

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
ENG. DE CONTROLE E AUTOMAÇÃO

MATEUS SCHOENARDIE HANEL - 262536

**ANÁLISE DE MÉTODOS DE DATA
AUGMENTATION PARA MELHORIA
DO DESEMPENHO DE UMA REDE
NEURAL DE DETECÇÃO DE DEFEITOS
EM SUPERFÍCIES METÁLICAS**

Porto Alegre
2021

MATEUS SCHOENARDIE HANEL - 262536

**ANÁLISE DE MÉTODOS DE DATA
AUGMENTATION PARA MELHORIA
DO DESEMPENHO DE UMA REDE
NEURAL DE DETECÇÃO DE DEFEITOS
EM SUPERFÍCIES METÁLICAS**

Trabalho de Conclusão de Curso (TCC-CCA) apresentado à COMGRAD-CCA da Universidade Federal do Rio Grande do Sul como parte dos requisitos para a obtenção do título de *Bacharel em Eng. de Controle e Automação*.

ORIENTADOR:

Prof. Dr. Heraldo José de Amorim

Porto Alegre
2021

MATEUS SCHOENARDIE HANEL - 262536

**ANÁLISE DE MÉTODOS DE DATA
AUGMENTATION PARA MELHORIA
DO DESEMPENHO DE UMA REDE
NEURAL DE DETECÇÃO DE DEFEITOS
EM SUPERFÍCIES METÁLICAS**

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção dos créditos da Disciplina de TCC do curso *Eng. de Controle e Automação* e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____
Prof. Dr. Heraldo José de Amorim, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil)

Banca Examinadora:

Prof. Dr. Heraldo José de Amorim, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil)

Prof. Dr. Altamiro Amadeu Susin, UFRGS
Doutor pela Institut National Polytechnique de Grenoble – Grenoble, França)

Prof. Dr. Herbert Martins Gomes, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil)

Mário Roland Sobczyk Sobrinho
Coordenador de Curso
Eng. de Controle e Automação

Porto Alegre, novembro de 2021.

AGRADECIMENTOS

Agradeço aos meus pais, Angela e Volnei, por sempre terem me apoiado e incentivado da melhor maneira possível, à minha vó Ilda, que sempre esteve comigo quando eu precisei, ao resto da minha família, que em diversos momentos me apoiou, aos meus colegas e amigos, que me ajudaram de tantas formas durante todo esse período de graduação, à minha namorada, Brandy, que me ajudou muito nesse último ano, e ao meu orientador, Heraldo, por ter sido um ótimo professor e auxiliado na elaboração deste trabalho.

RESUMO

Métodos de visão computacional e aprendizado de máquina estão sendo cada vez mais utilizados em ambientes industriais. Sistemas de inspeção visual automáticos permitem um monitoramento em tempo real e não destrutivo da produção, enquanto o uso de redes neurais permite a elaboração de soluções genéricas e adaptativas, sem a necessidade de programar e calibrar manualmente cada cenário de inspeção. Porém, um problema comum em soluções de aprendizado de máquina é a dificuldade de se obter um conjunto de dados suficientemente grande para o treinamento adequado de uma rede neural. Uma solução para esse problema é a criação de dados sintéticos a partir das imagens disponíveis. Neste trabalho, serão propostos dois métodos de *data augmentation* para melhorar as métricas de uma rede mista de detecção de defeitos em superfícies metálicas. Os métodos serão avaliados utilizando dois bancos de dados, e serão realizados testes com diversas proporções entre dados completamente classificados, em que se tem a informação píxel a píxel da localização dos defeitos, e fracamente classificados, em que se tem apenas a informação da presença ou ausência de defeitos a nível de imagem.

Palavras-chave: Processamento de Imagens, Redes Neurais, Geração de Dados Artificiais.

ABSTRACT

Computer vision and machine learning based methods are being increasingly used in an industrial context. Automated visual inspection systems allow real time and non-destructive production monitoring, while neural networks allow the development of generic and adaptive solutions, without the need to program and calibrate each inspection scenario. However, machine learning based solutions usually need a big training dataset, and in many applications it is hard to obtain a big enough dataset for an adequate training. A common solution for this problem is to augment the dataset with the creation of synthetic data. The current study proposes two data augmentation methods to improve the metrics of a mixed supervision surface-defect detection network. Both methods will be evaluated using two different datasets and different proportions of weakly and strongly labelled data for each dataset. With strongly labelled data, the image contains pixel-wise information of the defects location, and with weakly labeled data there is no segmentation mask, only the information that there is a defect in the image.

Keywords: Image Processing, Neural Networks, Data Augmentation.

SUMÁRIO

LISTA DE ILUSTRAÇÕES	8
LISTA DE TABELAS	9
LISTA DE ABREVIATURAS	11
1 INTRODUÇÃO	12
2 REVISÃO DA LITERATURA	13
2.1 Processamento de Imagens	13
2.1.1 Transformações Afins	14
2.1.2 Mesclagem de Imagens	14
2.2 Redes Neurais	17
2.2.1 Redes Neurais Convolucionais	18
2.3 Data Augmentation	19
3 METODOLOGIA	21
3.1 Rede Neural Escolhida	21
3.2 Métodos de <i>Data Augmentation</i>	22
3.2.1 Data Augmentation por Espelhamento	23
3.2.2 Data Augmentation por Mesclagem de Defeitos	23
3.3 Bancos de Dados Utilizados	26
3.4 Validação Experimental	28
4 RESULTADOS E DISCUSSÕES	30
4.1 Análise de Variância (ANOVA)	30
4.2 Análise dos Resultados	36
5 CONCLUSÃO	41
APÊNDICE A - PSEUDOCÓDIGO: DATA AUGMENTATION POR ES- PELHAMENTO	42
APÊNDICE B - PSEUDOCÓDIGO: DATA AUGMENTATION POR MES- CLAGEM DE DEFEITOS	43
APÊNDICE C - MÉTRICAS DOS TREINAMENTOS - SEVERSTAL	44
APÊNDICE D - MÉTRICAS DOS TREINAMENTOS - KSDD2	46
APÊNDICE E - ANOVA - SEVERSTAL	48

APÊNDICE F - ANOVA - KSDD2	51
REFERÊNCIAS	54

LISTA DE ILUSTRAÇÕES

1	Operação de mesclagem de imagens	15
2	Visualização das notações utilizadas	16
3	Diagrama de uma rede neural básica	17
4	Diagrama de uma rede neural convolucional básica	19
5	Detalhamento das camadas de convolução e de pooling	20
6	Exemplos de <i>data augmentation</i>	20
7	Representação da arquitetura da rede	21
8	Transformações afins utilizadas para o <i>data augmentation</i> independente de máscara	24
9	Transformações aplicadas nos defeitos: (A) original (B) piecewiseAffine;(C) rotate;(D) elastic;(E) scale;(F) fliplr; (G) flipud;	24
10	Exemplos de <i>data augmentation</i> por mesclagem de imagens. Defeitos originais (acima) e imagens sintéticas (abaixo)	25
11	Estrutura das pastas dos bancos de dados	27
12	Exemplo de defeito (esquerda) e máscara (direita) do banco de dados KSDD2	28
13	Exemplo de defeito (acima) e máscara (abaixo) do banco de dados Severstal	28
14	Gráficos de comparação para o banco de dados Severstal	37
15	Gráficos de comparação para o banco de dados KSDD2	39

LISTA DE TABELAS

1	Exemplos de transformações afins	14
2	Tranformações de espelhamento	23
3	Média das métricas - Severstal	31
4	Média das métricas - KSDD2	31
5	Quadro auxiliar PM - Severstal	32
6	Quadro ANOVA para a PM - Severstal	32
7	Desdobramento da interação do data augmentation para a PM - Severstal	33
8	Diferença das médias da PM - Severstal	34
9	Diferença das médias - Severstal	34
10	Quadro ANOVA da PM - KSDD2	34
11	Desdobramento da interação do data augmentation da PM - KSDD2 .	35
12	Diferença das médias da PM - KSDD2	35
13	Diferença das médias - KSDD2	36
14	Contagem de melhores métricas para cada tipo de treinamento - Severstal	38
15	Contagem de melhores métricas para cada tipo de treinamento - KSDD2	38
16	Métricas do primeiro lote de treinamento - Severstal	44
17	Métricas do segundo lote de treinamento - Severstal	45
18	Métricas do primeiro lote de treinamento - KSDD2	46
19	Métricas do segundo lote de treinamento - KSDD2	47
20	Quadro ANOVA da AUC - Severstal	48
21	Desdobramento da interação do data augmentation da AUC - Severstal	48
22	Quadro ANOVA da Exatidão - Severstal	48
23	Desdobramento da interação do data augmentation da Exatidão - Se- verstal	49
24	Quadro ANOVA da Precisão - Severstal	49
25	Desdobramento da interação do data augmentation da Precisão - Severstal	49
26	Quadro ANOVA da Revocação - Severstal	49
27	Desdobramento da interação do data augmentation da Revocação - Severstal	50
28	Quadro ANOVA da AUC - KSDD2	51
29	Desdobramento da interação do data augmentation da AUC - KSDD2	51
30	Quadro ANOVA da Exatidão - KSDD2	51
31	Desdobramento da interação do data augmentation da Exatidão - KSDD2	52
32	Quadro ANOVA da Precisão - KSDD2	52
33	Desdobramento da interação do data augmentation da Precisão - KSDD2	52
34	Quadro ANOVA da Revocação - KSDD2	52

35	Desdobramento da interação do data augmentation da Revocação - KSDD2	53
----	---	----

LISTA DE ABREVIATURAS

RGB	<i>Red, Green and Blue</i>
RNC	Rede Neural Convolucional
KSDD2	Kolektor Surface-Defect Dataset 2
GAN	<i>Generative Adversarial Network</i>
AUC	<i>Area under the ROC Curve</i>
PM	Precisão Média
GAN	<i>Generative Adversarial Network</i>

1 INTRODUÇÃO

A qualidade dos produtos em ambientes industriais pode ser afetada por diversos fatores, como a qualidade do processo, condições de trabalho e limitações tecnológicas. Com isso, o uso de métodos de detecção de falhas de produção se faz necessário (CHEN et al., 2021). Cada vez mais, sistemas baseados em visão computacional são propostos para realizar a detecção de defeitos superficiais com o objetivo de substituir operadores humanos e automatizar esse processo (MARTINS; PÁDUA; ALMEIDA, 2010). Essa substituição é vantajosa pois o trabalho de inspeção de defeitos normalmente é uma operação simples e repetitiva, e quando realizada por um operador humano, a natureza cansativa desse trabalho pode acarretar em falhas de processo.

Nos últimos anos, avanços de hardware, principalmente no que se refere às placas de processamento gráfico, e pesquisas na área causaram uma popularização do uso de redes neurais na solução de problemas de processamento de imagens, incluindo a área de detecção de defeitos em ambientes industriais (TULBURE; TULBURE; DULF, 2021).

Uma desvantagem comum em aplicações que utilizam inteligência artificial é a necessidade de um conjunto significativo de dados devidamente classificados para a realização do treinamento dos modelos. Uma tática que já se provou efetiva em diversas aplicações é a criação de dados artificiais a partir de dados originais, denominada de *data augmentation* (PEREZ; WANG, 2017).

Neste trabalho são implementados dois métodos de *data augmentation* com o objetivo de melhorar o desempenho de uma rede neural de detecção de defeitos em superfícies metálicas. Os resultados são validados com a realização de testes em dois bancos de dados, o Kolektor Surface-Defect Dataset e o Severstal Steel Defect Detection, e a rede neural que é utilizada como base é a rede mista de detecção de defeitos superficiais implementada por Božič, Tabernik e Skočaj (2021).

Na Seção 2, é apresentada uma revisão bibliográfica dos principais conceitos utilizados na elaboração desse trabalho, incluindo as técnicas de processamento de imagens utilizadas e definições importantes sobre redes neurais e *data augmentation*. A Seção 3 apresenta a metodologia do trabalho, na qual são detalhadas a arquitetura e o funcionamento da rede neural escolhida para os testes, o desenvolvimento dos algoritmos para geração das imagens, uma explicação sobre os bancos de dados utilizados e a definição de como é feita a validação dos resultados. Por fim, a Seção 4 apresenta diversos gráficos, tabelas e interpretações dos resultados, e uma comparação com trabalhos semelhantes na área de *data augmentation* para detecção de defeitos superficiais.

2 REVISÃO DA LITERATURA

Neste Capítulo, serão apresentados os principais conceitos necessários para a elaboração e entendimento deste trabalho. A Seção 2.1 aborda os conceitos básicos de processamento de imagens e detalha as técnicas que foram utilizadas no decorrer do trabalho. A Seção 2.2 explica o conceito, funcionamento e treinamento de redes neurais clássicas e convolucionais. Por fim, a Seção 2.3 explica e exemplifica o uso de *data augmentation* dentro do contexto de processamento de imagens.

2.1 Processamento de Imagens

Processamento de imagens é a área de estudo que busca obter informações a partir de imagens digitais. Essas informações podem ser outras imagens ou características extraídas destas (KANG, 2007), como a presença ou não de objetos ou defeitos, medições de distâncias entre pontos da imagem ou contagens de elementos.

Uma imagem digital pode ser definida por uma função discreta $f(x,y)$, onde x e y representam as coordenadas da imagem. Essa função pode ser representada por uma matriz bidimensional de tamanho $M \times N$, representada por

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \dots & a_{1,N-1} \\ \vdots & \vdots & & \\ a_{M-1,0} & a_{M-1,1} & \dots & a_{M-1,N-1} \end{bmatrix}, \quad (1)$$

em que cada elemento, chamado de píxel, possui um valor que representa sua intensidade a , onde $f(x,y) = a_{x,y}$. No caso de imagens em tons de cinza com 8 bits, os valores de intensidade de pixels são representados por inteiros contidos no intervalo $[0,255]$, em que o valor 0 é representado por pixels pretos, o valor 255 por pixels brancos, e todos os valores intermediários por diferentes valores na escala de cinza (GONZALES; WOODS, 2018a).

Para representar imagens coloridas, e não apenas em escala de cinza, normalmente são utilizadas câmeras RGB (*red, green and blue*), as quais possuem 4 sensores de luz para cada píxel, sendo 2 verdes, 1 vermelho e 1 azul. O uso desse tipo de câmera resulta em uma imagem de três canais, onde cada canal pode ser interpretado como uma imagem em escala de cinza.

As seções 2.1.1 e 2.1.2 apresentam as duas principais ferramentas de processamento de imagens utilizadas neste trabalho: transformações afins, que permitem a manipulação geométrica de imagens a partir de operações matriciais e interpolação de dados, e a mesclagem de imagens, que possibilita a geração de uma nova imagem a partir de outras duas imagens.

2.1.1 Transformações Afins

Transformações afins são operações geométricas em imagens compostas de duas etapas: a transformação espacial das coordenadas e a interpolação de intensidade, que define os valores dos pixels após a transformação. Essas transformações consistem nas operações de redimensionamento, espelhamento, translação, rotação e cisalhamento (GONZALES; WOODS, 2018b). Essas 5 transformações espaciais podem ser representadas pela equação matricial

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = A \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (2)$$

onde x e y são as coordenadas iniciais de um ponto, x' e y' são suas coordenadas finais e A é uma matriz cujos valores definem a natureza da operação desejada. A Tabela 1 apresenta cada uma das possíveis transformações afins em forma matricial e suas correspondentes equações de coordenada. Utilizando a representação matricial, é possível encadear diversas transformações afins realizando o produto de suas matrizes individuais. Essa propriedade é especialmente útil para realizar a correção do sistema de coordenadas da imagem utilizando a translação. Por exemplo, ao aplicar a operação de espelhamento horizontal, todas as coordenadas x da imagem estarão negativas. Assim, para a correta representação da imagem, é necessário realizar uma operação de translação com t_x igual à largura da imagem, retornando a mesma para sua posição original.

Tabela 1: Exemplos de transformações afins

Transformação	Matriz afim	Equações de coordenada
Redimensionamento e espelhamento	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = c_x x$ $y' = c_y y$
Rotação na origem	$\begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x \cos \theta - y \sin \theta$ $y' = x \sin \theta + y \cos \theta$
Translação	$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x + t_x$ $y' = y + t_y$
Cisalhamento horizontal	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x + s_h y$ $y' = y$
Cisalhamento vertical	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x$ $y' = s_v x + y$

Fonte: Adaptado de Gonzales e Woods (2018a)

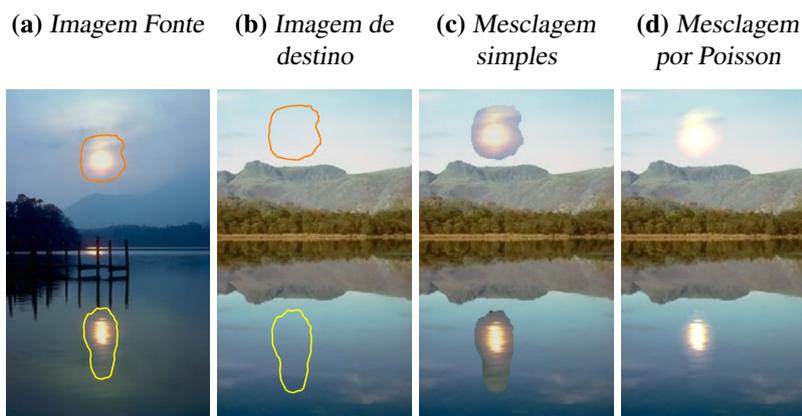
2.1.2 Mesclagem de Imagens

A mesclagem de imagens consiste na sobreposição de duas imagens diferentes, gerando uma nova imagem artificial. Para isso é necessário que se tenha uma imagem de destino,

uma imagem fonte e uma máscara binária que indique a região da imagem fonte que será sobreposta à imagem de destino, além de informar a localização final da imagem fonte na imagem de destino.

Existem diversas maneiras de realizar a mesclagem de imagens, a mais simples sendo a substituição dos pixels englobados pela máscara na imagem de destino pelos pixels da imagem fonte. Essa abordagem, no entanto, costuma gerar um resultado pouco natural, em que as fronteiras entre uma imagem e outra são facilmente notadas. Uma abordagem que apresenta resultados mais naturais é a mesclagem de imagens por Poisson, abordada por Pérez, Gangnet e Blake (2003). Esse método utiliza a informação do gradiente das imagens para realizar a mesclagem, e tem como restrição que a fronteira entre as imagens tenha os mesmos valores de pixel da imagem de destino. O gradiente de uma imagem é o campo vetorial obtido ao computar as derivadas parciais em x e y dos valores de intensidade de uma imagem, resultando em um campo com valores de maior magnitude em regiões com maiores variações de intensidade. As figuras 1a, 1b, 1c e 1d ilustram o procedimento de mesclagem, apresentando as imagens fonte, destino (com as máscaras binárias marcadas nas imagens) e os resultados dos dois métodos de mesclagem discutidos.

Figura 1: Operação de mesclagem de imagens



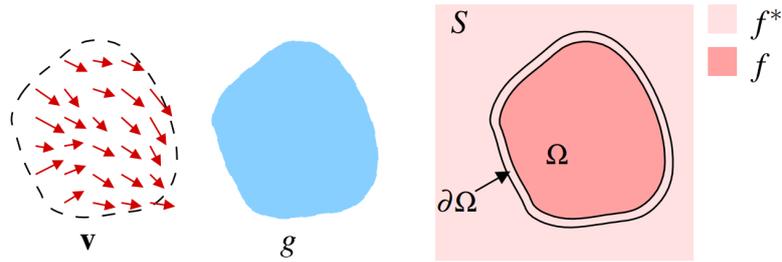
Fonte: Pérez, Gangnet e Blake (2003)

Para a explicação do método de mesclagem por Poisson, será utilizada a seguinte notação, demonstrada na Figura 2: S é um conjunto pertencente a \mathbb{R}^2 que representa o espaço definido pela imagem de destino, Ω é a região definida pela máscara binária, que possui uma fronteira $\partial\Omega$, f^* é uma função escalar conhecida (a imagem de destino) e f é a função escalar desconhecida que definirá os valores de intensidade dentro da região Ω . Por fim, o vetor \vec{v} é o chamado vetor guia, que em um primeiro momento será o gradiente da imagem fonte g , definida também dentro da região Ω .

Para realizar a mesclagem na região definida pela máscara binária, deve-se resolver o problema de minimização

$$\min_f = \iint_{\Omega} |\nabla f - \vec{v}|^2 \text{ com } f|_{\partial\Omega} = f^*|_{\partial\Omega} \quad (3)$$

onde $\nabla f = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}]$ representa o gradiente de f .

Figura 2: Visualização das notações utilizadas

Fonte: Pérez, Gangnet e Blake (2003)

A equação (3) possui uma solução única representada pela equação de Poisson com condições de fronteira de Dirichlet

$$\Delta f = \nabla \cdot \vec{v} \text{ sobre } \Omega \text{ com } f|_{\partial\Omega} = f^*|_{\partial\Omega} \quad (4)$$

onde $\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$ é o laplaciano de f , e $\nabla \cdot \vec{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$ representa o divergente do vetor de guia $\vec{v} = (u, v)$. Definindo o vetor guia como sendo o gradiente da imagem fonte g , ou seja, $\vec{v} = \nabla g$, obtém-se

$$\Delta f = \Delta g \text{ sobre } \Omega \text{ com } f|_{\partial\Omega} = f^*|_{\partial\Omega}. \quad (5)$$

O problema pode ser discretizado utilizando a grade de pixels de uma imagem. Mantendo as mesmas notações, S e Ω se tornam conjuntos finitos de pontos definidos no espaço discreto. Para cada píxel p em S , N_p é definido como sendo o conjunto de seus vizinhos 4-conectados (pixels diretamente acima, abaixo, à esquerda e à direita). Também se define $\langle p, q \rangle$ sendo um par de pixels tais que $q \in N_p$. A fronteira de Ω agora é definida como $\partial\Omega = \{p \in S \setminus \Omega : N_p \cap \Omega \neq \emptyset\}$. Definindo f_p como o valor de f em p , a tarefa é computar as intensidades $f|_{\Omega} = \{f_p, p \in \Omega\}$.

Discretizando a equação (3), temos:

$$\min_{f|_{\Omega}} = \sum_{\langle p, q \rangle \cap \Omega \neq \emptyset} (f_p - f_q - v_{pq})^2 \text{ com } f_p = f_p^* \text{ para todo } p \in \partial\Omega \quad (6)$$

onde v_{pq} é a projeção de $v(\frac{p+q}{2})$ na borda orientada $[p, q]$, ou seja, $v(\frac{p+q}{2}) \cdot \vec{p}q$. A solução desse problema de minimização se dá pelo seguinte conjunto de equações lineares:

$$\text{para todo } p \in \Omega, |N_p| f_p - \sum_{q \in N_p \cap \Omega} f_q = \sum_{q \in N_p \cap \partial\Omega} f_q^* + \sum_{q \in N_p} v_{pq} \quad (7)$$

A equação (7) forma um sistema linear esparço simétrico e positivo-definido, que pode ser resolvido por diversos métodos computacionais. O termo v_{pq} é obtido discretizando a função guia apresentada na equação (5), resultando em

$$\text{para todo } \langle p, q \rangle, v_{pq} = g_p - g_q. \quad (8)$$

O uso da função guia obtida na equação (8) resulta na perda completa da informação da imagem de destino f^* dentro da região Ω . Isso causa a perda da textura original da imagem e pode ser particularmente notado caso a imagem original seja rica em informação (gradiente elevado). Para reduzir os efeitos da perda de informação, é possível utilizar como função guia uma função que retenha o maior gradiente entre f^* e g , resultando em

$$\text{para todo } x \in \Omega, \vec{v}(x) = \begin{cases} \nabla f^*(x) & \text{se } |\nabla f^*(x)| > |\nabla g(x)| \\ \nabla g(x) & \text{se } |\nabla f^*(x)| < |\nabla g(x)| \end{cases}. \quad (9)$$

2.2 Redes Neurais

Redes neurais são um método de aprendizado de máquina que oferece soluções viáveis em diversas aplicações, como reconhecimento de voz e de padrões, diagnósticos médicos e detecção de fraudes (ABIODUN et al., 2018). Na área de visão computacional, o desenvolvimento e a implementação de redes neurais convolucionais, explicadas em maior detalhe na Seção 2.2.1, permitiram avanços em diversas áreas, como classificação de imagens, detecção e reconhecimento de objetos, segmentação de imagens e reconhecimento facial (LI, Z. et al., 2020).

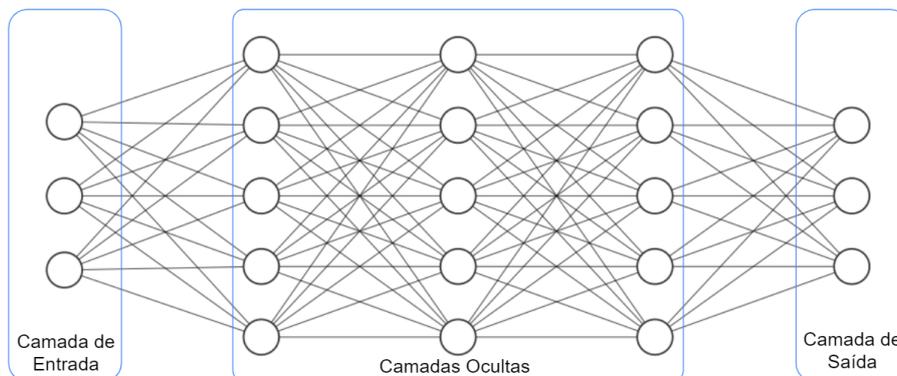
Redes neurais se baseiam no uso de um grande número de elementos de processamento interconectados, chamados de *neurons*. Seu nome se dá devido à semelhança entre o funcionamento de uma rede neural e o de um cérebro: ambos tem a capacidade de aprender como realizar tarefas por meio de exemplos (MAIND; WANKAR, 2014). Um *neuron* pode ser representado por

$$\hat{y} = \phi\left(\sum_{i=1}^m w_i x_i + b\right), \quad (10)$$

onde \hat{y} é a saída do *neuron*, x_i são as entradas, w_i e b são os parâmetros treináveis (peso e viés) e ϕ é uma função de ativação não linear, que tem como objetivo manter a saída em um intervalo específico.

Para formar uma rede neural, os *neurons* são organizados em camadas interconectadas. Normalmente, tem-se uma camada de entrada que contém a informação que será utilizada pela rede, seguida por camadas ocultas a partir das quais são extraídas as informações relevantes. Por fim, tem-se uma camada final que representa a saída da rede como um todo. Usualmente, o *neuron* de uma camada está conectado com todos os *neurons* da camada anterior. A Figura 3 mostra um diagrama com a organização básica de uma rede neural clássica completamente conectada.

Figura 3: Diagrama de uma rede neural básica



Fonte: Autor

A definição dos pesos dos *neurons* é realizada na etapa de treinamento. Como existe um número arbitrário de camadas entre a entrada e a saída do sistema, é necessário utilizar um algoritmo que seja capaz de atualizar os pesos de todos os *neurons* iterativamente,

buscando reduzir o erro entre a predição do sistema e a resposta esperada. Atualmente, o método mais comum de atualização dos pesos da rede é o algoritmo de *backpropagation* (LI, J. et al., 2012).

Intuitivamente, o algoritmo de *backpropagation* busca alterar os pesos de todos os *neurons* da rede de modo a reduzir uma função de custo. Essa função de custo deve quantificar a diferença entre o vetor de saída do sistema e a resposta esperada (por exemplo, calculando a diferença quadrada dos vetores). É computado o gradiente dessa função em relação a todos os pesos da rede, cujo valor indica como estes devem ser alterados de maneira a diminuir a função de custo. O algoritmo para o cálculo do gradiente de cada peso da rede é chamado de *backpropagation* pois o seu cálculo é realizado por camadas, começando pela última. O resultado do vetor de erro de uma camada é utilizado para estimar o gradiente da camada anterior (LI, J. et al., 2012).

Os pesos da rede são alterados utilizando o algoritmo de descida de gradiente, que altera levemente os pesos de cada *neuron* na direção negativa do gradiente calculado. Devido ao alto custo computacional envolvido no cálculo do gradiente, normalmente, divide-se o banco de dados em lotes menores, e a descida de gradiente de cada passo do treinamento da rede é calculada a partir de um desses lotes, e não do conjunto total de amostras.

Para evitar um viés na avaliação de uma rede neural, os dados são separados em três categorias: os dados de treinamento, que são utilizados para o aprendizado dos parâmetros da rede, os dados de validação, que permitem a comparação de treinamentos com diferentes configurações (pode se variar o número de camadas ocultas da rede, por exemplo), e os dados de teste, que são utilizados para obter uma avaliação final da rede escolhida por meio dos dados de validação (RIPLEY, 1996).

O uso dos dados de validação auxilia na detecção de problemas como o *overfitting*, em que um modelo de rede neural possui bons resultados com os dados utilizados durante o treinamento, mas apresenta perda de desempenho ao testar a rede com dados não vistos durante a etapa de treinamento. Já o uso dos dados de teste permite uma avaliação do modelo final sem levar em consideração o viés dos dados selecionados para treinamento ou dos dados utilizados para seleção do modelo final.

2.2.1 Redes Neurais Convolucionais

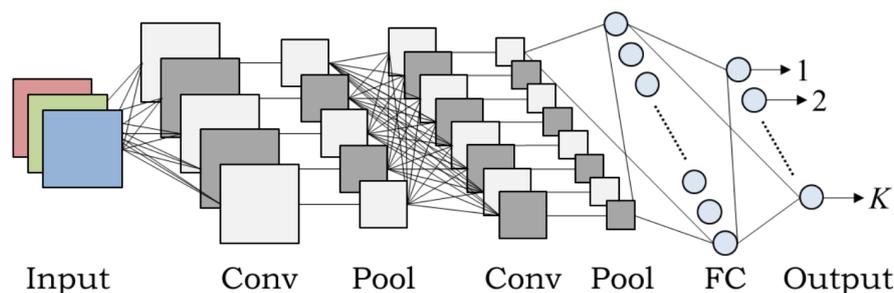
Redes neurais convolucionais (RNCs) são uma categoria de rede neural que se tornou popular para aplicações de visão computacional após a publicação dos resultados de uma rede desenvolvida por Krizhevsky, Sutskever e Hinton (2012) na tarefa de classificação de imagens no banco de dados ImageNet ILSVRC-2010. A RNC obteve erros top-1, que representa o índice de erro entre a predição correta e a obtida, e top-5, que indica o índice de erro entre a predição correta e as 5 maiores predições da rede, de 37,5%, e 17,0%. A melhor performance obtida até então, sem utilizar RNCs, era de 47,1% no erro top-1 e 28,2% no erro top-5.

A vantagem das redes neurais convolucionais no processamento de imagens está na arquitetura de suas camadas: a extração das características é feita pelas camadas de convolução e de *pooling*, e o mapeamento dessas características para a saída final da rede é feito por camadas completamente conectadas, como as vistas na Seção 2.2. A Figura 4 ilustra a arquitetura básica de uma RNC.

Uma camada de convolução é composta por uma série de filtros, também chamados de *kernels*. A diferença entre um filtro e os pesos das camadas completamente conectadas é que um filtro, assim como uma imagem, possui 3 dimensões (altura, largura e número de

canais), com cada um de seus elementos possuindo parâmetros treináveis. A aplicação do filtro se dá pela operação de convolução, em que se desloca o filtro por toda a imagem e computa-se um novo valor de pixel realizando o somatório entre os produtos dos pixels da imagem pelo valor do elemento do filtro na mesma posição. A Figura 5 apresenta uma operação de convolução entre uma imagem de dimensões $6 \times 6 \times 1$ e um filtro de dimensões $3 \times 3 \times 1$, seguido de uma operação de *pooling* a qual será explicada mais adiante.

Figura 4: Diagrama de uma rede neural convolucional básica



Fonte: Hidaka e Kurita (2017)

Assumindo que a imagem original possui dimensões $a_o \times l_o \times c_o$, e o filtro possui dimensões $a_k \times l_k \times c_o$, a imagem de saída possuirá dimensões $(a_o - a_k + 1) \times (l_o - l_k + 1) \times 1$. Como uma camada é composta por n kernels, as saídas dos filtros são combinadas, formando uma saída de tamanho $(a_o - a_k + 1) \times (l_o - l_k + 1) \times n$.

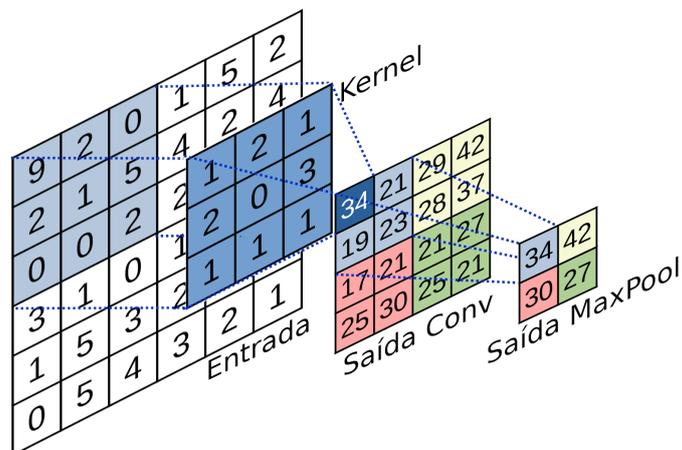
As camadas de *pooling* são utilizadas para reduzir a resolução das saídas dos filtros das camadas de convolução. Isso permite remover informações possivelmente redundantes que não são relevantes para o aprendizado. Esse *downsampling* é feito de maneira semelhante à convolução, porém o filtro realiza operações predefinidas, como o cálculo da média (*average pooling*) ou a obtenção do valor máximo dentro do *kernel* (*max pooling*). Também é introduzido o *stride*, que indica por quantos pixels o filtro será movido a cada operação. Por exemplo, uma camada de *max pooling* com um filtro 2×2 e um *stride* de 2 reduzirá as dimensões da sua entrada pela metade, como demonstrado na Figura 5.

Após diversas camadas de convolução e de *pooling*, as camadas finais de uma RNC podem conter camadas completamente conectadas, que realizam a decisão final. Também existem redes que têm como saída uma imagem, como é o caso das redes de segmentação semântica. O uso das camadas de convolução idealmente permite que as camadas próximas da entrada aprendam a detectar características espaciais de baixo nível, como linhas e cantos, e as camadas mais próximas da saída aprendam a reconhecer características de alto nível, como formas e objetos.

2.3 Data Augmentation

Um desafio comum na implementação de soluções baseadas em redes neurais é a grande quantidade de dados que normalmente é necessária para evitar o *overfitting* da rede e garantir a sua generalização (MIKOŁAJCZYK; GROCHOWSK, 2018). Muitos bancos de dados utilizados para testes de novas arquiteturas possuem um grande número de amostras por categoria como, por exemplo, o *ImageNet*, um banco utilizado para testes de classificação de imagens que possui mais de um milhão de amostras e 1000 categorias (DENG et al., 2009). Contudo, existem casos em que é muito custoso realizar a marcação de uma grande quantidade de dados e torna-se necessário buscar métodos alternativos para

Figura 5: Detalhamento das camadas de convolução e de pooling



Fonte: Autor

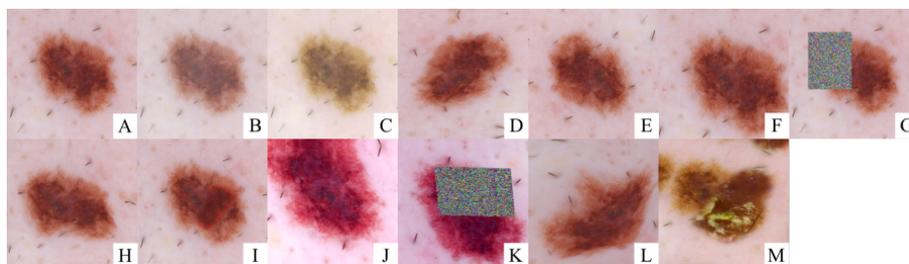
melhorar o desempenho e a capacidade de generalização de uma rede neural.

Um desses métodos é a *data augmentation*, que prevê, no contexto de visão computacional, o uso de diversas estratégias para gerar novas imagens a partir de um conjunto de dados já marcados. Caso as estratégias de *data augmentation* sejam implementadas de maneira adequada, o novo conjunto de dados representará de forma mais completa as possíveis entradas do sistema, e a rede possuirá uma melhor generalização.

Um método comum para a implementação de *data augmentation* envolve o uso de manipulações básicas de imagens, como espelhamentos, rotações, recortes, injeção de ruídos, manipulação do espaço de cores e combinações de imagens (SHORTEN; KHOSHGOFTAAR, 2019). Também é possível gerar imagens sintéticas utilizando os métodos de mesclagem de imagens abordados na Seção 2.1.2, ou redes neurais, como as *Generative Adversarial Networks* (GANs), que possuem a capacidade de criar imagens sintéticas baseadas nos seus dados de treinamento (BOWLES et al., 2018).

A Figura 6 apresenta exemplos de *data augmentation* utilizados em um contexto médico (análise de lesões de pele), na qual a imagem classificada como "A" é a original e as seguintes são imagens geradas a partir de métodos analíticos (sem o uso de redes neurais) de *data augmentation*.

Figura 6: Exemplos de *data augmentation*



Fonte: Perez, Vasconcelos et al. (2018)

3 METODOLOGIA

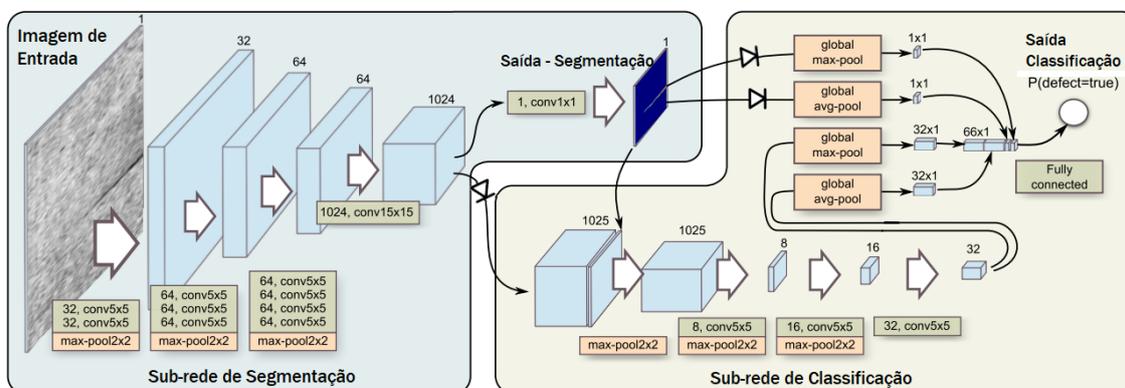
Neste Capítulo, será apresentada a metodologia utilizada para o desenvolvimento do presente trabalho. A Seção 3.1 descreve a arquitetura e o funcionamento da rede neural utilizada para avaliar os métodos de *data augmentation*, os quais se encontram detalhados na Seção 3.2. Na Seção 3.3 é feita uma descrição dos dois bancos de dados escolhidos e da estrutura dos arquivos utilizada. Por fim, a Seção 3.4 detalha o método escolhido para análise e validação dos resultados obtidos a partir dos treinamentos realizados com diferentes métodos de geração de imagens artificiais.

3.1 Rede Neural Escolhida

Para poder avaliar os resultados da inserção de dados artificiais em diversas condições de treinamento, optou-se por utilizar uma rede que permite o treinamento misto. Com isso, é possível avaliar se houve melhora utilizando apenas classificações a nível de imagem (fracamente classificadas), apenas classificações píxel a píxel (completamente classificadas) ou uma mistura destas.

A rede escolhida foi a implementada por Božič, Tabernik e Skočaj (2021) e é baseada em uma arquitetura previamente descrita por Tabernik et al. (2019). A rede é composta por duas sub-redes: uma rede de segmentação, que utiliza as marcações completas e amostras negativas para seu treinamento, e uma rede de classificação, que utiliza as informações a nível de imagem. A Figura 7 ilustra a arquitetura da rede.

Figura 7: Representação da arquitetura da rede



Fonte: Adaptado de Tabernik et al. (2019)

A sub-rede de segmentação tem como entrada uma imagem RGB ou em escala de cinza e possui um total de 11 camadas de convolução e 3 camadas *max-pool* de tamanho 2×2 .

Na última camada de convolução, é utilizado apenas um filtro de tamanho $1 \times 1 \times 1024$, resultando na imagem de segmentação, que é comparada com a máscara de segmentação, caso exista. A última e penúltima camadas da sub-rede de segmentação possuem uma dimensão 8 vezes menor que a imagem original, devido às 3 camadas de *max-pooling*.

A sub-rede de classificação tem como entrada a saída da sub-rede de segmentação e o mapa de características obtido na sua penúltima camada, composto por 1024 canais. Novamente, são realizadas operações de convolução e *maxpooling* até se obter uma saída composta por 32 canais. A penúltima camada é obtida realizando uma operação de *max-pool* global e média global em cada um dos 32 canais da camada anterior e da saída da rede de segmentação, resultando em uma camada de tamanho $1 \times 1 \times 66$, pois as operações globais obtêm o máximo e a média do mapa completo, resultando em apenas um valor por canal. Por fim, uma camada completamente conectada recebe como entrada esses 66 valores e tem como saída um valor que indica a chance de existir um defeito na imagem.

Para permitir o aprendizado misto é necessário realizar o treinamento das duas sub-redes de forma simultânea. Isso é feito combinando as funções de perda, resultando em

$$L_{total} = \lambda \cdot \gamma \cdot L_{seg} + (1 - \lambda) \cdot \delta \cdot L_{cls}, \quad (11)$$

onde L_{seg} e L_{cls} são as funções de perda das sub-redes de segmentação e classificação, δ é um peso adicional para a função de perda da sub-rede de classificação, λ é um fator de balanceamento que varia no decorrer do treinamento de acordo com a equação $\lambda = 1 - \frac{n}{n_{ep}}$, onde n é a época atual do treinamento e n_{ep} é o total de épocas. γ é um indicador da presença de marcação completa que permite que a sub-rede de segmentação seja treinada apenas com amostras negativas e amostras positivas com máscara, conforme a equação

$$\gamma = \begin{cases} 1, & \text{para amostras negativas} \\ 1, & \text{para amostras positivas com máscara} \\ 0, & \text{para amostras positivas sem máscara} \end{cases} \quad (12)$$

Para evitar uma interação indesejada entre as duas sub-redes, a propagação do erro do algoritmo de *backpropagation* é interrompida nos locais onde as sub-redes se conectam. Essa interrupção é representada por diodos na Figura 7. Com isso, o *backpropagation* da sub-rede de segmentação é calculado utilizando apenas a perda obtida a partir das máscaras das imagens.

3.2 Métodos de *Data Augmentation*

Devido ao desbalanceamento entre amostras positivas e negativas dos bancos de dados utilizados nesse trabalho, as operações de *data augmentation* foram realizadas somente nas amostras positivas. Além disso, a escolha da rede neural, abordada na Seção 3.1, permite a implementação e análise de dois tipos de métodos: um método independente de máscara, que pode ser utilizado em amostras fracamente ou completamente classificadas e que será detalhado na Seção 3.2.1, e um método dependente de máscara, que necessita da informação de localização e formato dos defeitos, que será detalhado na Seção 3.2.2. Os métodos descritos foram implementados em *Python*, utilizando as bibliotecas *Opencv* (BRADSKI, 2000) e *imgaug* (JUNG, 2018).

3.2.1 Data Augmentation por Espelhamento

Para a implementação do método independente de máscara, foi utilizada a transformação afim de espelhamento explicada na Seção 2.1.1, a qual permite a criação de três novas imagens para cada amostra (espelhamento vertical, horizontal e ambos). A Tabela 2 mostra as matrizes afins utilizadas e suas respectivas equações de coordenada, onde a_o é a altura da imagem (número de linhas na matriz) e l_o é a largura da imagem (número de colunas na matriz). O resultado desse método é exemplificado na Figura 8, onde a Figura 8a é uma amostra positiva do banco de dados Kolektor Surface-Defect Dataset 2 (KSDD2), descrito na Seção 3.3. As figuras 8b, 8c e 8d são os resultados das operações de espelhamento.

Tabela 2: *Tranformações de espelhamento*

Transformação	Matriz afim	Equações de coordenada
Espelhamento Vertical	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & a_o \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x$ $y' = a_o - y$
Espelhamento Horizontal	$\begin{bmatrix} -1 & 0 & l_o \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = l_o - x$ $y' = y$
Espelhamento Horizontal e Vertical	$\begin{bmatrix} -1 & 0 & l_o \\ 0 & -1 & a_o \\ 0 & 0 & 1 \end{bmatrix}$	$x' = l_o - x$ $y' = a_o - y$

Fonte: Autor

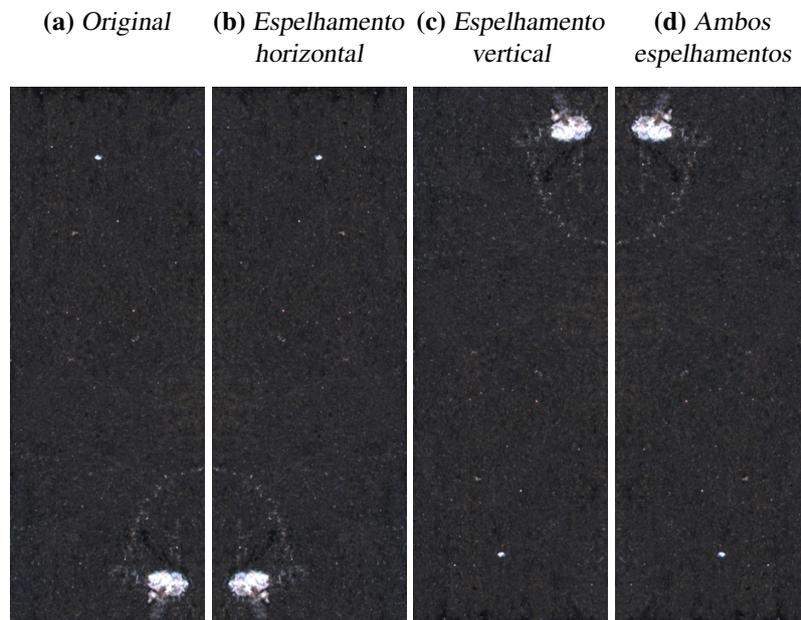
A operação de espelhamento foi escolhida porque não altera as dimensões ou a aparência geral da imagem, enquanto as operações afins de rotação, cisalhamento, translação e redimensionamento podem provocar anomalias não intencionais nas imagens geradas, prejudicando o treinamento. Com isso, basta aplicar a matriz de transformação na imagem original e, caso a amostra seja completamente classificada, aplicar a mesma matriz na sua máscara correspondente. O Apêndice A apresenta um pseudocódigo exemplificando como foi implementado o método de *data augmentation* por espelhamento.

3.2.2 Data Augmentation por Mesclagem de Defeitos

A implementação do segundo método de *data augmentation*, que consiste na geração de amostras positivas sintéticas e necessita de defeitos com marcações completas, foi realizada em duas etapas: a primeira consiste em aplicar transformações geométricas no recorte de um defeito e sua máscara correspondente, e a segunda consiste em utilizar o método de mesclagem de Poisson, visto na Seção 2.1.2, para inserir o novo defeito de forma natural em uma amostra negativa do banco de dados selecionada aleatoriamente. Como as transformações afins estão sendo aplicadas apenas nas regiões que contêm defeitos, é possível utilizar uma maior variedade de transformações.

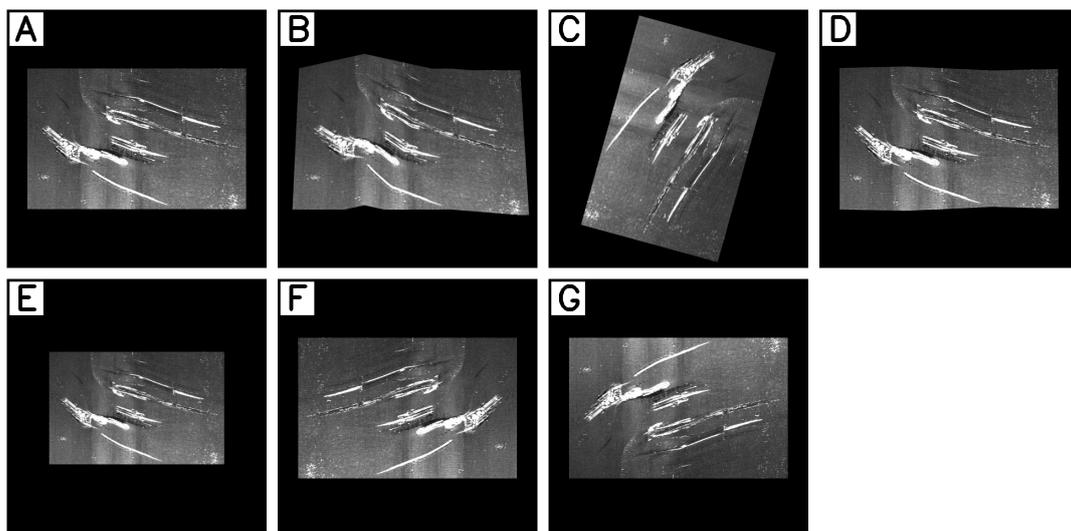
Para implementar as distorções dos defeitos, foi utilizada a biblioteca *imgaug*, que permite combinar e executar múltiplas transformações geométricas em uma imagem. Utilizando essa biblioteca, foram aplicadas as funções *PiecewiseAffine*, *Affine*, *Flipplr* e *Flipud*. A Figura 9 exemplifica cada uma dessas operações isoladamente, e os próximos parágrafos detalham o funcionamento de cada função de *data augmentation*.

Figura 8: Transformações afins utilizadas para o data augmentation independente de máscara



Fonte: Autor

Figura 9: Transformações aplicadas nos defeitos: (A) original (B) piecewiseAffine; (C) rotate; (D) elastic; (E) scale; (F) fliplr; (G) flipud;



Fonte: Autor

Para a implementação das transformações afins nos defeitos, foi utilizada a função *Affine*, que implementa algumas das transformações discutidas na Seção 2.1.1. Foram utilizadas as operações de redimensionamento (parâmetro *scale*), rotação (parâmetro *rotation*) e cisalhamento (parâmetro *shear*). Todos os parâmetros recebem os limites inferiores e superiores, e é escolhido de maneira pseudoaleatória um valor entre esses limites para cada imagem. Para a rotação, são definidos os ângulos; para o redimensionamento são definidos os fatores de escala; e para o cisalhamento são definidos os ângulos de cisalhamento.

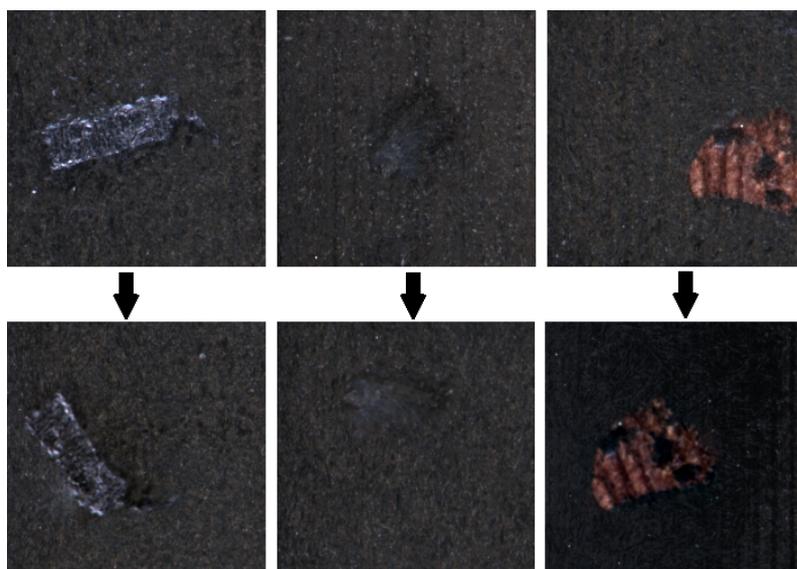
A função *PiecewiseAffine* gera uma grade de pontos na imagem (por padrão, 4X4) e movimentada esses pontos. Os pixels entre esses pontos serão movidos e interpolados de acordo com as posições finais dos pontos, gerando distorções locais. Os pontos se movem a partir de uma distribuição normal, onde o parâmetro *scale* indica os valores mínimos e máximos de movimentação em valores percentuais.

As funções *Fliplr* e *Flipud* fazem o espelhamento horizontal e vertical da imagem, respectivamente. Ambas possuem como parâmetro de entrada um número entre 0 e 1, que indica a probabilidade de a imagem ser espelhada.

O Apêndice B apresenta um pseudocódigo detalhando como é feito o *data augmentation* por mesclagem de defeitos. O algoritmo itera por todos os defeitos obtidos a partir das imagens completamente marcadas e insere uma quantidade arbitrária de defeitos sem possibilidade de sobreposição, representado pela variável *defeitosPorImagem*, em uma imagem base selecionada aleatoriamente. Esse processo pode ser repetido um número arbitrário de vezes (representado pela variável *numeroLoops*).

A Figura 10 exemplifica a geração de imagens sintéticas, utilizando imagens cortadas do banco de dados KSDD2. As transições entre a imagem original e o defeito colado são suaves, e os defeitos podem ficar mais claros ou mais escuros, dependendo das diferenças de cor dos fundos da imagem original e do defeito. Além disso, os defeitos das imagens sintéticas estão em posições e ângulos diferentes e apresentam deformações devido às transformações geométricas aplicadas antes da mesclagem.

Figura 10: Exemplos de *data augmentation* por mesclagem de imagens. Defeitos originais (acima) e imagens sintéticas (abaixo)



Fonte: Autor

Para cada defeito segmentado das imagens com marcação, foram geradas 3 imagens

sintéticas utilizando o método de Poisson com mistura de gradientes. Os parâmetros utilizados foram:

- 50% de chance de realizar espelhamento horizontal;
- 50% de chance de realizar espelhamento vertical;
- Rotação de 0 a 180°;
- Ângulo de cisalhamento entre -5 e 5°;
- Fator de escala entre 0,9 e 1,1;
- Para a *Piecewise Affine*, distorção de 1 a 5% em cada grade 4X4.

3.3 Bancos de Dados Utilizados

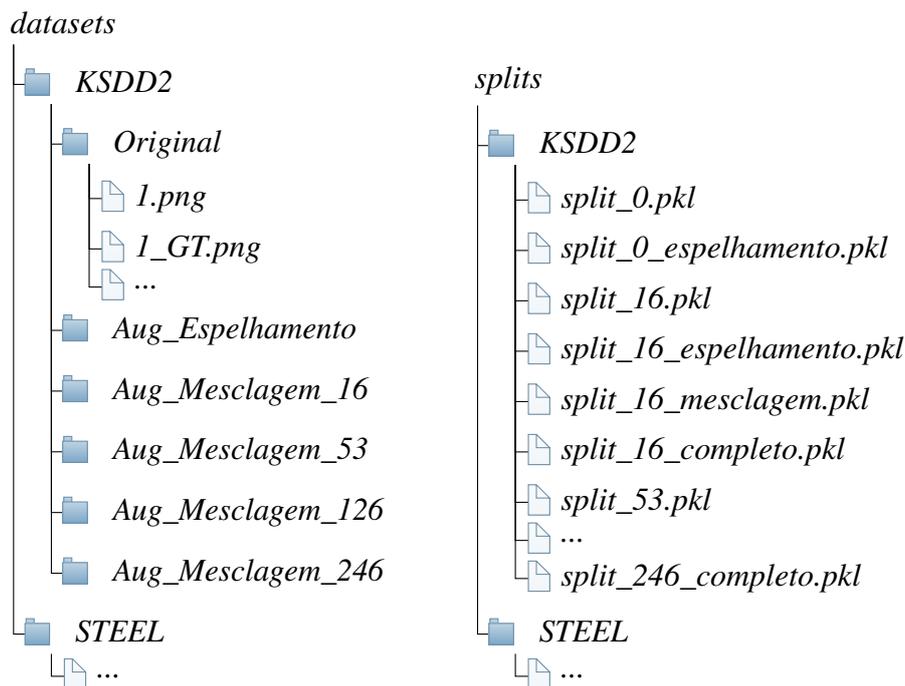
Para a avaliação dos novos treinamentos com dados artificiais, foram utilizados dois bancos de dados: o Kolektor Surface-Defect Dataset 2 (KSDD2) e o Severstal Steel Defect Detection (Severstal). Ambos são bancos de dados gerados a partir de imagens reais de superfícies metálicas que possuem marcações píxel a píxel.

Os bancos de dados utilizados possuem arquivos .pkl, uma extensão que permite o salvamento de variáveis do *python*. Esse arquivo possui três listas contendo as informações de dados de treinamento, de validação e de teste. Cada entrada dessas listas possui três informações: o nome da imagem, sua classificação (com ou sem defeito) e se deve ou não ser utilizada a marcação completa para a imagem. Com isso, é possível realizar treinamentos com diferentes proporções de dados completamente e fracamente classificados utilizando diferentes arquivos .pkl que apontam para os mesmos conjuntos de dados. Também é necessário informar para o script de treinamento a pasta raiz das imagens, já que o arquivo .pkl só possui o nome dos arquivos e não seu caminho completo.

Para permitir a adição de novos dados e facilitar a organização das pastas de trabalho, foi realizada uma modificação nos arquivos .pkl e nos scripts de treinamento: o campo de nome do arquivo para a lista de imagens de treinamento foi alterado para conter o caminho completo das imagens. Com isso é possível salvar as imagens sintéticas em diferentes pastas, de acordo com seu método de geração e dados de origem. Como os dados de validação e teste permanecerão os mesmos para todos os treinamentos, essa alteração foi feita somente para a lista que contém os dados de treinamento.

A estrutura das pastas dos bancos de dados e dos arquivos .pkl foi padronizada, resultando na estrutura ilustrada na Figura 11. Dentro de cada pasta do *dataset* existem outras 6 pastas, uma com as imagens originais e suas marcações e outras 5 que contêm as imagens geradas pelos métodos de *data augmentation*.

O banco de dados KSDD2 é composto por imagens de superfícies metálicas obtidas pelo grupo Kolektor. Ele é composto por imagens RGB com tamanho aproximado de 230 pixels de largura e 630 pixels de altura. O conjunto de treinamento possui 2085 amostras negativas e 246 amostras positivas, enquanto o conjunto de validação é composto por 894 amostras negativas e 110 positivas. As amostras negativas apresentam diversos defeitos com diferentes graus de severidade. A estrutura de pastas apresentada na Figura 11 foi baseada na estrutura padrão do banco de dados KSDD2, em que todas as imagens do *dataset* possuem uma máscara representada por uma imagem de mesmas dimensões, com os defeitos sendo representados por pixels brancos (valor 255), conforme apresentado na Figura 12.

Figura 11: Estrutura das pastas dos bancos de dados

Fonte: Autor

O banco Severstal Steel Defect Detection possui um número de amostras maior do que o KSDD2, possuindo com um total de 12568 imagens em escala de cinza. Além disso, as imagens possuem uma maior resolução, tendo 1600 pixels de largura e 256 pixels de altura. Para reduzir o tempo de treinamento e avaliar as imagens em uma situação em que se assume uma quantidade pequena de amostras, foi utilizado um conjunto de 300 amostras positivas e 4143 amostras negativas para o conjunto de treinamento, e 1200 amostras positivas e 1200 negativas para o conjunto de validação. Essa redução no tamanho do banco de dados é a mesma utilizada por Božič, Tabernik e Skočaj (2021) em um dos testes da rede neural que está sendo utilizada como base para este trabalho.

Quanto às marcações das amostras positivas, foi necessário criar um *script* em *python* para gerar as imagens das máscaras, buscando um resultado semelhante ao disponibilizado para o banco de dados KSDD2. Isso ocorreu porque, para reduzir o tamanho do dataset, as marcações dos defeitos do Severstal Steel Defect Detection são disponibilizadas em um arquivo de texto no formato de codificação *run-length*. Nesse método de codificação a imagem é transformada em uma lista de uma dimensão contendo os valores de intensidade de píxel. Cada imagem possui um conjunto de pares de valores que representam a posição do píxel na lista e a quantidade de pixels que representam um defeito a partir dele. Cada imagem possui um número arbitrário desses pares, possibilitando a representação de qualquer marcação de defeito.

O banco Severstal Steel Defect Detection possui uma maior variedade de superfícies e defeitos e suas marcações das amostras positivas são menos precisas. Devido a esses fatores, o Severstal é um banco com uma complexidade maior do que o KSDD2. Além disso, o maior número de amostras para treinamento e teste do Severstal pode gerar resultados mais consistentes. A Figura 13 apresenta uma amostra positiva do banco de

Figura 12: Exemplo de defeito (esquerda) e máscara (direita) do banco de dados KSDD2

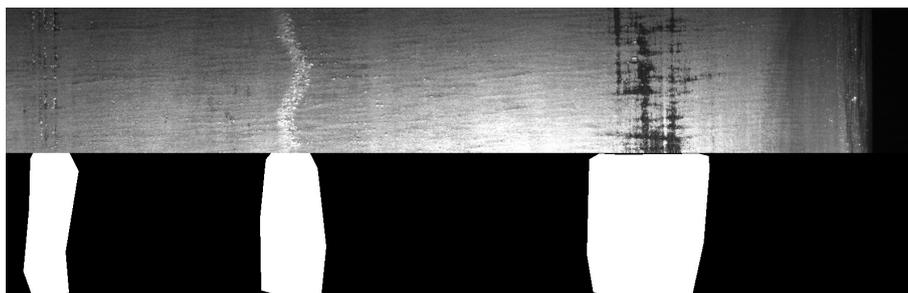


Nota: Na Figura é representada apenas a região com defeito da imagem

Fonte: Autor

dados Severstal e sua máscara decodificada.

Figura 13: Exemplo de defeito (acima) e máscara (abaixo) do banco de dados Severstal



Fonte: Autor

3.4 Validação Experimental

Para cada banco de dados foram criados 5 conjuntos de imagens e marcações, variando o número de amostras positivas com máscara de segmentação. Para o banco KSDD2 foram feitos conjuntos com 0, 16, 53, 126 e 246 amostras positivas utilizando as marcações completas, enquanto para o banco Severstal foram feitos conjuntos com 0, 10, 50, 150 e 300 amostras positivas com máscara, as mesmas divisões utilizadas por Božič, Tabernik e Skočaj (2021). Cabe ressaltar que as amostras sem máscara não foram excluídas do conjunto de treinamento, apenas ignorou-se suas marcações pixel a pixel. Assim, foi possível avaliar os métodos de *data augmentation* com bancos de dados fracamente e completamente supervisionados e com uma mistura entre marcações completas e a nível de imagem.

Para cada proporção de amostras positivas completamente marcadas foram realizados 4 treinamentos, que foram denominados da seguinte maneira: treinamento "base", sem utilização de amostras geradas artificialmente; "mesclagem", em que foi utilizado o método de mesclagem por Poisson em todas as amostras positivas com máscara; "espelhamento", em que foi utilizado o método de espelhamento em todas as amostras positivas; e "completo", em que foram utilizados ambos os métodos. Para o caso de 0 amostras com máscara, foram utilizados somente os treinamentos "base" e "espelhamento", pois não é possível gerar imagens pelo método de mesclagem sem amostras completamente classificadas.

Foram realizados dois treinamentos para cada combinação de método e número de marcações completas por banco de dados, totalizando 72 treinamentos. Em cada treinamento, foram registradas as seguintes métricas para comparação:

- Exatidão: percentual de amostras positivas e negativas corretamente classificadas. Calculado por $\frac{TP+TN}{TP+TN+FP+FN}$, onde TP e TN são as amostras positivas e negativas corretamente classificadas (*true positive* e *true negative*), e FP e FN são as amostras erroneamente classificadas como positivas e negativas (*false positive* e *false negative*).
- Precisão: dentre as amostras classificadas como positivas, quantas realmente são positivas. Calculado por $\frac{TP}{TP+FP}$.
- Revocação: dentro do total de amostras positivas, quantas foram classificadas como positivas. Calculado por $\frac{TP}{TP+FN}$.
- AUC (*Area under the ROC Curve*): Avalia a performance em todos os limiares de classificação. O limiar é variado entre 0 e 1 e é desenhada uma curva com a revocação e o índice de falsos positivos ($\frac{FP}{FP+TN}$) em um gráfico, denominada ROC (*receiver operating characteristic curve*). o AUC é a integral do gráfico desenhado.
- PM (precisão média): Semelhante ao AUC, porém é a área sob a curva precisão-revocação.

Para as métricas de exatidão, precisão e revocação, os valores utilizados foram os do limiar de reprovação da rede que apresentou o maior F1 score, definido como

$$F1 = 2 \times \frac{Pr \times Rev}{Pr + Rev}, \quad (13)$$

onde Pr é a precisão e Rev é a revocação.

A avaliação das métricas foi realizada qualitativamente por meio de gráficos e tabelas comparativas, e quantitativamente por meio da teoria de experimentos fatoriais com dois fatores, utilizando análise de variância (ANOVA), que permite verificar estatisticamente se os fatores que variam no decorrer dos testes influenciam os resultados de maneira significativa. Os fatores cuja significância será analisada serão os diferentes métodos de *data augmentation* e o número de amostras completamente classificadas por experimento. No caso deste trabalho, essa análise permite averiguar se houve uma diferença estatística nas métricas ao alterar os métodos de *data augmentation*, bem como se o uso dos métodos gera resultados com métricas acima da diferença mínima significativa calculada. Através desse método é realizado o cálculo da razão da variação entre as médias dentro de um fator e a variação dentro das amostras gerais. Caso esse valor seja maior que um valor tabelado, existe uma influência estatisticamente significativa causada pela variação do fator avaliado (RIBEIRO; CATEN, 2011).

4 RESULTADOS E DISCUSSÕES

Neste Capítulo, serão apresentados e analisados os resultados dos testes descritos na Seção 3.4. Na Seção 4.1, é feita uma análise de variância (ANOVA) para as métricas de ambos os bancos de dados, utilizando o conceito de experimentos fatoriais. A Seção 4.2 apresenta uma análise dos resultados dos ensaios com os bancos de dados Severstal Steel Defect Detection e KSDD2. Os parâmetros de treinamento foram os mesmos utilizados em Božič, Tabernik e Skočaj (2021) e os treinamentos duraram cerca de 3 horas para o banco de dados KSDD2 e 11 horas para o banco de dados Severstal, utilizando uma placa de vídeo NVidia RTX 2080 Super.

4.1 Análise de Variância (ANOVA)

Foram realizados 2 ensaios para cada combinação de método e número de marcações completas para cada banco de dados. A partir das médias dos resultados das métricas obtidas montou-se a Tabela 3, para o banco de dados Severstal, e a Tabela 4, para o banco de dados KSDD2. Os apêndices C e D contém as tabelas para cada um dos dois conjuntos de ensaios de cada banco de dados. Como o conjunto de dados de validação é menor para o banco KSDD2, houve uma diferença maior entre os resultados das métricas para as repetições dos ensaios.

A primeira análise de variância foi realizada utilizando os valores da precisão média para os ensaios realizados com o banco de dados Severstal Steel Defect Detection. Primeiramente, foi montado o quadro auxiliar mostrado na Tabela 5, onde cada valor é a soma da precisão média dos 2 ensaios realizados para a combinação de número de amostras completamente classificadas e método de *data augmentation*. Para simplificar os cálculos, não foram incluídos dados com 0 amostras, que possuem resultados apenas para os métodos base e espelhamento.

A partir do quadro auxiliar é possível calcular os valores da Tabela 6, onde GL é o número de graus de liberdade de cada fator. Definindo o fator amostras como A e o seu número de níveis como a , e o fator método como B e seu número de níveis como b , a interação como AB , o resíduo como R , o total como T e o número de repetições para cada experimento como n , temos que o $GL_A = (a - 1)$, $GL_B = (b - 1)$, $GL_{AB} = (a - 1)(b - 1)$, $GL_R = ab(n - 1)$ e $GL_T = abn - 1$. A coluna soma de quadrados (SQ) é calculada da seguinte forma, onde TOT é a soma dos valores de todos os experimentos, A_i é a soma dos experimentos no nível i do fator A , B_i é a soma dos experimentos no nível i do fator B e $(AB)_{ij}$ é a soma de experimentos no nível i de A e nível j de B (soma da repetição):

Tabela 3: Média das métricas - Severstal

Marcações Completas	Método	AUC	PM	Exatidão	Precisão	Revocação
0	Base	0,8718	0,8876	0,7838	0,7622	0,8271
0	Espelhamento	0,9046	0,9141	0,8269	0,8195	0,8388
10	Base	0,9170	0,9235	0,8338	0,8146	0,8646
10	Mesclagem	0,9209	0,9285	0,8452	0,8400	0,8529
10	Espelhamento	0,9447	0,9518	0,8733	0,8663	0,8846
10	Completo	0,9409	0,9497	0,8735	0,8723	0,8754
50	Base	0,9389	0,9472	0,8727	0,8838	0,8583
50	Mesclagem	0,9353	0,9455	0,8683	0,8789	0,8546
50	Espelhamento	0,9490	0,9565	0,8848	0,8935	0,8738
50	Completo	0,9555	0,9624	0,8973	0,9077	0,8846
150	Base	0,9607	0,9630	0,8988	0,9012	0,8958
150	Mesclagem	0,9617	0,9660	0,9038	0,9060	0,9013
150	Espelhamento	0,9745	0,9779	0,9260	0,9276	0,9242
150	Completo	0,9744	0,9787	0,9283	0,9310	0,9258
300	Base	0,9645	0,9699	0,9081	0,9124	0,9033
300	Mesclagem	0,9693	0,9742	0,9181	0,9285	0,9063
300	Espelhamento	0,9841	0,9858	0,9477	0,9602	0,9342
300	Completo	0,9744	0,9798	0,9392	0,9466	0,9308

Fonte: Autor

Tabela 4: Média das métricas - KSDD2

Marcações Completas	Método	AUC	PM	Exatidão	Precisão	Revocação
0	Base	0,9065	0,7503	0,9402	0,7786	0,6455
0	Espelhamento	0,9398	0,8380	0,9582	0,8467	0,7545
16	Base	0,9430	0,8621	0,9622	0,8914	0,7455
16	Mesclagem	0,9576	0,8946	0,9706	0,9307	0,7909
16	Espelhamento	0,9585	0,8898	0,9706	0,9328	0,7909
16	Completo	0,9612	0,9009	0,9706	0,9086	0,8136
53	Base	0,9507	0,8902	0,9721	0,9328	0,8045
53	Mesclagem	0,9530	0,9000	0,9726	0,9570	0,7864
53	Espelhamento	0,9679	0,9167	0,9761	0,9726	0,8045
53	Completo	0,9693	0,9225	0,9746	0,9831	0,7818
126	Base	0,9659	0,9202	0,9761	0,9433	0,8318
126	Mesclagem	0,9720	0,9254	0,9791	0,9406	0,8636
126	Espelhamento	0,9793	0,9320	0,9801	0,9412	0,8727
126	Completo	0,9823	0,9401	0,9801	0,9370	0,8773
246	Base	0,9674	0,9357	0,9811	0,9462	0,8773
246	Mesclagem	0,9835	0,9496	0,9776	0,8960	0,9000
246	Espelhamento	0,9826	0,9462	0,9796	0,9736	0,8364
246	Completo	0,9855	0,9492	0,9806	0,9599	0,8591

Fonte: Autor

Tabela 5: Quadro auxiliar PM - Severstal

Método \ Amostra	Base	Mesclagem	Espelhamento	Completo	Total
10	1,8470	1,8569	1,9035	1,8994	7,5068
50	1,8944	1,8909	1,9131	1,9249	7,6233
150	1,9259	1,9320	1,9557	1,9573	7,7710
300	1,9399	1,9484	1,9715	1,9596	7,8195
Total	7,6072	7,6283	7,7439	7,7412	30,7206

Fonte: Autor

$$TC = \frac{TOT}{abn} \quad (14)$$

$$SQ_A = \frac{\sum_{i=1}^a (A_i)^2}{bn} - TC \quad (15)$$

$$SQ_B = \frac{\sum_{i=1}^b (B_i)^2}{an} - TC \quad (16)$$

$$SQ_{AB} = \frac{\sum_{i,j=1}^{a,b} (AB)_{ij}^2}{n} - TC - SQ_A - SQ_B \quad (17)$$

$$SQ_{Tot} = \frac{(\text{soma de todos experimentos})^2}{abn} - TC \quad (18)$$

$$SQ_R = SQ_{Tot} - SQ_A - SQ_B - SQ_{AB} \quad (19)$$

A coluna dos quadrados médios é calculada por $QM = \frac{SQ}{GL}$, e a coluna do fator F por $F_{cal} = \frac{QM}{QM_R}$. Os valores da coluna F_{tab} são obtidos utilizando uma tabela da distribuição F com um grau de significância de 0,05.

Tabela 6: Quadro ANOVA para a PM - Severstal

Origem	GL	SQ	QM	F_{cal}	F_{tab}
Amostras	3	0,0076189	0,0025396	100,61	3,24
Método	3	0,0019761	0,0006587	26,09	3,24
Interação	9	0,0003357	0,0000373	1,48	2,54
Resíduo	16	0,0004039	0,0000252		
Total	31	0,0103347			

Fonte: Autor

Ao computar os valores da Tabela 6, pode-se verificar que $F_{cal} > F_{tab}$ para os fatores amostra e método, indicando que tanto a variação do número de marcações completamente classificadas e a variação dos métodos de *data augmentation* utilizados causaram uma

variação estatística significativa nos resultados do treinamento. Para a interação, $F_{cal} < F_{tab}$, indicando que não houve interação entre os fatores avaliados.

Também é possível fazer um desdobramento da interação realizando uma análise das variâncias do fator método dentro de cada nível do fator amostras, reduzindo o efeito da variação do número de amostras no cálculo de F_{cal} . Nesse caso, a equação para calcular a soma dos quadrados é

$$SQ_i = \sum_{j=1}^b \frac{(T_{ij})^2}{m_{ij}} - \frac{A_i^2}{n_i}, \quad (20)$$

onde i é o índice do nível do fator que não está sendo avaliado (fator amostras), n_i é o número de ensaios realizados com o nível i , A_i é a soma de todos os ensaios realizados com o nível i , T_{ij} é a soma dos ensaios realizados com o fator amostras i e fator método j e m_{ij} é o número desses ensaios.

Refazendo os cálculos analisando apenas a influência do método dentro de cada nível do fator amostras, é obtida a Tabela 7. Em todos os casos, $F_{cal} > F_{tab}$, o que indica que a variação do método causa uma variação estatística significativa em todos os casos para a métrica avaliada.

Tabela 7: Desdobramento da interação do data augmentation para a PM - Severstal

Causas da variação	GL	SQ	QM	F_{cal}	F_{tab}
Método/10	3	0,0012548	0,0004183	16,57	3,24
Método/50	3	0,0003845	0,0001282	5,08	3,24
Método/150	3	0,0003892	0,0001297	5,14	3,24
Método/300	3	0,0002834	0,0000945	3,74	3,24

Fonte: Autor

A partir do valor dos quadrados médios do resíduo QM_r , de um valor tabelado q , chamado de amplitude estudentizada, e do número de repetições de cada ensaio (m), é possível calcular a diferença mínima significativa, que é a diferença mínima entre médias dos ensaios que representa uma diferença significativa nos resultados. O valor da diferença mínima significativa é calculado por $\Delta = q\sqrt{\frac{QM_r}{m}}$. Utilizando o valor tabelado de $q = 4,046$, obtém-se que $\Delta = 0,01016$.

A Tabela 8 mostra a diferença entre a média da PM para os métodos de *data augmentation* e o treinamento base para 10, 50, 150 e 300 amostras. Os valores marcados em verde indicam que a diferença foi maior que a diferença mínima significativa calculada. É possível notar que os métodos completo e espelhamento apresentaram diferença significativa na maioria dos casos, enquanto o método de mesclagem não apresentou melhora significativa.

A ANOVA foi realizada com as outras 4 métricas obtidas nos experimentos com o banco Severstal, cujas tabelas montadas para a análise estão no Apêndice E. Foi calculada a diferença mínima significativa para cada métrica, obtendo os valores $\Delta_{AUC} = 0,01027$, $\Delta_{Exatidao} = 0,02006$, $\Delta_{Precisao} = 0,03476$ e $\Delta_{Revocacao} = 0,02623$. Utilizando esses valores, é possível montar a Tabela 9, que mostra a diferença das médias das métricas obtidas variando os métodos de *data augmentation* utilizados no treinamento e o treinamento base, com as diferenças maiores que a diferença mínima significativa marcadas novamente em verde.

Tabela 8: Diferença das médias da PM - Severstal

Método \ Amostra	Amostra			
	10	50	150	300
Completo	0,02620	0,01526	0,01686	0,00987
Espelhamento	0,02827	0,00936	0,01490	0,01582
Mesclagem	0,00496	-0,00171	0,00306	0,00425

Fonte: Autor

Tabela 9: Diferença das médias - Severstal

Marcações Completas	Método	AUC	PM	Exatidão	Precisão	Revocação
10	Mesclagem	0,0039	0,0050	0,0115	0,0253	-0,0117
10	Espelhamento	0,0277	0,0283	0,0396	0,0517	0,0200
10	Completo	0,0239	0,0262	0,0398	0,0577	0,0108
50	Mesclagem	-0,0036	-0,0017	-0,0044	-0,0049	-0,0037
50	Espelhamento	0,0101	0,0094	0,0121	0,0098	0,0154
50	Completo	0,0166	0,0153	0,0246	0,0240	0,0263
150	Mesclagem	0,0010	0,0031	0,0050	0,0048	0,0054
150	Espelhamento	0,0138	0,0149	0,0273	0,0264	0,0283
150	Completo	0,0138	0,0169	0,0404	0,0454	0,0350
300	Mesclagem	0,0048	0,0043	0,0100	0,0162	0,0029
300	Espelhamento	0,0196	0,0158	0,0396	0,0479	0,0308
300	Completo	0,0099	0,0099	0,0310	0,0343	0,0275

Fonte: Autor

A Tabela 9 mostra que o método de mesclagem não apresentou uma diferença significativa para nenhum dos casos; o método de espelhamento, porém, apresentou uma diferença significativa em 13 dos 20 casos, e o método completo apresentou uma diferença significativa em 15 dos 20 casos.

O mesmo procedimento de tabulação e cálculo da ANOVA foi realizado para os resultados dos ensaios do banco de dados KSDD2. A Tabela 10 apresenta os resultados do cálculo da ANOVA para a métrica precisão média.

Tabela 10: Quadro ANOVA da PM - KSDD2

Origem	GL	SQ	QM	F_{cal}	F_{tab}
Amostra	3	0,0155984	0,0051995	234,34	3,24
Método	3	0,0029253	0,0009751	43,95	3,24
Interação	9	0,0008573	0,0000953	4,29	2,54
Resíduo	16	0,0003550	0,0000222		
Total	31	0,0197360			

Fonte: Autor

Os cálculos da ANOVA para a precisão média dos ensaios com o banco KSDD2 mostram que $F_{cal} > F_{tab}$ para os fatores amostra e método, indicando que ambos os fatores causaram uma variação estatística significativa nos resultados do treinamento. Para a

interação, os cálculos mostram que $F_{cal} > F_{tab}$, indicando que houve interação entre os fatores avaliados. Com isso, é necessário realizar novamente o desdobramento da interação e avaliar a variância das médias em cada nível do fator amostra para eliminar a influência deste nos cálculos, conforme demonstrado na Tabela 11.

Tabela 11: Desdobramento da interação do data augmentation da PM - KSDD2

Causas da variação	GL	SQ	QM	F_{cal}	F_{tab}
Método/16	3	0,0017623	0,0005874	26,48	3,24
Método/53	3	0,0013261	0,0004420	19,92	3,24
Método/126	3	0,0004432	0,0001477	6,66	3,24
Método/246	3	0,0002510	0,0000837	3,77	3,24

Fonte: Autor

Para todos os níveis avaliados ao analisar o fator método, $F_{cal} > F_{tab}$, o que indica que em todos os casos houve uma variação significativa causada pela variação dos métodos de *data augmentation* utilizados. Realizando novamente o cálculo da diferença mínima significativa, obtém-se que $\Delta = 0,00954$. A Tabela 12 apresenta as diferenças das médias entre as métricas obtidas com os diferentes métodos de *data augmentation* e o treinamento base, com os valores marcados em verde sendo maiores que a diferença mínima significativa. Para o caso do banco de dados KSDD2, o único ensaio que não apresentou uma melhora acima da diferença mínima significativa foi aquele em que foi utilizado o método de mesclagem com 126 amostras completamente classificadas.

Tabela 12: Diferença das médias da PM - KSDD2

Método \ Amostra	Amostra			
	16	53	126	246
Completo	0,03886	0,03228	0,02902	0,01348
Espelhamento	0,02776	0,02647	0,01179	0,01043
Mesclagem	0,03253	0,00986	0,00515	0,01381

Fonte: Autor

Novamente, foi calculada a ANOVA a partir das outras 4 métricas, cujas tabelas montadas para análise estão no Apêndice F. Foi calculada a diferença mínima significativa para cada métrica, obtendo os valores $\Delta_{AUC} = 0,01042$, $\Delta_{Exatidao} = 0,00397$, $\Delta_{Precisao} = 0,04037$ e $\Delta_{Revocacao} = 0,04410$. Utilizando esses valores, é possível montar a Tabela 13, que mostra a diferença das médias das métricas obtidas entre os métodos de *data augmentation* utilizados no treinamento e o treinamento base, com as diferenças maiores que a diferença mínima significativa marcadas em verde.

A Tabela 13 mostra que o caso com 16 marcações completamente classificadas foi o que apresentou um número maior de métricas acima da diferença mínima significativa. Avaliando todos os níveis de marcação, percebeu-se que o método de mesclagem apresentou uma diferença das médias acima da diferença mínima significativa em 7 dos 20 casos e que o método de espelhamento apresentou uma diferença significativa em 13 dos 20 casos e o método completo apresentou uma diferença significativa em 12 dos 20 casos.

Tabela 13: Diferença das médias - KSDD2

Marcações Completas	Método	AUC	PM	Exatidão	Precisão	Revocação
16	Mesclagem	0,0146	0,0325	0,0085	0,0393	0,0455
16	Espelhamento	0,0156	0,0278	0,0085	0,0413	0,0455
16	Completo	0,0182	0,0389	0,0085	0,0172	0,0682
53	Mesclagem	0,0023	0,0099	0,0005	0,0242	-0,0182
53	Espelhamento	0,0172	0,0265	0,0040	0,0398	0,0000
53	Completo	0,0186	0,0323	0,0025	0,0503	-0,0227
126	Mesclagem	0,0061	0,0051	0,0030	-0,0028	0,0318
126	Espelhamento	0,0135	0,0118	0,0040	-0,0022	0,0409
126	Completo	0,0197	0,0290	0,0045	0,0165	0,0273
246	Mesclagem	0,0160	0,0138	-0,0035	-0,0502	0,0227
246	Espelhamento	0,0152	0,0104	-0,0015	0,0275	-0,0409
246	Completo	0,0181	0,0135	-0,0005	0,0137	-0,0182

Fonte: Autor

4.2 Análise dos Resultados

A partir das métricas obtidas nos ensaios realizados, foram montados gráficos de dois fatores ilustrando o desempenho do sistema para cada métrica avaliada. A Figura 14a compara a precisão média utilizando os 4 métodos de treinamento (base, mesclagem, espelhamento e completo). Cada barra vertical representa uma média da métrica, com sua cor indicando o número de amostras completamente classificadas utilizadas para geração das imagens sintéticas e treinamento da rede. No eixo horizontal do gráfico, é indicado o método de treinamento utilizado.

As figuras 14b, 14c, 14d e 14e apresentam gráficos de dois fatores onde as demais métricas são plotadas em função do método de treinamento para os diferentes números de amostras. É possível observar que todas as métricas possuem um comportamento semelhante, sendo que o gráfico da revocação é o único onde o desempenho com 10 amostras completamente classificadas é melhor do que com 50, com exceção do caso de treinamento completo. Para o caso com 0 amostras completamente classificadas, o método de espelhamento ocasionou uma melhora em todas as métricas.

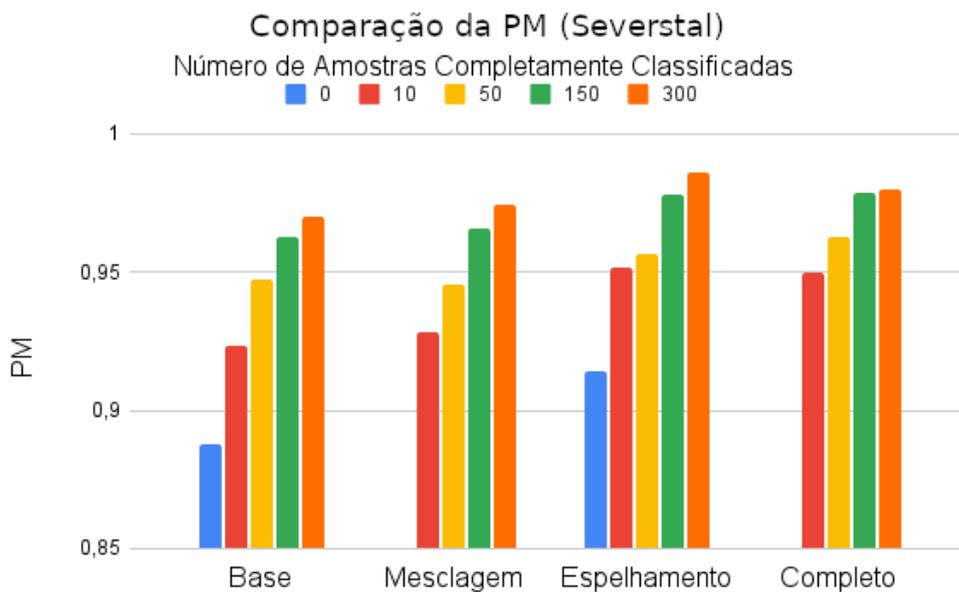
Comparando os métodos de espelhamento e completo com os treinamentos sem *data augmentation*, percebeu-se que ambos obtiveram métricas melhores em todos os casos, enquanto o método de mesclagem obteve métricas melhores em 14 dos 20 casos, atingindo um pior desempenho do que o treinamento base nas 5 métricas do treinamento com 50 amostras completas e na métrica de revocação com 10 amostras completas.

Comparando os dois métodos com melhor desempenho (espelhamento e completo), notou-se que ambos apresentaram resultados semelhantes, com o método completo obtendo melhor desempenho em 11 dos 20 casos. O método completo apresentou piores resultados nos treinamentos com o menor e maior número de amostras completamente classificadas (10 e 300, respectivamente). A Tabela 14 mostra em quantos casos cada um dos métodos de treinamento obteve a melhor métrica.

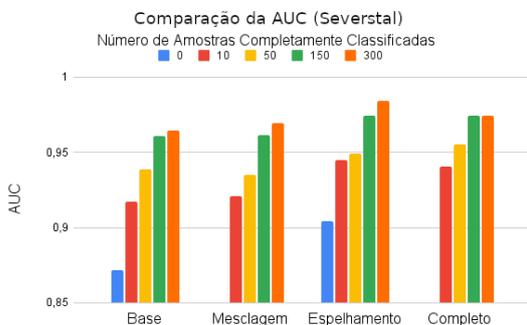
O mesmo procedimento descrito na Seção 4.2 foi realizado para o banco de dados KSDD2. As figuras 15a, 15b, 15c, 15d e 15e mostram os gráficos para as métricas calculadas. Para as métricas precisão média e AUC, é possível observar um comportamento

Figura 14: Gráficos de comparação para o banco de dados Severstal

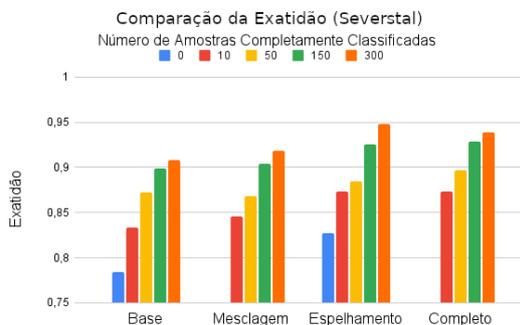
(a) Precisão Média



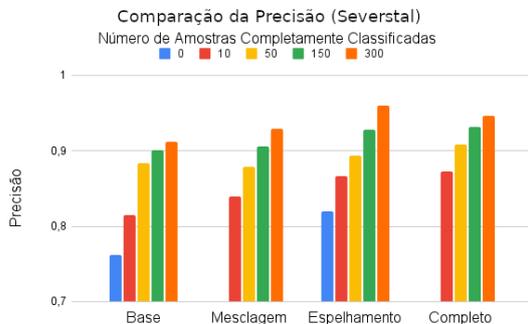
(b) AUC



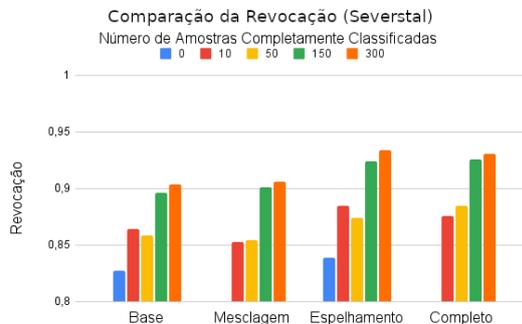
(c) Exatidão



(d) Precisão



(e) Revocação



Fonte: Autor

Tabela 14: Contagem de melhores métricas para cada tipo de treinamento - Severstal

Método	Contagem de melhores métricas
Base	0
Mesclagem	0
Espelhamento	14
Completo	11

Fonte: Autor

Nota: (1) A tabela inclui o caso fracamente supervisionado, em que só foram avaliados os métodos base e espelhamento

semelhante ao apresentado nos gráficos das métricas do banco de dados Severstal. A métrica de exatidão não apresentou melhoras aparentes, com exceção do caso com 0 amostras completamente classificadas, e as métricas de precisão e revocação, embora apresentem melhoras com o uso dos métodos de *data augmentation*, também apresentam quedas de desempenho em certos casos.

Comparando os 3 diferentes métodos com o treinamento base, o método de espelhamento apresentou uma melhora em 21 dos 25 casos, e os métodos de mesclagem e completo apresentaram melhora em 16 dos 20 casos. Novamente, ao comparar o método completo com o de espelhamento, percebeu-se resultados semelhantes, com o método completo apresentando métricas melhores em 13 dos 20 casos e apresentando o mesmo resultado em 2 casos. A Tabela 15 indica em quantos casos cada um dos métodos de treinamento obteve a melhor métrica.

Tabela 15: Contagem de melhores métricas para cada tipo de treinamento - KSDD2

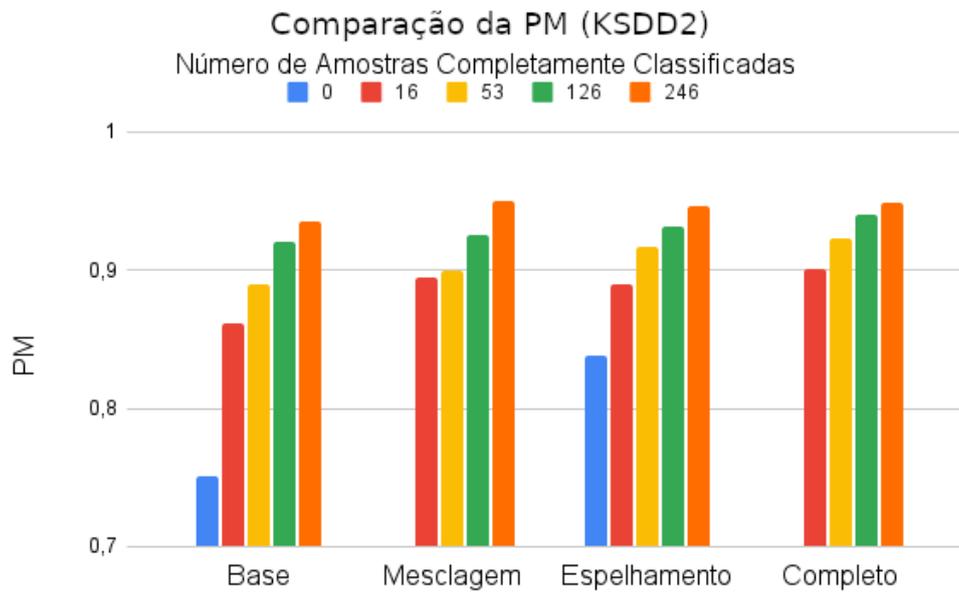
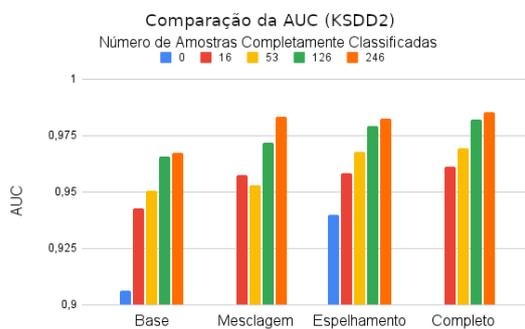
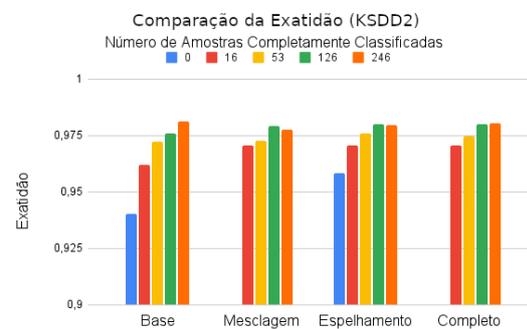
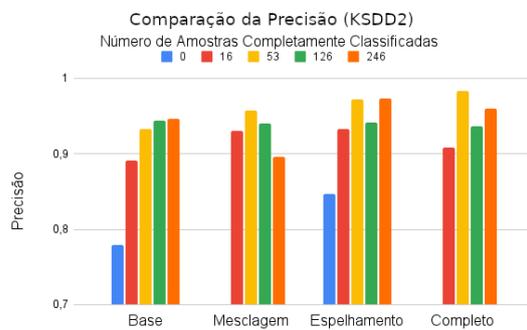
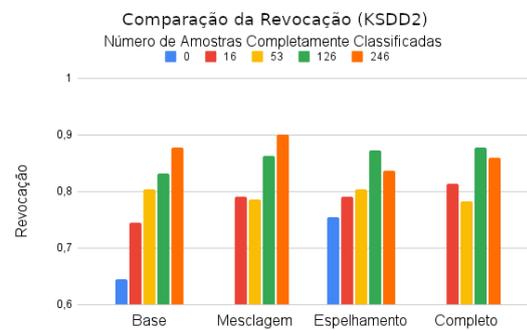
Método	Contagem de melhores métricas
Base	3
Mesclagem	3
Espelhamento	11
Completo	12

Fonte: Autor

Notas: (1) Caso dois métodos possuam o mesmo valor para uma métrica e este seja o maior valor, ambos são considerados como o melhor
 (2) A tabela inclui o caso fracamente supervisionado, em que só foram avaliados os métodos base e espelhamento

Ao avaliar os resultados de ambos os bancos de dados, percebe-se que o método de espelhamento apresentou melhoras significativas na maioria dos casos e o método de mesclagem obteve resultados mistos, em alguns casos piorando as métricas. Porém, em diversas situações, o uso de ambos os métodos de *data augmentation* em um mesmo treinamento apresentou as melhores métricas.

Esse comportamento pode ter se dado devido ao volume e à qualidade das imagens geradas: o método de espelhamento criou 3 novas imagens para cada amostra positiva do banco de dados, com ou sem máscara, e, como essas imagens não apresentam riscos

Figura 15: Gráficos de comparação para o banco de dados KSDD2**(a) Precisão Média****(b) AUC****(c) Exatidão****(d) Precisão****(e) Revocação**

Fonte: Autor

de geração de artefatos ou distorções, a qualidade delas pode ser considerada idêntica às originais. Já o método de mesclagem criou um volume menor de imagens sintéticas, pois gerou 3 imagens sintéticas para cada defeito das amostras negativas completamente classificadas, e, caso as imagens fonte e origem sejam muito diferentes ou a marcação manual do defeito estiver ruim, a qualidade da imagem sintética pode ser afetada. Quando os 2 métodos são utilizados simultaneamente, o maior número de amostras negativas sem risco de distorção reduzirá o efeito negativo de imagens sintéticas que não representam corretamente os defeitos que podem ser encontrados nas imagens reais.

Os resultados encontrados nos ensaios realizados condizem com outra publicação na área de *data augmentation* para identificação de defeitos superficiais: em Konovalenko et al. (2021), é realizado um treinamento de uma rede de classificação que varre trechos da imagem buscando por regiões com defeitos, utilizando o banco de dados Severstal para treinamento e validação. Como a entrada da rede é apenas um trecho da imagem, os autores realizaram três tipos de treinamento: um utilizando *data augmentation* mínimo, que apenas realiza o corte de trechos aleatórios da imagem, outro que utiliza a mesma técnica de espelhamentos utilizada neste trabalho para cada recorte de imagem, e outro que aplica operações de redimensionamento nas imagens recortadas e espelhadas. O modelo treinado com o *data augmentation* mínimo obteve uma exatidão de 0,9595, precisão de 0,9894 e revocação de 0,7692, enquanto o modelo treinado com métodos de *data augmentation* mais complexos apresentou uma exatidão de 0,9664 (maior), precisão de 0,9764 (menor) e revocação de 0,8212 (maior).

Para fins de comparação, também foram avaliadas publicações que implementam outros métodos de geração de dados sintéticos na área de detecção de defeitos: em Jain et al. (2020), é feita a comparação utilizando o banco de dados *Northeastern University Classification Dataset* entre uma rede treinada utilizando *data augmentation* clássico, que implementa rotações e transformações afins, e utilizando imagens sintéticas geradas por três arquiteturas diferentes de GANs. Enquanto o método clássico de *data augmentation* obteve uma exatidão de 0,9028, os treinamentos utilizando imagens geradas por GANs obtiveram exatidões de 0,9578, 0,9278 e 0,9486, demonstrando que o uso de redes neurais para geração de amostras artificiais é uma alternativa que pode ser analisada em futuros trabalhos.

Outra alternativa que demonstrou bons resultados na área de *data augmentation* para identificação de defeitos foi a criação de imagens sintéticas utilizando modelagem 3D, explorada em Boikov et al. (2021). Nesse trabalho, os autores utilizaram o software Blender para modelar e gerar um banco de dados semelhante ao Severstal Steel Defect Detection, que foi utilizado para validação dos resultados. O modelo treinado utilizando apenas dados sintéticos obteve um desempenho pior porém semelhante ao treinado com dados reais: obteve-se uma precisão de 0,81 e revocação de 0,89 para o treinamento com dados sintéticos, enquanto o treinamento com as amostras reais obteve uma precisão de 0,87 e revocação de 0,91.

5 CONCLUSÃO

Neste trabalho, foram propostos métodos de *data augmentation* para geração de imagens sintéticas a partir de um conjunto original de dados fracamente, parcialmente ou completamente classificados. Um dos métodos foi baseado no uso de espelhamentos de imagens e não necessita da informação da localização dos defeitos, enquanto o outro se baseou na criação de imagens sintéticas combinando defeitos cuja localização é conhecida com amostras negativas.

A validação dos métodos foi realizada por meio de uma série de treinamentos em dois bancos de dados diferentes, variando o número de amostras completamente classificadas disponíveis para treinamento e geração das imagens sintéticas. Essa validação demonstrou que o método de espelhamento apresentou bons resultados, principalmente no caso de treinamento fracamente supervisionado, e que o método de mesclagem não apresentou uma melhora significativa na maioria dos casos analisados. Também verificou-se que o uso simultâneo dos dois métodos gerou bons resultados.

Para trabalhos futuros, seria possível melhorar o desempenho do método de mesclagem realizando mais testes variando os parâmetros do método, buscando a geração de imagens mais próximas das reais ou utilizando outros métodos baseados em redes neurais para realizar a combinação dos defeitos segmentados com imagens sem defeitos. Outra possibilidade seria criar uma arquitetura de rede neural que permita a geração de defeitos sintéticos a partir de um banco de dados com diferentes tipos de anomalias e realizar a mesclagem com esses novos defeitos.

Outra proposta para futuros trabalhos seria a avaliação de outros métodos de geração de dados artificiais, como a utilização de GANs para geração de novas imagens sintéticas ou modelagem 3D das peças e defeitos utilizando softwares de modelagem como o Blender, dois métodos que apresentaram resultados positivos na literatura analisada.

APÊNDICE A - PSEUDOCÓDIGO: DATA AUGMENTATION POR ESPELHAMENTO

Algoritmo 1: Data augmentation por espelhamento

```
1: Entradas:  
2:   String pastaDefeitos  
3:   String pastaEspHorizontal  
4:   String pastaEspVertical  
5:   String pastaRot180  
6: Para imagem, mascara em pastaDefeitos Faça:  
7:   imagemVertical = espelhaV(imagem)  
8:   mascaraVertical = espelhaV(mascara)  
9:   imagemHorizontal = espelhaH(imagem)  
10:  mascaraHorizontal = espelhaH(mascara)  
11:  imagem180 = espelhaV(imagemHorizontal)  
12:  mascara180 = espelhaV(mascaraHorizontal)  
13:  salvarImagem(imagemVertical, pastaEspVertical)  
14:  salvarImagem(imagemHorizontal, pastaEspHorizontal)  
15:  salvarImagem(mascaraVertical, pastaEspVertical)  
16:  salvarImagem(mascaraHorizontal, pastaEspHorizontal)  
17:  salvarImagem(mascara180, pastaRot180)  
18:  salvarImagem(imagem180, pastaRot180)  
19: Fim
```

APÊNDICE B - PSEUDOCÓDIGO: DATA AUGMENTATION POR MESCLAGEM DE DEFEITOS

Algoritmo 2: Data augmentation por mesclagem de defeitos

```
1: Entradas:
2:   Inteiro defeitosPorImagem
3:   Inteiro numeroLoops
4:   String pastaRecorteDefeitos
5:   String pastaAmostrasNegativas
6: Inteiro contadorDefeitos = 0
7: imagemBase, mascaraBase = lerImagemAleatoria(pastaAmostrasNegativas)
8: Para loop 1,2, ... , numeroLoops Faça:
9:   pastaRecorteDefeitos = randomizar(pastaRecorteDefeitos)
10:  Para defeito, mascara em pastaRecorteDefeitos Faça:
11:    defeitoAug, mascaraAug = dataAugmentation(defeito, mascara)
12:    imagemBase, mascaraBase = adicionarDefeito(defeitoAug, mascaraAug)
13:    contadorDefeitos = contadorDefeitos + 1
14:    Se contadorDefeitos == defeitosPorImagem Faça:
15:      salvarImagem(imagemBase)
16:      salvarImagem(mascaraBase)
17:      imagemBase, mascaraBase = lerImagem(pastaAmostrasNegativas)
18:      contadorDefeitos = 0
19: Fim
```

APÊNDICE C - MÉTRICAS DOS TREINAMENTOS - SEVERSTAL

Tabela 16: Métricas do primeiro lote de treinamento - Severstal

Marcações Completas	Método	AUC	PM	Exatidão	Precisão	Revocação
0	Base	0,8601	0,8776	0,7667	0,7360	0,8317
0	Espelhamento	0,9062	0,9186	0,8283	0,8122	0,8542
10	Base	0,9053	0,9109	0,8108	0,7896	0,8475
10	Mesclagem	0,9227	0,9300	0,8463	0,8403	0,8550
10	Espelhamento	0,9416	0,9480	0,8654	0,8412	0,9008
10	Completo	0,9406	0,9492	0,8783	0,8828	0,8725
50	Base	0,9415	0,9465	0,8767	0,8890	0,8608
50	Mesclagem	0,9347	0,9462	0,8683	0,8864	0,8450
50	Espelhamento	0,9505	0,9575	0,8858	0,8985	0,8700
50	Completo	0,9580	0,9648	0,9008	0,9154	0,8833
150	Base	0,9605	0,9626	0,8975	0,9070	0,8858
150	Mesclagem	0,9616	0,9655	0,9071	0,9172	0,8950
150	Espelhamento	0,9705	0,9745	0,9183	0,9269	0,9083
150	Completo	0,9747	0,9792	0,9350	0,9500	0,9183
300	Base	0,9615	0,9677	0,9029	0,8986	0,9083
300	Mesclagem	0,9692	0,9745	0,9192	0,9192	0,9192
300	Espelhamento	0,9841	0,9859	0,9471	0,9543	0,9392
300	Completo	0,9785	0,9815	0,9392	0,9466	0,9308

Fonte: Autor

Tabela 17: Métricas do segundo lote de treinamento - Severstal

Marcações Completas	Método	AUC	PM	Exatidão	Precisão	Revocação
0	Base	0,8835	0,8976	0,8008	0,7883	0,8225
0	Espelhamento	0,9030	0,9096	0,8254	0,8268	0,8233
10	Base	0,9287	0,9360	0,8567	0,8397	0,8817
10	Mesclagem	0,9191	0,9269	0,8442	0,8396	0,8508
10	Espelhamento	0,9479	0,9555	0,8813	0,8914	0,8683
10	Completo	0,9412	0,9502	0,8688	0,8618	0,8783
50	Base	0,9364	0,9479	0,8688	0,8785	0,8558
50	Mesclagem	0,9359	0,9447	0,8683	0,8714	0,8642
50	Espelhamento	0,9475	0,9556	0,8838	0,8886	0,8775
50	Completo	0,9531	0,9601	0,8938	0,9001	0,8858
150	Base	0,9608	0,9633	0,9000	0,8954	0,9058
150	Mesclagem	0,9617	0,9665	0,9004	0,8948	0,9075
150	Espelhamento	0,9785	0,9812	0,9338	0,9284	0,9400
150	Completo	0,9741	0,9781	0,9217	0,9121	0,9333
300	Base	0,9675	0,9722	0,9133	0,9261	0,8983
300	Mesclagem	0,9693	0,9739	0,9171	0,9379	0,8933
300	Espelhamento	0,9841	0,9857	0,9483	0,9662	0,9292
300	Completo	0,9704	0,9781	0,9392	0,9466	0,9308

Fonte: Autor

APÊNDICE D - MÉTRICAS DOS TREINAMENTOS - KSDD2

Tabela 18: Métricas do primeiro lote de treinamento - KSDD2

Marcações Completas	Método	AUC	PM	Exatidão	Precisão	Revocação
0	Base	0,9057	0,7413	0,9343	0,7157	0,6636
0	Espelhamento	0,9436	0,8310	0,9542	0,8404	0,7182
16	Base	0,9362	0,8578	0,9612	0,8817	0,7455
16	Mesclagem	0,9575	0,8963	0,9721	0,9457	0,7909
16	Espelhamento	0,9519	0,8844	0,9711	0,9655	0,7636
16	Completo	0,9610	0,9058	0,9731	0,9192	0,8273
53	Base	0,9515	0,8933	0,9721	0,9556	0,7818
53	Mesclagem	0,9546	0,8991	0,9711	0,9765	0,7545
53	Espelhamento	0,9693	0,9215	0,9761	0,9674	0,8091
53	Completo	0,9708	0,9203	0,9721	0,9881	0,7545
126	Base	0,9668	0,9210	0,9761	0,9388	0,8364
126	Mesclagem	0,9674	0,9257	0,9781	0,9400	0,8545
126	Espelhamento	0,9791	0,9319	0,9821	0,9510	0,8818
126	Completo	0,9769	0,9397	0,9821	0,9510	0,8818
246	Base	0,9681	0,9330	0,9801	0,9500	0,8636
246	Mesclagem	0,9861	0,9502	0,9771	0,8991	0,8909
246	Espelhamento	0,9761	0,9431	0,9801	0,9688	0,8455
246	Completo	0,9894	0,9561	0,9821	0,9792	0,8545

Fonte: Autor

Tabela 19: Métricas do segundo lote de treinamento - KSDD2

Marcações Completas	Método	AUC	PM	Exatidão	Precisão	Revocação
0	Base	0,9072	0,7594	0,9462	0,8415	0,6273
0	Espelhamento	0,9361	0,8450	0,9622	0,8529	0,7909
16	Base	0,9497	0,8663	0,9631	0,9011	0,7455
16	Mesclagem	0,9576	0,8929	0,9691	0,9158	0,7909
16	Espelhamento	0,9651	0,8952	0,9701	0,9000	0,8182
16	Completo	0,9614	0,8960	0,9681	0,8980	0,8000
53	Base	0,9499	0,8870	0,9721	0,9100	0,8273
53	Mesclagem	0,9513	0,9010	0,9741	0,9375	0,8182
53	Espelhamento	0,9665	0,9118	0,9761	0,9778	0,8000
53	Completo	0,9679	0,9246	0,9771	0,9780	0,8091
126	Base	0,9650	0,9194	0,9761	0,9479	0,8273
126	Mesclagem	0,9765	0,9250	0,9801	0,9412	0,8727
126	Espelhamento	0,9796	0,9322	0,9781	0,9314	0,8636
126	Completo	0,9876	0,9405	0,9781	0,9231	0,8727
246	Base	0,9668	0,9385	0,9821	0,9423	0,8909
246	Mesclagem	0,9809	0,9489	0,9781	0,8929	0,9091
246	Espelhamento	0,9891	0,9492	0,9791	0,9785	0,8273
246	Completo	0,9816	0,9424	0,9791	0,9406	0,8636

Fonte: Autor

APÊNDICE E - ANOVA - SEVERSTAL

Tabela 20: Quadro ANOVA da AUC - Severstal

Origem	GL	SQ	QM	F_{cal}	F_{tab}
Amostra	3	0,0093972	0,0031324	121,52	3,24
Método	3	0,0021133	0,0007044	27,33	3,24
Interação	9	0,0003464	0,0000385	1,49	2,54
Resíduo	16	0,0004124	0,0000258		
Total	31	0,0122693			

Fonte: Autor

Tabela 21: Desdobramento da interação do data augmentation da AUC - Severstal

Causas da variação	GL	SQ	QM	F_{cal}	F_{tab}
Método/10	3	0,0011686	0,0003895	15,11	3,24
Método/50	3	0,0005151	0,0001717	6,66	3,24
Método/150	3	0,0003534	0,0001178	4,57	3,24
Método/300	3	0,0004225	0,0001408	5,46	3,24

Fonte: Autor

Tabela 22: Quadro ANOVA da Exatidão - Severstal

Origem	GL	SQ	QM	F_{cal}	F_{tab}
Amostra	3	0,0253169	0,0084390	85,86	3,24
Método	3	0,0062628	0,0020876	21,24	3,24
Interação	9	0,0005789	0,0000643	0,65	2,54
Resíduo	16	0,0015727	0,0000983		
Total	31	0,0337313			

Fonte: Autor

Tabela 23: Desdobramento da interação do data augmentation da Exatidão - Severstal

Causas da variação	GL	SQ	QM	F_{cal}	F_{tab}
Método/10	3	0,0024377	0,0008126	8,27	3,24
Método/50	3	0,0010176	0,0003392	3,45	3,24
Método/150	3	0,0013758	0,0004586	4,67	3,24
Método/300	3	0,0020107	0,0006702	6,82	3,24

Fonte: Autor

Tabela 24: Quadro ANOVA da Precisão - Severstal

Origem	GL	SQ	QM	F_{cal}	F_{tab}
Amostra	3	0,0350041	0,0116680	39,51	3,24
Método	3	0,0076570	0,0025523	8,64	3,24
Interação	9	0,0015015	0,0001668	0,56	2,54
Resíduo	16	0,0047250	0,0002953		
Total	31	0,0488877			

Fonte: Autor

Tabela 25: Desdobramento da interação do data augmentation da Precisão - Severstal

Causas da variação	GL	SQ	QM	F_{cal}	F_{tab}
Método/10	3	0,0042059	0,0014020	4,75	3,24
Método/50	3	0,0009704	0,0003235	1,10	3,24
Método/150	3	0,0013593	0,0004531	1,53	3,24
Método/300	3	0,0026230	0,0008743	2,96	3,24

Fonte: Autor

Tabela 26: Quadro ANOVA da Revocação - Severstal

Origem	GL	SQ	QM	F_{cal}	F_{tab}
Amostra	3	0,0175822	0,0058607	34,85	3,24
Método	3	0,0048268	0,0016089	9,57	3,24
Interação	9	0,0004465	0,0000496	0,30	2,54
Resíduo	16	0,0026906	0,0001682		
Total	31	0,0255462			

Fonte: Autor

Tabela 27: Desdobramento da interação do data augmentation da Revo-
cação - Severstal

Causas da variação	GL	SQ	QM	F_{cal}	F_{tab}
Método/10	3	0,0011233	0,0003744	2,23	3,24
Método/50	3	0,0011628	0,0003876	2,30	3,24
Método/150	3	0,0014322	0,0004774	2,84	3,24
Método/300	3	0,0015551	0,0005184	3,08	3,24

Fonte: Autor

APÊNDICE F - ANOVA - KSDD2

Tabela 28: Quadro ANOVA da AUC - KSDD2

Origem	GL	SQ	QM	F_{cal}	F_{tab}
Amostra	3	0,0032961	0,0010987	41,44	3,24
Método	3	0,0015034	0,0005011	18,90	3,24
Interação	9	0,0002131	0,0000237	0,89	2,54
Resíduo	16	0,0004242	0,0000265		
Total	31	0,0054369			

Fonte: Autor

Tabela 29: Desdobramento da interação do data augmentation da AUC - KSDD2

Causas da variação	GL	SQ	QM	F_{cal}	F_{tab}
Método/16	3	0,0004043	0,0001348	5,08	3,24
Método/53	3	0,0005691	0,0001897	7,15	3,24
Método/126	3	0,0003282	0,0001094	4,13	3,24
Método/246	3	0,0004150	0,0001383	5,22	3,24

Fonte: Autor

Tabela 30: Quadro ANOVA da Exatidão - KSDD2

Origem	GL	SQ	QM	F_{cal}	F_{tab}
Amostra	3	0,0006416	0,0002139	55,63	3,24
Método	3	0,0000727	0,0000242	6,30	3,24
Interação	9	0,0000908	0,0000101	2,62	2,54
Resíduo	16	0,0000615	0,0000038		
Total	31	0,0008666			

Fonte: Autor

Tabela 31: Desdobramento da interação do data augmentation da Exatidão - KSDD2

Causas da variação	GL	SQ	QM	F_{cal}	F_{tab}
Método/16	3	0,0001075	0,0000358	9,32	3,24
Método/53	3	0,0000203	0,0000068	1,76	3,24
Método/126	3	0,0000213	0,0000071	1,85	3,24
Método/246	3	0,0000143	0,0000048	1,24	3,24

Fonte: Autor

Tabela 32: Quadro ANOVA da Precisão - KSDD2

Origem	GL	SQ	QM	F_{cal}	F_{tab}
Amostra	3	0,0084255	0,0028085	7,05	3,24
Método	3	0,0039213	0,0013071	3,28	3,24
Interação	9	0,0081826	0,0009092	2,28	2,54
Resíduo	16	0,0063714	0,0003982		
Total	31	0,0269008			

Fonte: Autor

Tabela 33: Desdobramento da interação do data augmentation da Precisão - KSDD2

Causas da variação	GL	SQ	QM	F_{cal}	F_{tab}
Método/16	3	0,0023146	0,0007715	1,94	3,24
Método/53	3	0,0028658	0,0009553	2,40	3,24
Método/126	3	0,0000412	0,0000137	0,03	3,24
Método/246	3	0,0068823	0,0022941	5,76	3,24

Fonte: Autor

Tabela 34: Quadro ANOVA da Revocação - KSDD2

Origem	GL	SQ	QM	F_{cal}	F_{tab}
Amostra	3	0,0455165	0,0151722	31,93	3,24
Método	3	0,0020248	0,0006749	1,42	3,24
Interação	9	0,0106405	0,0011823	2,49	2,54
Resíduo	16	0,0076033	0,0004752		
Total	31	0,0657851			

Fonte: Autor

Tabela 35: Desdobramento da interação do data augmentation da Revo-
cação - KSDD2

Causas da variação	GL	SQ	QM	F_{cal}	F_{tab}
Método/16	3	0,0049070	0,0016357	3,44	3,24
Método/53	3	0,0008574	0,0002858	0,60	3,24
Método/126	3	0,0025207	0,0008402	1,77	3,24
Método/246	3	0,0043802	0,0014601	3,07	3,24

Fonte: Autor

REFERÊNCIAS

- ABIODUN, O. I. et al. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, v. 4, n. 11, e00938, 2018. ISSN 2405-8440. DOI: <https://doi.org/10.1016/j.heliyon.2018.e00938>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2405844018332067>>.
- BOIKOV, A. et al. Synthetic Data Generation for Steel Defect Detection and Classification Using Deep Learning. *Symmetry*, v. 13, n. 7, 2021. ISSN 2073-8994. DOI: 10.3390/sym13071176. Disponível em: <<https://www.mdpi.com/2073-8994/13/7/1176>>.
- BOWLES, C. et al. GAN Augmentation: Augmenting Training Data using Generative Adversarial Networks, 2018. arXiv preprint at arXiv: 1810.10863.
- BOŽIČ, J.; TABERNIK, D.; SKOČAJ, D. *Mixed supervision for surface-defect detection: from weakly to fully supervised learning*. [S.l.: s.n.], abr. 2021. Elsevier. Disponível em: <<https://arxiv.org/abs/2104.06064>>.
- BRADSKI, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- CHEN, Y. et al. Surface Defect Detection Methods for Industrial Products: A Review. *Applied Sciences*, v. 11, n. 16, 2021. ISSN 2076-3417. DOI: 10.3390/app11167657. Disponível em: <<https://www.mdpi.com/2076-3417/11/16/7657>>.
- DENG, J. et al. ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. [S.l.: s.n.], 2009. p. 248–255. DOI: 10.1109/CVPR.2009.5206848.
- GONZALES, R. C.; WOODS, R. E. Digital Image Processing. In: New York: Pearson, 2018a. Geometric Transformations, p. 65–70. ISBN 10: 1-292-22304-9.
- GONZALES, R. C.; WOODS, R. E. Digital Image Processing. In: New York: Pearson, 2018b. Geometric Transformations, p. 100–106. ISBN 10: 1-292-22304-9.
- HIDAKA, A.; KURITA, T. Consecutive Dimensionality Reduction by Canonical Correlation Analysis for Visualization of Convolutional Neural Networks. In: v. 2017, p. 160–167. DOI: 10.5687/sss.2017.160.
- JAIN, S. et al. Synthetic data augmentation for surface defect detection and classification using deep learning. *Journal of Intelligent Manufacturing*, Springer Science e Business Media LLC, nov. 2020. DOI: 10.1007/s10845-020-01710-x. Disponível em: <<https://doi.org/10.1007/s10845-020-01710-x>>.
- JUNG, A. B. *imgaug*. [S.l.: s.n.], 2018. <https://github.com/aleju/imgaug>. [Online; accessed 30-Oct-2018].

- KANG, B. A Review on Image and Video Processing. *International Journal of Multimedia and Ubiquitous Engineering*, v. 2, mai. 2007.
- KONOVALENKO, I. et al. Recognition of Scratches and Abrasions on Metal Surfaces Using a Classifier Based on a Convolutional Neural Network. *Metals*, v. 11, n. 4, 2021. ISSN 2075-4701. DOI: 10.3390/met11040549. Disponível em: <<https://www.mdpi.com/2075-4701/11/4/549>>.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet Classification with Deep Convolutional Neural Networks. In: PEREIRA, F. et al. (Ed.). *Advances in Neural Information Processing Systems 25*. [S.l.]: Curran Associates, Inc., 2012. p. 1097–1105. Disponível em: <<http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>>.
- LI, J. et al. Brief Introduction of Back Propagation (BP) Neural Network Algorithm and Its Improvement. In: JIN, D.; LIN, S. (Ed.). *Advances in Computer Science and Information Engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 553–558.
- LI, Z. et al. *A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects*. [S.l.: s.n.], 2020. arXiv: 2004.02806 [cs.CV].
- MAIND, S. B.; WANKAR, P. Research Paper on Basic of Artificial Neural Network. *International Journal on Recent and Innovation Trends in Computing and Communication*, v. 2, n. 1, p. 96–100, jan. 2014.
- MARTINS, L.; PÁDUA, F.; ALMEIDA, P. Automatic Detection of Surface Defects on Rolled Steel Using Computer Vision and Artificial Neural Networks. In: p. 1081–1086. DOI: 10.1109/IECON.2010.5675519.
- MIKOŁAJCZYK, A.; GROCHOWSK, M. Data augmentation for improving deep learning in image classification problem. In: PROCEEDINGS of the 2018 International Interdisciplinary PhD Workshop (IIPhDW). [S.l.: s.n.], 2018. p. 1–2.
- PEREZ, F.; VASCONCELOS, C. et al. Data Augmentation for Skin Lesion Analysis. *OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis*, Springer International Publishing, p. 303–311, 2018. ISSN 1611-3349. DOI: 10.1007/978-3-030-01201-4_33. Disponível em: <http://dx.doi.org/10.1007/978-3-030-01201-4_33>.
- PEREZ, L.; WANG, J. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *CoRR*, abs/1712.04621, 2017. arXiv: 1712.04621. Disponível em: <<http://arxiv.org/abs/1712.04621>>.
- PÉREZ, P.; GANGNET, M.; BLAKE, A. Poisson image editing. *ACM SIGGRAPH 2003 Papers*, 2003.
- RIBEIRO, J. L. D.; CATEN, C. S. TEN. *Série Monográfica Qualidade, Projeto de Experimentos*. Porto Alegre: [s.n.], abr. 2011. Universidade Federal do Rio Grande do Sul. Disponível em: <http://www.producao.ufrgs.br/arquivos/disciplinas/117_apostila_pe_2011.pdf>.
- RIPLEY, B. D. Pattern Recognition and Neural Networks. In: Cambridge: Cambridge University Press, 1996. p. 354. ISBN 10.1017/CBO9780511812651.
- SHORTEN, C.; KHOSHGOFTAAR, T. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, v. 6, n. 60, p. 2–5, jul. 2019.

TABERNIK, D. et al. Segmentation-based deep-learning approach for surface-defect detection. *Journal of Intelligent Manufacturing*, Springer Science e Business Media LLC, v. 31, n. 3, p. 759–776, mai. 2019. ISSN 1572-8145. DOI: 10.1007/s10845-019-01476-x. Disponível em: <<http://dx.doi.org/10.1007/s10845-019-01476-x>>.

TULBURE, A.-A.; TULBURE, A.-A.; DULF, E.-H. A review on modern defect detection models using DCNNs – Deep convolutional neural networks. *Journal of Advanced Research*, 2021. ISSN 2090-1232. DOI: <https://doi.org/10.1016/j.jare.2021.03.015>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2090123221000643>>.