também a função de ativação, responsável por limitar a saída de um neurônio dentro de um intervalo de valores e, para isso, as MLPs utilizam funções de ativação totalmente diferenciáveis (SILVA et al., 2010).

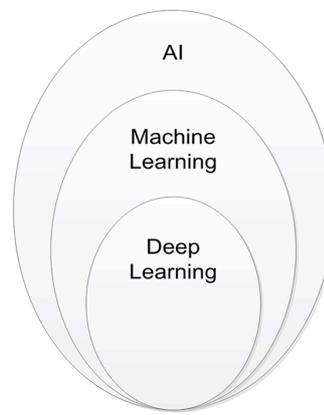
3.9 Deep Learning

O aprendizado profundo é apenas um subconjunto da Inteligência artificial. No entanto, possui grande importância e pode ser aplicado a várias tarefas. O *Deep Learning*

pode ser entendido como uma técnica de aprendizado de máquina que utiliza muitas camadas de informações não-lineares.

Os algoritmos de aprendizado profundo aprendem, geralmente, analisando grandes quantidades de dados, mas também podem aprender utilizando pequenas quantidades (MUELLER; MASSARON, 2019). O *Deep Learning* faz parte da IA e do aprendizado de máquina como mostra a figura 4.

Figura 4 – Representação da relação entre o campo da Inteligência Artificial e seus sistemas especialistas Machine Learning e Deep Learning.



Fonte: MUELLER; MASSARON, 2019

O aprendizado profundo imita a funcionalidade do cérebro humano, processando dados, tal como usando unidades de computação chamados de neurônios, organizados em seções ordenadas. A técnica da aprendizagem profunda é baseada no conceito de redes neurais artificiais (MUELLER; MASSARON, 2019).

Dependendo da rede neural e do número de camadas que ela possuir, deverá ser classificada como Rede Neural Profunda, podendo formar grandes redes e únicas capazes de realizar um aprendizado especializado, utilizando camadas de neurônios matemáticos para realizar essa tarefa.

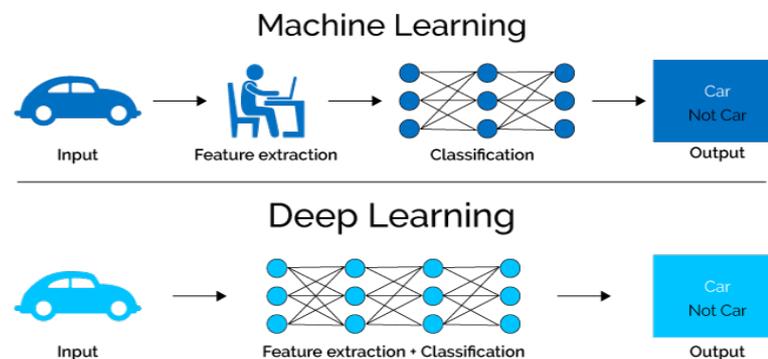
Tornou-se possível a utilização de *Deep Learning*, devido ao surgimento de computadores mais rápidos que passaram a utilizar GPUs (unidades de processamento gráficos). Com essas placas foi possível um incremento no poder de processamento em mais de 1000 vezes em 10 anos.

Em decorrência das características de implementação das MLP's, utilizando *Deep Learning*, seu treinamento pode ser muito complexo e demorado. Tudo isso depende da quantidade e características dos dados de entrada. Na maioria das vezes, é muito complexo e de grande volume.

Para simplificar essas redes e diminuir o tamanho e complexidade dos dados a serem treinados foi desenvolvido a Rede Neural Convolutacional (*Convolutional Neural Network* - CNN).

Pode-se afirmar, como sendo uma das grandes diferenças entre *Machine Learning* e *Deep Learning*, a forma como cada um deles extrai as características dos dados. No primeiro caso, essas características são extraídas por seres humanos. Enquanto, no segundo caso, essas características são extraídas pelo próprio modelo, de forma automática através do aprendizado. Na figura 5 é ilustrada essa diferença.

Figura 5 – Diferença entre aprendizado de máquina tradicional e aprendizado profundo.



Fonte: (AIMAGNIFI, 2017)

3.10 Redes Neurais Convolucionais

Deep Learning é um dos principais métodos de aprendizado de máquina utilizados para visão computacional, principalmente em classificação de imagens. Com a máquina podem-se aprender atributos da imagem automaticamente por meio de exemplos em um conjunto de imagens de treinamento (ZEILER; FERGUS, 2014). Dentro do aprendizado profundo, existem as redes neurais convolucionais, tornando-se populares para a classificação de imagens.

Essas redes utilizam características da visão biológica para realizar as classificações, desse modo, possuindo características diferentes de outros modelos de redes neurais. As redes neurais convolucionais são inspiradas pelo córtex visual humano ou biológico, contendo regiões muito pequenas das células. Elas são altamente sensíveis a certas áreas do campo visual humano (HAYKIN, 2003). Como nos neurônios biológicos nas CNNs, existem hierarquia nas camadas, sendo que as de nível inferior possuem a tarefa de

detectar padrões visuais mais simples, como segmentos de linha e bordas. À medida que a rede vai se aprofundando, são detectados padrões mais específicos (RUSSELL; NORVIG, 2013).

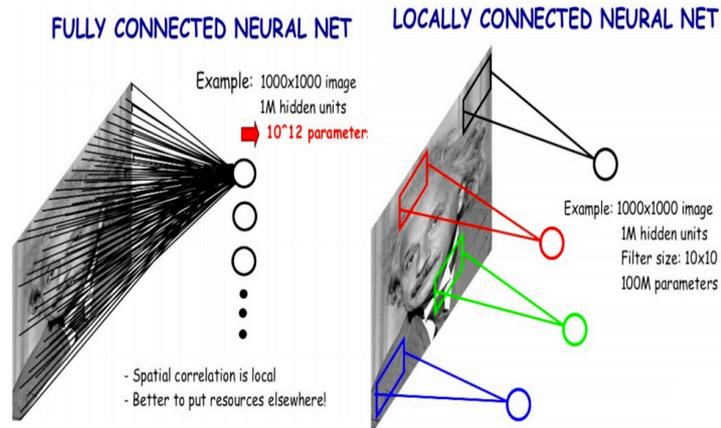
Outro diferencial é que o fato de conseguirem levar em conta a estrutura espacial da imagem. Dessa forma, permitem aos neurônios lidar com entradas semelhantes em diferentes locais espaciais de uma determinada imagem (ZEILER; FERGUS, 2014). Isso possibilita mostrar como as ativações dos neurônios eram espacialmente invariantes, significando que elas não eram dependentes da localização espacial dos padrões dados (PURKAIT, 2019). Em suma, a estrutura da informação de entrada é composta por até três dimensões. Normalmente duas dimensões para o nível de cinza e três dimensões para as imagens coloridas. Para a terceira dimensão, tem-se o canal de cores.

As CNNs estão sendo muito utilizadas em várias áreas da visão computacional como detecção de objetos, classificação de imagens, segmentação, reconhecimento de faces, sistemas de carro autônomo e outras aplicações. Essas tecnologias só foram possíveis, devido aos grandes avanços das tecnologias computacionais existentes hoje.

Nas redes neurais multicamadas todos os nós se conectam a todos os nós da próxima camada. Isso tem uma grande probabilidade de consumir muito tempo de processamento e levar a rede a ter um *overfitting* e também consumir muitos recursos computacionais. A CNN veio para corrigir este problema, pois não possui uma conexão para cada nó, assim realizando uma filtragem seletiva como ilustrado na figura 6 (SINGH, 2019).

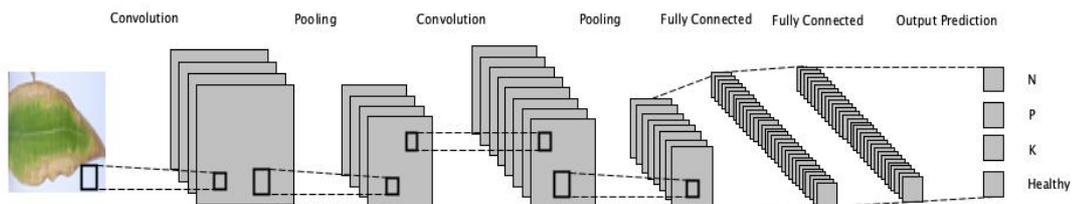
Além das camadas normais de redes neurais, existem as camadas extras que são chamadas de convolucionais. Nessas redes, existe uma divisão em duas etapas. A primeira etapa chamada de aprendizado de atributos ou características (*feature learning*) e a segunda, chamada de classificação (*classification*). Na etapa 1, o modelo irá aprender os detalhes das imagens, ou seja, suas características. Esse procedimento vai gerar um resultado utilizado para alimentar a segunda etapa da rede, essa sendo responsável por realizar a classificação. Na primeira etapa, basicamente, o modelo possui camadas convolucionais e de *downsampling* (subamostragem). Na segunda, camadas de uma rede totalmente conectada ou densa, como ilustrada na figura 7.

Figura 6 – Diferença entre uma camada tradicional totalmente conectada e uma camada convolucional.



Fonte:(ZAFAR et al., 2018a).

Figura 7 – Composição básica de um modelo de rede neural convolucional.



Fonte: autor da tese.

As CNNs são muito boas para extrair padrões locais mais específicos, ao contrário as redes totalmente conectadas, fracas para reconhecer padrões locais a partir de dados, mas é eficiente em capturar padrões globais envolvendo todos os pixels da imagem (PURKAIT, 2019).

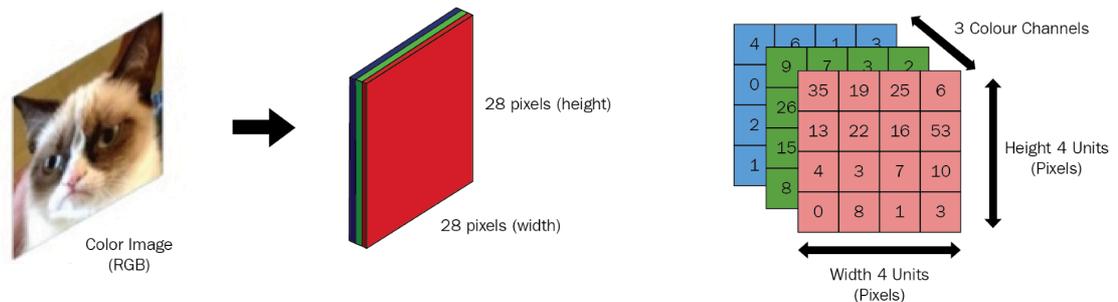
Como dito anteriormente, diferentemente das redes totalmente conectadas, nas CNNs consegue-se preservar a estrutura espacial da imagem. Essa rede será alimentada por um tensor de dimensão n , com isso cada pixel estará em sua posição original da matriz. Dessa forma, não será reduzida a um vetor único.

A principal vantagem da CNN em comparação a outros modelos. Nela é possível realizar a detecção automática dos recursos necessários sem nenhuma supervisão humana. Dadas as imagens, consegue extrair as características necessárias para identificar tal imagem, ou seja, ela aprende por si só. A CNN também é computacionalmente mais eficiente, pois utiliza funções especiais de convolução, *pool* e executa compartilhamento de parâmetros (PURKAIT, 2019).

A rede neural convolucional tem como entrada imagens “ $n \times n \times r$ ”. Para exemplificar, “ n ” representa a altura e a largura da imagem e “ r ” representam os canais de cores presentes na imagem. Por exemplo, uma imagem utilizada como entrada em uma CNN, poderá possuir o formato de $244 \times 244 \times 3$, sendo que 244 pixels de altura e largura e 3 canais de cores (RGB) como ilustrado na figura 8. É importante entender que uma imagem vai entrar na rede como um tensor tridimensional, preservando sua estrutura espacial.

Essas redes utilizam três ideias básicas, a seguir elencadas: a utilização de filtros receptivos locais, compartilhamento de pesos e *pooling*.

Figura 8 – Representação de uma imagen RGB para uma CNN.



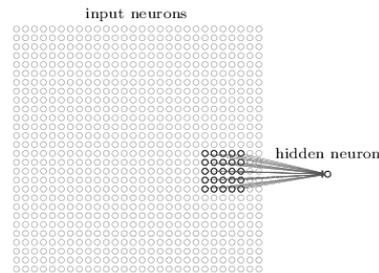
Fonte: (PURKAIT, 2019)

3.10.1 Campos Receptivos Locais

Diferentemente de redes totalmente conectadas, nas quais são conectados todos os pixels das imagens de entrada a todos os neurônios ocultos, na CNN realiza-se apenas conexões em pequenas regiões e em áreas localizadas na imagem (HAYKIN, 2003).

Na realidade, será conectado cada neurônio da primeira camada oculta a uma pequena região dos neurônios de entrada, por exemplo, uma região 5×5 , possuindo 25 pixels como entrada. Como pode ser ilustrado na figura 9.

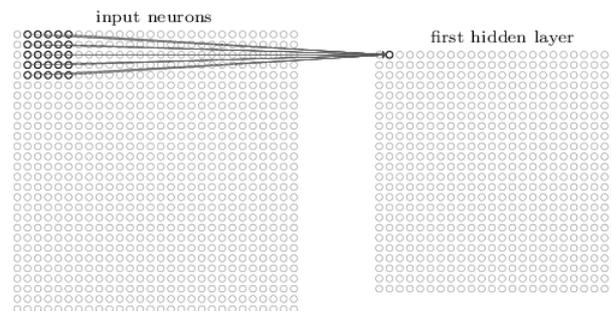
Figura 9 – Campo receptivo local neurônio de entrada.



Fonte: (ZAFAR et al., 2018a).

Essa pequena janela é chamada de campo receptivo local para o neurônio oculto. Para cada campo receptivo existe um neurônio oculto diferente na primeira camada oculta. A janela será deslizada por toda a imagem gerando a primeira camada oculta, como demonstrado na figura 10.

Figura 10 – Deslizando a janela sobre a imagem



hFonte: (ZAFAR et al., 2018a).

Essa movimentação do campo receptivo local pode ser feita com um pixel por vez, mas também com valores de passos maiores. Tudo isso vai depender do projeto e, logicamente, será definido pelos desenvolvedores, podendo experimentar valores maiores que 1. Vale ressaltar que é um dos hiperparâmetros de uma rede neural convolucional chamada de *stride length*.

Cada neurônio oculto tem um bias e pesos conectados ao seu campo receptivo local. Esses valores serão compartilhados com todos os neurônios daquela camada oculta realizando assim um compartilhamento de parâmetros diminuindo muito os dados a serem aprendidos em consonância com as redes neurais totalmente conectadas. Isso significa que

todos os neurônios de uma camada oculta, compartilham os mesmos pesos e bias. Assim, detectam os mesmos recursos, porém em lugares diferentes da imagem.

3.10.2 Camadas Convolucionais

As redes neurais convolucionais geralmente possuem muitas camadas, mas entre elas o papel central, nesse tipo de modelo, é a camada convolucional. Ela é a principal camada e faz a maior parte do processamento computacional pesado. A convolução é uma operação linear usada para mesclar mais conjuntos de informações e é aplicada aos dados de entrada da rede. Portanto, são aplicados filtros de convolução, gerando como resultado um mapa de características. A convolução é muito utilizada nas áreas de processamento de sinais e imagens (GONZALEZ; WOODS, 2010), e sua definição é feita pela equação 1.

$$y = x * w \rightarrow y[i] = \sum_{k=0}^{k=m-1} x^p[i + m - k]w[k] \quad (1)$$

As convoluções bidimensionais são utilizadas para implementar filtros de imagens (*kernel* ou detectores de recursos) aplicadas a toda área da mesma imagem e são utilizados para encontrar uma área específica na imagem ou alguma característica, utilizando transformações lineares. As camadas convolucionais filtram um tensor de entrada de uma imagem com uma pequena janela chamada de *kernel*. Ele define exatamente o que a convolução vai filtrar e extrair de características.

Pode-se presumir quais os parâmetros que uma camada de convolução deve aprender, neste caso, são os pesos do *kernel* da camada. Durante o treinamento da CNN, os valores desses filtros são ajustados de uma forma automática para extrair as características úteis da tarefa a ser executada (ZAFAR et al., 2018a). Esse *kernel* pode ser demonstrado na figura 11.

Figura 11 – Exemplo de operação de convolução.



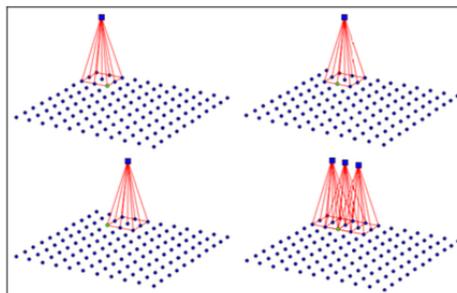
Fonte: autor tese.

Desta forma, a composição de uma CNN possui muitos *kernel's*, comportando-se como um neurônio. Ele pode formar uma ou mais camadas com um ou mais neurônios ou, nesse caso, *kernel's*.

Ao deslizar os *kernel* sobre a imagem (figura 12) os seus pesos não mudam. Eles são utilizados para calcular a ativação de todos os neurônios de saída. Isso é feito para realizar o compartilhamento de parâmetros, levando a CNN a ter um número muito menor de parâmetros a serem treinados e isso é feito por dois motivos:

- Reduzir o número de pesos, assim reduzindo o uso de recursos computacionais;
- Evitar ajustes excessivos (VASILEV et al., 2019).

Figura 12 – Filtro se movendo através da imagem calculando novos valores de ativação para os neurônios de saída.



Fonte: (VASILEV et al., 2019).

Os principais hiperparâmetros contidos em uma camada convolutacional utilizados para o controle de seu comportamento são:

- **Tamanho do kernel:** o kernel é a janela deslizante, passando sobre a imagem. Geralmente, essa janela é usada com tamanhos pequenos e seu valor é ímpar, como 1, 3, 5 e raramente 7.
- **Stride (s):** esse parâmetro é utilizado para definir o tamanho do passo do *kernel* responsável por deslizar sobre os pixels. Normalmente é definido em 1, mas pode ser definido com tamanho maiores.
- **Zero padding:** para conseguir deslizar a janela sobre toda a imagem e englobar todos os *pixels* sem perder os de borda, faz-se necessário o preenchimento dessas bordas com valores zero.
- **Número de filtros:** este hiperparâmetro irá controlar quantos filtros a camada de convolução terá e dependerá de quantas características ou recursos da imagem são preciso identificar (ZAFAR et al., 2018b).

Para ser aplicado a imagens coloridas é necessário considerar mais de um canal de cor. No caso de cores RGB são três cores: vermelho, verde e azul. Cada canal de cor seria uma fatia de profundidade, nas quais os valores são as intensidades de *pixel* para a cor especificada (R, G ou B).

Essa combinação de cores é chamada de volume de entrada com profundidade 3. Dessa forma, um filtro 3x3 seria aplicado a cada um dos canais de cores.

3.10.3 *Padding*

Algumas vezes, ao lidar com imagens, não é interessante que o resultado de uma operação de convolução tenha dimensões diferentes da imagem original. Nesse caso, o *padding* torna-se necessário. É importante utilizá-lo também, quando se quer realizar a convolução em todos os pixels da imagem, pois dependendo do tamanho da imagem e do tamanho do *kernel* será imprescindível realizar o ajuste com o *padding*. Para isso, é realizado o preenchimento com zeros (figura 13) nas bordas da imagens na parte superior, inferior, linhas e colunas para que, no final do processo de convolução, se mantenha o tamanho da imagem original ou se consiga agregar todos os pixels na convolução (MICHELUCCI, 2019).

Figura 13 – Exemplo de utilização de *padding*.

0	0	0	0	0	0
0	35	19	25	6	0
0	13	22	16	53	0
0	4	3	7	10	0
0	9	8	1	3	0
0	0	0	0	0	0

Fonte: autor tese.

3.10.4 Pooling

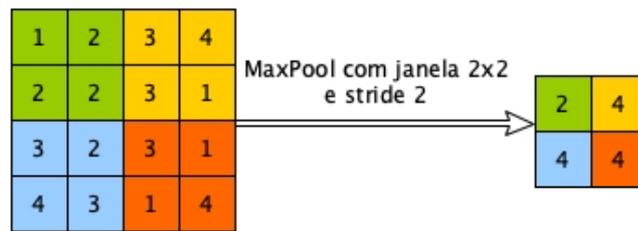
Além das camadas descritas anteriormente, esse modelo de rede possui camada de agrupamento ou *pooling*. Essas camadas são geralmente utilizadas logo depois das camadas convolucionais e simplificam as informações de saída da camada convolucional (HAYKIN, 2003).

Essa camada é utilizada para diminuir as dimensões espaciais dos tensores de ativação, mas não a profundidade do volume, sendo uma camada não paramétrica, ou seja, não possuidora de parâmetros (ZAFAR et al., 2018b).

Com esse tipo de camada obtém-se um ganho, pois é uma maneira simples e barata para resumir informações espacialmente em um tensor de entrada. Possui também uma grande vantagem de não possuir parâmetros a serem aprendidos, mas tem a desvantagem de talvez acabar deixando informações importantes de fora da rede.

Igual a camada de convolução, o *pooling* utiliza uma janela que desliza sobre a matriz de *pixels*, iniciando no canto superior esquerdo para o direito e de cima para baixo, em cada local por ela atingido.

Essa camada se difere da convolucional, pois possui operadores determinísticos que geralmente calculam o valor máximo (*max pooling*) ou médio (*average pooling*) dos elementos. O tipo mais comum de *pooling*, utilizado nos projetos de visão computacional, é o *max pooling* (SEWAK et al., 2018) como demonstrado na figura 14.

Figura 14 – *Max pooling* com tamanho 2x2.

Fonte: autor tese.

Na figura 14, com a configuração de janela e *stride* descritas, é obtido a redução pela metade da entrada, assim diminuindo o número de parâmetros e o tempo de treinamento.

Nas arquiteturas da CNN, o pool é normalmente realizado com janelas 2x2, passo 2 e sem preenchimento. Enquanto a convolução é feita com janelas 3x3, passo 1 e com preenchimento.

3.10.5 Função de ativação ReLU(*Rectified Linear Activation Function*)

A função de ativação é um elemento de extrema importância em redes neurais, basicamente, ela decide se um neurônio vai ser ativado ou não. Tal decisão vai dizer se essa informação é relevante ou pode ser ignorada (SILVA et al., 2010).

As funções de ativação fornecem a não linearidade ao modelo. Elas possibilitam a esses sistemas conseguirem aprender a realizar tarefas mais complexas, fazendo a transformação não-linear dos dados de entrada.

Existem diferentes tipos de funções de ativação (Ex.: sigmoid, tangente hiperbólica e ReLU), utilizadas em redes neurais. Sem essas funções, a rede seria um modelo linear facilmente descrito, utilizando multiplicação de matrizes.

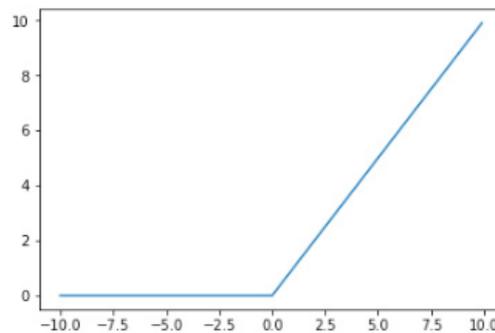
Nesta pesquisa, foi escolhida a função de ativação mais utilizada para projetos de redes neurais atualmente chamada de ReLU (KRIZHEVSKY et al., [S.d.]). Essa função é não-linear. Isso torna possível copiar os erros para trás e também ter várias camadas de neurônio ativados por ela.

A função leva vantagem sobre outras funções de ativação, pois não ativa todos os neurônios, ao mesmo tempo. Se a entrada for negativa, simplesmente ela será convertida para zero e o neurônio não será ativado como demonstra o gráfico da figura 15. Assim, apenas alguns neurônios serão ativados melhorando a eficiência, tornando a rede esparsa e

de fácil computação (GOODFELLOW et al., 2016). Nas redes neurais modernas, a recomendação é utilizar ReLU como padrão.

Também, faz-se necessário utilizar algum tipo de função de ativação para a camada de saída para que, de alguma forma, se tenha o resultado da avaliação executada pela rede neural. As funções de ativação Sigmoid e Softmax devem ser usadas para encontrar esse resultado em forma de uma probabilidade. A função Sigmoid é utilizada para problemas de classificação binária. Enquanto, a função Softmax pode ser usada para classificação binária ou multiclasse.

Figura 15 – Função de ativação RELU.



Fonte: autor da tese.

3.10.6 Função Softmax

A função de ativação Softmax transforma números em probabilidades e sua soma resulta 1. Ela gera um vetor que representa a distribuição de probabilidades de uma lista de possíveis resultados de saída. Ela é uma função muito utilizada em aprendizado profundo e, em aplicações de classificação, multiclasse. A classe a ser prevista por essa função é o item que tiver maior pontuação na lista gerada por ela.

Nessa função, é necessário aplicar uma exponencial para cada elemento da camada de saída e normalizar os valores, dividindo pelo somatório de todos os exponenciais, como ilustrado na figura 16.

Figura 16 – Funcionamento da função Softmax.

$$\begin{array}{ccc}
 \text{Camada de Saída} & & \text{Probabilidades} \\
 \begin{bmatrix} 1.3 \\ 5.1 \\ 2.2 \\ 0.7 \\ 1.1 \end{bmatrix} & \xrightarrow{\text{Função Softmax}} & \begin{bmatrix} 0.02 \\ 0.90 \\ 0.05 \\ 0.01 \\ 0.02 \end{bmatrix} \\
 & & \frac{\exp(X_i)}{\sum_j \exp(X_j)}
 \end{array}$$

Fonte: autor da tese.

3.10.7 Camadas Totalmente Conectadas (Densas)

Após as camadas de convolução e Pool, é realizada a etapa de classificação, utilizando algumas camadas totalmente conectadas. A função dessas é utilizar as características extraídas das camadas anteriores. Em resumo, realizar a classificação dos dados de entrada em várias classes baseada no conjunto de dados de treinamento.

Essas camadas são diferentes das camadas anteriores, pois possuem os pesos conectados a apenas uma determinada região. A camada totalmente conectada tem a característica de ser completamente conectada. Cada neurônio da camada anterior é conectado a cada um dos neurônios da próxima camada.

Diferentemente das camadas convolucionais, elas só aceitam dados em 1D, necessitando realizar uma conversão dos dados de 3D para 1D. Dessa forma, a entrada da camada totalmente conectada será um vetor de uma dimensão.

A camada totalmente conectada é uma rede MLP (Perceptron multicamadas). Nela, utiliza-se a função de ativação Softmax em sua saída. Há a transformação de um vetor de pontuação arbitrária de valores reais em um vetor de probabilidade entre valores de zero e um, e a soma de todos tem o valor 1.

3.10.8 Treinamento (Backpropagation)

As redes neurais convolucionais são treinadas, conforme as redes neurais artificiais multicamadas, utilizando o algoritmo de *backpropagation* (retropropagação com descida do gradiente). Esse é um algoritmo utilizado para encontrar os melhores pesos para mapear corretamente as entradas para saídas (LUGER, 2013).

Esse tipo de treinamento é realizado utilizando aplicações sucessivas de duas fases específicas, propagação adiante (*forward*) e propagação reversa (*backward*).

Na propagação adiante (*forward*), uma amostra do conjunto de dados de treinamento é apresentada a rede e são propagados para frente, camada a camada até gerar um resultado de saída. Os objetos são ponderados pelos pesos associados as suas conexões de entradas correspondentes. Em seguida, é aplicada uma função de ativação e a saída é produzida e utilizada como entrada para a próxima camada. Este processo é repetido até a camada de saída (LUGER, 2013).

Nessa fase, busca-se apenas obter os resultados de saída da rede, levando em consideração os valores atuais de pesos e limiares dos neurônios. Esses valores permanecerão inalterados durante essa fase.

Após esta etapa, o resultado de saída da rede é comparado com o valor desejado da amostra e será calculado o valor do erro da rede (SILVA et al., 2010).

Esse erro é calculado através de uma função de custo/perda, assim, é avaliado o desempenho do algoritmo de aprendizado. Essa função demonstra a capacidade do modelo em realizar as previsões para determinados valores de pesos. A função mais utilizada para realizar o aprendizado em *Deep Learning* multiclasse é a *Categorical Cross Entropy*, essa função é descrita pela equação (2).

$$Loss = -\sum_x p(x) \cdot \ln q(x) \quad (2)$$

Onde:

p = valor real;

q = Valor previsto.

O objetivo de qualquer algoritmo de aprendizado de máquina é realizar a minimização da função de custo, e para isso se realiza o treinamento. Diminuir a função de custo significa que, de alguma forma, o modelo está aprendendo sobre os dados de treino.

Com o valor do erro encontrado, é possível realizar o ajuste dos pesos da rede, assim realizando o seu aprendizado. Dessa maneira, possibilitando o reconhecimento de imagem. Para ajustar os pesos e bias, é realizada a propagação reversa (*backward*). Diferentemente da fase anterior, agora os seus valores são alterados e, dessa forma, a rede é treinada.

Para realizar essa otimização, é utilizado um algoritmo conhecido como descida do gradiente (*Gradient descent*). Esse algoritmo é utilizado para encontrar o mínimo para uma função de perda, como também para procurar qual a direção seguir, possibilitando alcançar esse mínimo da função e qual passo deve-se dar para chegar ao destino, ou melhor, ao erro mínimo.

Após encontrar o gradiente da função de perda na camada final, é necessário fazer o caminho reverso (*backward*). Nesse caso, é aplicado este gradiente na regra da cadeia de forma recursiva para atualizar os pesos na rede.

A regra da cadeia é utilizada para calcular funções compostas, e é matematicamente descrita como:

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y} \cdot \frac{\partial y}{\partial x} \quad (3)$$

Após o cálculo das derivadas é utilizada uma taxa de aprendizagem para forçar o peso a ser atualizado de forma suave e lenta da seguinte maneira:

$$\text{Novo Peso}(W_n) = \text{Peso Antigo}(W_a) - \text{Derivada} * \text{Taxa de Aprendizagem} \quad (4)$$

3.11 Redes VGGNet

O modelo de CNN VGGnet, foi inventado pelo VGG (Visual Geometry Group) da Universidade de Oxford.

A grande novidade nas redes VGG é a utilização de filtros menores. Enquanto outras redes como AlexNet e ZFNet utilizavam filtros 11x11 e 7x7, respectivamente, as redes VGG usam filtros 3x3. A vantagem em se utilizar filtros menores é a diminuição do número de parâmetros da rede, por exemplo:

- 1 camada de filtro 11x11 tem um total de $11 * 11 = 121$ parâmetros
- 5 camadas de filtros 3x3 tem um total de $3 * 3 * 5 = 45$ parâmetros;
- 1 camada de filtro 7x7 tem um total de $7 * 7 = 49$ parâmetros;
- 3 camadas de filtros 3x3 tem um total de $3 * 3 * 3 = 27$ parâmetros.

Pode-se usar 5 camadas de filtros 3x3 que irá cobrir a área de um filtro 11x11, mas utilizando menos parâmetros, exatamente 63% a menos. Da mesma forma, utilizando 3

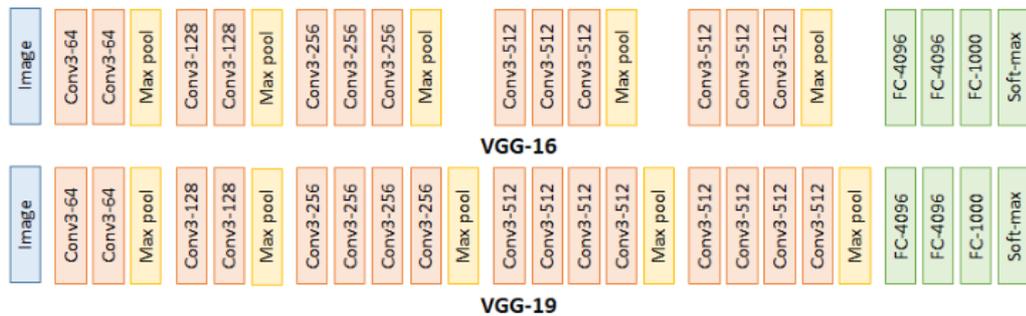
filtros 3x3, em contrapartida de filtros 7x7, tem-se uma economia de parâmetros, em torno de 45%.

Além da diminuição dos parâmetros, em cada uma das camadas, tem-se a introdução de mais funções de ativação ReLU, isso faz aumentar a não linearidade do modelo. Por exemplo, se utilizar 3 camadas 3x3 no lugar de 1 camada 7x7, serão 3 funções de ativação ao invés de apenas 1. Esse aumento na profundidade, e no número de não linearidade, ajuda o modelo a aumentar a capacidade de aprender (ZAFAR et al., 2018a).

No caso da VGG-19, há um aumento do número de camadas em comparação ao VGG-16. Foi verificado que esse aumento não estava resultando em melhoras no nível de eficácia. Com isso pode-se verificar que o aumento de camadas não se reflete no aumento de performance da rede.

Com menos parâmetros para serem aprendidos, é encontrada uma melhor convergência, aumentando sua rapidez e reduzindo o problema de sobre ajuste. Na figura 17, são descritos os modelos de VGG-16 e VGG-19.

Figura 17 – Modelos VGG-16 E VGG-19



Fonte: ZAFAR et al., 2018a.

3.12 Redes *Inception-V3*

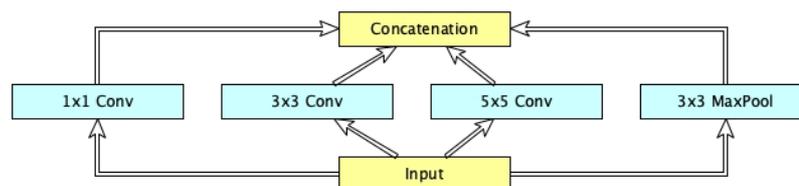
Uma maneira simples de aumentar o desempenho de uma rede neural profunda é aumentando seu tamanho. Dessa forma, seria aumentado o número de camadas e de neurônios em cada camada. Mas isso pode acarretar um aumento significativo no número de parâmetros da rede, necessitando de um poder computacional muito maior, o que pode, muitas vezes, inviabilizar a sua implementação. Outros problemas encontrados, alterando o tamanho da rede, são os do gradiente (Dissipação do gradiente), pois podem tornar a rede mais propensa a *overfitting*, principalmente, em casos em que o número de amostras de treinamento é limitado (SZEGEDY; LIU; et al., 2015).

As redes *inception* foram criadas com a premissa de resolver alguns problemas encontrados em arquiteturas antes apresentadas. As imagens podem ter variações muito grandes no tamanho e, assim, ocuparem áreas de tamanhos diferentes em imagens. Com essa enorme variação de tamanhos e localização das informações em uma imagem, a escolha do tamanho do *kernel* para a realização da operação de convolução se torna muito complexa.

Um kernel grande se faz necessário para identificar informações da imagem distribuídas mais globalmente, enquanto um kernel pequeno é utilizado preferencialmente para identificar informações localmente (SZEGEDY; LIU; et al., 2015).

Para resolver esses problemas, foi criado o modelo de rede neural profunda chamado de *Inception*. Nele foi realizado o seguinte questionamento: por que não criar filtros de vários tamanhos operando no mesmo nível? Dessa forma, tem-se uma rede mais ampla ao invés de mais profunda. Com essa ideia o modelo consiste em quatro caminhos paralelos que realizam convoluções de uma entrada utilizando 3 tipos de filtros convolucionais de diferentes tamanhos para extrair informações de tamanhos espaciais diferentes, além do *MaxPooling*. Após passarem por esses filtros e *MaxPooling* teriam suas saídas concatenadas como ilustrado na figura 18.

Figura 18 – Módulo *Inception*, versão naïve.

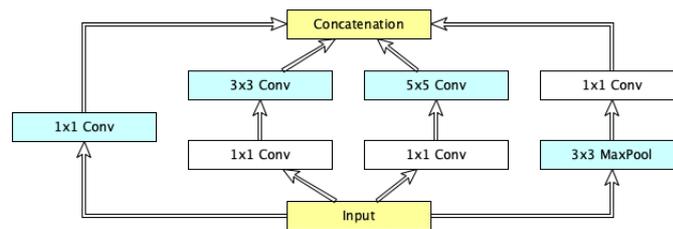


Fonte: (SZEGEDY; LIU; et al., 2015).

Como as redes neurais profundas exigem muito poder computacional, foi necessário limitar o número de canais de entrada da rede adicionando convoluções extras 1x1 antes das convoluções 3x3, 5x5 e, após a camada de pooling (Figura 19). Essas convoluções 1x1, são utilizadas como módulos de redução da dimensão, assim diminuindo o número de canais e diminuindo a complexidade do modelo. Todos esses caminhos utilizam algum tipo de preenchimento para que todas as saídas tenham o mesmo número de canais. Com essa redução, minimiza-se o problema de gargalo computacional e, dessa maneira, pode-se aumentar a profundidade e largura da rede.

Essas convoluções 1x1, além de ajudar a diminuir o tamanho do modelo, podem, de alguma forma, reduzir o *overfitting* da rede melhorando seu nível de precisão (SZEGEDY; LIU; et al., 2015).

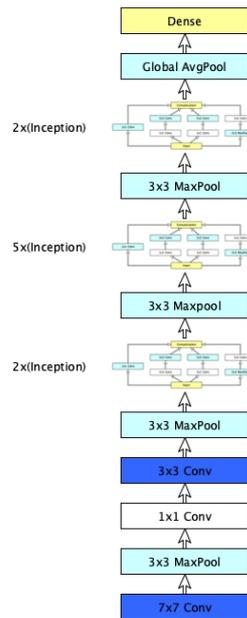
Figura 19 – Módulo Inception com dimensões reduzidas.



Fonte: (SZEGEDY; LIU; et al., 2015).

Utilizando 9 módulos *Inception* e realizando o empilhamento dos mesmos, chega-se assim, a uma configuração com 22 camadas. Com elas, gera-se o modelo de rede popularmente conhecida como GoogLeNet (*Inception-V1*), como demonstrada na figura 20. Este modelo foi o vencedor da competição ILSVRC2014, usando 12 vezes menos parâmetros que a arquitetura vencedora de 2012.

Figura 20 – Modelo GoogLeNet.



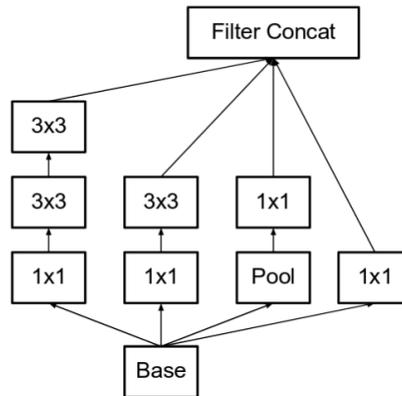
Fonte: (SZEGEDY; LIU; et al., 2015).

Após a criação da *Inception-V1*, foram apresentadas uma série de atualizações que aumentaram a precisão e diminuíram a complexidade da rede. Foram criadas as redes *Inception-V2* e *Inception-V3*. Essa última, utilizada no estudo em questão (SZEGEDY; VANHOUCKE; et al., 2015).

A *Inception-V1* foi refinada de várias maneiras. Primeiramente, pela introdução da ideia de fatoração. Depois, pela normalização em lote (SZEGEDY; VANHOUCKE; et al., 2015).

Nas redes *Inception-V2*, foram criados métodos inteligentes de fatoração. Com isso, as convoluções podem ser mais eficientes, ao pensar em complexidade computacional. Foi implementado um método de fatoração, utilizando duas convoluções 3x3, no lugar de uma convolução 5x5. Sendo assim, obtendo-se melhorias quanto à velocidade computacional. Uma convolução 5x5 é 2,78 vezes mais cara que uma convolução 3x3, conforme ilustrado na figura 21.

Figura 21 – Fatoração da convolução 5x5 agora representada por duas de 3x3.

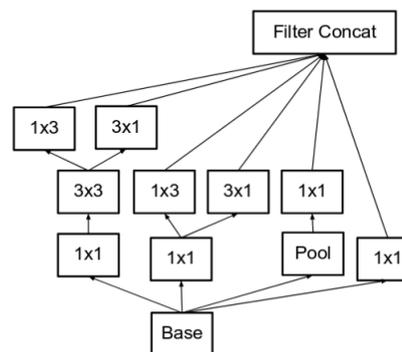


Fonte: (SZEGEDY; VANHOUCKE; et al., 2015).

Além dessa, vieram outras formas de fatorar o tamanho do filtro $n \times n$, em uma combinação de convoluções, $1 \times n$ e $n \times 1$. Em uma convolução 3×3 , é possível executar primeiro uma convolução 1×3 e, em seguida, outra 3×1 . O método é 33% mais barato computacionalmente de se executar apenas uma convolução 3×3 .

Outra opção, foram os bancos de filtros expandidos para remover o gargalo representacional. Se o módulo é aprofundado, pode haver uma redução excessiva nas dimensões e, com isso, perda de informação. Demonstra-se, essa afirmação, na figura 22.

Figura 22 – Bancos de filtros expandidos.



Fonte: (SZEGEDY; VANHOUCKE; et al., 2015).

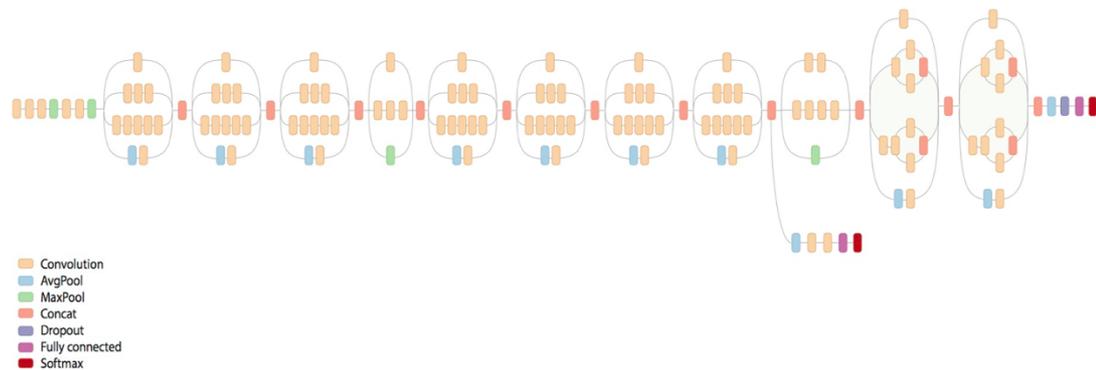
Juntamente com a *Inception-V2*, foi apresentada a *Inception-V3* que incorporou todas as atualizações da antecessora, mas com o incremento das seguintes melhorias:

1. Otimizador RMSProp;

2. Convolução 7x7 fatoradas;
3. BatchNorm nos classificadores auxiliares;
4. Suavização de *Label* (Impede o excesso de ajuste).

O modelo *Inception-V3* é descrito no diagrama abaixo, figura 23.

Figura 23 – Diagrama esquemático do Inception-V3.



Fonte: (SZEGEDY; VANHOUCKE; et al., 2015).

3.13 ResNet

A ResNet é uma rede muito profunda podendo chegar até 152 camadas, utilizando o aprendizado por representação residual, em vez de aprender diretamente a representação do sinal.

Essa tecnologia introduz conexão de salto. Logo, ajusta a camada anterior a próxima camada sem nenhuma modificação. Essa conexão de salto permite a rede ser muito mais profunda que as convencionais de aprendizado profundo, tornando-se o vencedor do ILSVRC 2015 em classificação, detecção e localização de imagens, como também o vencedor da detecção e segmentação do MS COCO 2015 (HE et al., 2016).

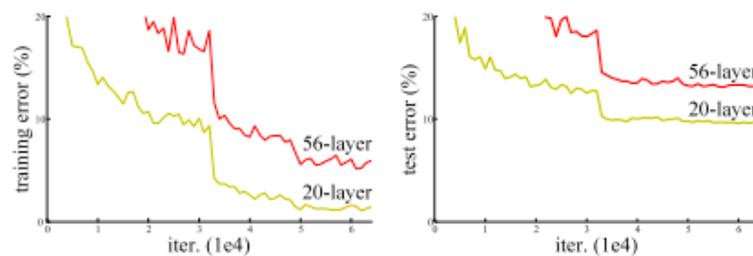
As redes convencionais de aprendizado profundo geralmente possuem camadas de convolução e camadas totalmente conectadas para realizar classificação como VGGNet e AlexNet, sem a utilização de nenhum tipo de atalho ou salto. Quando são criadas essas redes mais profundas, aumentando suas camadas na intenção de melhorar os níveis de precisão, pode acontecer um efeito contrário, pois com o aumento dessas camadas, os

problemas de explosão ou desaparecimento do gradiente podem ocorrer (*Vanishing/Exploding Gradients*) (HE et al., 2016).

Quando se estuda redes convencionais de aprendizado profundo é esperado que quanto mais profunda ela for, mais precisa será. No entanto, isso na maioria das vezes, não é correto. Observe um problema devido a *Vanishing/Exploding Gradients*, na figura 24, que mostra uma rede com 20 camadas, obtendo um melhor desempenho em comparação a uma rede com 56 camadas.

Durante a retropropagação, quando a derivada parcial da função de erro com relação ao peso em cada iteração do treinamento acontece, essa multiplicação de números pequenos ou grandes para calcular o gradiente das camadas, pode ocorrer um aumento do valor, levando o número a valores muito grandes (*Exploding Gradients*). É possível, também, ocorrer a diminuição desse número, levando a valores tão pequenos, podendo até tornar-se zero (*Vanishing*).

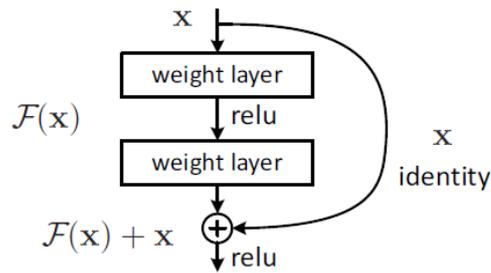
Figura 24 – Erro de treinamento (esquerda) e erro de teste (direita) no CIFAR-10 com redes convencionais de aprendizado profundo de 20 camadas e 56 camadas.



Fonte: (HE et al., 2016).

Para resolver o problema de *Vanishing/Exploding Gradients*, a ResNet criou uma conexão de pulo/atalho (skip/shortcut) que é adicionada da entrada X para à saída, após algumas camadas de peso, como demonstrado na figura 25.

Figura 25 – Elemento básico da rede residual

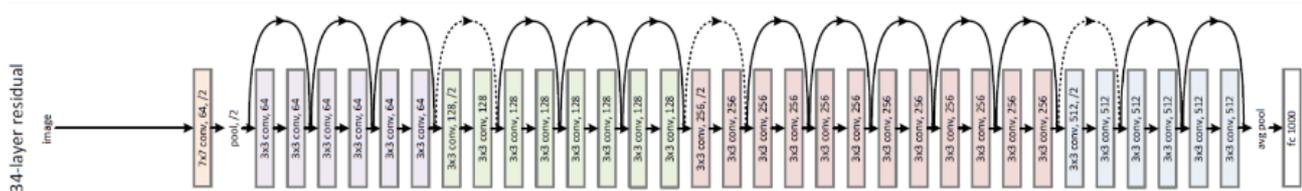


Fonte: (HE et al., 2016).

A saída deste bloco residual é $H(x) = F(x) + x$. As camadas de peso são na verdade uma forma de aprender um tipo de mapeamento residual. Mesmo com o *Vanishing/Exploding Gradients* para as camadas de peso, existe a identidade x para retornar às camadas anteriores.

A figura 26 abaixo, representa a arquitetura ResNet–34, com a descrição de suas 34 camadas.

Figura 26 – Modelo ResNet 34 camadas(HE et al., 2016)



Fonte: (HE et al., 2016).

Para a ResNet, existem 3 tipos de conexões de skip/shortcut quando as dimensões de entrada são menores do que as dimensões de saída.

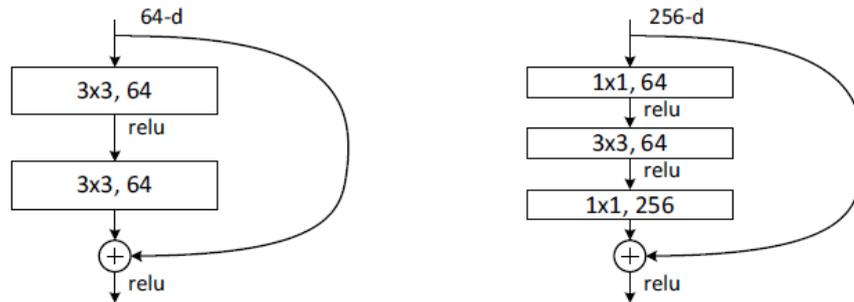
1. O atalho realiza mapeamento identidade com *padding* zero extra incrementando a dimensão. Desta forma não há parâmetros extras.
2. O atalho de projeção é usado apenas para aumentar as dimensões, os outros atalhos são de identidade. Parâmetros extras são necessários.
3. Todos os atalhos são projeções. Parâmetros extras são maiores que os do item 2.

Com o aumento da profundidade do modelo a complexidade do tempo é alta, um design de gargalo é utilizado para reduzir essa complexidade.

Como demonstrada na figura 27 (B), as camadas de convolução 1x1 são adicionadas no início e no fim da rede. Esta técnica foi utilizada pelos modelos “Network In

Network” e GoogleNet. Utilizando estas camadas adicionais 1x1 pode-se reduzir o número de conexões (parâmetros) sem que haja prejuízo no desempenho da rede.

Figura 27 – Bloco básico imagem (A) e bloco com design de gargalo (Bottleneck Design) (B).



Fonte: (HE et al., 2016).

Dessa forma, utilizando a *Bottleneck Design* a ResNet de 34 camadas, torna-se uma ResNet de 50 camadas. Esse modelo possui ainda suas versões de 101 e 152 camadas.

Foram realizados testes e comparações entre as redes convencionais de aprendizado profundo e ResNet, quando uma rede convencional de aprendizado profundo de 18 camadas é usada tem melhor desempenho que uma de 34 camadas, devido ao problema de *Vanishing/Exploding Gradients*.

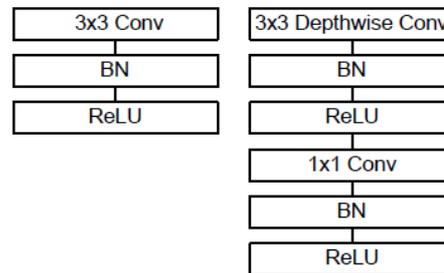
Já quando é utilizada uma ResNet, a rede com 34 camadas tem um desempenho melhor que a rede com 18 camadas. O problema encontrado nas redes convencionais de aprendizado profundo foi resolvido pelos atalhos de conexões.

3.14 MobileNetV2

As MobileNets são redes neurais convolucionais desenvolvidas pelo Google, que foram criadas para facilitar a utilização de CNNs em dispositivos móveis. Os pesquisadores buscaram criar um modelo que proporcionasse uma alta precisão, mas com o número de parâmetros e operações matemáticas o mais baixo possível (HOWARD et al., 2017).

A Diferença entre as redes neurais convolucionais em geral e a arquitetura MobileNet, é a divisão em convolução profunda e convolução ponto a ponto como ilustrado na figura 28.

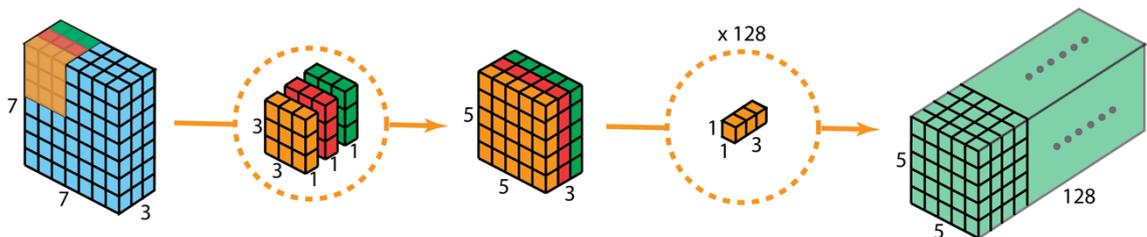
Figura 28 – Convolução padrão (A), convolução separável em profundidade (B) com BN e ReLU (HOWARD et al., 2017).



Fonte: (HOWARD et al., 2017).

As etapas das convoluções do MobileNet são resumidas na figura 29, abaixo. Nela, a entrada possui um volume de $7 \times 7 \times 3$ e ficou com um volume de saída $5 \times 5 \times 128$.

Figura 29 – O processo geral de convolução separável em profundidade.



Fonte: (BAI, 2019).

Realizando essa separação, consegue-se diminuir o número de multiplicações de forma muito significativa e o número de parâmetros da rede. Assim, haverá um aumento na performance, mas com uma diminuição da precisão.

Esse tipo de arquitetura possui desvantagens, em relação a convolução padrão, pois há uma redução do número de parâmetros da rede, caso a rede seja pequena. O modelo pode acabar com poucos parâmetros e, com isso, não conseguir aprender adequadamente, durante o treinamento. Se for utilizada adequadamente, no entanto, consegue-se aumentar a eficiência, sem reduzir, significativamente, a eficácia, reduzindo o custo computacional em um *kernel* de 3×3 em 8 a 9 vezes em relação a uma convolução padrão (HOWARD et al., 2017).

O MobileNetV2 (figura 30) se destaca de seu antecessor por basicamente adicionar duas ideias principais:

- Camada residual invertida;
- Camada de gargalo linear.

Figura 30 – Arquitetura geral do MobileNetV2.

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

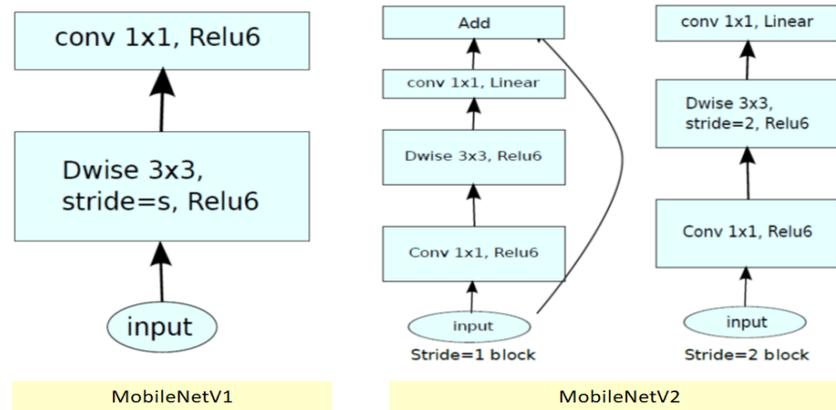
Fonte: (SANDLER et al., 2018).

Onde:

- t : fator de expansão;
- c : número de canais de saída;
- n : número de repetição e
- s : passo.

Essa nova camada convolucional, desenvolvida na MobileNetv2, consiste primeiro em realizar uma convolução ponto a ponto (1x1), sendo usada para expandir o mapa de recursos de entrada de baixa dimensão, para uma dimensão superior adequada para ativações não lineares. Com isso, o ReLU é aplicado, realizando uma filtragem espacial do tensor de maior dimensão. Em seguida, é realizada uma convolução 3x3 com ativação ReLU. Por fim, é realizada a volta para o subespaço de baixa dimensão, usando uma convolução ponto a ponto (1x1), ilustrada na figura 31 (SANDLER et al., 2018).

Figura 31 – Blocos convolucionais em rede MobileNetV1 e MobileNetV2.



Fonte: (SANDLER et al., 2018).

3.15 Técnicas de Regularização

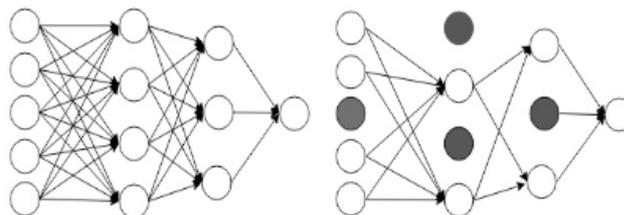
3.15.1 Dropout

É a técnica de regularização mais popular para redes neurais profundas, com ela é possível reduzir o *overfitting* (MOOLAYIL, 2019; ZAFAR et al., 2018b).

O *dropout* funciona utilizando o abandono a cada iteração de treinamento de alguns neurônios. Eles podem ser ligados e desligados utilizando uma probabilidade aleatória (geralmente 50%), ilustrado na figura 32 (SEWAK et al., 2018).

Utilizando essa técnica força-se a rede a aprender os mesmos conceitos, só que por caminhos diferentes. Após o treinamento todos os neurônios são ativados e assim os caminhos se comportam como várias redes, que a partir delas são calculadas a média do resultado, melhorando a generalização (ZAFAR et al., 2018b).

Figura 32 – Rede neural normal e rede neural depois do dropout



Fonte: (ZAFAR et al., 2018b).

O *dropout* pode ser aplicado aos nós da camada de entrada e camadas ocultas, mas não aos nós de saída.

3.15.2 *Data Augmentation*

Muitas vezes é encontrada aplicações problemáticas onde o número de amostras é pequeno, tornando-se insuficiente. Isso dificulta a rede neural de aprender e conseguir generalizar os problemas a serem classificados. A *Deep Learnig* necessita de uma quantidade grande de imagens para conseguir realizar o treinamento e obter resultados relevantes de reconhecimento, quando aplicados a imagens.

Para atenuar esses problemas, ocorridos em dados insuficientes, pode-se aplicar uma técnica conhecida como Aumento de Dados – *Data Augmentation*. Com a aplicação dessa ferramenta, amplia-se o tamanho do conjunto de dados, aplicando técnicas de processamento de imagens.

Os dados ou imagens podem ser gerados artificialmente de maneira inteligente, ajudando a resolver problemas de ajuste excessivo, dados insuficientes e conjunto de dados desequilibrados, assim melhorando o desempenho em classificação.

Várias técnicas de processamento de imagens são aplicadas para criar essas imagens, entre as mais utilizadas estão:

- Rotacionar as imagens em vários ângulos;
- Inversão da imagem;
- Alterar posicionamento da imagem;
- Adicionar ruído na imagem;
- Alterar as condições de iluminação da Imagem;
- Troca de canais de cores (ZAFAR et al., 2018a).

3.15.3 *Transfer Learning*

É possível treinar uma CNN do zero, dependendo da quantidade de imagens e poder computacional disponível. É algo que pode levar muito tempo. Nesse caso, é aplicada a transferência de aprendizado (*transfer learning*). Nela são utilizados pesos de um modelo já treinado (pré-treinado), geralmente com milhões de imagens e pertencentes a milhares de classes, utilizando uma quantidade de hardware muito potente. Muitas vezes, com várias placas GPUs de alta performance (ZAFAR et al., 2018a).

O *transfer learning* é uma técnica de aprendizado de máquina que consiste em utilizar o conhecimento já aprendido em problemas anteriores e usá-los para resolver novos problemas.

Com essa técnica, existem algumas vantagens. Com ela não é necessário um conjunto de dados muito grande e nem um poder computacional muito expressivo, pois uma grande quantidade de dados já foi aprendida.

Com a utilização de *transfer learning* é possível realizar um ajuste fino. Nesse ponto, são mantidas as camadas convolucionais do modelo escolhido. Todavia, modifica-se e treina-se somente as camadas totalmente conectadas (TOO et al., 2019).

É possível também treinar o modelo por inteiro utilizando os pesos já aprendidos em treinamentos anteriores, assim melhorando os níveis de precisão e diminuindo o número de épocas de treinamento.

3.16 Métricas de Avaliação

A qualidade de um modelo de Rede Neural Convolucional, geralmente é avaliada utilizando a acurácia e o custo como métricas de qualidade de predição.

Pode-se citar também, o coeficiente Kappa e a curva ROC, essa última criada e aplicada, na maioria das vezes, em modelos com duas classes. Dessa forma, embora bastante conhecida, a curva ROC, geralmente não se aplica à modelos multiclasse, como nos modelos dessa pesquisa.

Além destas métricas citadas anteriormente, existem outras necessárias para medir o quanto o resultado de uma classificação está próximo do ideal. Na literatura existem diversas métricas diferentes, podendo ser utilizadas individualmente ou em conjunto para possibilitar uma melhor avaliação do modelo, sendo as mais escolhidas, por pesquisadores, as descritas nos tópicos abaixo.

3.16.1 Matriz de Confusão

A matriz de confusão é uma tabela que mostra os valores observados e previstos em um problema de classificação. É utilizada para medir o desempenho de um modelo de classificação. Ela pode ser de duas ou mais saídas.

Uma matriz de confusão possui o mesmo número de linhas e colunas, mas com valores distintos de seus rótulos. Nas colunas da matriz, coloca-se cada valor do rótulo

previsto. Nas linhas da matriz, coloca-se cada valor de rótulo real, como apresentado na figura 34.

Figura 33 – Matriz de confusão.

	Predicted: Positive	Predicted: Negative
Actual: Positive	TP	FN
Actual: Negative	FP	TN

Fonte: autor da tese.

Para cada ponto de dados, são adicionadas às células correspondentes da matriz de confusão com base no valor previsto e real do rótulo. Essa matriz fornece uma visão não apenas dos erros cometidos pelo classificador, mas quais os tipos de erros são cometidos (DYRMANN; KARSTOFT; MIDTIBY, 2016).

Definição dos termos:

- Verdadeiro positivo (TP): O modelo previu como positivo e realmente é positivo;
- Verdadeiro negativo (TN): O modelo previu como negativo e realmente é negativo.
- Falso positivo (FP): O modelo previu positivo e na verdade é negativo;
- Falso negativo (FN): O modelo previu negativo e na verdade é positivo.

Estes dados extraídos da matriz de confusão, podem ser utilizados de forma extremamente útil para calcular várias outras métricas, como por exemplo *Accuracy*, *Recall*, *Precision* e *F1_Score*.

3.16.2 Acurácia (*Accuracy*)

A acurácia é uma medida geral do poder preditivo do modelo. Ela apresenta o número total de previsões verdadeiras dividido pelo total de previsões, como descrita na equação (NISBET et al., 2018):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

Problemas com acurácia são os seguintes:

- Supõe que a taxa de erro para cada classe de classificação seja igual.
- Uma precisão de 99% pode ser boa ou muito ruim, dependendo da natureza do problema e os custos diferenciais de ambos os erros.
- A acurácia em dados desbalanceados não garante uma avaliação precisa.

Existem outras métricas utilizadas para validar o modelo, como é o caso da *Recall*, *Precision* e *F1_Score* (BEN-DAVID, 2007).

3.16.3 Precisão (*Precision*)

A precisão mostra o grau de confiança que um modelo pode atribuir às suas predições, ou seja, das predições realizadas quantas realmente são corretas (NISBET et al., 2018). Essa métrica é calculada pela equação abaixo:

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

Utilizando essa métrica significa dizer que a maioria das classes detectadas pelo modelo como positivo são verdadeiras, ou seja, o número de falso positivo é baixo. Mas nessa métrica uma boa precisão não significa sucesso na identificação de todas as classes verdadeiras.

A precisão pode ser usada em uma situação em que os Falsos Positivos são considerados mais prejudiciais aos Falsos Negativos.

3.16.4 Revocação/Sensibilidade (*Recall*)

Na *Recall* é avaliado a proporção de valores positivos observados e previstos como positivo (NISBET et al., 2018). Essa métrica reflete a proporção total de observações positivas previstas corretamente pelo modelo.

Entretanto, ela não implica somente em classes positivas detectadas como verdadeiras.

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

A *recall* pode ser usada em uma situação em que os Falsos Negativos são considerados mais prejudiciais aos Falsos Positivos.

3.16.5 *F1_Score*

A métrica *F1_Score* pode ser aplicada nos casos onde é necessária uma combinação entre as métricas *Recall* e *Precision*. Ela utiliza uma média harmônica de *Precision* e *Recall*, utilizando a equação abaixo:

$$F1 = 2 * \frac{precision*recall}{precision+recall} \quad (8)$$

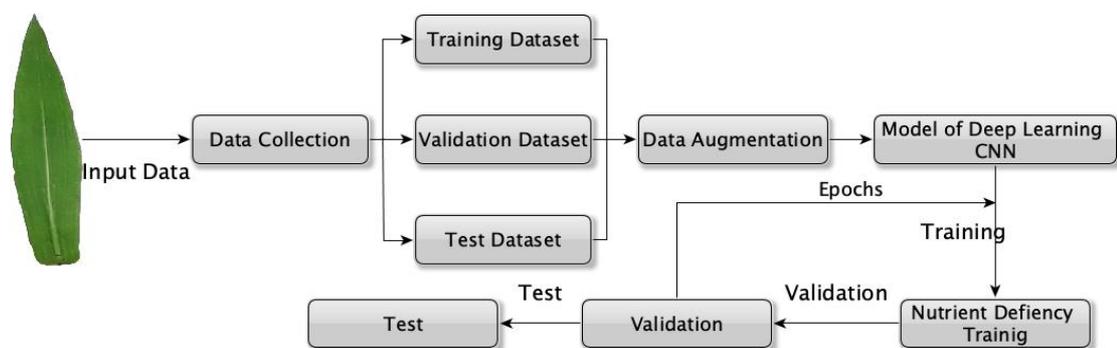
Para criar um modelo de classificação equilibrado, sendo ele o ideal entre *Recall* e *Precision*, então é necessário maximizar os valores desta métrica.

4 METODOLOGIA

Para realizar os estudos da identificação de deficiências nutricionais no solo, utilizando características das imagens das plantas, faz-se necessária a extração de informações. Essa pode ser realizada pelas suas folhas com a utilização da diagnose foliar. Nesta técnica são utilizadas as características visíveis nas folhas para a identificação de deficiências nutricionais (MALAVOLTA, 2006). Essas características são as informações ou dados utilizados como base de conhecimento para alimentar modelos computacionais, nos quais são utilizadas técnicas de Inteligência Artificial e, por meio disso, a realização da análise de deficiência do solo.

No processo de identificação nutricional da *Brachiaria brizantha* cv. Marandu, utilizaram-se as seguintes etapas: cultivo das amostras das pastagens em casa de vegetação; coleta das imagens das folhas cultivadas; criação do conjunto de imagens; estudo das arquiteturas de CNNs; proposição e testes de diferentes modelo de CNNs; teste do conjunto de imagens criado com várias arquiteturas de CNNs existentes e, por fim, um comparativo entre as arquiteturas propostas. A figura 34 apresenta uma visão geral das etapas para o sistema proposto de identificação de deficiência nutricional.

Figura 34 – Visão geral do sistema proposto de aprendizagem profunda.



Fonte: autor da tese.

De acordo com a figura 34, o sistema é implementado seguindo as seguintes etapas:

- O conjunto de dados de imagem é dividido em 3 categorias: treino, validação e teste;
- As imagens são aumentadas utilizando a técnica de *Data Augmentation*;
- Após o processamento das imagens um modelo de CNN é selecionado e treinado em várias épocas, utilizando o conjunto de dados de treino;

- Em cada época de treinamento o modelo é avaliado empregando o conjunto de dados de validação;
- Na última etapa, após o treinamento concluído, o modelo é testado e avaliado. Nesse caso, utiliza-se o conjunto de dados de teste.

Para atingir o objetivo de construir um modelo capaz de realizar a identificação da deficiência nutricional nas imagens das folhas da *Brachiaria brizantha* vs. Marandu, foi necessário elaborar um conjunto de dados de imagens processadas, extraíndo o seu fundo e rotulando-as, conforme sua deficiência.

Além disso, foram implementados vários modelos de CNNs. Primeiramente, um modelo desenvolvido com base na rede *AlexNet* e modelos pré-treinados VGG-16, VGG-19, *Inception-V3*, *Resnet-50* e o modelo para tecnologia móvel *MoblileNetV2*. As redes foram testadas e comparadas para, assim, analisar qual seria a mais apropriada à aplicabilidade em questão, como também qual atingiria o melhor desempenho com níveis de precisão aceitáveis.

4.1 Cultivo e Aquisição das Imagens

Um dos pré-requisitos fundamentais para a construção de modelos computacionais, capaz de reconhecer padrões em imagens, é o seu conjunto de dados. No caso de modelos supervisionados, que utilizam imagens para aprender padrões e, com isso conseguir generalizar o processo de reconhecimento de imagens, faz-se necessário um conjunto com uma quantidade de imagens com o intuito de proporcionar as informações para, dessa forma, o sistema conseguir aprender as características necessárias e, assim, classificar tais imagens. Também é importante ter um bom conjunto de dados para conseguir realizar uma divisão em dados de treino, validação e teste.

Para criar esse conjunto de dados foram realizados o cultivo e aquisição das imagens no Instituto Federal de Educação, Ciência e Tecnologia de Rondônia – *Campus Colorado do Oeste*, localizado às margens da BR 435, km 63, zona rural do município de Colorado do Oeste, estado de Rondônia.

O experimento consistiu na caracterização de sintomas visuais de deficiência de Nitrogênio, Fósforo e Potássio na cultura da *Brachiaria brizantha* cv. Marandu, cultivada em casa de vegetação. Os sintomas foram avaliados por meio da análise de imagens

associadas a análises foliar e do solo, durante vários estádios de desenvolvimento da forrageira.

As amostras foram desenvolvidas seguindo os três princípios básicos da estatística experimental: repetição, casualização e controle local (FERREIRA, 2000). Sendo assim, o delineamento experimental foi em blocos casualizados (DBC), arranjado em esquema fatorial 3 x 3, sendo três nutrientes (N, P e K) e três níveis de adubação (nível inicial do próprio substrato, 50% e 100% das doses recomendadas para a cultura) (FERREIRA, 2000).

As avaliações foram realizadas com cinco repetições, totalizando 45 unidades experimentais. A unidade experimental foi composta por tubos de PVC de 500 mm (Figura 35), cortados com comprimento de 0,6 metros tendo capacidade para 50 kg, sendo preenchidos com areia de baixa fertilidade e posteriormente adicionado os nutrientes para cada tratamento. A quantidade de areia foi a mesma para todos os vasos e a irrigação foi realizada em dias alternados utilizando uma quantidade de água conhecida como a capacidade de "vaso", com valor calculado utilizando 80% da capacidade de campo (CASAROLI; DE JONG VAN LIER, 2008).

As imagens foram tomadas no terço médio da planta, utilizando de três a quatro folhas diagnósticas por vaso e registradas semanalmente como descrito na tabela 4.

Tabela 4 – Período de plantio e registro das imagens.

Etapa	Data	Avaliação e fotografia
Plantio	08/01/2019	Não
Primeira avaliação	16/02/2019	Sim
Segunda avaliação	22/02/2019	Sim
Terceira avaliação	15/03/2019	Sim
Corte	19/03/2019	Não
Quarta avaliação	07/06/2019	Sim
Quinta avaliação	15/06/2019	Sim
Sexta avaliação	21/06/2019	Sim

Fonte: autor da tese.

Figura 35 – Unidade experimental composta por tubos de PVC.



Fonte: autor da tese.

O substrato foi inicialmente submetido à caracterização química e granulométrica para ajuste e testes das doses dos macronutrientes e, também, para adição dos demais nutrientes.

Para a semeadura foram utilizadas 20 sementes puras viáveis por vaso, semeadas na profundidade de dois centímetros como ilustrado na figura 36. Aproximadamente, 15 dias após a emergência (DAE), foi realizado o desbaste, permanecendo seis plantas por vaso. Aos 30 DAE, foi realizado o primeiro corte para uniformizar e estimular o perfilhamento.

Figura 36 – Semeadura e emergência do capim.



Fonte: autor da tese.

Para a adubação foi preparada solução nutritiva de macronutrientes composta pelos produtos comerciais ureia na concentração de 21,7 g/l (fonte de Nitrogênio), Superfosfato Triplo na concentração de 37,11 g/l (fonte de Fósforo) e Cloreto de Potássio na concentração de 16,3 g/l (fonte de Potássio), sendo recomendado 5 ml para 1 kg de solo. A solução foi preparada em laboratório individualmente para cada macronutriente e aplicada conforme os tratamentos (figura 37).

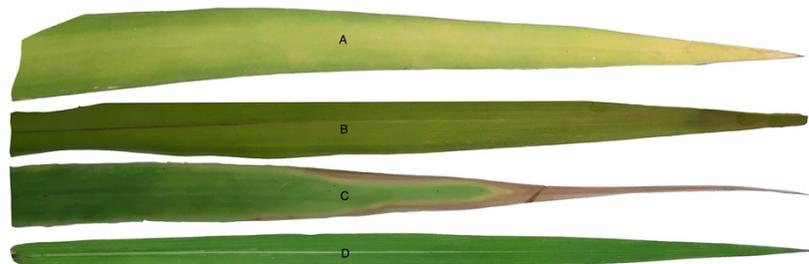
Figura 37 – Adubação dos devidos tratamentos.



Fonte: autor da tese.

Após o primeiro corte, semanalmente, foram registradas fotos de cada unidade experimental (Figura 38) e de quatro folhas diagnósticas de cada vaso. Essas foram representadas pela segunda folha completamente expandida, do ápice para a base dos perfilhos.

Figura 38 – Imagem folha com deficiência de Nitrogênio (A), Fósforo (B), Potássio (C) a nível do substrato e planta saudável (D).



Fonte: autor da tese.

A partir dessas amostras, as imagens das folhas foram capturadas com um aparelho celular Samsung Galaxy J7 Pro, equipado com uma câmera de 13 MP, f/1.9, LED

flash, 1080p e mini *Studio* fotográfico portátil para as três deficiências nutricionais, bem como para a planta com nutrição completa, totalizando 284 imagens.

Essas imagens foram catalogadas, de acordo com cada estado nutricional, sendo rotuladas e divididas em pastas, conforme seu período de cultivo, nutriente avaliado e nível de adubação.

É possível a utilização de banco de dados para o armazenamento destas imagens, principalmente quando a quantidade de informações é muito grande.

Para armazenamento de dados não estruturados, como é o caso de imagens, sendo que em aplicações de *Deep Learning* a quantidade é normalmente muito grande, recomenda-se a utilização do banco de dados NoSQL. Estes bancos de dados são projetados e otimizados para alcançar um melhor desempenho quando existe um grande volume de dados. Além disso, é melhor para dados semiestruturados e não estruturados (FOWLER, 2015).

4.2 Processamento das Imagens

O processamento das imagens foi realizado com um sistema desenvolvido na linguagem *python* e com uma biblioteca de processamento de imagem chamada OPENCV. Ela é vastamente utilizada por grandes empresas de renome mundial, como o Google, Yahoo, Microsoft, IBM, entre outras, tal como grupos de pesquisa e até entidades governamentais (OPENCV, 2016). Essa ferramenta é uma biblioteca de processamento de imagem de código aberto, contendo em sua implementação mais de 2.500 algoritmos otimizados. Eles podem ser aplicados para realizar várias funções de tratamento de imagem, como manipulação, segmentação, rastreamento e reconhecimento de padrões.

O processamento realizado nas imagens coletadas e armazenadas no banco de dados é mínimo, pois a tecnologia de rede neural escolhida para realizar esse estudo utiliza pouca engenharia de recursos para realizar seu trabalho. Estas características (*features*) necessárias para realizar a classificação são encontrados pela própria rede por meio do seu treinamento, como descrito no item 3.9 (LECUN et al., 2015)(MOHANTY et al., 2016).

Inicialmente foi realizada a segmentação da imagem. O fundo da imagem original foi removido, deixando apenas a folha a ser analisada com um fundo branco (ZÚÑIGA, 2012).

A segmentação da imagem utilizou a técnica de segmentação por limiarização de imagem, o qual divide a imagem em duas regiões (objeto e fundo), considerando o histograma de níveis de cinza uma distribuição bimodal. Esse limiar separa os pixel acima do limiar e abaixo dele. Com isso é possível identificar os contornos e extrair somente a folha a ser analisada e, com essa informação, criado o conjunto de dados das imagens (CHITYALA; PUDIPEDDI, 2014; HAMUDA et al., 2016; ROMUALDO et al., 2014; WILLIAM; OLIVEIRA, 2015; XU et al., 2011; ZÚÑIGA, 2012). A ferramenta “Adobe Photoshop CC 2019” foi usada para ajustes nas imagens, quando necessário.

Essas imagens segmentadas foram divididas em seus respectivos grupos nutricionais. Dessa forma, foram divididas em deficiência de N, P, K e folha saudável, sendo armazenadas separadamente cada uma em seu grupo representativo.

Após a execução da segmentação e o salvamento das imagens, nas respectivas pastas, foram carregadas em memória. A seguir, houve a criação de uma lista contendo a imagem e seu respectivo rótulo, como mostrado na figura 39.

Parte do sistema desenvolvido é responsável por navegar nas pastas, nas quais encontram-se as imagens, e verificar qual é o nome da pasta, criar o rótulo para ela. Em seguida, houve o armazenamento de todas essas imagens em uma lista.

Figura 39 – Matriz das imagens segmentadas.

	Images	Labels
0	[[[255, 255, 255], [255, 255, 255], [255, 255, ...	2
1	[[[255, 255, 255], [255, 255, 255], [255, 255, ...	0
2	[[[255, 255, 255], [255, 255, 255], [255, 255, ...	1
3	[[[255, 255, 255], [255, 255, 255], [255, 255, ...	1
4	[[[255, 255, 255], [255, 255, 255], [255, 255, ...	0

Fonte: autor da tese.

Após o carregamento de todo o conjunto de dados, foi necessário realizar partição dos dados em 3 subconjuntos (figura 40). Cada um deles com uma finalidade diferente. Na sequência, é apresentada uma explicação de cada um deles:

- **Treinamento:** esse subconjunto é responsável por alimentar o treinamento do modelo e é composto por 80% do total das amostras. Ele pode ser usado para treinar vários modelos com diferentes configurações. No entanto, a avaliação do desempenho não é realizada por esse conjunto. Ele é utilizado para identificar a melhor função que representará as classes do conjunto de dados.

- Validação: esse subconjunto é utilizado para realizar uma avaliação imparcial de cada modelo enquanto ajusta os hiperparâmetros e é composto por 10% das amostras. É utilizado para verificar a eficiência do modelo de aprendizado e com isso identificar possíveis problemas do sobreajuste (*overfitting*). Depois de executar diferentes configurações de hiperparâmetros e analisando o desempenho de cada modelo treinado com esses dados de validação, um modelo vencedor é selecionado.
- Teste: para realizar testes do sistema foi separado 10% das amostras. Esses dados são utilizados como uma avaliação final dos modelos, após o treinamento e validação, isso ajuda a medir o desempenho com dados reais para futuras previsões. Com esses dados de teste pode-se analisar e verificar o quão generalista está o modelo de reconhecimento de imagens proposto.

Figura 40 – Divisão do conjunto de dados.



Fonte: autor da tese.

Antes da execução da divisão dos dados, as imagens foram organizadas de forma aleatória para possuir amostras de todas os estados nutricionais nos três subconjuntos.

Além dessa divisão, e do pré-processamento nas imagens, ainda foi necessário realizar uma técnica utilizada muitas vezes em *Deep Learning* chamada de *Data Augmentation*, descrita no tópico a seguir.

Após a ampliação do número de amostras também foi necessário realizar o redimensionamento do tamanho das imagens, pois quanto maior o tamanho da imagem mais recursos computacionais serão necessários para execução da tarefa de treinamento. Esse redimensionamento também foi aplicado para atender as características de entrada das redes convolucionais pré-treinadas.

4.2.1 Aplicação do *Data Augmentation*

Em *Deep Learning*, quanto maior a quantidade de dados disponível, para treinar e testar o modelo, mais o modelo aprende, bem como mais generalista ele se torna. Sendo assim, neste projeto foi necessário realizar a implantação da técnica de aumento de dados, pois o *dataset* é relativamente pequeno. Como descrito anteriormente, para melhorar o

desempenho da CNN, diminuindo o *overfitting*, é possível criar imagens de forma artificial utilizando uma técnica conhecida como *Data Augmentation* (Aumento de dados) (DYRMANN; KARSTOFT; SKOV, 2016; ZHANG, X. et al., 2018b)(WANG et al., 2018).

O conjunto de dados foi aumentado em 9 vezes, realizando espelhamento, rotacionamento das imagens e mudança de brilho. Dessa forma, gerou-se um total de 2556 amostras (LIU et al., 2017; MA et al., 2018b; ZHANG, X. et al., 2018b; ZHU et al., 2018), descrito na tabela 5 e ilustrado na figura 41.

Figura 41 – Exemplo de aumento de dados usados nas imagens.



Fonte: autor da tese

Tabela 5 – Lista de nutrientes e o número de amostras após o Data Augmentation.

Nutrientes	No. de amostras	No. de amostras Data Augmentation
Deficiência de Nitrogênio (N)	104	936
Deficiência de Fósforo (P)	35	315
Deficiência de Potássio (K)	100	900
Planta com nutrição completa	45	405
Total	284	2556

Fonte: autor da tese

4.2.2 Normalização dos Dados com Min-Max

Utilizou-se o método conhecido como mín-máx (SINGH, 2019) a fim de realizar a normalização dos dados. Nesse caso, simplesmente, realizando divisões dos pixels por 255, deixando os valores de pixel entre 0 e 1 (TODA; OKURA, 2019) (ALLA; ADARI, 2019). Assim, este método é definido como:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (9)$$

Com os dados normalizados no intervalo de 0 a 1, consegue-se a estabilidade numérica nos algoritmos, como também, o convergir mais rapidamente.

Sem a escala dos dados, em muitos problemas, os valores dos pesos podem aumentar de maneira indesejável, levando a instabilidade da rede e aumentando o erro de generalização. Dessa forma, garantiu-se uma mesma escala, evitando que o algoritmo interpretasse os dados de forma diferente (ACCOUNT; SHARMA, 2018).

Após a realização de todas as etapas, os dados ficaram prontos para construir o vetor capaz de armazenar os pixels das imagens segmentadas e normalizadas, ficando aptas a serem apresentadas ao modelo para treinamento e testes.

Outro passo importante a ser analisado é o tamanho do vetor de característica ou dados. Se o vetor for muito grande pode levar a um consumo muito grande também de recursos computacionais para a execução das avaliações. Em consequência disso, exigindo um hardware muito mais poderoso e também um aumento no tempo de execução. Em contrapartida, um número muito pequeno de características pode levar a avaliações imprecisas (ARRIBAS et al., 2011).

4.3 Modelos Desenvolvidos

Foram desenvolvidos modelos de redes neurais convolucionais para realizar o reconhecimento de deficiências nutricionais na *Brachiaria brizantha* cv. Marandu. No total foram implementados 6 modelos diferentes. Um deles foi desenvolvido baseado no AlexNet e realizou a implementação de todas as camadas e os modelos pré-treinados VGG-16, VGG-19, *Inception-V3*, Resnet-50 e MobileNetV2.

4.3.1 Configurações dos parâmetros das Redes Neurais Convolucionais aplicadas

Para o desenvolvimento de modelos de redes neurais convolucionais muitos hiperparâmetros são utilizados, destacando-se a taxa de aprendizado (*learning rate*). Pode-se afirmar que esse parâmetro é considerado como um dos mais importantes a serem gerenciados em modelos de CNN. Ele afeta diretamente o aprendizado dos modelos, pois controla a quantidade de alterações nos pesos, em cada época de treinamento. Isso influencia diretamente, também, na velocidade em que o modelo irá aprender (GOODFELLOW et al., 2016) (BENGIO, 2012).

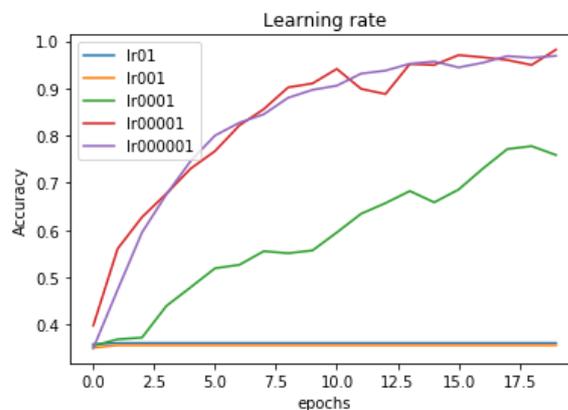
Com uma taxa adequadamente configurada, o modelo aprenderá mais rapidamente a aproximar a função dos recursos disponíveis para um determinado número de épocas. Na maioria das vezes, uma taxa de aprendizado muito grande, resultará em uma

atualização de pesos muito grande, em vez de diminuir o erro de treinamento. Já muito pequena, além de um treinamento mais lento, pode nunca convergir e ficar presa em uma solução abaixo do ideal (GOODFELLOW et al., 2016). Portanto, não se deve escolher uma taxa de aprendizado muito grande ou muito pequena, mas configurar a taxa para que o modelo consiga encontrar pesos suficientemente bons para alcançar resultados adequados ao problema avaliado.

Existem estudos referentes ao valor ideal para a taxa de aprendizado, segundo os quais, esses valores precisam ser menores que 1 e maiores que 10^{-6} (BENGIO, 2012).

Nesta pesquisa foram testadas várias taxas de aprendizado como, por exemplo, os valores 0,1, 0,01, 0,001, 0,0005 e 0,0001. Dentre esses valores, para redes VGG-16 e 19 e a baseada na AlexNet, os melhores resultados foram alcançados ao se utilizar um “*learning-rate=0,0001*”. Já para redes *Inception-V3*, *Resnet-50* e *MobileNetV2*, os melhores resultados foram com um “*learning-rate=0,00001*”. Na figura 42, observa-se um exemplo dos resultados na avaliação do modelo VGG-16, variando as taxas de aprendizado.

Figura 42 – VGG-16 acurácia do modelo em relação a taxa de aprendizado.



Fonte: autor da tese.

Para as redes *Inception-V3*, *Resnet-50*, *VGG-16*, *VGG-19* e *MobileNetV2*, os pesos utilizados foram previamente treinados e carregados da base de dados do *ImageNet*, que contém cerca de 1,2 milhão de imagens e 1000 classes de categorias. Esses dados foram utilizados por transferência de aprendizado. Para a arquitetura, na qual não houve a utilização de pesos pré-treinados, o treinamento foi executado por completo, sendo eles carregados de forma aleatória.

Os modelos utilizaram a função de ativação ReLU (GLOROT et al., 2011) e foram aplicados a todas as camadas. Exceto na camada de saída implementada com a *Softmax* (DUCH; JANKOWSKI, 1999).

4.3.2 Tecnologia Utilizada

Os processos de treinamento e teste dos modelos CNN foram implementados, utilizando as bibliotecas *scikit-learn*, *Keras*, *TensorFlow*, *pillow*, *numpy*, *matplotlib*, *opencv*, entre outros. Elas utilizam a linguagem de programação *python*. O *hardware* utilizado nos modelos foi uma máquina com CPU Core i5-7400k, 16GB de RAM e uma GPU NVIDIA TITAN XP com arquitetura Pascal, 12GB de memória, 3.840 núcleos CUDA (*Compute Unified Device Architecture*) e com o desempenho de 12 TFLOPS.

Essa GPU possui suporte à tecnologia CUDA. Trata-se de uma plataforma de computação paralela, criada para que desenvolvedores possam usufruir de forma mais precisa e livre do potencial de processamento paralelo proporcionado pelas placas de vídeo. Em suma, os desenvolvedores têm acesso a todo poderio dos processadores gráficos das placas de vídeo da Nvidia. No caso, ao se utilizar essa tecnologia, todo trabalho sequencial é processado pela CPU do computador, enquanto a parte intensiva de computação é executada por milhares de núcleos de GPU, em paralelo (NVIDIA, 2020).

Vale ressaltar que essa tecnologia de computação, baseada na utilização de GPUs, trouxe a possibilidade de processar muito mais que imagens e modelos de CNN, como também permitiu desenvolver aplicativos para diversos fins, desde finanças até *software* para medicina e, principalmente, em *Deep Learning*.

A tecnologia CUDA foi utilizada no desenvolvimento desta pesquisa. Com ela todo o processo de treinamento foi acelerado. Dessa forma, o tempo de treinamento foi reduzido em até 15 vezes. Empregou-se essa em conjunto com as plataformas TensorFlow e Keras.

4.3.3 Treinamento das redes neurais convolucionais

Todos os modelos implementados, nesta pesquisa, utilizaram o mesmo algoritmo para realização do treinamento da rede. Esse procedimento consiste em alterar os valores dos pesos das camadas que foram inicializados com os valores aleatórios ou transferidos do *ImageNet*. Logo, o mesmo foi realizado utilizando o otimizador “Adam”, uma variação da técnica de *Stochastic Gradient Descent* (SGD), um excelente algoritmo de

primeira ordem para otimização baseada em gradiente de funções objetivas estocásticas, tomando como base uma estimativa adaptada de momentos de baixa ordem (KINGMA; LEI BA, [S.d.]).

Foram utilizados, em cada experimento, a métrica de acurácia e a função de perda/custo *categorical cross-entropy (loss)* para avaliar os modelos. Essa função é própria para a determinação de perda/custo em classificação multiclasse.

Todos os modelos foram treinados com um total de 80 épocas e lote de tamanho 32. A taxa de aprendizado teve variações em cada modelo, como descrito no tópico 4.3.1.

4.3.4 Modelo baseado na AlexNet

Na proposta inicial realizaram-se os primeiros testes, utilizando a tecnologia baseada em redes neurais convolucionais. Assim, foi implementado um modelo para reconhecimento de deficiência nutricional, utilizando as folhas da *Brachiaria brizantha*. Essa implementação baseou-se no modelo AlexNet, que foi a vencedora do *Large Scale Visual Recognition Challenge (ILSVRC)*, em 2012.

A arquitetura foi projetada com um modelo sequencial (Figura 43), sendo que a primeira e a segunda camadas são de convolução com sessenta e quatro mapas de característica, com campos receptivos locais (*kernel*), dimensão de 3x3 e um input com o formato de 299x299 pixels e três canais de cores. São seguidas por uma camada de subamostragem (*MaxPooling*), usando filtro 2x2. A terceira e quarta são camadas convolucionais, com cento e vinte e oito mapas de características, campos receptivos locais com dimensão de 3x3. A quinta é uma camada convolucional com duzentos e cinquenta e seis mapas de característica, com campos receptivos locais com dimensão de 3x3, seguida por um *MaxPooling* com filtro 2x2.

As camadas de convolução ainda possuem alguns hiperparâmetros que devem ser configurados de forma adequada para melhorar o desempenho da rede, segue abaixo alguns deles:

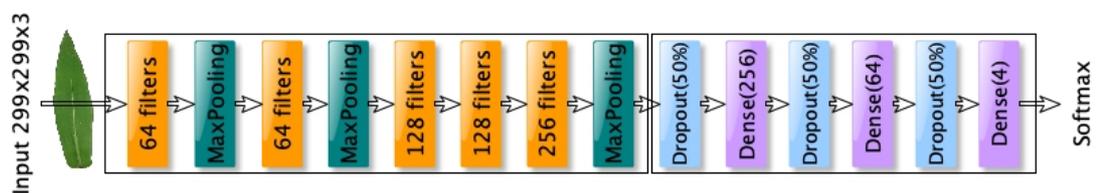
- **Função de Ativação:** foi utilizada a função de ativação do tipo ReLU, que é muito utilizada em projeto de reconhecimento de imagens;
- **Stride:** o passo em que as janelas passam realizando a convolução, foi utilizado passo (*Stride*) igual a 1;
- **Padding:** para manter a dimensionalidade da imagem foi implementado o *padding*, que nada mais é que o preenchimento das bordas da imagem com valores igual a 0;

Após as camadas convolucionais, foram implementadas camadas totalmente conectadas (*Full Connected* - FC). Nelas foram adicionadas 3 camadas. Na primeira, com 256 neurônios densamente conectada com a camada seguinte com 64 neurônios, a qual densamente conectada com a de saída (*output*) com 4 neurônios, que calcula a probabilidade da imagem pertencer à cada uma das quatro classes predeterminadas. Dessa forma, o maior valor é o considerado para predição. Esse modelo foi treinado e testado em oitenta épocas.

Nas camadas totalmente conectadas, além do número de neurônios, foram configurados basicamente os seguintes hiperparâmetros:

- **Função de Ativação:** foi utilizada a função de ativação do tipo ReLU, que é muito utilizada em projeto de reconhecimento de imagens e a função *Softmax* que é responsável pela classificação final;
- **Dropout:** Foi utilizado um dropout de 50%, que é responsável pela desativação de 50% dos neurônios na fase de treinamento de modo aleatório.

Figura 43 – Arquitetura em camadas do modelo proposto da CNN.



Fonte: autor da tese.

As várias camadas de convolução, implementadas neste modelo, geram muitas representações de dados e atuam como descritores de características, inicialmente nas primeiras camadas com informações mais comuns. Em seguida, houve o aumento e o detalhamento das características, conforme o número de camadas a serem aprofundadas (GEETHARAMANI; J., 2019)(LU et al., 2017c)(ZHANG, S. et al., 2019)(KOZLOWSKI; GÓRECKI; SZCZYPIŃSKI, 2019).

Além das camadas de convolução, tem-se *MaxPooling*, que é utilizada para diminuir a dimensionalidade da imagem para que seja possível um treinamento mais rápido com uma quantidade de parâmetros menor (SLADOJEVIC et al., 2016).

As camadas totalmente conectadas – FC, são responsáveis por realizar a classificação usando as saídas das camadas de convolução e de pooling. A última destas

camadas possui 4 saídas responsáveis pela realização da classificação utilizando uma função *Softmax* que normaliza exponencialmente a entrada recebida realizando a distribuição dos valores das 4 classes, ela transforma as saídas de cada classe para valores entre 0 e 1 e realiza a divisão pela soma das saídas. Permite, assim, a probabilidade de uma entrada estar em uma das saídas (MOHANTY et al., 2016). Nelas também foram adicionadas camadas de *dropout* com a função de reduzir o *overfitting* da rede, eliminando alguns neurônios da rede, definindo-os para zero (WANG et al., 2018).

5 RESULTADOS

Neste capítulo são apresentados os resultados obtidos com os seis modelos de CNNs desenvolvidos. Foram avaliados, inicialmente, os resultados dos modelos na fase de treinamento e validação, utilizando o conjunto de dados para este fim. Em seguida, foram realizados os testes finais dos modelos, avaliando sua capacidade de classificação com dados nunca antes apresentados. Após estas avaliações foi realizado um comparativo dos resultados de todos os modelos.

5.1 Resultados do modelo baseado na AlexNet

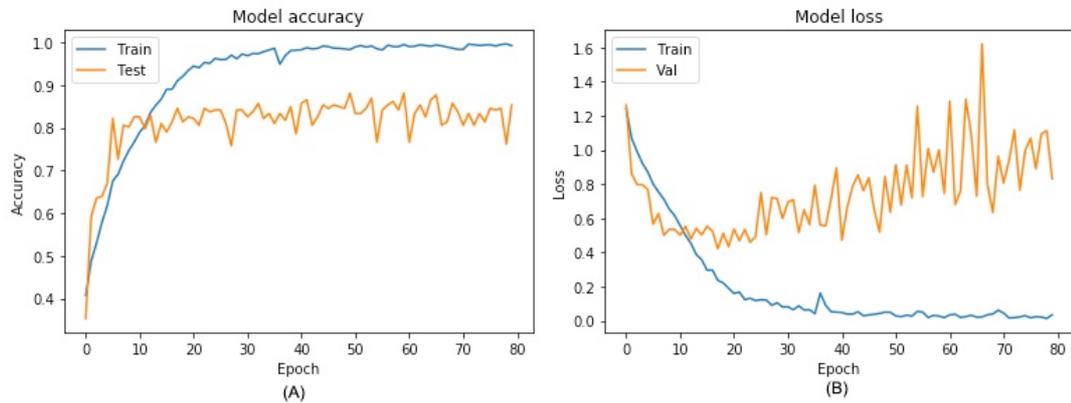
Nesta proposta, o resultado de acurácia foi de 77%, utilizando o conjunto de dados de teste. Todavia, ao se tratar de modelos de *Deep Learnig*, com as tecnologias disponíveis hoje no mercado, esses valores são considerados baixos e com potencial de melhorias com a utilização de outras técnicas e modelos disponíveis em visão computacional.

Essa rede ficou com o número de parâmetros total de 90.291.076, com um tempo de treinamento de 36 minutos e 12 segundos. O gráfico de acurácia (figura 44 A) mostra o modelo variando com as épocas em uma média entre 75% a 85%, tendo a função de parada selecionado na época 60 como a melhor acurácia encontrada pelo treinamento.

Ao analisar o gráfico de perda (*loss*) da figura 44 B, é possível verificar que tal modelo tende a sofrer de *overfitting*. Isso ocorre devido ao excesso de ajuste sobre os dados não conseguirem mais generalizá-los. Uma das maneiras de resolver essa problemática é realizar a parada do treinamento no momento em que o gráfico de *loss*, em validação, começar a aumentar seus valores. Isso também pode ocorrer pelo número limitado de amostras presentes no conjunto de dados de treinamento.

Após o treinamento, realizou-se o teste do modelo, utilizando um conjunto de dados nunca apresentado. Sendo assim, verificou-se o quanto ele conseguiu generalizar e foi possível constatar o desempenho em identificar o estado nutricional da cultura.

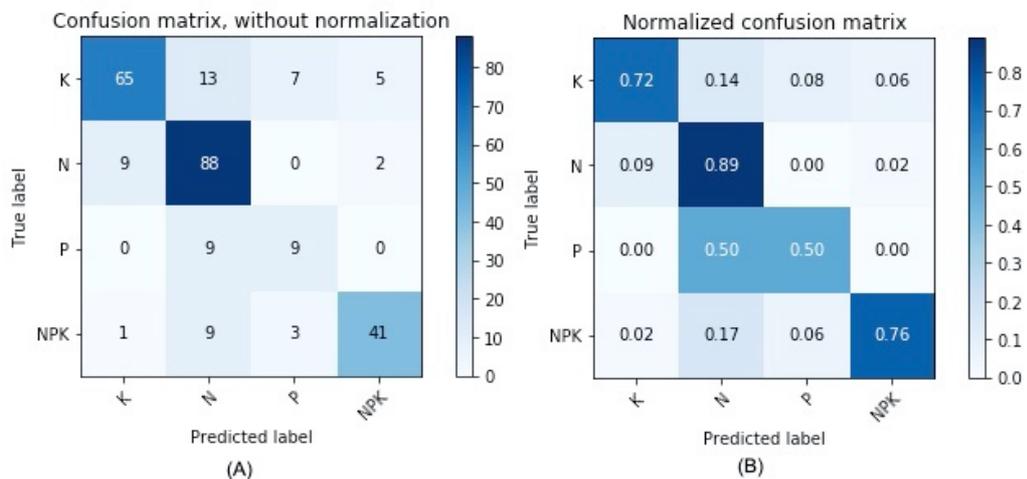
Figura 44 – Gráficos de acurácia(A) e gráfico de perda(B).



Fonte: autor da tese.

Na figura 45 é demonstrada a matriz de confusão baseada em dados de teste. Nessa etapa, tornou-se possível avaliar o quanto o modelo está generalizando, ou seja, o quanto ele conseguiu extrair características representativas das imagens apresentadas.

Figura 45 – Matriz de confusão de modelo utilizando dados de teste.



Fonte: autor da tese.

A matriz de confusão da figura 45 A, apresenta as quantidades das imagens analisadas e quantas foram avaliadas corretamente. Na figura 45 B, é a matriz normalizada demonstrando a proporção de positivos reais que foram identificados corretamente, que resultou nos valores em teste (GOOGLE DEVELOPERS, 2020) para cada classe de 72% para o Potássio, 89% para o Nitrogênio, 50% para o Fósforo, 76% para planta saudável com uma média de 72%. Esses valores também são os encontrados na métrica *Recall*.

Além do *Recall* pode-se utilizar a *Precision* como métrica de avaliação. Com isso é possível ter uma melhor análise dos resultados encontrados pelo modelo, avaliando quantas previsões efetuadas realmente eram corretas. No caso dos resultados em teste a precisão é de 87% para o Potássio, 74% para o Nitrogênio, 47% para o Fósforo e 85% para amostra saudável com a média de 73%.

As matrizes de confusão (figura 45), indicam que o Fósforo está abaixo dos outros nutrientes, isso demonstra a dificuldade do modelo em aprender as características de como identificá-lo. Um dos prováveis motivos para o menor desempenho, deve-se a quantidade de imagens da deficiência deste nutriente, pois, ele está um pouco abaixo das outras amostras e o modelo não conseguiu extrair as características para realizar a classificação de modo equivalente aos demais.

Analisando os nutrientes individualmente utilizando o *Recall* e o *Precision*, verificou-se que no caso do Fósforo, os valores estão muito abaixo dos demais nutrientes, sendo 47% e 50% respectivamente, comparando a acurácia do sistema que é 77,77%. Isso demonstra que o modelo pode mascarar o desempenho e estar aquém do ideal, por este motivo é importante utilizar além da acurácia outras métricas justificadas no texto.

Para o modelo ser considerado eficiente além da acurácia avalia-se um equilíbrio entre *Recall* e *Precision*. Para isso é possível utilizar a métrica *f1_score*, que combina as métricas buscando um equilíbrio entre elas. Com ela é possível avaliar o equilíbrio individualmente dessas métricas em cada uma das classes. Na tabela 6 é possível visualizar os valores encontrados para *Recall*, *Precision* e *f1_score*.

Tabela 6 – Resultados utilizando as métricas *Precision*, *Recall* e *F1_Score* para amostras de teste.

Nutriente	<i>Precision</i>	<i>Recall</i>	<i>F1_Score</i>	Amostras
Potássio	87%	72%	79%	90
Nitrogênio	74%	89%	81%	99
Fósforo	47%	50%	49%	18
Saudável	85%	76%	80%	54
Macro AVG	73%	72%	72%	261

Fonte: autor da tese.

Assim, como demonstrado anteriormente, esse modelo apresentou, aparentemente, uma eficiência satisfatória, mas existe um nutriente sem a mesma performance ficando com valores abaixo de 50%. Esse valor muito baixo acaba

inviabilizando a utilização deste modelo, por não trazer uma confiança aceitável na classificação.

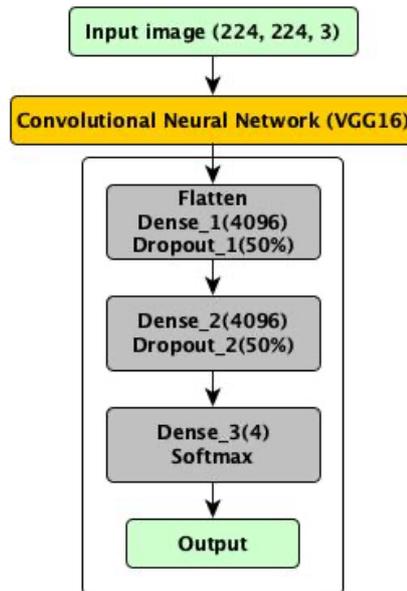
5.2 Modelo VGG-16 modificado

Após os testes iniciais com um modelo baseado na AlexNet, foi avaliada a possibilidade de usar outros modelos já implementados e disponíveis para utilização em sistemas de reconhecimento de imagens.

O primeiro modelo pré-treinado, utilizado na pesquisa, foi o VGG-16, vencedor da competição ILSVR (*ImageNet*), em 2014. Ele foi considerado com excelente arquitetura para visão computacional, até os dias de hoje. Possui alta capacidade de extrair *features* (características). Esse modelo utiliza uma rede de 16 camadas, sendo 13 camadas convolucionais e 3 camadas totalmente conectadas (FC) e uma camada *Softmax* de saída.

Inicialmente é necessário uma modificação, nesse modelo, que consiste na alteração das últimas camadas conhecidas como totalmente conectadas ou camadas densas, ilustrado na figura 46. No caso da VGG-16 padrão, este valor seria 1000 classes, mas para esse estudo foi considerado a identificação de apenas 4 classes (deficiências de Nitrogênio, Fósforo, Potássio e planta saudável). Dessa maneira, existiu a necessidade da alteração da camada de saída para 4, ou seja, o total de classes a serem classificadas. Outra alteração, realizada para melhorar o desempenho do modelo, foi a inserção de camadas de Dropout, melhorando a generalização.

Figura 46 – Arquitetura VGG-16 modificada.



Fonte: autor da tese.

Para implementação desse modelo, foi utilizada a técnica de *transfer learning*, baseada em um modelo pré-treinado da VGG-16. Isso significa que a rede já foi treinada anteriormente utilizando um conjunto de dados. Nesse caso, foram utilizados dados do *ImageNet*.

Nesse modelo foram realizados como testes iniciais o treinamento somente das últimas camadas, utilizando ajuste fino. Contudo, os resultados não foram satisfatórios, atingido valores muito baixos de reconhecimento. Por esse motivo, foi necessário realizar o treinamento de todas as camadas para alcançar bons resultados de reconhecimento, atingindo valores aceitáveis para implementação da aplicação. Vale afirmar que o treinamento em todas as camadas foi realizado a partir dos pesos já extraídos do *ImageNet*. Dessa forma, os valores iniciais de toda rede já estavam treinados com os pesos da base citada, acelerando o processo e atingindo melhores resultados.

Cabe lembrar que isso ocorreu devido a especificidade das imagens criadas para este projeto, tendo características próprias. Houve, também, a necessidade de um treinamento completo das camadas.

As configurações das camadas convolucionais foram as do modelo VGG-16, mas como comentado anteriormente, as camadas totalmente conectadas tiveram alterações. Elas foram configuradas da seguinte maneira: 2 camadas com 4096 neurônios e, sua última

camada, com 4 neurônios. São as responsáveis pela classificação e possuem basicamente os seguintes hiperparâmetros:

- **Função de Ativação:** utilizou-se a função de ativação do tipo ReLU;
- **Dropout:** usou-se um dropout de 50%, que é responsável pela desativação de 50% dos neurônios na fase de treinamento de modo aleatório;
- **SoftMax:** normaliza exponencialmente a entrada recebida, realizando a distribuição dos valores das 4 classes, da probabilidade de uma entrada estar em uma das saídas.

As imagens de entrada da rede foram redimensionadas para um tamanho fixo de 224x224 RGB padrão para o modelo VGG-16.

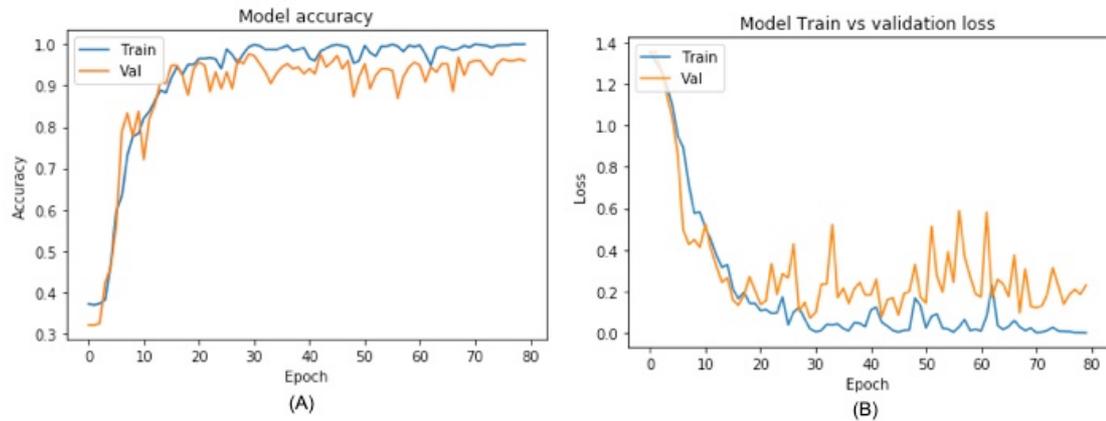
5.2.1 Resultados do modelo VGG-16

Os resultados da acurácia obtidos neste modelo foram de 97,61%, quando utilizado o conjunto de dados de validação e 96,93%, utilizando o de teste.

Nessa rede, com as alterações de suas camadas totalmente conectadas, o número de parâmetros foi alterado, chegando a um total de 134.276.923, com um tempo de treinamento de 24 minutos e 12 segundos. O tamanho desse modelo é uma desvantagem para a sua utilização, pois essa quantidade de parâmetros torna a rede mais lenta para ser treinada.

O gráfico da figura 47 B apresenta a perda (*loss*), que é indicativa do nível de aprendizado do modelo. Ele está demonstrando um possível *overfitting* (excesso de ajuste), a partir de época 50. Dessa forma, a melhor decisão é realizar a parada do treinamento, salvando o modelo com melhores resultados em validação, na época 28.

Figura 47 – Gráficos de acurácia e gráfico de perda.



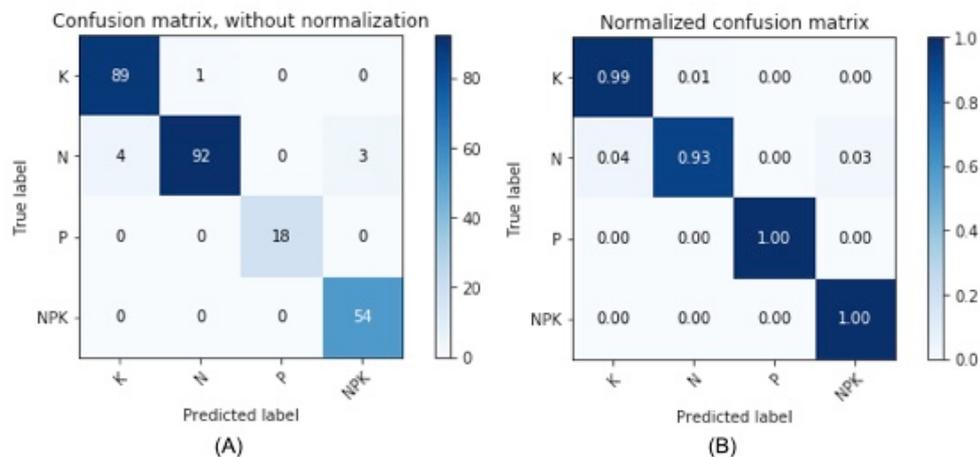
Fonte: autor da tese.

No gráfico de acurácia (figura 47 A), verificou-se o aprendizado do modelo em relação as épocas, possuindo um valor mais acelerado, aproximadamente até a época 18, com valores muito próximos do nível em treinamento, demonstrando que o modelo aprende muito bem com os exemplos e consegue atingir uma boa generalização.

Avaliou-se a eficiência com os dados de teste, totalizando 261 imagens, inicialmente utilizando a métrica da acurácia atingindo o valor de 96,93%. Isso demonstra o ótimo poder de generalização do modelo, conseguindo aprender muito bem as características das amostras de treinamento.

É possível verificar nas matrizes de confusão o comportamento do modelo (figura 48) trazendo os valores reais das amostras e os valores preditos pelo classificador. Nestas matrizes são identificados o estado nutricional diagnosticado de cada uma das amostras apresentadas.

Figura 48 – Matriz de confusão de modelo utilizando dados de teste.



Fonte: autor da tese.

A tabela 7 demonstra a existência um bom equilíbrio entre as métricas *Recall* e *Precision*, possuindo valores de *f1_score* muito bons. Isso comprova a qualidade do modelo avaliando as imagens nunca a ele apresentadas, no caso o conjunto de dados de teste. Analisando os números de *f1_score*, é possível verificar que o modelo está com um ótimo valor, chegando a 98%.

Tabela 7 – Resultados utilizando as métricas *Precision*, *Recall* e *F1_Score* para amostras de teste (Modelo VGG-16).

Nutriente	<i>Precision</i>	<i>Recall</i>	<i>F1_Score</i>	Amostras
Potássio	96%	99%	97%	90
Nitrogênio	99%	93%	96%	99
Fósforo	100%	100%	100%	18
Saudável	95%	100%	97%	54
Macro AVG	97%	98%	98%	261

Fonte: autor da tese.

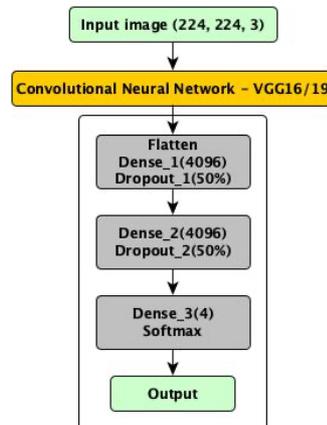
Isso comprova que o modelo está apresentando ótimos resultados de generalização. Logo, possui uma alta precisão de avaliação e um alto número de acerto das classes.

5.3 Modelo VGG-19 modificado

Para o modelo VGG-19 tem-se um aumento no número de camadas convolucionais em relação a VGG-16 descrito no tópico anterior. O aumento do número de camadas, muitas vezes, possibilita que o modelo alcance melhores resultados. No entanto, para o modelo VGG-19 em relação a rede VGG-16, com 3 camadas convolucionais a menos, isso não foi constatado.

Essas 3 camadas a mais deixam o modelo com 16 camadas convolucionais, 3 camadas totalmente conectadas e a camada de saída Softmax (figura 49).

Figura 49 – Arquitetura VGG-19 modificada.



Fonte: autor da tese.

Nesse modelo foi realizada a modificação das últimas camadas, conhecidas como camadas totalmente conectadas, alterando a quantidade de neurônios na camada de saída de 1000 para 4, sendo essa a quantidade de classes a serem identificadas. Além disso, foram adicionadas camadas de *dropout*, entre as camadas densas, para ajudar a melhorar a generalização.

Essa rede, com a alteração executadas, ficou com o número de parâmetros total de aproximadamente 139.586.628, com um tempo de treinamento de 28 minutos e 30 segundos. Pode-se verificar que quanto mais parâmetros a rede possuiu, maior foi o tamanho do modelo e mais tempo e recursos computacionais foram necessários para o treinamento.

O modelo, também, utilizou em suas camadas totalmente conectada (FC), a função de ativação ReLU, *dropout de 50%* e *Softmax* na camada de saída.

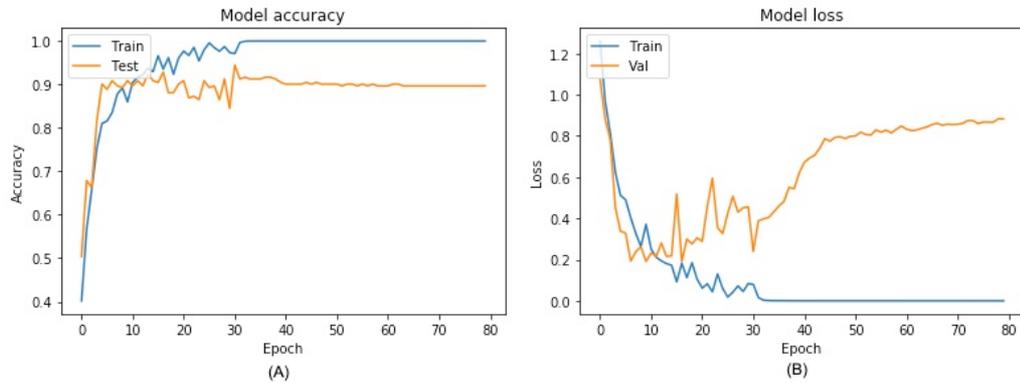
5.3.1 Resultados do modelo VGG-19

No modelo VGG-19 foi obtido um valor de acurácia em base de dados de teste de 86,20%, bem abaixo dos 96,93% alcançados pela VGG-16.

Isso demonstra que o aumento do número de camadas não implica necessariamente em melhoria do modelo, podendo até mesmo prejudicar gerando uma redução nos valores de classificação, inviabilizando sua utilização. O aumento do número de camadas pode levar a um problema já descrito neste trabalho, chamado de *vanishing/exploding* gradientes.

No gráfico A da figura 50, é apresentada a acurácia em treino e validação ao longo das épocas. Na época 5, os valores estão acima de 80% para validação e para treinamento e a partir da época 30 inicia uma estabilidade na acurácia.

Figura 50 – Gráficos de acurácia e gráfico de perda.



Fonte: autor da tese.

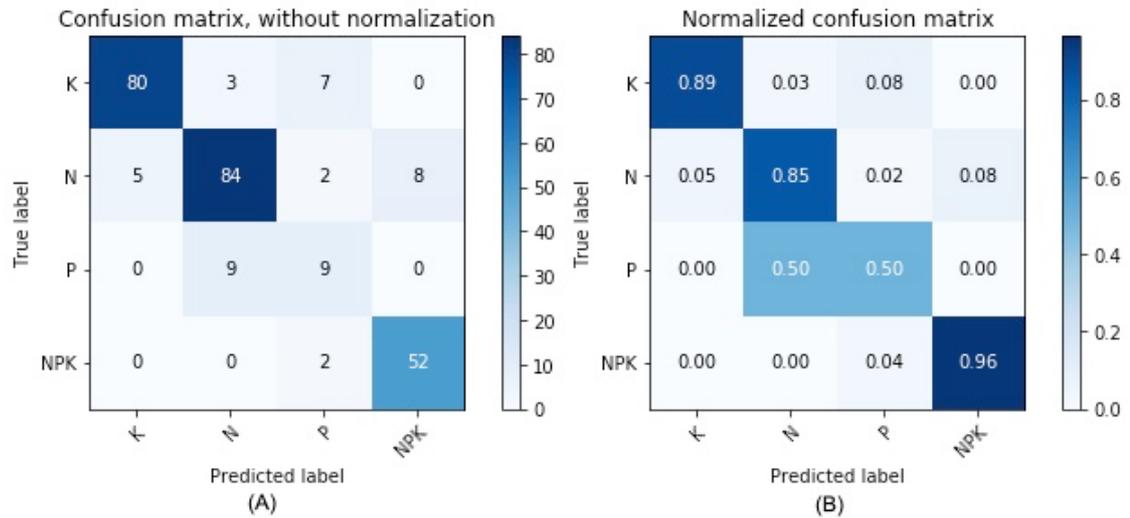
No gráfico B, da figura 50, é possível avaliar a perda (*loss*) ao longo das épocas. Nota-se, claramente, a partir da época 30, um aumento significativo na perda mostrando que o modelo está passando por um caso de *overfitting*. Isso demonstra a aprendizagem de forma excessiva das características das amostras no momento do treinamento e não está conseguindo mais generalizar os dados de validação. Para esse caso, o treinamento deverá ser parado antes do início do *overfitting*, ou se possível aumentar a quantidade de dados de treinamento.

No caso desse modelo, o melhor valor de acurácia em validação foi encontrado na época 31, onde atingiu o valor de 94,44%.

Para validar o modelo foi realizada a avaliação da performance com a base de dados de teste atingindo um valor de acurácia de 86,20%.

Na tabela 8 e figura 51, é possível verificar os valores de acurácia no conjunto de dados de teste não refletem nos valores preditos para o Fósforo, pois possui uma precisão e *recall* com os valores de 45% e 50%, respectivamente, e valores de *f1_score* de 47%. Esses valores encontrados de Fósforo ficam muito abaixo dos demais, com isso este modelo fica muito frágil para análise de NPK, pois um de seus nutrientes não traz segurança de classificação.

Figura 51 – Matriz de confusão de modelo utilizando dados de teste.



Fonte: autor da tese.

Tabela 8 – Resultados utilizando as métricas *Precision*, *Recall* e *F1_Score* para amostras de teste (Modelo VGG-19).

Nutriente	<i>Precision</i>	<i>Recall</i>	<i>F1_Score</i>	Amostras
Potássio	94%	89%	91%	90
Nitrogênio	88%	85%	86%	99
Fósforo	45%	50%	47%	18
Saudável	87%	96%	91%	54
Macro AVG	78%	80%	79%	261

Fonte: autor da tese.

Nesse modelo é possível verificar uma grande dificuldade em reconhecer o nutriente que possui a menor quantidade de amostras, assim o modelo não conseguiu aprender as características que descrevem tal deficiência. Como dito anteriormente, uma das alternativas para melhorar a acurácia deste modelo seria um aumento no número de amostras.

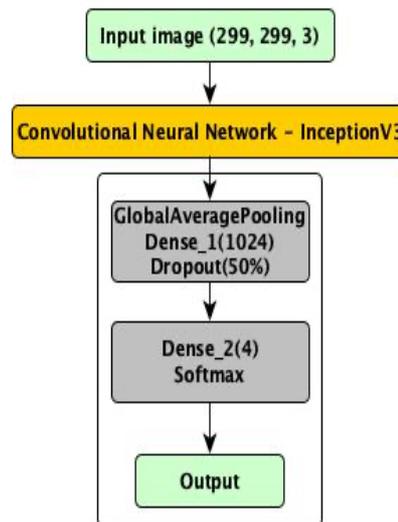
5.4 Modelo *Inception-V3* modificada

Outra abordagem utilizada foi o modelo *Inception-V3*, sendo um modelo menor que os descritos anteriormente e com resultados muito significativos no *ImageNet* e, também, na pesquisa em questão.

Nessa abordagem, foram necessárias alterações de suas camadas totalmente conectadas, deixando o modelo apto a classificar o estado nutricional da pastagem pesquisada.

As camadas totalmente conectadas foram implementadas com 1 camada densa de 1024 neurônios, com um *dropout* de 50%, seguida por uma camada com 4 neurônios para classificar o estado nutricional e a camada de saída *Softmax* como ilustrado na figura 52.

Figura 52 – Arquitetura Resnet-50 modificada.



Fonte: autor da tese.

Realizou-se o dimensionamento das imagens para 299x299x3 para atender o padrão de entrada exigido pelo modelo em questão.

Com a alteração da última camada, o modelo ficou com o número de parâmetros total de aproximadamente 21.810.980 e um tempo de treinamento de 24 minutos e 25 segundos.

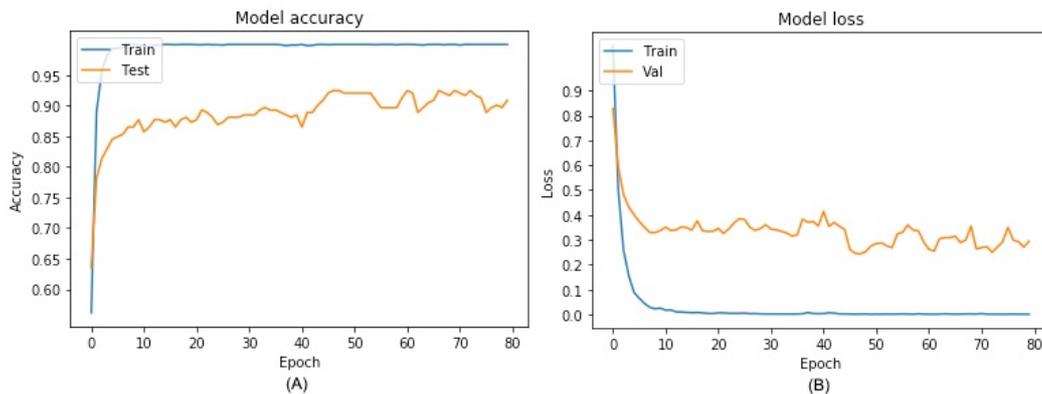
5.4.1 Resultados do modelo Inception-V3

Realizando o treinamento do modelo *inception-v3*, verifica-se nos gráficos da figura 53, a existência de pequenas oscilações na validação e se mantendo durante o processo de treinamento.

No gráfico de acurácia (figura 53 A) o modelo teve seu melhor resultado na época 67, nesse momento foi atingido uma acurácia em validação de 92,46% para base de dados de validação.

Ainda, analisando os gráficos de acurácia e de perda (*loss*) da figura 53, verifica-se que o modelo está apresentando uma tendência de aumento de acurácia e apresentando diminuição da perda (*loss*). Isto é um indicativo da existência de uma possível melhora do desempenho nos dados de treinamento e validação, podendo chegar a índices de acurácia melhores do que os já apresentados, simplesmente aumentando o número de iterações. Como todos os modelos testados utilizaram a mesma quantidade de épocas para os treinamentos, não foi possível avaliar se realmente esta tendência é verdadeira.

Figura 53 – Gráficos de acurácia e gráfico de perda.

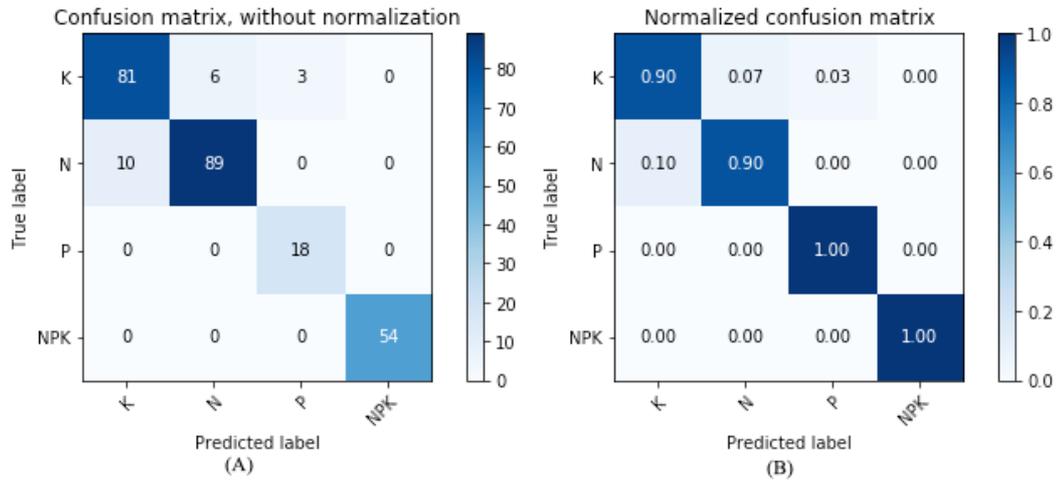


Fonte: autor da tese.

Para esse modelo, o valor de acurácia em base de dados de teste foi de 92,72%, índice muito significativo para esse tipo de aplicação.

Ao analisar a matriz de confusão na figura 54, é possível verificar a dificuldade apresentada pelo modelo para a detecção das deficiências do Potássio e Nitrogênio. Nelas, aparentemente, o sistema está confundindo as características aprendidas dos dois nutrientes. No entanto, ainda, os resultados são satisfatórios como mostra a tabela 9, apresentando uma adequada métrica geral para *Recall*, mas a *Precision* com valores mais baixos.

Figura 54 – Matriz de confusão de modelo utilizando dados de teste.



Fonte: autor da tese.

Tabela 9 – Resultados utilizando as métricas *Precision*, *Recall* e *F1_Score* para amostras de teste (modelo Inception-V3).

Nutriente	<i>Precision</i>	<i>Recall</i>	<i>F1_Score</i>	Amostras
Potássio	89%	90%	90%	90
Nitrogênio	94%	90%	92%	99
Fósforo	86%	100%	92%	18
Saudável	100%	100%	100%	54
Macro AVG	92%	95%	93%	261

Fonte: autor da tese.

Avaliando os valores de *f1_score*, o modelo possui níveis satisfatórios, obtendo índices acima de 90%, chegando a um valor geral de 93%.

Esse modelo atingiu valores de classificação satisfatórios, com uma grande possibilidade de melhorar seus resultados aumentando o conjunto de dados e o número de épocas, sendo uma opção viável para aplicações desta natureza.

5.5 Modelo ResNet-50 modificada

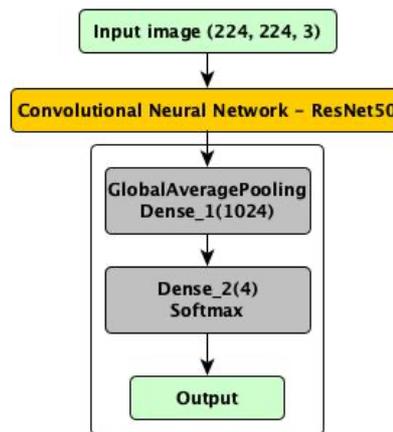
O modelo ResNet possuiu várias configurações em relação a sua arquitetura, que podem ser aplicadas, por exemplo, com 34, 50, 101 e 152 camadas.

Neste estudo foi implementada a arquitetura ResNet-50, um modelo muito utilizado para projetos de reconhecimento de imagens que possui resultados significativos e é uma arquitetura relativamente pequena em comparação às ResNet's maiores.

Para utilizá-lo, foi necessário realizar alterações em suas camadas totalmente conectadas, ficando com duas camadas densas. A primeira com 1024 neurônios, seguida por uma camada com 4 neurônios e uma camada de saída Softmax, como ilustrado na figura 55.

No conjunto de dados, foi necessário realizar o redimensionamento das imagens para atender o padrão da arquitetura ResNet, no formato de 224x224x3.

Figura 55 – Arquitetura Resnet-50 modificada.



Fonte: autor da tese.

Esse modelo, após as alterações realizadas, ficou com uma quantidade de parâmetros de 25.689.988 e um tempo de treinamento de 19 minutos e 50 segundos.

5.5.1 Resultados do modelo ResNet-50

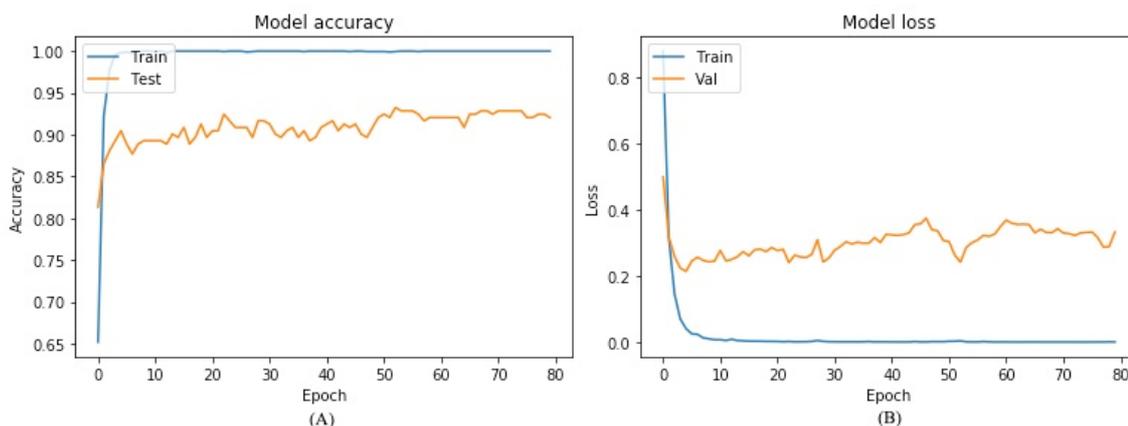
Aqui, utilizou-se uma taxa de aprendizado menor em relação aos anteriores. Isso ocorreu depois de testes realizados no treinamento apresentarem uma instabilidade. Foi necessário realizar a modificação da taxa de aprendizado para o modelo executar passos menores no treinamento e, com isso, melhorar sua performance.

Analisando o gráfico A da figura 56, a acurácia, já na primeira época, possui valores acima de 80%, tanto em treino como em validação, encaminhando para a estabilidade. Nesse treinamento, o modelo chegou ao seu melhor desempenho, na época 53, atingindo a acurácia de 93,25%.

Na avaliação da eficiência do modelo, utilizando a base de dados de teste, foi encontrado o valor de acurácia de 93,10%. Nos dados de validação e teste foram obtidos valores muito próximos. Isso demonstra que o modelo conseguiu extrair bem as

características no treinamento, possuindo uma boa generalização. De acordo com o gráfico de *loss* (figura 56 B), na época 60, há o início de um possível *overfitting*, demonstrando a necessidade da parada do treinamento antes que isso aconteça.

Figura 56 – Gráfico de acurácia e gráfico de perda.



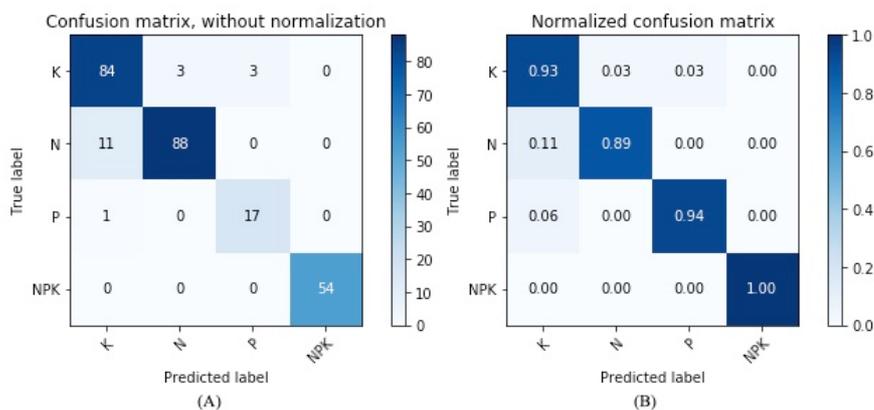
Fonte: autor da tese.

Na tabela 10 e figura 57, é possível verificar que para o Fósforo não foram encontrados valores ideais de precisão. Algumas amostras foram identificadas de forma equivocada, sendo confundida com Potássio. Já para a métrica *Recall*, esse nutriente apresentou um bom nível de acerto, chegando a 94%.

O mesmo acontece com o Potássio, com uma precisão de acerto de 88% e um *recall* de 93%, demonstrando níveis mais baixos de precisão, no caso, desse nutriente.

As amostras com deficiência de Nitrogênio e com nutrição completa, tiveram os melhores resultados, tanto para *Precision* como para *Recall*.

Figura 57 – Matriz de confusão de modelo utilizando dados de teste.



Fonte: autor da tese.

Tabela 10 – Resultados utilizando as métricas *Precision*, *Recall* e *F1_Score* para amostras de teste (modelo ResNet-50).

Nutriente	<i>Precision</i>	<i>Recall</i>	<i>F1_Score</i>	Amostras
Potássio	88%	93%	90%	90
Nitrogênio	97%	89%	93%	99
Fósforo	85%	94%	89%	18
Saudável	100%	100%	100%	54
Macro AVG	92%	94%	93%	261

Fonte: autor da tese.

Analisando individualmente os nutrientes avaliados, é possível verificar que o modelo está apresentando um pequeno desequilíbrio entre as métricas *Precision* e *Recall*. Ele possui uma boa acurácia, mas não tem uma precisão com os mesmos resultados, classificando algumas amostras falsas, como positivas.

Na avaliação geral da métrica *f1-score*, os resultados são satisfatórios, com um valor geral de 93% no conjunto de dados de teste

Na análise de todos os resultados apresentados pelo modelo, é possível dizer que o mesmo possuiu valores de classificação satisfatórios, tendo sua acurácia com valores acima dos 93%, o que demonstra sua qualidade.

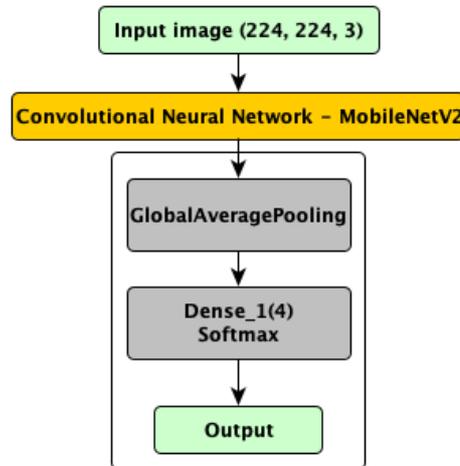
5.6 Modelo MobileNetV2 modificada

Nos dias atuais, com o aumento no uso de tecnologias no campo, principalmente a necessidade de atender pequenos produtores que não possuem de grandes recursos para a compra de equipamentos caros ou se encontram em áreas afastadas de grandes centros tecnológicos, torna-se indispensável, pensar em tecnologias mais baratas e de fácil acesso.

Ao pensar nisso, há a necessidade de tecnologias capazes de atender produtores de forma mais simples e com custos menores, mesmo resultando em níveis de precisão, um pouco inferiores a tecnologias mais robustas. Com o advento de dispositivos móveis, com poder computacional cada vez maiores e valores mais baixos, foram avaliados e realizados ensaios com um modelo possível de ser aplicado em tais tecnologias como, por exemplo, tablets e celulares.

Na busca de atender esse mercado de dispositivos móveis, foi utilizado um modelo de redes neurais convolucionais conhecido como MobileNetV2 (figura 58). Como descrito no item 3.14, é um modelo projetado para suporte à classificação e outras aplicações.

Figura 58 – Arquitetura MobileNetV2 modificada.



Fonte: autor da tese.

Para utilizar esse modelo, realizou-se a modificação da camada totalmente conectada (FC), alterando seu tamanho de saída para a quantidade de classes que será classificada, assim, foi alterada de 1000 para 4 neurônios.

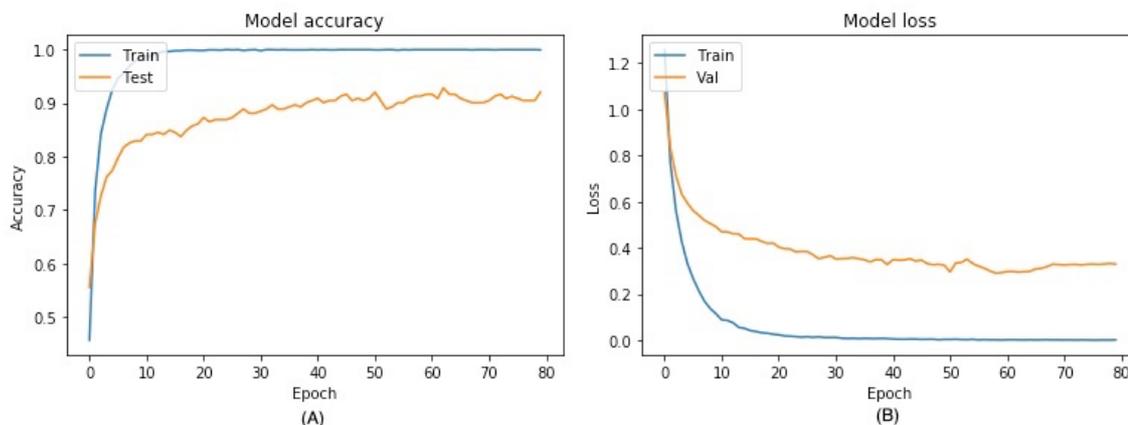
Essa rede com a alteração de sua última camada ficou com o número de parâmetros total de aproximadamente 2.263.108. Como possuiu poucos parâmetros em comparação a outros modelos como as redes VGG e *Inception*. Ele possui um tempo de treinamento de 13 minutos e 45 segundos, o menor de todos os modelos testados.

5.6.1 Resultados do modelo MobileNetV2

Esse modelo alcançou uma acurácia em validação de 92,85% e de 92,33%, utilizando o conjunto de dados de teste.

Ao observar e analisar o gráfico A da figura 59, é possível verificar que o modelo está aprendendo as características das amostras e, com isso, conseguiu ter uma boa generalização, classificando-as de forma satisfatória.

Figura 59 – Gráficos de acurácia e gráfico de perda.



Fonte: autor da tese.

O gráfico B da figura 59, demonstra a diminuição da perda durante o treinamento, demonstrando que o modelo está aprendendo sobre os dados. Isso também acontece nos dados de validação, obtendo resultados próximos as amostras de treinamento, indicando o nível de aprendizagem do modelo e, dessa forma, conseguindo generalizar.

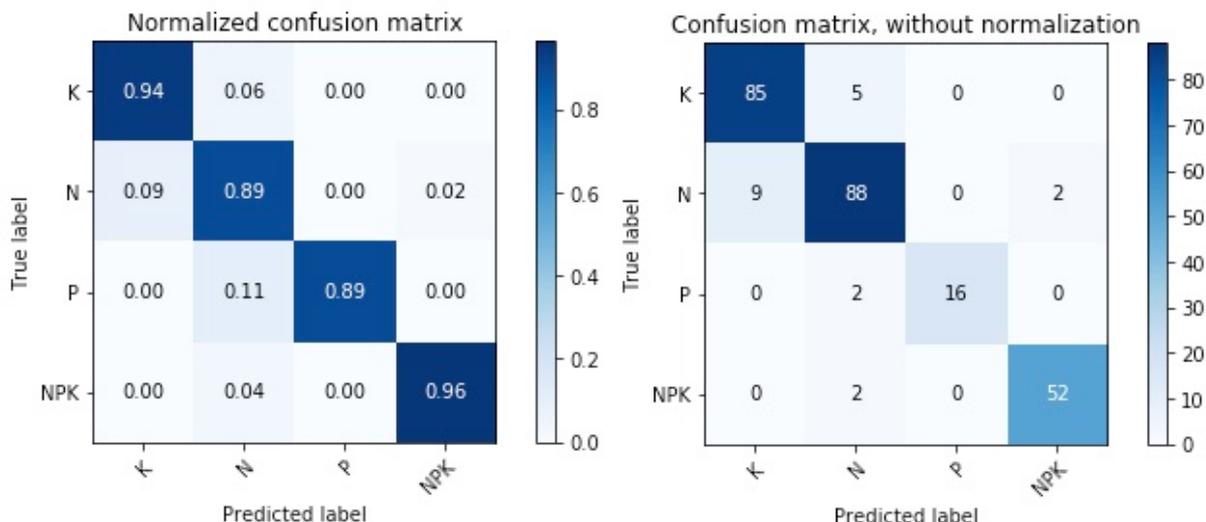
Na matriz de confusão na figura 60, verifica-se um equilíbrio no reconhecimento de todos os nutrientes, obtendo uma boa qualidade de classificação.

Já na tabela 11, é possível verificar no caso do nutriente Fósforo, seus níveis de precisão (*Precision*) com valores ótimos, atingindo 100% de reconhecimento em dados de teste, demonstrando que todas as amostras classificadas como Fósforo eram corretas. Já na métrica Recall atingiu 89% de acerto, o que significa que nem todas as amostras que eram Fósforo foram reconhecidas como tal. Isso demonstra que esse modelo é muito preciso para este nutriente, mas que não consegue classificar todas as amostras desta deficiência na mesma proporção.

No caso do Nitrogênio e planta saudável, tiveram um equilíbrio entre as métricas *Precision* e *Recall*. No primeiro caso, com *Precision* de 91% e *Recall* de 89%. No segundo, ficando com 96% em ambas as métricas.

Já o Potássio, também ficou com valores um pouco diferentes entre as duas métricas, ficando com 90% de *Precision* e 94% de *Recall*. Logo, mostra que no caso desse nutriente, o sistema tem menos precisão, mas conseguiu identificar mais amostras.

Figura 60 – Matriz de confusão de modelo utilizando dados de teste.



Fonte: autor da tese.

Na avaliação do modelo, utilizando a métrica *f1_score*, nota-se um equilíbrio ao ter um valor geral de 93% no conjunto de dados de teste. Individualmente pode-se verificar a existência de um pequeno desequilíbrio do Nitrogênio em relação aos outros nutrientes, mas isso não inviabiliza o modelo. Na tabela 11, pode-se verificar todos os resultados alcançados pelo modelo.

Tabela 11 – Resultados utilizando as métricas *Precision*, *Recall* e *F1_Score* para amostras de teste (Modelo MobileNetV2).

Nutriente	<i>Precision</i>	<i>Recall</i>	<i>F1_Score</i>	Amostras
Potássio	90%	94%	92%	90
Nitrogênio	91%	89%	90%	99
Fósforo	100%	89%	94%	18
Saudável	96%	96%	96%	54
Macro AVG	94%	92%	93%	261

Fonte: autor da tese.

Com os dados apresentados é possível verificar que esse modelo possuiu um nível de desempenho muito bom. Contudo, houve um pequeno desequilíbrio entre as métricas avaliadas, demonstrando valores aceitáveis de classificação. Com esse desempenho não existiu nenhuma avaliação de estado nutricional com níveis muito baixos, apresentando um bom nível de confiabilidade de classificação. Esses resultados foram obtidos, com uma quantidade muito menor de parâmetros. Cerca de 60 vezes o valor do maior modelo aqui

testado. Consequentemente isso exigiu um poder computacional menor que outros modelos, dando a ele uma vantagem muito importante.

5.7 Desempenho dos diversos modelos testados

Após a análise dos resultados dos modelos testados, foi elaborada a tabela 12. Ela apresenta como todos estão se comportando, tanto em dados de validação como em teste, além do número de parâmetros.

Na avaliação dos modelos baseado no AlexNet e modelo VGG-19, foram encontrados valores bem abaixo de outros testados neste estudo. No caso, os valores de acurácia abaixo do recomendado para aplicações em *Deep Learning* em comparação a modelos atuais. O primeiro modelo ficou 19,16% inferior e o segundo 10,73%. Todos em comparação ao melhor resultado encontrado.

Os modelos Resnet-50 e *Inception-V3*, obtiveram valores muito bons de acurácia atingindo, respectivamente, 92,72% e 93,10% em dados teste. Esses modelos, no geral, podem se tornar viáveis para serem aplicados a este tipo de avaliação, pois são modelos que utilizam menos parâmetros, tornando-os menores e mais rápidos para treinamento.

É possível que ao se aumentarem os dados de treinamento, tais modelos tenham um incremento em sua acurácia, ficando bem mais próximos ao modelo mais bem avaliado. No entanto, como as amostras eram limitadas a quantidade descrita em capítulos anteriores, não foi possível fazer essa avaliação. Mesmo assim, os modelos tiveram uma diferença de acurácia de 4,21% em relação VGG-16.

Para o modelo *Inception-V3*, o gráfico B da figura 54, demonstra a existência de uma possibilidade de melhora de resultados, pois é possível verificar uma tendência de queda nesse valor da acurácia, indicando que o modelo ainda está aprendendo em seu treinamento e possui espaço para melhorar o resultado de classificação.

Outro modelo que atingiu bons resultados é o MobileNetV2, mesmo com um tamanho reduzido, necessitando de menor poder computacional, atingiu uma acurácia de 92,33% no conjunto de dados de teste.

Tabela 12 – Resultados de acurácia para validação e teste para todos os modelos avaliados.

Classificador	Validação	Teste	Parâmetros
Modelo baseado na AlexNet	88,10%	77,77%	90.291.076
VGG-16	97,61%	96,93%	134.276.923
VGG-19	94,44%	86,20%	139.586.628
<i>Inception-V3</i>	92,46%	92,72%	23.905.060
Resnet-50	93,25%	93,10%	25.689.988
MobileNetV2	92,85%	92,33%	2.263.108

Fonte: autor da tese.

Já para o VGG-16 foram obtidos resultados de acurácia em dados de validação de 97,61% e teste de 96,93%. Em resumo, esse foi o melhor resultado, dentre todos os modelos testados. Esse modelo conseguiu extrair as melhores *features* (características) das imagens de treino e, por esse motivo, conseguiu a melhor generalização e os melhores resultados em teste.

Ao comparar os resultados obtidos com outros estudos, como por exemplo em TRAN et al. (2019), que utilizou uma rede neural convolucional conhecida como Inception-ResNet onde foi alcançado uma acurácia de 87,27%, avaliando Cálcio, Nitrogênio e Potássio, o modelo VGG-16 implementado nesta tese, conseguiu uma melhor acurácia atingindo 96,93% avaliando Nitrogênio, Fósforo e Potássio.

Já GHOSAL et al. (2018) utilizou uma rede neural convolucional para classificar as deficiências nutricionais de Ferro (Fe), Potássio (K), problemas de doenças e lesão química. Neste artigo foi alcançado uma acurácia geral de 94,13%, resultado um pouco abaixo dos 96,93% do estudo desta tese.

É importante salientar que os estudos de TRAN et al. (2019) e de GHOSAL et al. (2018) avaliam outras culturas.

6 CONCLUSÃO

Ao se considerarem os resultados apresentados, pode-se afirmar que a utilização do aprendizado profundo e CNN para a classificação do estado nutricional da *Brachiaria brizantha* cv. Marandu mostrou-se eficiente, atingindo valores capazes de confirmarem a técnica.

Para produzir os resultados, foram desenvolvidas muitas variações de arquiteturas de redes neurais artificiais com aprendizado profundo, aplicando CNN, tendo modelos iniciais implementados por completo e a utilização de modelos pré-treinados.

Por se tratar de modelos implementados e testados empiricamente, não se descartou testes exaustivos com diferentes combinações dos números das camadas e valores de seus hiperparâmetros. Foram esses testes que nortearam a busca por um modelo mais eficiente. Outrossim, por meio do estudo, contribuiu para a produção de modelos com bons resultados de classificação.

Foram implementados 6 modelos neste estudo: modelo desenvolvido com base na AlexNet e os modelos pré-treinados VGG-16, VGG-19, *Inception-V3*, Resnet-50 e MobileNetV2.

Entre todos os modelos, o que obteve melhores resultados de reconhecimento dos estados nutricionais foi o modelo VGG-16, alcançando uma acurácia em dados de teste 96,93%, sendo 99% para identificação da deficiência de Potássio, 93% para deficiência de Nitrogênio, 100% para deficiência de Fósforo e 100% em amostras saudáveis.

O modelo *Inception-V3* também obteve bons resultados, atingindo valores próximos ao VGG-16, mas com possibilidade de um incremento de sua performance, realizando alterações de hiperparâmetros e incremento do conjunto de dados. Outro grande destaque é seu número de parâmetros reduzidos em comparação ao outro modelo, conseqüentemente levando a uma redução de seu tamanho, o que é uma vantagem em relação ao modelo anterior.

Outro modelo com bons resultados é o MobileNetV2 que atingiu 92,33% em dados de teste. Este modelo se apresentou muito promissor, pois possuiu um número reduzido de parâmetros lavando-o a ter um menor custo computacional sem uma perda muito grande em sua acurácia. Com essas características podem ser aplicados em dispositivos móveis, sendo equipamentos que geralmente possuem pequeno poder computacional, mas acessíveis até para pequenos produtores rurais, beneficiando a região amazônica, principalmente o estado de Rondônia.

Essas tecnologias aplicadas as pastagens, com a precisão apresentada, demonstram que ajudarão o homem do campo a melhorá-las, realizando análises regulares sem a necessidade de avaliação laboratorial e de especialistas, que possuem altos custos e tempo para a realização. Além disso, as análises de solo laboratoriais são geralmente realizadas em alguns pontos da propriedade, porém, com a aplicação das tecnologias de inteligência artificial, pode-se realizar as análises em quantos pontos da propriedade se fizerem necessários sem aumentar significativamente os custos.

Acredita-se que os resultados podem ser melhores aumentando a quantidade de amostras que, apesar de não proibitiva, foi limitante. Pretende-se futuramente, produzir mais amostras desta pastagem para incrementar o conjunto de dados, realizando a criação de amostras tanto em casa de vegetação como em ambiente de produção e para que se consiga atender uma maior quantidade de áreas de cultivo. Considera-se a criação de outros conjuntos de dados com outras pastagens utilizadas na Amazônia, tudo isso como ação complementar a esta pesquisa.

Os resultados apresentados por esta tecnologia são promissores, com valores de reconhecimento ótimos para este tipo de aplicação, conferindo confiança em sua implementação na prática. Todos os modelos alcançaram resultados satisfatórios, no entanto, os que aplicaram *transfer learning* foram mais robustos, demonstrando o quanto a quantidade de dados influencia na qualidade do aprendizado.

7 TRABALHOS FUTUROS

Essa pesquisa estabelece bases sólidas para desenvolvimentos futuros. É possível sugerir novos trabalhos, como:

- 1) criação de novas amostras utilizando outros cultivares para avaliação de seu estado nutricional;
- 2) utilização da sobreposição de deficiências nutricionais;
- 3) adição da variável ataque de doenças
- 4) adição da avaliação de incidência de pragas;
- 5) utilização de imagens multiespectrais;
- 6) adição de imagens de aparelhos do tipo *Drones* ou câmeras integradas à equipamentos agrícolas, entre outras possibilidades.

REFERÊNCIAS BIBLIOGRÁFICAS

- ABRAFRIGO. Exportação Brasileira de Carnes Bovina e Derivados. p. 63, 2019.
- ACCOUNT, C.; SHARMA, A. *Convolutional Neural Networks with TensorFlow*. [S.l.]: Packt Publishing, 2018.
- AIMAGNIFI. *What is the difference between machine learning and deep learning*. Disponível em: <https://aimagnifi.com/blog/index.php/2017/10/13/what-is-the-difference-between-machine-learning-and-deep-learning/>. Acesso em: 24 jul. 2020.
- ALLA, S.; ADARI, S. K. *Beginning Anomaly Detection Using Python-Based Deep Learning With Keras and PyTorch*. California: Apress Media, LLC, 2019.
- ALVIM, M. J.; BOTREL, M. de A.; XAVIER, D. F. As principais espécies de Brachiaria utilizada no país. *Comunicado Técnico 22*, p. 3–6, 2002.
- AMARA, J.; BOUAZIZ, B.; ALGERGAWY, A. A Deep Learning-based Approach for Banana Leaf Diseases Classification. *Btw*, p. 79–88, 2017. Disponível em: <https://pdfs.semanticscholar.org/adae/9446cb66eaa6645dca78fd81b21d43aebdda.pdf>. Acesso em: 24 jun. 2019.
- ARNAL BARBEDO, J. G. Plant disease identification from individual lesions and spots using deep learning. *Biosystems Engineering*, v. 180, p. 96–107, 1 abr. 2019. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1537511018307797>. Acesso em: 19 ago. 2019.
- ARRIBAS, J. I.; SÁNCHEZ-FERRERO, G. V.; RUIZ-RUIZ, G.; GÓMEZ-GIL, J. Leaf classification in sunflower crops by computer vision and neural networks. *Computers and Electronics in Agriculture*, v. 78, n. 1, p. 9–18, 1 ago. 2011. Disponível em: <https://www.sciencedirect.com.ez67.periodicos.capes.gov.br/science/article/pii/S0168169911001220?via%3Dihub>. Acesso em: 19 maio 2018.
- BAESSO, M. M. Análise Do Status Do Nitrogênio No Feijoeiro Utilizando Imagens Digitais. 2005.
- BAI, K. *A Comprehensive Introduction to Different Types of Convolutions in Deep Learning*. Disponível em: <https://towardsdatascience.com/a-comprehensive-introduction-to-different-types-of-convolutions-in-deep-learning-669281e58215>. Acesso em: 21 set. 2020.
- BARBEDO, J. G. A. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Computers and Electronics in Agriculture*, v. 153, p. 46–53, 1 out. 2018. Disponível em: <https://www.sciencedirect.ez67.periodicos.capes.gov.br/science/article/pii/S0168169918304617>. Acesso em: 12 fev. 2019.
- BEN-DAVID, A. A lot of randomness is hiding in accuracy. *Engineering Applications of Artificial Intelligence*, v. 20, n. 7, p. 875–885, 1 out. 2007.
- BENGIO, Y. *Practical Recommendations for Gradient-Based Training of Deep*

Architectures. [S.l: s.n.], 2012. Disponível em: <http://deeplearning.net/software/pylearn2>. Acesso em: 17 jan. 2020.

BIANCO, S.; TONHÃO, M. A. R.; PITELLI, R. A. Crescimento e nutrição mineral de capim-braquiária. *Planta Daninha*, v. 23, n. 3, p. 423–428, set. 2005. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-83582005000300005&lng=pt&tlng=pt. Acesso em: 01 out. 2019.

BONFIM-SILVA, E. M.; MONTEIRO, F. A. Nitrogênio e enxofre na adubação e em folhas diagnósticas e raízes do capim-braquiária em degradação. *Revista Brasileira de Zootecnia*, v. 39, n. 8, p. 1641–1649, 2010.

BRÁSILIA, E.; HUGO, A.; MANUEL, Z.; MOTTA, C.; ARMINDO, M.; KICHEL, N.; GIOLO DE ALMEIDA, R. *Degradação, recuperação e renovação de pastagens*. [S.l: s.n.], 2012. Disponível em: <http://www.cnpqg.embrapa.br/publicacoes/doc/DOC189.pdf>. Acesso em: 14 ago. 2018.

CASAROLI, D.; DE JONG VAN LIER, Q. *CRITÉRIOS PARA DETERMINAÇÃO DA CAPACIDADE DE VASO (I)*. [S.l: s.n.], 2008.

CHEN, J.; LIU, Q.; GAO, L. Visual tea leaf disease recognition using a convolutional neural network model. *Symmetry*, v. 11, n. 3, p. 343, 7 mar. 2019. Disponível em: <https://www.mdpi.com/2073-8994/11/3/343>. Acesso em: 19 jul. 2019.

CHITYALA, R.; PUDIPEDDI, S. *Image Processing and Acquisition using Python*. [S.l.]: CRC Press, Taylor & Francis Group, 2014. 390 p. Disponível em: https://books.google.com/books?id=iJ_AAqAAQBAJ&pgis=1

COSTA, K. A. de P.; FAQUIN, V.; OLIVEIRA, I. P. De; RODRIGUES, C.; SEVERIANO, E. da C. Doses E Fontes De Nitrogênio Em Pastagem. *Revista Brasileira de Ciência do Solo*, v. 32, n. 1, p. 1601–1607, 2008. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-06832008000400024&lng=pt&nrm=iso&tlng=pt

COSTA, K. A. de P.; OLIVEIRA, I. P. De; FAQUIN, V. Adubação Nitrogenada para Pastagens do Gênero. *Documentos / Embrapa Arroz e Feijão*, v. 1, p. 60, 2006.

DIAS-FILHO, M. B. Diagnóstico das Pastagens. n. September 2014, , 2015.

DUCH, W.; JANKOWSKI, N. Survey of neural transfer functions. *Neural Computing Surveys*, v. 2, p. 163–212, 1999.

DYRMANN, M.; KARSTOFT, H.; MIDTIBY, H. S. Plant species classification using deep convolutional neural network. *Biosystems Engineering*, v. 151, p. 72–80, 1 nov. 2016. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1537511016301465>. Acesso em: 26 ago. 2019.

DYRMANN, M.; KARSTOFT, H.; SKOV, H. ScienceDirect Plant species classification using deep convolutional neural network. *Biosystems Engineering*, v. 151, n. 2005, p. 72–80, 2016. Disponível em: <http://dx.doi.org/10.1016/j.biosystemseng.2016.08.024>

FAQUIN, V. De Plantas No Agronegócio Diagnose Do Estado Nutricional Das Plantas. *Ufla/Faepe*, p. 1–75, 2002.

FAQUIN, V. Solos E Meio Ambiente Nutrição Mineral De Plantas. *Universidade Federal de Lavras-UFLA*, v. III, n. FAEPE, p. 186, 2005.

FERREIRA, P. V. *Estatística experimental aplicada às Ciências Agrárias*. 1ª edição, [S.l.]: EDITORA UFV, 2000. v. Livro. 550 p.

FOWLER, A. *NoSQL For Dummies*. New Jersey: John Wiley & Sons, Inc., 2015.

GEETHARAMANI, G.; J., A. P. Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Computers and Electrical Engineering*, v. 76, p. 323–338, 1 jun. 2019. Disponível em: <https://www-sciencedirect.ez67.periodicos.capes.gov.br/science/article/pii/S0045790619300023>. Acesso em: 12 jul. 2019.

GHOSAL, S.; BLYSTONE, D.; SINGH, A. K.; GANAPATHYSUBRAMANIAN, B.; SINGH, A.; SARKAR, S. An explainable deep machine vision framework for plant stress phenotyping. *Proceedings of the National Academy of Sciences of the United States of America*, v. 115, n. 18, p. 4613–4618, 1 maio. 2018. Disponível em: Fiss. Acesso em: 19 ago. 2019.

GIRARDIN, P.; TOLLENAAR, M.; MULDOON, J. F.; NNTIONAL, I.; RECHERCHE, D. Rate and Chlorophyll Content of Maize. v. 500, n. July, p. 491–500, 1985.

GLOROT, X.; BORDES, A.; BENGIO, Y. Deep Sparse Rectifier Neural Networks. *Proceedings of Machine Learning Research*, v. 15, , 2011.

GONZALEZ, R. C.; WOODS, R. E. *Processamento Digital de Imagens*. 3. ed., [S.l.]: Pearson Prentice Hall, 2010.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning (Adaptive Computation and Machine Learning series)*. [S.l.: s.n.], 2016. 800 p.

GOOGLE DEVELOPERS. *Machine Learning Crash Course*. Disponível em: <https://developers.google.com/machine-learning/crash-course/>. Acesso em: 26 fev. 2020.

GRIMM, J.; HERZOG, K.; RIST, F.; KICHERER, A.; TÖPFER, R.; STEINHAGE, V. An adaptable approach to automated visual detection of plant organs with applications in grapevine breeding. *Biosystems Engineering*, v. 183, p. 170–183, 1 jul. 2019. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1537511018311838>. Acesso em: 13 ago. 2019.

HAMUDA, E.; GLAVIN, M.; JONES, E. A survey of image processing techniques for plant extraction and segmentation in the field. *Computers and Electronics in Agriculture*, v. 125, p. 184–199, 1 jul. 2016. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168169916301557>. Acesso em: 11 maio 2018.

HAYKIN, S. *Redes Neurais: Princípios e Prática*. [S.l.]: Bookman, 2003. 898 p.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep Residual Learning for Image Recognition. In: 2016 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2016, p. 770–778. Disponível em: <http://ieeexplore.ieee.org/document/7780459/>. Acesso em: 10 set. 2019.

HOWARD, A. G.; ZHU, M.; CHEN, B.; KALENICHENKO, D.; WANG, W.; WEYAND, T.; ANDREETTO, M.; ADAM, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. 16 abr. 2017. Disponível em: <http://arxiv.org/abs/1704.04861>. Acesso em: 16 mar. 2020.

HU, G.; WU, H.; ZHANG, Y.; WAN, M. A low shot learning method for tea leaf's disease identification. *Computers and Electronics in Agriculture*, v. 163, , 1 ago. 2019.

HU, G.; YANG, X.; ZHANG, Y.; WAN, M. Identification of tea leaf diseases by using an improved deep convolutional neural network. *Sustainable Computing: Informatics and Systems*, v. 24, p. 100353, 1 dez. 2019.

IPNI. Manual Internacional de Fertilidade do Solo. *Manual internacional de fertilidade do solo*, v. 2, p. 89–117, 1998.

JI, M.; ZHANG, K.; WU, Q.; DENG, Z. Multi-label learning for crop leaf diseases recognition and severity estimation based on convolutional neural networks. *Soft Computing*, 30 mar. 2020. Disponível em: <http://link.springer.com/10.1007/s00500-020-04866-z>

KINGMA, D. P.; LEI BA, J. *ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION*. [S.l: s.n.], [S.d.]Disponível em: <https://arxiv.org/pdf/1412.6980v8.pdf>. Acesso em: 19 jul. 2019.

KOZŁOWSKI, M.; GÓRECKI, P.; SZCZYPIŃSKI, P. M. Varietal classification of barley by convolutional neural networks. *Biosystems Engineering*, v. 184, p. 155–165, 1 ago. 2019. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1537511018304021>. Acesso em: 08 ago. 2019.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. *ImageNet Classification with Deep Convolutional Neural Networks*. [S.l: s.n.], [S.d.]Disponível em: <http://code.google.com/p/cuda-convnet/>. Acesso em: 25 jun. 2019.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, v. 521, n. 7553, p. 436–444, 27 maio. 2015. Disponível em: <http://www.nature.com/articles/nature14539>. Acesso em: 02 jul. 2019.

LIU, B.; ZHANG, Y.; HE, D.; LI, Y.; LIU, B.; ZHANG, Y.; HE, D.; LI, Y. Identification of Apple Leaf Diseases Based on Deep Convolutional Neural Networks. *Symmetry*, v. 10, n. 1, p. 11, 29 dez. 2017. Disponível em: <http://www.mdpi.com/2073-8994/10/1/11>. Acesso em: 06 fev. 2019.

LU, Y.; YI, S.; ZENG, N.; LIU, Y.; ZHANG, Y. Identification of rice diseases using deep convolutional neural networks. *Neurocomputing*, 2017. a.

LU, Y.; YI, S.; ZENG, N.; LIU, Y.; ZHANG, Y. Identification of rice diseases using deep

convolutional neural networks. *Neurocomputing*, v. 267, p. 378–384, 6 dez. 2017. b.

LU, Y.; YI, S.; ZENG, N.; LIU, Y.; ZHANG, Y. Identification of rice diseases using deep convolutional neural networks. *Neurocomputing*, v. 267, p. 378–384, 6 dez. 2017. c.

Disponível em:

<https://www.sciencedirect.com/science/article/pii/S0925231217311384?via%3Dihub>.

Acesso em: 24 jun. 2019.

LUGER, G. F. *Inteligência Artificial*. 6. ed., São Paulo: Pearson Education do Brasil, 2013. 614 p.

MA, J.; DU, K.; ZHENG, F.; ZHANG, L.; GONG, Z.; SUN, Z. A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. *Computers and Electronics in Agriculture*, v. 154, p. 18–24, nov. 2018. a.

Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0168169918309360>. Acesso em: 14 fev. 2019.

MA, J.; DU, K.; ZHENG, F.; ZHANG, L.; GONG, Z.; SUN, Z. A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. *Computers and Electronics in Agriculture*, v. 154, p. 18–24, 1 nov. 2018. b.

Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168169918309360>. Acesso em: 14 fev. 2019.

MALAVOLTA, E. *Elementos de nutrição mineral de plantas*. São Paulo, SP: [s.n.], 1980.

MALAVOLTA, E. *Manual de nutrição mineral de plantas*. São Paulo, SP: [s.n.], 2006.

MALAVOLTA, E. *Manual de química agrícola*. São Paulo, SP: [s.n.], 1976.

MALAVOLTA, E.; ROMERO, P. J. *Manual de adubação*. 2ed., São Paulo: [s.n.], 1975.

MARCHESIN, W.; HENRIQUE, P.; LUZ, D. C.; HERLING, V. R. Guia de identificação de deficiências nutricionais em *Brachiaria brizantha* cv. marandu. *Comunicado Técnico 76*. [S.l: s.n.], 2007. .

MARIN, M. A. *Sistema de visão artificial para a diagnose nutricional de ferro , boro , zinco e cobre em plantas de milho Pirassununga-SP MÁRIO ANTONIO MARIN Sistema de visão artificial para a diagnose nutricional de ferro , boro , zinco e cobre em*. [S.l: s.n.], 2012.

MARTINEZ, H. E. P.; CARVALHO, J. G.; SOUZA, R. B. Recomendações para o uso de corretivos e fertilizantes em Minas Gerais. *UFV*, v. 5, p. 143–168, 1999.

MATTOS, W. T. DE; MONTEIRO, F. A. Respostas de braquiária *brizantha* a doses de potássio. p. 1–9, 2018.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, v. 5, n. 4, p. 115–133, dez. 1943.

Disponível em: <http://link.springer.com/10.1007/BF02478259>

MEHMET, ♂; ODABAS, S.; SENYER, N.; KAYHAN, G.; ERGUN, E. Estimation of Chlorophyll Concentration Index at Leaves using Artificial Neural Networks. 2016.

Disponível em: <https://www.worldscientific.com/doi/pdf/10.1142/S0218126617500268>. Acesso em: 20 maio 2018.

MICHELUCCI, U. *Advanced Applied Deep Learning*. Berkeley, CA: Apress, 2019. Disponível em: <http://link.springer.com/10.1007/978-1-4842-4976-5>

MOHANTY, S. P.; HUGHES, D. P.; SALATHÉ, M. Using Deep Learning for Image-Based Plant Disease Detection. *Frontiers in Plant Science*, v. 7, p. 1419, 22 set. 2016. Disponível em: <http://journal.frontiersin.org/article/10.3389/fpls.2016.01419/full>. Acesso em: 02 jul. 2019.

MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. *Foundations of Machine Learning*. Second edi, [S.l.]: The MIT Press, 2018.

MOOLAYIL, J. *Learn Keras for Deep Neural Networks*. Berkeley, CA: Apress, 2019. Disponível em: <http://link.springer.com/10.1007/978-1-4842-4240-7>

MOTTA, T. P. *Índices espectrais de imagens coloridas na avaliação do estado nutricional de fósforo, enxofre e magnésio no capim-marandu*. 2016. Universidade de São Paulo, 2016.

MUELLER, J. P.; MASSARON, L. *Deep Learning For Dummies*. Hoboken, New Jersey: John Wiley & Sons, Inc., 2019.

NISBET, R.; MINER, G.; YALE, K. *HANDBOOK OF STATISTICAL ANALYSIS AND DATA MINING APPLICATIONS*. Academic P, [S.l.]: Candice Janco, 2018.

NVIDIA. *TRAIN MODELS FASTER*. Disponível em: <https://developer.nvidia.com/cuda-zone>. Acesso em: 27 jan. 2020.

PASTAGEM.ORG. *Atlas Digital das Pastagens Brasileiras*. Disponível em: <https://pastagem.org/atlas/map>

PURKAIT, N. *Hands-On Neural Networks with Keras Design*. 1. ed., [S.l.]: Packt Publishing, 2019. 838 p.

ROMUALDO, L. M.; LUZ, P. H. C.; DEVECHIO, F. F. S.; MARIN, M. A.; ZÚÑIGA, A. M. G.; BRUNO, O. M.; HERLING, V. R. Use of artificial vision techniques for diagnostic of nitrogen nutritional status in maize plants. *Computers and Electronics in Agriculture*, v. 104, n. June, p. 63–70, 2014. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0168169914000696>

ROSENBLATT, F. *THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN I*. Psychological Review[S.l: s.n.], 1958.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, v. 323, n. 6088, p. 533–536, 9 out. 1986. Disponível em: <http://www.nature.com/doi/10.1038/323533a0>. Acesso em: 21 ago. 2018.

RUSSELL, S.; NORVIG, P. *Inteligência Artificial*. 3rd ed., Rio de Janeiro: Elsevier Editora Ltda, 2013.

SAHA, P. K.; UDUPA, J. K. Optimum image thresholding via class uncertainty and region homogeneity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 23, n. 7, p. 689–706, jul. 2001.

SALEH, H. *Machine Learning Fundamentals: Use Python and scikit-learn to get up and running with the hottest developments in machine learning (English Edition)*. 1. ed., [S.l.]: Packt Publishing, 2018. 240 p.

SANDLER, M.; HOWARD, A.; ZHU, M.; ZHMOGINOV, A.; CHEN, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. 12 jan. 2018. Disponível em: <http://arxiv.org/abs/1801.04381>

SEWAK, M.; KARIM, M. R.; PUJARI, P. *Practical Convolutional Neural Networks*. [S.l.]: Packt Publishing Ltd., 2018. .

SILVA, I. N. Da; SPATTI, D.; FLAUSINO, R. A. *Redes neurais artificiais: para engenharia e ciências aplicadas*. São Paulo: Artliber Editora Ltda, 2010.

SINGH, H. *Practical Machine Learning and Image Processing*. Allahabad, Uttar Pradesh, India: Apress, 2019.

SUH, H. K.; IJSSELMUIDEN, J.; HOFSTEE, J. W.; VAN HENTEN, E. J. Transfer learning for the classification of sugar beet and volunteer potato under field conditions. *Biosystems Engineering*, v. 174, p. 50–65, 1 out. 2018. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1537511016308777>. Acesso em: 12 ago. 2019.

SUN, J.; HE, X.; GE, X.; WU, X.; SHEN, J.; SONG, Y. Detection of key organs in tomato based on deep migration learning in a complex background. *Agriculture (Switzerland)*, v. 8, n. 12, p. 196, 11 dez. 2018. Disponível em: <http://www.mdpi.com/2077-0472/8/12/196>. Acesso em: 28 jan. 2020.

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCHE, V.; RABINOVICH, A. Going deeper with convolutions. In: PROCEEDINGS OF THE IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2015, v. 07-12-June-2015, p. 1–9.

SZEGEDY, C.; VANHOUCHE, V.; IOFFE, S.; SHLENS, J.; WOJNA, Z. Rethinking the Inception Architecture for Computer Vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, v. 2016-Decem, p. 2818–2826, 1 dez. 2015. Disponível em: <http://ieeexplore.ieee.org/document/7780677/>

TABORDA, J. M. M. Desenvolvimento da Pecuária Bovina no Estado de Rondônia: Contextualização Histórica e Indicadores Zootécnicos. *Dissertação (Mestrado em Desenvolvimento Regional) - Núcleo de Ciências Exatas e da Terra*, p. 73, 2015.

TM, P.; PRANATHI, A.; SAIASHRITHA, K.; CHITTARAGI, N. B.; KOOLAGUDI, S. G. Tomato Leaf Disease Detection Using Convolutional Neural Networks. In: 2018 11TH INTERNATIONAL CONFERENCE ON CONTEMPORARY COMPUTING, IC3 2018, 2018, p. 1–5. Disponível em: <https://ieeexplore.ieee.org/document/8530532/>. Acesso em: 19 jul. 2019.

TODA, Y.; OKURA, F. How Convolutional Neural Networks Diagnose Plant Disease. *Plant Phenomics*, v. 2019, p. 1–14, 26 mar. 2019. Disponível em: <https://spj.sciencemag.org/plantphenomics/2019/9237136/>. Acesso em: 15 ago. 2019.

TOO, E. C.; YUJIAN, L.; NJUKI, S.; YINGCHUN, L. A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, v. 161, p. 272–279, jun. 2019. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0168169917313303>

TOWNSEND, C.R.; COSTA, N. DE L.; MENDES, A.M.; PEREIRA, R. DE G.A.; MAGALHÃES, J. . Nutrientes Limitantes em Solo de Pastagens Degradadas de Brachiaria Brizantha Cv. Marandu em Porto Velho-RO. *Reunião Anual da Sociedade Brasileira de Zootecnia*, 2001.

TRAN, T.-T.; CHOI, J.-W.; LE, T.-T.; KIM, J.-W.; TRAN, T.-T.; CHOI, J.-W.; LE, T.-T. H.; KIM, J.-W. A Comparative Study of Deep CNN in Forecasting and Classifying the Macronutrient Deficiencies on Development of Tomato Plant. *Applied Sciences*, v. 9, n. 8, p. 1601, 17 abr. 2019. Disponível em: <https://www.mdpi.com/2076-3417/9/8/1601>. Acesso em: 24 jun. 2019.

VASILEV, I.; SLATER, D.; SPACAGNA, G.; ROELANTS, P.; ZOCCA, V. *Python Deep Learning*. Second Edi, BIRMINGHAM - MUMBAI: Published by Packt Publishing Ltd., 2019.

WANG, S.-H.; TANG, C.; SUN, J.; YANG, J.; HUANG, C.; PHILLIPS, P.; ZHANG, Y.-D. Multiple Sclerosis Identification by 14-Layer Convolutional Neural Network With Batch Normalization, Dropout, and Stochastic Pooling. *Frontiers in Neuroscience*, v. 12, p. 818, 8 nov. 2018. Disponível em: <https://www.frontiersin.org/article/10.3389/fnins.2018.00818/full>. Acesso em: 22 jul. 2019.

WILLIAM, M.; OLIVEIRA, S. Análise de textura em imagens de folha para diagnose nutricional precoce em culturas de milho. 2015.

XU, G.; ZHANG, F.; SHAH, S. G.; YE, Y.; MAO, H. Use of leaf color images to identify nitrogen and potassium deficient tomatoes. *Pattern Recognition Letters*, v. 32, n. 11, p. 1584–1590, 2011. Disponível em: <http://dx.doi.org/10.1016/j.patrec.2011.04.020>

ZAFAR, I.; TZANIDOU, G.; BURTON, R.; PATEL, N.; ARAUJO, L. *Hands-On Convolutional Neural Networks with TensorFlow*. [S.l.]: Packt Publishing Ltd., 2018a. 272 p.

ZAFAR, I.; TZANIDOU, G.; BURTON, R.; PATEL, N.; ARAUJO, L. Hands-On Convolutional Neural Networks with TensorFlow. *Hands-On Convolutional Neural Networks with TensorFlow*. [S.l.: s.n.], 2018b. .

ZEILER, M. D.; FERGUS, R. *LNCS 8689 - Visualizing and Understanding Convolutional Networks*. [S.l.: s.n.], 2014.

ZHANG, S.; HUANG, W.; ZHANG, C. Three-channel convolutional neural networks for vegetable leaf disease recognition. *Cognitive Systems Research*, v. 53, p. 31–41, 1 jan. 2019. Disponível em: <https://www->

sciencedirect.ez67.periodicos.capes.gov.br/science/article/pii/S1389041717303236.
Acesso em: 12 fev. 2019.

ZHANG, X.; QIAO, Y.; MENG, F.; FAN, C.; ZHANG, M. Identification of maize leaf diseases using improved deep convolutional neural networks. *IEEE Access*, v. 6, p. 30370–30377, 2018. a. Disponível em: <https://ieeexplore.ieee.org/document/8374024/>. Acesso em: 19 jul. 2019.

ZHANG, X.; QIAO, Y.; MENG, F.; FAN, C.; ZHANG, M. Identification of Maize Leaf Diseases Using Improved Deep Convolutional Neural Networks. *IEEE Access*, v. 6, p. 30370–30377, 2018. b. Disponível em: <https://ieeexplore.ieee.org/document/8374024/>. Acesso em: 06 fev. 2019.

ZHU, X.; ZHU, M.; REN, H. ScienceDirect Method of plant leaf recognition based on improved deep convolutional neural network Action editor : Ali Minai. *Cognitive Systems Research*, v. 52, p. 223–233, 2018. Disponível em: <https://doi.org/10.1016/j.cogsys.2018.06.008>

ZÚÑIGA, A. M. G. Sistema de visão artificial para identificação do estado nutricional de plantas. *Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP*, 2012.

GLOSSÁRIO

Blocos Casualizados – distribuição dos tratamentos às parcelas de forma casual, para se evitar que um determinado tratamento venha a ser beneficiado (ou prejudicado) por sucessivas repetições em parcelas melhores (ou piores).

Cloroplastos – é uma organela presente nas células das plantas e outros organismos fotossintetizadores, como as algas e alguns protistas. Possui clorofila, pigmento responsável pela sua cor verde. É um dos três tipos de plastos pigmentados, ou cromoplasto, sendo os outros dois os cromoplastos e os leucoplastos.

Clorofila – é a designação de um grupo de pigmentos fotossintéticos presente nos cloroplastos das plantas (em sentido geral, incluindo também as algas, cianofíceas e diversos protistas anteriormente considerados "algas" ou "plantas", como as algas vermelhas ou castanhas).

Cultivar – Uma planta cultivada.

Delineamento Experimental – é o plano utilizado para realizar o experimento. Esse plano implica na maneira como os diferentes tratamentos deverão ser distribuídos nas parcelas experimentais, e como serão analisados os dados a serem obtidos.

Experimento Ou Ensaio – é um trabalho previamente planejado, que segue determinados princípios básicos, no qual se faz a comparação dos efeitos dos tratamentos.

Fotossíntese – é um processo pelo qual ocorre a conversão da energia solar em energia química para realização da síntese de compostos orgânicos.

FORAGEIRAS – São as plantas usadas como fonte de alimento para os animais, podendo ser forrageiras gramíneas ou forrageiras leguminosas.

Granulométrica – é um estudo da distribuição das dimensões dos grãos de um solo.

Kernel – Um kernel é uma matriz, que é deslizada pela imagem e multiplicada com a entrada de forma que a saída seja aprimorada de uma certa maneira desejável.

Matéria Orgânica – é o conjunto de compostos químicos formados por moléculas **orgânicas** encontradas em ambientes naturais sendo eles terrestres ou aquáticos.

Pigmentos – são compostos químicos responsáveis pelas cores das plantas ou animais.

Perfilho – é a unidade básica de crescimento das gramíneas.

Tratamento – é uma denominação genérica, para designar qualquer método, elemento ou material, cujo efeito desejamos medir e comparar.

Unidade Experimental ou Parcela – é a unidade na qual o tratamento é aplicado. É na parcela que obtemos os dados que deverão refletir o efeito de cada tratamento nela aplicado.

INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES
Diretoria de Pesquisa, Desenvolvimento e Ensino
Av. Prof. Lineu Prestes, 2242 – Cidade Universitária CEP: 05508-000
Fone/Fax(0XX11) 3133-8908
SÃO PAULO – São Paulo – Brasil
<http://www.ipen.br>

**O IPEN é uma Autarquia vinculada à Secretaria de Desenvolvimento,
associada
à Universidade de São Paulo e gerida técnica e administrativamente pela
Comissão Nacional de Energia Nuclear, órgão do
Ministério da Ciência, Tecnologia, Inovações e Comunicações.**
