

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA
CAMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA
ENGENHARIA MECATRÔNICA**

Matheus Espindola

**Detecção Automatizada de Defeitos em Madeira Serrada Empregando Redes Neurais
Convolucionais**

Florianópolis, 2022

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA
CAMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA
ENGENHARIA MECATRÔNICA**

Matheus Espindola

**Deteccção Automatizada de Defeitos em Madeira Serrada Empregando Redes Neurais
Convolucionais**

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Santa Catarina como parte dos requisitos para obtenção do título de Engenheiro em Engenharia Mecatrônica

Orientador:

Prof. Raimundo Ricardo Matos da Cunha, DR. ENG.

Florianópolis, 2022

Ficha de identificação da obra elaborada pelo autor.

Espindola, Matheus

**Detecção Automatizada de Defeitos em Madeira Serrada
Empregando Redes Neurais Convolucionais / Matheus Espindola;
orientação de Raimundo Ricardo Matos da Cunha;
coorientação de Mauricio Edgar Stivanello. - Florianópolis,
SC, 2022.**

45 p.

**Trabalho de Conclusão de Curso (TCC) - Instituto Federal
de Santa Catarina, Câmpus Florianópolis. Bacharelado
em Engenharia Mecatrônica. Departamento
Acadêmico de Metal Mecânica.
Inclui Referências.**

1. **Aprendizado de máquinas.** 2. **Redes Neurais Convolucionais.**
3. **Inspeções de defeitos em madeira.** 4. **UNet.**
5. **FastAI.** I. **Matos da Cunha, Raimundo Ricardo.** II.
Edgar Stivanello, Mauricio. III. **Instituto Federal
de Santa Catarina.** IV. **Detecção Automatizada de Defeitos
em Madeira Serrada Empregando Redes Neurais Convolucionais.**

Detecção Automatizada de Defeitos em Madeira Serrada Empregando Redes Neurais Convolucionais

MATHEUS ESPINDOLA

Este trabalho foi julgado adequado para obtenção do título de Engenheiro em Engenharia Mecatrônica e aprovado na sua forma final pela banca examinadora do Curso do Instituto Federal de Educação Ciência e Tecnologia de Santa Catarina.

Florianópolis, 05 de Dezembro de 2022.

Banca Examinadora:



Documento assinado digitalmente
RAIMUNDO RICARDO MATOS DA CUNHA
Data: 15/02/2023 10:42:06-0300
Verifique em <https://verificador.itl.br>

Prof.Orientador: Raimundo Ricardo Matos da Cunha



Documento assinado digitalmente
GREGORY CHAGAS DA COSTA GOMES\;gregory.c
Data: 17/02/2023 16:49:33-0300
CPF: 073.008.969-01
Verifique as assinaturas em <https://v.ifsc.edu.br>

Prof. Avaliador 1: Gregory Chagas

IFSC



Documento assinado digitalmente
DELCINO PICININ JUNIOR
Data: 16/02/2023 14:00:26-0300
Verifique em <https://verificador.itl.br>

Prof. Avaliador 2: Delcino Picinin Júnior

IFSC



INSTITUTO FEDERAL
SANTA CATARINA

MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS

DECLARAÇÃO DE FINALIZAÇÃO DE TRABALHO DE CURSO

Declaro que o estudante **MATHEUS ESPÍNDOLA**, matrícula nº 1620054752, do Curso de Engenharia Mecatrônica, defendeu o trabalho intitulado **DETECÇÃO AUTOMATIZADA DE DEFEITOS EM MADEIRA SERRADA EMPREGANDO REDES NEURAIAS CONVOLUCIONAIS** o qual está apto a fazer parte do banco de dados da Biblioteca Hercílio Luz do Instituto Federal de Santa Catarina, Campus Florianópolis.

Florianópolis, 15 de Fevereiro de 2023.



Documento assinado digitalmente
RAIMUNDO RICARDO MATOS DA CUNHA
Data: 15/02/2023 10:39:26-0300
Verifique em <https://verificador.jfsc.br>

RAIMUNDO RICARDO MATOS DA CUNHA
Prof. Orientador do TCC

AGRADECIMENTOS

Gostaria de agradecer primeiramente a Deus por sempre me proteger e orientar durante os tempos difíceis vividos durante a minha caminhada.

Agradecer aos meus familiares, especialmente a minha mãe, Maria Inês, e meu pai Elias, que sempre fizeram o possível e o impossível para que eu tivesse um ensino de qualidade toda a minha vida, ao meu irmão Alessandro, por sempre ser fazer presente, e me proporcionar momentos de descontração e alegria, e a minha irmã Elaine, por sempre demonstrar o amor que sente por mim, e me apoiar em qualquer decisão que tive.

Aos meus amigos que cultivei durante toda minha vida, seja ela, acadêmica ou pessoal, sem eles eu com certeza, não teria chegado até aqui, todos os momentos de risadas e dificuldades proporcionaram experiências essenciais para meu desenvolvimento tanto pessoal, como profissional.

A todos os professores do IFSC que tive a oportunidade de receber seus ensinamentos, sempre buscando fornecer o ensino de melhor qualidade para seus alunos a fim de formar profissionais do mais alto nível.

E por fim, ao a instituição IFSC, que me proporcionou todos os acontecimentos citados a cima.

RESUMO

Este trabalho propõe a elaboração e avaliação de um algoritmo para a classificação de madeira serrada empregando o aprendizado de máquinas com foco em utilizar redes neurais convolucionais para a detecção de defeitos. Para exposição do conteúdo foi realizada uma contextualização para o setor madeireiro, descrevendo os defeitos encontrados, bem como a importância de uma classificação dos mesmos. Também serão abordados conceitos de redes neurais artificiais e suas funcionalidades. Para a elaboração do algoritmo foi utilizado a biblioteca FastAI juntamente com a linguagem de programação *Python* para realizar os treinamentos do modelo, devido a ampla documentação disponível para estudo. Por fim foram expostos e avaliados os resultados obtidos após a implementação do modelo e uma possível continuação para trabalhos futuros.

Palavras-chave: Aprendizado de máquinas, Redes Neurais Convolucionais, Inspeções de defeitos em madeiras, OpenCV, Segmentação Semântica, FastAI, UNet.

ABSTRACT

This project proposes the elaboration and evaluation of an algorithm for the classification of lumber using machine learning focused on using convolutional neural networks for the detection of defects. To expose the content, a contextualization was carried out for the timber sector, describing the defects found, as well as the importance of classifying them. Concepts of artificial neural networks and their functionalities will also be addressed. For the elaboration of the algorithm, was used the FastAI library for Python programming language to carry out the training of the model, due to the extensive documentation available for study. Finally, the results obtained after the implementation of the model and a possible continuation for future work were exposed and evaluated.

Keywords: Machine Learning, Covolutional Neural Network, Wood defect inspections, OpenCV, Semantic Segmentation, FastAI, UNet.

LISTA DE FIGURAS

Figura 1 - Defeitos presentes na base de dados	14
Figura 2– Definições dadas a inteligência artificial ao longo dos anos	15
Figura 3 - Comparação entre Neurônio e Perceptron.....	16
Figura 4 - Detalhamento de um perceptron	17
Figura 5 - Funções de ativação	17
Figura 6 - Estrutura básica de uma RNC	18
Figura 7 – Exemplo de segmentação semântica	19
Figura 8 - Arquitetura UNet.....	20
Figura 9 - Exemplos de classificação da métrica.....	21
Figura 10 – Equipamento de captura de imagens	25
Figura 11 - Imagem da madeira serrada e identificação dos defeitos.....	27
Figura 12 - Máscara fornecida a rede neural.	29
Figura 13 – Perda versus Taxa de aprendizado para o modelo de multi-classes.....	30
Figura 14 – Máscara fornecida ao modelo de classificação binária.....	31
Figura 15 – Perda versus a taxa de aprendizado do modelo de classificação binária.....	31
Figura 16 - Comparação dos resultados esperados versus obtidos (resnet18).....	34
Figura 17 – Comparação dos resultados esperados versus obtidos (resnet34).	35
Figura 18– Comparação mascara modelo próprio versus original. (Múltiplas classes)	36
Figura 19 – Comparação do resultado do modelo de classificação binária (resnet18).....	38
Figura 20– Comparação dos resultados obtidos pelo modelo de classificação binária(resnet34).	39
Figura 21 – Comparações das máscaras geradas pelo modelo para ambas as arquiteturas com a máscara original do banco de dados.	41

LISTA DE TABELAS

Tabela 1 -	Aparição dos defeitos em fotos da base de dados.....	26
Tabela 2 -	Exemplo de como é realizada a rotulação dos defeitos.....	28
Tabela 3 -	Resultados dos treinamentos do modelo de classificação com múltiplas classes.....	33
Tabela 4 -	Resultados dos treinamentos do modelo de classificação binária.....	37
Tabela 5 -	Comparativo de resultados com outros modelos.....	42

SUMÁRIO

1 INTRODUÇÃO	11
1.1 Contextualização.....	11
1.3 Objetivos.....	11
1.3.1 Principais	11
1.3.2 Específicos	12
2 FUNDAMENTAÇÃO TEÓRICA	13
2.1 O setor madeireiro e a busca por automatização de serviços manuais	13
2.2 Defeitos da madeira.....	13
2.3 Aprendizado de máquinas e as redes neurais artificiais.....	14
2.3.1 Inteligência Artificial e suas ramificações	14
2.3.4 Redes Neurais Convolucionais.....	18
2.2.5 UNet	19
2.4 Métricas para avaliação de resultados.....	21
2.5 Trabalhos Correlatos	22
3 METODOLOGIA	24
3.1 Requisitos de projeto	24
3.2 Descrição da solução implementada	24
3.3 Base de dados.....	24
3.4 Abordagem 1: Modelo de classificação de múltiplas classes	28
3.5 Abordagem 2: Modelo de treinamento com apenas uma classe de defeito	30
4. RESULTADOS OBTIDOS.....	33
4.1 Modelo de classificação de apenas uma classe	37
4.2 Modelo de classificação com múltiplas classes	33
5. CONSIDERAÇÕES FINAIS	42
REFERÊNCIAS	44

1 INTRODUÇÃO

1.1 Contextualização

Como toda matéria prima, a madeira não é isenta da presença de defeitos, sejam eles naturais ou causados por algum processo de fabricação. Com isso em mente, a indústria madeireira demanda que seja feita a triagem destas imperfeições, devido à presença ou não de tais defeitos alterarem a qualidade e o valor do produto destinados ao comércio (Hittawe, et al., 2015). A inspeção de defeitos em madeiras processadas, geralmente é feita de forma manual, por um profissional treinado, porém conforme a jornada de trabalho avança, a precisão das verificações começa a diminuir (Dellabetha, et al, 2018).

A busca por automatização na detecção de defeitos na indústria madeireira se faz necessária, uma vez que a produção acelerada e as inspeções manuais acabam por vezes não gerando resultados satisfatórios. Dentro destas técnicas de inspeção automatizada tem se destacado o emprego de visão computacional para a substituição de operadores humanos. Dentre as técnicas utilizadas na área, observa-se o aumento do uso de técnicas que utilizam aprendizado de máquina, mais especificamente as Redes Neurais Convolucionais (CNN ou ConvNet). A utilização de Redes Neurais Convolucionais para realizar a inspeção de defeitos vem se mostrando muito eficiente nos últimos anos, alcançando sucesso similar ou superior àqueles alcançados por operadores humanos levando em conta critérios como tempo de resposta, taxas de acerto e efetividade da inspeção ao longo do trabalho.

Observando a importância assumida por tais técnicas no desenvolvimento de sistemas de inspeção voltadas à indústria, propõe-se um estudo e avaliação dos resultados frente ao caso de inspeção específica de madeira serrada, tradicionalmente solucionado empregando técnicas de visão clássicas.

1.3 Objetivos

As seções a seguir descrevem quais foram os objetivos principais e específicos propostos no início da pesquisa.

1.3.1 Principal

O objetivo principal do trabalho foi desenvolver um algoritmo capaz de localizar e classificar os diferentes defeitos presentes em madeiras serradas.

1.3.2 Específicos

Os objetivos específicos a serem alcançados durante o desenvolvimento do presente trabalho foram:

- Estudar a implementação de redes neurais convolucionais;
- Estudar os defeitos mais relevantes encontrados em madeira serrada;
 - Sendo eles: nós vivos, nós mortos, nós rachados, nós vazados, quartizite, rachaduras, crescimento excessivo, bolsa de resina, medula e manchas azuladas;
- Identificar e/ou preparar uma base de dados a ser utilizada no treinamento dos classificadores e na validação da solução proposta;
- Selecionar e implementar classificadores capazes de detectar os defeitos de interesse;
- Validar a solução proposta.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo foi feita uma breve contextualização do setor madeireiro, dos defeitos presentes no material e uma breve introdução do ponto de vista de um usuário sobre redes neurais artificiais e suas possíveis aplicações.

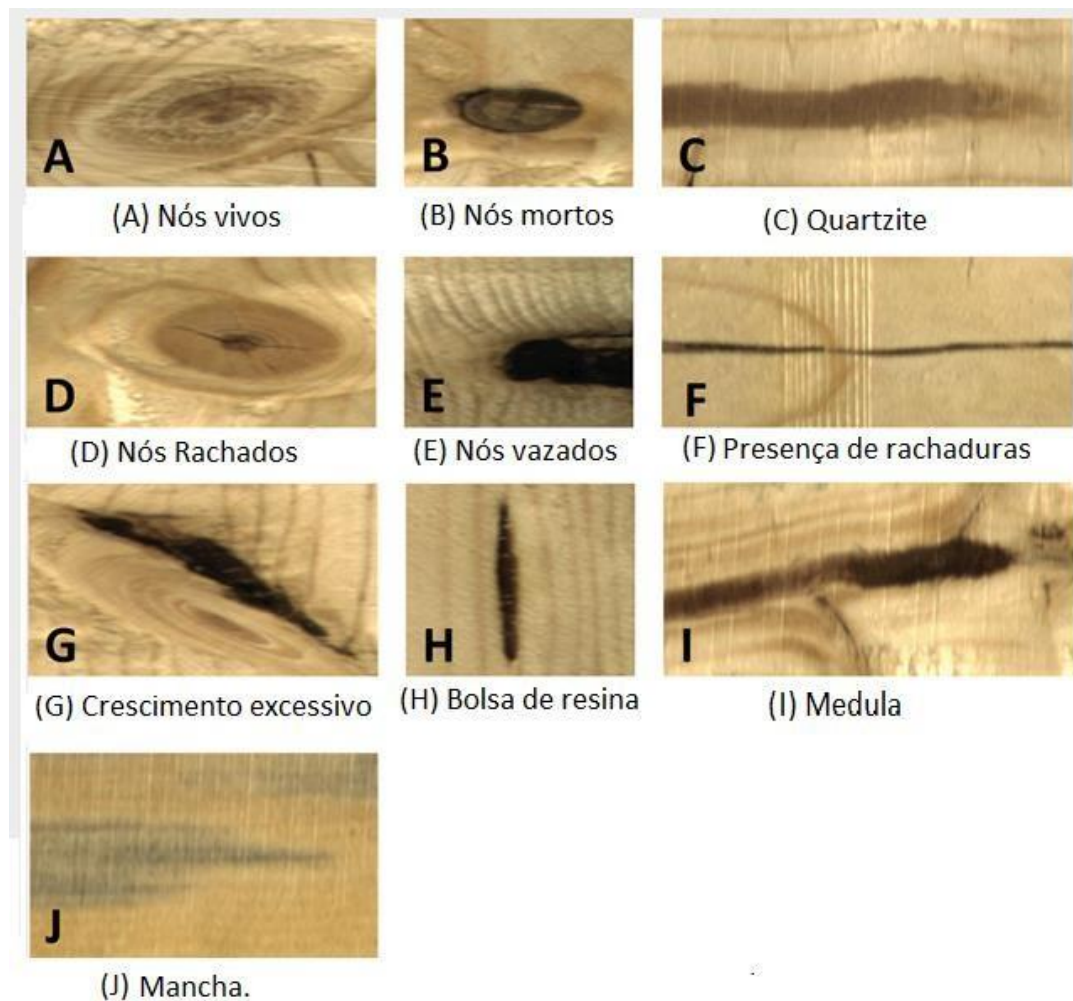
2.1 O setor madeireiro e a busca por automatização de serviços manuais

Segundo a Associação Brasileira da Indústria de Madeira Processada Mecanicamente (ABIMCI), o setor madeireiro se faz muito presente na economia do Brasil. A exportação de madeiras serradas, portas, pisos e pallets, demonstrou um crescimento em 2021 se comparados com os dados de 2020. Com o crescimento do mercado e o desenvolvimento de novas ferramentas e tecnologias, surge a necessidade de inovar e automatizar processos, a fim de se melhorar os resultados obtidos. Uma tecnologia relativamente nova que vem ganhando espaço no meio das indústrias é a utilização do aprendizado de máquinas para a substituição de mão de obra manual. Apesar de novo, o método vem ganhando bastante destaque devido aos resultados gerados serem satisfatórios, com resultados médios de reconhecimento acima de 90% de defeitos em madeiras, cerca de 20% maior se comparados com os resultados de um operador humano que ficam na faixa de 70% de reconhecimento (Kodytek; Bodzas; Bilik, 2021, p.3).

2.2 Defeitos da madeira

Nesta seção foram abordados os defeitos classificados pelo algoritmo, a lista foi confeccionada com o auxílio de um profissional experiente, segundo (P. Kodytek, et. Al.). Tais defeitos geralmente surgem na madeira quando se realiza algum processo de fabricação sobre o material, porém eles são causados por fatores naturais. Observando na Figura 1, são exemplificados os defeitos presentes na base de dados bem como suas nomenclaturas estipuladas pelos autores do banco de dados.

Figura 1 - Defeitos presentes na base de dados



Fonte: Adaptado de Kodytek P. et. Al (2021).

2.3 Aprendizado de máquinas e as redes neurais artificiais

Para um melhor entendimento do trabalho foram abordados os conceitos de Redes neurais artificiais e de aprendizados de máquinas nas sessões a seguir, a fim de se facilitar a leitura bem como contextualizar os termos utilizados no meio de visão computacional.

2.3.1 Inteligência Artificial e suas ramificações

O conceito da Inteligência Artificial (IA) se baseia em 4 vieses, cada qual tende a ditar o comportamento dos sistemas de máquinas: o primeiro remete ao lado do pensamento humano, o segundo, busca o pensamento racional, o terceiro visa o ato de agir humanamente e o último se concentra em agir racionalmente (RUSSELL; NORVIG, 1994)

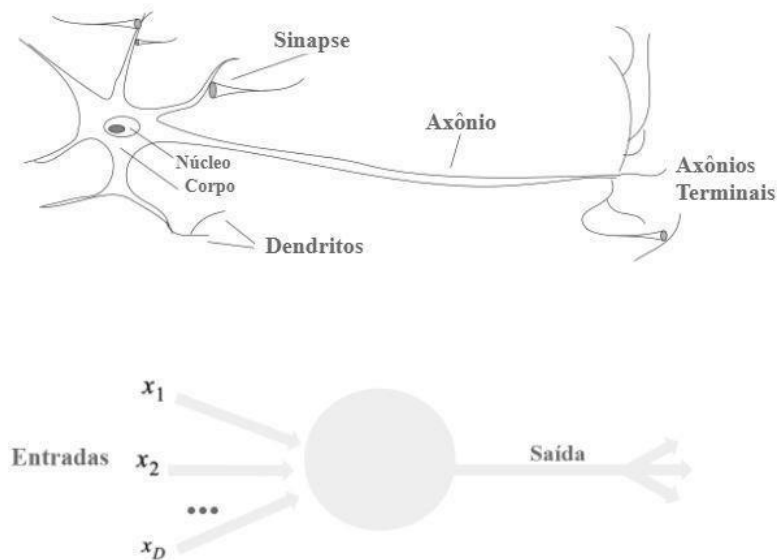
Figura 2– Definições dadas a inteligência artificial ao longo dos anos.

Pensando como um humano	Pensando racionalmente
“O novo e interessante esforço para fazer os computadores pensarem (...) <i>máquinas com mentes</i>, no sentido total e literal.” (Haugeland, 1985) “[Automatização de] atividades que associamos ao pensamento humano, atividades como a tomada de decisões, a resolução de problemas, o aprendizado...” (Bellman, 1978)	“O estudo das faculdades mentais pelo uso de modelos computacionais.” (Charniak e McDermott, 1985) “O estudo das computações que tornam possível perceber, raciocinar e agir.” (Winston, 1992)
Agindo como seres humanos	Agindo racionalmente
“A arte de criar máquinas que executam funções que exigem inteligência quando executadas por pessoas.” (Kurzweil, 1990) “O estudo de como os computadores podem fazer tarefas que hoje são melhor desempenhadas pelas pessoas.” (Rich and Knight, 1991)	“Inteligência Computacional é o estudo do projeto de agentes inteligentes.” (Poole <i>et al.</i>, 1998) “AL... está relacionada a um desempenho inteligente de artefatos.” (Nilsson, 1998)

Fonte: Adaptado de (RUSSEL, at., al 2013)

Pertencente ao ramo de IA o conceito do aprendizado de máquinas diz respeito a transmitir o comportamento humano a computadores através da aplicação de técnicas de programação com a finalidade de capacitar máquinas a processarem dados apresentados a elas de maneira prévia (LEMOS, 2021). Com o intuito de capacitar os computadores a tal comportamento, uma das formas criadas foi o desenvolvimento de Redes Neurais Artificiais (RNA), que buscam imitar um cérebro humano com suas redes neurais biológicas, onde neurônios se conectam através de sinapses para transmitir informações. Uma rede neural artificial também é composta por “neurônios”, chamados de perceptrons, onde esse neurônio é responsável por receber um dado, processar, armazenar essa informação e transmiti-la em formato de um sinal aos demais. Na Figura 3 é possível verificar a semelhança entre um neurônio e um perceptron.

Figura 3 - Comparação entre Neurônio e Perceptron

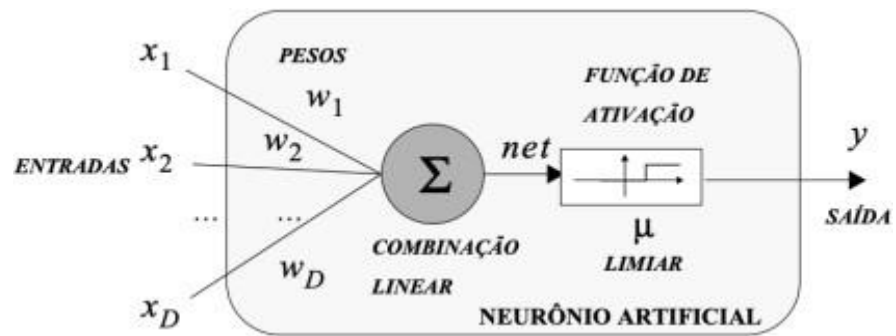


Fonte: Adaptado de BRIGNOLI, RAMON (2021)

A representação exposta anteriormente, diz respeito ao modelo inicial do perceptron, desenvolvido por McCulloch e Pitts, onde cada entrada do perceptron (representadas na figura por $x_1, x_2, \dots, x_D$) são alimentadas com saídas provenientes de outro perceptron que por sua vez alimenta outro perceptron, formando uma rede de neurônios artificiais.

Além disso, todas as entradas do perceptron são vinculadas com matrizes de pesos ($w_1, w_2, \dots, w_D$) onde é realizado um produto entre os pesos e as entradas e esse valor por sua vez passa por uma combinação linear, resultando no processamento do dado recebido pelo perceptron. A figura 4 mostra o corpo do perceptron em maiores detalhes.

Figura 4 - Detalhamento de um perceptron

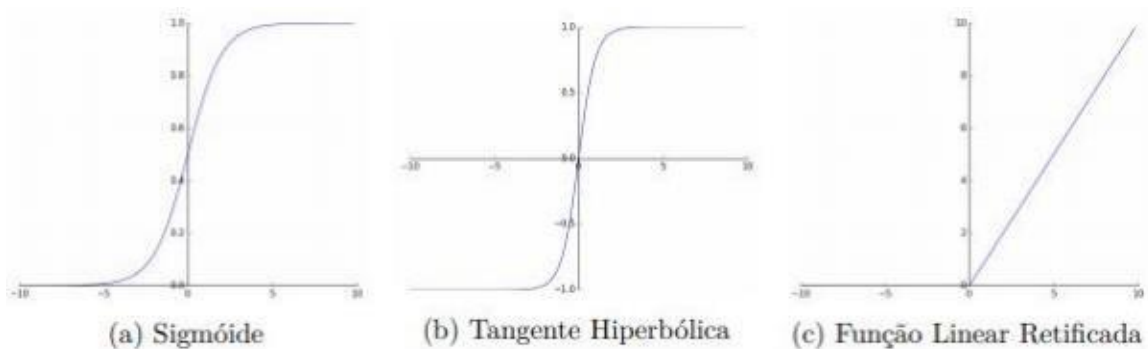


Fonte: RAUBER (1997)

Após a combinação linear do produto entre os pesos e as entradas, o resultado que se obtém é nomeado de *net* como visto na figura acima, essa *net* por sua vez, passa por uma função de ativação específica dentre diversos tipos, que é responsável por processar matematicamente os dados fornecidos ao perceptron. Cada função de ativação tem suas características próprias bem como utilizações.

Dentre essas funções de ativações tem destaque as funções Sigmóide, Tangente Hiperbólica e Retificadora (ReLU), sendo a última a mais utilizada atualmente. Os gráficos das funções podem ser observados na Figura 5 na sequência:

Figura 5 - Funções de ativação



Fonte: FERREIRA, Arthur

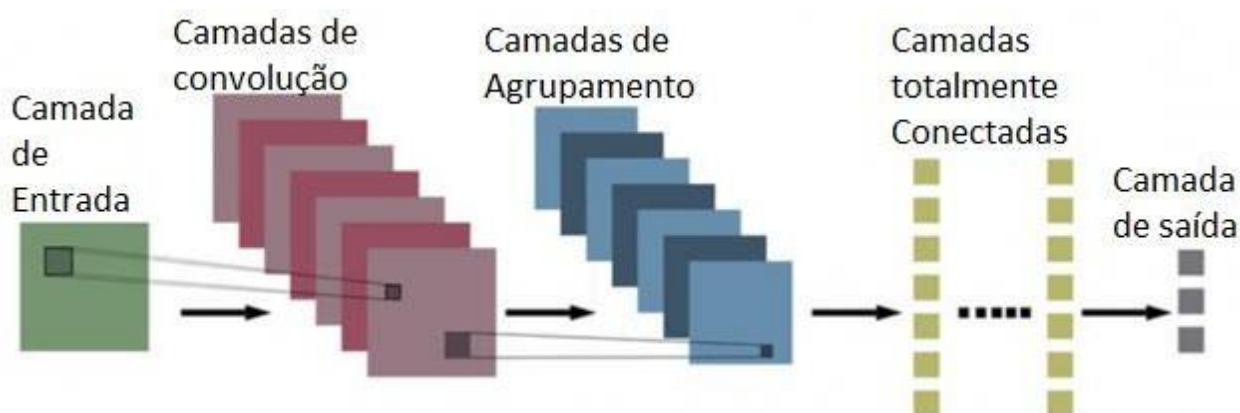
2.3.4 Redes Neurais Convolucionais

Redes Neurais Convolucionais (RNCs) são arquiteturas derivadas das RNAs com base na convolução de *kernel* e sua utilização amplamente visada no meio de visão computacional.

A classificação de imagens utiliza-se das RNCs devido ao seu comportamento em atribuir pesos a diversas características das imagens, podendo facilmente diferenciá-las umas das outras. Como sua antecessora a RNA, a RNC também está estruturada em camadas e as informações fornecidas as camadas de entradas são mantidas e passadas pelas outras até chegarem às camadas de saída. Um diferencial das RNCs é que suas camadas não são completamente conectadas, esse fato é muito importante na área de processamento de imagens, pois as conexões dos neurônios aumentam de maneira exponencial ao tamanho das camadas da arquitetura, se somarmos isso ao fato das imagens serem matrizes de pixels, quando tivermos uma imagem de 100x100 seriam necessários uma grande quantidade de parâmetros em cada camada para realizar o processamento desta imagem.

Para as RNCs apenas os pixels nas proximidades influenciam na região de interesse, essa informação é passada para os neurônios de camadas adjacentes o que possibilita a classificação de padrões complexos, de maneira mais eficiente. Na Figura 6 é possível observar um exemplo da estrutura de uma RNC:

Figura 6 - Estrutura básica de uma RNC



Fonte: Adaptado de KUMAR

Observando a figura acima nota-se uma divisão na arquitetura, onde temos a camada de entrada, mais à esquerda, em seguida temos as camadas de convolução responsáveis por aplicar filtros à imagem e coletar características da figura, à medida que o número de camadas aumenta essa coleta de informações se torna mais refinada possibilitando a obtenção de

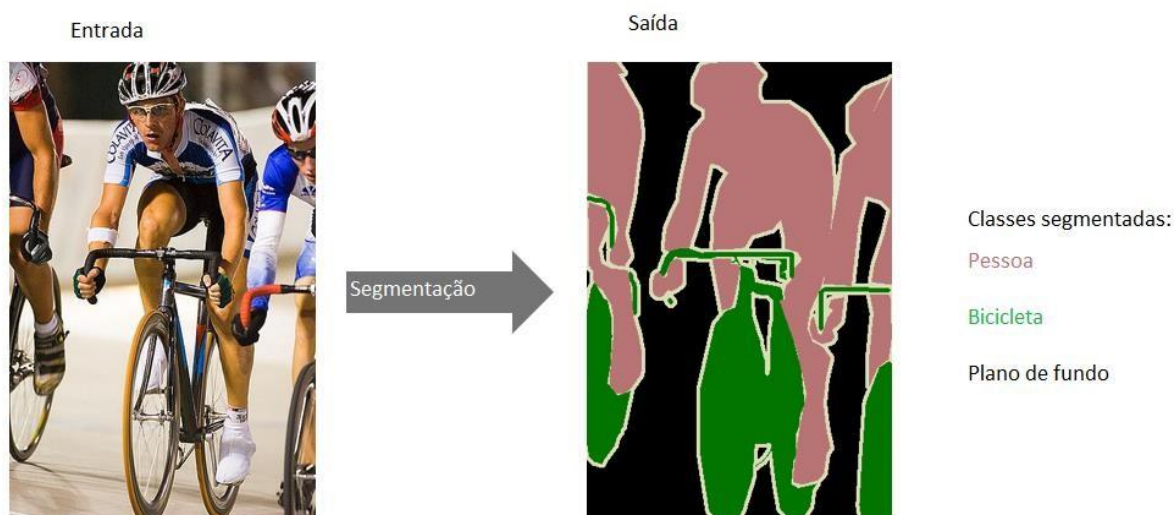
detalhes mais sutis. O resultado passado adiante por essas camadas é chamado de mapa de características, e é enviado ao próximo agrupamento de camadas, que são responsáveis por diminuir o tamanho da entrada facilitando o processamento e diminuindo a memória necessária, além de diminuir os números de parâmetros e acelerarem os treinamentos. Dentro desse processo são realizadas as operações de *Max-pooling* (Agrupamento máximo) e *Average-pooling* (Agrupamento mediano) não necessariamente juntas. Após isso, a estrutura é composta de camadas totalmente conectadas, que são responsáveis por prever o resultado com base nas informações coletadas nas camadas anteriores e informar ao usuário.

2.2.5 UNet

A área de aprendizagem profunda é repleta de modelos para utilização, cada qual com sua devida funcionalidade. O presente projeto fez uso da arquitetura conhecida como UNet, devida a sua recorrente utilização para casos de segmentação de imagens.

O processo de segmentação ou processamento de imagem, consiste em pegar uma imagem e através da aplicação de técnicas de visão computacional classificar cada pixel da imagem a uma classe específica. No caso cada região da figura seria representada por uma coloração diferente correspondente a uma classe específica. A Figura 7 exemplifica a funcionalidade da segmentação semântica.

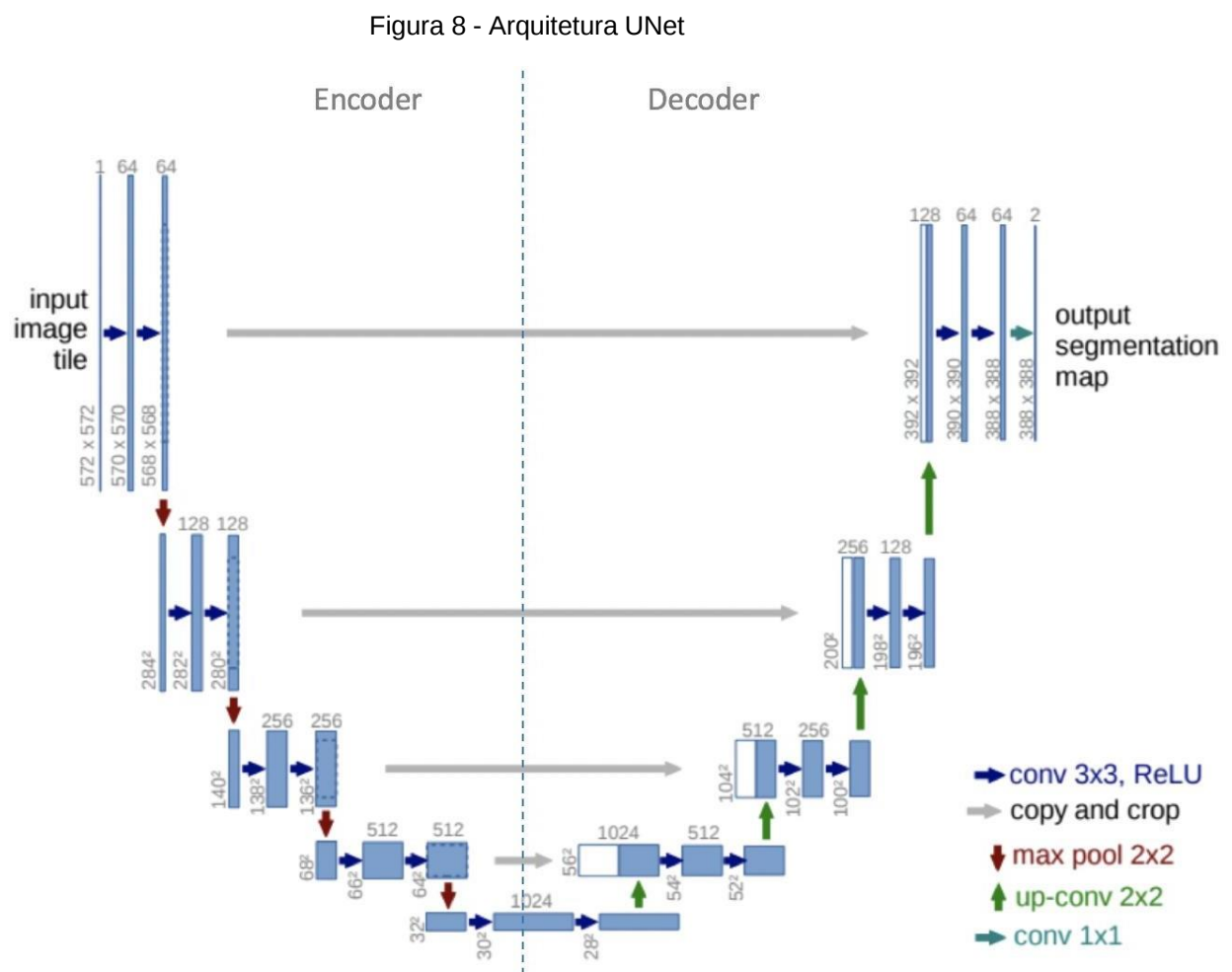
Figura 7 – Exemplo de segmentação semântica



Fonte: Adaptado de LAMBA, 2019

Os treinamentos de redes neurais por vezes requerem um grande número de amostras para apresentarem resultados satisfatórios. A arquitetura UNet, aborda esta etapa com técnicas diferentes das demais arquiteturas, resultando em uma utilização mais eficiente dos exemplos disponíveis para treinamento.

Montada a partir de uma modificação da Rede Neural Totalmente Convolutiva ou do inglês FCNN, o principal objetivo da UNet é reduzir o número de imagens necessárias para o treinamento e fornecer segmentações mais precisas. O nome se dá devido ao formato em que as camadas de contração estão dispostas. Observando a Figura 8 é possível notar a diferença se comparada com as demais arquiteturas.



Fonte: Ronnebløe et al. (2015).

A parte, a esquerda da arquitetura, é responsável por codificar a imagem de entrada. geralmente composta de uma rede de classificação pré-treinada ResNet, que realiza blocos de convoluções seguidos de operações de agrupamento máximo (*Max-pooling*) e redução da

resolução (*downsampling*), para representar em diferentes níveis as características das entradas. Essas características são classificadas a nível de pixels, onde a resolução da imagem é reduzida para que cada pixel seja alcançado.

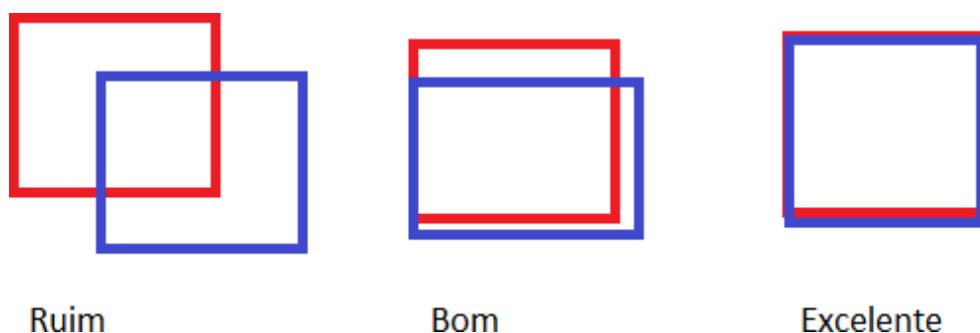
A parte à direita da arquitetura, fica responsável por decodificar a informação obtida pela parte anterior. De forma semântica, ela projeta as características conseguidas em baixas resoluções, e aloca em conjunto de pixels agora numa resolução maior para obter uma classificação mais densa. De forma contrária à primeira metade, as entradas sofrem as operações de aumento de resolução (*upsampling*) e concatenação seguidas de convoluções normais.

2.4 Métricas para avaliação de resultados

A validação de um modelo de RNC é feita por uma métrica que quantifica o quão eficiente é o modelo treinado. Há diversas métricas que realizam essa avaliação, porém há certas áreas que algumas se destacam mais que outras.

Para o caso de segmentação semântica pode ser empregada a técnica conhecida como União sobre Intersecção, ou do inglês, *Intersection over Union* (IoU). A métrica IoU, é utilizada para avaliar a precisão na detecção de objetos em figuras. A Figura 9 demonstra como a métrica classifica a assertividade do modelo

Figura 9 - Exemplos de classificação da métrica.



Fonte: Elaboração própria (2022).

Na figura está representada como acontece a validação da precisão do modelo utilizando a métrica Intersecção sobre união e é com base nesses resultados que a precisão da rede neural é avaliada.

Como o nome sugere ela compara os resultados fornecidos pelo modelo com os resultados reais e sobrepõem os mesmos para avaliar o quão próximos eles são através da Equação 1:

$$IoU = \frac{\text{Área de intersecção}}{\text{Área de união}} \quad (1)$$

O resultado da Equação 1 é um valor positivo menor ou igual a um, onde quanto mais próximo de zero pior é a precisão do modelo e menos sobrepostas estão as áreas, e quanto mais próximo a um, melhor é a precisão do modelo e as áreas estão, praticamente sobrepondo-se.

2.5 Trabalhos Correlatos

Em relação a trabalhos correlatos foi possível encontrar diversos trabalhos que abordam o tema de classificação ou detecção de defeitos em madeira empregando visão computacional. Porém durante a abordagem a grande maioria trata de focar apenas um defeito específico e utilizar outras maneiras de classificação, sendo elas visão computacional clássica, RNCs por região conhecida por R-CNN do inglês, entre outras técnicas.

Dando destaque a dois trabalhos onde ambos classificam madeiras, o primeiro desenvolvido por Alexander Augusto Dellabetha (2018) avalia apenas o defeito nós, classificados entre, vivos, mortos, podres ou pinos, no seu projeto Augusto desenvolve dois modelos de classificadores, um deles o MultiCamadas de Perceptrons e o outro Máquina de vetor de suporte, obtendo uma acurácia de 94% na classificação de nós, uma característica a ser ressaltada por Alexander é que a medida em que o número de nós presentes nas imagens aumentava a precisão do modelo diminuía.

Essa mesma característica, se fez presente no segundo trabalho desenvolvido por Ting He e colaboradores (2019), onde também é realizada a classificação e localização de defeitos presentes em madeiras. Porém de forma diferente do primeiro trabalho, este por sua vez classifica seis defeitos, dentre eles: nós vivos, nós mortos, pigmentação azulada, rachaduras, pigmentação marrom e *pitch streak*. No projeto é implementado um modelo chamado de *Fully*

Convolutional Neural Network e obteve resultados distintos para diferentes métricas utilizadas.

Porém vale destacar o uso da métrica, Intersecção sobre união, ou do inglês *intersection over union*, que teve resultado uma assertividade de 62%. É importante ressaltar que ambos os trabalhos possuem abordagens bem distintas, mudando defeitos estudados, métricas, modelos e arquiteturas implementadas.

3 METODOLOGIA

No presente capítulo foi explicitado todo o processo de desenvolvimento seguido para a confecção de um algoritmo de detecção automatizada de defeitos em madeiras serradas empregando redes neurais convolucionais.

Primeiramente foram explorados os requisitos necessários para a criação do algoritmo. Na sequência foram abordadas quais ferramentas seriam utilizadas para realizar a tarefa. Na seção 3.3 será contextualizada a base de dados empregada no treinamento do modelo. Subsequentemente nas seções 3.4 e 3.5 serão abordados as estratégias utilizadas para treinamento do modelo, bem como a configuração e preparação do ambiente de desenvolvimento utilizado.

3.1 Requisitos de projeto

Para início do projeto foi realizado um levantamento de todos os requisitos a serem atingidos com a criação do algoritmo. Com os padrões estipulados o desenvolvimento do projeto passou pelas seguintes etapas:

- 1) O algoritmo desenvolvido deve ser capaz de identificar defeitos em madeiras;
- 2) A partir da identificação dos defeitos classificá-los em diferentes classes presentes na base de dados.

3.2 Descrição da solução implementada

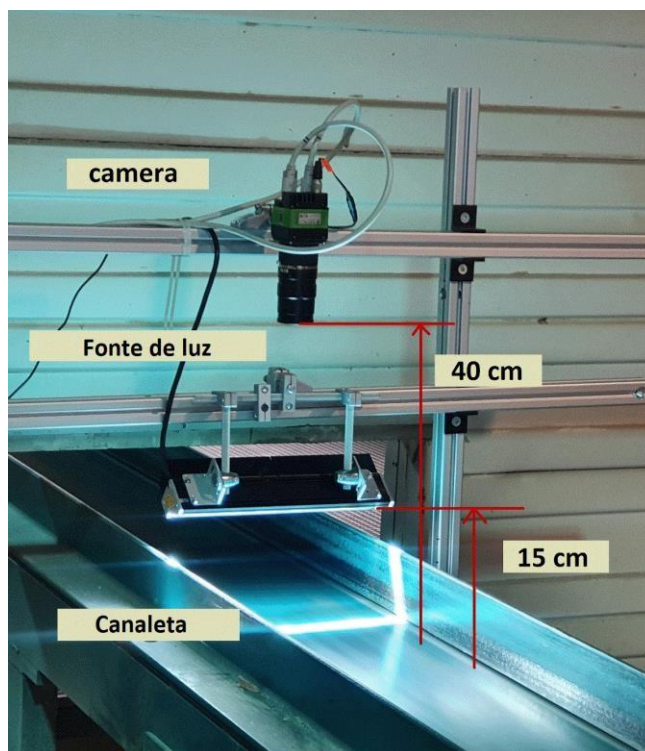
Com os requisitos do projeto voltados para a identificação e classificação dos defeitos em madeiras, foi optado pela criação de um algoritmo implementando técnicas de aprendizado profundo baseado em redes neurais convolucionais. Para a criação do algoritmo também foi utilizada a biblioteca *FastAI*. Além da biblioteca de programação, foram utilizados o *Google Colab* e o *Google Drive* para desenvolvimento do código, e armazenamento e manipulação das imagens, respectivamente.

3.3 Base de dados

Para o presente trabalho optou-se por utilizar uma base de dados de madeiras serradas disponibilizada na internet por Kodytek. P. et. Al (2021) . Segundo Kodytek. P. et. Al. (2021),

grande parte dos trabalhos na área da indústria madeireira com base em aprendizado profundo, são feitos com imagens coletadas pelos próprios pesquisadores, portanto o número de amostras obtidas se limita a uma baixa quantidade de imagens. A base de dados utilizada, consiste em aproximadamente 21.000 imagens de tábuas de madeira serrada retiradas diretamente da linha de produção de uma madeireira e fornece aos pesquisadores amostras na área de controle de qualidade automatizado da indústria. Na Figura 10 é possível observar o equipamento utilizado para realizar a captura das imagens.

Figura 10 – Equipamento de captura de imagens



Fonte: Adaptado de Kodytek. P. et. Al. (2021)

As fotos foram capturadas em blocos de 1024 linhas, o que resultam em imagens com resolução de 1024 x 4096 pixels. Como elas foram capturadas diretamente da linha de produção algumas destas imagens possuem apenas a canaleta por onde passavam as tábuas, portanto foi necessário um tratamento para filtrar e descartar estas imagens da base de dados. Além da filtragem é realizado um corte para reduzir o fundo da imagem a fim de diminuir o tamanho do arquivo bem como retirar áreas indesejadas da foto. Na Tabela 1 podemos ver a distribuição dos defeitos dentro das imagens capturadas no processo.

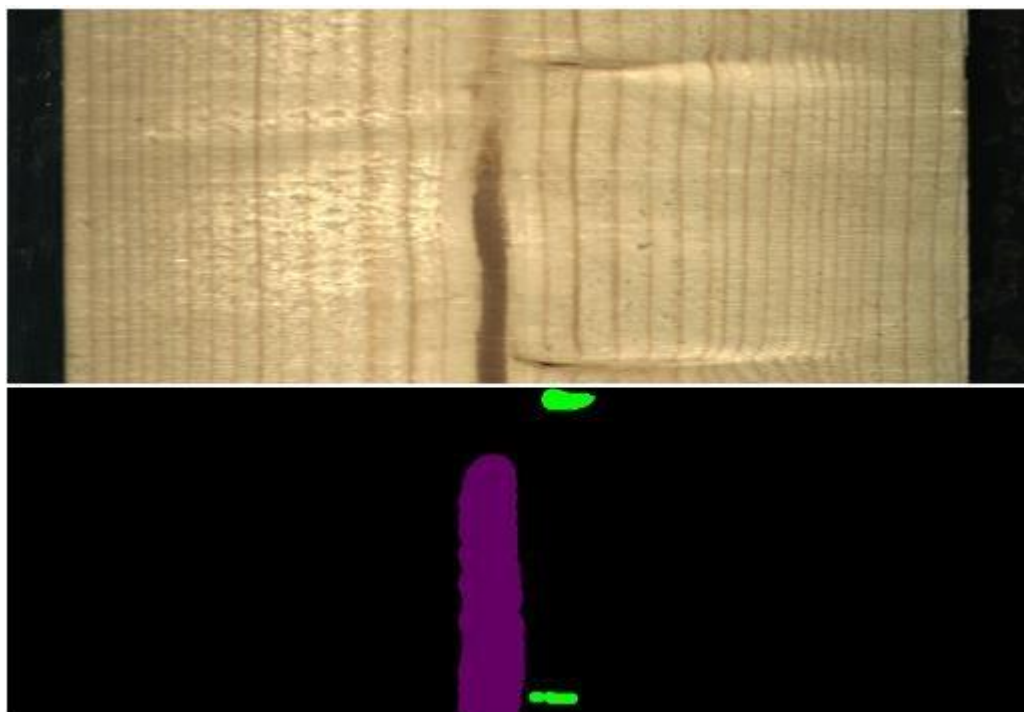
Tabela 1 – Aparição dos defeitos em fotos da base de dados.

Defeito	Cores das classes	Imagens com o defeito	Ocorrência geral no dataset (%)
Nó vivo	Verde	11.912	58,8
Nó morto	Vermelho	8.350	41,2
Nó com rachadura	Amarelo escuro	1.835	9,1
Rachaduras	Rosa	1.578	7,8
Presença de resina	Magenta	2.624	12,9
Medulas	Azul	1.060	5,2
Quartizito	Roxa	847	4,2
Nós faltantes	Laranja	478	2,4
Manchas azuladas	Ciano	77	0,4
Crescimento Excessivo	Verde escuro	6	0,03

Fonte: Adaptado de Bilik P. et. Al. (2021)

Na base de dados também é disponibilizado além da imagem da madeira, uma imagem onde seus pixels assumem valores associados às diferentes classes de defeitos consideradas. Mostrado na Figura 11.

Figura 11 - Imagem da madeira serrada e identificação dos defeitos.



Fonte: Elaboração própria (2022).

Na imagem, na parte superior temos a foto da madeira captura pelo equipamento, e na parte inferior, a indicação dos defeitos presentes na mesma foto. A identificação dos defeitos foi realizada por um profissional especialista com o auxílio de uma ferramenta confeccionada para essa finalidade em específico. Além da imagem indicando os defeitos, foi gerada uma caixa delimitadora para indicar a posição deles. Cada defeito é identificado com uma cor específica e listado na base de dados como foi descrito na Tabela 1 a cima.

A classificação desses defeitos é necessária devido ao fato da presença ou não deles alterarem a qualidade e valor de mercado da madeira em si, portanto a devida classificação dos defeitos se faz necessária para se obter um melhor aproveitamento do material (Hittawe, et al., 2015).

Além das fotos da madeira e da máscara dos defeitos, também foram disponibilizados arquivos de texto contendo as coordenadas das caixas delimitadoras dos defeitos encontrados em cada imagem. Foram disponibilizadas duas maneiras de localizar os defeitos para que os pesquisadores pudessem optar por qual maneira utilizar, no caso do projeto. Foi-se utilizado o método de segmentação semântica utilizando as imagens coloridas.

Durante o processo de criação do algoritmo, surgiu a necessidade de realizar uma alteração na base de dados referente a dimensão das imagens e também a coloração dos defeitos. Essas modificações foram realizadas através de um código desenvolvido em C++ utilizando técnicas de visão computacional clássica onde as imagens passaram por um redimensionamento de 70% do tamanho original, e as cores foram alteradas de tons de RGB para tons de cinza, com a intensidade da iluminação do pixel variando de 1 até 10 sendo atribuídos um valor para cada defeito. Como exposto na Tabela 2 na sequência.

Tabela 2 – Exemplo de como é realizada a rotulação dos defeitos

'Wood'	0
'Live_knot'	1
'Death_knot'	2
'Knot_missing'	3
'Knot_With_Crack'	4
'Crack'	5
'Quartzizity'	6
'Resin'	7
'Marrow'	8
'Blue_Stain'	9
'Overgrow'	10

Fonte: Elaboração própria (2022).

Essa enumeração dos defeitos corresponde com a cor que o pixel foi colorido na máscara informada ao modelo, e é com base nela que o modelo saberá que defeito está presente na imagem em questão. A título de exemplo, na imagem fornecida a rede neural, conter um pixel em que seu valor seja igual a 2 significa que esse pixel será classificado como um defeito do tipo "Death_knot".

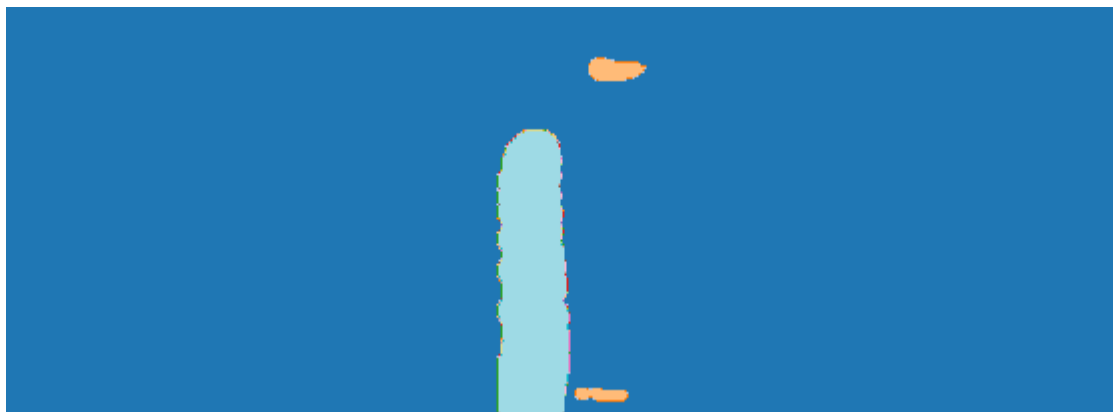
Com todas as alterações realizadas, a base de dados foi enviada para o repositório na conta do *Google Drive*, para em seguida ser utilizada na etapa de treinamento do modelo.

3.4 Abordagem 1: Modelo de classificação com múltiplas classes

Como citado em seções anteriores o ambiente utilizado foi o *Google Collaboratory*, junto das linguagens de programação Python, a biblioteca *FastAI*, e o repositório do *Google Drive* para armazenar as imagens. Com o ambiente inicializado, foi então alocado uma GPU para o desenvolvimento e treinamento do algoritmo utilizando a função *unet_learner*.

Para realizar o treinamento do modelo, inicialmente foram necessários três parâmetros principais, onde o primeiro é o *DataBlock*, que gerencia toda a parte de imagens do banco de dados. A Figura 12, mostra um exemplo de como a rede neural recebe a máscara.

Figura 12 - Máscara fornecida a rede neural.



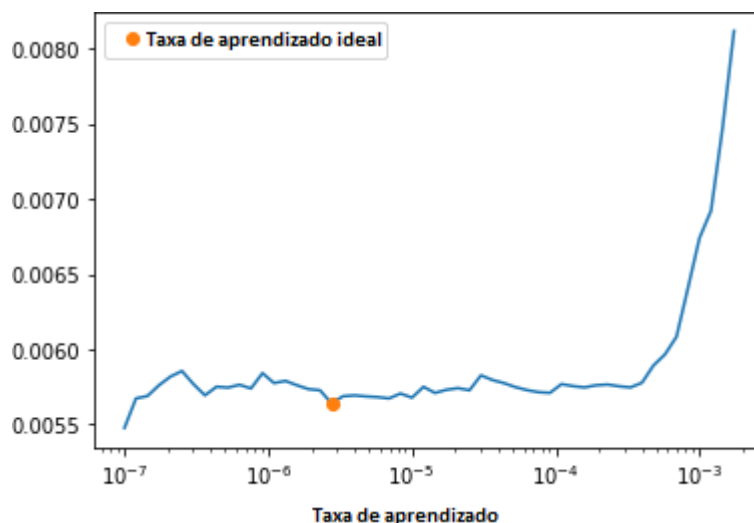
Fonte: Elaboração própria (2022)..

Após as imagens serem fornecidas ao modelo foi necessário dividi-las em imagens para treinamento e imagens para validação, e para isso foi utilizado o parâmetro *Random_splitter*, onde é definido que 70% das imagens do banco de dados serão voltadas ao treinamento, enquanto o restante será para a validação. Por último foi utilizada uma arquitetura *resnet34*, que é uma arquitetura treinada de maneira prévia. O método de treinamento empregado foi o de transferência de aprendizado (*transfer learning*), justamente por ser utilizada uma arquitetura pré-treinada como parâmetro da função de aprendizado. A prática de utilizar arquiteturas treinadas previamente acelera o processo de extração das características e ajustes de resultados, devido ao fato destas arquiteturas, já serem treinadas utilizando pesos definidos de maneira não aleatória, e treinadas em milhares de categorias distintas (GOODFELLOW, et al., 2016). Dentro do método de transferência de aprendizado, foi utilizada a técnica de ajuste fino (*fine_tuning*) que consiste em ajustar os pesos dos modelos treinados previamente conforme se finaliza as épocas de treinamento.

O segundo parâmetro a ser definido foi o número de épocas que o modelo foi treinado, como o nome sugere, esse parâmetro diz respeito a quantidade de vezes em que o modelo será exposto as imagens do banco de dados, este parâmetro é estipulado de forma experimental, podendo variar de uma até centenas de épocas.

Por último, o parâmetro final a ser definido foi a taxa de aprendizado do modelo, que pode ser obtida utilizando uma função onde é realizada a amostragem de uma pequena quantidade de imagens com taxas de aprendizado variadas. Após essa amostragem, o algoritmo gera um gráfico contendo a perda obtida no treinamento em relação a taxa de aprendizado, observado na Figura 13.

Figura 13 – Perda versus Taxa de aprendizado para o modelo de multi-classes.



Fonte: Elaboração própria (2022).

Observando o gráfico é notável que os valores entre 10^{-7} até 10^{-4} o valor de perda se mantém basicamente constante. Já de 10^{-3} em diante, os valores sobem exponencialmente. Se baseando no gráfico e após testes práticos foi estipulado uma taxa de aprendizado de 10^{-4} para o modelo em questão.

Em virtude do *Google Colab* não permitir longas operações por sessões devido ao grande poder de processamento necessário para processar os dados do modelo, foi estipulado um valor de 10 épocas para a realização do treinamento. Com todos os parâmetros definidos é por fim realizado o treinamento e os resultados serão discutidos na seção seguinte.

3.5 Abordagem 2: Modelo de classificação binária

A título de curiosidade um segundo modelo foi treinamento com outra alteração feita na base de dados, onde as classes de defeitos foram agrupadas em apenas uma classe chamada apenas de defeitos, onde o modelo então iria realizar a classificação da madeira, entre defeituosa ou não defeituosa além de localizar os defeitos nas imagens. A classificação acontece do mesmo modo que o modelo anterior, a diferença é que neste segundo caso teria

apenas as classes: “Madeira” com valor igual a 0 e “Defeito” com valor igual a 1. Ou seja, para a rede neural, na imagem sendo fornecida a ela, se o valor do pixel for igual a um, significa que há um defeito na região. Na Figura 14 já é possível notar a diferença na máscara fornecida ao modelo.

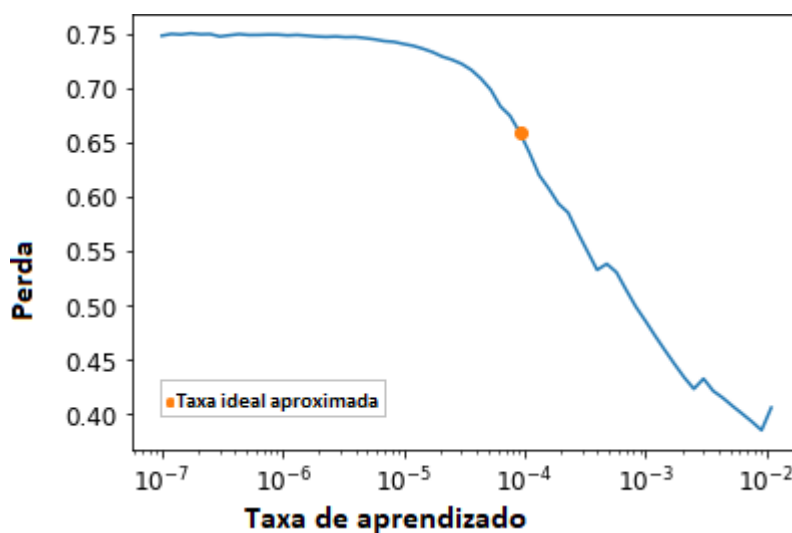
Figura 14 – Máscara fornecida ao modelo de classificação binária.



Fonte: Elaboração própria (2022)..

A definição dos parâmetros para o segundo modelo seguiu os mesmos procedimentos do primeiro, porém o valor para taxa de aprendizado sofreu uma leve alteração. Observando a Figura 15 podemos ver a taxa para o segundo modelo desenvolvido.

Figura 15 – Perda versus a taxa de aprendizado do modelo de classificação binária.



Fonte: Elaboração própria (2022)..

Como visto na figura anterior o valor ideal para a taxa de aprendizado seria entre os valores 10^{-4} e 10^{-2} . Com isso em mente o valor escolhido para realizar o treinamento foi de 10^{-3} .

No próximo capítulo foram expostos os resultados obtidos de ambos os modelos após os treinamentos bem como as considerações finais referentes ao projeto.

4. RESULTADOS OBTIDOS

Nesta seção foi discutido os resultados obtidos após a conclusão do treinamento de ambos os modelos utilizando as arquiteturas resnet18 e resnet34, respectivamente. Como o ambiente de desenvolvimento, GPU e os parâmetros eram semelhantes para os modelos treinados os resultados não se alteraram de maneira expressiva.

4.1 Modelo de classificação com múltiplas classes

Na Tabela 3 são expostos os resultados do modelo de classificação com múltiplas classes, observando as alterações causadas referentes a mudança no uso de arquiteturas no treinamento.

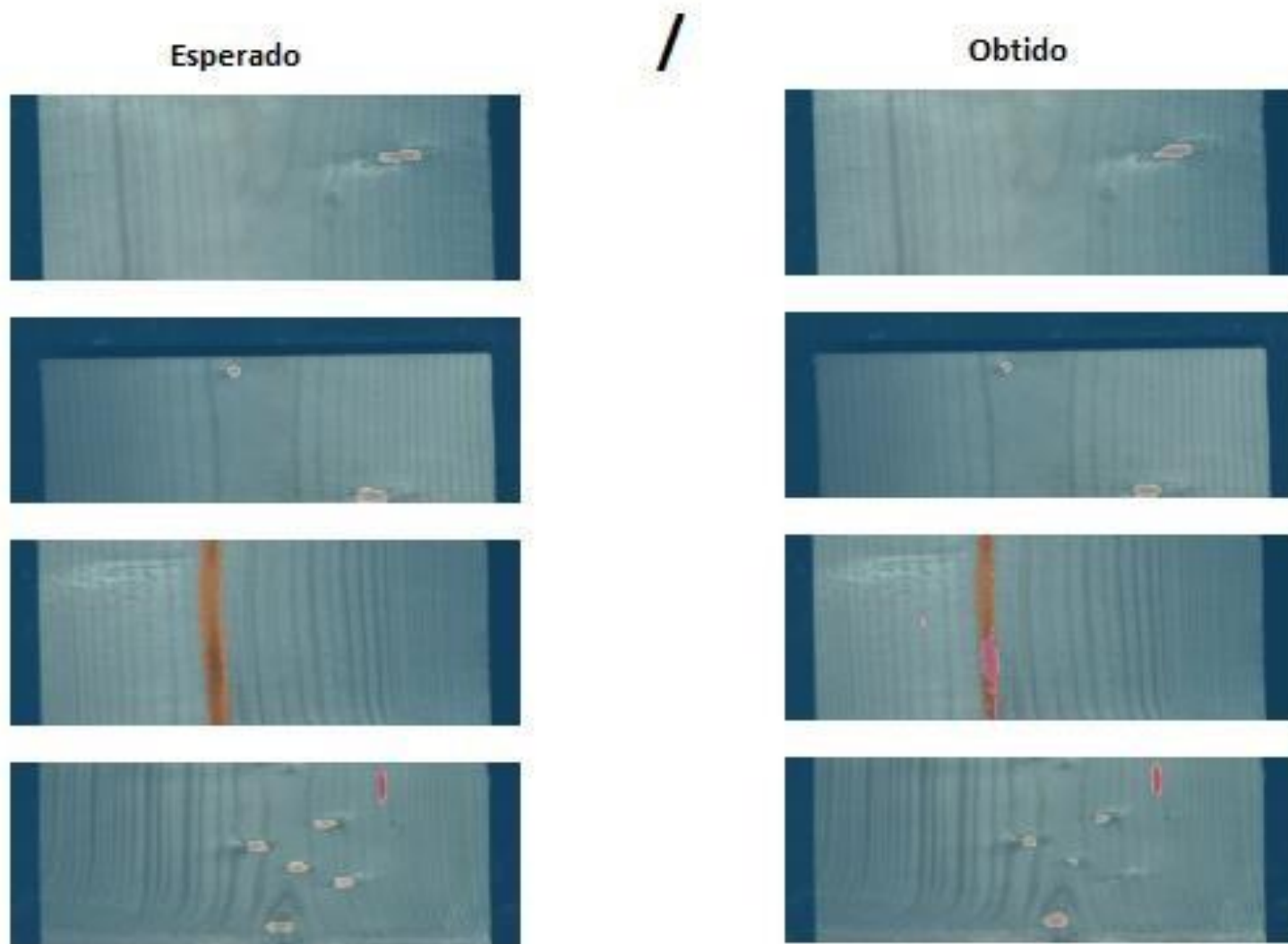
Tabela 3 – Resultados dos treinamentos do modelo de classificação com múltiplas classes

Arquitetura	Rodadas	Taxa de aprendizado	IoU média	Tempo de treinamento (min)
Resnet18	10	10^{-4}	0,6200	95
Resnet34	10	10^{-4}	0,5488	105

Fonte: Elaboração própria (2022).

Como foi exposto na seção 2.4 a métrica utilizada para medir a assertividade dos modelos foi a Intersecção sobre união, e também como exposto o resultado varia entre zero e um. Com base na tabela 3 é possível averiguar que o modelo utilizando a resnet18 se destaca com uma assertividade aproximadamente 10% maior e tempo de treinamento 6% menor que a resnet34. A Figura 16 demonstra comportamento do modelo em relação a classificação dos defeitos utilizando a resnet18.

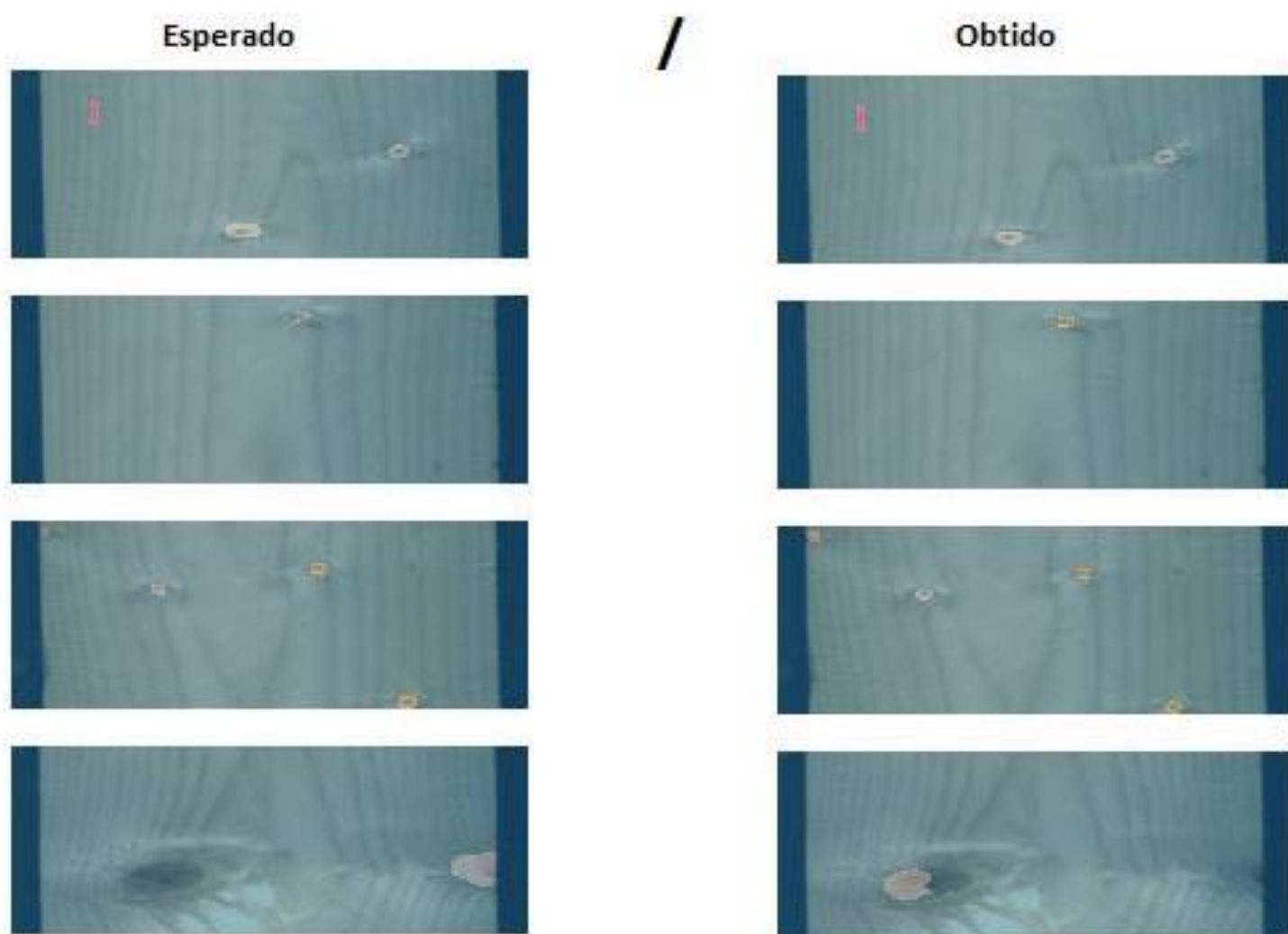
Figura 16 - Comparação dos resultados esperados versus obtidos (resnet18).



Fonte: Elaboração própria (2022).

Na sequência, a Figura 17 traz os resultados esperados e obtidos após o treinamento empregando a resnet34.

Figura 17 – Comparação dos resultados esperados versus obtidos (resnet34).



Fonte: Elaboração própria (2022).

É possível notar que ambos os modelos tiveram erros distintos, enquanto o modelo utilizando as resnet18 errou a classe estipulada ao defeito. O modelo que utilizou a resnet34, errou a localização deles.

Para realizar a comparação, foi exportado uma máscara gerada pelo modelo, que pode ser comparada diretamente com a máscara original. Essa comparação pode ser vista na Figura 18.

Figura 18– Comparação mascara modelo próprio versus original. (Múltiplas classes)



Original



Resnet18



Resnet34

Fonte: Elaboração própria (2022).

4.2 Modelo de classificação binária

De maneira semelhante ao capítulo anterior aqui são expostos os resultados que foram obtidos após o treinamento de dois modelos semelhantes aos anteriores em relação as arquiteturas, porém para o caso de classificação binária

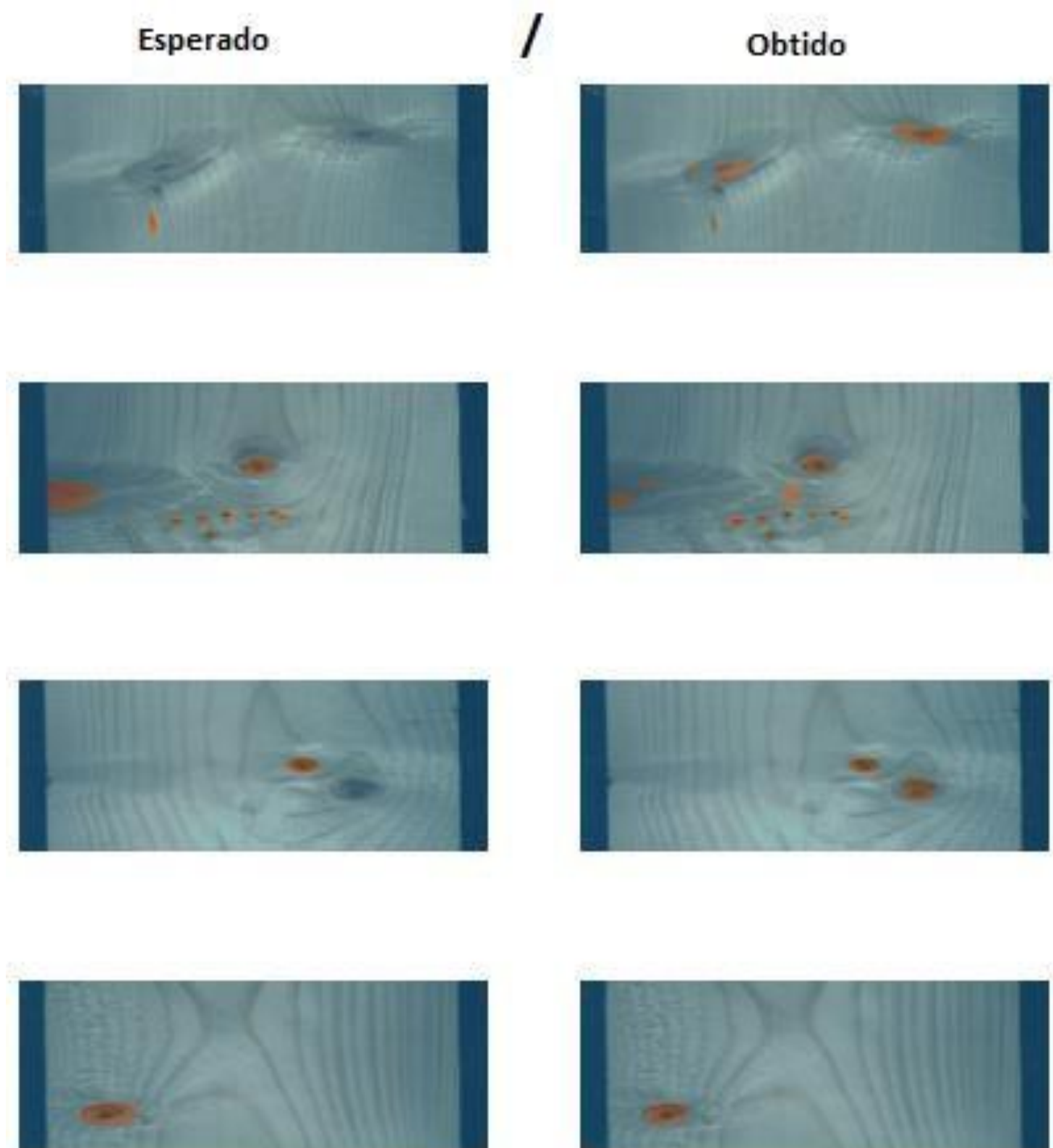
Tabela 4 – Resultados dos treinamentos do modelo de classificação binária

Arquitetura	Rodadas	Taxa de aprendizado	IoU Média	Tempo de treinamento
Resnet18	10	10^{-3}	0.6575	84.3 minutos
Resnet34	10	10^{-3}	0.7660	90 minutos

Fonte: Elaboração própria (2022).

Com base na tabela a cima, é notável que o modelo treinado utilizando a resnet34 se destaca com uma precisão de 0.76 com apenas 6 minutos de diferença no tempo total do treinamento. Nas Figuras 19 e 20, são mostrados a comparação dos resultados esperados e obtidos do modelo para ambas as arquiteturas, respectivamente.

Figura 19 – Comparação do resultado do modelo de classificação binária (resnet18).



Fonte: Elaboração própria (2022).

A comparação obtida demonstra que o modelo desenvolvido utilizando as resnet18 classifica de maneira semelhante ao esperado, errando poucas vezes em relação a área defeituosa.

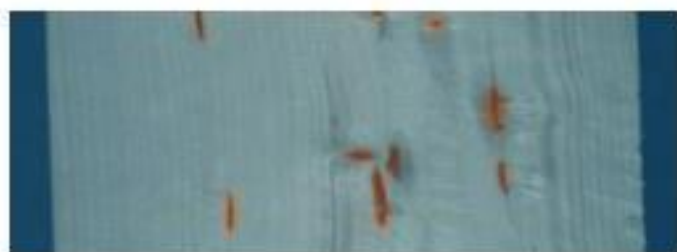
Figura 20— Comparação dos resultados obtidos pelo modelo de classificação binária(resnet34).

Esperado



/

Obtido



Fonte: Elaboração própria (2022).

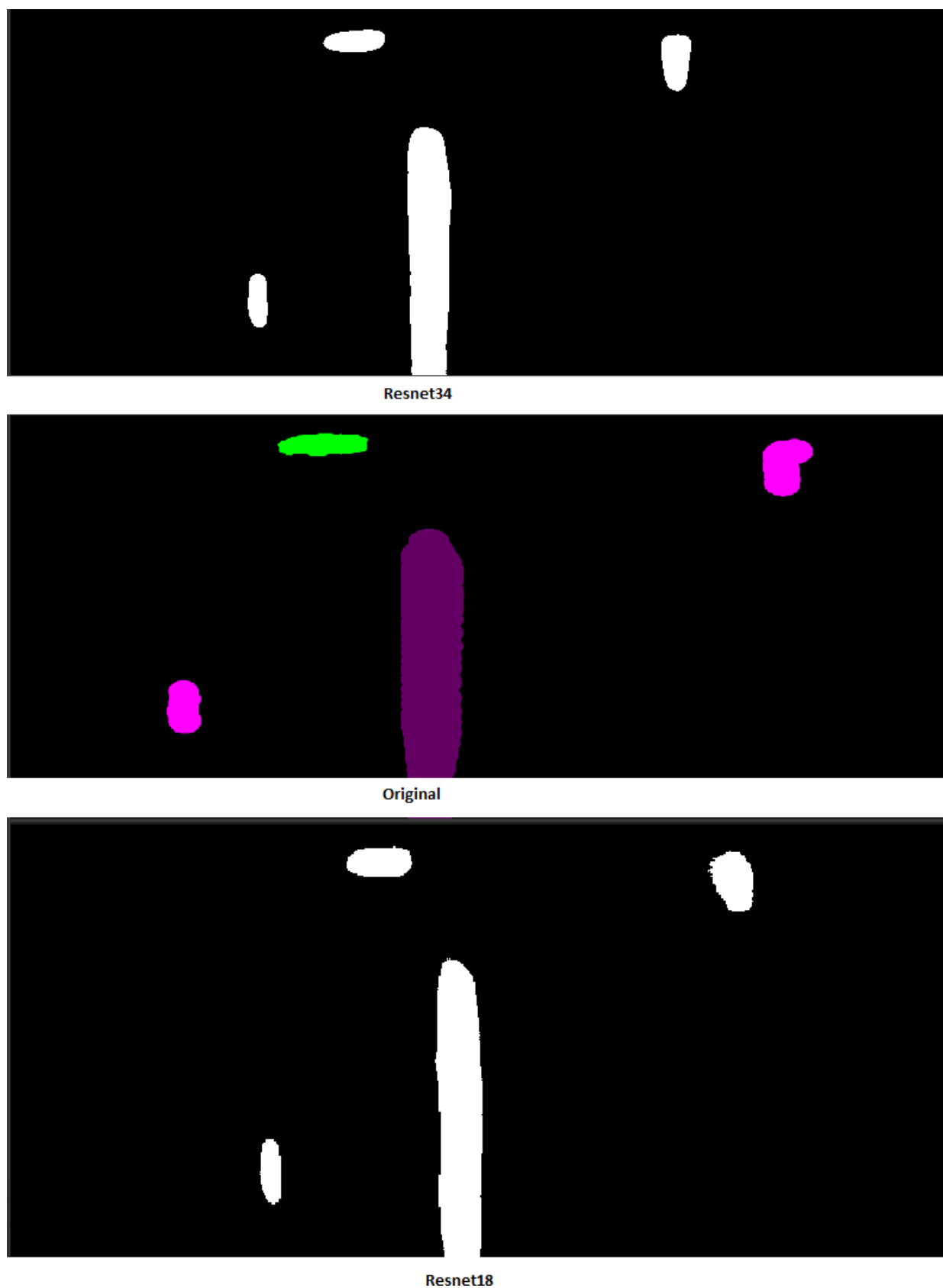
Para o modelo que utiliza a arquitetura resnet34 a classificação de regiões defeituosas ocorreu de maneira semelhante a esperada.

Essa diferença na classificação dos defeitos se dá devido ao fato do treinamento ser realizado utilizando arquiteturas pré-treinadas diferentes para os dois treinamentos.

Ambos os modelos se mostraram promissores, analisando os dados comentados previamente, é possível notar que o modelo da resnet34 se sobressai em relação ao modelo utilizando a arquitetura resnet18. E esse resultado é facilmente notado, quando se compara as Figuras 19 e 20, onde o primeiro demonstrou uma falha em classificar o defeito da madeira, enquanto o segundo demonstra os resultados praticamente iguais aos esperados.

De forma semelhante ao modelo anterior de classificação de múltiplas classes este modelo também permite exportar as máscaras geradas pela própria rede neural. A Figura 21 demonstra essa comparação com a amostra original.

Figura 21 – Comparações das máscaras geradas pelo modelo para ambas as arquiteturas com a máscara original do banco de dados.



Fonte: Elaboração própria (2022).

Os resultados são satisfatórios quando comparados com a amostra original disponível, pois a classificação realizada pelo modelo se assemelha muito com a das amostras originais. Onde as arquiteturas que tiveram destaque para as abordagens de classificação binária e de múltiplas classes, foram respectivamente, a retnet34 e a resnet18

A título de validação dos resultados obtidos pelo treinamento em ambas as abordagens realizadas, foi utilizado o trabalho dos pesquisadores Ting He e colaboradores (2019), citado na seção 2.5, onde foi aplicado a mesma métrica de validação que o presente projeto e teve seu valor calculado em 0,62 de Intersecção sobre união. Segundo Ting He, os resultados obtidos no método proposto tiveram resultados excelentes na detecção de defeitos em madeira. Com isso em mente, e analisando os dados gerados pelos treinamentos é possível dizer que o modelo satisfaz os requisitos estipulado na seção 3.1. A tabela X demonstra os resultado obtidos por Ting He e os do presente projeto, comparando ambos os resultados notasse uma superioridade na métrica utilizada para validação.

Tabela 5 – Comparativo de resultados com outros modelos.

Autoria	Abordagem	Arquiteturas	IoU Média
Ting He, et al.		SegNet	62,93%
Própria	1	ResNet18	62%
	2	ResNet34	76%

Fonte: Do autor.

5. CONSIDERAÇÕES FINAIS

O presente trabalho descreve o desenvolvimento de um algoritmo de classificação de defeitos em madeiras. Para que os requisitos estipulados no início do projeto fossem cumpridos foram aplicadas técnicas de aprendizado de máquina focadas na área de aprendizado profundo.

O trabalho demonstra que é possível a confecção de um algoritmo empregando técnicas de visão computacional para a solução de problemas repetitivos no ramo da indústria madeireira afim de trazer automação ao procedimento de controle de qualidade da madeira. Os modelos desenvolvidos foram ambos treinados pensando não só na classificação de regiões defeituosas como também em uma classificação mais específica do defeito em si, podendo diferencia-los uns dos outros. Para a classificação binário de defeitos o modelo empregando a arquitetura resnet34 demonstrou resultados superiores, com uma precisão de 0,65 sendo treinado por apenas 10 épocas e tempo de 90 minutos. Já o modelo com múltiplas classes de defeito, o que se destacou foi o modelo utilizando a arquitetura resnet18, com uma precisão de 0,62 e um tempo de treinamento de aproximadamente 95 minutos.

Para o presente trabalho foram utilizadas duas arquiteturas com base em estudos de trabalhos correlatos disponíveis, tais arquiteturas treinadas previamente foram utilizadas para que a coleta de características das imagens ocorresse de maneira mais acelerada. A título de validação e sugestão para trabalhos futuros, sugere-se a comparação dos resultados obtidos pelos presentes modelos com outros desenvolvidos utilizando arquiteturas diferentes das expostas no trabalho, bem como modelos de classificação diferentes dos que utilizam da segmentação semântica. Outra sugestão seria a utilização de uma métrica que permita a criação de uma matrix de confusão para a indicação dos acertos e erros do modelo durante o treinamento.

Também se faz necessário ressaltar a utilização de ferramentas disponibilizadas de forma gratuita, o que gera uma limitação na parte de manipulação das imagens do banco de dados. Como por exemplo uma limitação de memória no *Google Drive*, o que força o usuário utilizar parcialmente a base de dados disponível. Caso não houvesse tal limitação seria possível fornecer mais material de aprendizado ao modelo desenvolvido bem como uma possível melhora nos resultados apresentados. Além disso há uma limitação no *Google Colab* que força o usuário a limitar os números de épocas em que o modelo será treinado, o que pode ser contornado comprando uma licença para o mesmo.

REFERÊNCIAS

1. M. M. Hittawe, S. M. Muddamsetty, D. Sidibé and F. Mériaudeau, "Multiple features extraction for timber defects detection and classification using SVM," 2015 IEEE International Conference on Image Processing (ICIP), 2015, pp. 427-431, doi: 10.1109/ICIP.2015.7350834.
2. Kodytek P, Bodzas A and Bilik P. **A large-scale image dataset of wood surface defects for automated vision-based quality control processes [version 1; peer review: 1 approved with reservations]**. F1000Research 2021.
3. Dellabetha, A. A.. **Sistema Industrial Para Aquisição de Imagens e Classificação de Nós em Madeira Serrada**. UNIVERSIDADE DE CAXIAS DO SUL ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E ENGENHARIAS. 2018.
4. ABIMCI, **Setor madeireiro em 2021, crescimento consolidado**. ABIMCI, 2021. Disponível em: <https://abimci.com.br/setor-madeireiro-em-2021-crescimento-consolidado/>. Acesso em: 10 Mar. 2022.
5. ABIMCI, **Exportações primeiro trimestre de 2021**. ABIMCI, 2021. Disponível em: <https://abimci.com.br/exportacoes-primeiro-trimestre-de-2021/> . Acesso em: 10 Mar. 2022
6. DIJKSTRA, E.W. **A Discipline of Programming**. Prentice-Hall, Englewood Cliffs, NJ. 1976
7. RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence: A Modern Approach Third Edition**. [S.l.: s.n.], 1994.
8. BRIGNOLI, RAMON, **Estudo de Redes Neurais Convolucionais E Sua Aplicação Para Classificação De Ambientes**. INSTITUTO FEDERAL DE CIÊNCIA E TECNOLOGIA DE SANTA CATARINA. 2021
9. FERNEDA, Edberto. **Redes Neurais e Sua Aplicação em Sistemas de Recuperação de Informação**, Brasília, v. 35, n. 1, p. 25-30, jan./abr. 2006. Disponível em: <https://www.scielo.br/j/ci/a/SQ9myjZWLxnyXfstXMgCdch/?format=pdf&lang=pt>. Acesso em: 30 Set 2022.

10. RONNEBERGER, Olaf, FISCHER, BROX, Thomas. **U-Net: Convolutional Networks for Biomedical Image Segmentation**, 2015; Disponível em: [arXiv:1505.04597](https://arxiv.org/abs/1505.04597) Acesso em 14 Out. 2022.

11. KUMAR, Ajutesh. **Different Types of CNN Architectures Explained**. Data Analytics. Disponível em: <https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/> Acesso em: 16 Out. 2022.

12. LAMBA, H. Understanding Semantic Segmentation with UNET. 2019. Disponível em: <https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47> Acesso em: 14 Set. 2022.

13. Gu, I.Y.H., Andersson, H., Vicen, R. (2009). **Automatic Classification of Wood Defects Using Support Vector Machines**. In: Bolc, L., Kulikowski, J.L., Wojciechowski, K. (eds) Computer Vision and Graphics. ICCVG 2008. Lecture Notes in Computer Science, vol 5337. Springer, Berlin, Heidelberg. Disponível em: https://doi.org/10.1007/978-3-642-02345-3_35 Acesso em: 20 Nov. 2022

LEMOS, Yessica Maria Valencia. **Inspeção automática de defeitos em ovos comerciais usando visão computacional**. 2021. Dissertação (Mestrado em Engenharia de Automação e Sistemas) – UFSC, Florianópolis, 2021. Disponível em: <https://repositorio.ufsc.br/handle/123456789/227084> . Acesso em: 20 jun. 2021.

He, Ting & Liu, Ying & Xu, Chengyi & Zhou, Xiaolin & Hu, Zhongkang & Fan, Jianan. (2019). **A Fully Convolutional Neural Network for Wood Defect Location and Identification**. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2937461.