UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE FÍSICA E ESCOLA DE ENGENHARIA

Uso de redes neurais de convolução para classificação de anomalias em arames metálicos a partir de imagens de inspeção por partícula magnética

Julia Corrêa Remus

Porto Alegre, 15 de fevereiro de 2023

SUMÁRIO

1 Introdução 1
1.1 Definição do problema 2
1.2 Classificação de imagens 2
1.3 Objetivos específicos 3
2 Revisão Bibliográfica 4
2.1 Redes neurais de convolução 4
2.1.1 Camada de convolução 4
2.1.2 Camada de pooling 5
2.1.3 Camadas densas 5
2.1.4 Função de ativação 5
2.1.5 Arquitetura geral das redes neurais de convolução 6
2.2 Treinamento 6
2.2.1 *Data augmentation* 7
2.2.2 Custo 7
2.2.3 Regularização 8
2.2.4 Aprendizado por transferência 8
2.3 Trabalhos relacionados 9
3 Metodologia 10
3.1 Descrição do conjunto de dados 10
3.1.1 Classificação dos defeitos 10
3.1.2 Banco de dados 10
3.2 Algoritmo proposto 10
3.2.1 Arquitetura utilizada 14
3.2.2 Treinamento 14
3.3 Sistema utilizado 14
3.4 Métricas de avaliação 15
4 Resultados e Discussão 16
4.1 Classificação das Imagens 16
4.2 Classificação das Regiões 16
4.3 Classificação dos Defeitos 19
5 Conclusão 23

REFERÊNCIAS BIBLIOGRÁFICAS 25

SIGLAS

CIMM Centro de Informação Metal Mecânica.

CNN Redes Neurais de Convolução.

DL Deep Learning.

LIME Local Interpretable Model-Agnostic Explanation.

ML Machine Learning.

PM Partícula Magnética.

ReLU Rectified Linear Units.

1 INTRODUÇÃO

Dutos flexíveis são estruturas utilizadas em sistemas de produção submarinos nas indústrias de óleo e gás, em especial, em condições de altas temperaturas e pressões. São formados por múltiplas camadas poliméricas com alta rigidez e camadas metálicas em formato helicoidal, as quais permitem tanto a flexibilidade quanto a habilidade de aguentar os ambientes em que são expostos. Por suas características, são muito utilizados em poços no Brasil, no Mediterrâneo, no Extremo Oriente, entre outros locais que possuem condições de temperatura e pressão elevadas (BAI; BAI, 2014a; BAI; BAI, 2014b).

Na Figura 1.1 é apresentado um exemplo esquemático de duto flexível. Observando da parte interna para a externa, é formado pelas camadas: carcaça (metálica), barreira de pressão (polimérica), armadura de pressão (metálica), fita anti-desgaste (polimérica), armadura de tração (metálica), fita anti-desgaste (polimérica), armadura de tração (metálica), camada *antibirdingcaging* (polimérica) - o nome remete a um modo de falha potencial - e a capa externa (polimérica). Cada uma das camadas possui um objetivo específico e suas características físicas e geométrias podem variar conforme a aplicação e o entendimento dos fabricantes, também podem variar a sua disposição e quantidade (BAI; BAI, 2014a; BAI; BAI, 2014b).

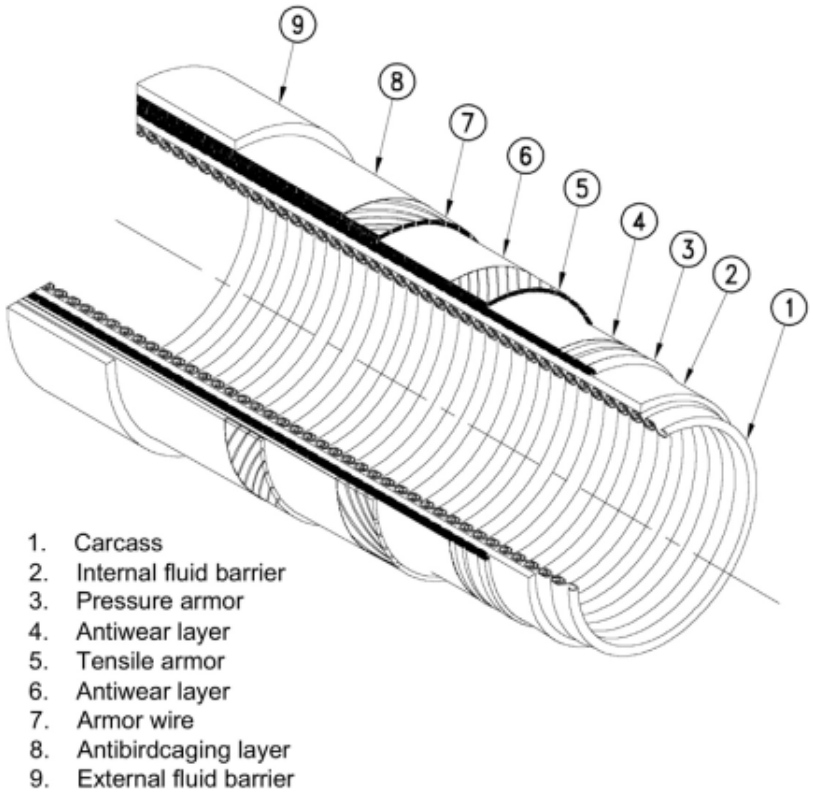


Figura 1.1 – Esquemático de um duto flexível (BAI; BAI, 2014a).

Segundo Bai e Bai (2014b), a complexidade das estruturas desenvolvidas, associada com os ambientes em que são aplicadas, ressalta a importância de possuir um plano de gerenciamento de integridade dos dutos flexíveis. Conforme Souza et al. (2003), esse plano busca mitigar perdas humanas e de produção, evitar a poluição ambiental e aumentar o aproveitamento de equipamentos e instalações. Ademais, os

autores afirmam que para desenvolver e embasar as decisões é necessário a maior quantidade possível de dados históricos, compreendendo todas falhas encontradas.

Realizando a inspeção por Partícula Magnética (PM) nos arames metálicos dos dutos consegue-se entender melhor se aquela estrutura conseguiu alcançar os objetivos esperados ou se houve formação de defeitos nos materiais. Os defeitos nos arames podem ser motivados pela fadiga da estrutura e corrosão por entrada de água e presença de CO_2 e H_2S nas camadas metálicas, conforme apontado por Bai e Bai (2014b).

O uso da inspeção por PM tem como objetivo identificar descontinuidades superficiais e sub-superficiais em materiais ferromagnéticos. É um método não destrutivo aplicado nas indústrias ferroviária, aeroespacial, entre outras, para avaliação de integridade desses materiais (ANDREUCCI, 2007; ASTM INTERNATIONAL, 2021; YANG et al., 2022; TOUT et al., 2021).

Conforme ANDREUCCI (2007):

> O processo consiste em submeter a peça, ou parte desta, a um campo magnético. Na região magnetizada da peça, as descontinuidades existentes, ou seja, a falta de continuidade das propriedades magnéticas do material, irão causar um campo de fuga do fluxo magnético. Com a aplicação das partículas ferromagnéticas, ocorrerá a aglomeração destas nos campos de fuga, uma vez que serão por eles atraídas devido ao surgimento de pólos magnéticos. A aglomeração indicará o contorno do campo de fuga, fornecendo a visualização do formato e da extensão da descontinuidade (ANDREUCCI, 2007).

1.1 Definição do problema

O seguinte projeto tem como objetivo conseguir classificar os tipos de defeitos encontrados em arames metálicos a partir de imagens de inspeção por Partícula Magnética. Para isso, o trabalho utilizará o banco de dados da empresa SIMEROS[1] e as suas definições de defeitos.

1.2 Classificação de imagens

Na literatura é possível encontrar trabalhos com classificação de defeitos de superfícies que podem ser divididos entre os que utilizam algoritmos de *Machine Learning (ML)* (como *support vector machine* (SVM) e *random forests*) e os de *Deep Learning (DL)* (como as redes neurais de convolução). Indolia et al. (2018) cita que a grande diferença entre os dois métodos é a necessidade da extração prévia das características para o primeiro, enquanto no segundo, as características que melhor definem os dados são obtidos ao longo do aprendizado da rede neural (YANG et al., 2022; CUNHA, 2020).

Além disso, conforme apontado por Cunha (2020) e por Yang et al. (2022), algoritmos de *DL* conseguem extrair os resultados necessários de melhor forma - comparado aos de *ML* - em imagens com maior ruído de fundo e variabilidade de iluminção, contraste, entre outras características.

Assim sendo, a partir da revisão bibliográfica e considerando que as imagens cadastradas tem alta variabilidade de fundo, é escolhido utilizar a arquitetura de *Deep*

[1] Site da empresa: https://simeros.com/en/

Learning chamada Redes Neurais de Convolução (CNN) para a realização da classificação das imagens. Também foi levado em conta a dificuldade de seleção das características que definem os grupos estabelecidos, a fim de realizar sua extração e posterior aplicação de algoritmos de *ML*.

1.3 Objetivos específicos

Desta forma, o trabalho tem como objetivo o desenvolvimento e aplicação de um algoritmo de CNN que consiga classificar os defeitos definidos pela empresa SIMEROS, proprietária do banco de dados. Os defeitos são classificados em quatro grupos, podendo ser separados em dois grupos de tipos de defeito e especificados na quantidade de defeitos encontrados; esse tópico será exposto com mais detalhe na seção 3.1.1 - Classificação dos defeitos.

Com a finalidade de construir o projeto, serão levados em conta trabalhos relacionados, como os publicados por Tout et al. (2021), Yang et al. (2022), Kou et al. (2022) e Lee et al. (2019). Para o treinamento e desenvolvimento, serão aplicadas técnicas de *data augmentation* e testado a aplicabilidade do aprendizado por transferência; tais conceitos serão abordados melhor na seção 2 - Revisão Bibliográfica.

A avaliação do algoritmo se dará a partir da estatística de acurácia de teste e a matriz de confusão (verificar seção 3.4). É de extrema importância que a rede seja capaz de verificar a existência do defeito e classificá-lo em um dos dois grandes grupos.

O algoritmo proposto consiste em subdividir as imagens originais em tamanhos menores e realizar duas classificações nesses registros: primeiro a verificação do tipo de região encontrada, se é imagem de fundo ou realmente se encontra o defeito, com posterior classificação nas classes definidas pela empresa. Entende-se que o uso das imagens inteiras para a classificação traria uma dificuldade para o modelo entender qual a região correta para realizar a escolha; mesmo que para esse treinamento a estrutura de dados já esteja definida e, por conseguinte, poderia ser utilizado um maior número de registros para o treinamento. Essa hipótese é verificada e apresentada na seção 4 - Resultados e Discussão juntamente com os resultados do algoritmo proposto.

2 REVISÃO BIBLIOGRÁFICA

2.1 Redes neurais de convolução

Segundo Goodfellow et al. (2016) (tradução nossa): "Redes Neurais de Convolução (CNN) são um tipo especializado de rede neural para processamento de dados que possuem uma topologia conhecida do tipo matriz". São redes neurais que aplicam convolução (em pelo menos uma camada) ao invés de realizar uma multiplicação matricial de toda entrada com os vetores peso, como no caso de outras redes neurais.

Ainda conforme Goodfellow et al. (2016), utilizar a operação de convolução faz com que o modelo necessite de menos memória, melhore a eficiência estatísitica e que seja necessário menos operações para computar a saída.

De acordo com Indolia et al. (2018), uma CNN consiste em quatro elementos principais: a camada de convolução, a camada de *pooling*, função de ativação e a camada densa ou *fully connected layer*.

2.1.1 Camada de convolução

A camada de convolução é a responsável pela extração de propriedades da matriz de entrada, I, por meio de filtros (matrizes de dimensões menores), k, exemplificada na Figura 2.1. A partir da aplicação de k filtros em uma matriz de entrada são obtidas k matrizes de saída ou *feature maps* (JAMES et al., 2021).

A matriz I é formada pelos *pixels* da imagem original ou das matrizes convolucionadas que já passaram pela camada de *pooling* (seção 2.1.2) e pela função de ativação (seção 2.1.4). Já os filtros são matrizes de dimensão menor com valores específicos para extração de propriedades. Alguns exemplos de atributos a serem retirados são as bordas presentes, uma curvatura específica ao longo da imagem, intensidade de cores. No modelo CNN, os parâmetros quantitativos que compõe os filtros são aprendidos ao longo do treinamento da rede, esse aprendizado é realizado escolhendo os parâmetros que minimizem a função de custo (seção 2.2) (JAMES et al., 2021).

Quando as imagens são coloridas, a entrada é formada por três canais de cor (sistema RGB), ou seja, são três matrizes de entrada. Elas são convolucionadas separadamente pelos filtros que possuem profundidade igual a da quantidade de canais (BEZDAN T., 2019).

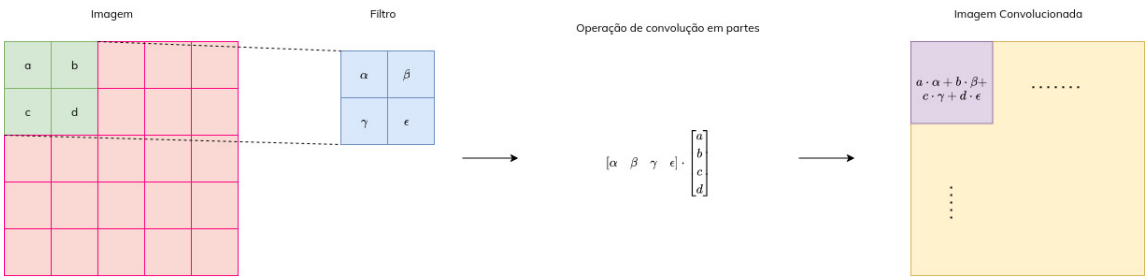


Figura 2.1 – Operação de convolução.

Outros dois parâmetros importantes na convolução são o *stride* e o *padding*. O primeiro significa a quantidade de *pixels* que a operação de convolução pulará, quando o *stride* é igual a um todos os pixels sofrem a operação. O segundo parâmetro mostra como os filtros atuam em entradas com tamanho não múltiplo (em especial nas bordas da matriz): aplicando a convolução apenas onde o filtro consiga sobrepor *pixels* válidos

ou completando com valores nulos nos espaços faltantes. Ambos, juntamente com o número de filtros e o tamanho da entrada, determinam o tamanho da saída (BEZDAN T., 2019).

Segundo Goodfellow et al. (2016), a operação de convolução utilizando a entrada de imagem, I, com um filtro também com duas dimensões, k, gera uma matriz de saída S descrita pela equação (2.1); nela i são os índices que correspondem às linhas e j as colunas, de mesma forma os valores m e n.

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i-m, j-n) \cdot k(m,n) \tag{2.1}$$

2.1.2 Camada de pooling

Essa camada realiza a diminuição de dados gerados pela camada de convolução. A forma mais utilizada de redução de dimensionalidade é o *max pooling*, o qual seleciona os maiores valores dentro de um intervalo de dados da matriz (JAMES et al., 2021; INDOLIA et al., 2018).

Com essa camada, conforme Goodfellow et al. (2016), mantém-se o resultado invariante a pequenas modificações da entrada. Também de acordo com os autores, essa é uma ótima característica se o objetivo for localizar a região, mas, não necessariamente, saber o exato local do foco da busca.

2.1.3 Camadas densas

A camada densa (ou *fully connected*) é utilizada após repetidas aplicações das camadas de convolução e de *pooling*, quando os dados já estão suficientemente reduzidos. Nesse estágio, os dados podem ser representados em um vetor ou em uma matriz de baixa dimensionalidade (mais conhecido como *flatten*), ao invés da matriz de entrada formada pela imagem (JAMES et al., 2021).

Nela é realizado a multiplicação matricial entre a saída do estágio de extração de características e redução de dimensionalidade (verificar seção 2.1.5) e um vetor de pesos. Esse processo pode acontecer com apenas uma mas também com várias camadas densas. Por fim, o resultado é passado em uma função de ativação *softmax* (ver seção 2.1.4) para geração das probabilidades da classificação do modelo (INDOLIA et al., 2018).

2.1.4 Função de ativação

A função de ativação adiciona a não linearidade ao modelo, dando a possibilidade para o aprendizado. É utilizada pelo método após a aplicação das convoluções nos dados gerados e após as camada densas (GOODFELLOW et al., 2016).

A função Rectified Linear Units (ReLU) é a função recomendada para redes neurais *feedfowards* (como a CNN), sendo descrita pela equação (2.2) e apresentado pela Figura 2.2. É utilizada nas camadas de convolução e nas camadas densas que antecedem a última camada de classificação (JAMES et al., 2021; GOODFELLOW et al., 2016).

$$f(z) = \begin{cases} 0, z < 0 \\ z, z >= 0 \end{cases} . \tag{2.2}$$

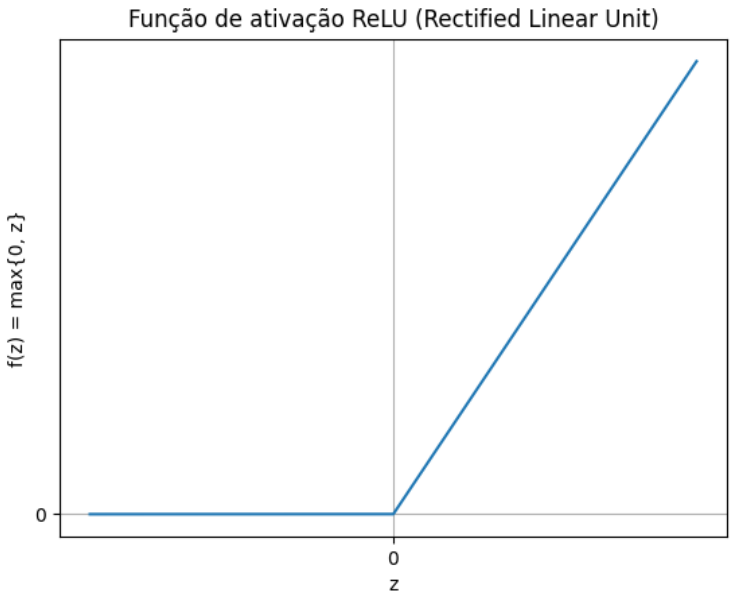


Figura 2.2 – Função de ativação ReLU.

Já para a classificação das caracteríticas, é utilizada a função *softmax*, por retornar as probabilidades de cada tipo de classificação. Pode ser utilizada ao longo do modelo quando o objetivo é escolher entre parâmetros, no entando, é mais utilizada após a última camada densa. Ela é descrita pela equação (2.3), onde $\mathbf{x}$ é um vetor de pesos, i é o índice de um valor desse vetor que contém um total de n valores (GOODFELLOW et al., 2016).

$$f(\mathbf{x})_i = \frac{\exp x_i}{\sum_j^n \exp x_j}. \tag{2.3}$$

2.1.5 Arquitetura geral das redes neurais de convolução

De modo geral, observando as considerações dos autores Goodfellow et al. (2016), James et al. (2021), e Indolia et al. (2018), o algoritmo de CNN pode ser descrito pelo fluxograma apresentado na Figura 2.3.

São exemplos de arquiteturas já treinadas de CNN as redes VGG, ResNet e GoogLeNet, elas possuem diferentes tipos de estruturas de camadas e funções de ativação (ALMEIDA, 2020).

2.2 Treinamento

O treinamento é o processo de, a partir de um conjunto de dados conhecido, ajustar o modelo com seus parâmetros, a fim de obter os resultados apresentados. Uma definição importante para o treinamento é o número de épocas utilizado: ele consiste no número de vezes que o algoritmo de ajuste (para mais informações sobre cálculo de custo e otimização ver 2.2.2) passa pelo conjunto de treinamento (JAMES et al., 2021).

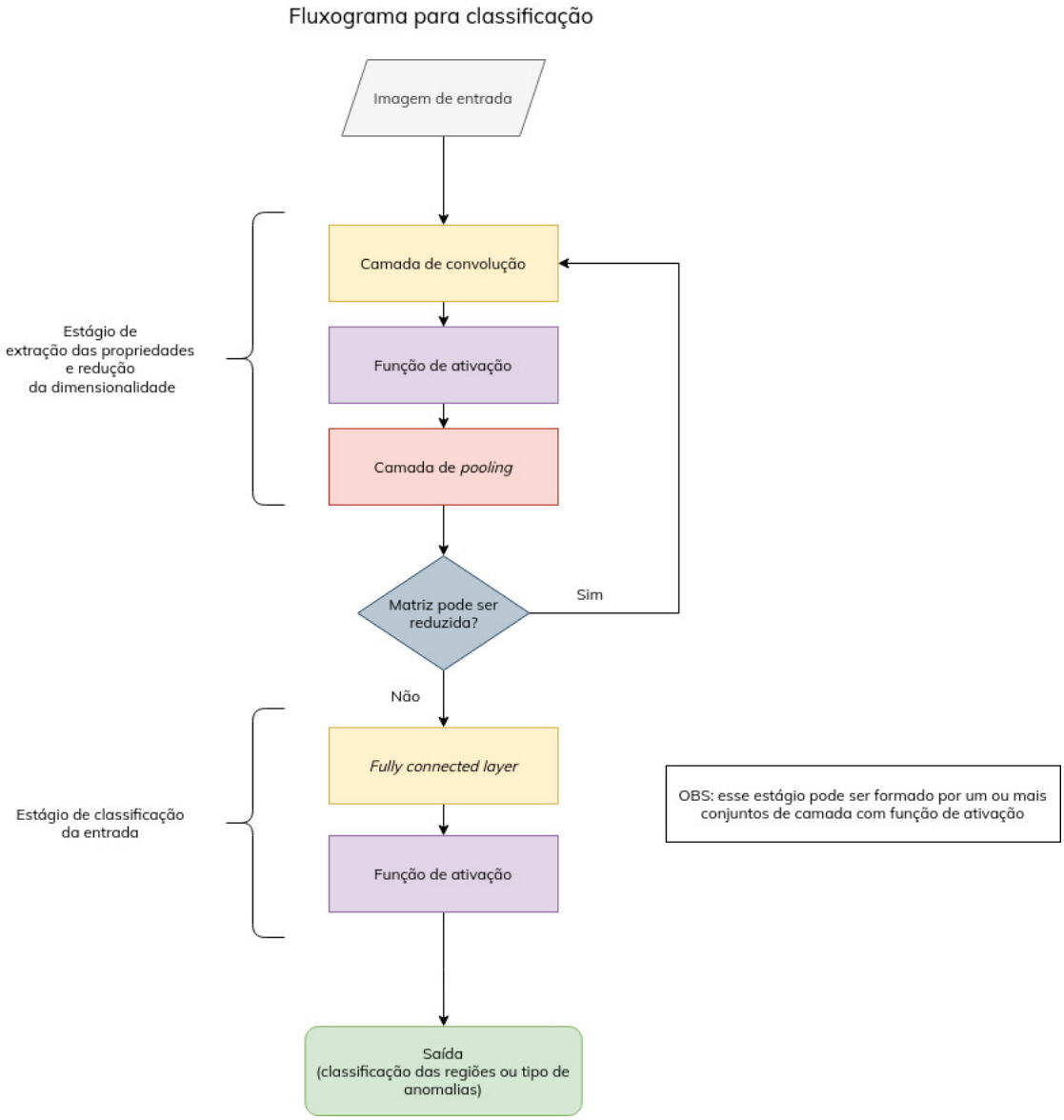


Figura 2.3 – Fluxograma para classificação a partir do algortimo CNN.

2.2.1 *Data augmentation*

Data augmentation é o termo utilizado para a ação de criação de dados falsos a partir de modificações que não interferem na classificação da imagem de entrada. Exemplos de modificações que podem ser aplicadas são rotações, mudanças no contraste, brilho, cisalhamento das imagens. Esse operação auxilia a rede no entendimento de quais parâmetros realmente devem ser levados em conta para a escolha da classe, diminuindo o *overfitting* do modelo (GOODFELLOW et al., 2016).

2.2.2 Custo

A atualização dos parâmetros (e por consequência realizar o aprendizado da rede) é realizada aplicando algoritmos otimizadores que buscam a minimização da função de custo; são exemplos: o gradiente descendente estocástico, Adam e RMSProp (GOODFELLOW et al., 2016).

Para a estimativa do custo pode ser utilizada a função *cross-entropy*, definida

pela equação (2.4), onde N é a quantidade total de amostras para treinamento, M a quantidade de classificações do modelo, $y_{i,j}$ são os valores verdadeiros e $\hat{y}_{i,j}$ as predições pelo modelo (JAMES et al., 2021).

$$g(\hat{y}) = -\sum_{i=0}^{N}\sum_{j=0}^{M} y_{i,j}\log(\hat{y}_{i,j}). \tag{2.4}$$

2.2.3 Regularização

A regularização é aplicada tanto em *Machine Learning (ML)* quanto em *Deep Learning (DL)* a fim de diminuir o *overfitting* do modelo. Existem alguns métodos de regularização que são aplicadas em *DL*, entre eles estão a normalização por lotes, *dropout* e as regularizações L^1 e L^2 (GOODFELLOW et al., 2016).

A normalização por lotes introduz uma forma de reparametização na rede neural. Ela é utilizada principalmente em redes que possuem muitas camadas internas, para melhorar a perfomance do aprendizado. Com ela, as modificações nos pesos das camadas apresentadas pela minimização do custo causarão menos impacto não esperado no cálculo. Desta forma, é possível aplicar taxas de aprendizado maiores, agilizando o processo de treinamento (GOODFELLOW et al., 2016).

A normalização é descrita pela equação (2.5) e pode ser aplicada tanto nas saídas das camadas internas quanto nas entradas da rede, ambas descritas por H, onde que μ é a média e σ é o desvio padrão dos valores de ativação daquele lote (GOODFELLOW et al., 2016).

$$H' = \frac{H - \mu}{\sigma}. \tag{2.5}$$

Outra possibilidade de regularização que pode ser aplicada no treinamento é o uso de *dropouts*. Esse processo remove parte do conjunto de neurônios das camadas densas em um período de treinamento, realiza a classificação da entrada e atualiza os pesos dos neurônios mantidos. Para as rodadas seguintes do treinamento realiza o mesmo procedimento, escolhendo de forma aleatória os neurônios a serem retirados (GOODFELLOW et al., 2016).

Por fim, os métodos L^1 e L^2 aplicam uma penalidade na função de custo. O primeiro aplica um termo de penalidade que é igual a soma absoluta de um conjunto de pesos escolhido, enquanto no segundo o termo é calculado como a soma dos quadrados dos pesos. Desse modo, ao utilizar a penalidade L^1 o modelo força alguns parâmetros a se manterem zerados, condicionando a escolha de parâmetros da rede, o que não acontece com a penalidade L^2. Com essa característica, o primeiro consegue gerar modelos que são mais facilmente interpretáveis, enquanto o segundo, é mais indicado para modelos com grande número de variáveis preditoras (JAMES et al., 2021).

2.2.4 Aprendizado por transferência

Outro conceito que pode ser empregado durante o desenvolvimento de modelos de aprendizado é o aprendizado por transferência. Conforme Almeida (2020): "*transfer learning* pode ser definido como uma técnica em que um modelo treinado para resolver um problema específico é reutilizado para a solução de um novo problema relacionado ao primeiro".

Assim, ao invés de inicializar o modelo com pesos distribuídos de forma randômica, pode ser utilizado modelos já treinados e aperfeiçoados em algumas partes do modelo proposto. Por exemplo, utilizar nas primeiras camadas para a extração de características de imagens, mantendo os parâmetros de escolhas mais específicas com a necessidade de aprendizado (ALMEIDA, 2020).

Ainda segundo o autor Almeida (2020), esse método é muito importante considerando os gastos financeiros e de tempo para se realizar a obtenção do conjunto de dados, já que ele tende a diminuir o tempo de aprendizado das tarefas.

2.3 Trabalhos relacionados

Em específico para detecção de defeitos em imagens de inspeção por Partícula Magnética, podem ser citados os trabalhos de Kou et al. (2022), Tout et al. (2021), Konovalenko et al. (2020) e Yang et al. (2022). Neles é possível concluir que o modelo CNN é uma boa ferramenta para identificação dos defeitos no material, mesmo com imagens complexas. Outras referências para de classificação são os algortimos desenvolvidos por Prihatno et al. (2021) e Lee et al. (2019). Em especial o útlimo pode ser ressaltado por ter um conjunto de dados pequeno (300 imagens por categoria) e com alta acurácia de teste, ademais, o problema de classificação é parecido com o do presente projeto.

Tout et al. (2021) compara três arquiteturas de CNN para realizar a classificação dos locais de defeitos em metais. Ele apresenta dois métodos que retornam regiões de presença de falhas (um por classificação de imagens e outro por detecção de objetos) e um método que detecta precisamente onde a falha se encontra no ambiente (utilizando segmentação semântica). O método de classificação de imagens varre a imagem de entrada buscando se a região possui o defeito ou não; o de detecção de objetos busca as regiões que possuem falhas; já o segmentação semântica classifica *pixel* a *pixel* os locais na imagem, além disso, retorna a imagem com a classificação binária (há ou não defeito), utilizando uma técnica chamada *upsampling*. Kou et al. (2022) também apresenta um modelo de segmentação semântica como alternativa para métodos físicos de detecção de defeitos em trilhos com acurácia de teste chegando a 99%. Esse método de classificação se destaca pela quantidade baixa de imagens (algumas centenas) e alta acurácia de teste em ambos artigos.

3 METODOLOGIA

3.1 Descrição do conjunto de dados

3.1.1 Classificação dos defeitos

As imagens cadastradas podem ser subdivididas entre com e sem defeitos. Os defeitos podem ser subdivididos entre trincas e pites, estes podem ser encontrados tanto sozinhos quanto em conjunto (colônias de defeitos).

Segundo o Centro de Informação Metal Mecânica (CIMM), trinca é o defeito superficial que ocorre quando a tensão local de ruptura excede os limites do material e pite é uma depressão no material causado por corrosão.

Na Tabela 3.1 está descrita a sigla dada a cada tipo de anomalia, essa denominação é seguida pela empresa SIMEROS, proprietária do banco de dados. Na Figura 3.3 é possível verificar essa classificação com as imagens reais.

Tabela 3.1 – Descrição dos tipos de anomalias encontradas.

Sigla	Descrição
IP	Anomalia do tipo trinca
IPC	Anomalia do tipo colônia de trincas
IV	Anomalia do tipo pite
IVC	Anomalia do tipo colônia de pites

3.1.2 Banco de dados

O banco de dados de imagens e classificações é pertencente a empresa SIMEROS. A distribuição entre a classificação dos defeitos é de 64,70% como IPC, 15,65% como IP, 14,05% como IVC e 5,60% IV (verificar a Tabela 3.1 para descrição das siglas). Cada imagem com defeito possui um tipo de classificação associado.

As imagens disponíveis possuem tamanho de 3264 x 2448 *pixels* e são coloridas. Exemplos de imagens podem ser observados na Figura 3.1.

3.2 Algoritmo proposto

Observando as imagens cadastras (Figura 3.1), conclui-se que a classificação da imagem completa traria dificuldades para o modelo entender onde está contido o arame com o defeito, principalmente, pela falta de padronização da sua posição, interferência do fundo e grande variedade na iluminação das imagens.

Desta forma, em consonância com o trabalho desenvolvido por Tout et al. (2021), as iamgens originais foram recortadas em registros de 200 x 200 *pixels* e classificados em cinco categorias. Além das categorias de defeitos, inclui-se a categoria de fundo que engloba maior parte da imagem. Exemplos podem ser observados nas Figuras 3.2 e 3.3.

Visto a natureza das imagens, em especial a grande variedade de tipos de imagens de fundo, é escolhido trabalhar com dois modelos de classificação. Um com o resultado binário que diz se a região possui ou não anomalia e outro, se com anomalia, que retorna o tipo de defeito. Esses modelos serão chamados de classificação de região e de defeitos, respectivamente. Assim, o fluxograma de classificação se dá com

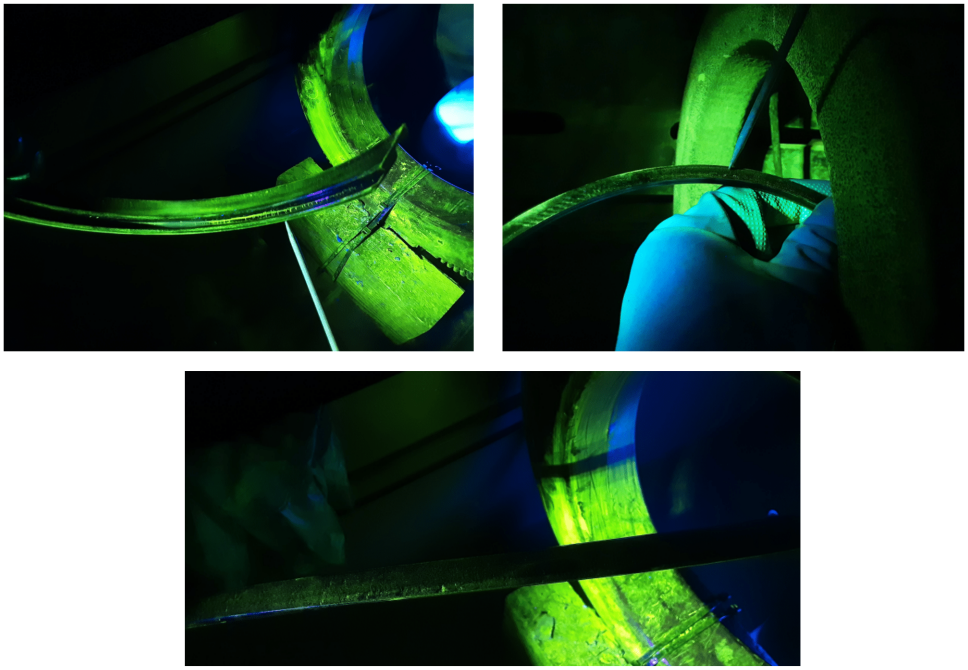


Figura 3.1 – Exemplos de imagens cadastradas no banco de dados.

uma varredura em toda a imagem com uma janela de tamanho fixo, onde primeiro é verificado se há anomalia e, após, se detectada, é classificada na sua respectiva classe.

Para verificar a hipótese de que o treinamento com as imagens completas não atenderia ao objetivo, também será testado um algoritmo utilizando esse tipo de entrada. Nesse caso, será aplicada apenas uma rede neural para fazer a classificação dos defeitos. As entradas, nesse caso, sofrerão uma redução da dimensionalidade para 512 x 512 *pixels*, a fim de conseguir treinar o modelo no sistema utilizado.

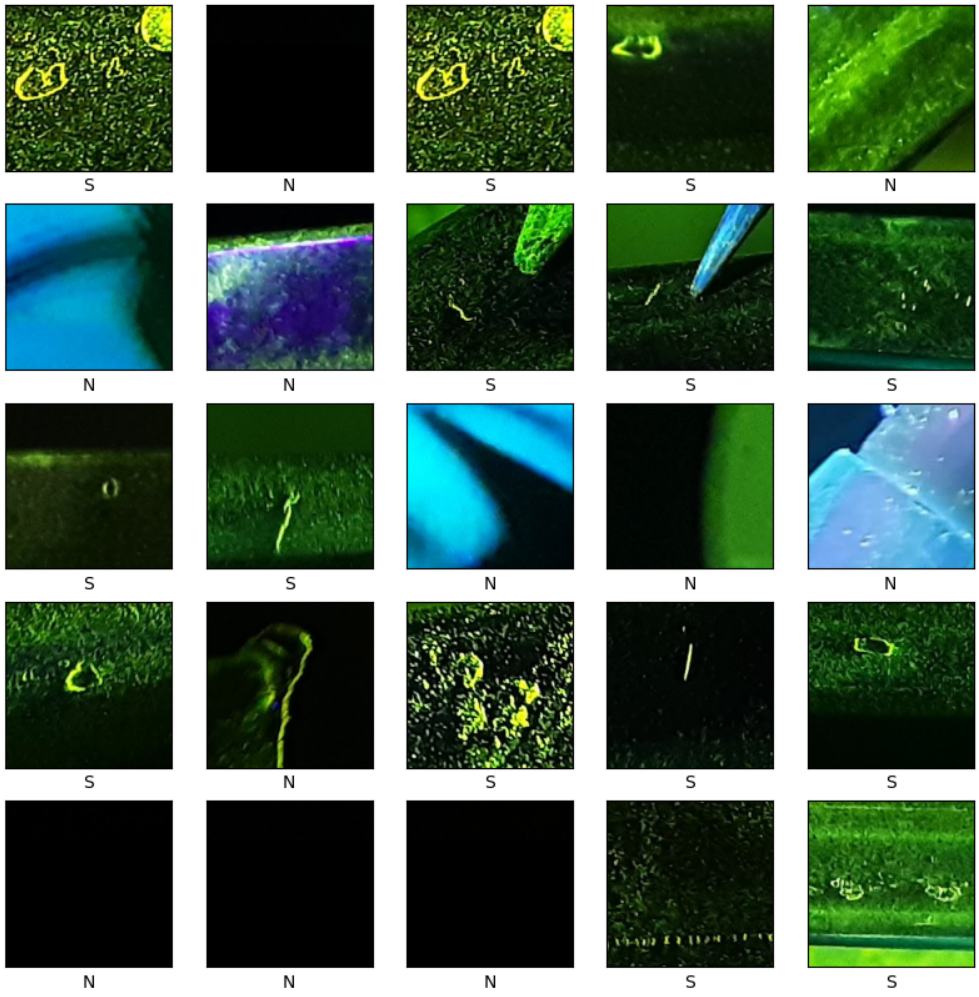


Figura 3.2 – Entradas do modelo de classificação de regiões, onde S representa a região com defeito e N a região sem defeito.

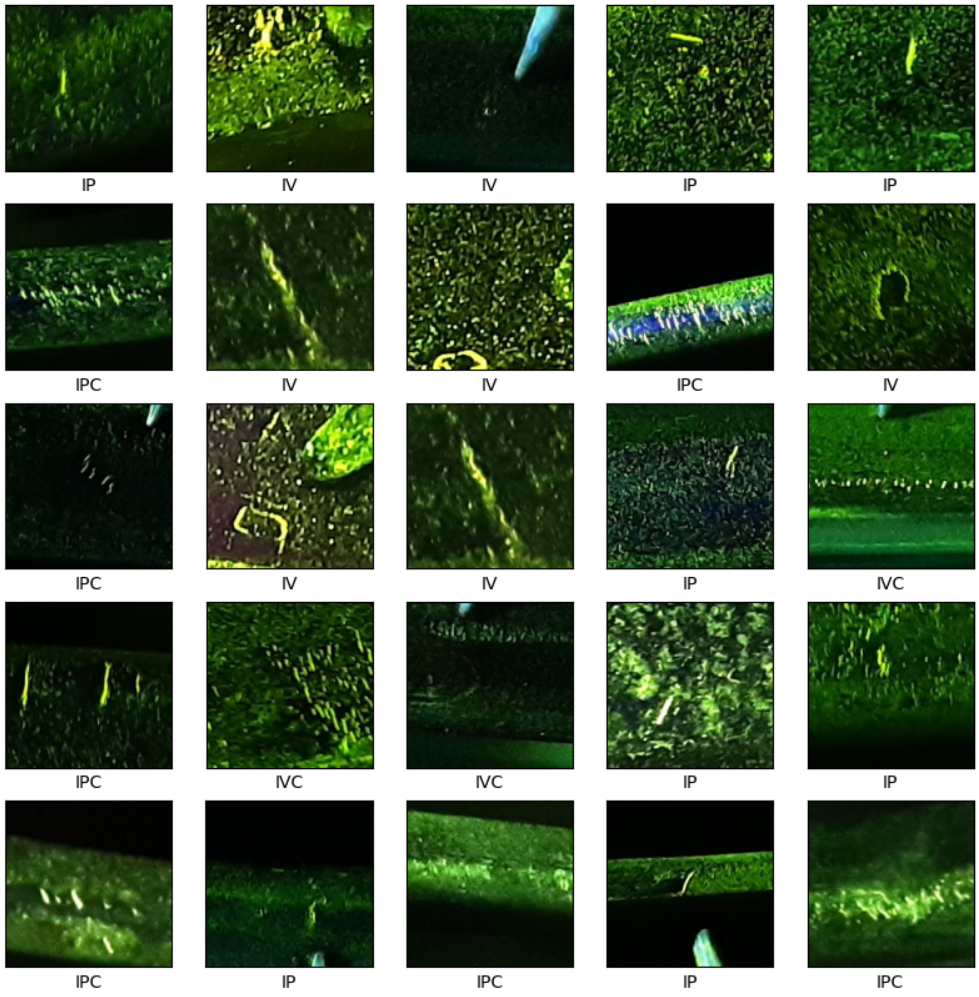


Figura 3.3 – Entradas do modelo de classificação de defeitos, com as siglas descritas na Tabela 3.1.

3.2.1 Arquitetura utilizada

Foi escolhido aplicar o aprendizado por transferência com a rede ResNet50, tal como Kou et al. (2022) e como Konovalenko et al. (2020). Tout et al. (2021) também utilizaram rede residual no seu modelo de classificação, o que resultou em melhores resultados se comparada a arquiteura sem o uso dos pesos treinados pela rede residual. Por utilizar os pesos já treinados a partir do conjunto de dados ImageNet, as entradas utilizadas se encontram no sistema RGB. A arquitetura de camadas adicional à rede residual está disponível na Tabela 3.2.

Para ambos os modelos de classificação foi utilizada a mesma estrutura de camadas, modificando apenas o número de canais da última camada densa, sendo 2 para a classificação de regiões e 4 para a classificação dos defeitos.

No caso do modelo de clasificação de defeitos foram realizados outros testes com redes com maior número de neurônios e camadas densas, porém, todas resultaram em valores parecidos de acurácia. Desse modo, optou-se por manter a estrutura mais concisa. Além de testar a mudança nas camadas da arquitetura proposta, foi verificada a possibilidade de aplicar a rede apresentada por Lee et al. (2019), os resultados estão disponíveis na seção 4.3 - Classificação dos Defeitos.

Tabela 3.2 – Arquitetura de camadas para os modelos

Tipo de camada	Canais	Regularização	Ativação
ResNet50	-	-	-
Densa	512	$L^2 = 0,1$	ReLU
Dropout	-	-	-
Densa	512	$L^2 = 0,1$	ReLU
Densa	2 ou 4	-	Softmax

3.2.2 Treinamento

Para o treinamendo foi utilizado o otimizador Adam com taxa de aprendizado de 0,00001; o custo foi calculado com o método *categorical cross-entropy*, ambos implementados pelo *framework Tensorflow*. O motivo da taxa de aprendizado baixa, mesmo com o uso de aprendizado por transferência, pode ser explicado pela baixa quantidade de entradas.

Para a classificação de regiões foram utilizadas 1088 imagens por categoria (com e sem defeito), enquanto para a classificação de defeitos foram utilizadas 200 imagens por tipo (verificar a Tabela 3.1). Ambos conjuntos de dados foram dividios em 80% para o treinamento, 10% para a validação e 10% para o teste.

A fim de aumentar o número de dados e evitar o *overfitting* do modelo, foi aplicado *data augmentation* no treinamento com o uso de rotação, giro horizontal e *zoom*. O número baixo de imagens, em especial para o treinamento do segundo tipo de clasificação, se dá pelo trabalho de subdividir cada uma das imagens, mesmo possuindo um método semi-automático.

3.3 Sistema utilizado

O computador utilizado para o treinamento do modelo possui sistema operacional Windows 10, processador Intel Core i7-8700 com memória RAM de 32GB.

O projeto é desenvolvido com a linguagem *Python* 3.7, com o *framework TensorFlow* versão 2.11 e *API Keras*.

3.4 Métricas de avaliação

Para avaliação dos algortimos e da escolha da arquitetura de camadas do modelo CNN serão utilizados a acurácia e a matriz de confusão.

A acurácia (ACC) é apresentada na equação (3.1). São definidas também a precisão (p) e o *recall* (r, ou sensibilidade), respectivamente (3.2) e (3.3). Onde TP são as valores verdadeiros positivos, TN são os valores verdadeiros negativos, FP são os falsos positivos e FN são os falsos negativos. No caso da classificação entre imagens com ou sem defeitos, essas variáveis representam as imagens com defeitos, as imagens sem defeitos, as imagens sem defeitos, mas que foram detectadas com defeitos e as imagens com defeitos em que o modelo não detectou, respectivamente.

Com os valores de TP, TN, FP e FN é possível montar a matriz de confusão, tabela que informa o número de previsões corretas e incorretas do modelo. Os resultados previstos estão dispostos nas colunas e os resultados reais as linhas, com isso, a diagonal principal possui os corretos de previsão (BRUCE; BRUCE, 2019).

$$ACC = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.1}$$

$$p = \frac{TP}{TP + FN} \tag{3.2}$$

$$r = \frac{TP}{TP + FP} \tag{3.3}$$

Além disso, será verificada as regiões que são utilizadas para a classificação através do metódo desenvolvido por Ribeiro et al. (2016) chamado *Local Interpretable Model-Agnostic Explanation (LIME)*, aplicando a biblioteca desenvolvida pelos autores em conjunto com o modelo preditor. Essa ferramenta permite visualizar as regiões mais utilizadas, as quais são coloridas de verde; enquanto as menos utilizadas são coloridas de vermelho. Outra possibilidade permitida pela ferramenta é só manter na imagem original as regiões utilizadas para classificação. Exemplos aplicados são apresentados ao longo da seção 4 - Resultados e Discussão.

4 RESULTADOS E DISCUSSÃO

Nos modelos propostos, tanto para o modelo de classificação das regiões quanto para a identificação de defeitos foi utilizada a rede neural de convolução ResNet50 já treinada com as imagens do banco de dados chamado ImagesNet, acrescentada das camadas apresentadas na Tabela 3.2.

Ao longo dos testes realizados com as arquiteturas propostas, foram testados diferentes tipos de valores para a taxa de aprendizado e para os regularizadores. A taxa de aprendizado padrão definida pela biblioteca utilizada foi diminuída, pois percebeu-se que não havia uma melhora nas métricas ao longo do treinamento. De maneira similar, o valor de penalidade padrão do método L^2 foi aumentado, após se realizar com testes de valores entre 0,5 e 0,01 (valor definido pelo *Tensorflow*). O uso do regularizador L^1 não melhorou a performance ao ser utilizado em conjunto com o L^2 e, ao ser testado sozinho, não obteve resultados melhores do que as redes treinadas somente com o método L^2. Por fim, a taxa utilizada na camada de *dropout* foi definida como 0,2, a apartir de testes utilizando um intervalo de 0,1 a 0,5.

4.1 Classificação das Imagens

A classificação da imagem inteira em quatro defeitos também utilizou a arquitetura já citada: uso de ResNet50 com a adição das camadas densas disponíveis na Tabela 3.2.

O conjunto de dados de treinamento contém 500 imagens por categoria descritas na Tabela 3.1. Foram utilizadas 35 épocas. As imagens de entrada foram reduzidas a 512 x 512 *pixels* e também foi aplicado *data augmentation*. O conjunto de dados foi separado em 80% para treinamento, 10% para validação e 10% para teste. Com o uso de imagens maiores, o treinamento do modelo demorou em torno de oito horas.

Mesmo com a acurácia de teste chegando a 73,00%, ao verificar com a ferramenta desenvolvida por Ribeiro et al. (2016) as regiões utilizadas pela rede, observa-se que não são capturados os locais de interesse na classificação - que utiliza majoritariamente o fundo para realizar a escolha. Na Figura 4.1 pode ser observado as regiões que o modelo usa para a selação: na primeira coluna somente as regiões mais importantes são mantidas, na segunda coluna as que são marcadas em vermelho são as menos utilizadas e as em verde as mais utilizadas, já na terceia coluna está a imagem original. Esse resultado corrobora que o uso de imagens menores e uma varredura ao longo da imagem completa, utilizando uma janela fixa e classificação por parte analisada, pode trazer resultados mais confiáveis.

Além de não conseguir verificar corretamente os arames e seus defeitos, o treinamento dessa rede se torna mais custoso por utilizar imagens de maior tamanho.

4.2 Classificação das Regiões

O teste do modelo de classificação de regiões chegou em uma acurácia de teste de 92,13%, sendo utilizada 108 imagens para cada categoria no teste. Durante o processo de treinamento, a média de acurácia de treinamento e validação ficaram em 95,18% e 87,51%, respectivamente. O treinamento dessas redes utilizando o sistema apresentado na seção 3.3 demorou em torno de quatro horas.

Na matriz de confusão apresentada na Tabela 4.1 é possível identificar que os falsos positivos causados pela classificação errada do tipo sem defeito são os erros mais frequentes do modelo; os valores previstos estão dispostos nas colunas e os

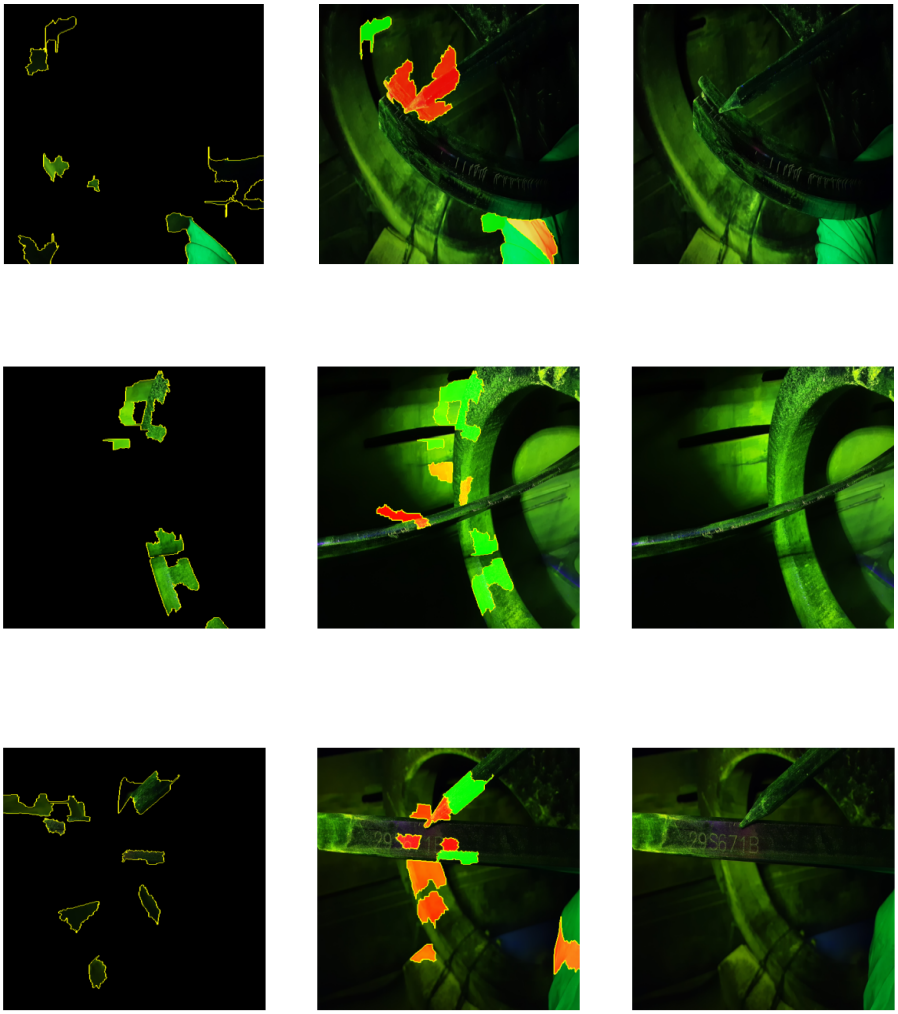


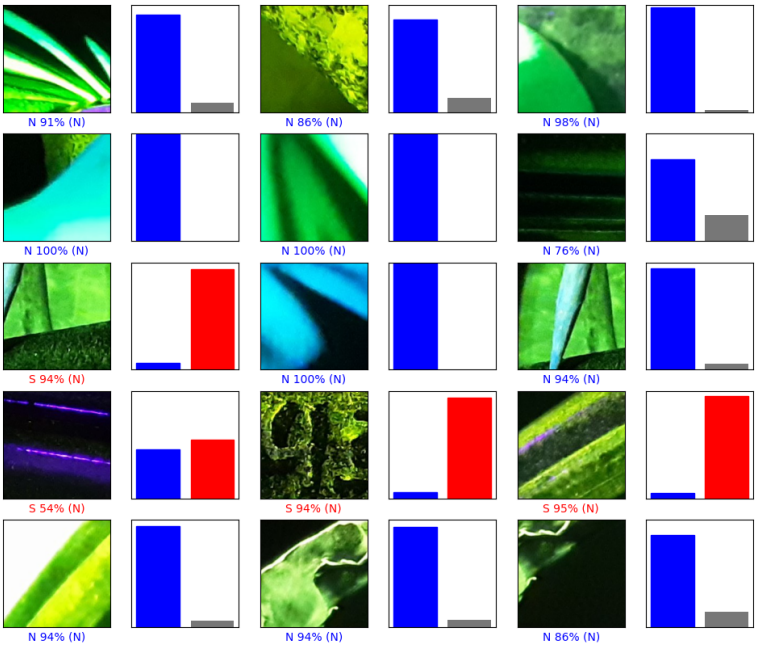
Figura 4.1 – Regiões utilizadas para a classificação, segundo a biblioteca desenvolvida por Ribeiro et al. (2016).

valores reais nas linhas. Isso é evidenciado na Figura 4.2a, onde pode-se perceber que o modelo tem dificuldade de realizar a escolha certa em imagens que possuem arames ou suas bordas presentes. Uma hipótese que pode ser apontada é a baixa quantidade desses tipos de imagens dentro do conjunto de dados de treinamento e validação.

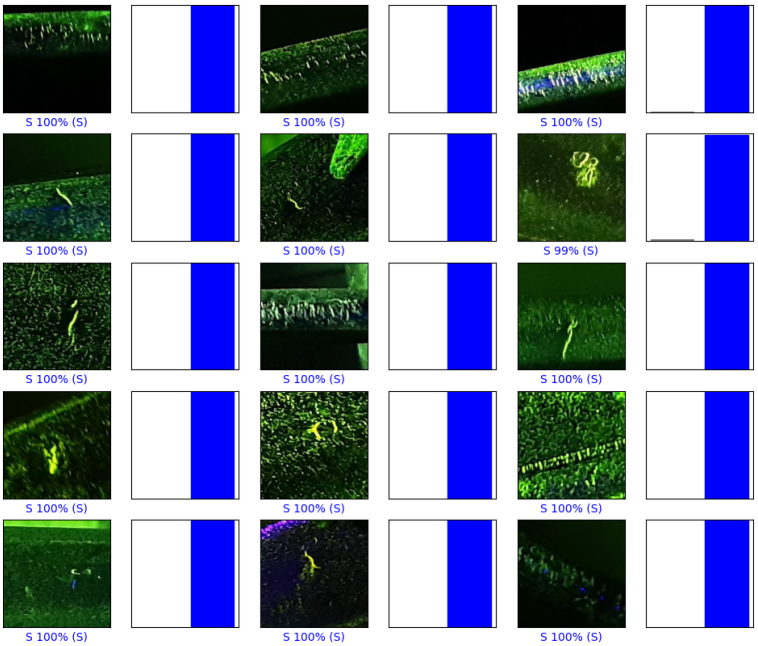
Tabela 4.1 – Matriz de confusão do modelo de classificação de regiões

	Sem defeito	Com defeito
Sem defeito	91	17
Com defeito	0	108

Na Figura 4.2 são apresentadas algumas imagens utilizadas para o teste do modelo, primeiro são mostradas as imagens utilizadas e ao seu lado o gráfico de probabilidade do resultado da rede. Abaixo da imagem está descrito o tipo que retornou o resultado com maior probabilidade, a probabilidade associada com a escolha e, entre parênteses, a classificação correta.



(a) Regiões sem defeito.



(b) Regiões com defeito.

Figura 4.2 – Exemplos de resultados do teste de classificação de escolhas de tipos de região, onde S denomina a região com defeito e N a região sem defeito.

4.3 Classificação dos Defeitos

O teste do modelo de classificação de defeitos chegou em uma acurácia de 66,25%, sendo utilizada 20 imagens para cada categoria no teste. Durante o processo de treinamento, a acurácia de treinamento e validação ficaram em 92,71% e 76,25%, respectivamente. O histórico do treinamento pode ser verificado no gráfico presente na Figura 4.3, foi utilizado total de 100 épocas para treinamento, levando cerca de duas horas para treinamento. Observou-se também que, mesmo treinando a rede um maior número de vezes, a acurácia de validação e teste não aumentaram proporcionalmente, evidenciando a falta de dados como problema para melhoria.

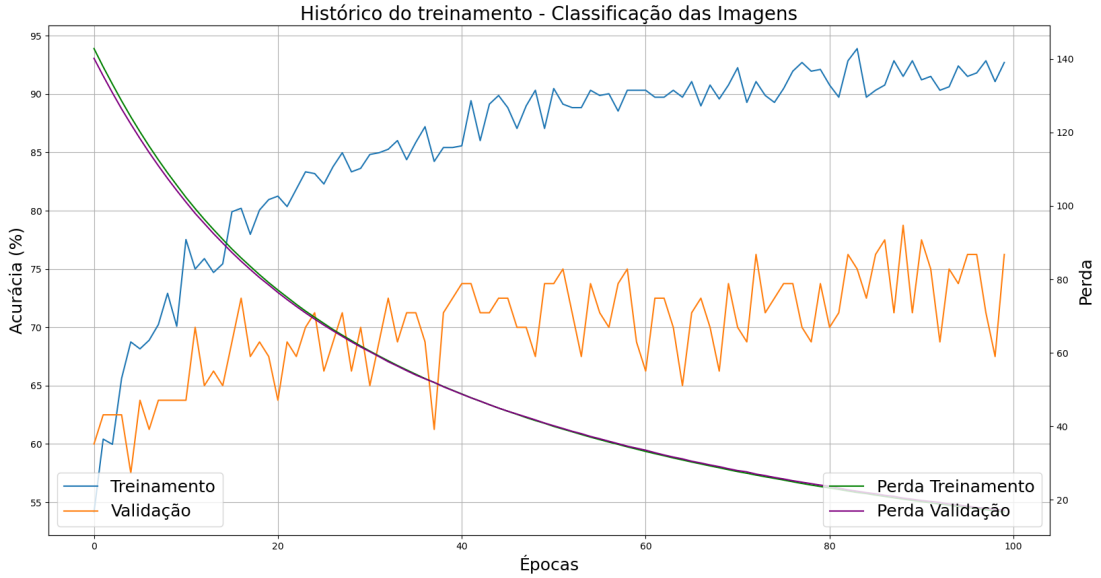


Figura 4.3 – Gráfico de acurácia e perdas de treinamento e validação ao longo do processo de treinamento do modelo de classificação defeitos.

A matriz de confusão é apresentada na Tabela 4.2, com ela percebe-se que os valores falsos indicados são minoria, possuindo a diagonal principal com os maiores valores; os valores previstos estão dispostos nas colunas e os valores reais nas linhas. Além disso, pode ser verificado que, ao unir os grupos conforme o tipo de defeito e anulando a divisão por quantidade de defeitos presente (IV e IVC, IP e IPC como uma classificação de apenas dois grupos, não quatro), a acurácia de teste se eleva para 78,75%. Nota-se que a classificação com menor acertos é a tipo IVC onde se tem o maior número de escolhas erradas do que corretas.

Tabela 4.2 – Matriz de confusão do modelo de classificação de regiões

	IP	IPC	IV	IVC
IP	14	1	3	2
IPC	2	17	1	0
IV	4	0	13	3
IVC	6	1	4	9

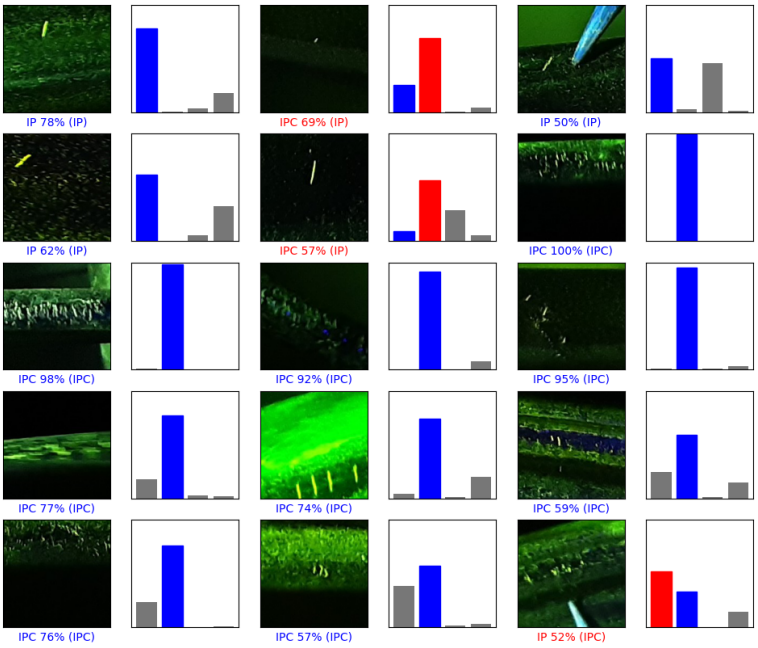
Exemplos dos resultados de teste do modelo estão apresentados na Figura 4.4. Ao visualizar as imagens que compõe o conjunto de dados pode ser notada a

dificuldade de identificar a partir das fotos a diferença entre os grupos. Na figura também podem ser verificados os resultados incorretos com a escolha de defeito correta (pite ou trinca) mas a identificação da quantidade de defeitos errada.

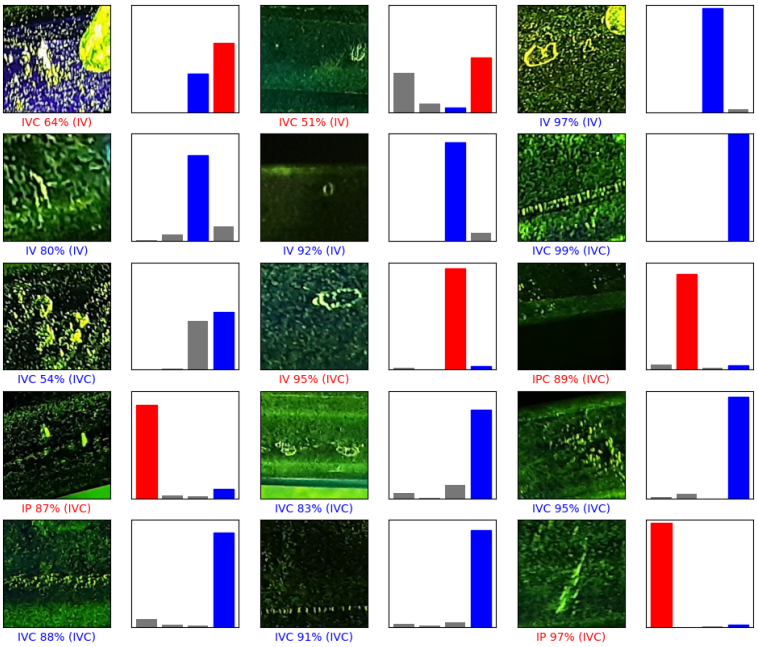
Além disso, com o uso do método *Local Interpretable Model-Agnostic Explanation (LIME)* (RIBEIRO et al., 2016), conclui-se que a rede neural está utilizando as regiões esperadas para a classificação, conforme apresentado na Figura 4.5.

A arquitetura proposta por Lee et al. (2019) foi treinada com o conjunto de dados de defeitos. Foi escolhido testar essa possibilidade, pois os autores utilizaram 300 imagens por classe (número de dados parecido com o utilizado no trabalho) e o objetivo da rede apresentada consistia em subdividir as imagens entre seis tipos de defeito em metal.

Utilizando 200 épocas para treinamento e mesma divisão do conjunto de dados, obteu-se 42,50% de acurácia para validação, 94,49% para treinamento e 47,50% para teste. As perdas ficam abaixo do modelo com a ResNet50 para extração das características, enquanto a distribuição de dados na matriz de confusão ficou parecida com o modelo anterior. Notou-se que a acurácia de treinamento ao longo do treinamento possui evolução lenta, enquanto a de validação e teste não apresentou melhora, se mantendo constante. Mesmo utilizando mais épocas para treinamento e, em consequência, demorando mais o processo, a acurácia de teste não alcançou os resultados apresentados pelos autores (99,44%) nem o apresentado anteriormente com o uso de aprendizado por tranferência.

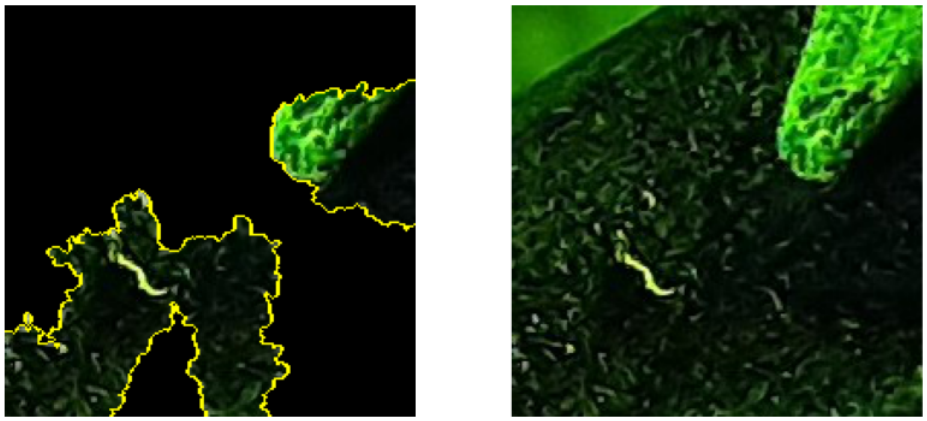


(a) Defeitos tipo IP e IPC.

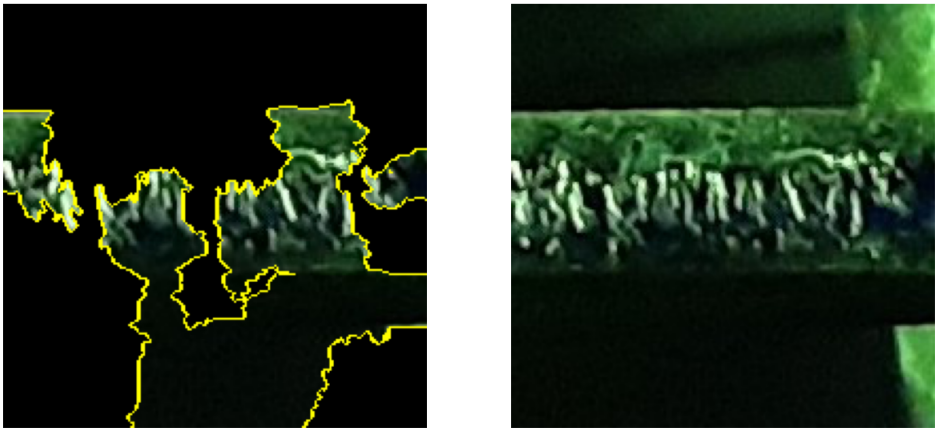


(b) Defeitos tipo IV e IVC.

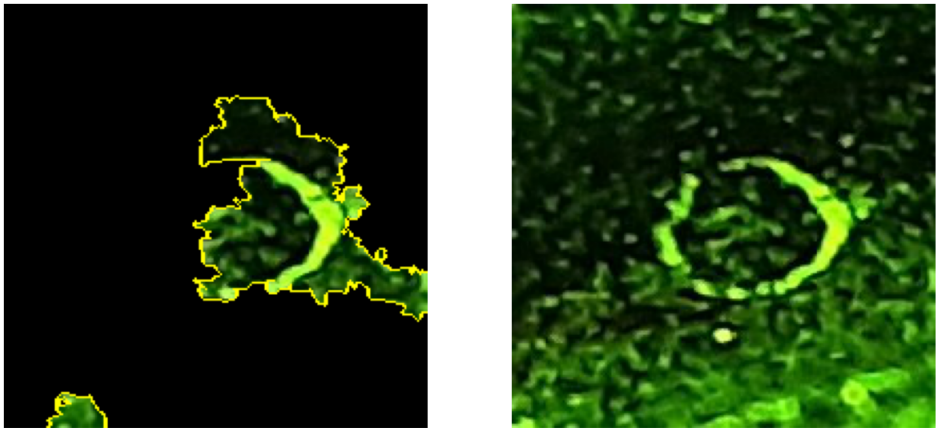
Figura 4.4 – Exemplos de resultados do teste de classificação de escolhas de defeitos.



(a) Exemplo de defeito IP.



(b) Exemplo de defeito IPC.



(c) Exemplo de defeito IV.

Figura 4.5 – Regiões utilizadas para a classificação (imagem da esquerda) e o registro original (imagem da direita), segundo a biblioteca desenvolvida por Ribeiro et al. (2016).

5 CONCLUSÃO

O objetivo deste projeto era o desenvolvimento de uma ferramenta capaz de classificar diferentes tipos de defeitos presentes em arames metálicos analisados com inspeção por partícula magnética. Esses defeitos podem ser subdivididos em quatro grupos: pite (IV), pite em colônia (IVC), trinca (IP) e trinca em colônia (IPC). Com o uso de redes neurais de convolução, aplicado com o *framework Tensorflow*, foi possível propor uma solução viável, dado o conjunto de dados disponível.

O algoritmo proposto é subdividido em duas partes: primeiro a verificação do tipo de região, com ou sem defeito, com a posterior classificação do tipo de defeito. Foi utilizado um conjunto de dados contendo 1088 imagens para cada classe para o treinamento, validação e teste do primeiro modelo e 200 para o segundo. A partir disso, consegue-se distinguir com 92,13% de acurácia de teste se uma região contém ou não um defeito e com 66,25% de acurácia de teste classifica-se os defeitos em quatro tipos e com 78,75% nos grupos defeito tipo pite e trinca (IVC e IV, IPC e IP, respectivamente).

Observou-se que a divisão do algoritmo em duas classificações, utilizando uma janela fixa e de menor tamanho na imagem original, consegue atingir resultados esperados nas imagens provenientes do banco de dados da empresa SIMEROS. A classificação das imagens inteiras resultou no uso regiões de fundo para a decisão do modelo, como pode ser observado com ferramenta criada por Ribeiro et al. (2016). Se observadas as métricas do treinamento e teste, poderia ser concluido que o desenvolvimento melhor das camadas e parâmetros resultaria em um modelo com alta capacidade de classificação. Porém, com a baixa padronização das imagens e como mostrado pelo método *Local Interpretable Model-Agnostic Explanation (LIME)*, não seria possível confiar nesses resultados. Ademais, ao reduzir o tamanho da imagem a ser classificada, não é perdida a resolução original, tal qual aconteceria ao utilizar técnicas de redução de dimensionalidade para o treinamento da rede neural com as fotos completas, além disso, pode ser treinado o modelo com mais dados ou com um número de épocas maior utilizando o mesmo tempo de processamento.

Pelos resultados apresentados, nota-se que a baixa quantidade de imagens é um grande fator para a melhoria das métricas do modelo. Em específico, percebe-se que diversificar mais o conjunto para o treinamento das regiões, especialmente a quantidade de fotos que contém arames, seus defeitos naturais e suas bordas, seria de fundamental importância. De mesmo modo, traria resultados mais precisos melhorar a qualidade das imagens de defeitos, focalizando mais no defeito do que no seu ambiente e aumentando o número de dados associado para o treinamento, teste e validação.

Alguns trabalhos recentes que buscam a classificação ou detecção de defeitos podem ser utilizados para comparação. Tout et al. (2021), no seu algoritmo de classificação das regiões (com ou sem defeitos) utilizando ResNet34 no aprendizado por transferência, conseguiu acurácia de 99,93%, porém utilizou o conjunto de dados com 62740 imagens - cada classe com metade do conjunto. De mesma forma, os autores Konovalenko et al. (2020) ao aplicarem ResNet50 para classificação de três tipos de anomalias em metais, mesmo em um conjunto de dados distribuído de forma não homogênea, atingiram acurácia de teste média de 96,91%, com um conjunto com 17723 imagens com defeitos (820, 14576, 2327, respectivamente para cada classe). Esses resultados destacam o uso de maiores quantidades de imagens como ponto para melhora da acurácia do modelo, principalmente no caso da classificação de defeitos.

Além disso, ao aplicar a arquitetura proposta por Lee et al. (2019), não alcançou-se os resultados apresentados pelos autores, mesmo utilizando a quantidade de

dados parecida com a descrita. Para a sua rede proposta era necessário classificar as imagens em seis categorias, possuindo uma base de dados com 300 imagens por classe possuindo 200 x 200 *pixels*. Após o treinamento, a acurácia de teste para o conjunto de imagens do presente trabalho foi de 47,50% comparada com 99,44% do artigo.

Outras possibilidades de melhoria modelo proposto, ademais o conjunto de dados, são o uso de diferentes redes neurais de convolução já treinadas no aprendizado por transferência, a modificação dos parâmetros de regularização utilizados e a adição ou remoção de camadas. Também pode ser estudado uma forma de tratamento das imagens com defeitos de maneira a realçar as suas características, como realizado por Kou et al. (2022). Por fim, como já apontado pelos trabalhos dos autores Tout et al. (2021) e Kou et al. (2022), a utilização da técnica de segmentação semântica para a classificação de defeitos com baixa quantidade de dados também é uma alternativa que consegue obter resultados muito promissores (mais de 95% de acurácia de teste) mesmo em imagens com múltiplas bordas.

REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, J. R. de. *Transfer learning e convolutional neural networks para a classificação de imagens e reconhecimento de objetos no âmbito da perícia criminal*. Dissertação (Mestrado) — Universidade de Brasília, Instituto de Ciências Exatas, Departamento de Ciência da Computação, Brasília, 2020. 6, 8, 9

ANDREUCCI, R. *Ensaios por Partículas Magnéticas*. [S.l.], 2007. 2

ASTM INTERNATIONAL. *ASTME 709 08 - Standard Guide for Magnetic Particle Testing*. Estados Unidos, 2021. 2

BAI, Q.; BAI, Y. 24 - flexible pipe. In: BAI, Q.; BAI, Y. (Ed.). *Subsea Pipeline Design, Analysis, and Installation*. Boston: Gulf Professional Publishing, 2014. p. 559–578. ISBN 978-0-12-386888-6. Disponível em: <https://www.sciencedirect.com/science/article/pii/B9780123868886000249>. 1

BAI, Y.; BAI, Q. Chapter 5 - integrity management of flexible pipes. In: BAI, Y.; BAI, Q. (Ed.). *Subsea Pipeline Integrity and Risk Management*. Boston: Gulf Professional Publishing, 2014. p. 101–124. ISBN 978-0-12-394432-0. Disponível em: <https://www.sciencedirect.com/science/article/pii/B9780123944320000056>. 1, 2

BEZDAN T., B. D. N. Convolutional neural network layers and architectures. In: *Sinteza 2019 - International Scientific Conference on Information Technology and Data Related Research*. [S.l.: s.n.], 2019. p. 445–451. 4, 5

BRUCE, P.; BRUCE, A. *Estatística Prática para Cientistas de Dados*. [S.l.]: O'Reily, 2019. 15

CUNHA, B. S. da. *Development Of Computer Vision Based Models For Automated Crack Detection*. Dissertação (Mestrado) — UNIVERSIDADE FEDERAL DE PERNAMBUCO, Recife, 2020. 2

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. 4, 5, 6, 7, 8

INDOLIA, S. et al. Conceptual understanding of convolutional neural network- a deep learning approach. *International Conference on Computational Intelligence and Data Scienc*, 2018. 2, 4, 5, 6

JAMES, G. et al. *An Introduction to Statistical Learning*. New York: Springer, 2021. 4, 5, 6, 8

KONOVALENKO, I.; MARUSCHAK, P.; BREZINOVá, J.; VIňáš, J.; BREZINA, J. Steel surface defect classification using deep residual neural network. *Metals*, v. 10, n. 6, 2020. ISSN 2075-4701. Disponível em: <https://www.mdpi.com/2075-4701/10/6/846>. 9, 14, 23

KOU, L.; SYSYN, M.; FISCHER, S.; LIU, J.; NABOCHENKO, O. Optical rail surface crack detection method based on semantic segmentation replacement for magnetic particle inspection. *Sensors*, v. 22, n. 21, 2022. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/22/21/8214>. 3, 9, 14, 24

LEE, S. Y.; TAMA, B. A.; MOON, S. J.; LEE, S. Steel surface defect diagnostics using deep convolutional neural network and class activation map. *Applied Sciences*, December 2019. 3, 9, 14, 20, 23

PRIHATNO, A. T. et al. Metal defect classification using deep learning. *Conference: 2021 Twelfth International Conference on Ubiquitous and Future Networks (ICUFN)*, August 2021. 9

RIBEIRO, M. T.; SINGH, S.; GUESTRIN, C. "Why should I trust you?": Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. [S.l.: s.n.], 2016. p. 1135–1144. 15, 16, 17, 20, 22, 23

SOUZA, L.; FILHO, S.; CARPIGIANI, M.; FREITAS, J. Flexible Pipe Integrity Management. In: . [s.n.], 2003. (International Conference on Offshore Mechanics and Arctic Engineering, Volume 2: Safety and Reliability; Pipeline Technology), p. 711–719. Disponível em: <https://doi.org/10.1115/OMAE2003-37337>. 1

TOUT, K. et al. Automated vision system for magnetic particle inspection of crankshafts using convolutional neural networks. *The International Journal of Advanced Manufacturing Technology*, 2021. 2, 3, 9, 10, 14, 23, 24

YANG, Y. et al. Automatic defect identification method for magnetic particle inspection of bearing rings based on visual characteristics and high-level features. *Applied Sciences*, 2022. 2, 3, 9