

**UNIVERSIDADE ESTADUAL DE PONTA GROSSA
SETOR DE CIÊNCIAS AGRÁRIAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA**

BRUNA NEVES MACHADO

**REDE NEURAL CONVOLUCIONAL E PADRÃO DE METADADOS NA
CLASSIFICAÇÃO DE GRÃOS DE SOJA**

PONTA GROSSA

2021

BRUNA NEVES MACHADO

**REDE NEURAL CONVOLUCIONAL E PADRÃO DE METADADOS NA
CLASSIFICAÇÃO DE GRÃOS DE SOJA**

Dissertação apresentada ao Programa de Pós-Graduação em Computação Aplicada, curso de Mestrado em Computação Aplicada da Universidade Estadual de Ponta Grossa, como requisito parcial para obtenção do título de Mestre.

Orientadora: Prof^ª. Dr^ª. Maria Salete Marcon Gomes Vaz

PONTA GROSSA

2021

M149 Machado, Bruna Neves
Rede neural convolucional e padrão de metadados na classificação de grãos de soja / Bruna Neves Machado. Ponta Grossa, 2021.
89 f.

Dissertação (Mestrado em Computação Aplicada - Área de Concentração: Computação para Tecnologias em Agricultura), Universidade Estadual de Ponta Grossa.

Orientadora: Profa. Dra. Maria Salete Marcon Gomes Vaz.

1. Metadados. 2. Padrão de metadados. 3. Rede neural convolucional. 4. Classificação de imagens. 5. Deep learning. I. Vaz, Maria Salete Marcon Gomes. II. Universidade Estadual de Ponta Grossa. Computação para Tecnologias em Agricultura. III.T.

CDD: 004



UNIVERSIDADE ESTADUAL DE PONTA GROSSA
Av. General Carlos Cavalcanti, 4748 - Bairro Uvaranas - CEP 84030-900 - Ponta Grossa - PR - <https://uepg.br>

TERMO

TERMO DE APROVAÇÃO

Bruna Neves Machado

REDE NEURAL CONVOLUCIONAL E PADRÃO DE METADADOS NA CLASSIFICAÇÃO DE GRÃOS

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre no Programa de Pós-Graduação em Computação Aplicada da Universidade Estadual de Ponta Grossa, pela seguinte banca examinadora:

Prof(a). Dra. Maria Salete Marcon Gomes Vaz(UEPG)

Prof. Dr. Alceu De Souza Britto Junior (UEPG)

Prof. Dr. Erickson Freitas de Moraes (UTFPR)



Documento assinado eletronicamente por **Alceu de Souza Britto Junior, Professor(a)**, em 25/03/2021, às 11:14, conforme art. 1º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **Maria Salete Marcon Gomes Vaz, Professor(a)**, em 25/03/2021, às 11:14, conforme art. 1º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **Erikson Freitas de Moraes, Usuário Externo**, em 25/03/2021, às 11:16, conforme art. 1º, III, "b", da Lei 11.419/2006.



A autenticidade do documento pode ser conferida no site <https://sei.uepg.br/autenticidade> informando o código verificador **0439884** e o código CRC **E64C1844**.

AGRADECIMENTOS

Agradeço à Deus, por estar presente em todos os momentos, me dando força e fé para a conclusão dessa dissertação. Pelas amizades que conquistei durante o período do curso.

Agradeço minha família por ter me apoiado e me incentivado a correr atrás dos meus sonhos. Especialmente minha vó Maria Inah, minha mãe Maria Rute e meu namorado Miguel Angelo, por sempre terem apoiado meus estudos e por terem acreditado em mim, apesar das dificuldades.

Agradeço minha orientadora Prof^a. Dr^a. Maria Salete Marcon Gomes Vaz, pelas oportunidades, incentivos e ensinamentos, nesse e em outros projetos.

Agradeço a todos os professores do programa que contribuíram com o meu amadurecimento como aluna, pesquisadora e pessoa. Especialmente o Prof. Dr. José Carlos Ferreira da Rocha e a Prof^a. Dr^a. Elaine Guimarães por todos os ensinamentos e conhecimentos compartilhados durante o curso.

Agradeço aos amigos e colegas acadêmicos que contribuíram com esse trabalho de diversas maneiras, ensinando-me e apoiando-me.

Agradeço ao Serviço Nacional de Aprendizagem Rural (SENAR) por ter contribuído com os conhecimentos agrônômicos práticos para o desenvolvimento da pesquisa.

Agradeço a Universidade Estadual de Ponta Grossa e ao Programa de Pós-graduação em Computação Aplicada, pela oportunidade e por disponibilizar toda a sua estrutura, equipamentos, para essa pesquisa.

Agradeço a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior pelo suporte financeiro.

E finalmente, a todos que contribuíram direta ou indiretamente para a realização dessa pesquisa.

“As nuvens mudam sempre de posição, mas são sempre nuvens no céu. Assim devemos ser todo dia, mutantes, porém leais com o que pensamos e sonhamos; lembre-se, tudo se desmancha no ar, menos os pensamentos”. (Paulo Beleki)

RESUMO

A classificação de grãos na agricultura é feita de forma manual, onde deve ser analisado grão a grão para que os grãos sejam alocados em suas respectivas classes. A identificação das classes dos grãos é feita visualmente, onde é feita a análise da cor e o estado em que o grão se encontra. Essa classificação é feita, geralmente, por profissionais especializados na área que podem demandar muito tempo para classificar uma pequena quantidade de grãos. O tempo para realizar a classificação manual é fundamental para que os grãos sejam avaliados e enviados para os consumidores o mais rápido possível. Neste trabalho foi apresentado um modelo de padrão de metadados para a classificação de grãos, um aplicativo para estruturar os metadados em HTML ou em XML, além de um método computacional para a classificação de grãos de soja da espécie *Glycine max* onde redes neurais convolucionais foram utilizadas. As redes neurais convolucionais utilizadas foram as *Resnet34*, *MobileNet* e *VGG19*. A rede foi treinada com base nos dados de defeitos de grão. Foram consideradas 9 (nove) classes de grãos de soja, onde cada classe continha 100 (cem) imagens. O resultado alcançado da *Resnet34* foi de 99,55% de acurácia com o uso de *data augmentation*, e 80,85% sem o uso dessa técnica. As redes *MobileNet* e *VGG19* obtiveram uma acurácia de 97,33% e 97,22% com o uso do *data augmentation* respectivamente.

Palavras-chave: Metadados; Padrão de Metadados; Rede Neural Convolucional; Classificação de Imagens; *Deep Learning*

ABSTRACT

The classification of grains in agriculture is done manually, where grain by grain must be analyzed so that the grains are allocated to their respective classes. The identification of the classes of the grains is done visually, where the analysis of the color and the state in which the grain is made. This classification is usually done by professionals specialized in the field, which can take a long time to classify a small amount of grains. The time to carry out the manual classification is essential for the grains to be evaluated and sent to consumers as soon as possible. In this work, a metadata standard model for grain classification was presented, an application to structure metadata in HTML or XML, as well as a computational method for the classification of soybeans of the *Glycine max* species where convolutional neural networks were used. The convolutional neural networks used were Resnet34, MobileNet and VGG19. The network was trained on the basis of grain defect data. Nine (9) classes of soybeans were considered, where each class contained 100 (one hundred) images. The result achieved by Resnet34 was 99.55% accurate with the use of data augmentation, and 80.85% without the use of this technique. The MobileNet and VGG19 networks achieved an accuracy of 97.33% and 97.22% with the use of data augmentation, respectively.

Keywords: Metadata; Metadata Standard; Convolutional Neural Network; Images Classification; Deep Learning

LISTA DE FIGURAS

Figura 1 – Alguns dos defeitos encontrados nos grãos de soja: (a) grão com o defeito esverdeado, (b) grão com o defeito germinado, (c) grão com o defeito mofado, (d) grão com o defeito ardido, (e) grão com o defeito fermentado, e (f) grão com o defeito imaturo.	15
Figura 2 – Grão de soja sem nenhum defeito.	15
Figura 3 – Ciclo dos objetos da informação.	21
Figura 4 – Etapas do processamento digital de imagens.....	31
Figura 5 – Uma imagem em escala de cinza e a convenção utilizada para o par de eixos (x,y).	33
Figura 6 – Imagem representada por matriz.	33
Figura 7 – Esquema em cubo de cores RGB.....	34
Figura 8 – (a) imagem em escala cinza; (b) histograma da imagem.....	35
Figura 9 – (a) Imagem na escala RGB; (b) histograma.....	36
Figura 10 – Imagem em processo de segmentação. (a) imagem original; (b) imagem com filtro; e (c) imagem segmentada.	37
Figura 11 – Extração de característica (área) de um grão de soja.....	38
Figura 12 – Uma imagem cinza com pontos de interesse detectados.....	39
Figura 13 – Imagem integral.....	40
Figura 14 – Esquema que mostra divisões na área de Inteligência Artificial.	42
Figura 15 – Arquitetura de uma Rede Neural Convolucional.	44
Figura 16 – Representação matricial de uma imagem RGB sendo convoluída com um filtro (kernel).....	46
Figura 17 – Exemplo de pooling de uma imagem 4x4.	47
Figura 18 – Arquitetura de um bloco residual.....	49
Figura 19 – Arquiteturas de redes neurais convolucionais: (a) Arquitetura de uma VGG-19; (b) Arquitetura de uma rede simples com 34 camadas e (c) Arquitetura de uma rede com 34 camadas em blocos.	50
Figura 20 – Evolução da produção de soja no Brasil de 1961 a 2018.	53
Figura 21 – Folhas de soja do PDDB: (a) peste bacteriana, (b) vírus do mosaico da soja, (c) fitotoxicidade do cobre, (d) podridão de carvão, (e) saudável, (f) folha cercospora, (g) míldio penugento, (h) oídio, (i) oídio e ferrugem, (j)	

ferrugem, (k) ferrugem e ponto, (l) mancha marrom, (m) praga do sul, (n) desconhecido.....	56
Figura 22 – Grãos de Café sobre um papel fosco para a classificação: (A) Grãos normais; (B) grãos quebrados e (C) impurezas.....	58
Figura 23 – Grãos de soja sobre EVA preto para a classificação.	61
Figura 24 – Imagens de soja com defeitos: (a) grão de soja ardido; (b) grão de soja imaturo e (c) grão de soja partido.....	62
Figura 25 – Script da criação das imagens com as transformações.	63
Figura 26 – Grão de soja ardido: (a) Imagem original. (b) Imagem criada com as transformações.	63
Figura 27 – Código em Python para o recorte e redimensionamento das imagens..	64
Figura 28 – (a) Tela inicial do aplicativo e (b) Tela de cadastro dos metadados.	70
Figura 29 – (a) Estrutura dos metadados em XML e (b) Estrutura dos metadados em HTML.	71
Figura 30 – Resultados da porcentagem de treino e validação do modelo. (a) Resultados com data augmentation. (b) Resultados sem data augmentation.	72
Figura 31 – Matriz de confusão de 225 imagens usadas no teste com a técnica de data augmentation.	74
Figura 32 – Matriz de confusão de 47 imagens usadas no teste sem a técnica de data augmentation.	77
Figura 33 – Grão de soja classificado erroneamente.	80
Figura 34 – Resultados da porcentagem de treino e validação do modelo. (a) Rede Neural VGG19. (b) Rede Neural MobileNet.	81
Figura 35 – Gráfico que ilustra os learning rate obtidos e suas perdas.	82
Figura 36 – Resultados da porcentagem de treino e validação do modelo com learning rate de 10^{-2}	83

LISTA DE QUADROS

Quadro 1 – Tipos de Metadados.	19
Quadro 2 – Princípios para aplicação dos metadados.	23
Quadro 3 – Elementos do DC em nível simples.	25
Quadro 4 – Elementos do DC em nível qualificado.	26
Quadro 5 – Defeitos dos grãos avariados da soja.	53
Quadro 6 – Metadados das características dos grãos	65
Quadro 7 – Metadados relacionados à região dos grãos.	68

LISTA DE TABELAS

Tabela 1 – Valores das cores no formato RGB.....	35
Tabela 2 – Valores dos defeitos em gramas encontradas nos grãos de soja.....	61
Tabela 3 – Número de imagens coletadas por classe.....	62
Tabela 4 – Resultados da rede neural com data augmentation.....	73
Tabela 5 – Resultados da rede neural sem data augmentation.....	73
Tabela 6 – Assertividade por classe com a técnica de data augmentation.....	74
Tabela 7 – Porcentagem incorreta por classes com a técnica de data augmentation.....	75
Tabela 8 – Porcentagem de acertos por classes com a técnica de data augmentation.....	76
Tabela 9 – Assertividade por classe sem a técnica de data augmentation.....	78
Tabela 10 – Porcentagem incorreta por classes sem a técnica de data augmentation.....	78
Tabela 11 – Porcentagem de acertos por classes sem a técnica de data augmentation.....	79
Tabela 12 – Comparação dos índices de assertividade com e sem o data augmentation.....	79
Tabela 13 – Redes Neurais treinadas e suas taxas de acerto.....	80
Tabela 14 – Valores dos learning rate utilizados e as acurácias obtidas.....	81
Tabela 15 – Resultado da rede neural com um learning rate de 10^{-2}	82

LISTA DE SIGLAS

ANSI	American National Standards Institute
CONAB	Companhia Nacional de Abastecimento
DC	Dublin Core
DCAT	Data Catalog Vocabulary
DCMI	Dublin Core Metadata Initiative
DL	Deep Learning
HTML	Hypertext Markup Language
MAPA	Ministério da Agricultura, Pecuária e Abastecimento
ML	Machine Learning
NERC	Conselho de Pesquisa Ambiental Natural
NISO	National Information Standards Organization
PDI	Processamento Digital de Imagens
PURE	Probability, Uncertainty and Risk in the Environment
RDF	Resource Description Framework
RGB	Red Green Blue
CNN	Rede Neural Convolucional
SGBD	Sistema de Gerenciamento de Banco de Dados
SIFT	Scale-Invariant Feature Transform
SURF	Speeded Up Robust Features
XHTML	eXtensible Hypertext Markup Language
XML	Extended Markup Language
WWW	Web World Wide

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 MOTIVAÇÃO.....	14
1.2 OBJETIVOS.....	16
1.2.1 Objetivo Geral.....	16
1.2.2 Objetivos Específicos.....	16
1.3 ESTRUTURA DA DISSERTAÇÃO.....	17
2 REVISÃO DE LITERATURA.....	18
2.1 METADADOS.....	18
2.2 PRINCÍPIOS GERAIS DE PADRÃO DE METADADOS.....	22
2.3 PADRÕES DE METADADOS.....	24
2.4 LINGUAGEM XML.....	29
2.5 LINGUAGEM HTML.....	30
2.6 PROCESSAMENTO DIGITAL DE IMAGENS.....	31
2.6.1 Imagens Digitais.....	32
2.6.2 Histograma.....	35
2.6.3 Segmentação.....	36
2.7 EXTRAÇÃO DE CARACTERÍSTICAS.....	38
2.8 MACHINE LEARNING.....	40
2.9 DEEP LEARNING.....	41
2.10 CONVOLUÇÃO.....	42
2.10.1 Definição da Convolução Discreta.....	43
2.10.2 Formulação Discreta.....	43
2.11 REDE NEURAL CONVOLUCIONAL.....	44
2.11.1 Camada Convolutacional.....	45

2.11.2 Camada de POOLING.....	47
2.11.3 Camada DROPOUT.....	48
2.11.4 Camada Totalmente Conectada.....	48
2.11.5 ResNet34.....	49
2.12 TENSORFLOW.....	51
2.13 FAST.AI.....	51
2.14 CULTURA DA SOJA.....	52
3 TRABALHOS CORRELATOS.....	55
4 MATERIAIS E MÉTODOS.....	60
4.1 AQUISIÇÃO DAS IMAGENS.....	60
4.2 SISTEMA COMPUTACIONAL.....	64
4.3 METADADOS.....	65
5 RESULTADOS E DISCUSSÃO.....	72
6 CONCLUSÕES E TRABALHOS FUTUROS.....	84
REFERÊNCIAS.....	86

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Com o aumento do desenvolvimento de aplicações para manipular dados, é necessário o uso de ferramentas específicas e padrões para obter informações rápidas e precisas. É através da definição de metadados de um objeto de estudo que é possível identificar, descrever, explicar, localizar, facilitar a recuperação e o uso de dados (MANOUSELIS et al., 2010; NISO, 2017; STEINACKER; GHAVAM; STEINMETZ, 2001). Empresas que gerenciam grande número de dados necessitam realizar tomadas de decisões para um determinado problema, analisando informações utilizando padrão de metadados.

Com o uso de um padrão de metadados, a gerência das informações é facilitada na identificação, recuperação e, principalmente, na busca. A falta de uma padronização para organizar os dados afeta negativamente na manipulação dos mesmos. Podemos citar como exemplos, a dificuldade de recuperação da informação, informações importantes não sendo registradas, entre outros.

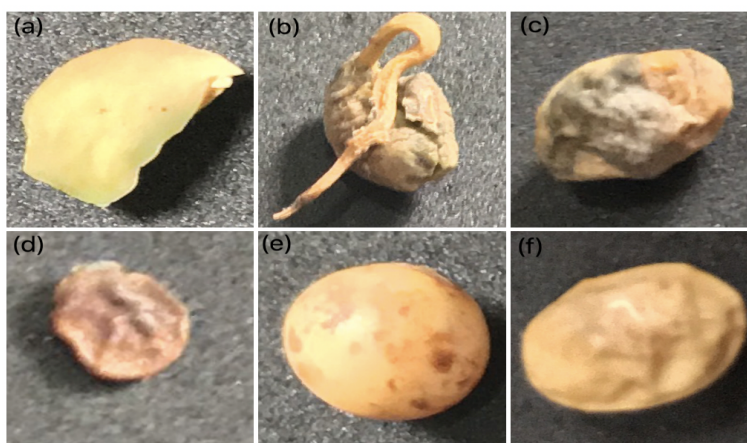
Além de utilizar um padrão de metadados para fazer a gerência dos dados, é muito importante adotar técnicas computacionais para capturar informações. O Processamento digital de imagens (PDI), é utilizado para obter informações de uma imagem em análise além de fornecer tratamentos para a melhoria da qualidade das mesmas. A adoção de PDI analisa diferentes abordagens, sendo utilizada para analisar a qualidade de diversas culturas e com diferentes abordagens, como por exemplo, o desenvolvimento de um sistema de monitoramento para a classificação de grãos de café (PINTO et al., 2017).

Devido à quantidade de informação e com o aprimoramento de processos computacionais, a classificação de imagens tem aumentado (GERON, 2019). A classificação de grãos na agricultura é realizada, geralmente, por profissionais especializados na área, gerando uma dependência desses profissionais. Como parte do processo para a classificação dos grãos, é necessário analisar grão a grão para que sejam classificados em suas devidas classes, sendo essa etapa um processo demorado, podendo levar aproximadamente 2 (duas) horas para classificar 200 gramas de grãos. Além disso, é um trabalho exaustivo, pois o classificador fica muito

tempo em uma posição, podendo causar problemas de saúde e até mesmo afetar o resultado.

A identificação correta sobre as classes que os grãos pertencem é de vital importância, seja na pesquisa, no controle dos danos, na ingestão humana e de animais. A Figura 1 mostra alguns dos defeitos encontrados nos grãos de soja.

Figura 1 – Alguns dos defeitos encontrados nos grãos de soja: (a) grão com o defeitos esverdeado, (b) grão com o defeito germinado, (c) grão com o defeito mofado, (d) grão com o defeito ardido, (e) grão com o defeito fermentado, e (f) grão com o defeito imaturo.



Fonte: A autora

A soja da espécie *Glycine max*, como mostrada na Figura 2, possui uma cor amarelada com formato oval, entretanto a forma e a cor podem variar conforme a espécie de soja analisada.

Figura 2 – Grão de soja sem nenhum defeito.



Fonte: A autora

Analisando esses aspectos, um modelo de padrão de metadados e um método computacional para a classificação de grãos de soja podem ser ferramentas usadas pelos produtores rurais ou qualquer pessoa que tenha interesse na classificação de grãos.

Este trabalho teve como objetivo a definição de metadados para classificação de grãos, a criação de uma base de dados rotuladas com imagens de grãos de soja e o desenvolvimento de um método convolucional para a classificação de grãos de soja.

Com esse trabalho, muitos profissionais da área poderão utilizar essa ferramenta para auxiliar na classificação de grãos, tornando o trabalho mais rápido e menos exaustivo. Portanto técnicas de processamento de imagens e redes neurais (métodos computacionais) podem ser empregadas para a identificação dos variados defeitos encontrados nos grãos de soja.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

O objetivo geral desta dissertação é desenvolver um método convolucional para classificação automática de grãos de soja.

1.2.2 Objetivos Específicos

Como objetivos específicos, têm-se:

- Compilar uma base de dados de imagens de defeitos dos grãos de soja;
- Definir o padrão de metadados sobre os grãos;
- Desenvolver um aplicativo para gerar os metadados para a classificação dos grãos;
- Definir um XML e HTML para a padronização da estrutura dos metadados;
- Implementar redes neurais convolucionais para a classificação de grãos de soja;
- Avaliar a acurácia encontrada com os algoritmos das redes neurais convolucionais.

1.3 ESTRUTURA DA DISSERTAÇÃO

Este trabalho está dividido em mais 5 (cinco) capítulos, além deste introdutório. O segundo capítulo trata sobre os fundamentos e conceitos inerentes às técnicas de processamento digital de imagens, metadados e redes neurais convolucionais. No capítulo seguinte, trata sobre os trabalhos correlatos na área de metadados e redes neurais. No Capítulo 4 é descrita a metodologia e os materiais utilizados para o desenvolvimento desta dissertação. O Capítulo 5 aborda os resultados obtidos e discussões. E, finalmente, no Capítulo 6 são apresentadas as conclusões e perspectivas de trabalhos futuros.

2 REVISÃO DE LITERATURA

2.1 METADADOS

Cada vez mais, a tecnologia de metadados está sendo usada como uma ferramenta de gerenciamento e análises dos dados e informações. Através das análises sobre os dados é gerado conhecimento e inteligência para tomadas de decisões nas organizações. Segundo Prusak (1997), inteligência e conhecimento estão conectados, sendo que a inteligência é necessária para a geração do conhecimento, onde o mesmo fornece a base a qual a inteligência pode ser aplicada.

Devido ao fato do desenvolvimento da *World Wide Web (WWW)*, o número de dados presentes na internet levam a adoção de ferramentas que facilitem a recuperação e gerenciamento dos dados armazenados em recursos eletrônicos. Para que seja feita a busca sobre um assunto em documentos de forma mais rápida e precisa, metadados são utilizados. Inicialmente, os metadados foram criados como uma marca comercial, e com o tempo, foi se desenvolvendo para descrever recursos de informações, passando a ser utilizados em várias áreas do conhecimento.

Primeiramente, o termo metadado começou a surgir na literatura de sistemas de gestão de base de dados (SGBD) com o objetivo de descrever os dados contidos nas bases de dados. A partir da década de 1990, os metadados foram utilizados em áreas específicas, como na área de dados geoespaciais, podendo ser considerada uma das primeiras aplicações dos metadados.

Segundo Méndez Rodriguez (2002), os metadados estão sendo utilizados como descrições por eles mesmos, apresentando um significado reduzido e vago quando não contextualizados em suas aplicações, domínio e conteúdo. Vale ressaltar que, os metadados podem ser considerados apenas dados quando utilizados para descrever eles mesmos.

Existem vários conceitos a respeito do termo metadados, mas a definição mais clássica e comum encontrada é "dados sobre dados" citada por Madsen, Foog e Ruggles (1994). Eles fornecem informações sobre um determinado objeto de estudo, facilitando a interoperabilidade, identificação, compartilhamento, integração, utilização, reutilização, gerenciamento e recuperação de maneira mais eficiente.

Segundo Nascimento (2008), os metadados tem como objetivos:

- Auxiliar na recuperação de informações e dados;
- Estabelecer comunicação consistente entre banco de dados;
- Garantir o registro de informações relevantes;
- Facilitar o uso e compartilhamento de informações e conhecimentos, para humanos e máquinas; e
- Auxiliar a troca de informações entre diferentes sistemas e banco de dados.

Assim, analisando as definições dos metadados, pode-se concluir que os metadados estão vinculados a uma estrutura que segue um padrão, com o objetivo de representar uma aplicação de um domínio específico, seja em dados de bibliotecas, da internet, entre outros. De maneira geral, os metadados são dados estruturados e padronizados que dão significado a uma informação, facilitando a sua identificação para localização, busca e recuperação em um sistema (ALVES, 2005).

Tendo em vista a grande quantidade de dados gerados em diferentes plataformas, há a necessidade de separar os metadados em diferentes tipos.

De acordo com Gilliland-Swetland (1998), as bibliotecas vêm compartilhando metadados descritivos em sistemas informatizados, desde os anos de 1960, na qual utilizavam normas e estruturas de descrição, tal como o formato MARC. Eles também apontam que os metadados podem ser divididos em 5 (cinco) tipos (Quadro 1), em função da sua funcionalidade nos meios eletrônicos.

Quadro 1 – Tipos de Metadados.

TIPO	DEFINIÇÃO
Administrativos	Usados na gestão e administração de recursos de informação.
Descritivos	Usados para descrever informação sobre recursos
De Conservação	Relacionados com a conservação de recursos de informação.
Técnicos	Relacionados com o funcionamento dos sistemas e os comportamentos dos metadados.
De uso	Relacionados com o nível e o tipo do uso dos recursos de informação.

FONTE: (GILLILAND-SWETLAND, 1998)

Segundo Baca (2008), os tipos de metadados possuem funções que podem ser executadas em um sistema de informação digital:

- Administrativos - Informações de aquisição, rastreamento de direitos e reprodução, documentação dos requisitos legais de acesso, informações de localização, critérios de seleção para digitalização.
- Descritivos - Registros de Catalogação, encontrar auxílios, diferenciações entre versões, índices especializados, informação curatorial, relações com hiperlink entre recursos, anotações de criadores e usuários.
- De Conservação - Documentação da condição física dos recursos, documentação de ações tomadas para preservar versões físicas e digitais de recursos, por exemplo, atualização e migração de dados, documentação de quaisquer alterações que ocorram durante a digitalização ou preservação.
- Técnicos - Documentação de hardware e software, informações técnicas de digitalização, por exemplo, formatos, taxas de compressão, rotinas de dimensionamento, rastreamento dos tempos de resposta do sistema, dados de autenticação e segurança, por exemplo, chaves de criptografia, senhas.
- De uso - Registros de circulação, registros de exposições físicas e digitais, rastreamento de uso e usuário, reutilização de conteúdo e informações de várias versões, logs de pesquisa, metadados de direitos.

Também é definido atributos chaves para os metadados de acordo com Gilliland-Swetland (1998):

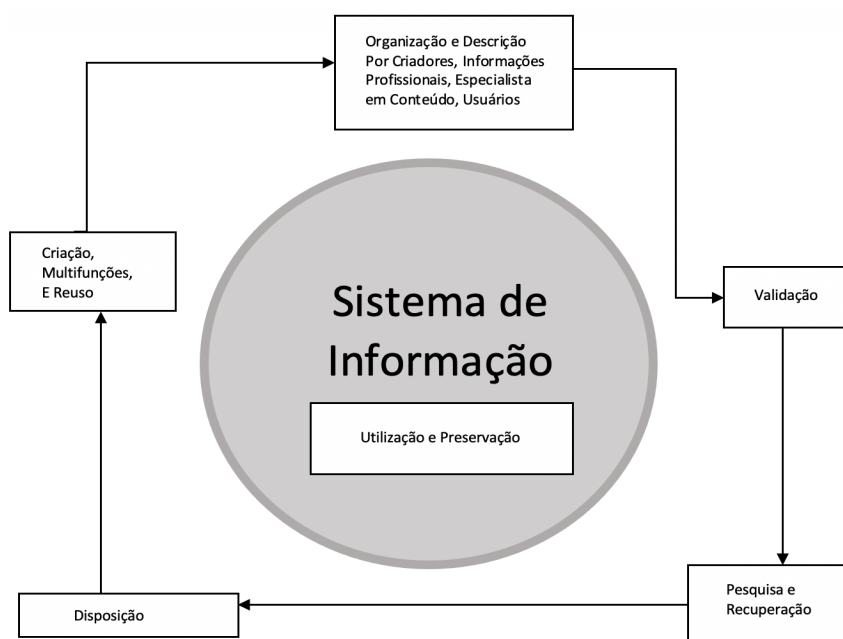
- Nível: metadados de coleções de itens ou objetos individuais.
- Semântica: metadados que seguem uma regra e que estão catalogados em um vocabulário padrão.
- Estrutura: metadados que adotam uma estrutura padronizada.
- Situação: metadados podem ser estáticos ou dinâmicos.
- Caráter: metadados criados por especialistas ou pelo coordenador do documento.
- Método de criação: metadados criados por ferramentas computacionais.
- Fonte: metadados que podem ser gerados durante o processo de criação do documento ou após.

Com o conceito sobre metadados, muitos trabalhos utilizam essa definição (não somente essa) para solucionar problemas com a criação de um padrão de metadados. Riddick et al. (2019) criaram um esquema de metadados para modelos de risco ambiental. Para isso, eles fizeram um estudo sobre alguns padrões existentes, Dublin Core, ISO 19115, Inspire, DCAT entre outros, para que o padrão fosse proposto.

Atualmente, este modelo encontra-se no Portal PURE, o qual é uma ação priorizada pelo Conselho de Pesquisa Ambiental Natural (NERC).

A criação e gerenciamento de metadados são tarefas que exigem concentração, pois, a base de dados a ser manipulada pode ser complexa dependendo do seu tamanho. Logo, a informação dos objetos, os quais estão relacionados com os metadados, podem assumir diferentes processos durante um ciclo, sendo de suma importância que os metadados definidos sejam confiáveis. A Figura 3 ilustra as fases pelas quais os objetos de informação passam.

Figura 3 – Ciclo dos objetos da informação.



FONTE: (Adaptado de BACA, 2008).

Os elementos do ciclo dos objetos da informação são descritos como segue:

- **Criação, Multifunções, Reuso:** Os objetos são criados digitalmente ou convertidos em formato digital. Esses objetos podem possuir diversas funções como, preservação, pesquisa, exibição, disseminação ou até mesmo para fins de desenvolvimento de produtos.
- **Organização e Descrição:** Os metadados assumem a função principal de descrever e ordenar os objetos em um repositório. Esses objetos podem ser organizados de forma automática ou manual, e podem incluir descrições geradas pelo criador original.

- Validação: Os metadados são examinados por usuários para verificar a autoridade e a confiabilidade desses recursos.
- Pesquisa e Recuperação: Metadados descritivos são essenciais para a capacidade dos usuários de encontrar e recuperar os objetos relevantes de metadados e informações.
- Utilização e Preservação: Com o passar do tempo, os objetos de informação ficam sujeitos a diferentes tipos de usos, podendo ser modificados. Os objetos digitais podem sofrer atualizações, migração e verificação de integridade para garantir sua disponibilidade contínua nos processos de preservação.
- Disposição: Os metadados são componente-chave na documentação da disposição de objetos e itens originais.

2.2 PRINCÍPIOS GERAIS DE PADRÃO DE METADADOS

Com o estudo da definição de metadados feito na seção 2.1, a adoção de um padrão de metadados mostra-se preciso para o gerenciamento dos dados. Os padrões de metadados contém elementos que descrevem objetos em um domínio específico. Segundo Smith e Schirling (2006), a adoção de um padrão de metadados possibilita melhor armazenamento, manipulação, recuperação e troca de informações, favorecendo a redução de trabalho e custos.

Padrões de metadados garantem que as informações sobre temas e áreas que sejam comuns, facilitem a compreensão, integração e compartilhamento de informações. (NASCIMENTO, 2008). Um esquema de padrão de metadados é feito por pessoas que possuem conhecimento para um determinado tipo de assunto, sendo possível extrair diversas características do objeto em estudo. Essas características formarão um padrão, com uma estrutura a ser seguida. Portanto, o objetivo de um padrão de metadados é descrever uma entidade (objeto) gerando uma representação padronizada que possa ser utilizada para a recuperação da mesma. Um atributo ou metadado irá representar uma única característica da entidade, enquanto que o conjunto de metadados representados em um padrão irá representar toda a entidade. Diante disso, os metadados devem identificar de maneira absoluta uma entidade, garantindo a sua unicidade.

Para elaborar metadados e padrões de metadados é preciso que seu desenvolvimento esteja baseado em princípios de modo que sua criação não seja feita de modo aleatório. Com o crescente uso de metadados e criação de padrões de metadados foram sendo desenvolvidos princípios para sua criação, produção e estabelecimento que atuam sobre a entidade em estudo.

Uma organização sem fins lucrativos conhecida como NISO (National Information Standards Organization) credenciada pela ANSI (American National Standards Institute), é responsável por gerenciar normas técnicas, estabelecendo princípios que devem ser seguidos para a criação de metadados (NISO, 2004).

No Quadro 2 será descrito seis princípios para a aplicação de metadados, sendo utilizado o termo “*good metadata*” que se entende como metadados de qualidade, criadas no guia para a construção de coleções digitais “*Framework of Guidance for Building Good Digital Collections*” (NISO, 2004):

Quadro 2 – Princípios para aplicação dos metadados.

PRINCÍPIOS	DESCRIÇÃO
Princípio 1	Metadados de qualidade estão em conformidade com os padrões da comunidade de uma forma que seja apropriada para os materiais da coleção e usos atuais e futuros da coleção.
Princípio 2	Metadados de qualidade suportam interoperabilidade.
Princípio 3	Metadados de qualidade usam controle de autoridade e padrões de conteúdo para descrever objetos e colocar objetos relacionados.
Princípio 4	Metadados de qualidade incluem uma declaração clara das condições e termos de uso do objeto digital.
Princípio 5	Metadados de qualidade apoiam a curadoria e preservação de longo prazo de objetos em coleções.
Princípio 6	Registros de metadados de qualidade são objetos em si e, portanto, devem ter as qualidades de bons objetos, incluindo autoridade, autenticidade, arquivabilidade, persistência e identificação única.

FONTE: (NISO, 2004).

Após um período de tempo, surgiu o Dublin Core Metadata Initiative (DCMI), uma organização dedicada a difundir a adoção de padrões de metadados e o desenvolvimento de vocabulário especializado de metadados, tendo como objetivo

criar mecanismos que facilitem a recuperação de recursos na internet. Suas principais características são:

- Simplicidade: Pode ser gerado pelo responsável do documento sem a necessidade de treinamentos.
- Interoperabilidade Semântica: Descrição de modelos diferentes interferem na habilidade das pesquisas entre áreas. Com a existência de um modelo comum, há um aumento da interoperabilidade dessas áreas.
- Consenso Internacional: A participação de mais de vinte países do DCMI contribui para um consenso internacional.
- Extensibilidade: *Dublin Core* possui flexibilidade e extensibilidade na elaboração de modelos, permitindo que novos elementos possam ser acrescentados para atender a uma necessidade de descrição de um determinado recurso.
- Flexibilidade: Os elementos podem ser repetidos caso necessário, e modificáveis.

Portanto, após a revisão de alguns conceitos citados, serão abordados a seguir alguns padrões de metadados que são utilizados para descrever um objeto.

2.3 PADRÕES DE METADADOS

Para representar objetos, as instituições relacionadas com tecnologia e ciência da informação têm desenvolvido padrões para serem utilizados em diferentes casos utilizando os metadados. Dentre as instituições que trabalham no desenvolvimento dos padrões, podemos citar a DCMI.

Nos dias atuais encontramos padrões em diferentes áreas para agilizar um processo. Na área de computação, um padrão de metadados pode ser usado para descrever objetos digitais, como por exemplo: vídeos, áudios, imagens, etc. Com o seu uso a busca e recuperação das informações são melhoradas, além de preservar documentos digitais. Além dessa melhoria, a padronização facilita a interoperabilidade, ou seja, sistemas se comunicam entre si de forma a compartilhar dados. Portanto, padrões podem ser definidos como ferramentas que utilizam normas para que algo seja realizado, por grupos de pessoas e/ou instituições. Segundo Moura

(2001) os padrões de metadados na área de ciência da informação tem os seguintes objetivos:

- Definir padrões de metadados, modelos, protocolos e sua integração de busca na web.
- Adquirir resultados mais precisos no processo de busca de recursos de informação na web.
- Dotar um conjunto de termos para descrição de uma ou mais categorias de recursos.

Pode-se afirmar que com os avanços da tecnologia a ciência da computação tem se aperfeiçoado para o desenvolvimento dessas ferramentas para se chegar a uma padronização para que sistemas de recuperação manuais ou automatizados façam uso dessa ferramenta.

O Dublin Core (DC) é um padrão de metadados desenvolvido na língua inglesa, para obter informações em recursos eletrônicos da web. Foi desenvolvido especificamente para o domínio da Web. O padrão de metadados Dublin Core é constituído em dois níveis:

- Nível Simples: possui quinze elementos de descrição.
- Nível Qualificado: Além dos quinze elementos de descrição, possui sete elementos adicionais e duas classes de qualificadores.

As classes de qualificadores são:

- Elementos de Refinamento: São utilizados para refinar o significado dos elementos descritivos.
- Esquemas de codificação: Identificam esquemas para o valor do elemento (metadado), podendo citar o vocabulário controlado, notações de sistemas de classificação, etc.

No Quadro 3 (três) temos os elementos do DC em nível simples, onde são apresentados 15 (quinze) elementos (metadados) e as suas definições.

Quadro 3 – Elementos do DC em nível simples.

(continua)

ELEMENTOS	DEFINIÇÃO
Título	Um nome dado ao recurso.
Autor	Uma entidade principalmente responsável por criar o recurso.
Assunto	O tópico do recurso.

Quadro 3 – Elementos do DC em nível simples.

(conclusão)

ELEMENTOS	DEFINIÇÃO
Descrição	Uma conta do recurso.
Editor	Uma entidade responsável por disponibilizar o recurso.
Colaborador	Uma entidade responsável por fazer contribuições para o recurso.
Data	Um ponto ou período de tempo associado a um evento no ciclo de vida do recurso.
Tipo	A natureza ou gênero do recurso.
Formato	O formato do arquivo, meio físico ou dimensões do recurso.
Identificador	Uma referência inequívoca ao recurso em um determinado contexto.
Fonte	Um recurso relacionado do qual o recurso descrito é derivado.
Linguagem	Um idioma do recurso.
Relação	Um recurso relacionado.
Cobertura	O tópico espacial ou temporal do recurso, a aplicabilidade espacial do recurso ou a jurisdição sob a qual o recurso é relevante.
Direitos	Informações sobre direitos mantidos no recurso e sobre ele.

FONTE: (DCMI, 2012)

No Quadro 4 são mostrados os elementos do DC em nível qualificado onde se encontram 7 (sete) elementos (metadados) e as suas definições.

Quadro 4 – Elementos do DC em nível qualificado.

(continua)

ELEMENTOS	DEFINIÇÃO
Público	Uma classe de agentes para os quais o recurso se destina ou é útil.
Proveniência	Uma declaração de quaisquer mudanças na propriedade e custódia do recurso desde sua criação que sejam significativas para sua autenticidade, integridade e interpretação.
Periodicidade de Acúmulo	A frequência com que os itens são adicionados a uma coleção.
Política de Acúmulo	A política que rege a adição de itens a uma coleção.

Quadro 4 – Elementos do DC em nível qualificado.

(conclusão)

ELEMENTOS	DEFINIÇÃO
Detentor de Direitos	Uma pessoa ou organização que possui ou gerencia direitos sobre o recurso.
Método Instrucional	Um processo, usado para gerar conhecimentos, atitudes e habilidades, que o recurso descrito visa apoiar.

(FONTE: DCMI, 2012)

Os elementos de nível simples e nível qualificado podem ser descritos em uma linguagem de marcação, como o HTML do inglês *Hypertext Markup Language* que significa Linguagem de Marcação de Hipertexto. Essa descrição pode ser feita utilizando *tags*: <nome=atributo propriedade=valor>. A seguir, é ilustrado, um exemplo de como os elementos são representados no Padrão Dublin Core.

- <meta name=DC.Titulo" content="Padrão de Metadados Dublin Core">

A utilização de todos os elementos do padrão DC é opcional e eles podem ser repetidos, isso mostra a flexibilidade do padrão. Com isso, existe uma diversidade de estrutura com diferentes conjuntos de metadados, sendo essa estrutura formada conforme a necessidade de cada um. Neste padrão são adotados dois princípios.

- a) princípios relacionados aos metadados e aos recursos que eles descrevem.

Este princípio cita três princípios básicos que relacionam os metadados com os recursos:

- Princípio um para um: No geral os elementos do padrão DC descrevem uma manifestação ou um recurso, o qual corresponde a uma representação, ao invés de relacionar manifestações. Podemos citar como exemplo uma imagem sendo reproduzida conforme uma imagem original, portanto a imagem reproduzida não terá na descrição do padrão DC uma relação com a imagem original. De acordo com Hillmann (2005), cada imagem terá uma descrição diferente, não sendo relacionada.
- Princípio "Dumb-down": Esse princípio é utilizado para que o implementador do padrão possa ignorar qualificadores e que possa utilizar o valor absoluto, por mais que haja perda de especificidade, o valor do metadado (elemento) deve ser válido e útil.

- Valor apropriado: Um determinado elemento ou qualificadores podem adotar boas práticas em contextos variados, porém, devem-se manter o seu valor.
- b) princípios gerais para o estabelecimento da estrutura do esquema de metadados: Este princípio aponta quatro metas:
 - Simplicidade de descrição e manutenção: O padrão DC foi desenvolvido com o objetivo de possuir um número mínimo de metadados que seja descrito e manipulado por pessoas que não sejam da área de informação, sendo sua recuperação garantida pela descrição mínima (HILLMANN, 2005).
 - Semântica comumente compreendida: A busca de informação é dificultada pelas diferenças terminológicas e práticas descritivas quando o domínio são distintos. Os metadados do DC são um esquema genérico com semântica compreendida (significado dos metadados) o que garante a interoperabilidade entre domínios diferentes (HILLMANN, 2005).
 - Alcance Internacional: Vários países participam do DCIM, buscando um escopo internacional para definir uma estrutura para o DC. (HILLMANN, 2005).
 - Extensibilidade: O DC consegue estender e acrescentar elementos descritivos ao conjunto do esquema de metadados. Com isso, o conjunto de metadados pode ser estendido conforme o seu domínio. (HILLMANN, 2005).

Neste padrão pode-se utilizar diferentes sintaxes como por exemplo: a linguagem HTML, linguagem XHTML, linguagem XML, RDF / XML entre outras.

- Exemplo na linguagem HTML:

```
<html>
<head>
  <meta name="DC.creator" content="Bruna Neves"/>
</head>
<body>
  ...
</body>
```

</html>

- Exemplo na linguagem XML

<metadata>

<dc:creator>Bruna Neves</dc:creator>

<dc:identifier>

....

</dc:identifier/>

</metadata>

O uso de diferentes sintaxes não influencia na semântica estabelecida pelo padrão DC, ou seja, foi projetado para ser independente de modo que, estabelece uma estrutura e uma semântica para os dados não se importando com o tipo de sintaxe que se utilizará. Vale ressaltar que o uso de sintaxes é muito utilizado para representar os dados da Web, o que oferece diversas funcionalidades para o seu uso, como por exemplo a recuperação de informações. Este padrão pode ser utilizado em conjunto com outros padrões de metadados, tendo como objetivo se relacionar com outros padrões na descrição de recursos.

2.4 LINGUAGEM XML

A linguagem de marcação estendida, do inglês *Extended Markup Language* (XML) publicada em 1997 pelo *W3Consortium* é utilizada para descrever conteúdo, permitindo maior facilidade para a interpretação dos dados pelos computadores, sendo assim, maior facilidade para construção de aplicativos, páginas webs etc. Seu uso permite a criação de marcações pelo autor do documento para que a comunicação seja feita.

Sua estrutura é caracterizada com a utilização de sinais "<" e ">" onde ocorre o início da estrutura de marcação e os sinais "<" e ">" para o fechamento da estrutura. A seguir um exemplo de uma estrutura em XML e seus dados textuais.

- Estrutura XML

<Pessoa>

<nome>Bruna</nome>

<sobrenome>Neves</nome>

```
<email>teste@gmail.com</email>
```

```
</Pessoa>
```

- Dados Textuais

```
Bruna
```

```
Neves
```

```
teste@gmail.com
```

Um documento em XML é bem formado quando sua estrutura possui um único elemento raiz, e seus elementos são aninhados adequadamente, além de seguir um conjunto de regras chamado de DTD (*Data Type Definition*).

2.5 LINGUAGEM HTML

O HTML foi desenvolvido para descrever a apresentação de dados em páginas da internet. Seu funcionamento pode ser visto em qualquer navegador, como o Google Chrome ou Mozilla Firefox. Sua linguagem consiste em uma série de tags que são uma estrutura hierárquica para apresentação das informações. Assim como no XML, no HTML é utilizada a sintaxe de abertura e fechamento de tags.

Na linguagem HTML as tags utilizadas são fixas, isso não ocorre com o XML onde as tags são criadas conforme a necessidade do desenvolvedor. A seguir, um fragmento em HTML onde pode-se observar algumas *tags* fixas dessa linguagem.

- Fragmento de um Estrutura em HTML

```
<h1>Autora desta dissertação</h1>
```

```
<p>Bruna Neves, teste@gmail.com</p>
```

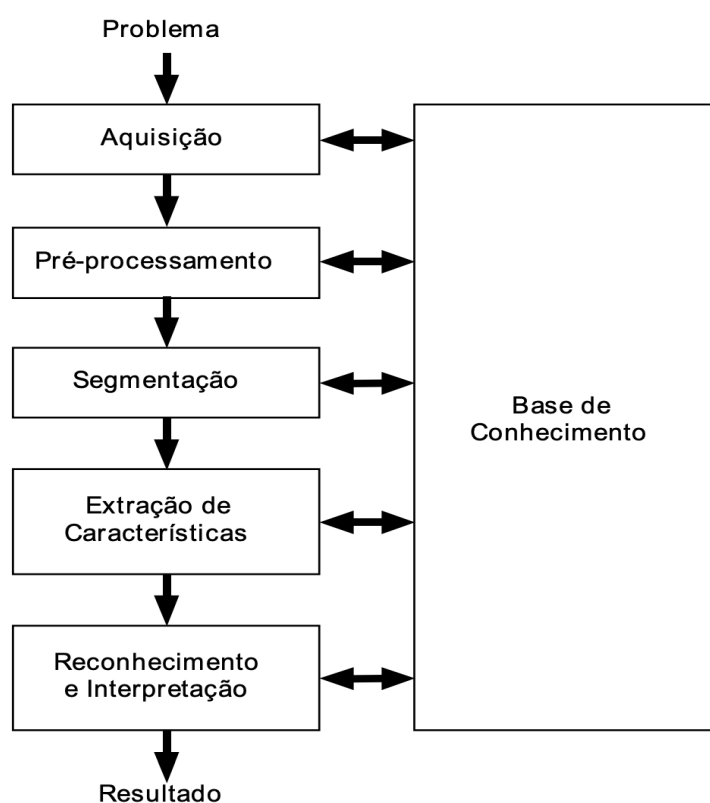
Ao contrário do que acontece no XML, no fragmento acima, os desenvolvedores não criam *tags*, mas sim, utilizam as *tags* já definidas na linguagem para descrever e mostrar os dados em uma página web. Portanto o HTML descreve a aparência e ações em uma página enquanto o XML descreve o conteúdo de um documento.

2.6 PROCESSAMENTO DIGITAL DE IMAGENS

O Processamento Digital de Imagens (PDI) é uma área que está em crescente evolução. É através desta técnica computacional que é possível obter informações das imagens para tomar uma decisão. Seu uso se deu no início da década de 1920, mas apenas em 1960 que houve maior avanço. (MARQUES FILHO; VIEIRA NETO, 1999).

Uma abordagem que explica o funcionamento do processamento digital de imagens é mostrada na Figura 4, onde a primeira etapa consiste na aquisição das imagens.

Figura 4 – Etapas do processamento digital de imagens.



FONTE: (MARQUES FILHO; VIEIRA NETO, 1999)

A primeira etapa explica a forma como as imagens de estudos serão obtidas. Nesta etapa ocorre a escolha do equipamento (sensor) que será utilizado para a captura das imagens tendo em vista a problemática a ser resolvida. A etapa de pré-processamento, após feita a captura das imagens, é analisar os possíveis problemas

encontrados nas imagens, como por exemplo: ruídos, dimensões das imagens, bordas entre outros (MARQUES FILHO; VIEIRA NETO, 1999), para que possíveis melhorias sejam aplicadas nas imagens.

A etapa de segmentação procura identificar um objeto específico na imagem para realizar os estudos, no qual a imagem é dividida em segmentos. É considerada uma das tarefas mais difíceis dentre todas as etapas, pois seu resultado depende diretamente de um bom reconhecimento na segmentação dos objetos nas imagens (GONZALEZ; WOODS, 2010). A extração de características busca selecionar informações específicas nas imagens referente ao problema a ser solucionado. Muitas vezes a entrada desta etapa é uma imagem já segmentada.

Com todas as características extraídas, essas informações são a entrada para a etapa seguinte (reconhecimento e interpretação) sendo que o reconhecimento é a etapa que define um rótulo a um determinado objeto, onde o rótulo é definido com base nas informações fornecidas pelos respectivos descritores. Todas as etapas necessitam de uma base de conhecimento sobre um determinado problema. Entretanto, nem todas as etapas mostradas na Figura 3 estão presentes no processamento digital de imagens, dependendo diretamente do problema abordado.

2.6.1 Imagens Digitais

Segundo Marques Filho e Vieira Neto (1999) uma imagem monocromática pode ser definida matematicamente por uma função $f(x,y)$ da intensidade luminosa, sendo seu valor, em qualquer ponto de coordenadas espaciais (x,y) , proporcional ao brilho, conhecido também como imagem binária naquele ponto.

Os seres vivos interpretam de maneira distinta as imagens em comparação com computadores. Computadores interpretam números, portanto as imagens são convertidas em um conjunto numérico, conhecido como pixels. Os *pixels* ou *pels*, abreviação de "*picture elements*", são os menores elementos constituintes nas imagens. A Figura 5 mostra uma imagem em escala de cinza, cujo os elementos dessa matriz (imagem) digital são chamados de pixels significando "elementos de figura" (GONZALEZ; WOODS, 2010; PEDRINI; SCHAWARTZ, 2008).

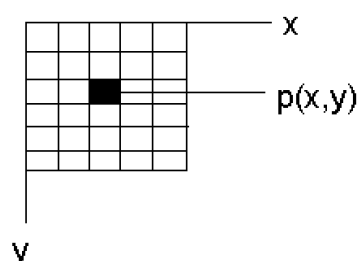
Figura 5 – Uma imagem em escala de cinza e a convenção utilizada para o par de eixos (x,y) .



FONTE: (MARQUES FILHO; VIEIRA NETO, 1999).

A Figura 5 mostra a convenção que os autores adotaram, diferente do sistema cartesiano ortogonal, onde a origem da imagem está situada no canto superior esquerdo, sendo que neste trabalho esse sistema também será adotado. Essa escolha, não admite que as coordenadas X e Y sejam negativas (MARQUES FILHO; VIEIRA NETO, 1999). Como ilustra a imagem 4, as imagens digitais podem ser representadas como uma matriz de duas dimensões, onde no ponto $(0,0)$ está concentrado o valor de um pixel. Para melhor ilustração, a Figura 6 ilustra uma representação matricial.

Figura 6 – Imagem representada por matriz.

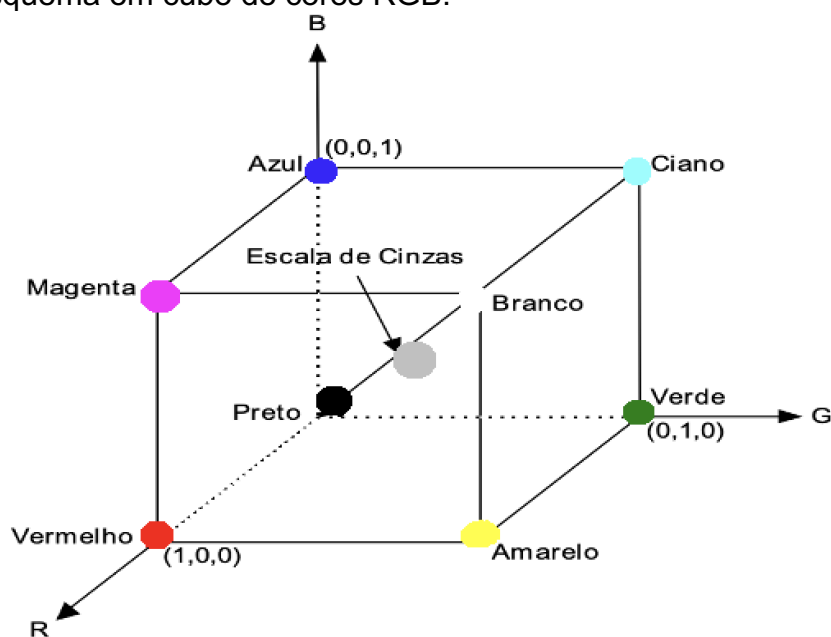


Fonte: A autora

Conforme ilustra a Figura 6, a imagem representada por meio de uma matriz, as coordenadas do eixo X e Y identificam um ponto na imagem, sendo que o valor neste ponto corresponde ao nível de cinza naquela coordenada. A distribuição de níveis cinza nas imagens monocromáticas podem ser representadas por um histograma.

Além de imagens digitais em escala de cinza, temos as imagens digitais coloridas, que são formadas por matrizes tridimensionais, compondo uma imagem multiespectral. As cores visíveis pelo olho humano representada pelas cores RGB (Red, Green, Blue – Vermelho, Verde, Azul) que pode ser representado por um cubo. Na Figura 7 ilustra o modelo RGB.

Figura 7 – Esquema em cubo de cores RGB.



(FONTE: MARQUES FILHO; VIEIRA NETO, 1999).

Os vértices ilustram as cores primárias (Figura 7) as quais representam os canais RGB, e ao longo da diagonal principal se encontram os valores que representam os tons de cinzas. Na representação de cores, o padrão RGB como mostra a Tabela 1, possui um limite na sua representação, sendo a cor preta representada por RGB (0,0,0) sendo o menor valor possível e a cor branca RGB (255, 255, 255) sendo o maior valor possível.

Tabela 1 – Valores das cores no formato RGB.

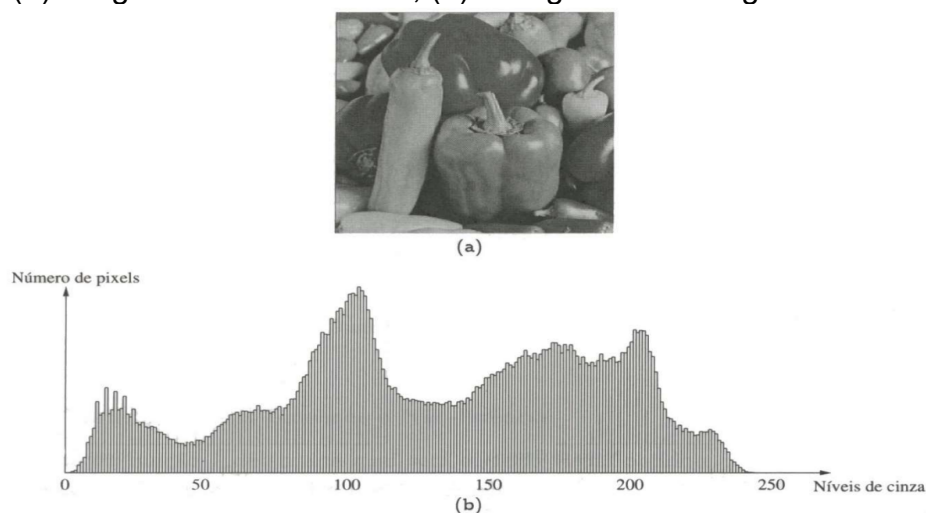
Cor	Valor (RGB)
Preto	0, 0, 0
Magenta	255, 0, 255
Verde	0, 255, 0
Azul	0, 0, 255
Branco	255, 255, 255
Vermelho	255, 0, 0
Amarelo	255, 255, 0
Ciano	0, 255, 255

Fonte: A autora

2.6.2 Histograma

Para descrever imagens digitais podemos utilizar histogramas, não considerando a localização dos pixels. A Figura 8(a) ilustra uma imagem em escala cinza, onde os pixels da imagem (conjunto numérico) são representados pelo histograma (Figura (b)) que indica os níveis de cinza da imagem. Esses valores são visualizados em um gráfico de barras que mostra para cada nível cinza o número de pixels correspondentes na imagem.

Figura 8 – (a) imagem em escala cinza; (b) histograma da imagem.



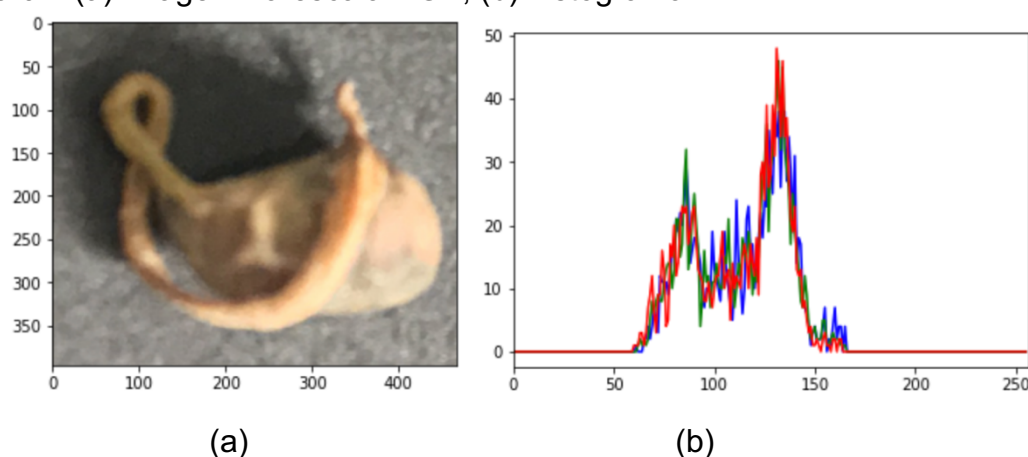
FONTE: (PEDRINI; SCHWARTZ, 2008)

O histograma pode ser expresso por:

$$p_k(f) = \frac{n_k}{n} \quad (1)$$

onde $p_k(f)$ é a probabilidade de ocorrência de (f) , n_k representa a quantidade de *pixels* de nível cinza k e n representa o total de *pixels* na imagem. Em imagens digitais coloridas também é possível calcular o histograma com a diferença que serão vistos três canais de cores, o RGB.

Figura 9 – (a) Imagem na escala RGB; (b) histograma.



Fonte: A autora

Conforme a Figura 9(b) podemos ver o histograma da Figura 9(a) onde no eixo x temos as frequências de 0 a 255 e no eixo y, temos os valores dos *pixels*.

2.6.3 Segmentação

Como o próprio nome diz, é a etapa de separação de uma imagem em unidades de interesse. A quantidade de unidades de interesse da imagem é dividida conforme o problema que será resolvido. A sua utilização possui grande influência no resultado, pois, quanto mais precisa a segmentação for, a chance de sucesso para o reconhecimento dos objetos aumenta. (GONZALEZ; WOODS, 2010).

Para as imagens em tons de cinza existem algoritmos de segmentação baseados em descontinuidade ou similaridade. A descontinuidade ocorre quando os níveis de tons de cinza mudam bruscamente, e é ela que identifica as bordas presentes nas imagens. As bordas delimitam duas regiões na imagem com tons de cinza distintos. Já a similaridade baseia-se na limiarização, ou *thresholding*, que se

baseia na diferença dos níveis de cinza, compondo diferentes objetos em uma imagem.

Com a limiarização é possível classificar objetos de uma mesma imagem em duas classes distintas usando um valor de limiar. Esse processo é conhecido como binarização, pois produz uma imagem binária como saída. Os objetos de interesse na imagem podem ser frontais, de fundo ou um objeto escolhido para ser detectado, sendo adotado a cor, borda ou similaridade como características definidas. (SOLEM, 2012).

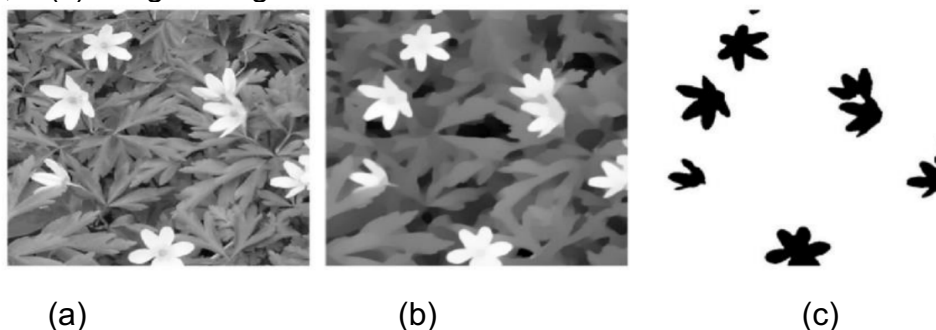
Algumas ferramentas são utilizadas no pré-processamento de imagens para remoção de ruídos, os filtros. Os filtros atuam na imagem com o objetivo de reduzir ou eliminar imperfeições realizando a manipulação dos valores de intensidade do *pixel*. É através dessa etapa de pré-processamento que obtemos imagens de maior qualidade do que a original. Os filtros podem ser expressos pela Equação 1.

$$g(x, y) = T [f(x, y)] \quad (1)$$

onde:

- $f(x, y)$ é a imagem original,
- $g(x, y)$ é a imagem transformada,
- T é uma transformação da imagem em uma vizinhança na posição x, y .

Figura 10 – Imagem em processo de segmentação. (a) imagem original; (b) imagem com filtro; e (c) imagem segmentada.



FONTE: (SOLEM, 2012)

A Figura 10(a) apresenta uma imagem contendo flores e folhas onde se quer isolar as flores do restante da imagem, ou seja, segmentar. É possível notar que os

pixels das flores estão em tons próximos do branco, logo, utilizando um limiar com tons de pixels nessa cor na imagem, resolveria o problema de segmentação.

Na Figura 10(b) foi aplicado um filtro da média na imagem, devido algumas regiões da haste e de folhas possuírem tons próximos ao branco também e isso apareceria na imagem final. Com isso, o limiar foi aplicado e a imagem foi segmentada com sucesso (Figura 10(c)).

2.7 EXTRAÇÃO DE CARACTERÍSTICAS

A extração de características visa transcrever as informações contidas em pontos de interesse de uma imagem buscando selecionar características que sejam importantes para a classificação e análise de uma forma mais precisa. Essa etapa ocorre logo após a etapa de segmentação, a qual diversas características podem ser obtidas, como histograma, área, perímetro, contorno etc.

É importante ressaltar que, ainda na etapa de extração de características a entrada é uma imagem, e a saída é um conjunto de características correspondentes à imagem. Por meio de descritores as informações das imagens são caracterizadas, onde os descritores são representados por uma estrutura de dados conveniente ao algoritmo de classificação.

Figura 11 – Extração de característica (área) de um grão de soja.



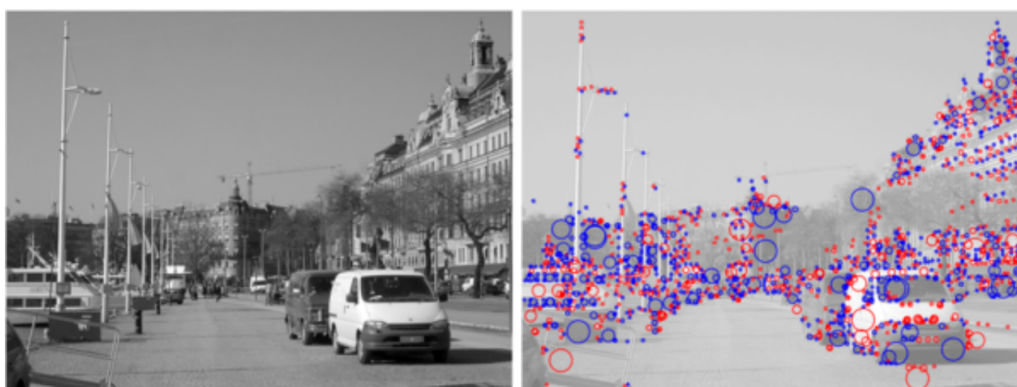
Fonte: A autora

Na Figura 11 é mostrado um exemplo de extração de características, onde foi detectado o contorno com a cor vermelha ao redor do grão de soja. A computação com suas ferramentas consegue resolver vários problemas utilizando visão

computacional, área que está em alta. Existem algoritmos descritores que realizam a extração de características, como o SIFT e o SURF.

Transformação de recurso de escala invariável, do inglês *Scale-Invariant Feature Transform* (SIFT), foi desenvolvido por Lowe (2004) e é baseado em Diferença de Gaussianas. Este descritor é usado em inúmeras finalidades na visão computacional relacionados à correspondência de pontos de uma cena 3D, rotação, transformações e intensidade. Inicialmente o algoritmo SIFT utilizava um método que detectava pontos de interesse ou pontos-chave de uma imagem em escala cinza para que fosse combinado em pontos de interesse correspondentes entre imagens diferentes, logo depois, o SIFT começou a ser aplicado em grades densas, demonstrando um melhor desempenho em diferentes tarefas como por exemplo: classificação de texturas, biometria etc.

Figura 12 – Uma imagem cinza com pontos de interesse detectados.



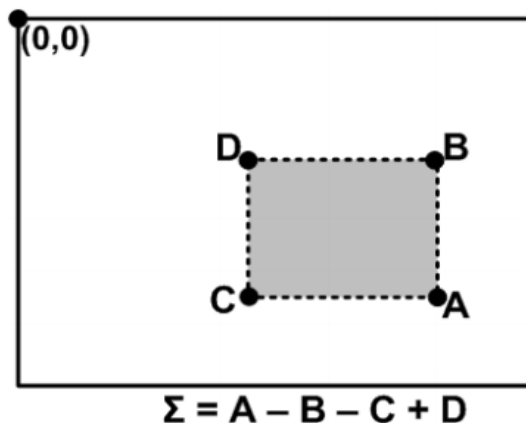
FONTE: (LINDBERG, 2015)

A Figura 12 ilustra os pontos de interesse na imagem em escala cinza, onde os raios dos círculos ilustram os pontos de interesse, sendo que os círculos vermelhos representam características claras da imagem e os círculos azuis representam as características escuras da imagem.

Características Robustas Aceleradas do inglês *Speeded Up Robust Features* (SURF), foi proposto por Bay et al. (2008), o qual é um detector e descritor de características invariante em relação à escala e rotação. Esse extrator computa pontos de interesse e vetores de características muito mais rápidas que o SIFT. O detector e o descritor usam técnicas de imagem integral o qual reduz o tempo de computação devido aos filtros de convolução serem baseados em caixa (Figura 13).

Cada elemento de uma imagem integral representa a soma de todos os pixels da imagem original que estão contidos na região retangular situada entre a origem e (x,y).

Figura 13 – Imagem integral.



FONTE: (MARTINS, 2014)

O descritor utiliza a distribuição (derivada de primeira ordem) de *Wavelet de Haar* para descrição. Os autores propuseram um descritor baseado em matrizes Hessianas com metade do número de elementos do SIFT o que garante mais estabilidade, melhor performance no tempo computacional e melhores taxas de acerto.

2.8 MACHINE LEARNING

Aprendizado de máquina, ou *machine learning* (ML), são métodos matemáticos que buscam padrões para que máquinas e algoritmos aprendam sozinhos e possam realizar tomadas de decisões para resolução de problemas. ML pode ser definido como um conjunto de algoritmos que dão sentido a uma base de dados, transformando os dados em conhecimento (RASCHKA, 2015). De forma simplificada, o algoritmo de aprendizado de máquina resolve um problema diversas vezes, utilizando uma medida de avaliação. Com isso, o algoritmo busca otimizar a medida para que o problema possa ser solucionado. Segundo Raschka os modelos de aprendizado de máquina podem ser divididos em três tipos de aprendizados de máquinas:

- **Aprendizado supervisionado:** Neste caso a base de dados que está sendo utilizada contém uma variável observada que indica qual a classificação de cada dado. Essa variável é comumente chamada por rótulos ou *labels*.
- **Aprendizado não supervisionado:** Neste caso a base de dados não apresenta uma variável dependente. Com isso, os algoritmos têm como objetivo comparar os elementos da base de dados entre eles.
- **Aprendizado por reforço:** Neste tipo de aprendizagem, o algoritmo aprenderá qual resultado ou tomada de decisão a ser obtido sobre as circunstâncias a qual for submetido.

Os três algoritmos citados podem consumir muitos recursos computacionais devido ao número de repetições para o aprendizado da rede ou pela complexidade algorítmica. Nesses casos, é necessário utilizar ferramentas que aumentem a capacidade computacional, e uma forma de obter isso é usando computação distribuída ou paralela.

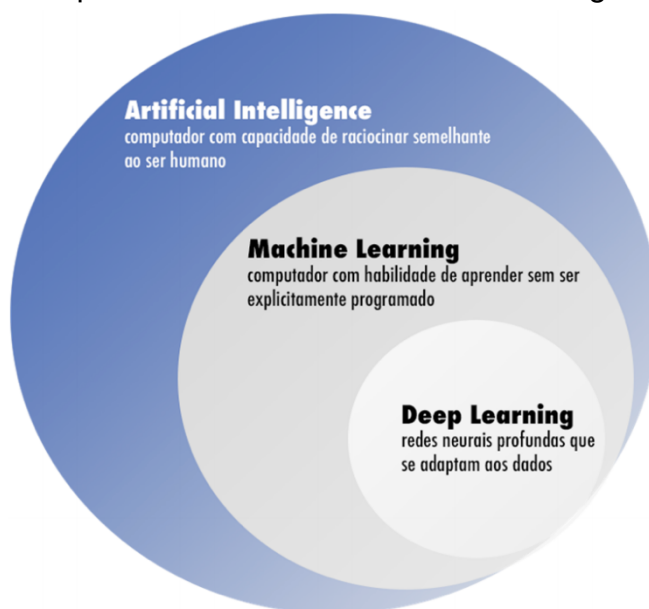
Com *machine learning* é possível desenvolver diferentes tarefas tais como: regressão, segmentação, clusterização entre outras. Neste projeto o foco será em problemas de classificação. Podemos citar por exemplo uma base de dados de motos e informarmos ao algoritmo que essas imagens são de motos, com isso caberá ao algoritmo de classificação descobrir qual é a lógica por trás das informações. Isso se refere ao aprendizado supervisionado, já na aprendizagem não supervisionada, não informamos ao algoritmo qual é a resposta para os dados de entrada, sendo que o algoritmo fica encarregado de encontrar um padrão para os dados de saída.

2.9 DEEP LEARNING

Deep Learning (DL) ou Aprendizagem profunda é uma subárea do ML, o qual é uma subárea de inteligência artificial como mostra a figura 14. Ela é um tipo de modelo de *machine learning* que se assemelha com os neurônios cerebrais humanos. O DL utiliza um modelo bioinspirado, como a rede neural artificial, onde os neurônios artificiais estão organizados em camadas interconectadas (ARROYO-FIGUEROA et al., 2000).

É uma das técnicas utilizadas para aprendizado de máquina que utiliza redes neurais artificiais para processar informações e realizar aprendizagem para que a rede seja capaz de identificar, de maneira correta, o objeto em estudo, podendo citar, áudio no reconhecimento de fala, imagens no reconhecimento facial, entre outros.

Figura 14 – Esquema que mostra divisões na área de Inteligência Artificial.



FONTE: (MENEZES, 2020).

2.10 CONVOLUÇÃO

A palavra convolução vem da ideia de “mudança de estado”, que na matemática, analogicamente, define-se como sendo a modificação de um sinal de entrada para a geração de um sinal de saída. A convolução é utilizada para se referir à operações lineares, que a partir de duas funções dadas, resulta numa terceira função que mede a soma do produto dessas funções. Logo, essa transformação é um sistema linear de sinal invariante ao deslocamento. Portanto, o cálculo da convolução discreta é relevante em diversos estudos.

A Óptica de Fourier se relaciona com a convolução, pois ela descreve o sinal de saída de um sistema linear invariante ao deslocamento como a superposição ponderada das respostas ao impulso deslocadas no tempo (FIGUEIREDO, 1977; PEREIRA, 2011). *Convolution Neural Network* (CNN) podem ser utilizadas em diversas aplicações de processamento digital de sinais, pois são redes de múltiplas

camadas, que implementam filtros baseados na convolução discreta. As CNN utilizam técnicas computacionais através de um modelo matemático baseando-se na estrutura neural de organismos inteligentes, adquirindo conhecimento através da experiência, ou seja, do treinamento da rede. Elas podem ser aplicadas em diferentes áreas como por exemplo na detecção e reconhecimento de faces.

2.10.1 Definição da Convolução Discreta

O sinal discreto é definido como (n) , e sua representação pode ser vista na Equação 2.

$$(n) = \{ 1, \text{se } n = 0, 0, \text{se } n \neq 0. \} \quad (2)$$

Como pode ser visto na Equação 1, o sinal discreto determina um único valor no instante $n=0$. Caso deseje deslocar o impulso unitário no tempo, é necessário mudar o argumento da função (n) . Por exemplo, para $k = 2$, a Equação 3 teria a seguinte mudança:

$$(n - 2) = \{ 1, \text{se } n = 2, 0, \text{se } n \neq 2. \} \quad (3)$$

2.10.2 Formulação Discreta

O esforço computacional de uma CNN é relativo ao cálculo do algoritmo de convolução discreta. Sendo assim, para ter maior eficiência, adota-se o cálculo das multiplicações da convolução em paralelo, somando as multiplicações.

Portanto, adotou-se a seguinte formulação, como mostra a Equação 4.

$$z_{ij} = y_{ij} + \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} x_{i+m,j+n} w_{mn} \quad (4)$$

Onde:

- x_{ij} é o valor da posição ij da matriz referente à imagem de entrada,
- w_{mn} representa a matriz $K \times K$ referente ao *kernel*,
- y_{ij} é um viés que é combinado com o resultado,

- z_{ij} é o valor de saída na matriz respectiva.

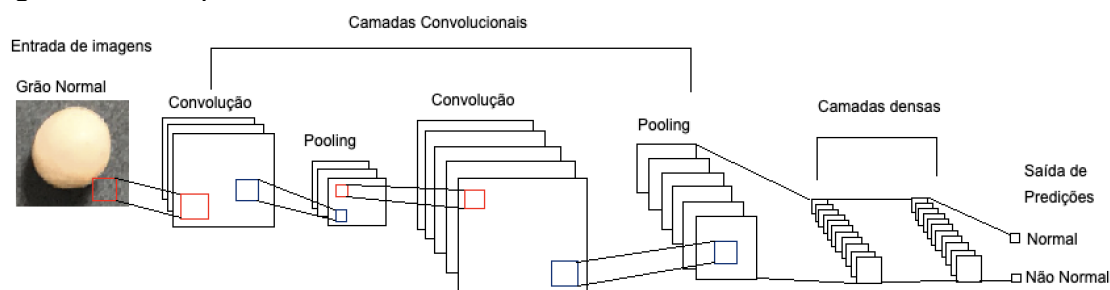
2.11 REDE NEURAL CONVOLUCIONAL

CNN também conhecida como *ConvNet* é uma rede que é utilizada em análises de imagens digitais. Uma CNN pode ser considerada uma rede *feedforward* com várias camadas de convolução intercaladas de subamostragem (camada de *pooling*), funções de ativação e uma camada totalmente conectada. Esta rede possui neurônios que possuem pesos e bias que são treinados durante o processamento dos dados. Elas utilizam imagens RGB ao invés de imagens na escala cinza, proporcionando uma melhora significativa nos resultados de classificação, detecção e reconhecimento em imagens e vídeos.

LeCun et al. (1998) desenvolveram uma arquitetura neural conhecida como LeNet5 que foi desenvolvida para reconhecer dígitos à mão, tendo atingindo 99,2% de acurácia na base de dados MNIST.

As CNN são caracterizadas por serem compostas por camadas convolucionais as quais processam as entradas (base de conhecimento), mais conhecidas como camadas de entradas, para extrair as características. A camada de entrada geralmente é composta por uma imagem com largura, altura e profundidade. Além da camada de entrada, também é encontrada a camada convolucional, camada de *pooling* e camada totalmente conectada. A Figura 15 ilustra a arquitetura de uma CNN.

Figura 15 – Arquitetura de uma Rede Neural Convolucional.



Fonte: A autora

A Figura 15 mostra uma rede neural convolucional recebendo uma imagem como uma matriz utilizando os canais RGB. Essa matriz vai passando pelos filtros

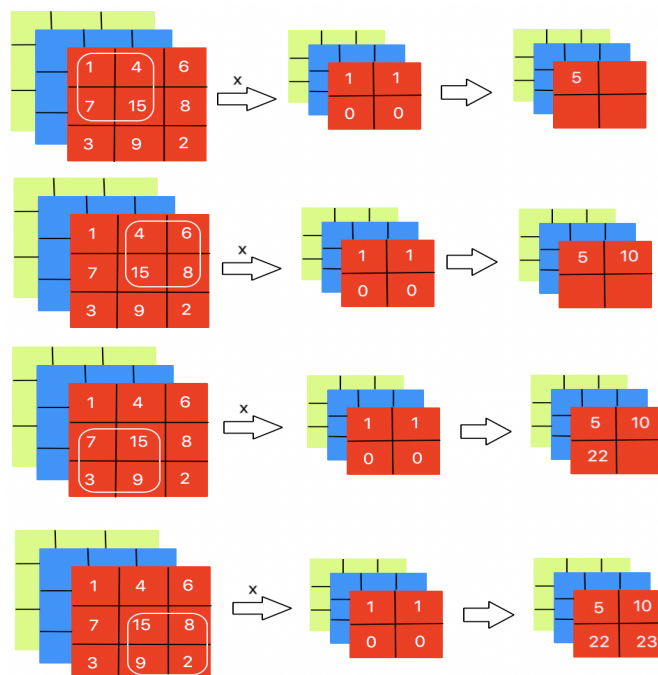
convolucionais, pelo *pooling*, funções de ativação, onde em cada saída é gerado um mapa de características até chegar nas camadas densas (totalmente conectadas) para que a classificação seja realizada.

2.11.1 Camada Convolucional

É considerada a camada mais importante da rede, sendo realizada a parte mais pesada do processamento computacional. Esta camada é composta por um conjunto de filtros ou *kernels* que são capazes de aprender conforme o treinamento da rede. Os filtros podem ser interpretados como pesos que são convoluídos com os pixels das imagens para obter mapa de características. Estes, por sua vez, indicam regiões características específicas em relação ao filtro.

Ao longo do processamento da rede os filtros são alterados para que a rede aprenda a identificar regiões importantes para a extração de características da base de conhecimento. A Figura 16 ilustra uma imagem 3x3x3 e um filtro (*kernel*) 2x2x3, tendo como resultado uma matriz 2x2x3. A matriz resultante é devido ao produto escalar entre o filtro e a imagem. Essa etapa é chamada de convolução. O filtro percorre a imagem calculando o produto escalar, utilizando um deslocamento de 1 pixel para cima ou para baixo. Esse deslocamento é conhecido com *stride* (um hiperparâmetro da CNN), que neste caso é 1. O resultado final deste procedimento é denominado mapa de características.

Figura 16 – Representação matricial de uma imagem RGB sendo convoluída com um filtro (*kernel*).



Fonte: A autora

Com a matriz de resultado formada (Figura 16), é aplicada uma função de ativação, sendo que a mais comum é a função *ReLU* (KRIZHEVSKY et al, 2012). Dentre as características da CNN podemos citar: Conectividade esparsa, campos receptivos locais e compartilhamento de parâmetros.

- Conectividade esparsa: Esta conectividade ocorre devido ao fato que o filtro (*kernel*) utilizado no processo de convolução apresenta dimensões muito menores do que a matriz de entrada (imagem). Com isto, ocorre uma correlação espacial local através de um padrão de conectividade local entre neurônios de camadas adjacentes.
- Campos receptivos locais: uma região na matriz de entrada representa o tamanho do filtro que será utilizado, que corresponde ao único neurônio na camada.
- Compartilhamento de parâmetros: São aplicados na matriz, nas diferentes regiões os mesmos filtros.

Segundo Goodfellow et al., (2016) a conectividade esparsa, campos receptivos locais, compartilhamento de parâmetros, permitem que a CNN armazenem menos parâmetros, fazendo com que reduza os requisitos de memória do modelo, além de

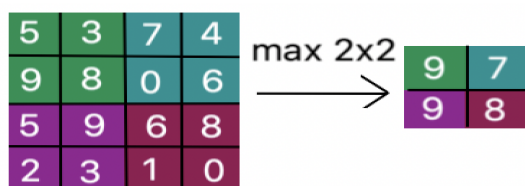
melhorar a eficiência estatística. Além do mais, os resultados obtidos requerem menores operações, o que por consequência, possibilita uma melhoria na eficiência destes modelos.

É importante ressaltar dois hiperparâmetros em relação aos filtros utilizados nas camadas convolucionais, são eles: *wide* ou *narrow convolution* e o *stride*. O *wide* define a maneira de como aplicar o filtro, ou seja, analisando uma matriz de dados, como aplicar o filtro aos elementos da matriz que não possuem vizinhos, esses elementos são conhecidos como elementos na borda da matriz. Nesses casos, é comumente utilizado um *padding* (preenchimento) que tem como objetivo expandir a matriz de entrada. Quando o uso do *padding* é necessário, ou seja, adicionar elementos na borda da matriz de entrada com valor zero, este hiperpâramento é conhecido como *wide convolution*. Entretanto, caso não seja necessário a utilização do preenchimento, o hiperpâramento é conhecido como *narrow convolution*. O *stride*, como dito anteriormente, é responsável por determinar o tamanho do deslocamento do filtro a cada passo.

2.11.2 Camada de *POOLING*

A camada de *pooling* tem como objetivo reduzir o tamanho espacial das matrizes resultantes da convolução, diminuindo a quantidade de parâmetros a serem aprendidos na rede, controlando o *overfitting*. Também é necessário determinar o tamanho do filtro (*kernel*) e o *stride*. Segundo Goodfellow et al., (2016) as saídas de determinadas regiões das camadas convolucionais são substituídas pelas camadas de *pooling* por estatística resumida das saídas mais próximas, o que pode gerar uma redução dos neurônios da camada anterior. Na Figura 17 é ilustrado o *pooling* sendo aplicado no resultado de uma convolução.

Figura 17 – Exemplo de *pooling* de uma imagem 4x4.



Fonte: A autora

A Figura 17 mostra uma imagem 4x4 e o filtro do *pooling* com dimensões 2x2 com *stride* igual a 2. Além disso, é aplicado a função de agregação $\max(X)$ que escolhe o maior valor da região. Outras funções podem ser utilizadas, como a média, a normal L2 ou a média ponderada da região analisada. As camadas de *pooling* são utilizadas depois das camadas convolucionais, e além de reduzir o tamanho das matrizes, elas são responsáveis por encontrar as informações mais importantes das camadas convolucionais.

2.11.3 Camada *DROPOUT*

Essa camada (HINTON et al., 2012) tem como finalidade definir estocasticamente zeros a saída de alguns neurônios de uma determinada camada com probabilidade de $p=0,5$. Os neurônios não modificados são treinados com o algoritmo *backpropagation*. Para cada época do treinamento, todo esse procedimento é repetido. O *dropout* é utilizado para regular um problema que as CNNs apresentam. As CNNs possuem menos conexões e parâmetros devido a presença de conectividade local e armazenamento de filtros (*kernel*) e por essa razão essas arquiteturas utilizadas podem apresentar milhares de camadas de processamento aumentando a quantidade de parâmetros na rede, devido a este acontecimento as CNNs podem apresentar um problema conhecido como sobreajuste do inglês *overfitting*, que significa quando o modelo se ajusta muito bem aos dados que foram utilizados no treinamento e por consequência não generaliza bem.

2.11.4 Camada Totalmente Conectada

Como o próprio nome sugere, é a camada que está completamente conectada com a camada anterior. Ela sempre vem após uma camada convolucional ou de *pooling*, sendo necessário conectar cada elemento das matrizes de saídas em um neurônio de entrada. O objetivo desta camada é usar as características que são geradas das camadas anteriores para fazer a classificação dos dados de entrada, podendo ser imagem, texto ou som.

Nesta camada cada neurônio de uma camada anterior é conectada a cada um dos neurônios da próxima camada, essa camada é composta de uma última camada

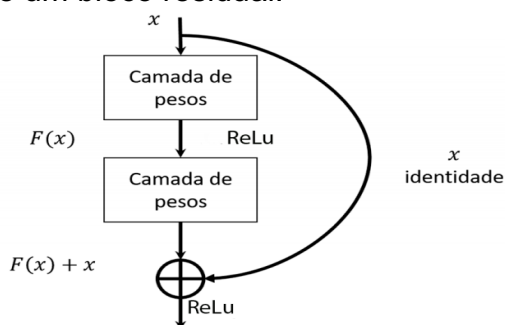
denominada camada de saída, onde cada neurônio representa uma classe do modelo, sendo assim, o número de neurônios da camada de saída representa o número de classes do modelo que está sendo treinado.

2.11.5 ResNet34

Pesquisadores da *Microsoft Research* desenvolveram as redes neurais residuais (*ResNet*). A rede venceu em primeiro lugar o ILSVRC 2015 entre as cinco melhores, com erro de 3,6% no conjunto de imagens da *ImageNet*. A ResNet-34 é uma rede residual profunda e o número 34 se refere à quantidade de camadas que ela possui. É uma subclasse de redes neurais convolucionais, junto com a *Resnet*, mais utilizada para classificação de imagens.

Inicialmente a *Resnet* era composta por 152 camadas e formada por blocos residuais. Os blocos residuais são utilizados para criar atalhos de uma camada para a outra. Como mostra a Figura 18, uma entrada x passa por vários processos de convolução, a formulação $F(x) + x$ pode ser feita por redes neurais *feedforward* que utilizam conexões de atalho, ou seja, uma camada pode pular para outra camada sem estar na sequência.

Figura 18 – Arquitetura de um bloco residual.



FONTE: (Adaptado de HE, K. et al. 2016)

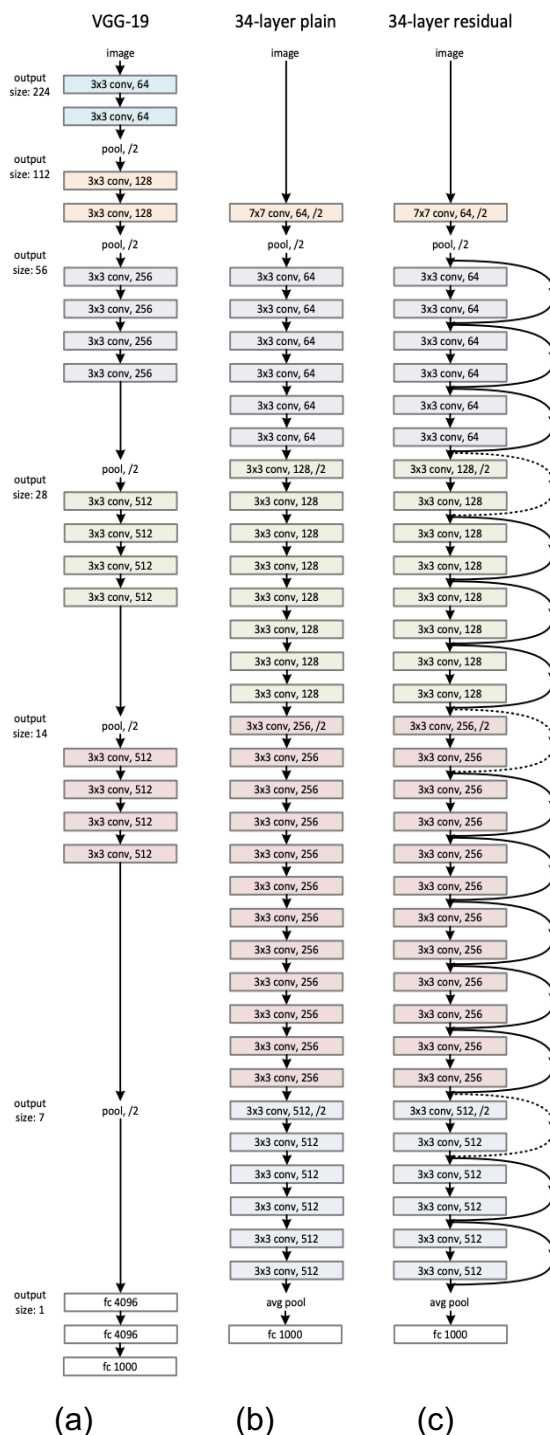
O resultado da $F(x)$ é adicionado à entrada x como pode ser visto da Equação 5.

$$H(x) = F(x) + x \quad (5)$$

Nas redes neurais convolucionais tradicionais, o resultado de $H(x)$ é igual ao $F(x)$, portanto, o espaço de saída é alterado em relação a entrada x . Na *ResNet*, a

função $F(x)$ funciona como um regularizador, ou seja, o espaço de saída é somente uma alteração do espaço de entrada. A Figura 19 ilustra uma arquitetura da Resnet34 além de comparar com outros tipos de arquiteturas.

Figura 19 – Arquiteturas de redes neurais convolucionais: (a) Arquitetura de uma VGG-19; (b) Arquitetura de uma rede simples com 34 camadas e (c) Arquitetura de uma rede com 34 camadas em blocos.



FONTE: (Adaptado de HE, K. et al. 2016)

2.12 TENSORFLOW

O desenvolvimento de uma rede neural é uma tarefa complexa, contendo quantidade distintas de camadas, onde ocorre o processamento dos dados para que seja obtido um resultado. Com o grande número de modelos de redes neurais implementadas, vários frameworks foram desenvolvidos para facilitar o treinamento das redes neurais, sendo o *tensorflow* uma ferramenta específica para esse caso.

O *tensorflow* foi criado pela equipe do *Google Brain*, o qual é um framework de código aberto, ou seja, todas as pessoas podem utilizá-lo de forma gratuita. É uma biblioteca multiplataforma podendo ser utilizado em diferentes sistemas operacionais, como por exemplo, *Windows*, *MacOS* e *Linux*. Sua utilização vem crescendo cada vez mais devido a grande necessidade de solucionar problemas com *machine learning* e *deep learning*. No *tensorflow* grafos são utilizados para realizar os cálculos dos dados que são inseridos nas redes neurais, cada nó do grafo é uma operação matemática e as arestas são os *tensors* (matrizes de dados multidimensionais) que são trocados entre os nós.

Com o uso desse *framework* é possível agilizar o processo de obter os dados, ou seja, grandes bases de dados estão nesse framework sendo de fácil utilização, treinar modelos com a própria elaboração de uma base de conhecimento, realizar previsões e refinar resultados. (TENSORFLOW, 2020). Essa biblioteca pode ser executada tanto na CPU quanto em GPUs, servidores ou dispositivos móveis.

O *tensorflow* é uma biblioteca que foi desenvolvida na linguagem de programação Python e C++, sendo sua arquitetura dividida em três partes principais (ARAUJO et al. 2017):

- Pré-processamento dos dados
- Construção dos modelos
- Treinamento e estimativas do modelo criado

2.13 FAST.AI

O *Fast.AI* é um framework de código aberto hospedado no GitHub onde todos podem utilizar. Foi desenvolvido para resolver problemas de *deep learning* de forma

simples e fácil. O fast.ai é um grupo de pesquisa que não possui fins lucrativos, sendo fundado em 2016.

Os desenvolvedores, desse framework, disponibilizam um curso online aberto, chamado “*Practical Deep Learning for Coders*” que significa Aprendizado Prático Profundo para Codificadores e para realizar esse curso não é necessário pré-requisitos, apenas ter o conhecimento da linguagem de programação *Python*.

2.14 CULTURA DA SOJA

A soja (*Glycine max* [L.] Merril) é uma planta que pertence à família Fabaceae com origem asiática, sendo utilizada na fabricação de diversos produtos, possuindo grande importância econômica e alimentícia. No final do século XV e início do século XVI a soja chegou ao ocidente, aproximadamente dois milênios após ser domesticada, por conta da agricultura chinesa não ter sido levada a outros lugares do mundo (HARLAN, 1992). Possui uma ótima adaptação a diferentes condições climáticas, possuindo diversas áreas cultivadas em todas as regiões do país, influenciando no desenvolvimento econômico nessas regiões (ROCHA et al., 2018).

Além de ser utilizada na alimentação e matéria prima como o óleo e o farelo, o grão de soja tem utilidade na produção de chocolates, massas e temperos. Também se tornou importante na fabricação de biodiesel, combustível utilizado por transportes elétricos. Esse combustível possui na sua composição o diesel que é derivada do petróleo e o óleo de soja, o qual é responsável por mais de 80% da demanda total de biodiesel no Brasil. (APROSOJA BRASIL, 2018).

No início da década de 1960, há registros sobre a produção de soja no Brasil, com crescimento acentuado até os dias de hoje. A Figura 20 ilustra a evolução da soja no Brasil.

Figura 20 – Evolução da produção de soja no Brasil de 1961 a 2018.



Fonte: (Adaptado de dados da FAO em 2016)

Segundo dados da CONAB (2019) as exportações da soja no primeiro semestre de 2019 foram de 56,14 milhões de toneladas, e nesse mesmo período no ano de 2018 foi de 64,60 milhões de toneladas. No segundo semestre de 2019 as exportações somaram em torno de 17,70 milhões de toneladas sendo que em 2018 foi de 18,66 milhões de toneladas.

Devido ao seu elevado grau de importância na alimentação, alguns procedimentos devem ser adotados para realizar a classificação dos grãos. A soja possui diferentes classes (defeitos) que devem ser classificados. O Ministério da Agricultura, Pecuária e Abastecimento (MAPA) determinam regras a serem seguidas, como a Instrução Normativa 11/2007 onde grãos de soja podem ser classificados em avariados, amassados, esverdeados, partidos e quebrados, matérias estranhas e impurezas, entre outros. O Quadro 5 mostra os tipos de grãos encontrados na cultura da soja.

Quadro 5 – Defeitos dos grãos avariados da soja.

(continua)

AVARIADOS	DEFINIÇÃO
Queimados	grãos ou pedaços de grãos carbonizados.
Ardidos	grãos fermentados em sua totalidade e com coloração marrom escura acentuada, afetando o cotilédone.
Mofados	grãos ou pedaços de grãos que se apresentam com fungos.

Quadro 5 – Defeitos dos grãos avariados da soja.

(conclusão)

AVARIADOS	DEFINIÇÃO
Fermentados	grãos ou pedaços de grãos que, em razão do processo de fermentação, tenham sofrido alteração visível na cor do cotilédono que não aquela definida para os ardidos.
Germinados	grãos ou pedaços de grãos que apresentam visivelmente a emissão da radícula.
Danificados	grãos ou pedaços de grãos que se apresentam com manchas na polpa alterados e deformados, perfurados ou atacados por doenças ou insetos, em qualquer de suas fases evolutivas.
Imaturos	grãos de formato oblongo, que se apresentam intensamente verdes, por não terem atingido seu desenvolvimento fisiológico completo e que podem se apresentar enrugados.
Chochos	grãos com formato irregular que se apresentam enrugados, atrofiados e desprovidos de massa interna.

FONTE (MAPA, 2018)

Ao avaliar a revisão da literatura e os trabalhos correlatos, pode-se observar a importância de um padrão de metadados para recuperar informações de forma mais rápida, para tomada de decisão e, métodos computacionais para classificação de grãos com o intuito de diminuir o tempo demandado para a realização da classificação manualmente realizado por trabalhadores da área. Com este método de classificação qualquer pessoa que tenha interesse em realizar a classificação de grãos poderá exercer a prática na área não necessitando de conhecimentos prévios.

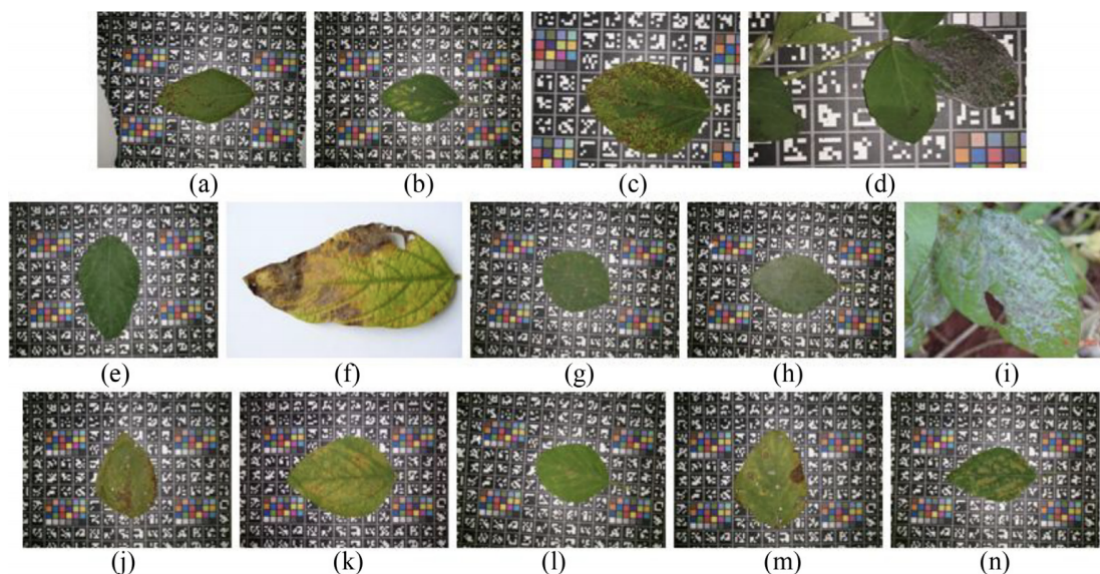
3 TRABALHOS CORRELATOS

Assim como existem trabalhos sobre classificação de grãos em imagens, pode-se classificar diferentes imagens digitais dependendo do problema a ser solucionado. As técnicas de computação vem auxiliando cada vez mais na busca de soluções para diversos ramos na pesquisa. As redes neurais e os padrões de metadados são ferramentas que auxiliam e melhoram a vida de diversas pessoas que precisam realizar uma tarefa de forma rápida e com bons resultados. A seguir será citado alguns trabalhos que utilizam essas ferramentas em áreas parecidas e distintas.

Com o aumento da população em 9 bilhões em 2050, segundo estudos, a procura por alimentos aumentaria em 60%. Aumentar a produção e a qualidade de grãos é importante para suprir a demanda. Muitas doenças acabam causando estragos no desenvolvimento de uma cultura, podendo diminuir drasticamente a colheita. No trabalho de (KARLEKAR; SEAL, 2020) foi adotado uma abordagem computacional com dois módulos para classificar doenças em folhas de soja.

O primeiro módulo extrai parte da folha da imagem e subtrai o fundo. No segundo módulo é utilizado uma rede neural de convolucional para o reconhecimento de doenças nas folhas de soja. As imagens das plantas de soja foram do banco de dados de imagens de doenças de plantas sintomas (PDDB), que consiste em 486 imagens no padrão RGB, consistindo em 3 categorias: saudável, infectado e desconhecido. As folhas da classe saudável são divididas em 14 subcategorias, sendo todas as imagens com as mesmas dimensões.

Figura 21 – Folhas de soja do PDDDB: (a) peste bacteriana, (b) vírus do mosaico da soja, (c) fitotoxicidade do cobre, (d) podridão de carvão, (e) saudável, (f) folha cercospora, (g) míldio penugento, (h) oídio, (i) oídio e ferrugem, (j) ferrugem, (k) ferrugem e ponto, (l) mancha marrom, (m) praga do sul, (n) desconhecido.



FONTE: (KARLEKAR; SEAL, 2020).

Portanto foram utilizadas 16 categorias neste trabalho. O modelo de rede neural convolucional proposto SoyNet atingiu uma precisão de 98,14%, sendo comparada com outros modelos de rede, como a VGG19, GoogleLeNet, Dense121, XceptionNet, LeNet e ResNet50.

Deslizamentos de terra e inundações são fenômenos da natureza que causam perda de vidas e danos econômicos em um país. Com a utilização de modelos é possível entender a frequência e ocorrência desses fenômenos. Porém, para que sejam desenvolvidos modelos, os recursos devem ser significativos e disponibilizados. Neste trabalho (RIDDICK et al., 2019) foi feito um esquema de metadados através de um modelo de questionário. Esse esquema utilizou padrões já existentes, como base do novo padrão.

Primeiramente, os autores deste projeto distribuíram um questionário online para trezentos e vinte pessoas interessadas de universidades, organizações comerciais e outras organizações de pesquisa. Das 320 apenas 108 respostas foram recebidas, das quais foram analisadas por especialistas na modelagem ambiental. Através dos questionários foi possível perceber que há uma falta de padrões de metadados sobre modelos ambientais, e quais atributos eram mais relevantes para o

desenvolvimento do modelo de metadados. O modelo desenvolvido foi implementado no portal PURE o qual disponibiliza modelos para metadados.

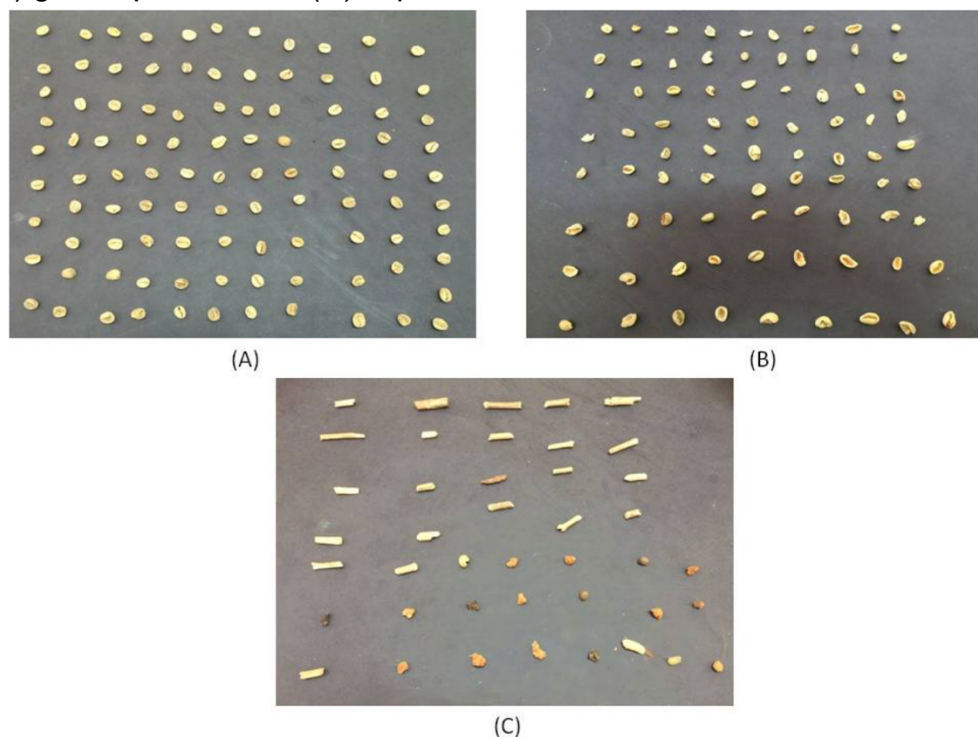
Grande parte da produção de cereais é destinado ao processamento agroalimentar, implicando o desenvolvimento de pesquisas e sistemas modernos para armazenagem dos cereais. Grãos armazenados necessitam de cuidados, pois, muitos fatores podem causar danos, como a cariopse causada pelo gorgulho do grão (*Sitophilus granaries* L).

Neste trabalho (BONIECKI et al., 2020) foi identificado as características físicas da cariopse de trigo afetada pelo gorgulho dos grãos. Foi desenvolvido um sistema visual automatizado para a detecção dos grãos danificados e gorgulhos adultos, os dados referentes aos grãos (massa, diâmetro, umidade e dureza) foram utilizados para construir o conjunto de treinamento das redes neurais. Através dos testes conclui-se que as redes neurais artificiais podem ser usadas como ferramentas para a identificação de pragas, sendo a topologia FBR (função de base radial) a melhor topologia de rede neural gerada, as quais são redes aplicadas na aproximação não linear de variáveis numéricas.

A produção de café no Brasil possui alta importância devido ao fato de ser exportada para diversos países, onde é feita a distribuição do produto em todo o mercado. Quando realizada a colheita os grãos são armazenados para serem lavados e realizar a limpeza, removendo impurezas, como folhas, pedras, galhos entre outros materiais que podem estar presente. Após os procedimentos de limpeza, é feita a classificação dos grãos em diferentes classes, como por exemplo: grãos quebrados, impuros etc.

No trabalho de (OLIVEIRA et al., 2019) foi feita a classificação de grãos de café normais, quebrados e impurezas utilizando rede neural convolucional e redes neural artificial do tipo *Perceptron* Multicamadas. A base de imagens foi preparada em laboratório, onde os autores utilizaram uma câmera para a captura das imagens. A Figura 22 mostra as classes de grãos que foram estudadas neste trabalho.

Figura 22 – Grãos de Café sobre um papel fosco para a classificação: (A) Grãos normais; (B) grãos quebrados e (C) impurezas.



FONTE: (OLIVEIRA et al., 2019)

Com a CNN obteve-se uma acurácia de 96,60%, resultado superior ao *Perceptron* Multicamadas que teve uma acurácia de 78,13%, onde foram realizadas 30 execuções dos algoritmos. O uso de processamento digital de imagens foi utilizado no *Perceptron* Multicamadas para extrair informações relevantes para a classificação.

A aveia é um cereal utilizado tanto na alimentação humana quanto na alimentação animal. É um cereal que possui diversos benefícios como por exemplo seu elevado percentual de fibra alimentar. Por esse motivo, este cereal é muito utilizado em dietas.

No trabalho de (PATRÍCIO, 2019) foi utilizado técnicas de aprendizado profundo em imagens digitais para duas finalidades. A primeira foi usada para identificar as espécies *Avena sativa* e *Avena strigosa*, e a segunda finalidade foi classificar os grãos para a espécie *Avena sativa*. Para a aquisição das imagens foi utilizado um scanner fotográfico, onde a superfície dos grãos de aveia não tiveram contato direto com o vidro do scanner por possuírem uma forma cilíndrica.

Foram utilizadas diferentes redes neurais como a LeNet5, AlexNet, VGG16, InceptionV3, Resnet e a DenseNet21. Entretanto, a rede que obteve a maior acurácia

foi a DenseNet com 99,7% para identificação das espécies de aveia e 89,7% para classificação da cultivar de aveia.

Zapotoczny et al. (2020) utilizaram grãos de cevadas infectados pelo fungo *Fusarium* e foram visualizados em luz polarizada e as imagens obtidas foram analisadas utilizando métodos de reconhecimento de padrões e redes neurais. As imagens dos grãos de cevadas concentrou-se em: grãos saudáveis, grãos com ponta preta e grãos afetados por mofo, sendo obtido 100 imagens para cada classe, totalizando 300 imagens.

Para realizar a classificação, foi utilizado um *perceptron* multicamada e uma função de base radial, sendo aplicados 12 neurônios de entrada, uma camada oculta e 3 saídas onde 70% das imagens foram destinadas a teste e 30% para treino (ZAPOTOCZNY et al., 2020). Segundo Zapotoczny et al., no conjunto de teste dos grãos afetados por mofo obteve um acerto 100% e nos demais grupos ocorreu uma variação de 75% a 88% pelo fato de ter ocorrido uma classificação incorretamente entre esses grupos.

4 MATERIAIS E MÉTODOS

Para solucionar o problema de classificação dos grãos de soja da espécie *Glycine max*, foram estudados artigos e demais materiais bibliográficos que tratam sobre padrões de metadados, classificação de imagens utilizando redes neurais e processamento digital de imagens.

A base de conhecimento utilizada neste trabalho foi coletada durante o curso realizado no Serviço Nacional de Aprendizagem Rural (SENAR) e não foram submetidas ao processamento de mudança de cores. As imagens foram redimensionadas devido a escolha da rede neural necessitar de dimensões específicas para que se possa utilizar a rede.

Com formas e cores distintas é possível classificar objetos em imagens. Portanto, nas imagens dos grãos de soja a identificação das diferentes classes utilizadas neste trabalho foram feitas através das formas e cores. As redes neurais possuem métodos nas suas camadas onde extraem características, sem que isso seja feito de maneira manual.

Neste capítulo são apresentados os métodos utilizados para a definição dos metadados para a criação do padrão de metadados para a classificação dos grãos. Também é apresentado a criação da base de dados que foi utilizada nas redes neurais convolucionais para desenvolver um método computacional para a classificação de grãos de soja.

4.1 AQUISIÇÃO DAS IMAGENS

Na Seção 2.6 foi mostrado as etapas do processamento digital de imagens (MARQUES FILHO, VIEIRA NETO, 1999), mostrando a primeira etapa, uma das mais importantes, sendo a aquisição de imagens. Para obter as imagens dos grãos de soja, primeiramente foi feito um curso de classificação de grãos no SENAR no período de 07/10/2019 à 09/10/2019, sendo que a quantidade em gramas analisadas dos grãos de soja foram de 170 gramas. A Figura 23 mostra os grãos de soja sendo classificados manualmente.

Figura 23 – Grãos de soja sobre EVA preto para a classificação.



Fonte: A autora

Após feita a classificação dos grãos, as imagens foram obtidas. A quantidade em gramas obtidas dos grãos e seus devidos defeitos ou, classes são mostrada na Tabela 2.

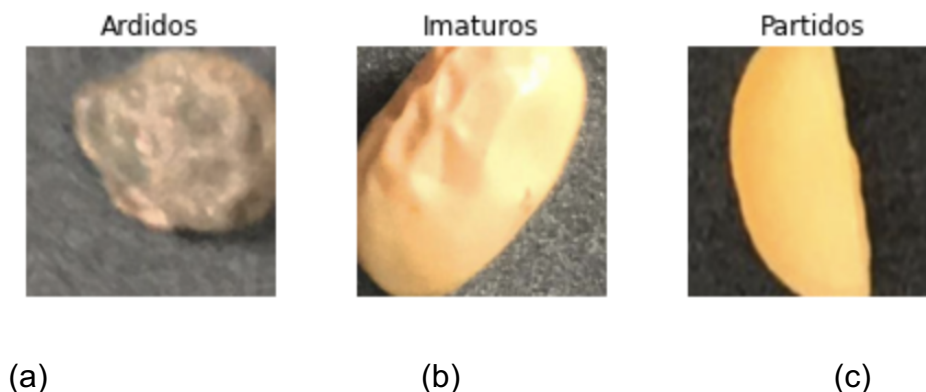
Tabela 2 – Valores dos defeitos em gramas encontradas nos grãos de soja.

CLASSES	QUANTIDADE EM GRAMAS (g)
Ardidos	0,4
Mofados	8,4
Fermentados	58,1
Germinados	0,4
Imaturos	0,4
Esverdeados	0,3
Partidos e Quebrados	2
MEI	1,6

Fonte: A autora

A Figura 24 mostra grãos de soja separada nas classes: Ardidos, Imaturos e Partidos.

Figura 24 – Imagens de soja com defeitos: (a) grão de soja ardido; (b) grão de soja imaturo e (c) grão de soja partido.



Fonte: A autora

Com a falta de uma base de dados pronta com imagens dos variados defeitos dos grãos de soja, fica uma tarefa desafiadora para que se possa obter um classificador satisfatório, sendo que é necessário capturar um número considerável de imagens. Para separar os grãos em uma única imagem foram feitos recortes manualmente e recortes com a utilização de um algoritmo em *Python*. No total foram realizados 21 (vinte e um) recortes para cada classe. O número das imagens coletadas e suas classes são mostradas na Tabela 3.

Tabela 3 – Número de imagens coletadas por classe.

CLASSES	QUANTIDADE EM GRAMAS (g)
Ardidos	21
Mofados	21
Fermentados	21
Germinados	21
Imaturos	21
Esverdeados	21
Partidos e Quebrados	21
MEI	21
Normais	21

Fonte: A autora

Foram coletadas 189 imagens, com média de 21 imagens por classe. Como mostra a Tabela 3, apenas 9 classes foram classificadas neste trabalho devido a dificuldade de encontrar grãos com os defeitos citados anteriormente.

Com o número de imagens coletados, a partir de 16 imagens de cada classe foram criadas 756 imagens novas (84 imagens novas para cada classe) obtendo um total de 100 imagens por classes, utilizando *data augmentation* que é um técnica utilizada para gerar novos dados, com o intuito de aumentar a generalidade do modelo. As outras 5 imagens foram separadas para realizar predições no modelo.

Figura 25 – Script da criação das imagens com as transformações.

```
import Augmentor

pipe = Augmentor.Pipeline("Partidos")
pipe.rotate(probability=1, max_left_rotation=5, max_right_rotation=5)
pipe.flip_left_right(probability=0.5)
pipe.zoom_random(probability=0.5, percentage_area=0.8)
pipe.sample(84)

Processing <PIL.Image.Image image mode=RGB size=212x176 at 0x1485836A0>:
8%|██████████| 7/84 [00:00<00:03, 25.41 Samples/s]

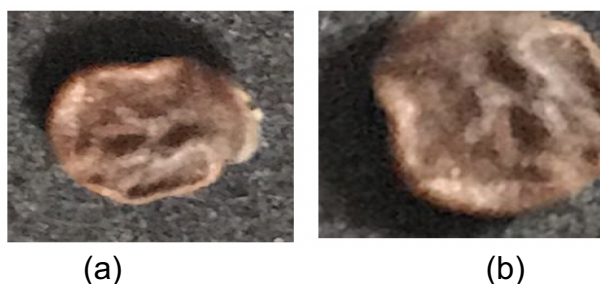
Initialised with 16 image(s) found.
Output directory set to Partidos/output.

Processing <PIL.Image.Image image mode=RGB size=212x212 at 0x14859C940>:
100%|██████████| 84/84 [00:00<00:00, 152.52 Samples/s]
```

Fonte: A autora

Como mostra a Figura 25, foi aplicado a técnica de *data augmentation* em 16 imagens de cada classe, onde passaram por transformações de rotação, inversão de lados e aproximação.

Figura 26 – Grão de soja ardido: (a) Imagem original. (b) Imagem criada com as transformações.



Fonte: A autora

Como pode ser visto na Figura 26(a) temos uma imagem de grão de soja ardido que passou pela técnica de *data augmentation*, gerando uma nova imagem (Figura 25(b)). Também foi feita a classificação dos grãos de soja sem o uso desta técnica.

4.2 SISTEMA COMPUTACIONAL

O sistema de processamento desenvolvido para classificar os defeitos ou classes dos grãos de soja, foi implementado na linguagem de programação orientada à objetos, *Python* na versão 3.7.3.

A plataforma de desenvolvimento adotada foi o *Google Colab* que é um ambiente de *notebooks jupyter* que não requer configuração e é executado na nuvem e que auxilia na visualização dos dados e resultados de análises juntamente com o código e o computador em uso, possui um processador Intel i5, com memória RAM de 8GB e sistema operacional MACOS.

A plataforma de aprendizagem profunda utilizada foi o *Tensorflow* e o *Fast.ai* para treinar e testar as redes neurais. Essas redes neurais foram previamente treinadas com um conjunto de imagens da *ImageNet*. Quando a aprendizagem profunda é iniciada do zero, seu treinamento pode levar dias para ser concluído, portanto, reutilizando os modelos por meio do aprendizado de transferência, pode-se fazer o treinamento em pouco tempo, dependendo da base de imagens. Para utilizar as imagens de grãos de soja, elas foram recortadas e redimensionadas para 224 x 224 pixels, tornando-as compatíveis com as dimensões originais das imagens da base *ImageNet*, para serem utilizadas nas redes Resnet34, MobileNet e VGG19. A Figura 27 mostra o código desse procedimento.

Figura 27 – Código em Python para o recorte e redimensionamento das imagens.

```
def cropped_images(imagens):
    imagens_novas = []
    for img in imagens:
        h, w = img.shape[:2]
        if (h > w):
            aux = int((h-w)/2)
            img = img[aux:w+aux, 0:w]
        else:
            aux = int((w-h)/2)
            img = img[0:h, aux:h+aux]

        imagem = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        imagem = cv2.resize(imagem, (224,224))
        imagens_novas.append(imagem)

    return imagens_novas
```

Fonte: A autora

A base de dados da *ImageNet* contém mais de 1.000 classes, somando mais de 14 milhões de imagens, portanto, o modelo treinado foi adaptado para a base de imagens de grãos de soja que contém 9 classes. Foram testadas diferentes taxas de aprendizado, do inglês *learning rate*, para chegar na melhor taxa de acerto, sendo que, por padrão, as redes neurais utilizam 0.001.

4.3 METADADOS

A definição dos metadados para a criação do modelo do padrão de classificação de grãos foram levantadas com base em:

- Curso técnico de classificação de grãos de forma manual realizado no SENAR na cidade de Ponta Grossa, onde o objetivo do curso visou ensinar métodos de classificação de diferentes grãos, além de citar as doenças encontradas nos mesmos.
- Estudo comparativo dos padrões de metadados encontrados na literatura.

Pode-se definir cada grão como sendo um objeto e cada característica deste objeto como metadado. Este modelo de padrão de metadados para classificação de grãos ajuda no processo de recuperação dos objetos com suas devidas características, proporcionando vantagens no tempo demandado em operações distintas relacionadas ao objeto com seus metadados. Além disso, objetos já existentes com suas informações poderão ser herdados para o cadastro de outros objetos, sendo viável o compartilhamento dos metadados.

O Quadro 6 apresenta os metadados definidos em relação às características dos grãos, onde é mostrada a descrição simbólica para a construção do padrão de metadados para grãos. Além dos metadados, valores que os mesmos podem assumir.

Quadro 6 – Metadados das características dos grãos

(continua)

METADADOS	VALORES
Grão	Feijão, soja, milho, etc
Cor	Amarelo, preto, marrom, etc
Formato	Redondo, oval, etc
Tamanho	Tamanho

Quadro 6 – Metadados das características dos grãos.

(conclusão)

METADADOS	VALORES
Peso	Peso
Tecido	Endosperma, cotilédones, etc
Defeitos	Fermentados, ardidos, etc
Doença	Fungo, insetos, etc

Fonte: A autora

Os metadados propostos presentes no Quadro 6, podem possuir diferentes valores, sendo que esses valores estão relacionados com o tipo de grão que será cadastrado, podendo ser feijão, soja, milho, etc.

a) Grão:

- descrição: nome dado ao tipo de grão que será classificado;
- descrição eletrônica: <META NAME = "GM.Grao" CONTENT = "Milho">;

b) Cor:

- descrição: nome dado a cor do grão;
- descrição eletrônica: <META NAME = "GM.Cor" CONTENT = "Amarelo">;

c) Formato:

- descrição: nome dado a forma do grão;
- descrição eletrônica: <META NAME = "GM.Formato" CONTENT = "Padrão">;

d) Tamanho:

- descrição: nome dado a medida do comprimento do grão;
- descrição eletrônica: <META NAME = "GM.Tamanho" CONTENT = "5 centímetros">;

e) Peso:

- descrição: nome dado ao peso do grão podendo ser em qualquer unidade de medida;
- descrição eletrônica: <META NAME = "GM.Peso" CONTENT = "150 gramas">;

f) Tecido:

- descrição: nome dado ao tipo de revestimento do grão;

- descrição eletrônica: <META NAME = "GM.Tecido" CONTENT = "Cotilédone">;

g) Defeitos:

- descrição: nome dado aos defeitos encontrados no grão
- descrição eletrônica: <META NAME = "GM.Grao" CONTENT = "Fermentado">;

h) Doença:

- descrição: nome dado às doenças que os grãos podem desenvolver;
- descrição eletrônica: <META NAME = "GM.Grao" CONTENT = "Fungo">;

Em relação à sintaxe utilizada, a tag META NAME utilizada na linguagem de marcação HTML tem como objetivo fornecer um padrão que resulta na precisão da busca pelos grãos em páginas web. Assim como é utilizado no padrão Dublin Core o prefixo DC, na sintaxe para a descrição dos grãos é utilizado o prefixo GM (Grão Metadados) que é o prefixo na tag <link>. A seguir a representação dos metadados na linguagem HTML.

```
<META NAME = "GM.Grao" CONTENT = "Milho">
<META NAME = "GM.Cor" CONTENT = "Amarelo">
<META NAME = "GM.Formato" CONTENT = "Padrão">
<META NAME = "GM.Tamanho" CONTENT = "5 centímetros">
<META NAME = "GM.Peso" CONTENT = "150 gramas">
<META NAME = "GM.Tecido" CONTENT = "Cotilédone">
<META NAME = "GM.Grao" CONTENT = "Fermentado">
<META NAME = "GM.Grao" CONTENT = "Fungo">
```

Os metadados podem ser representados na linguagem XML. A seguir uma estrutura em XML representando a classificação de grãos.

```
<ListaGraos>
  <Grao>
    <Nome name="Nome">Milho</Nome>
    <Cor name="Cor">Amarelo</Cor>
    <Formato name="Formato">Padrão</Formato>
    <Tamanho name="Tamanho">5 centímetros</Tamanho>
    <Peso name="Peso">150 gramas</Peso>
```

```

<Tecido name="Tecido">Cotilédone</Tecido>
<Defeito name="Defeito">Fermentado</Defeito>
<Doença name="Doença">Fungo</Doença>
</Grao>
</ListaGraos>

```

Além dos metadados de características dos grãos, podemos definir alguns metadados relacionados com a região em que os grãos foram cultivados.

Quadro 7 – Metadados relacionados à região dos grãos.

METADADOS	VALORES
Cidade	Ponta Grossa, Florianópolis, etc
Estado	Paraná, Santa Catarina, etc
País	Brasil, Estados Unidos, etc
Região	Norte, Sul, etc
Estação	Verão, Inverno, etc

Fonte: A autora

Conforme mostra o Quadro 7 os grãos podem ser cultivados em diferentes lugares, sendo necessário criar metadados que recuperem as informações conforme a região do grão. A seguir, os metadados sendo representado na linguagem HTML e XML respectivamente.

a) Cidade:

- descrição: nome dado a região o grão foi cultivado;
- descrição eletrônica: <META NAME = "GM.Cidade" CONTENT = "Ponta Grossa">;

b) Estado:

- descrição: nome dado ao estado onde o grão foi cultivado;
- descrição eletrônica: <META NAME = "GM.Estado" CONTENT = "Paraná">;

c) País:

- descrição: nome dado ao país de origem onde o grão foi cultivado;
- descrição eletrônica: <META NAME = "GM.Pais" CONTENT = "Brasil">;

d) Região:

- descrição: nome dado onde o grão foi cultivado;
- descrição eletrônica: <META NAME = "GM.Regiao" CONTENT = "Sul">;

e) Estação:

- descrição: nome dado em qual estação o grão foi cultivado;
- descrição eletrônica: <META NAME = "GM.Estacao" CONTENT = "Primavera">

Metadados na representação HTML:

<META NAME = "GM.Cidade" CONTENT = "Ponta Grossa">

<META NAME = "GM.Estado" CONTENT = "Paraná">

<META NAME = "GM.Pais" CONTENT = "Brasil">

<META NAME = "GM.Regiao" CONTENT = "Sul">

<META NAME = "GM.Estacao" CONTENT = "Primavera">

Metadados na representação XML:

<ListaGraos>

<Grao>

<Cidade name="Cidade">Ponta Grossa</Cidade>

<Estado name="Cor">Paraná</Estado>

<Pais name="Formato">Brasil</Pais>

<Regiao name="Tamanho">Sul</Regiao>

<Estacao name="Peso">Primavera</Estacao>

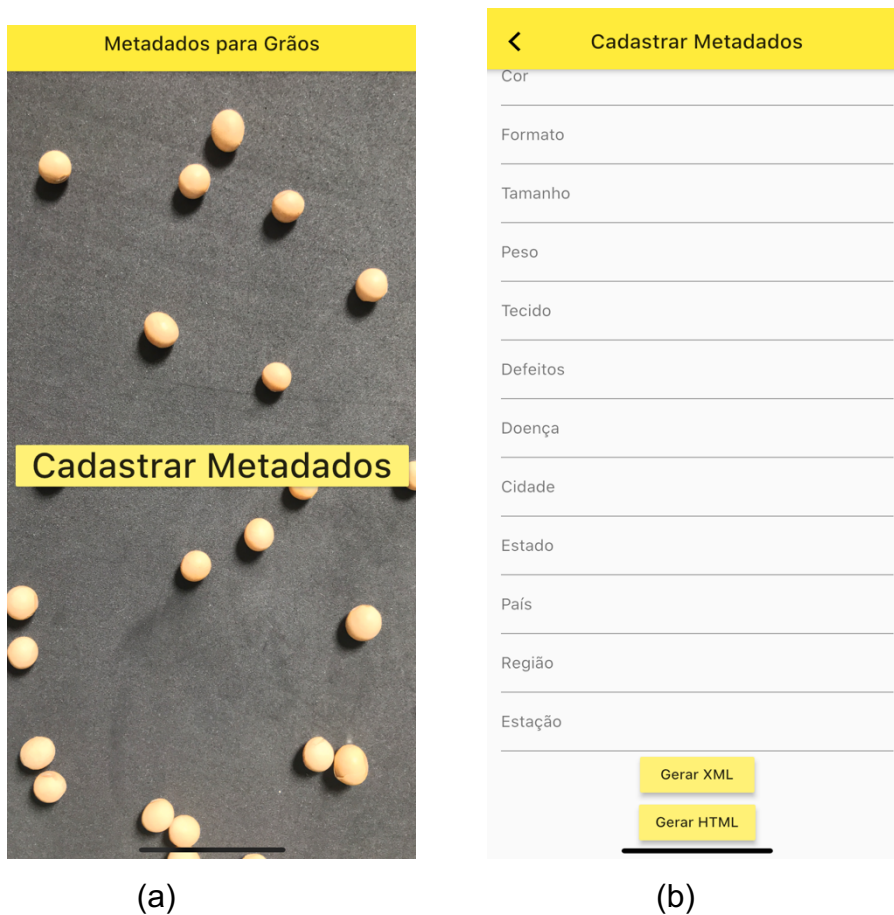
<Grao>

</ListaGraos>

Através da estrutura XML os metadados são representados, facilitando o entendimento, pois, podem ocorrer elementos com valores multivalorados, relações de hierarquia, etc.

Com os metadados definidos baseados em normas do MAPA e através do curso realizado no SENAR foi desenvolvido um aplicativo para auxiliar os usuários da área a gerarem seus metadados na estrutura XML e HTML. A Figura 28(a) mostra a tela inicial da ferramenta e Figura 28(b) mostra os campos dos metadados criados onde o usuário preenche todos os campos e escolhe se deseja representar os elementos na linguagem XML ou HTML, podendo ser gerado nas duas linguagens caso necessário.

Figura 28 – (a) Tela inicial do aplicativo e (b) Tela de cadastro dos metadados.



Fonte: A autora

O intuito do aplicativo que foi desenvolvido é ser fácil o seu uso e intuitivo para que todas as pessoas possam utilizar, principalmente as pessoas que são especialistas na área. As Figuras 29(a) e 29(b) mostram o resultado dos metadados inseridos nas estruturas XML e HTML.

Figura 29 – (a) Estrutura dos metadados em XML e (b) Estrutura dos metadados em HTML.

< Metadados em XML	< Metadados em HTML
<pre> <ListaGraos> <Grao> <Nome name="Nome">Milho</Nome> <Cor name="Cor">Amarelo</Cor> <Formato name="Formato">Padrão </Formato> <Tamanho name="Tamanho">5 centímetros </Tamanho> <Peso name="Peso">150 gramas</Peso> <Tecido name="Tecido">Cotilédone </Tecido> <Defeito name="Defeito">Fermentado</Defeito> <Doenca name="Doenca">Fungo</Doenca> </Grao> </ListaGraos> </pre>	<pre> <META NAME = "GM.Grao" CONTENT = "Milho"> <META NAME = "GM.Cor" CONTENT = "Amarelo"> <META NAME = "GM.Formato" CONTENT = "Padrão "> <META NAME = "GM.Tamanho" CONTENT = "5 centímetros "> <META NAME = "GM.Peso" CONTENT = "150 gramas"> <META NAME = "GM.Tecido" CONTENT = "Cotilédone "> <META NAME = "GM.Defeitos" CONTENT = "Fermentado"> <META NAME = "GM.Doenca" CONTENT = "Fungo"> <META NAME = "GM.Cidade" CONTENT = "Ponta Grossa"> <META NAME = "GM.Estado" CONTENT = "Paraná "> <META NAME = "GM.Pais" CONTENT = "Brasil"> <META NAME = "GM.Regiao" CONTENT = "Sul"> <META NAME = "GM.Estacao" CONTENT = "Primavera"> </pre>

(a)

(b)

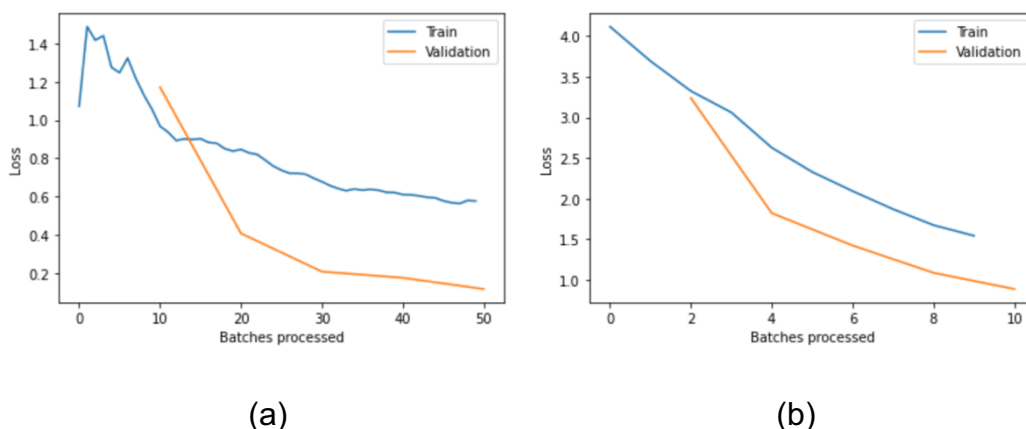
Fonte: A autora

Como é visto nas Figuras 29(a) e 29(b) preenchendo todos os campos dos metadados no aplicativo é gerado um código de metadados sendo sua saída em XML ou HTML.

5 RESULTADOS E DISCUSSÃO

A rede neural convolucional utilizada foi a Resnet34 que funciona por blocos, obteve uma taxa de 97,77% de acerto com a utilização da técnica de *data augmentation* e 80,85% de acerto sem essa técnica com um *learning rate* de 10^{-3} . A porcentagem de imagens reservadas para teste, ou seja, para a validação da rede treinada foi de 25%, essas imagens não foram utilizadas no processo de treinamento da rede, sendo seu uso restrito na etapa de validação.

Figura 30 – Resultados da porcentagem de treino e validação do modelo. (a) Resultados com *data augmentation*. (b) Resultados sem *data augmentation*.



Fonte: A autora

Conforme mostra os gráficos acima (Figura 30(a) e 30(b)) temos um caso que não é comum de acontecer, porém possível, onde o erro de validação subestima o erro de treinamento. Isso acontece por alguns motivos: o primeiro é que a base de dados não possui um grande número de imagens, sendo assim, o modelo não consegue aprender caso ocorra diferentes situações. O segundo motivo se dá ao fato das imagens de treino serem muito parecidas com as imagens de teste, como por exemplo podemos citar o fundo das imagens, onde neste trabalho foi utilizado um EVA preto, porém a cor do fundo não traria uma diferença significativa para o modelo, podendo ser qualquer outra cor.

Entretanto, os modelos se saíram muito bem para esta base de dados seguindo o padrão que foi realizado neste trabalho. Como pode ser visto na Figura 30(a) e 30(b) a perda foi menor com o uso da técnica de *data augmentation*, sendo assim, a acurácia do modelo com o uso dessa técnica foi maior. Por fim, a acurácia e o erro do

modelo obtiveram valores melhores são considerados bons resultados, uma vez que a base de imagens não possui um número significativo de imagens.

Tabela 4 – Resultados da rede neural com *data augmentation*.

Épocas	Perda no Treino	Perda na Validação	Acurácia	Taxa de Erro	Tempo
0	1.057.069	1.172.182	0.844444	0.155556	00:08
1	0.837853	0.407960	0.928889	0.071111	00:08
2	0.695821	0.207992	0.960000	0.040000	00:08
3	0.621038	0.175690	0.960000	0.040000	00:08
4	0.577147	0.116773	0.977778	0.022222	00:08

Fonte: A autora

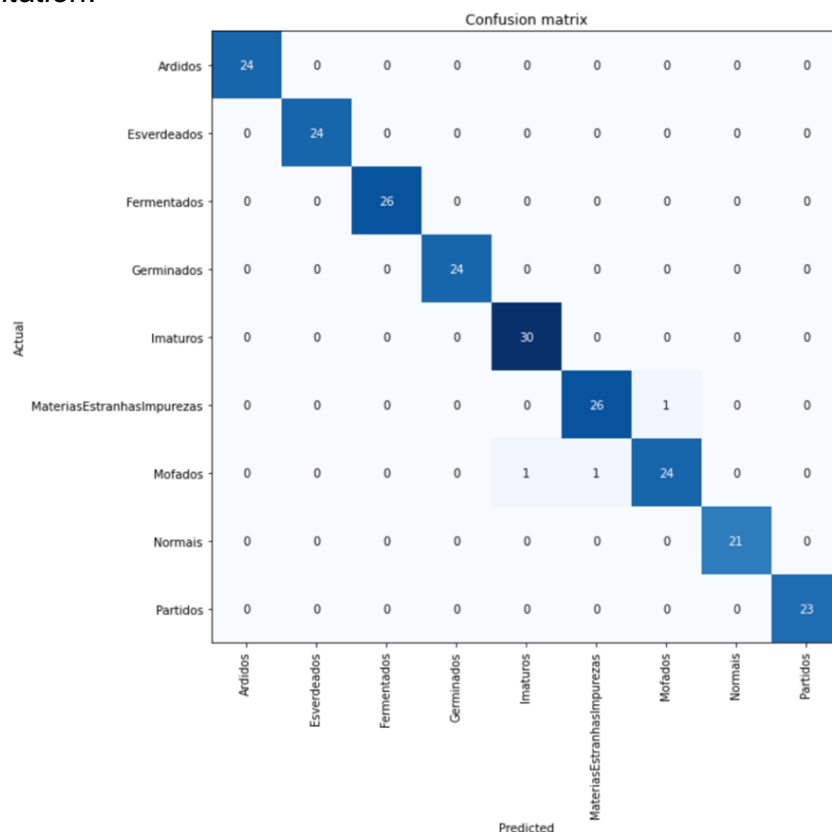
Tabela 5 – Resultados da rede neural sem *data augmentation*.

Épocas	Perda no Treino	Perda na Validação	Acurácia	Tempo
0	0.334442	1.044.622	0.659574	00:02
1	0.319128	0.940667	0.680851	00:02
2	0.291233	0.881485	0.744681	00:02
3	0.255072	0.768678	0.765957	00:02
4	0.234375	0.680335	0.808511	00:02

Fonte: A autora

Com o decorrer das etapas de treinamento, a acurácia do modelo tende a aumentar (Tabela 4 e Tabela 5). Ao término de cada etapa de treinamento uma avaliação é executada, dessa maneira, é possível executar a tarefa de classificação em imagens que ainda não foram analisadas pelo modelo. Assim, fica claro como o modelo irá se comportar.

Figura 31 – Matriz de confusão de 225 imagens usadas no teste com a técnica de *data augmentation*.



Fonte: A autora

Ao avaliarmos a matriz de confusão (Figura 31) vemos que a classe Mofados foi a que apresentou mais erros que as outras classes, onde a rede classificou 1 vez a classe como sendo Matérias Estranhas e Impurezas, e 1 vez como sendo Imaturos. A classe Mofados foi a que apresentou o menor índice de acerto quando comparada às outras classes, obtendo 92,30% de assertividade.

Como mostra a Tabela 6, a classe Matérias Estranhas e Impurezas obteve índice de 96,29% de assertividade, onde apenas 1 vez foi classificada erroneamente em Mofados. Já as classes: Ardidos, Esverdeados, Fermentados, Germinados, Normais e Partidos, obtiveram um índice de 100% de assertividade.

Tabela 6 – Assertividade por classe com a técnica de *data augmentation*.
(continua)

Classes	Índice de Assertividade
Ardidos	100%
Esverdeados	100%

Tabela 6 – Assertividade por classe com a técnica de *data augmentation*.
(conclusão)

Classes	Índice de Assertividade
Fermentados	100%
Germinados	100%
Imaturos	100%
Normais	100%
Partidos	100%
Matérias Estranhas e Impurezas	96,29%
Mofados	92,30%

Fonte: A autora

Na matriz de confusão foram testadas 225 imagens variadas nas diferentes classes apresentadas. Conforme podemos visualizar, a somatória das classificações incorretas foi de 3. Esse número corresponde a 1,33% do total de imagens reservadas para realização de testes.

Tabela 7 – Porcentagem incorreta por classes com a técnica de *data augmentation*.

Classes	Número de Imagens	Erros	Porcentagem
Ardidos	24	0	0%
Esverdeados	24	0	0%
Fermentados	26	0	0%
Germinados	24	0	0%
Imaturos	30	0	0%
Normais	21	0	0%
Partidos	23	0	0%
Matérias Estranhas e Impurezas	27	1	3,70%
Mofados	26	2	7,69%

Fonte: A autora

A Tabela 7 mostra 3 erros de classificação de imagens, 1 erro aconteceu na classificação da classe de Matérias Estranhas e Impurezas que contém 27 imagens e 2 erros aconteceram na classe Mofados, com um total de 26 imagens. A porcentagem do erro foi de 3,7% e 7,69, respectivamente.

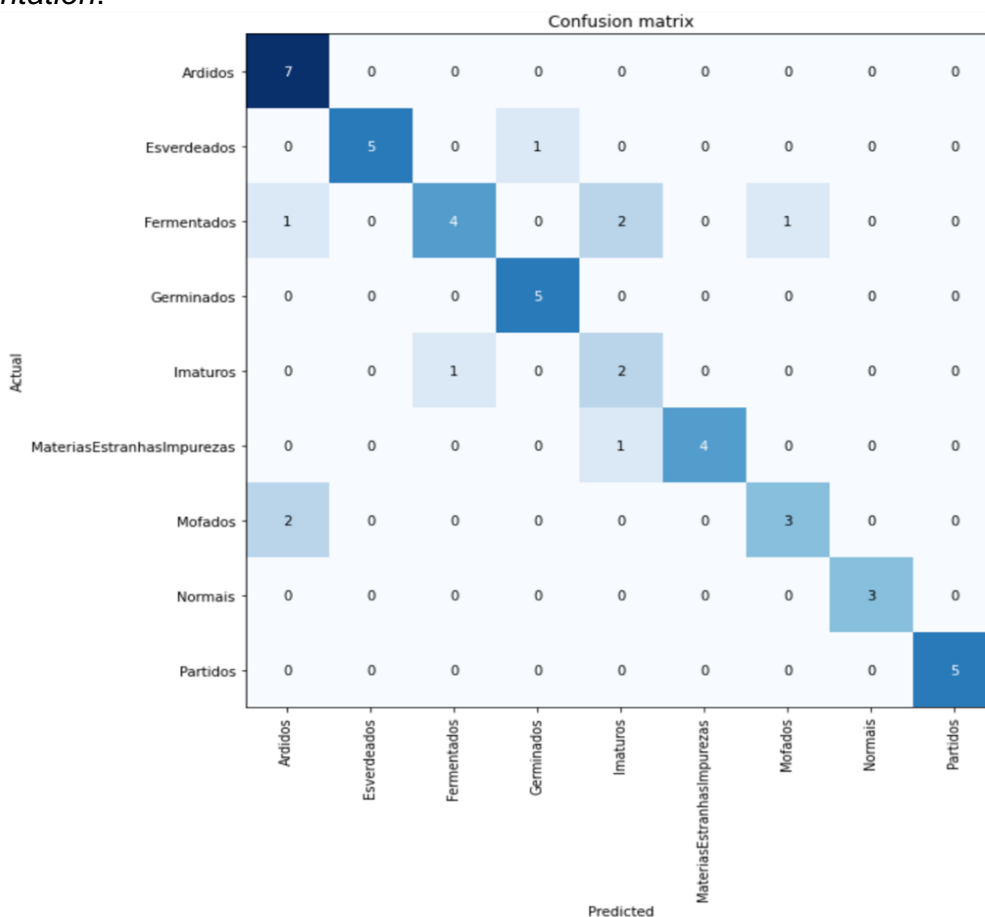
Tabela 8 – Porcentagem de acertos por classes com a técnica de *data augmentation*

Classe	Número de Imagens	Acertos	Porcentagem
Ardidos	24	0	100%
Esverdeados	24	0	100%
Fermentados	26	0	100%
Germinados	24	0	100%
Imaturos	30	0	100%
Normais	21	0	100%
Partidos	23	0	100%
Matérias Estranhas e Impurezas	27	1	96,29%
Mofados	26	2	92,30%

Fonte: A autora

A Tabela 8 mostra a porcentagem de acertos por classes, onde apenas as Matérias Estranhas e Impurezas e Mofados, tiveram acerto de 96,30% e 92,30% respectivamente. Também foi feita a matriz de confusão das imagens que não foram aplicadas o *data augmentation*, como mostra a Figura 32.

Figura 32 – Matriz de confusão de 47 imagens usadas no teste sem a técnica de *data augmentation*.



Fonte: A autora

Ao avaliarmos a matriz de confusão (Figura 32) vemos que a classe Fermentados foi a que apresentou mais erros que as outras classes, onde a rede classificou 1 vez a classe como sendo Ardidos, 2 vezes como sendo Imaturos e 1 vez como sendo Mofado. A classe Fermentados foi a que apresentou o menor índice de acerto quando compara as outras classes, obtendo 50% de assertividade.

Como mostra a Tabela 9, a classe Esverdeados obteve 83,33% de assertividade, onde a rede classificou 1 vez a classe como sendo Germinado. A classe Imaturos teve 66,66% de assertividade, pois a rede classificou 1 vez como sendo Fermentado. A classe Matérias Estranhas e Impurezas obteve 80% de assertividade, onde classificou 1 vez como sendo da classe Imaturos. A classe Mofados obteve 60% de assertividade, onde classificou 2 vezes como sendo Ardidos, e por fim as classes Ardidos, Germinados, Normais e Partidos obtiveram um índice de 100% de assertividade.

Tabela 9 – Assertividade por classe sem a técnica de *data augmentation*.

Classes	Índice de Assertividade
Ardidos	100%
Germinados	100%
Normais	100%
Partidos	100%
Esverdeados	83,33%
Matérias Estranhas e Impurezas	80%
Imaturos	66,66%
Mofados	60%
Fermentados	50%

Fonte: A autora

Na matriz de confusão (Figura 32) foram testadas 47 imagens variadas nas diferentes classes apresentadas. Conforme podemos visualizar, a somatória das classificações incorretas foi de 9. Esse número corresponde a 19,14% do total de imagens reservadas para realização de testes.

Tabela 10 – Porcentagem incorreta por classes sem a técnica de *data augmentation*.

Classes	Número de Imagens	Erros	Porcentagem
Ardidos	7	0	0%
Germinados	5	0	0%
Normais	3	0	0%
Partidos	5	0	0%
Esverdeados	6	1	16,67%
Matérias Estranhas e Impurezas	5	1	20%
Imaturos	3	1	33,33%
Mofados	5	2	40%
Fermentados	8	4	50%

Fonte: A autora

A Tabela 10 mostra os 9 erros de classificação de imagens, como pode ser observado, a classe Esverdeados que contém 6 imagens obteve 1 erro, a classe

Matérias Estranhas e Impurezas com 5 imagens obteve 1 erro, a classe Mofados com 5 imagens teve 2 erros e por fim a classe Fermentados com 8 imagens com 4 erros. A percentagem do erro foi de 16,67%, 20%, 33,33%, 40% e 50% respectivamente.

Tabela 11 – Percentagem de acertos por classes sem a técnica de data augmentation.

Classes	Número de Imagens	Acertos	Porcentagem
Ardidos	7	0	100%
Germinados	5	0	100%
Normais	3	0	100%
Partidos	5	0	100%
Esverdeados	6	1	83,33%
Matérias Estranhas e Impurezas	5	1	80%
Imaturos	3	1	66,66%
Mofados	5	2	60%
Fermentados	8	4	50%

Fonte: A autora

A Tabela 11 mostra a percentagem de acertos por classes, onde as classes Esverdeados, Matérias Estranhas e Impurezas, Imaturos, Mofados e Fermentados , tiveram acerto de 83,33%, 80%, 66,66%, 60% e 50% respectivamente.

Tabela 12 – Comparação dos índices de assertividade com e sem o *data augmentation*.

(continua)

Classes	Índice de Assertividade com data augmentation	Índice de Assertividade sem data augmentation
Ardidos	100%	100%
Germinados	100%	100%
Normais	100%	100%
Partidos	100%	100%
Esverdeados	100%	83,33%
Matérias Estranhas e Impurezas	96,29%	80%
Imaturos	100%	66,66%

Tabela 12 – Comparação dos índices de assertividade com e sem o *data augmentation*

(conclusão)		
Classes	Índice de Assertividade com <i>data augmentation</i>	Índice de Assertividade sem <i>data augmentation</i>
Mofados	92,30%	60%
Fermentados	100%	50%

Fonte: A autora

Conforme é visto na Tabela 12, os índices de assertividade com a utilização da técnica de *data augmentation* obtiveram um resultado maior do que com a base de imagens, sem a utilização dessa técnica. A Figura 33 ilustra um grão classificado de forma errada, onde a predição da rede neural foi da classe mofados, entretanto a sua classe atual é Matérias Estranhas e Impurezas. Teve-se uma perda de 1,46% e uma probabilidade de acerto de 0,23%.

Figura 33 – Grão de soja classificado erroneamente.

Mofados/MateriasEstranhasImpurezas / 1.46 / 0.23



Fonte: A autora

Além de utilizar a rede neural Resnet34, foi feito testes com outras três redes neurais convolucionais, onde as redes foram treinadas na base de dados com o *data augmentation* pela base de dados ser muito pequena. A Tabela 13 mostra as redes utilizadas e seus valores de taxa de acerto treinados.

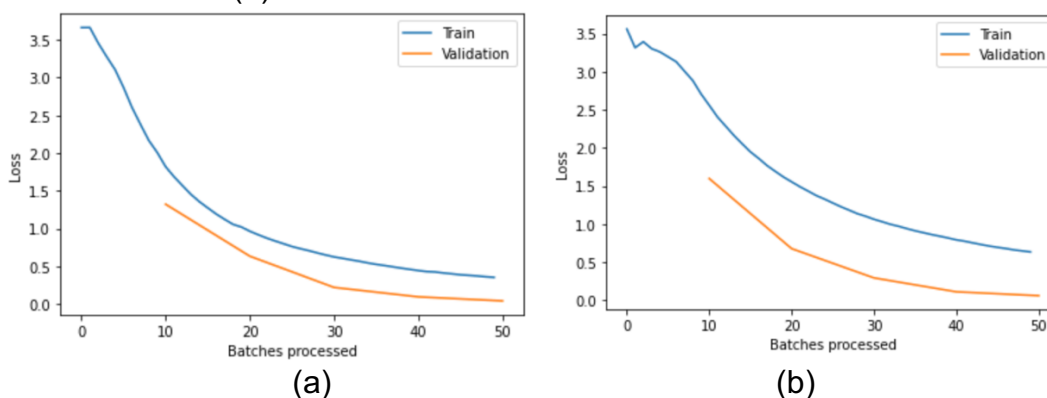
Tabela 13 – Redes Neurais treinadas e suas taxas de acerto.

Redes Neurais	Acurácia
MobileNet	97,33%
VGG19	97,22%

Fonte: A autora

Como pode ser observado na Tabela 13 as redes neurais treinadas obtiveram um valor de acurácia menor do que a *Resnet34*. As redes *MobileNet* e a *VGG19* obtiveram uma taxa de acerto muito próximas, sendo de 97,33% e 97,22% respectivamente. A Figura 34(a) e (b) mostram os resultados da porcentagem de treino e validação do modelo.

Figura 34 – Resultados da porcentagem de treino e validação do modelo. (a) Rede Neural *VGG19*. (b) Rede Neural *MobileNet*.



Fonte: A autora

Conforme os resultados apresentados, foi visto que a técnica de *data augmentation* obteve uma taxa de assertividade maior, portanto, foi executado o *learning rate* (taxa de aprendizado) nessa base de dados, na rede *Resnet34* por ter obtido um valor maior de acurácia. O *learning rate* é um parâmetro que tem influência durante o processo de treinamento de uma rede neural, podendo aumentar ou diminuir a acurácia da rede, por isso seu valor numérico é importante para se obter uma taxa de assertividade maior. Foram testados diferentes valores de *learning rate*, como mostra a Tabela 14.

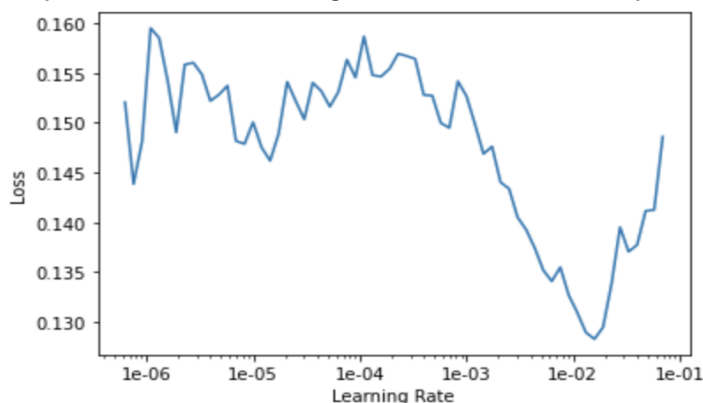
Tabela 14 – Valores dos *learning rate* utilizados e as acurácias obtidas.

Learning Rate	Acurácia
10^{-1}	84%
10^{-2}	99,55%
10^{-3}	97,77%
10^{-4}	94,66%
10^{-5}	95,11%
10^{-6}	95,11%

Fonte: A autora

Por padrão a Resnet34 utiliza um *learning rate* de 10^{-3} , como é visto na Tabela 14 utilizando este valor, a rede neural obteve uma taxa de 97,77%. Entretanto, a melhor acurácia obtida foi com o *learning rate* 10^{-2} com uma acurácia de 99,55%. A Figura 35 ilustra as taxas de aprendizado obtidas.

Figura 35 – Gráfico que ilustra os *learning rate* obtidos e suas perdas.



Fonte: A autora

Conforme a Figura 35 mostra, o melhor *learning rate* foi 10^{-2} , obtendo uma perda de aproximadamente 0.130. A Figura 36 mostra o treinamento da rede e sua validação com o *learning rate* que teve a menor perda na validação da rede.

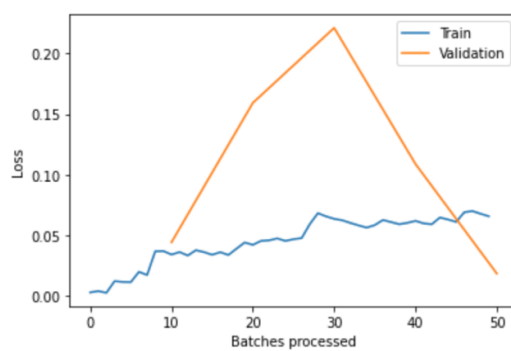
Tabela 15 – Resultado da rede neural com um *learning rate* de 10^{-2}

Épocas	Perda no Treino	Perda na Validação	Acurácia	Tempo
0	0.030200	0.000551	1.000000	00:09
1	0.025033	0.129092	0.973333	00:09
2	0.038099	0.120440	0.955556	00:09
3	0.043116	0.051511	0.986667	00:09
4	0.045499	0.004160	0.995556	00:09

Fonte: A autora

Os resultados da porcentagem de treino e validação do modelo são mostrados na Tabela 15. Como pode-se observar, o gráfico mostra o erro de validação subestimando o erro de treinamento. Isto aconteceu pelo mesmo motivo que foi explicado na Figura 30(a) e 30(b).

Figura 36 – Resultados da porcentagem de treino e validação do modelo com *learning rate* de 10^{-2} .



Fonte: A autora

6 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi apresentado um modelo de padrão de metadados para a classificação de grãos, além de um método computacional para a classificação de grãos de soja onde redes neurais convolucionais foram utilizadas. As redes neurais convolucionais utilizadas foram as Resnet34, MobileNet e VGG19. A rede foi treinada com base nos dados de defeitos de grão.

A base de dados foi criada a partir de imagens de defeitos de grãos de soja, onde 16 imagens de cada classe passaram por técnicas de rotação, inversão de dados e aproximação, gerando 100 imagens por classe com um total de 900 imagens.

A Resnet34 foi treinada com e sem a utilização dessa técnica. As formas de treinamento de validação para todas as redes foram de 75% das imagens usadas para treinamento e 25% usadas para validação, essa separação das imagens foram feitas aleatoriamente. A rede Resnet34 obteve 97,77% de taxa de assertividade com o uso da técnica de *data augmentation*, e 80,85% sem essa técnica. As redes MobileNet e VGG19 obtiveram 97,33% e 97,22% respectivamente, com o uso do *data augmentation*. Como a Resnet34 foi a rede que obteve maior acurácia nos treinamentos, foram testados diferentes *learning rate* com o objetivo de aumentar a acurácia, obtendo uma taxa de assertividade de 99,55%. Podemos dizer que é um excelente resultado, já que não foram utilizadas técnicas de pré-processamento nas imagens, e também em relação ao tamanho da base de imagens, por não conter um número alto de amostras, a rede obteve um índice de acerto muito bom para ambos os casos.

O aplicativo desenvolvido para realizar o gerenciamento dos metadados de classificação de grãos é de fácil manuseio e intuitivo para que todas as pessoas possam usar sem dificuldades, não necessitando de um hardware avançado para sua execução. O aplicativo fornece duas estruturas para que os metadados sejam gerados, em XML e em HTML, sendo possível que o usuário possa fazer o uso das duas estruturas.

Em relação aos trabalhos futuros, é interessante aumentar a base de imagens com diversas imagens que não sigam um padrão e realizar um novo treinamento para que a taxa de assertividade possa chegar aos 100% ou o mais próximo possível, aprimorar os metadados apresentados neste trabalho, além de, desenvolver uma

esteira para as indústrias de classificação de grãos para que seja feita a classificação de forma automática.

REFERÊNCIAS

- ALVES, R. C. V. **Web Semântica: uma análise focada no uso de metadados**. 2005, 180f. Dissertação (Mestrado em Ciência da Informação) Faculdade de Filosofia e Ciências, Universidade Estadual Paulista, Marília, 2005.
- APROSOJA BRASIL. **Uso da Soja**. Disponível em: <https://aprosojabrasil.com.br/a-soja>. Acesso em: 11 jan. 2020.
- ARAÚJO, F. H. D.; CARNEIRO, A. C.; SILVA, R. R. V.; MEDEIROS, F. N. S.; USHIZIMA, D. M. Redes Neurais Convolucionais com Tensorflow: Teoria e Prática. III **Escola Regional de Informática do Piauí**, Piauí, v. 1, n. 1 p. 382-406, 2017.
- ARROYO-FIGUEROA, G.; SUCAR, L.; VILLAVICENCIO, A. Fuzzy intelligent system for the operation of fossil power plants. **Engineering Applications of Artificial Intelligence**, Mexico, v. 13, n. 4, p. 431–439, 2000.
- BACA, M. **Introduction to Metadata**. Los Angeles: Getty Research Institute, 2008.
- BAY, H. et al. Speeded-up robust features (surf). **Computer Vision and Image Understanding**, Suíça, v. 110, n. 3, p. 346-359, 2008.
- BONIECKI, P.; KOSZELA, K.; SWIERCZYNSKI, K.; SKWARCZ, J.; ZABOROWICZ, M.; PRZYBYL, J. Neural Visual Detection of Grain Weevil (*Sitophilus granaries* L). Polonia, **Agriculture**, v. 10, n.1 p. 2-9, 2020.
- BRASIL. Ministério da Agricultura, Pecuária e Abastecimento – MAPA. **Instrução Normativa nº 11 de 15 de maio de 2007**. Sistema Integrado de Legislação. Brasília: Mapa/ACS, 2018.
- CONAB. **Conjuntura Mensal de soja de Dezembro 2019**. Disponível em: <https://www.conab.gov.br/info-agro/analises-do-mercado-agropecuario-e-extrativista/analises-do-mercado/historico-mensal-de-soja?start=10>. Acesso em: 15 jun. 2020.
- DCMI. **The Dublin Core Metadata Initiative**. Disponível em: <http://dublincore.org>. Acesso em: 10 fev. 2020.
- DCMI. **The Dublin Core Metadata Initiative**. Disponível em: <https://www.dublincore.org/specifications/dublin-core/dces/>. Acesso em: 20 fev. 2020.
- FAO. **Statistical Databases**. Disponível em: <http://www.fao.org/faostat>. Acesso em: 20 set. 2020.
- FIGUEIREDO, D. G. **Análise de Fourier e Equações Diferenciais Parciais**. 4. ed. Rio de Janeiro, RJ: Impa, 2000.

GERON, A. *Mãos à Obra: Aprendizado de Máquina com Scikit-Learn TensorFlow*. Rio de Janeiro: Alta Books, 2019.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge: MIT Press, 2016.

GONZALEZ, R. C.; WOODS, R. E. **Processamento digital de imagens**. São Paulo: Pearson Prentice Hall, 2010.

GILLILAND-SWETLAND, A. J. **Introducción a los metadatos vías a la información digital**. Los Angeles: Murtha Baca, 1998.

HARLAN, J. R. **Crops and man**. America Society of Agronomy. Cambridge, v. 8, n. 1, p. 47-48, 1992.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. Las Vegas, p. 770-778, 2016.

HILLMANN, D. **Using Dublin Core**. Disponível em: <https://www.dublincore.org/specifications/dublin-core/usageguide/2005-11-07/>. Acesso em: 5 jul. 2020.

KARLEKAR, A.; SEAL, A. SoyNet: Soybean leaf diseases classification. **Computers and Electronics in Agriculture**. Florida, v. 172, 2020.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. **Communications of the ACM**. Toronto, v. 60, n. 6, p.1097-1105, 2012.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, New York, v. 86, n.11, p. 2278-2324, 1998.

LINDEBERG, T. **Scale Invariant Feature Transform**. Suecia: Stockholm, 2015.

LOWE, David G. Distinctive image features from scale-invariant keypoints. **International Journal of Computer Vision**, Colombia, v. 60, n. 2, p. 91-110, 2004.

MADSEN, M. S.; FOOG, I.; RUGGLES, C. Metadata systems: integrative information technologies. **De Gruyter Saur**, United Kingdom, v. 44, n. 3, p. 237-257, 1994.

MANOUSELIS, N. et al. Metadata interoperability in agricultural learning repositories: An analysis. **Computers and Electronics in Agriculture**. v. 70, n. 2, p. 302-320, 2010.

MARQUES FILHO, O.; VIEIRA NETO, H. **Processamento Digital de Imagens**. Rio de Janeiro: Brasport. 1999.

MARTINS, J. G. **Identificação de Espécies Florestais utilizando Seleção Dinâmica de Classificadores no Espaço de Dissimilaridade**. Tese de Doutorado. Programa de Pós-Graduação em Informática do Setor de Ciências Exatas da Universidade Federal do Paraná, Curitiba, 2014.

MÉNDEZ RODRÍGUEZ, E. **Metadados y recuperación de información: estándares, problemas y aplicabilidad en bibliotecas digitales**. Gijón: Trea, 2002.

MENEZES, C. **Reconhecimento de Imagens por Redes Neurais Convolucionais (CNN) em Sistemas Embarcados**. Disponível em: <https://www.embarcados.com.br/webinars/webinar-reconhecimento-de-imagens-por-redes-neurais-convolucionais>. Acesso em: 15 jul. 2020

MOGHADDAM, B; NASTAR, C; PENTLAND, A. **A Bayesian similarity measure for deformable image matching**. Image and Vision computing, Cambridge, v. 19, n. 5, p. 235-244, 2001

MOURA, A.; FERNANDES, S. **A Metadata Approach to Represent and Visualize Sites on the Web**. International Workshop on Information Integration Technologies on the Web. Rio de Janeiro, 2001.

NASCIMENTO, L. U. **Um Padrão de Metadados para Indexação e Recuperação de Objetos Multimídia**. Dissertação (Mestrado em Informática). Universidade Federal do Paraná, Curitiba, 2008.

NISO. **UNDERSTANDING METADATA: WHAT IS METADATA, AND WHAT IS IT FOR**. Disponível em: <https://www.niso.org/publications/understanding-metadata-2017>. Acesso em: 4 jan. 2020.

NISO. **Understanding Metadata**. Disponível em: <https://www.niso.org/publications/understanding-metadata-2017>. Acesso em: 8 jun. 2020.

OLIVEIRA, B. A. S.; CARDOSO, G. M. M.; DIAS, F. P.; SANTOS, F. H. W.; GUIMARÃES, F. G. **Sistema para Classificação de Grãos de Café: Uma Comparação entre Redes Neurais Convolucionais e Processamento Digital de Imagens com MLP**. Simpósio Brasileiro de Automação Inteligente, Minas Gerais, p. 256-261, 2019.

PATRÍCIO, D. I. **SeedFlow: Sistema de Visão Computacional para Classificação de Grãos de Aveia**. Dissertação do Programa de Pós-Graduação da UPF. Passo Fundo, 2019.

PEDRINI, H.; SCHWARTZ, W. R. **Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações**. São Paulo: Thomson Learning, 2008.

PEREIRA, A. G. **Séries de Fourier e aplicações**. Disponível em: <http://dspace.bc.uepb.edu.br:8080/xmlui/handle/123456789/715>. Acesso em: 15 jan. 2020.

PINTO, C.; FURUKAWA, J.; FUKAI, H.; TAMURA, S. **Classification of green coffee bean images basec on defect types using convolutional neural network (CNN)**. IEEE Xplore. Japan, 2017.

PRUSAK, L. **Knowledge in Organizations (Resources for the Knowledge-Based Economy)**. Boston: Butterworth-Heinemann, 1997.

RASCHKA, S. **Python Machine Learning**. Birmingham: Packt Publishing Ltd, 2015.

RIDDICK, A. T.; HEAVEN, R.; ROYSE, K. R., Singh, A., Hughes, A.G. **A model metadata schema for environmental hazard models and its implementation in the PURE portal**. Environmental Modelling and Software. United Kingdom, v. 124, p. 104597, 2019.

ROCHA, B.G.R.; AMARO, H.T.R; PORTO, E.M.V.; GONÇALVES, C.C.; DAVID, A.M.S.S.; LOPES, E.B. Sistema de semeadura cruzada na cultura da soja: avanços e perspectivas. **Revista de Ciências Agrárias**, Minas Gerais, v. 41, n.2, p. 91-100, 2018.

SOLEM, J. E. **Programming Computer Vision with Python: Tools and algorithms for analyzing images**. Beijing: O'Reilly Media, 2012.

SMITH, J. R.; e SCHIRLING, P. **Metadata Standards Roundup**. Disponível em: <http://ieeexplore.ieee.org>. Acesso em: 20 fev. 2020.

TENSORFLOW. **Tensors**. Disponível em: <https://www.tensorflow.org/guide/tensor>. Acesso em: 22 de set. 2020.

ZAPOTOCZNY, P.; REINER, J.; MRZYGLOD, M.; LAMPA, P. The use of polarized light and image analysis in evaluations of the severity of fungal infection in barley grain. **Computers and Electronics in Agriculture**. Poland, v. 169, p.105154, 2020.