

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

VINICIOS RODRIGUES

**CLASSIFICAÇÃO DE DEFEITOS SUPERFICIAIS EM PLACAS DE AÇO
LAMINADAS A QUENTE COM ASSISTÊNCIA DE UMA REDE NEURAL**

PATO BRANCO

2023

VINICIOS RODRIGUES

**CLASSIFICAÇÃO DE DEFEITOS SUPERFICIAIS EM PLACAS DE AÇO
LAMINADAS A QUENTE COM ASSISTÊNCIA DE UMA REDE NEURAL**

**Classification of a superficial defect in hot rolled steel with assistance of a
neural network**

Trabalho de conclusão de curso de graduação apresentada como requisito para obtenção do título de Bacharel em Engenharia Mecânica da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador(a): Bruno Bellini Medeiros

PATO BRANCO

2023



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

VINICIOS RODRIGUES

**CLASSIFICAÇÃO DE DEFEITOS SUPERFICIAIS EM PLACAS DE AÇO
LAMINADAS A QUENTE COM ASSISTÊNCIA DE UMA REDE NEURAL**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do título de
Bacharel em Engenharia Mecânica na Universidade
Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 01/12/2023

Fabio Rodrigo Mandello Rodrigues
Doutorado
Universidade Tecnológica Federal do Paraná

Silvana Patricia Verona
Doutorado
Universidade Tecnológica Federal do Paraná

Bruno Bellini Medeiros
Doutorado
Universidade Tecnológica Federal do Paraná

PATO BRANCO

2023

Dedico este trabalho a meus amigos, pelos momentos
de sorriso durante meu curso.

AGRADECIMENTOS

Primeiramente quero agradecer aos amigos que fiz durante o curso, com certeza não teria conseguido tal ato sem a ajuda deles, posso dizer que fazem parte da minha família, pois por 7 anos foram a família que possuí, além de me proporcionarem muitos momentos de alegria durante o curso.

Agradecer ao meu irmão Breno Rodrigues, o qual me orientou durante duros percursos da minha vida, sendo sempre um alicerce de conhecimento e conduta.

Agradecer ao professor Bruno Bellini Medeiros, o qual me ajudou durante todo o período da faculdade, bem como me ajudou no meu crescimento pessoal, sendo sempre um professor atencioso com a vida pessoal do aluno.

Em seguida agradecer a minha noiva Mariane Tibes, a qual me ajudou durante o desenvolvimento deste trabalho, tendo um grande impacto na minha vida e me mostrando as cores do mundo.

E por último fazer um auto agradecimento por ter conquistado tal amplitude de conhecimento, sempre tendo dificuldades, mas me levantando e seguindo em frente até a chegada desse momento.

RESUMO

Este trabalho explora o uso de redes neurais para identificação de falhas superficiais em placas laminadas a quente e discute-se primeiramente a importância da detecção de falhas no setor industrial enfatizando sua influência sobre o padrão de qualidade dos produtos e integridade estrutural. Neste texto estão reunidos alguns dos tópicos mais importantes em relação as redes neurais incluindo informações sobre sua função, tipo e modo de aprendizagem. No presente contexto mencionam-se as principais formas das redes neurais que se destacam por serem mais adequadas ao processamento dos dados em forma visual através do uso das *convolutional neural networ* (CNNs). Além disso, no desenrolar deste estudo adotou-se uma metodologia minuciosa que abrangeu desde a preparação inicial dos dados até o ajuste fino das características relevantes utilizadas para capacitar o modelo. Há ênfase no uso de transferência de aprendizado, que consiste em aproveitar conhecimentos prévios de redes neurais treinadas em grandes conjuntos de dados genéricos. São apresentados estudos anteriores que utilizaram redes neurais para a detecção de defeitos em placas de aço, demonstrando a eficácia dessa abordagem. Além disso, são discutidos os principais avanços, desafios e limitações enfrentados nessa área de pesquisa. Por fim, são abordados os tipos de defeitos superficiais comumente encontrados em placas de aço laminado a quente, como fissuras, inclusões, manchas, buracos, desgaste por laminação e riscos. Também são mencionadas pesquisas anteriores relacionadas a esses tipos de defeitos. O trabalho oferece uma visão abrangente sobre a detecção de defeitos superficiais em placas de aço utilizando redes neurais, discutindo desde os conceitos básicos até a aplicação prática e os desafios enfrentados nessa área de estudo.

Palavras-chave: defeitos em lâminas de aço; defeitos superficiais; rede neural; detecção.

ABSTRACT

This work addresses the use of neural networks for the detection of surface defects in hot-rolled steel plates. Initially, the importance of defect detection in the industry is discussed, highlighting its impacts on product quality and structural safety. Next, the basic concepts of neural networks are presented, including their structure, functioning, and learning from data. The main types of neural networks used in this context are mentioned, with a focus on convolutional neural networks (CNNs), which are suitable for processing image data. The methodology employed in the work is detailed, covering data preprocessing, selection of relevant features, and model training. The use of transfer learning is highlighted, which involves leveraging prior knowledge from neural networks trained on large generic datasets. Previous studies that used neural networks for defect detection in steel plates are presented, demonstrating the effectiveness of this approach. Additionally, the main advancements, challenges, and limitations in this research area are discussed. Finally, the commonly encountered types of surface defects in hot-rolled steel plates, such as crazing, inclusion, patches, pitted, rolled, and scratches, are addressed. Previous research related to these types of defects is also mentioned. Overall, the work provides a comprehensive overview of the detection of surface defects in steel plates using neural networks, discussing everything from basic concepts to practical application and the challenges faced in this field of study.

Keywords: steel plate defects detection; surface defects; neural network; detection.

LISTA DE ILUSTRAÇÕES

Figura 1 - Sobreposto	18
Figura 2 - Subdecapado.....	18
Figura 3 - Superdecapado	19
Figura 4 - Vergão	20
Figura 5 - Verruga.....	20
Figura 6 - Emenda defeituosa	21
Figura 7 - Escama.....	21
Figura 8 - Esfoliação	22
Figura 9 - Defeito esponjoso	22
Figura 10 - Ferrugem.....	23
Figura 11 - Casca de laranja	23
Figura 12 - Costela	24
Figura 13 - Depósito carbonoso.....	24
Figura 14 - Gota fria	25
Figura 15 - Linhas brancas.....	25
Figura 16 - Linhas de distensão.....	26
Figura 17 - Linhas pretas	27
Figura 18 - Marca de arame	27
Figura 19 - Marca de areia	28
Figura 20 - Marca de colamento.....	29
Figura 21 - Marca de deslizamento	30
Figura 22 - Marca de prego.....	30

Figura 23 - Marca de retífica	31
Figura 24 - Marca de rolo	32
Figura 25 - Marca de sujeira	32
Figura 26 - Marca de topo quente	33
Figura 27 - Marca transversal.....	34
Figura 28 - Quebra de bordas.....	34
Figura 29 - Quebra de selamento.....	35
Figura 30 - Quebra de superfície.....	35
Figura 31 - Refratário incrustado	36
Figura 32 - Ruga de laminação.....	36
Figura 33 - Trincas	37
Figura 34 - Inclusões.....	38
Figura 35 - Manchas d'água.....	39
Figura 36 - Mancha de recozimento.....	40
Figura 37 - Mancha.....	40
Figura 38 - Carepas	41
Figura 39 - Cavidade de Carepas	42
Figura 40 - Carepa laminada.....	43
Figura 41 - Riscos	44
Figura 42 - Modelo de Neurônio Artificial.....	45
Figura 43 - Arquitetura da Rede Neural	47
Figura 44 - Rede neural feedforward	48
Figura 45 - Rede neural feedback	48

Figura 46 - Pooling	49
Figura 47 - Aprendizado Convolucional	50
Figura 48 - Rede Convolucional.....	51
Figura 49 - Vencedor visual recognition challenge 2014	52
Figura 50 - Redução de ruído	54
Figura 51 - Fluxograma de etapas	61
Figura 52 - Base de dados	63
Figura 53 - Importando bibliotecas da rede neural	65
Figura 54 - Geração aleatória	65
Figura 55 - Diretórios	65
Figura 56 - Hyper parâmetros.....	66
Figura 57 - Melhoramento de imagens	66
Figura 58 - Base de dados para treinamento	67
Figura 59 - VGG16	67
Figura 60 - Congelamento de camadas	68
Figura 61 – Camada de classificação	69
Figura 62 - Compilação	70
Figura 63 - Treinamento	70
Figura 64 - Salvando o modelo	70
Figura 65 - Critérios de validação	71
Figura 66 - Validação	72
Figura 67 - Importando bibliotecas para teste	73
Figura 68 - Diretórios de teste e modelo	73

Figura 69 - Carregando modelo	73
Figura 70 - Definição de classes	74
Figura 71 - Variáveis da matriz de confusão e rótulos.....	74
Figura 72 - Iteração na base de dados de teste	75
Figura 73 - Plotagem da matriz confusão.....	76
Figura 74 - Plotagem F1-score	76
Figura 75 - Acurácia da validação.....	77
Figura 76 - Matriz confusão	78
Figura 77 - F1-score	78
Figura 78 - Curva ROC	79

SUMÁRIO

1	INTRODUÇÃO.....	12
1.1	Justificativa.....	13
1.2	Objetivo geral	13
1.3	Objetivos específicos	13
2	REVISÃO BIBLIOGRÁFICA	15
2.1	Placas de aço laminadas a quente	15
2.2	Fabricação do aço laminado a quente	15
2.3	Defeitos superficiais em placas de aço laminadas a quente	16
2.3.1	Sobreposto.....	17
2.3.2	Subdecapado	18
2.3.3	Superdecapado.....	18
2.3.4	Trinca térmica	19
2.3.5	Vergão.....	19
2.3.6	Verruga.....	20
2.3.7	Emenda defeituosa.....	20
2.3.8	Escama	21
2.3.9	Esfoliação.....	21

2.3.10	Defeito esponjoso	22
2.3.11	Ferrugem	22
2.3.12	Casca de laranja.....	23
2.3.13	Costela	23
2.3.14	Depósito carbonoso	24
2.3.15	Gota fria.....	24
2.3.16	Linhas brancas.....	25
2.3.17	Linhas de distensão	25
2.3.18	Linhas pretas	26
2.3.19	Marca de arame.....	27
2.3.20	Marca de areia	27
2.3.21	Marca de colamento	28
2.3.22	Marca de deslizamento	29
2.3.23	Marca de prego.....	30
2.3.24	Marca de retífica	31
2.3.25	Marca de rolo.....	31
2.3.26	Marca de sujeira	32
2.3.27	Marca de topo quente	32
2.3.28	Marca transversal	33
2.3.29	Quebra de bordas	34
2.3.30	Quebra de selamento	34
2.3.31	Quebra de superfície	35
2.3.32	Refratário incrustado.....	35

2.3.33	Ruga de laminação	36
2.3.34	Trincas	36
2.3.35	Inclusões	38
2.3.36	Manchas.....	39
<u>2.3.36.1</u>	<u>Mancha d'água.....</u>	<u>39</u>
<u>2.3.36.2</u>	<u>Mancha de recozimento</u>	<u>39</u>
2.3.37	Carepas.....	41
<u>2.3.37.1</u>	<u>Sal e Pimenta</u>	<u>42</u>
<u>2.3.37.2</u>	<u>Carepa laminada</u>	<u>42</u>
2.3.38	Riscos.....	43
2.4	Redes neurais	44
2.4.1	Neurônio Artificial	45
2.4.2	Características de redes neurais	45
2.4.3	Arquitetura de redes neurais	46
2.4.4	Tipos de redes neurais	47
2.4.5	Arquitetura das redes neurais	48
<u>2.4.5.1</u>	<u>VGG16</u>	<u>51</u>
2.4.6	Pré-processamento	53
2.4.7	Pós-processamento.....	53
2.4.8	Algoritmos de aprendizado.....	55
<u>2.4.8.1</u>	<u>Função de perda</u>	<u>56</u>
<u>2.4.8.2</u>	<u>Otimização</u>	<u>57</u>
<u>2.4.8.3</u>	<u>Transfer learning</u>	<u>57</u>

2.4.8.4	<u>Validação.....</u>	<u>58</u>
2.4.9	Utilização de redes neurais para detecção de defeitos:	59
3	METODOLOGIA	60
3.1	Coleta de dados	62
3.2	Desenvolvimento.....	63
3.2.1	Desenvolvimento dos algoritmos.....	63
3.2.2	Teste da rede neural treinada	72
4	RESULTADOS E DISCUSSÕES	76
5	CONCLUSÃO	80
	REFERÊNCIAS.....	81

1 INTRODUÇÃO

Recentemente muitos métodos de aumento de eficiência têm sido implementados para reduzir custos e aprimorar a qualidade de placas de aço garantindo a qualidade dos produtos e a segurança da utilização das placas de aço, sendo minimamente automatizados e em sua maioria com inspeções manuais. (Ghorai, et al, 2012).

A detecção de defeitos superficiais em placas de aço desempenha um papel crucial na indústria. É essencial para garantir a qualidade dos produtos que a detecção dos defeitos superficiais em placas de aço seja precisa e eficiente, evitando falhas estruturais e custos indevidos, que se não detectados a tempo, podem levar a problemas graves, como a formação de trincas e a redução da vida útil das estruturas metálicas. Portanto, a detecção precoce e precisa de defeitos, é fundamental para evitar perdas financeiras e riscos potenciais de acidentes e falhas em estruturas metálicas. (Ghorai, et al, 2012).

A variedade dos defeitos em placas de aço laminadas a quente é grande, muitos destes defeitos possuem sub grupos de classificação, resultando em um desafio maior para diferenciar defeitos. Juntamente com a falta de padrão de tipos de defeitos, a caracterização e classificação variam de uma planta industrial para outra e de operador para operador, assim como a diferença de variáveis durante o processo. Sendo desejável métodos automáticos para inspeção. (Ashour et al, 2019).

Entre as várias técnicas automatizadas de detecção de defeitos superficiais, vem se destacando as redes neurais artificiais convolucionais. A utilização de redes neurais também permite a aplicação de técnicas de *transfer learning*, o que significa aproveitar conhecimentos prévios de modelos de redes neurais treinados em conjuntos de dados genéricos. Isso reduz a necessidade de treinar a rede do zero em um conjunto de dados específico e acelera o processo de desenvolvimento de um sistema de detecção de defeitos.

1.1 Justificativa

A detecção de defeitos superficiais em placas de aço laminado a quente é de suma importância para a indústria siderúrgica. A localização manual de falhas é uma tarefa demorada, também propensa a erros humanos. Adicionalmente, a verificação convencional por meio da examinação visual muitas vezes não é capaz de observar erros nos estágios iniciais do processo produtivo, podendo levar a dificuldades futuras, como perda de materiais e custos indevidos. Cada vez mais é preciso recorrer a técnicas automatizadas e confiáveis para detectar defeitos.

Utilizar redes neurais nesse ambiente é altamente benéfico, elas possuem a habilidade de aprender padrões complexos e as *convolutional neural network* (CNNs) ou rede neurais convolucionais, se destacam pela eficiência no processamento de imagens. A capacidade das redes em identificar e classificar corretamente diversos tipos de defeitos superficiais em placas de aço melhora ao passo que seus bancos de dados são alimentados com informações.

Com um sistema automatizado de detecção de defeitos baseado em redes neurais, espera-se melhorar significativamente a eficiência e a precisão dos processos de controle de qualidade na indústria. Isso resultará em produtos de melhor qualidade, redução de desperdício de material e custos associados a retrabalhos. Além disso, a detecção precoce de defeitos permitirá a implementação de medidas corretivas de forma imediata.

1.2 Objetivo geral

Este projeto tem como objetivo principal desenvolver e analisar uma rede neural artificial capaz de detectar defeitos superficiais em placas metálicas laminadas a quente.

1.3 Objetivos específicos

Para atingir a finalidade primária os seguintes objetivos específicos devem ser alcançados:

- Identificar os defeitos em placas de aço laminadas a quente.
- Pesquisar e selecionar uma base de dados de imagens de defeitos superficiais
- Desenvolver um algoritmo em *python* de uma rede neural artificial para detecção de defeitos superficiais.
- Otimizar parâmetros, validar e determinar a acurácia da rede neural artificial desenvolvida.

2 REVISÃO BIBLIOGRÁFICA

A revisão bibliográfica é um alicerce teórico para o entendimento de um trabalho como esse, além de fornecer o embasamento científico necessário para o pesquisador, por meio desta, poderemos entender a utilização de redes neurais em processos industriais, visando a diminuição de erros humanos e a eficácia do procedimento, mantendo a qualidade do produto e diminuindo o tempo relativo.

2.1 Placas de aço laminadas a quente

A produção de placas de aço laminado a quente é um processo essencial na indústria siderúrgica, fornecendo materiais utilizados em diversas aplicações, como construção civil, indústria automotiva e indústria naval. Ao longo dos anos, muitos estudos têm sido realizados para compreender e melhorar a qualidade das placas de aço laminado a quente, bem como identificar e resolver os problemas associados a defeitos superficiais.

Laminação é o método mais comum da deformação plástica do aço de sua condição de fundição. A deformação quebra a microestrutura fundida, produzindo um aço no formato utilizável e melhorar as propriedades mecânicas. As propriedades do aço laminado dependem do quanto o material foi deformado em estágios diferentes do processo de laminação, além da composição química do aço. (Tamura et al, 1988).

A propriedade mecânica do produto é uma característica de preocupação na indústria de laminação. Dependendo deste fator, produtos laminados são desviados da linha de produção. Ao se pensar em produtividade é importante que os especialistas entendam a razão que qualquer produto foi rejeitado, bem como seu motivo, dessa forma, utilizando os dados para melhorar sua produtividade e minimizar os erros na produção. (Das et al, 2006).

2.2 Fabricação do aço laminado a quente

A fabricação de aço laminado a quente é um processo crucial na indústria siderúrgica, onde lingotes de aço são aquecidos a altas temperaturas e posteriormente laminados para obter produtos acabados com propriedades

mecânicas desejadas. Esse processo envolve várias etapas, desde a preparação da matéria-prima até a obtenção do produto final laminado.

A primeira etapa do processo de fabricação de aço laminado a quente é a preparação da matéria-prima, que geralmente envolve a fusão de minério de ferro em um alto-forno para produzir o ferro-gusa. Em seguida, o ferro-gusa é refinado para reduzir o teor de impurezas e ajustar a composição química do aço. Esse processo de refino pode ser realizado por meio de conversores a oxigênio, fornos elétricos a arco ou outros métodos. (Roberts, 1983).

Após a preparação da matéria-prima, o aço é aquecido a altas temperaturas em um forno. Esse aquecimento é essencial para amolecer o aço e torná-lo maleável para o processo de laminação subsequente. Durante o processo de aquecimento, o aço pode passar por tratamentos térmicos específicos para melhorar suas propriedades mecânicas. (Roberts, 1983).

A etapa seguinte é a laminação a quente, em que o aço aquecido é passado por cilindros de laminação para reduzir sua espessura e dar forma ao produto final. Esse processo de laminação é realizado em temperaturas elevadas para facilitar a deformação plástica do aço. Durante a laminação, o aço pode passar por várias passagens para atingir a espessura desejada e obter propriedades mecânicas específicas. (Roberts, 1983).

Após a laminação, o aço laminado a quente pode ser submetido a tratamentos adicionais, como recozimento para aliviar tensões e melhorar a ductilidade, bem como revestimentos de proteção para prevenir a corrosão. Esses tratamentos são importantes para garantir a qualidade do produto final e atender aos requisitos específicos do mercado. (Roberts, 1983).

2.3 Defeitos superficiais em placas de aço laminadas a quente

Como mencionado na justificativa a variedade dos defeitos em placas de aço laminadas a quente é grande, muitos destes defeitos possuem sub grupos de classificação, resultando em um desafio maior para diferenciar defeitos. Juntamente com a falta de padrão de tipos de defeitos, a caracterização e classificação variam de

uma planta industrial para outra e de operador para operador, assim como a diferença de variáveis durante o processo. (Ashour et al, 2019).

Mesmo com muitos pesquisadores coletando imagens de defeitos em placas de aço, não está disponível grande quantidade de dados e muitas empresas não permitem o acesso a sua base de dados. Até a presente data, a base de dados mais citada para defeitos em chapa de aço a quente é a NEU-DET, ou base de dados de defeitos superficiais em placas de aço laminadas a quente da *Northeaster Univeristy*. (Tang, 2023).

Utilizou-se a NBR 16552 produtos laminados, a fim de caracterizar os defeitos de superfície de acordo com a norma brasileira, com a intenção de relacionar os defeitos superficiais em chapas de aço laminadas a quente fornecida pela base de dados de NEU, com a NBR 16552 da Associação Brasileira de Normas Técnicas, ou ABNT.

2.3.1 Sobreposto

“Defeito de superfície causado pela laminação de material com alguma protuberância, ou pelo fechamento de cavidades superficiais durante a laminação afetando apenas uma face”, Figura 1. (ABNT NBR 16552, 2016).

Figura 1 - Sobreposto

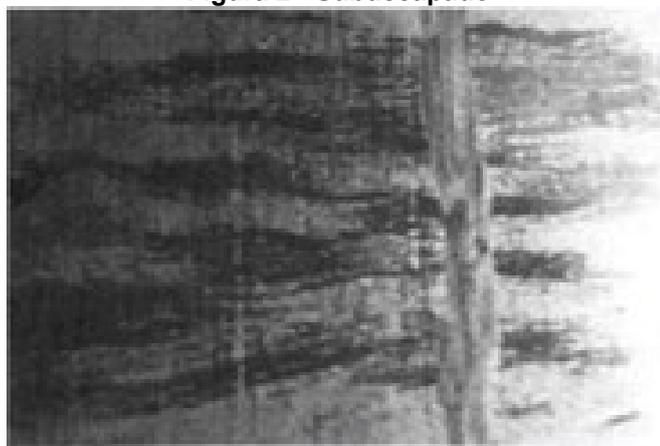


Fonte: NBR 16552 (2016)

2.3.2 Subdecapado

“Defeito de superfície causado pela não eliminação total de óxido natural de laminação a quente pela decapagem”, Figura 2. (ABNT NBR 16552, 2016).

Figura 2 - Subdecapado

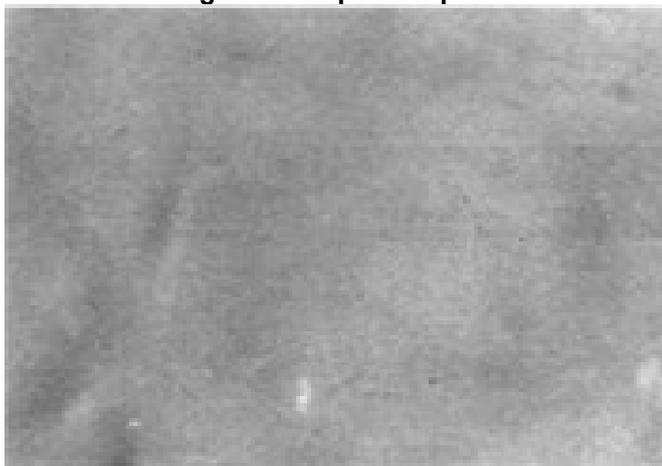


Fonte: NBR 16552 (2016)

2.3.3 Superdecapado

“Defeito de superfície causado pela existência na superfície áspera e corroída, causada por ataque excessivo ao aço durante a decapagem”, Figura 3. (ABNT NBR 16552, 2016).

Figura 3 - Superdecapado



Fonte: NBR 16552 (2016)

2.3.4 Trinca térmica

“Defeito de superfície causado por fissuras nos cilindros de laminação a quente; apresenta-se como um alto relevo de forma rendilhada, distribuído aleatoriamente na superfície do material”. (ABNT NBR 16552, 2016).

2.3.5 Vergão

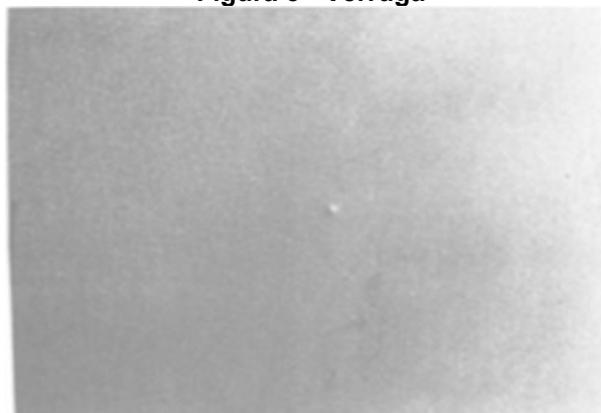
“Defeito de superfície caracterizado pela acomodação de leves ondulações durante a laminação, não chegando a configurar rugas de laminação”, Figura 4. (ABNT NBR 16552, 2016).

Figura 4 - Vergão

Fonte: NBR 16552 (2016)

2.3.6 Verruga

“Defeito de superfície caracterizado pela presença de um ou mais pontos salientes originados durante a laminação por depressões nos cilindros, provocado por jateamento inadequado”, Figura 5. (ABNT NBR 16552, 2016).

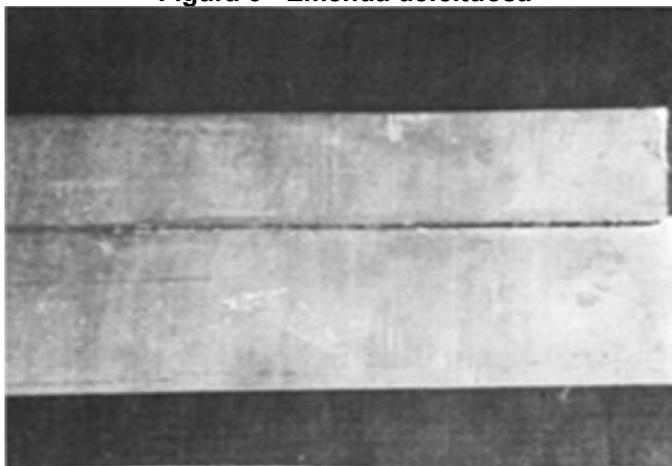
Figura 5 - Verruga

Fonte: NBR 16552 (2016)

2.3.7 Emenda defeituosa

“Defeito de superfície caracterizado pela existência de emenda por solda no material”, Figura 6. (ABNT NBR 16552, 2016).

Figura 6 - Emenda defeituosa

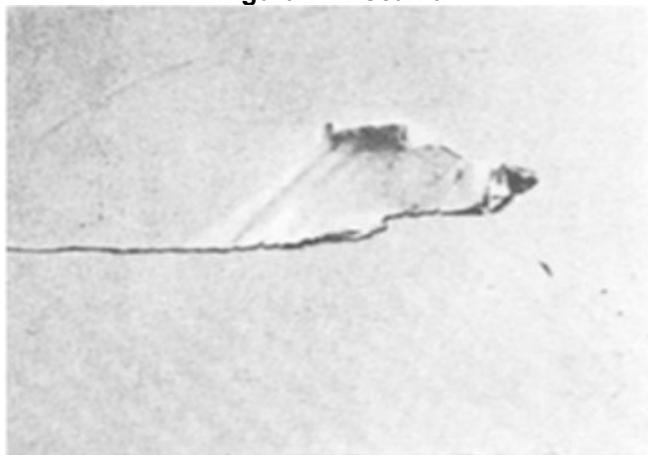


Fonte: NBR 16552 (2016)

2.3.8 Escama

“Defeito de superfície caracterizado pela presença de finas lâminas de aço, oxidadas ou não, com uma das extremidades fixas no aço-base e o restante levemente aderente”, Figura 7. (ABNT NBR 16552, 2016).

Figura 7 - Escama

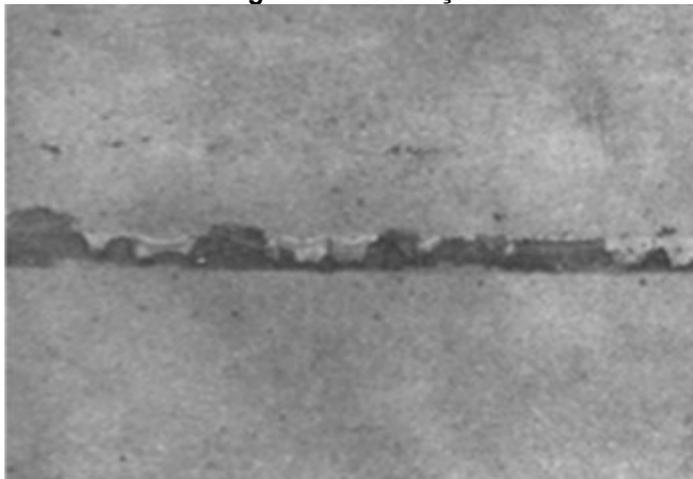


Fonte: NBR 16552 (2016)

2.3.9 Esfoliação

“Defeito de superfície caracterizado pela existência de uma película de aço parcialmente aderente ao material que por ação mecânica se destaca”, Figura 8. (ABNT NBR 16552, 2016).

Figura 8 - Esfoliação

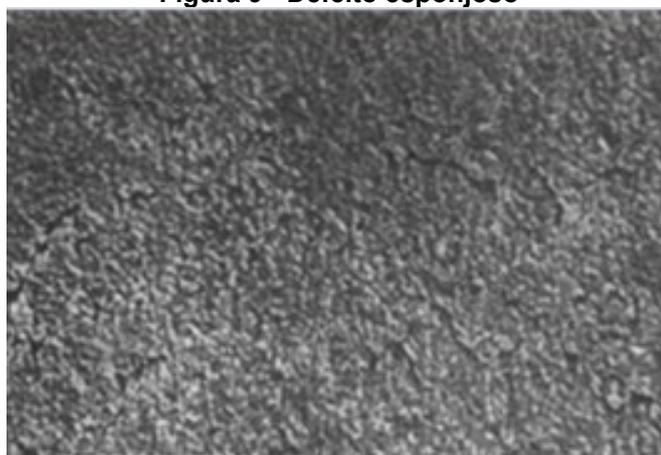


Fonte: NBR 16552 (2016)

2.3.10 Defeito esponjoso

“Defeito de superfície caracterizado por um fendilhamento de aspecto irregular”, Figura 9. (ABNT NBR 16552, 2016).

Figura 9 - Defeito esponjoso

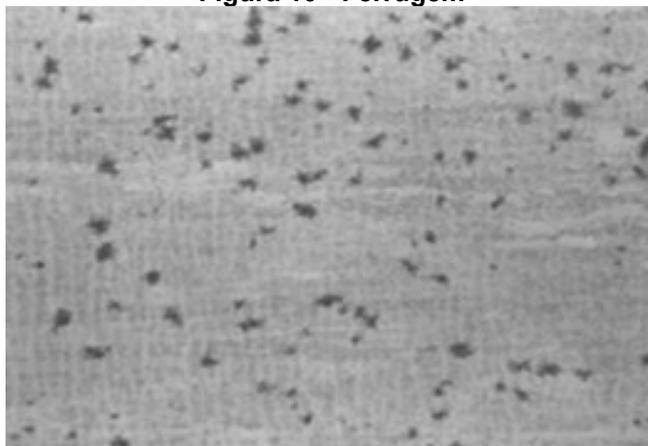


Fonte: NBR 16552 (2016)

2.3.11 Ferrugem

“Defeito de superfície caracterizado pela existência de regiões oxidadas, de coloração avermelhada, com intensidade variável. Pode se apresentar em forma de pontos ou manchas com extensão variável”, Figura 10. (ABNT NBR 16552, 2016).

Figura 10 - Ferrugem

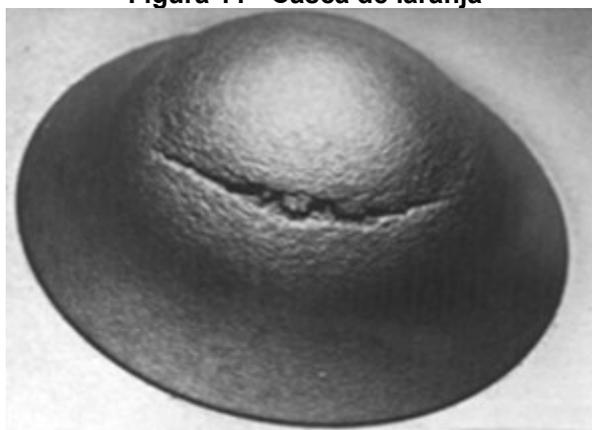


Fonte: NBR 16552 (2016)

2.3.12 Casca de laranja

“Defeito de superfície caracterizado pela formação de contornos arredondados de aspecto granular, que ocorre somente após a conformação de material com granulação grosseira”, Figura 11. (ABNT NBR 16552, 2016).

Figura 11 - Casca de laranja

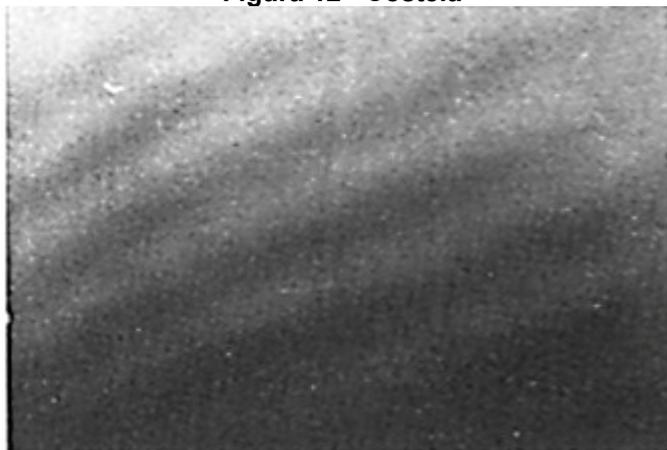


Fonte: NBR 16552 (2016)

2.3.13 Costela

“Defeito de superfície ou de forma caracterizado por uma sucessão de ondas diagonais à direção de laminação”, Figura 12. (ABNT NBR 16552, 2016).

Figura 12 - Costela

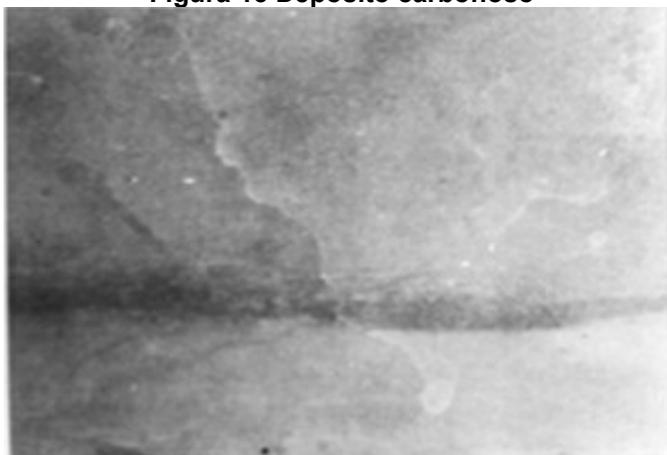


Fonte: NBR 16552 (2016)

2.3.14 Depósito carbonoso

“Defeito de superfície caracterizado pela existência de manchas, de cor variando do cinza esbranquiçado ao negro, causado pela decomposição de resíduos carbonosos”, Figura 13. (ABNT NBR 16552, 2016).

Figura 13 Depósito carbonoso

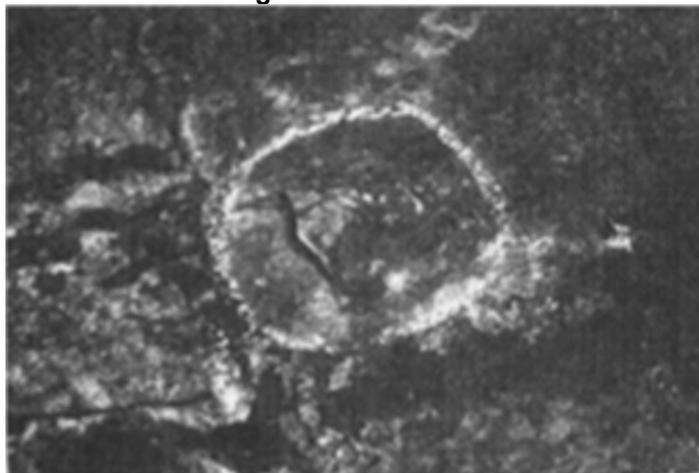


Fonte: NBR 16552 (2016)

2.3.15 Gota fria

“Defeito de superfície caracterizado pela existência de massa metálica de tamanho pequeno incrustada à superfície do material”, Figura 14. (ABNT NBR 16552, 2016).

Figura 14 - Gota fria

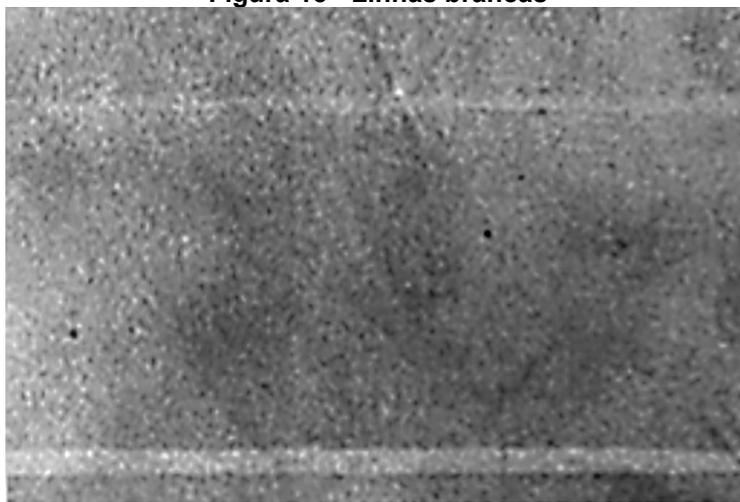


Fonte: NBR 16552 (2016)

2.3.16 Linhas brancas

“Defeito de superfície caracterizado pela existência de linhas claras paralelas à direção de laminação aleatoriamente distribuídas sobre a superfície do material”, Figura 15. (ABNT NBR 16552, 2016).

Figura 15 - Linhas brancas



Fonte: NBR 16552 (2016)

2.3.17 Linhas de distensão

“Defeito de superfície que ocorre somente após a conformação do material, caracterizado pela existência de estrias inclinadas de aproximadamente 45° em relação à direção de deformação; podem também ocorrer na forma de contornos

arredondados de aspecto granular, quando podem ser confundidos com “casca de laranja”.”, Figura 16. (ABNT NBR 16552, 2016).

Figura 16 - Linhas de distensão

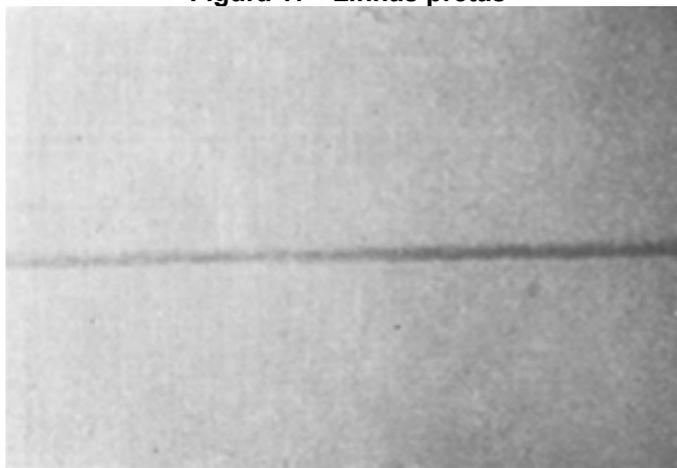


Fonte: NBR 16552 (2016)

2.3.18 Linhas pretas

“Defeito de superfície caracterizado pela existência de linhas escuras paralelas à direção da laminação, aleatoriamente distribuídas sobre a superfície do material”, Figura 17. (ABNT NBR 16552, 2016).

Figura 17 - Linhas pretas



Fonte: NBR 16552 (2016)

2.3.19 Marca de arame

“Defeito de superfície caracterizado por depressões, saliências e riscos, consequentes da utilização de arames para separação das espiras no processo de decapagem por imersão da bobina”, Figura 18. (ABNT NBR 16552, 2016).

Figura 18 - Marca de arame

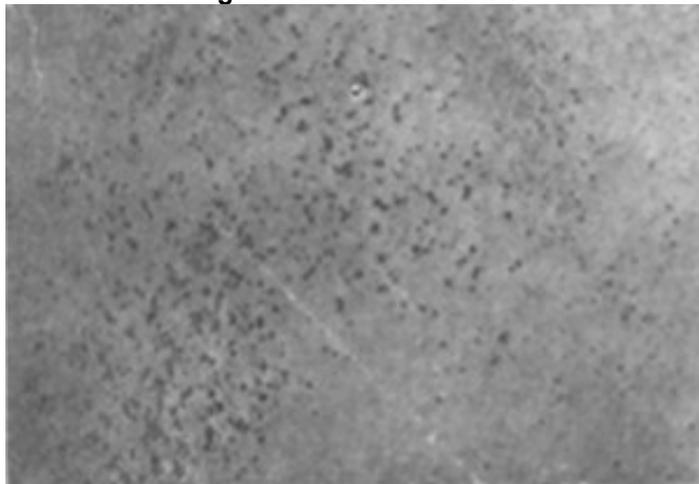


Fonte: NBR 16552 (2016)

2.3.20 Marca de areia

“Defeito de superfície caracterizado por cavidades deixadas em material a frio por grãos de areia provenientes do setor dos abafadores dos fornos de recozimento. Localizam-se geralmente próximos às bordas”, Figura 19. (ABNT NBR 16552, 2016).

Figura 19 - Marca de areia

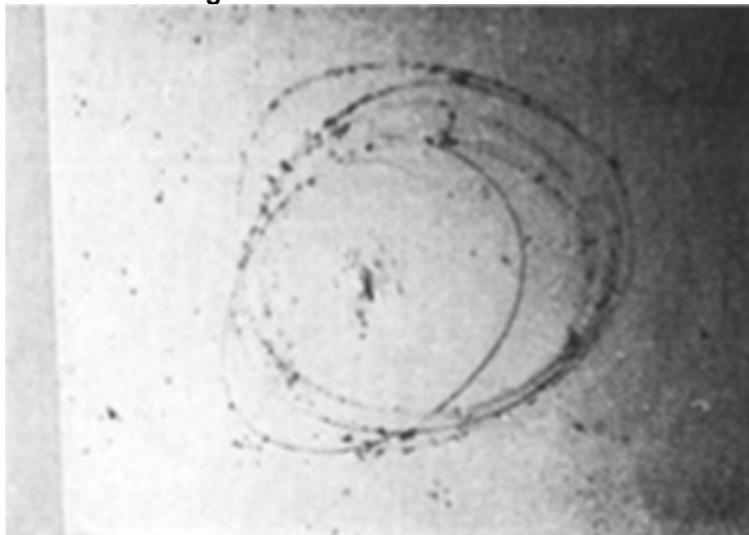


Fonte: NBR 16552 (2016)

2.3.21 Marca de colamento

“Defeito de superfície caracterizado pelo destacamento de partículas de metal de uma das faces do material em sequência do caldeamento, em pontos isolados, de espiras de bobina durante o recozimento. Localiza-se em geral no centro do material”, Figura 20. (ABNT NBR 16552, 2016).

Figura 20 - Marca de colamento

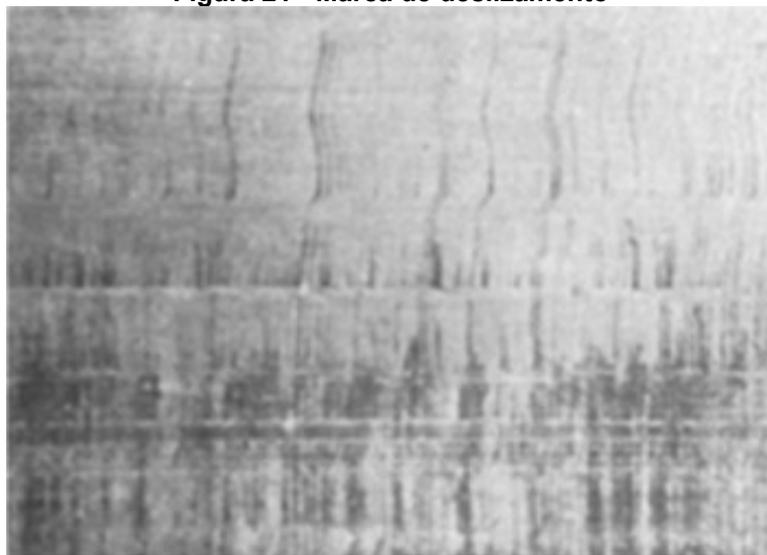


Fonte: NBR 16552 (2016)

2.3.22 Marca de deslizamento

“Defeito de superfície caracterizado pela existência de danificação, em formas de riscos e/ou cavidades, provocados pelo deslizamento entre espiras, durante o enrolamento ou desenrolamento de bobinas”, Figura 21. (ABNT NBR 16552, 2016).

Figura 21 - Marca de deslizamento

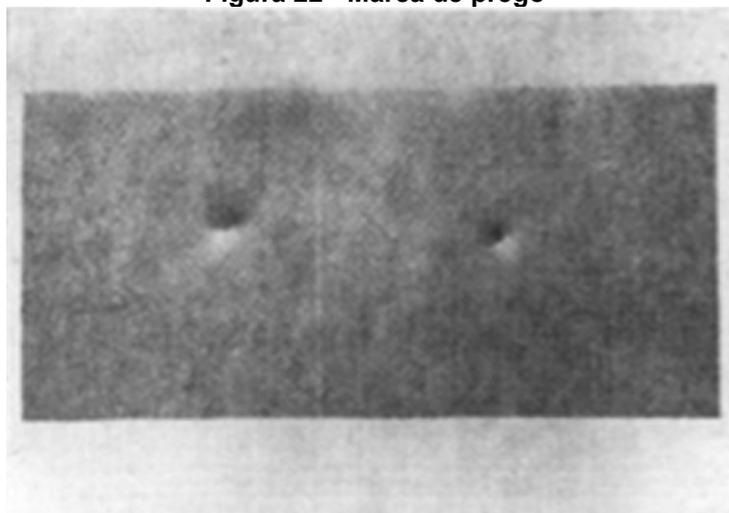


Fonte: NBR 16552 (2016)

2.3.23 Marca de prego

“Defeito de superfície caracterizado pela existência de depressões e saliências causadas por pregos utilizados na confecção dos estrados da embalagem”, Figura 22. (ABNT NBR 16552, 2016).

Figura 22 - Marca de prego

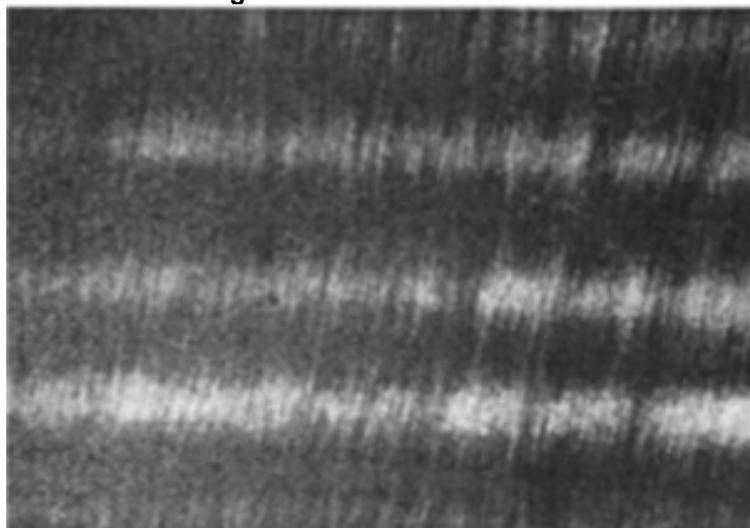


Fonte: NBR 16552 (2016)

2.3.24 Marca de retífica

“Defeito de superfície caracterizado por pequenas linhas longitudinais homogeneamente espaçadas e distribuídas por toda a superfície da chapa. São marcas provocadas por cilindros com retificação inadequada”, Figura 23. (ABNT NBR 16552, 2016).

Figura 23 - Marca de retífica

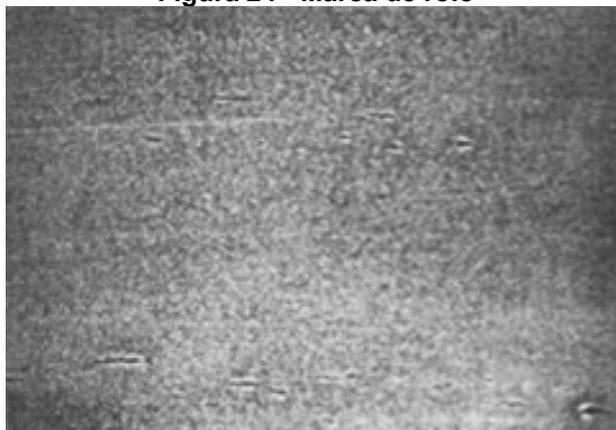


Fonte: NBR 16552 (2016)

2.3.25 Marca de rolo

“Defeito de superfície caracterizado por cavidades repetidas periodicamente ao longo do material, originadas por defeitos dos rolos”, Figura 24. (ABNT NBR 16552, 2016).

Figura 24 - Marca de rolo

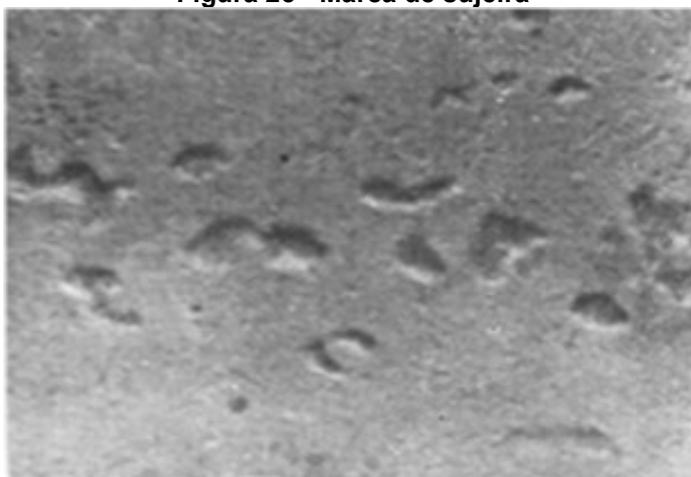


Fonte: NBR 16552 (2016)

2.3.26 Marca de sujeira

“Defeito de superfície caracterizado pela existência de depressões originadas por acúmulo de sujeira nos cilindros”, Figura 25. (ABNT NBR 16552, 2016).

Figura 25 - Marca de sujeira

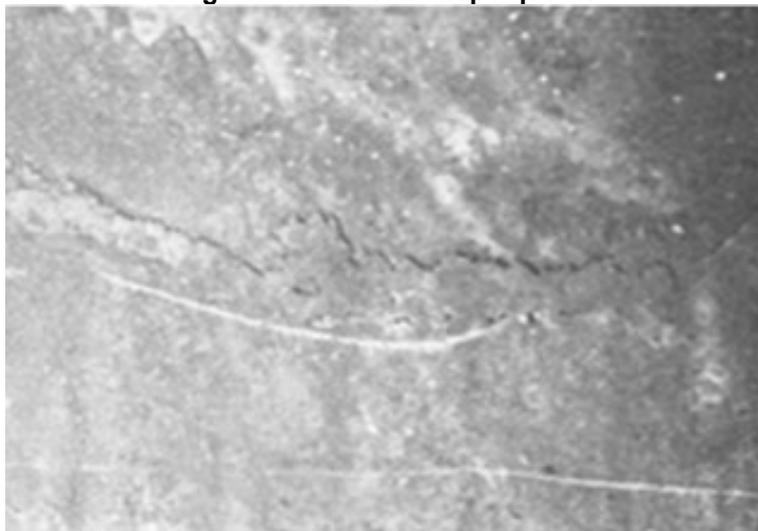


Fonte: NBR 16552 (2016)

2.3.27 Marca de topo quente

“Defeito de superfície caracterizado pela existência de marcas rugosas e profundas nas faces e bordas das chapas grossas na região oriunda do topo dos lingotes de aço acalmados ou estabilizados decorrentes da utilização de placas isolantes nas lingoteiras”, Figura 26. (ABNT NBR 16552, 2016).

Figura 26 - Marca de topo quente

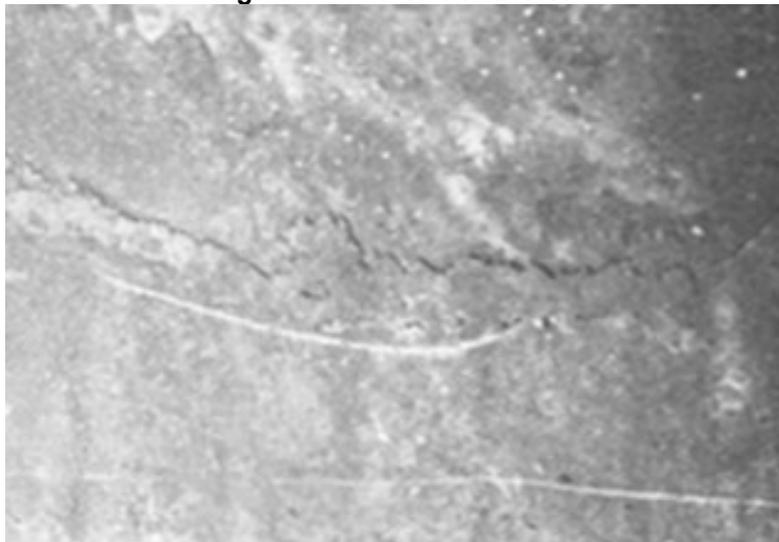


Fonte: NBR 16552 (2016)

2.3.28 Marca transversal

“Defeito de superfície caracterizado por estrias transversais, de intensidade variável, podendo apresentar-se com espaçamento constante ou uniformemente variado”, Figura 27. (ABNT NBR 16552, 2016).

Figura 27 - Marca transversal

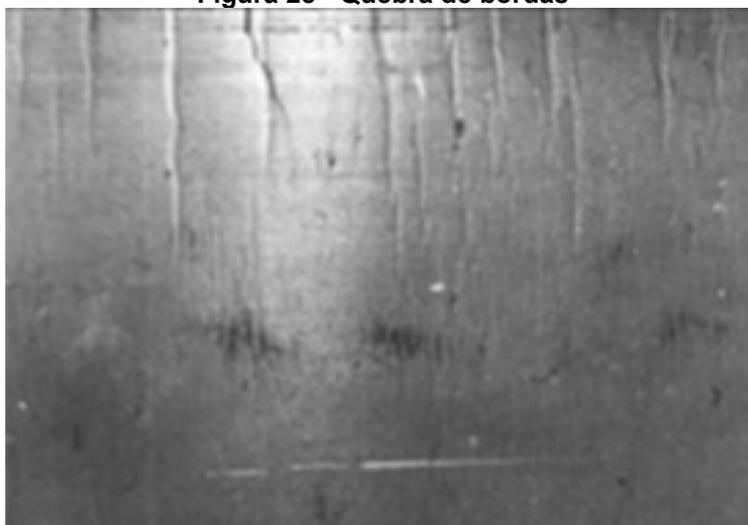


Fonte: NBR 16552 (2016)

2.3.29 Quebra de bordas

“Defeito de superfície caracterizado por linhas de distinção transversais que se estendem a partir das bordas do material, sem regularidade quanto à intensidade, espaçamento e extensão”, Figura 28. (ABNT NBR 16552, 2016).

Figura 28 - Quebra de bordas



Fonte: NBR 16552 (2016)

2.3.30 Quebra de selamento

“Defeito de superfície caracterizado por estrias semicirculares nas duas faces do material, a intervalos regulares ou não, sem localização preferencial quanto à

largura, originadas durante o desbobinamento de material recozido”, Figura 29. (ABNT NBR 16552, 2016).

Figura 29 - Quebra de selamento

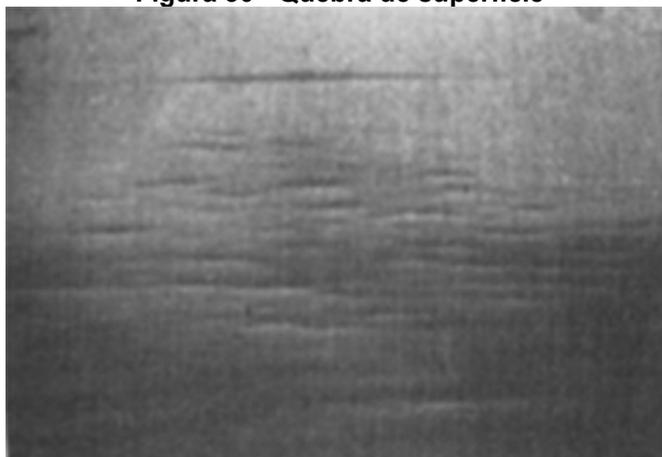


Fonte: NBR 16552 (2016)

2.3.31 Quebra de superfície

“Defeito de superfície caracterizado por estrias transversais, de intensidade variável, sensíveis ou não ao tato”, Figura 30. (ABNT NBR 16552, 2016).

Figura 30 - Quebra de superfície



Fonte: NBR 16552 (2016)

2.3.32 Refratário incrustado

“Defeito de superfície caracterizado pela incrustação de materiais refratário”, Figura 31. (ABNT NBR 16552, 2016).

Figura 31 - Refratário incrustado

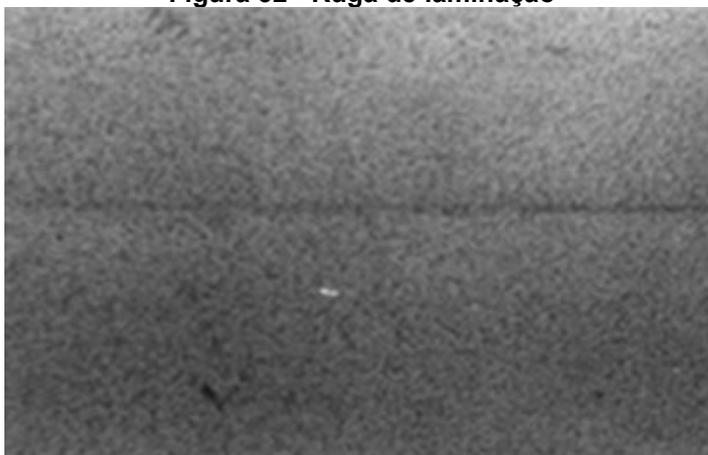


Fonte: NBR 16552 (2016)

2.3.33 Ruga de laminação

“Defeito de superfície caracterizado pela acomodação de ondulações durante a laminação, não chegando a configurar dobra”, Figura 32. (ABNT NBR 16552, 2016).

Figura 32 - Ruga de laminação



Fonte: NBR 16552 (2016)

2.3.34 Trincas

As trincas são caracterizadas por aberturas lineares na superfície do material, como demonstrado na Figura 33 abaixo e podendo se estender para camadas subsuperficiais. A formação de fissuras está relacionada a fatores como tensões residuais, composição química do aço e condições de resfriamento.

Durante o processo de laminação a quente, o aço é submetido a altas temperaturas e deformação plástica, o que pode gerar tensões residuais no material. Essas tensões residuais podem ser introduzidas durante a laminação ou podem se desenvolver durante o resfriamento do aço. As fissuras podem se formar como resultado dessas tensões, especialmente em regiões onde ocorre uma mudança brusca de espessura ou forma. (Mintz, 1988).

A composição química do aço também desempenha um papel importante na formação de fissuras. A presença de elementos como enxofre, fósforo e oxigênio podem aumentar a susceptibilidade à fissuração. Por exemplo, o enxofre e o fósforo podem levar à formação de inclusões de sulfeto e fosfeto, que atuam como locais de iniciação das fissuras. (Mintz, 1988).

As condições de resfriamento após a laminação também podem contribuir a formação de fissuras. Taxas de resfriamento mais rápidas podem resultar em gradientes de temperatura mais acentuados e, conseqüentemente, em maior geração de tensões residuais, aumentando a probabilidade de fissuração. (Mintz, 1988).

“Defeito de superfície caracterizado por aberturas lineares ou não, nas face ou bordas do material”, Figura 33. (ABNT NBR 16552, 2016).

Figura 33 - Trincas



Fonte: NBR 16552 (2016)

No entanto, a inspeção visual apresenta limitações em termos de eficiência e precisão, especialmente quando se trata de fissuras microscópicas ou de pequenas dimensões. Por essa razão, muitos estudos têm se concentrado no desenvolvimento de métodos não destrutivos automatizados para a detecção de fissuras superficiais em placas de aço laminado a quente.

2.3.35 Inclusões

As inclusões em placas de aço laminado a quente são caracterizadas pela presença de materiais estranhos incorporados ao aço durante o processo de fabricação. Esses materiais podem ser compostos por óxidos, sulfetos, silicatos e outros elementos não metálicos. A formação de inclusões ocorre devido a várias razões, incluindo a presença de impurezas nos materiais de partida, reações químicas durante o processo de fabricação e contaminação ambiental.

Durante o processo de laminação a quente, o aço é submetido a altas temperaturas e pressões, o que leva à formação de inclusões. Essas inclusões podem se originar de vários pontos ao longo da cadeia de produção, como a escória presente nos materiais de partida, as impurezas presentes no metal líquido, os revestimentos usados para proteção do aço, entre outros, Figura 34. (Luo, 2001).

Figura 34 - Inclusões



Fonte: NBR 16552 (2016)

As inclusões podem ser classificadas em diferentes tipos, dependendo de sua composição química e origem. Alguns dos tipos comuns de inclusões incluem inclusões de óxido (como óxido de ferro), inclusões de sulfeto (como sulfeto de manganês) e inclusões de silicato (como silicato de cálcio).

A presença de inclusões em placas de aço laminado a quente pode afetar negativamente a qualidade e as propriedades mecânicas do aço. Essas inclusões podem atuar como pontos de concentração de tensões, reduzindo a resistência à fadiga e aumentando o risco de falhas prematuras. (Luo, 2001).

2.3.36 Manchas

São defeitos superficiais que podem ocorrer em placas de aço laminado a quente, afetando sua qualidade e desempenho. Essas oxidações são caracterizadas por áreas localizadas com uma aparência diferente da superfície circundante, apresentando variações de cor, textura e brilho. (Ashour et al, 2019).

Manchas de óxido são formadas em aço primeiramente de três formas, hematita, magnetita e wustita, normalmente tem uma camada estrutural de hematita na camada superficial, uma camada intermediária de magnetita e uma camada interna de wustita, ou o produto da decomposição. (Chen, 2003)

2.3.36.1 Mancha d'água

“Defeito de superfície caracterizado pela existência no produto final de regiões oxidadas, de coloração variada, em consequência de ação de água”, Figura 35. (ABNT NBR 16552, 2016).

Figura 35 - Manchas d'água

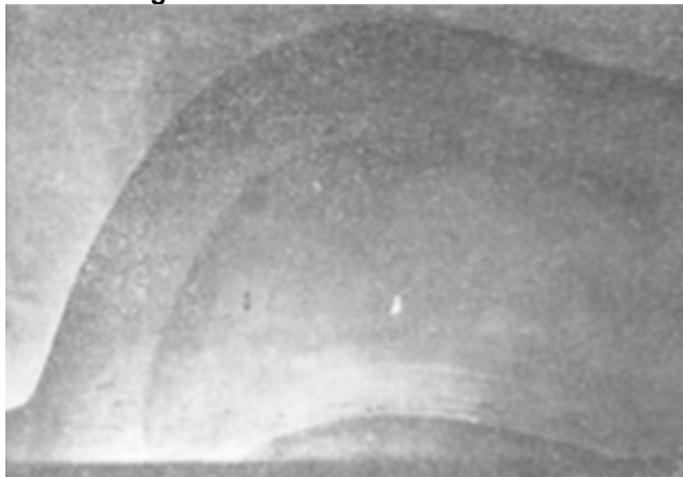


Fonte: NBR 16552 (2016)

2.3.36.2 Mancha de recozimento

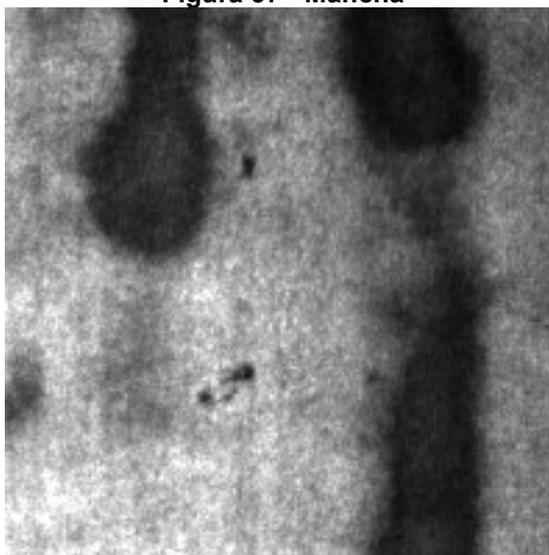
“Defeito de superfície caracterizado por alteração da coloração de material laminado a frio e recozido, variando do amarelo ao azul escuro; localiza-se geralmente junto às bordas”, Figura 36. (ABNT NBR 16552, 2016).

Figura 36 - Mancha de recozimento



Fonte: NBR 16552 (2016)

Figura 37 - Mancha



Fonte: NEU (2020)

A oxidação é caracterizada presença de manchas na superfície da chapa de aço laminadas a quente, Figura 37, onde o aspecto visível se trata de diferença na intensidade de cor, como manchas em áreas isoladas ou de maior abrangência. (Martins, 2009).

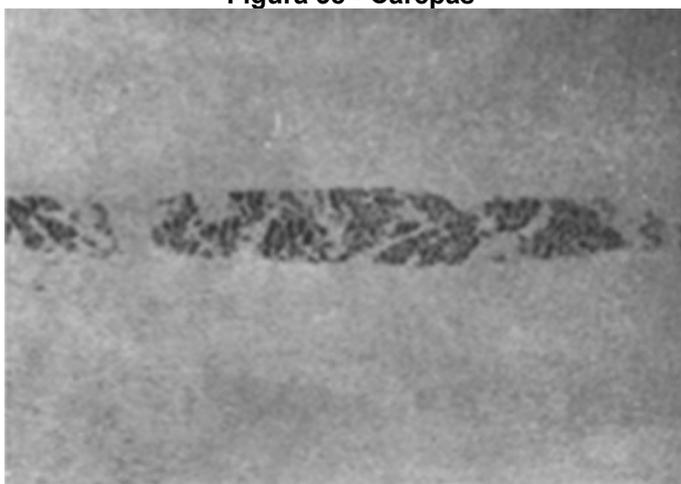
2.3.37 Carepas

São defeitos superficiais que podem ocorrer em placas de aço laminado a quente, resultando em pequenas depressões ou cavidades na superfície do material. Essas cavidades são geralmente irregulares em forma e distribuição, e podem comprometer a qualidade e o desempenho da chapa de aço.

A formação está relacionada a vários fatores durante o processo de fabricação do aço laminado a quente. Um dos principais fatores é a presença de inclusões não metálicas, como sulfetos e óxidos, que podem atuar como sítios preferenciais para a corrosão eletroquímica na superfície da chapa, Figura 38.

Defeito de superfície caracterizado pela incrustação de óxidos na superfície da chapa durante a laminação a quente. (ABNT NBR 16552, 2016).

Figura 38 - Carepas



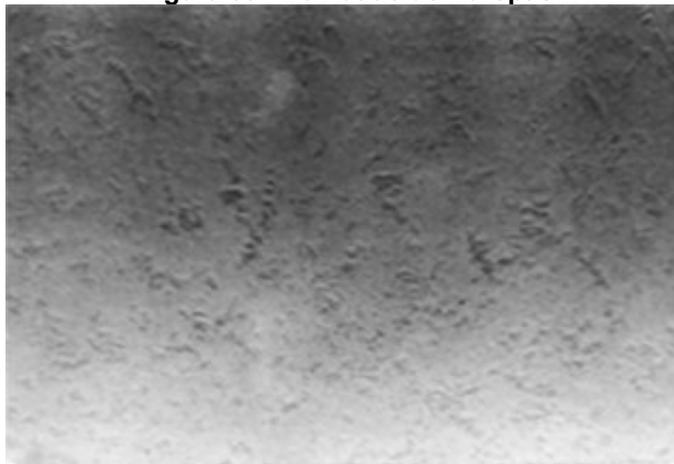
Fonte: NBR 16552 (2016)

Durante a laminação a quente e subsequente resfriamento das chapas de aço laminadas a quente, oxidação é formada na superfície do aço. Geralmente, a camada de oxidação é compacta e presente na superfície, esta apresenta uma importante função de resistência a corrosão. Entretanto, quando a camada é danificada ou quebrada, o aço sofre uma séria corrosão localizada, formando cavidades na superfície do material. (Meng, 2008).

2.3.37.1 Sal e Pimenta

Se trata de uma subclasse, dentro da classificação de carepas, onde pequenos pontos de defeito são observados dentro da placa de aço laminado, Figura 39. (Dey, 2013).

Figura 39 - Cavidade de Carepas



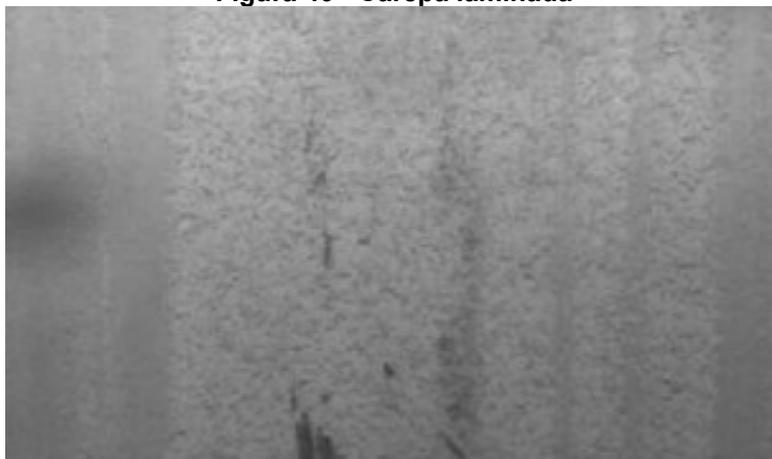
Fonte: NBR 16552 (2016)

A gravidade do defeito pode variar, desde desgastes superficiais leves até desgastes profundos que afetam a integridade estrutural da chapa de aço. Além disso, o defeito pode afetar negativamente a qualidade da superfície e a aparência estética da chapa de aço.

2.3.37.2 Carepa laminada

É um defeito superficial, o qual ocorre por condições térmicas ou mecânicas, normalmente originadas nos primeiros processos do término do processo de acabamento, por conta do alto desgastes dos rolos de laminação, Figura 40. (Braz, 2019).

Figura 40 - Carepa laminada



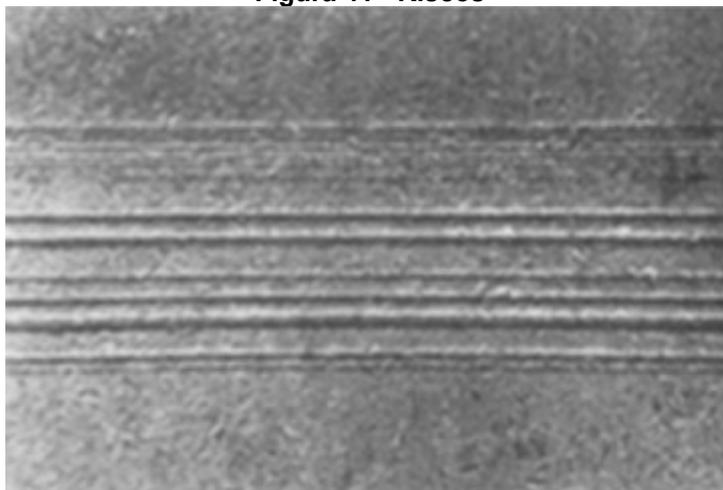
Fonte: Braz (2019)

2.3.38 Riscos

A fratura da camada de oxidação é o primeiro passo para o defeito de colagem durante o processo de laminação, concluiu-se que um risco profundo o suficiente para quebrar a camada de proteção de oxidação do aço inicia o defeito de colagem, onde a placa se solta da camada de proteção. (Jin, 2000).

A ocorrência de riscos nas placas de aço laminado a quente pode afetar comprometer a qualidade da superfície e a integridade estrutural do material. Riscos profundos podem levar a pontos de concentração de tensão, comprometendo ainda mais a resistência do aço, Figura 41.

“Defeitos de superfície caracterizado um ou mais sulcos de pequena profundidade geralmente paralelos à direção de laminação”. (ABNT NBR 16552, 2016).

Figura 41 - Riscos

Fonte: NBR 16552 (2016)

Estudos têm sido realizados para entender as causas desses defeitos, desenvolver técnicas de inspeção e detecção, bem como propor soluções para minimizá-los. Alguns desses estudos têm se concentrado no uso de técnicas de processamento de imagem, análise de padrões e redes neurais para a detecção e classificação de defeitos superficiais em placas de aço laminado a quente. (Wang et al. 2021).

2.4 Redes neurais

As redes neurais artificiais têm sido amplamente estudadas e aplicadas em diversas áreas devido à sua capacidade de aprender a partir dos dados. Elas são modelos de aprendizado de máquina que imitam o funcionamento do cérebro humano, permitindo a extração de informações relevantes a partir de dados complexos. (Ahmad, 2021).

Inspirado pelo sofisticado arbitramento do cérebro humano, onde bilhões de neurônios são interligados, processando informações em paralelo. Exemplos notáveis dessas estruturas são os softwares de tradução e reconhecimento de padrões. (Philipp, 2008).

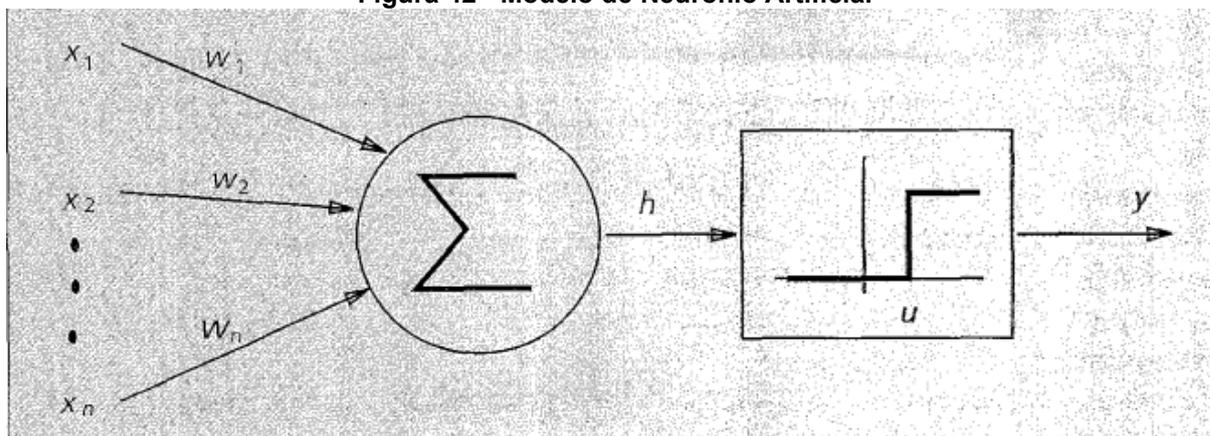
Enquanto as simulações de consciência e emoção humana ainda estão nos meios da ficção científica, consideramos a rede neural como um aproximador de funções cognitivas. (Wang, 2003).

2.4.1 Neurônio Artificial

Os pioneiros McCulloch e Pitts (1940) publicaram a proposta de um neurônio artificial, conforme pode-se observar na Figura 42, o qual tem uma entrada de uma soma de números, onde o neurônio tem sua entrada como $x_1, x_2, x_3, \dots, x_n$, resultando em uma saída de 1 ou 0, onde 1 seria o caso de uma soma maior que um limite proposto, ou resultado 0 o qual retrataria a soma inferior ao limite. Os pesos demonstrados como $w_1, w_2, w_3, \dots, w_n$, quantificam a importância de tal caminho adotado pelo neurônio.

Os dois provaram que pesos selecionados fazem com que neurônios artificiais façam operações diversas, sendo possível uma gama imensa de entradas, porém a saída do problema será resolvida. Há aqui uma analogia grosseira com um neurônio biológico: fios e interconexões modelam axônios e dendritos, os pesos das conexões representam a sinapses, ou o local onde ocorre a transmissão de impulsos nervosos, e a função retrata a operação de uma soma. Entretanto McCulloch e Pitts fazem algumas simplificações que não refletem a verdadeira natureza de um neurônio biológico. (Jain, 1996).

Figura 42 - Modelo de Neurônio Artificial



Fonte: McCulloch e Pitts (1996)

2.4.2 Características de redes neurais

As redes neurais podem ser reduzidas a quatro características básicas, dentro das quais:

1 - Não são lineares. A sua não linearidade é uma clara relação com a natureza, sendo que a inteligência humana não é um fenômeno linear. Neurônios artificiais são ativados ou inibidos, esse comportamento é a não linearidade de uma relação matemática. Uma rede de neurônios consegue atingir capacidades maiores de performance e armazenamento de dados. (Wu, 2018).

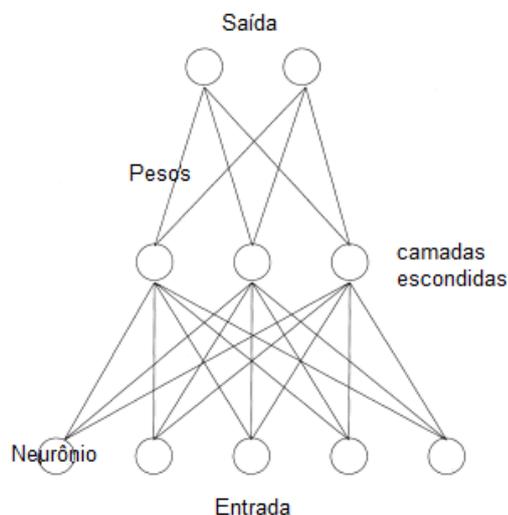
2 – Não são limitadas. Uma rede neural normalmente consiste em múltiplos neurônios conectados. O comportamento do sistema não depende apenas da característica de apenas um neurônio e sim da interação entre os neurônios conectados, podemos observar esse comportamento pelo número de interações entre neurônios. (Wu, 2018).

3 – Não qualitativa. A rede neural artificial pode ser auto adaptativa, assim como auto arranjada e auto didata. Redes neurais processam informações que podem conter várias alterações, além de computar as informações. Sua não linearidade a permite ficar em constante mudança, com as suas interações mostrando a evolução do sistema. (Wu, 2018).

4 – Não são convexas. O efeito de constante mudança do sistema, sobre certas condições, depende do estado das funções. Isto significa que a evolução da rede ocorre com diversidade. (Wu, 2018).

2.4.3 Arquitetura de redes neurais

Uma das vantagens das redes neurais é sua capacidade de lidar com problemas de alta dimensionalidade. Elas são capazes de aprender e representar relações complexas entre os dados, o que as torna adequadas para problemas de reconhecimento de padrões e classificação, Figura 43. (Vargas et al, 2016).

Figura 43 - Arquitetura da Rede Neural

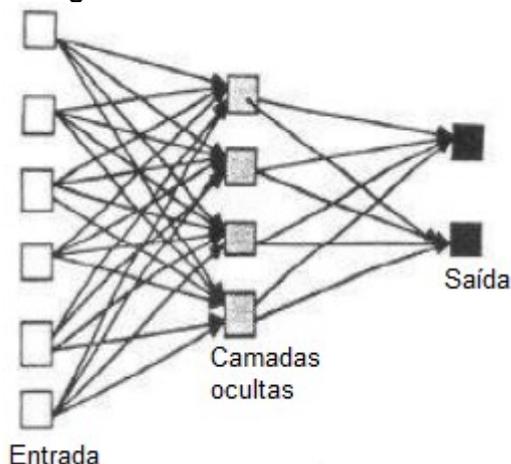
Fonte: Adaptado de Wang (2003)

2.4.4 Tipos de redes neurais

As redes neurais podem ser organizadas em diferentes tipos, dependendo da natureza do problema e dos dados disponíveis. Uma arquitetura comumente utilizada é a rede neural feedforward, Figura 44, que possui camadas de neurônios dispostas em uma sequência, sem ciclos de retroalimentação. Essas redes são capazes de mapear uma entrada para uma saída através de uma série de transformações não-lineares. (Wilamowski, 2009).

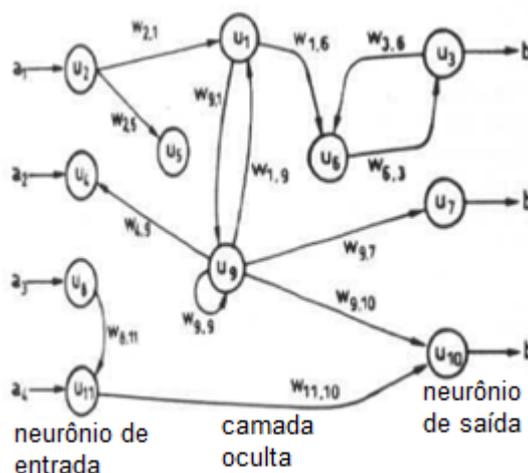
Nas redes neurais feedforward, o dado é transportado por um único caminho de neurônios artificiais, porém em redes neurais feedback o dado é transportado nas duas direções por loops, Figura 45. (Sharma, 2012).

Figura 44 - Rede neural feedforward



Fonte: Adaptado de Christos (1996)

Figura 45 - Rede neural feedback



Fonte: Adaptado de Christos (1996)

2.4.5 Arquitetura das redes neurais

As redes neurais artificiais podem ser implementadas em diferentes arquiteturas, cada uma com suas características e aplicações específicas. Ao longo dos anos, várias arquiteturas têm sido propostas e estudadas para melhorar o desempenho e a eficácia das redes neurais.

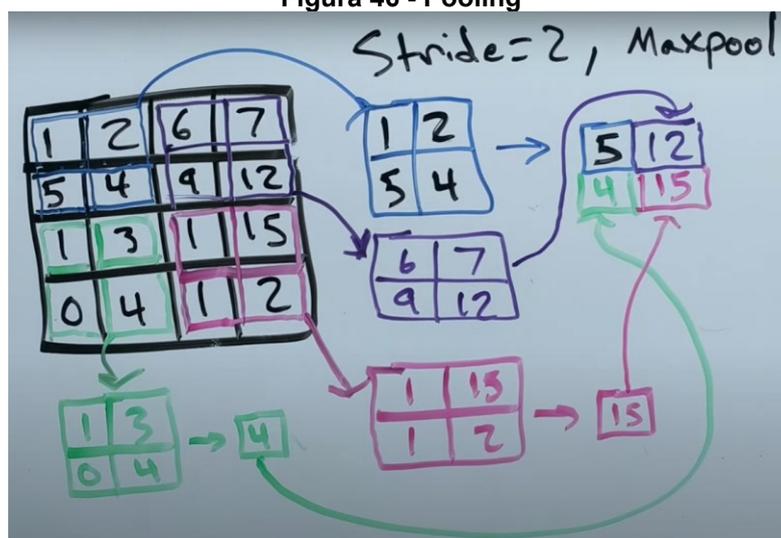
Outra arquitetura amplamente utilizada é a Rede Neural Convolutiva (CNN), a qual será o foco do projeto, uma arquitetura dentro do tipo Feedforward, que foi desenvolvida especificamente para tarefas de processamento de imagem e visão computacional. As CNNs são projetadas para extrair características locais e

invariantes de posição dos dados de entrada, por meio do uso de camadas convolucionais e camadas de pooling. Essa arquitetura se mostrou altamente eficaz em tarefas como classificação de imagens e detecção de objetos. (Tu et al, 2017).

A rede neural convolucional tem este nome pela operação entre matrizes, chamada de convolução. Esta rede possui múltiplas camadas, incluindo camadas de convolução, camadas de pooling e camadas inteiramente conectadas. As camadas de convolução e inteiramente conectadas tem parâmetros, entretanto as camadas de pooling não necessitam. (Albawi, 2017)

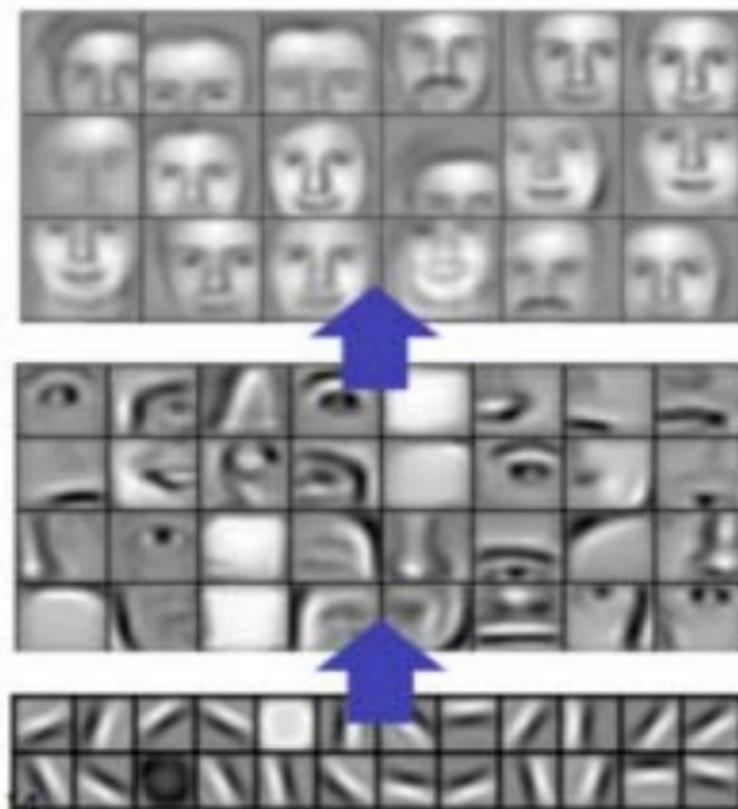
O objetivo da camada do pooling é reduzir a quantidade de dados, fazendo uma filtragem de dados para preservar os mais importantes, normalmente utilizando o valor máximo ou médio da operação, utilizando uma parte da matriz seleccionada pela configuração do passo do pooling, mostrado na Figura 46. (Sighn, 2020).

Figura 46 - Pooling



Fonte: IntuitiveML (2021)

Outro aspecto importante da CNN, é a obtenção de características da entrada, os quais nas próximas camadas passa a serem aproveitadas, como por exemplo, a primeira camada pode detectar as bordas, enquanto a segunda pode detectar a forma da imagem, assim atingindo uma grande qualidade de atributos, como descrito em Figura 47. (Albawi, 2017).

Figura 47 - Aprendizado Convolutional

Fonte: Albawi (2017)

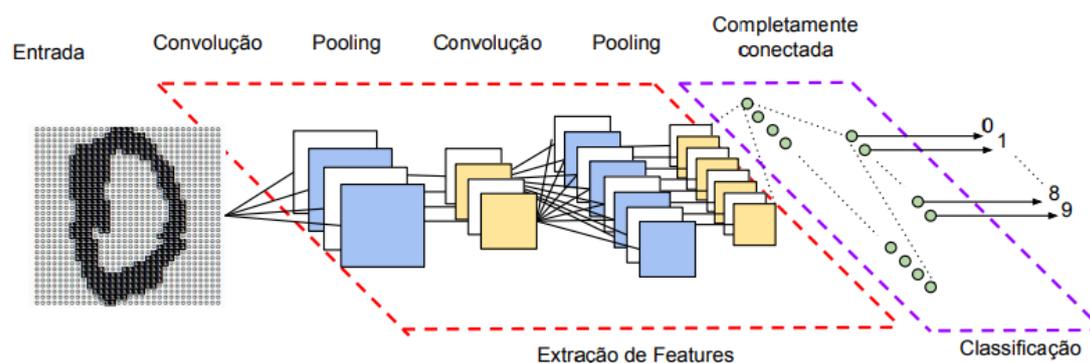
Rede neurais convolucionais profundas tem tido um grande impacto em várias aplicações assistidas por computador, como classificação de imagens, detecção de objetos, e câmeras de vigilância. Entretanto, a quantidade grande de dados fornecidos e complexidade da rede neural convolutiva, como demonstrado em Figura 48, gera um desafio grande para a eficiência eletrônica e performance, onde a eficiência elétrica pode ser definida pelo gasto eletrônico de um processamento de dado. (Le, 2013).

As CNNs são capazes de aprender automaticamente características discriminativas a partir das imagens de entrada por meio de camadas convolucionais e de pooling. Essas camadas aplicam filtros convolucionais e operações de redução de dimensionalidade para extrair características hierárquicas de diferentes níveis de abstração. (Li et al, 2021).

A arquitetura da CNN utilizada foi a conhecida como VGGNet, proposta por Simonyan e Zisserman (2014). Essa arquitetura se destaca por sua profundidade e por ser capaz de aprender representações robustas de imagens. Ao utilizar uma rede

pré-treinada com pesos aprendidos em um grande conjunto de dados, é possível aproveitar o conhecimento prévio da rede e acelerar o processo de treinamento. (Liu et al, 2017).

Figura 48 - Rede Convolutacional



Fonte: Vargas et al (2016).

Além das arquiteturas, os algoritmos de aprendizado também desempenham um papel fundamental no treinamento das redes neurais, alimentando-se de alguma base de dados de treinamento e aplicando em uma base de dados para teste, previsão e classificação. (Mahesh, 2020).

2.4.5.1 VGG16

Dentro das redes neurais convolucionais notáveis está a VGG-16, sugerida pelo grupo de geometria visual, os parâmetros para treinamento são encontrados dentro das 16 camadas, além de camadas com pooling. Essa arquitetura de rede neural é a sucessora da VGG, a qual foi o primeiro lugar da 2014 visual recognition challenge Figura 49, na categoria de classificação e localização, onde era proposto o uso de dados da ImageNet, um repositório de imagens com 14.197.122 imagens. (ImageNet, 2014).

Figura 49 - Vencedor visual recognition challenge 2014
Classification+localization

Task 2a: Classification+localization with provided training data

Classification+localization with provided training data: Ordered by localization error

Team name	Entry description	Localization error	Classification error
VGG	a combination of multiple ConvNets (by averaging)	0.253231	0.07405
VGG	a combination of multiple ConvNets (fusion weights learnt on the validation set)	0.253501	0.07407
VGG	a combination of multiple ConvNets, including a net trained on images of different size (fusion done by averaging); detected boxes were not updated	0.255431	0.07337
VGG	a combination of multiple ConvNets, including a net trained on images of different size (fusion weights learnt on the validation set); detected boxes were not updated	0.256167	0.07325
GoogLeNet	Model with localization ~26% top5 val error.	0.264414	0.14828
GoogLeNet	Model with localization ~26% top5 val error, limiting number of classes.	0.264425	0.12724
VGG	a single ConvNet (13 convolutional and 3 fully-connected layers)	0.267184	0.08434
SYSU_Vision	We compared the class-specific localization accuracy of solution 1 and solution 2 by the validation set. Then we chosen better solution on each class based on the accuracy. General speaking, solution 2 outformed solution 1 when there were multiple objects in the image or the objects are relatively small.	0.31899	0.14446

Fonte: (ImageNet, 2014).

A entrada foi fixada em 224 x 224 pixels em uma imagem RGB, computada na base de dados de treino, para cada pixel. A imagem passa por um compilado de camadas de convoluções, onde elas são filtradas em pequenos campos de 3 x 3 (o menor campo possível para ainda ter a noção de baixo e cima, esquerda e direita). Em uma das configurações também utilizaram filtros convolucionais de 1 x 1, os quais podem ser observados como uma transformação linear dos dados de entrada, seguidos por camadas não lineares. (Simonyan, 2014).

O passo da camada convolucional, seria o tamanho da matriz, foi fixado em 1 pixel. É preservada a resolução especial da camada de entrada após a convolução, feito então o pooling pelo valor máximo em cima de uma janela de 2 x 2 pixels, utilizando um passo de tamanho 2. (Simonyan, 2014).

Então um compilado de camadas convolucionas com diferentes profundidades e arquiteturas são adicionadas, seguida por três camadas inteiramente conectadas. As duas primeiras camadas inteiramente conectadas têm 4096 canais cada, e a terceira faz 1000 iterações de maneiras de classificar, assim contendo 1000 canais. A camada final é uma camada softmax. (Simonyan, 2014).

A camada softmax é uma camada específica para redes neurais de classificação, tendo uma função crucial no cálculo da função de perda utilizada para otimizar os parâmetros da rede neural. (Hu, 2014).

Rectified Linear Unit (ReLU) é normalmente utilizada como uma função ativadora para redes neurais, com softmax sendo a função de classificação. Então

essas redes neurais utilizam softmax com função de perda cross-entropy para aprender os parâmetros de peso da rede neural. (Agarap, 2018).

2.4.6 Pré-processamento

O primeiro passo para o processamento de imagens é o pré-processamento, desempenhando um papel fundamental na preparação e melhoria das imagens antes da realização de tarefas de análise e processamento.

O pré-processamento, o qual é normalmente subestimado, se dá por qualquer operação que consista em carregar um dado, no qual o resultado é uma imagem. Operações de pré-processamento geralmente ficam em uma de três categorias, a primeira é a reconstrução de imagem, usada para reconstruir uma imagem a partir de uma medição de dados. A segunda sendo restauração de imagem, utilizada para retirar ruídos e outras impurezas da imagem. Por fim o melhoramento de imagem, acentuando alguns pontos importantes da imagem, o que facilita no processamento de reconhecimento de objetos. (Egmont, 2002).

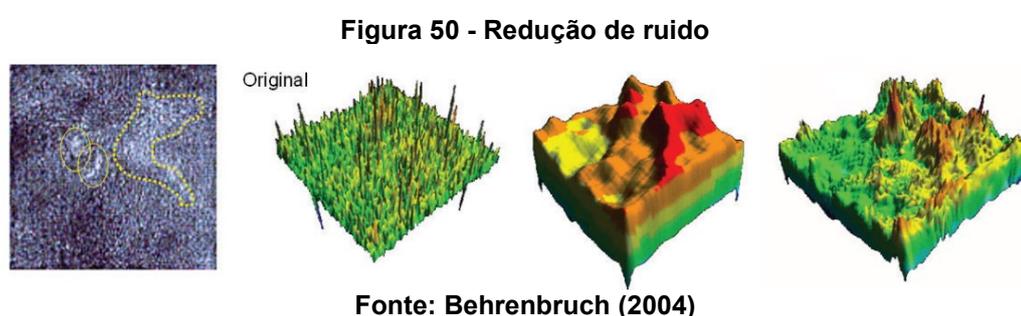
Visto que a imagem por natureza é de grande dimensão, mecanismos de extração de características são necessários para a segmentação ou reconhecimento do objeto. Além de diminuir o custo computacional, extração de características é uma ferramenta para controlar a tão famosa maldição da dimensionalidade. (Devijver, 1982).

Segmentação é uma partição da imagem em partes coerentes dentro de um critério. Quando observamos a tarefa de classificação o propósito da segmentação fica ainda mais evidente, utilizada para colocar um rotulo em cada pixel ou voxel da imagem. Alguns algoritmos de redes neurais fazem a segmentação diretamente nos dados de pixels, obtidos por uma janela de convolução ou sendo fornecida para uma classificação neural em forma de características locais. (Egmont, 2002).

2.4.7 Pós-processamento

O pós-processamento de imagem desempenha um papel crucial na melhoria da qualidade e na manipulação de imagens.

Uma das técnicas comumente utilizadas é a filtragem de imagem, que tem o objetivo de reduzir o ruído e melhorar a nitidez das imagens. Dentre os métodos de filtragem, destaca-se o filtro de média, que suaviza a imagem ao calcular a média dos valores dos pixels em uma vizinhança. Além disso, o filtro de mediana mostrado na Figura 50, é amplamente utilizado para remover ruídos impulsivos, substituindo o valor de um pixel pela mediana dos valores dos pixels em sua vizinhança. (Behrenbruch, 2004).



Outra técnica importante é a segmentação de imagem, que visa identificar regiões de interesse ou objetos em uma imagem. A segmentação pode ser realizada por meio de métodos baseados em limiarização, como o Método de Otsu, que determina automaticamente o valor de limiar ótimo para separar as regiões da imagem. (Talab, 2016).

O melhoramento de dados é uma técnica crucial usada em aprendizado de redes neurais, particularmente em imagens, quando são utilizadas para classificação, o principal motivo é para aumentar a diversidade dos dados de treinamento, assim aumentando a habilidade do modelo de generalizar com uma quantidade limitada de imagens. (Egmont, 2002).

A restauração de imagem é uma técnica fundamental para recuperar informações perdidas ou danificadas durante o processo de aquisição. Um método amplamente utilizado nesse contexto é a Restauração de Wiener, que utiliza o conhecimento do modelo de degradação da imagem para restaurá-la de maneira ótima. (Baselice et al, 2018).

Outra técnica comumente aplicada é a transformação de cores, que permite alterar as características de cores de uma imagem, é amplamente utilizada para

expandir a faixa dinâmica das imagens, realçando detalhes em regiões de baixa intensidade. (Lukac, 2018).

2.4.8 Algoritmos de aprendizado

O aprendizado mais comum de uma rede neural convolucional, se dá por meio de um gradiente estocástico descente, juntamente com uma função de perda Cross Entropy. A função de perda é popular devido sua excelente velocidade de convergência, além da performance de ponta em acurácia em classificação. (Martinez, 2019).

No amplo campo de aprendizado profundo de máquinas, aleatoriedade é fundamental em vários aspectos do modelo de treino, como, os primeiros pesos, embaralhamento da base de dados e melhora nos dados fornecidos. Entretanto, a aleatoriedade pode introduzir variabilidade nos experimentos, sendo mais desafiador reproduzir os mesmos resultados entre compilações do mesmo código. (Scardapane, 2017).

Os algoritmos de aprendizado no treinamento de modelos de inteligência artificial e aprendizado de máquina, são responsáveis por permitir que os modelos aprendam a partir dos dados e sejam capazes de fazer previsões ou tomar decisões com base nesse aprendizado.

Support vector machines (SVMs), foram introduzidas por Vapnik em 1990, sendo uma família de algoritmos para aprendizado discriminando função de duas classes da base de dados de treino. Elas são aplicadas em uma variedade de áreas, incluindo categorização de texto, reconhecimento de imagem, reconhecimento de escrita, entre outros. O método combina ideias do aprendizado por estatística e otimização via convex, para achar limites na base de dados entre duas classes. (Mammone, 2009).

Outro algoritmo de aprendizado amplamente utilizado é a árvore de decisão, sendo um dos algoritmos de aprendizado com maior sucesso, por ter vários mecanismos, como por exemplo, simplicidade, entendimento, nenhum parâmetro, e possibilidade de utilizar uma base de dados de diferentes assuntos. No aprendizado de árvore de decisão, a árvore é induzida a características da base de

dados, rotulando os dados, representado por uma lista de valores de atributos e um rótulo de classe. (Su, 2006).

No aprendizado profundo de uma rede neural, os hyper parâmetros são um fator crítico que pode influenciar o treinamento e a performance da rede neural, os dois mais fundamentais são o tamanho do lote e o número de períodos utilizados no treinamento.

Batch Size: O batch size, ou tamanho do lote, se refere ao número de amostras utilizadas em cada iteração do processo de treinamento, é um hyper parâmetro pivô que afeta tanto a eficiência do treino quanto a convergência do modelo, tamanho de lotes menores, como 32 ou 64, fazem com que os pesos atualizem frequentemente, fazendo com que o programa convirja rapidamente, porem com potencialmente menos qualidade, tamanhos de lotes maiores, antagonicamente, providenciam uma qualidade maior, mas uma convergência mais lenta, a escolha do tamanho normalmente envolve em um acordo entre velocidade de treinamento e qualidade de convergência. (Smith, 2018).

Training Epochs: O número de períodos utilizados define quantas vezes a rede neural passa pela base de dados de treinamento, é um fator crítico que determina a duração do treinamento e o risco de sobre ajustes. Treinando por períodos pequenos pode resultar em um modelo fraco, enquanto treinar por muito tempo pode resultar em um modelo com sobre ajustes ou vícios, onde o modelo pode aprender a computar os erros, achar o ponto certo é normalmente feito por experimentação. (Schwenk, 2000).

2.4.8.1 Função de perda

A função de perda tem duas principais ações no treinamento de uma rede neural convencional, primeiro medindo o índice de classificação e então gerando gradientes que direcionam o treinamento da rede. (Li, 2020).

A com maior destaque na categoria de funções de perda baseadas na probabilidade, é a cross entropy, a qual é a generalização de uma regressão lógica para um cenário de múltiplas classes, sendo proposta por Bridle. (1990).

2.4.8.2 Otimização

O gradiente estocástico é um ponto crucial na otimização em muitos campos da ciência e engenharia. Muitos problemas podem ser considerados como a otimização de alguma função objetivo parametrizada, requerendo maximização ou minimização dependendo dos seus parâmetros. (Kingma, 2014).

Foi proposto por Kingma em 2014 o método Adam de otimização, um método estocástico que requer apenas um gradiente de primeira ordem, com pouco custo de memória. O método computa separadamente o ritmo de aprendizado para diferentes parâmetros, estimados do primeiro e segundo momento do gradiente, seu nome Adam vem de Adaptive Moment Estimation.

2.4.8.3 Transfer learning

A metodologia de treinamento para redes neurais, se trata de um treinamento em uma base de dados fornecida, juntamente com o teste, retirados do mesmo fornecimento, porém isso não é verdade para operações em que a base de dados é de difícil obtenção ou de exorbitante preço. Assim, a necessidade de criar uma rede neural pré treinada é grande, está sendo construída em cima de uma base de dados de fácil acesso, essa metodologia é conhecida como transfer learning. (Weiss, 2016).

Utilizando uma base de dados pequena, o método de transferência de conhecimento ou transfer learning pode ser usado para detecção de defeitos de superfície em placas de aço. Com o progresso da tecnologia, o processo de laminação de aço foi melhorado, onde a velocidade de uma planta moderna de laminação chega a 45 m/s, enquanto a velocidade de laminação de fio excedeu 130 m/s, o que é um grande desafio para a detecção de defeitos. Sendo o controle de qualidade em tempo real, primordial. (Tang, et al, 2023).

Isso é particularmente valioso quando a rede neural pré-treinada compartilha de padrões ou características com a tarefa nova. Isso habilita o modelo para aprender uma ação específica enquanto se beneficia do grande conhecimento prévio de uma rede neural pré treinada. (Kruithof et al, 2016).

2.4.8.4 Validação

Uma das principais métricas utilizadas para avaliar o desempenho do modelo é a matriz de confusão, Figura 76. Essa matriz permite visualizar o número de acertos e erros de classificação para cada classe. Com base nesses valores, é possível calcular métricas adicionais, como acurácia, precisão e recall, que fornecem uma medida quantitativa do desempenho do modelo em detectar e classificar os defeitos nas placas de aço. (Krstinić, 2020).

Além da matriz de confusão, é comum utilizar a curva (Receiver Operating Characteristic) ROC para avaliar o desempenho do modelo em diferentes limiares de decisão, Figura 78. A curva ROC mostra a taxa de verdadeiros positivos em relação à taxa de falsos positivos para diferentes limiares. A área sob a curva (AUC) é uma medida que resume a capacidade do modelo de distinguir corretamente as classes. (Woods, 1997).

Validação é um aspecto crucial de qualquer modelo. Entretanto, não existe um método formulado ou metodologia teórica para a validação de modelos de redes neurais, normalmente para validar o modelo é utilizado uma base de dados de validação, uma base de dados que não foi utilizada durante o treinamento. (Twomey et al, 1997).

Existe a tentativa de formular uma metodologia teórica para validação de modelos de redes neurais, dentro da necessidade industrial de tal formulação Taylor, 2006, propõe uma metodologia a ser seguida.

A F1-score é uma métrica estatística que combina precisão e recall em um único valor, oferecendo uma medida equilibrada do desempenho de um modelo em problemas de classificação. Essa métrica é especialmente útil quando as classes têm desequilíbrios significativos, garantindo uma avaliação mais robusta do modelo. A F1-score varia de 0 a 1, onde 1 representa um desempenho perfeito, indicando um equilíbrio ideal entre precisão e recall. (Gambäck, 2017)

2.4.9 Utilização de redes neurais para detecção de defeitos:

A detecção de defeitos em diferentes materiais e superfícies desempenha um papel crucial em várias indústrias, como a de manufatura, inspeção de qualidade e manutenção preditiva. A aplicação de redes neurais tem se mostrado uma abordagem promissora para a detecção automatizada de defeitos, oferecendo alta precisão e eficiência.

As redes neurais convolucionais (CNNs) têm sido amplamente utilizadas para a detecção de defeitos em diversos materiais, incluindo metais, plásticos e tecidos. Essas redes são projetadas para extrair características relevantes das imagens e identificar padrões associados aos defeitos. A arquitetura das CNNs, com camadas convolucionais e de pooling, permitem uma aprendizagem hierárquica das características, tornando-as adequadas para a detecção de defeitos em diferentes escalas. (Gu et al, 2018).

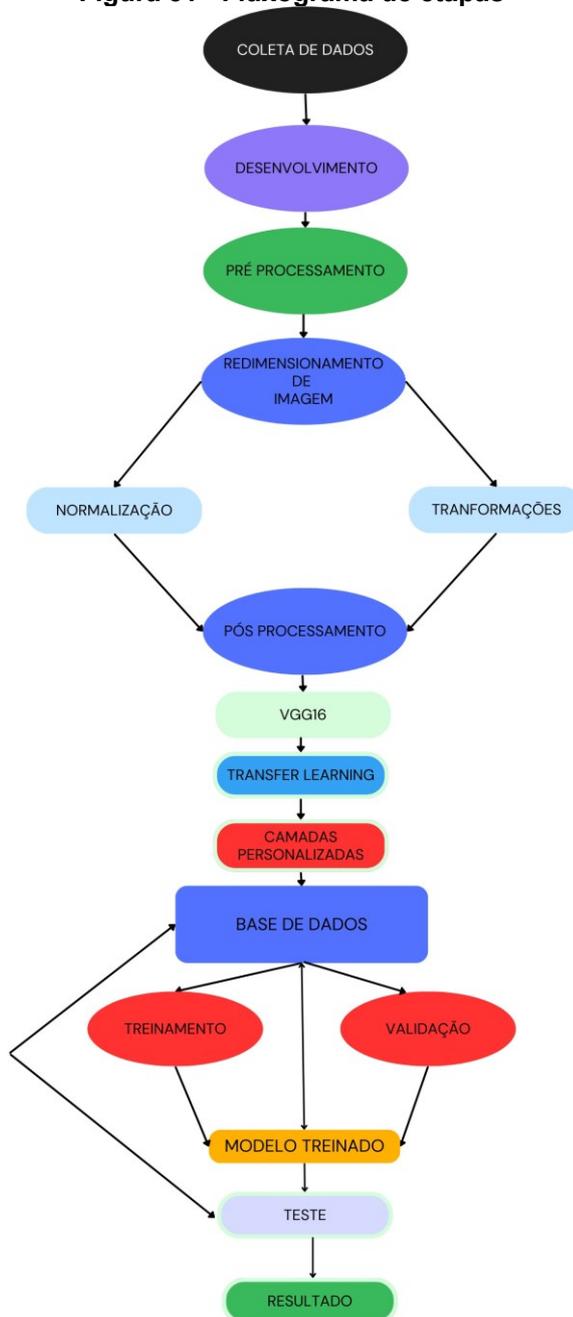
Além das CNNs, outros tipos de redes neurais, como redes neurais profundas (DNNs) e redes neurais recorrentes (RNNs), também têm sido empregadas na detecção de defeitos. As DNNs são capazes de aprender representações complexas dos dados e identificar padrões sutis, enquanto as RNNs são adequadas para a detecção de defeitos em sequências, como em inspeções contínuas de superfícies. (Yin et al, 2017).

A utilização da VGG16 como base de modelos se mostra eficiente, reduzindo a utilização do sistema e o consumo de memória, pela redução de tamanho e profundidade da estrutura da rede neural, além de adicionar uma camada de normalização que acelera o tempo de convergência do modelo. Baseado na base de dados de NEU Wang et al. (2021) conseguiu um resultado surpreendente de 99,63% de acurácia e um tempo de reconhecimento de 333 Frames por segundo. No mesmo contexto os resultados obtidos por Zhao, 2023, utilizando um modelo de detecção de defeitos em placas de aço laminadas a quente foi uma rede neural convolucional de paralelismo profundo, resultaram em 99,57% e 99,6% de precisão.

3 METODOLOGIA

A fim de esclarecer a metodologia utilizada, foi formulado um fluxograma de etapas, neste relatado todo processo do projeto Figura 51.

Figura 51 - Fluxograma de etapas



Fonte: Autoria própria (2023)

3.1 Coleta de dados

Uma base de dados confiável é de suma importância para uma rede neural eficiente, do mesmo modo, toda formatação da base de dados e arquitetura da rede neural, assim fornecendo uma boa base para uma rede neural eficaz.

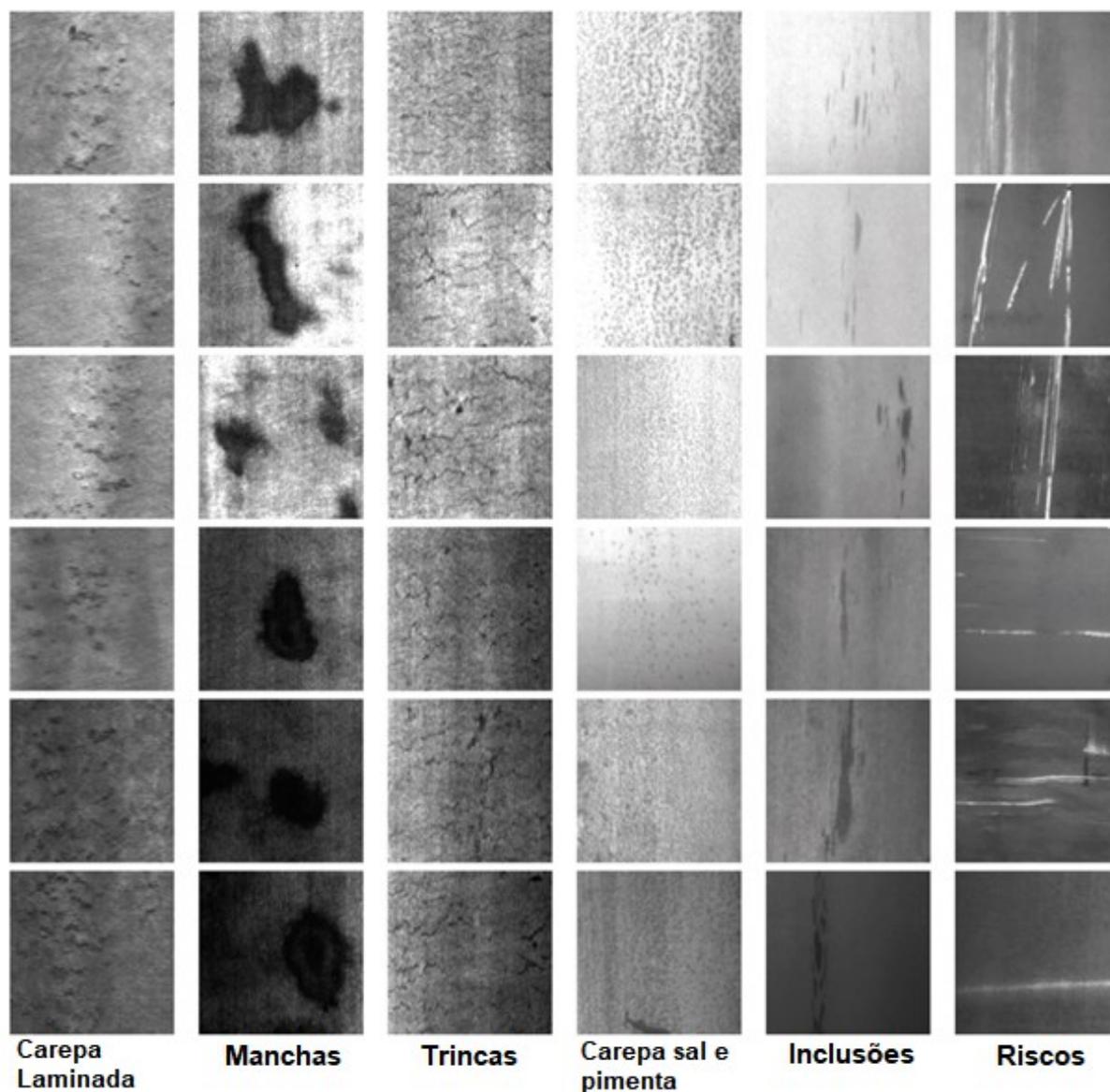
A dificuldade em obter dados representativos de defeitos em placas de aço laminado a quente é uma questão relevante que precisa ser considerada em estudos nessa área. A colaboração entre a indústria siderúrgica e a comunidade acadêmica é essencial para superar esses desafios e obter conjuntos de dados adequados para análise e desenvolvimento de modelos de detecção de defeitos.

Foram utilizados os dados fornecidos pela Northeastern University, sendo amplamente utilizada em outros algoritmos e trabalhos com redes neurais, como no trabalho de (He et al, 2020)

A construção da base de dados fornecido pela Northeastern University (NEU), foi feita em seis tipos de defeitos superficiais em placas de aço laminadas a quente, as quais foram, rolled in scales (RS) ou Carepa laminada, patches (Pa) como Manchas, crazing (Cr) demonstrado por Trincas, pitted surface (PS) utilizada como Carepa tipo sal e pimenta, inclusion (In) caracterizado por inclusões, e scratches (Sc) sendo reconhecida por riscos. A base de dados inclui 1.800 imagens em escala de cinza, como na Figura 52, onde é possível observar exemplos de imagens de seis defeitos diferentes com resolução original de 200x200 pixels. Sendo possível observar nas classes internas dos defeitos, como os riscos podendo ser tanto na horizontal quanto na vertical, existindo uma grande diferença na aparência, enquanto as classes macro se diferindo em iluminação, e diferenças em materiais. Tendo como desafio, as classes internas com uma diferença muito grande de imagens, e as classes macros com defeitos parecidos. (Song, 2014)

A base de dados separa em 300 imagens para cada defeito, tendo 72 imagens separadas para validação e 72 imagens para teste de modelos.

Figura 52 - Base de dados



3.2 Desenvolvimento

3.2.1 Desenvolvimento dos algoritmos

Para o desenvolvimento foram revisadas atentamente as características da operação sugerida, uma rede neural, capaz de identificar defeitos em placas de aço laminadas a quente.

O código começa importando as bibliotecas necessárias. Essas bibliotecas são essenciais para o modelo utilizado, visando a verificação de imagens de placas de aço laminados a quente, a fim de encontrar defeitos de superfície, além de validar o código posteriormente, como podemos observar por código demonstrado na Figura 53.

Primeiramente foi importado o programa tensorflow, uma estrutura popular de aprendizado profundo de código aberto, o qual nos permite importar redes neurais pré-treinadas.

Em seguida foi utilizado o tensorflow.keras.preprocessing.image para importar o ImageDataGenerator, o qual nos providencia as ferramentas necessárias para melhorar as imagens coletadas na base de dados utilizada, além de carregar as mesmas para o treino da rede neural.

Posteriormente o tensorflow.keras.applications.vgg16, onde foi importado uma ferramenta crucial para o desenvolvimento do algoritmo, a VGG16, onde conseguimos carregar uma rede neural pré-treinada conhecida por classificar imagens, assim conseguindo um resultado aprimorado, visto que a VGG16 é treinada em uma base de dados da ImageNET.

Novamente foi utilizado o tensorflow.keras.models importando o sequential, um modelo utilizado para construção de redes neurais, fazendo com que as camadas do modelo fiquem em uma sequência.

Após foi importado do tensorflow.keras.layers, a Dense e a Flatten, as quais são camadas utilizadas na rede neural, onde a camada Dense é uma camada totalmente conectada e a Flatten é utilizada para remodelar a camada de saída do modelo VGG16.

Finalmente foi importado a biblioteca numpy, que foi utilizada para operações matemáticas, além de matrizes.

Figura 53 - Importando bibliotecas da rede neural

```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten
import numpy as np
```

Fonte: Autoria própria (2023)

No código foi utilizada a `tf.random.set seed(42)`, Figura 54, assim a geração aleatória ficou estabelecida no número 42, desta maneira processos dentro do tensorflow, como os pesos de início ou embaralhamento da base de dados, ficariam com a mesma sequência de dados aleatórios, toda vez que o código fosse executado.

Figura 54 - Geração aleatória

```
# Configurando a geração aleatória.
tf.random.set seed(42)
```

Fonte: Autoria própria (2023)

Em seguida foram fornecidas as pastas que o programa vai ter que acessar a fim de ter acesso a base de dados de treinamento e de validação, Figura 55, os quais serão ambos utilizados no programa.

Figura 55 - Diretórios

```
# Computando os caminhos para a base de dados de treinamento e validação
train_dir = r'D:\Vini\TCC\base de dados\NEU Metal Surface Defects Data\train'
valid_dir = r'D:\Vini\TCC\base de dados\NEU Metal Surface Defects Data\valid'
```

Fonte: Autoria própria (2023)

O tamanho do lote escolhido foi de 32, visto que não possuímos uma base de dados grande, entretanto foi um número de períodos de 10, logo foram passados dez vezes pela base de dados de treinamento sendo assim a rede neural consegue realizar a sua função com eficiência e qualidade, como demonstrado em Figura 56.

Figura 56 - Hyper parâmetros

```
# Colocando os Hyper parâmetros
batch_size = 32
epochs = 10
```

Fonte: Autoria própria (2023)

Neste caso foi feita uma alteração na escala das imagens, além de rotacionar as mesmas, aumentar a espessura e comprimento, colocar zoom, rotacionar horizontalmente e preencher, Figura 57, fazendo assim nossa pequena quantidade de dados, parecer uma quantidade suficiente.

Pré-processamento é o redimensionamento das imagens. O redimensionamento consiste em ajustar todas as imagens para um tamanho padrão, a fim de garantir que a entrada da rede neural tenha dimensões consistentes. De acordo com Egmont, 2002, o redimensionamento das imagens pode ajudar a reduzir o processamento desnecessário e facilitar a extração de características relevantes.

Além disso, a normalização é uma técnica amplamente utilizada no pré-processamento de imagens. As imagens foram normalizadas dividindo cada pixel pelo valor máximo de intensidade, ou seja, 255. Essa técnica é útil para garantir que todos os valores de pixel estejam na mesma faixa, o que ajuda no treinamento eficiente da rede neural.

Figura 57 - Melhoramento de imagens

```
# definindo o melhoramento de dados
train datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
```

Fonte: Autoria própria (2023)

O treinamento e a validação de base de dados são carregados usando o programa ImageDataGenerator. Essa ferramenta prepara os dados para treinamento,

aplicando a melhora de imagem nos dados de treinamento e remodelando ambos para 1/255 do tamanho original.

Neste caso, foram aplicadas algumas maneiras de aumentar ainda mais a quantidade de dados, fazendo o embaralhamento dos mesmos. Após isso os dados de validação do treinamento foram carregados, Figura 58.

Figura 58 - Base de dados para treinamento

```
valid datagen = ImageDataGenerator(rescale=1./255)

# carregando a base de dados de treinamento
train data = train datagen.flow from directory(
    train dir,
    target size=(224, 224),
    batch size=batch size,
    class mode='categorical',
    shuffle=True,
    seed=42
)
```

Fonte: Aatoria própria (2023)

Então foi inserido a VGG16 no código, excluindo a camada superior, visando que utilizaremos classificações diferentes da qual a VGG16 fora treinada, assim classificando os nossos dados de acordo com a necessidade do projeto, Figura 59.

Figura 59 - VGG16

```
# Inserir a VGG16 porem excluindo a camada superior.
base model = VGG16(weights='imagenet', include_top=False, input_shape=(224,
224, 3))
```

Fonte: Aatoria própria (2023)

Congelando as camadas pré-treinadas da VGG16, visando fazer uma transferência de conhecimento entre as redes neurais, onde a rede neural pré-treinada é adaptada para uma nova tarefa, congelar as camadas pré-treinadas é fundamental para integridade da rede neural, mantendo assim os mecanismos da rede pré-treinada.

A VGG16, uma rede neural pré-treinada conhecida por classificar imagens, serve como uma base para a nova tarefa estipulada de classificar defeitos na camada superficial de placas de aço laminadas a quente.

Para garantir as características aprendidas pela rede neural pré-treinada, na VGG16, o algoritmo congela as camadas, assim os pesos e parâmetros não são atualizados durante o treinamento. Como resultado o modelo apenas adiciona novos parâmetros e camadas para a classificação de defeitos, enquanto as camadas pré-treinadas agem como uma base para extração de pesos e características.

Pelo congelamento de camadas pré-treinadas, Figura 60, o modelo pode efetivamente ser construído em cima das características de uma rede pré-treinada.

Figura 60 - Congelamento de camadas

```
# Congelamento das camadas pré treinadas
for layer in base_model.layers:
    layer.trainable = False
```

Fonte: Autoria própria (2023)

Seguindo a base de modelo da VGG16, uma camada Flatten é introduzida. Essa camada remodela a saída da camada pré treinada em um vetor unidimensional. Essa transformação é essencial para conectar os mapas convolucionais com a subsequente camada completamente conectada.

Após a camada Flatten foram adicionadas duas camadas Dense, Figura 61, as camadas densas são conhecidas como camadas totalmente conectadas, normalmente utilizadas nos estágios finais da tarefa de classificação, nas camadas adicionadas.

Uma vez que foi definida a arquitetura da rede neural de forma que a primeira camada personalizada é a camada Flatten. Essa camada é usada para converter o tensor tridimensional resultante das camadas convolucionais em um vetor unidimensional. Isso é necessário porque as camadas Dense (totalmente conectadas) que virão a seguir requerem entradas unidimensionais.

Após a camada Flatten, uma camada totalmente conectada (Dense) é adicionada com 256 unidades. A ativação ReLU (Rectified Linear Unit) é aplicada a

cada unidade. Essa camada tem a tarefa de aprender representações mais abstratas e complexas das características extraídas pelas camadas convolucionais.

A última camada é outra camada totalmente conectada (Dense) com 6 unidades, que corresponde ao número de classes. A ativação Softmax é usada nesta camada para converter as saídas em probabilidades, indicando a probabilidade de cada classe, o próximo passo é o de compilação do modelo

Figura 61 – Camada de classificação

```
# Adicionando uma nova camada de classificação
model = Sequential()
model.add(base_model)
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dense(6, activation='softmax'))
```

Fonte: Autoria própria (2023)

A compilação envolve especificamente componentes que vão guiar o treinamento da rede neural, como otimização, função de perda e métricas de validação.

Otimização é responsável pelo ajuste dos pesos dos modelos e vieses durante o treinamento, minimizando a função de perda, foi utilizado o otimizador adam, o qual é conhecido pela eficiência em treinamento de redes neurais, o qual combina mecanismos (Zhang, 2018).

A escolha da função de perda é crucial e depende de qual função é exigida da rede neural, foi utilizada uma cross-entropy, Figura 62, a qual é normalmente utilizada para rede neurais com classificação de múltiplas classes. Ela quantifica e diferencia a classe prevista e a classe atual, sendo apropriada para otimização do modelo de classificação de defeitos na superfície de metais (Zhang, 2018).

A validação é usada para verificar a performance do modelo durante e depois do treinamento, recorreremos a acurácia como métrica de validação. Acurácia mede a proporção de eventuais classificações corretas dentro do total de classificações, provendo uma medida de performance do modelo.

Figura 62 - Compilação

```
# Compilando o modelo
model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
```

Fonte: Autoria própria (2023)

O processo de treinar uma rede neural, envolve iterações e atualizações dos parâmetros do modelo, como pesos e vieses, para minimizar a função de perda e fazer uma rede precisa.

O método de treinamento utilizado foi o fit, Figura 63, esse método atualiza os pesos utilizando o mecanismo de backpropagation e otimizadores, neste caso o Adam, esse treinamento é um processo fundamental na criação da rede neural.

Figura 63 - Treinamento

```
# Treinamento do modelo
history = model.fit(
    train data,
    epochs=epochs,
    batch size=batch size,
    validation data=valid data
)
```

Fonte: Autoria própria (2023)

Após o treinamento o modelo é salvo no diretório requisitado, Figura 64, onde podemos observar no código fornecido.

Figura 64 - Salvando o modelo

```
# Salvar o modelo
model.save('D:\Vini\TCC\saved_model.h5')
```

Fonte: Autoria própria (2023)

A performance do modelo é validada na base de dados para validação, Figura 65, sendo mostrado na tela a acurácia e a perda, após a validação.

Figura 65 - Critérios de validação

```
# Validação do modelo
loss, accuracy = model.evaluate(valid data)
print('Validation Loss:', loss)
print('Validation Accuracy:', accuracy)
```

Fonte: Autoria própria (2023)

O código em seguida recolhe os dados da validação e os calcula separando as previsões por classes, Figura 66, sendo assim, obtém-se um resultado na validação para compreender a acurácia e possíveis erros do projeto.

Assim, o algoritmo apresenta todos os processos de construção de uma rede neural, compreendendo o treinamento e validação de uma rede neural para classificação de imagens utilizando o modelo pré treinado (VGG16) e melhoramento de dados.

Figura 66 - Validação

```

# Coletar previsões da validação do modelo
valid_data.reset()
predictions = model.predict(valid_data)
predicted_labels = np.argmax(predictions, axis=1)
true_labels = valid_data.classes

# Coletar as classes
class_labels = list(valid_data.class_indices.keys())

# Calcula o número de previsões e de acertos de cada classe
class_counts = {}
for i in range(len(class_labels)):
    class_name = class_labels[i]
    true_indices = np.where(true_labels == i)[0]
    predicted_indices = np.where(predicted_labels == i)[0]
    correct_count = np.intersect1d(true_indices, predicted_indices).size
    wrong_count = true_indices.size - correct_count
    class_counts[class_name] = {'Correct': correct_count, 'Wrong':
wrong_count}

# Mostra os resultados para cada classe
for class_name, counts in class_counts.items():
    print('Class:', class_name)
    print('Correctly Classified Images:', counts)

```

Fonte: Autoria própria (2023)

3.2.2 Teste da rede neural treinada

Após a validação e treinamento, uma parte crucial do projeto é a fase de testes da rede neural, para observarmos a eficiência em uma base de dados nunca vista, para isso utilizamos outro algoritmo.

Foram importadas as bibliotecas necessárias para o desenvolvimento do código de teste, Figura 67, sendo este utilizado para validar e apresentar resultados reais.

A biblioteca os foi utilizada para funções relacionadas com o sistema operacional, neste caso abrir a janela de gráfico e da matriz confusão.

O programa Numpy foi fundamental para a fabricação matemática dos gráficos a serem apresentados.

O programa Tensorflow.keras providenciou ferramentas para carregarmos a rede neural treinada.

O programa Matplotlib.pyplot forneceu uma biblioteca para visualização dos resultados matemáticos em forma de gráficos.

O programa Sklearn.metrics ajudou a medir a performance do modelo, incluindo a matriz de confusão.

Figura 67 - Importando bibliotecas para teste

```
import os
import numpy as np
from tensorflow.keras.preprocessing.image import load_img, img_to_array
from tensorflow.keras.models import load_model
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, classification_report
```

Fonte: Autoria própria (2023)

Primeiramente foi computado os diretórios do modelo salvo e da base de dados de teste fornecida pela NEU, Figura 68, (Universidade de Northeastern) Metal Surface Defects Data.

Figura 68 - Diretórios de teste e modelo

```
# Diretório para de teste e modelo
test_dir = r'D:\Vini\TCC\base de dados\NEU Metal Surface Defects Data\test'
model_dir = r'D:\Vini\TCC\saved_model.h5'
```

Fonte: Autoria própria (2023)

Carregando o modelo salvo após o treinamento e validação, Figura 69, assim sendo utilizado para teste.

Figura 69 - Carregando modelo

```
# Carregando o modelo
model = load_model(model_dir)
```

Fonte: Autoria própria (2023)

Definindo as classes para a caracterização do projeto, sendo utilizado os nomes em inglês dos mesmos, para não haver alteração na base de dados, sendo estes Crazing (Trincas), Inclusion (inclusões), Patches (manchas), Pitted (Sal e pimenta), Rolled (Carepa laminada) e Scratches (riscos), Figura 70.

Figura 70 - Definição de classes

```
# Definindo as classes
class_names = ['Crazing', 'Inclusion', 'Patches', 'Pitted', 'Rolled',
'Scratches']
```

Fonte: Autoria própria (2023)

Após o estabelecimento das classes, foram definidas as variáveis para matriz confusão, Figura 71.

Figura 71 - Variáveis da matriz de confusão e rótulos

```
# Inicializar rótulos reais e previstos
true_labels = []
predicted_labels = []

# Variáveis da matriz de confusão
num_classes = len(class_names)
confusion_matrix_array = np.zeros((num_classes, num_classes))
```

Fonte: Autoria própria (2023)

O código faz a iteração entre as imagens de teste para cada classe, dentro do loop, ele processa cada imagem, carregando a mesma, após isso remodelando a imagem para o tamanho de 224x224 pixels, e convertendo para um arranjo numérico para a rede neural. O valor dos pixels também é normalizado dividindo por 255.

A imagem carregada passa pela rede neural treinada, onde a rede neural faz uma previsão da classe da imagem. A classe atual da imagem é determinada pelo loop do nome da classe, ambos a previsão e a atual classe são mostradas, Figura 72.

Figura 72 - Iteração na base de dados de teste

```

# Iteração pelas imagens do teste
for class name in class names:
    class path = os.path.join(test dir, class name)
    image files = os.listdir(class path)

    for image file in image files:
        image path = os.path.join(class path, image file)
        image = load_img(image path, target_size=(224, 224))
        image array = img to array(image)
        image array = image array / 255.0
        image array = np.expand dims(image array, axis=0)

        # Predizer a classe
        predicted label = model.predict(image array)
        class index = np.argmax(predicted label)
        predicted class name = class names[class index]

        true labels.append(class names.index(class name))
        predicted labels.append(class index)

        print(f"Actual Class: {class name}, Predicted Class:
        {predicted class name}")

```

Fonte: Autoria própria (2023)

Após o processamento de imagem para cada classe específica, o código atualiza a matriz de confusão. Isso aumenta a contagem de previsão da classe e a classe atual, dentro da variável da matriz confusão, esse processo se repete para cada classe.

O Código utiliza matplotlib para criar um mapa de calor da matriz de confusão normalizada, (Figura 73). Este mapa apresenta o percentual de acurácia do modelo e a previsão para cada classe, providenciando uma boa noção de como o modelo está classificando todas as imagens.

Figura 73 - Plotagem da matriz confusão

```

# Criar a matriz de confusão
confusion_mat = confusion_matrix(true_labels, predicted_labels,
labels=np.arange(num_classes))

# Plotar a matriz de confusão com valores anotados
fig, ax = plt.subplots()
im = ax.imshow(confusion_mat, interpolation='nearest', cmap=plt.cm.Blues)
ax.figure.colorbar(im, ax=ax)
ax.set(xticks=np.arange(confusion_mat.shape[1]),
yticks=np.arange(confusion_mat.shape[0]),
xticklabels=class_names, yticklabels=class_names,
title='Confusion Matrix',
ylabel='True Class',
xlabel='Predicted Class')

# Adicionar anotações à matriz de confusão
for i in range(num_classes):
    for j in range(num_classes):
        text = ax.text(j, i, f'{confusion_mat[i, j]:.0f}', ha='center',
va='center', color='black')

```

Fonte: Autoria própria (2023)

E por fim calcular e retornar o resultado da F1-score, (Figura 74) outra forma de controle de acertos da rede neural.

Figura 74 - Plotagem F1-score

```

# Calcular e mostrar o resultado da f1-score
classification_rep = classification_report(true_labels, predicted_labels,
labels=np.arange(num_classes), target_names=class_names)
print("Classification Report:\n", classification_rep)

# Mostrar o gráfico
plt.show()

```

Fonte: Autoria própria (2023)

4 RESULTADOS E DISCUSSÕES

A avaliação dos resultados é uma etapa fundamental no processo de detecção de defeitos superficiais em placas de aço laminado a quente. Após a execução do modelo de detecção, são obtidos os resultados preditos para cada

imagem de teste. Esses resultados são então comparados com as classes verdadeiras das imagens para avaliar o desempenho do modelo.

No período de validação da rede neural, foi utilizado o código para fazer a validação e nos retornar os valores fazendo jus a acurácia, Figura 75, com isso observa-se que das 72 imagens selecionadas para validação a rede neural errou uma, tendo 98,61% de acurácia durante a validação.

Figura 75 - Acurácia da validação

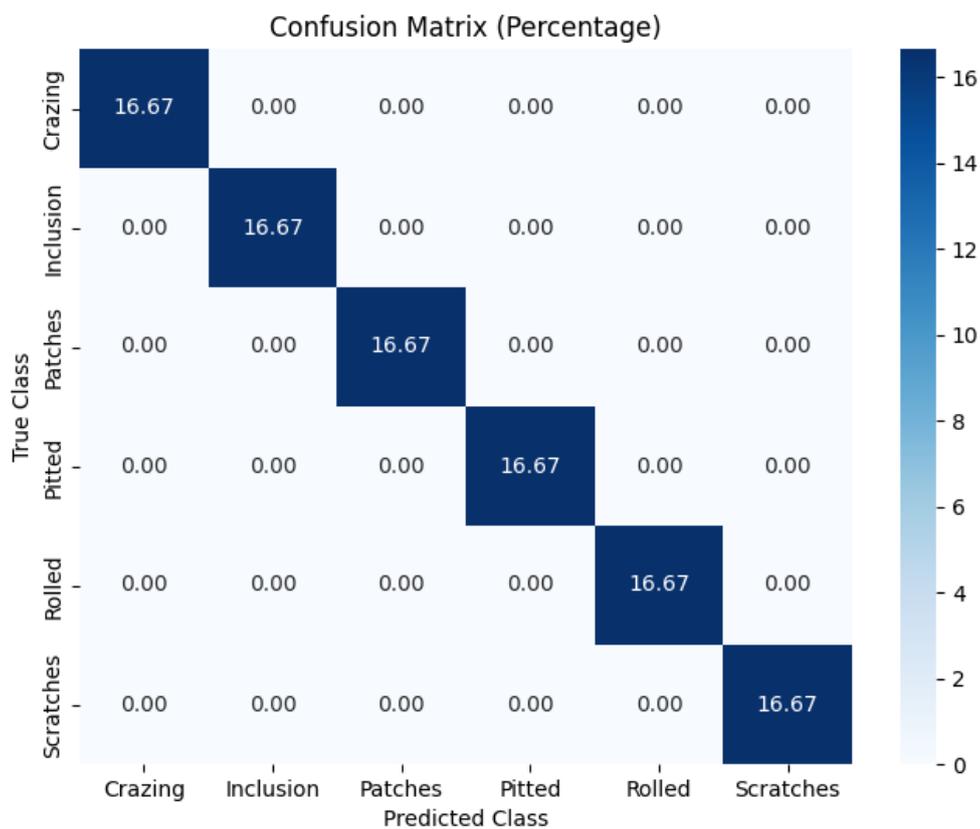
```
C:\Users\vinic\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\src\engine\training.py:3079:
considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')`.
  saving_api.save_model(
3/3 [=====] - 5s 1s/step - loss: 0.0161 - accuracy: 0.9861
Validation Loss: 0.01607760600745678
Validation Accuracy: 0.9861111044883728
3/3 [=====] - 5s 1s/step
Class: Crazying
Correctly Classified Images: {'Correct': 12, 'Wrong': 0}
Class: Inclusion
Correctly Classified Images: {'Correct': 12, 'Wrong': 0}
Class: Patches
Correctly Classified Images: {'Correct': 11, 'Wrong': 1}
Class: Pitted
Correctly Classified Images: {'Correct': 12, 'Wrong': 0}
Class: Rolled
Correctly Classified Images: {'Correct': 12, 'Wrong': 0}
Class: Scratches
Correctly Classified Images: {'Correct': 12, 'Wrong': 0}
PS D:\Vini\TCC\Rede\Base de dados NEU\rede>
```

Fonte: Autoria própria (2023)

Partindo para o teste em si, observa-se que a rede neural, teve 100% de acurácia no teste em 72 imagens, as quais estavam separadas do montante treinado desde o início do projeto, dentro da base de dados utilizada.

Considerando esses detalhes foram analisados os resultados obtidos pela rede neural, através de uma matriz de confusão, juntamente com uma curva e um F1-score (Figura 77).

Figura 76 - Matriz confusão



Fonte: Autoria própria (2023)

Figura 77 - F1-score

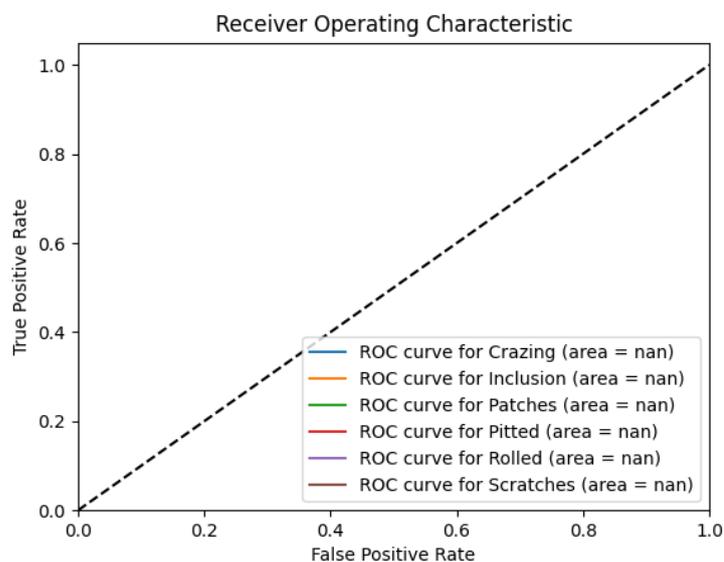
```

Classification Report:

```

	precision	recall	f1-score	support
Crazing	1.00	1.00	1.00	12
Inclusion	1.00	1.00	1.00	12
Patches	1.00	1.00	1.00	12
Pitted	1.00	1.00	1.00	12
Rolled	1.00	1.00	1.00	12
Scratches	1.00	1.00	1.00	12
accuracy			1.00	72
macro avg	1.00	1.00	1.00	72
weighted avg	1.00	1.00	1.00	72

Fonte: Autoria própria (2023)

Figura 78 - Curva ROC

Fonte: Autoria própria (2023)

Observa-se que a rede neural obteve 100% de acurácia, é um resultado bastante impressionante e indica que o modelo de detecção de defeitos utilizado no código possui uma capacidade excelente de classificar corretamente as imagens de teste. Isso significa que o modelo foi capaz de identificar e distinguir com precisão os diferentes tipos de defeitos nas placas de aço laminado a quente.

Wang 2021, utilizou para os seis tipos de defeito, 50% da base de dados para treino, no caso 150 imagens para cada defeito, selecionadas de forma aleatória, as separadas foram usadas para validação e teste, tendo obtido um resultado impressionante de 99.63% de acurácia, mesmo com uma base de dados pequena, isto pode ter impactado em seu resultado, mas ao mesmo tempo tendo uma apuração mais concreta, visto a sua seleção maior para teste.

Já Zhao, (2023) para aprimorar a acurácia de sua rede neural, utilizou outras métricas de desenvolvimento da rede neural, sendo aplicada modelos totalmente diferentes dos apresentados, estes de rede neural convolucional profunda em atenção paralela, conseguindo uma acurácia de 99.57%, 99.6% e 99.8%

5 CONCLUSÃO

Neste trabalho foi desenvolvida uma rede neural convolucional para a classificação de defeitos superficiais em chapas laminadas a quente. Foram utilizadas 1800 imagens de um banco de dados construído pela Northeastern University.

A validação da rede neural nos mostrou um desempenho funcional ótimo da rede neural, tendo como resultado 98,61% de acurácia na base de dados de validação, a qual foi extraída da base de dados para treinamento, tendo sido validada em 72 imagens, com erro em apenas 1 imagem.

O teste utilizando a base de dados NEU, com 72 imagens selecionadas mostrou um excelente desempenho tendo como resultado 100% de acuraria, indicada através de matriz de confusão, F1-Score e curva ROC, significando que todas as imagens foram identificadas e classificadas.

Os resultados foram obtidos a partir de uma base de dados relativamente pequena, mas se credenciam o modelo a ser testado em bases de dados maiores e mais detalhadas e para ser utilizado em situações práticas na indústria siderúrgica.

É fundamental continuar aprimorando e refinando o modelo de detecção de defeitos, buscando constantemente melhorias na sua capacidade de generalização e na detecção de casos mais desafiadores. A aplicação de técnicas avançadas de aprendizado de máquina, a expansão do conjunto de dados de treinamento e a utilização de estratégias de validação e teste mais robustas podem ser consideradas.

REFERÊNCIAS

Agarap, Abien Fred. Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375, 2018.

Albawi, Saad; MOHAMMED, Tareq Abed; AL-ZAWI, Saad. Understanding of a convolutional neural network. In: 2017 international conference on engineering and technology (ICET). Ieee, 2017. p. 1-6.

Ahmad, Zubair et al. Artificial intelligence (AI) in medicine, current applications and future role with special emphasis on its potential and promise in pathology: present and future impact, obstacles including costs and acceptance among pathologists, practical and philosophical considerations. A comprehensive review. Diagnostic pathology, v. 16, p. 1-16, 2021.

Associação Brasileira de Normas Técnicas. ABNT NBR 16552: Produtos planos laminados. Rio de Janeiro: ABNT, 2016.

Ashour, M. W, Khalid, F, Abdul Halin, A, Abdullah, L. N, & Darwish, S. H. (2019). Surface defects classification of hot-rolled steel strips using multi-directional shearlet features. Arabian Journal for Science and Engineering, v. 44, p. 2925-2932.

Baselice, F, Ferraioli, G, Ambrosanio, M, Pascazio, V, & Schirinzi, G. (2018). Enhanced Wiener filter for ultrasound image restoration. Computer methods and programs in biomedicine, v. 153, p. 71-81.

Behrenbruch, C. P, Petroudi, S, Bond, S, Declerck, J. D, Leong, F. J, & Brady, J. M. (2004). Image filtering techniques for medical image post-processing: an overview. The British journal of radiology, v. 77, n. 2, p. S126-S132.

Biles, John; Anderson, Peter; Loggi, Laura. (1996). Neural network fitness functions for a musical IGA.

Bridle, John S. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In: Neurocomputing: Algorithms, architectures and applications. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990. p. 227-236.

Chen, R. Y.; Yeun, W. Y. D. (2003). Review of the high-temperature oxidation of iron and carbon steels in air or oxygen. Oxidation of metals, v. 59, n. 5-6, p. 433-468.

Das, Prasun; Bhattacharyay, Bidyut Kumar; Datta, Shubhabrata. (2006). A comparative study for modeling of hot-rolled steel plate classification using a statistical approach and neural-net systems. Materials and manufacturing processes, v. 21, n. 8, p. 747-755.

Devijver, Pierre A.; Kittler, Josef. (1982). Pattern recognition: A statistical approach.

Dey, Arthita; BHATTACHARYYA, Tanmay; DHARA, Sharmistha. Characterization of hot rolled scales-root cause identification and remedial action. *Engineering Failure Analysis*, v. 34, p. 478-487, 2013.

Egmont, Petersen Michael; De Ridder, Dick; Handels, Heinz. (2002). Image processing with neural networks—a review. *Pattern recognition*, v. 35, n. 10, p. 2279-2301.

Forsyth, David A.; Ponce, Jean. (2002). *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference.

Gambäck, Björn; SIKDAR, Utpal Kumar. Using convolutional neural networks to classify hate-speech. In: *Proceedings of the first workshop on abusive language online*. 2017. p. 85-90.

Ghorai, S, Mukherjee, A, Gangadaran, M, & Dutta, P. K. (2012). Automatic defect detection on hot-rolled flat steel products. *IEEE Transactions on Instrumentation and Measurement*, v. 62, n. 3, p. 612-621.

Gu, J, Wang, Z, Kuen, J, Ma, L, Shahroudy, A, Shuai, B, ... & Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern recognition*, v. 77, p. 354-377.

Hu, Ruofei et al. Efficient hardware architecture of softmax layer in deep neural network. In: *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*. IEEE, 2018. p. 1-5.

ImageNet (repositório online de imagens). Large Scale Visual Recognition Challenge 2014. Disponível em <https://www.image-net.org/challenges/LSVRC/2014/results.php>. Acesso em: 10 mar. 2022

Jain, Anil K.; MAO, Jianchang; MOHIUDDIN, K. Moidin. Artificial neural networks: A tutorial. *Computer*, v. 29, n. 3, p. 31-44, 1996.

Jin, Won; Choi, Jeom-Yong; Lee, Yun-Yong. (2000). Nucleation and growth process of sticking particles in ferritic stainless steel. *ISIJ international*, v. 40, n. 8, p. 789-793.

Khalid, F, Abdul Halin, A, Abdullah, L. N, & Darwish, S. H. (2019). Surface defects classification of hot-rolled steel strips using multi-directional shearlet features. *Arabian Journal for Science and Engineering*, v. 44, p. 2925-2932.

Kingma, Diederik P.; BA, Jimmy. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

Krstinić, Damir et al. Multi-label classifier performance evaluation with confusion matrix. *Computer Science & Information Technology*, v. 1, 2020.

Kurita, Takio. (2019). Principal component analysis (PCA). *Computer Vision: A Reference Guide*, p. 1-4.

Le, Quoc V. (2013). Building high-level features using large scale unsupervised learning. In: 2013 IEEE international conference on acoustics, speech and signal processing. IEEE, p. 8595-8598.

LI, Li; DOROSLOVAČKI, Miloš; LOEW, Murray H. Approximating the gradient of cross-entropy loss function. IEEE access, v. 8, p. 111626-111635, 2020.

Li, Z, Liu, F, Yang, W, Peng, S, & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. IEEE transactions on neural networks and learning systems.

Liu, H, Simonyan, K, Vinyals, O, Fernando, C, & Kavukcuoglu, K. (2017). Hierarchical representations for efficient architecture search. arXiv preprint arXiv:1711.00436.

Luo, Chunhui; Ståhlberg, Ulf. (2001). Deformation of inclusions during hot rolling of steels. Journal of Materials Processing Technology, v. 114, n. 1, p. 87-97.

Lukac, Rastislav; Plataniotis, Konstantinos N. (2018). Color image processing: methods and applications. CRC Press.

Martinez, Manuel; STIEFELHAGEN, Rainer. Taming the cross entropy loss. In: Pattern Recognition: 40th German Conference, GCPR 2018, Stuttgart, Germany, October 9-12, 2018, Proceedings 40. Springer International Publishing, 2019. p. 628-637.

Martins, Luiz AO; PÁDUA, Flávio LC; DE ALMEIDA, Paulo EM. Sistema Baseado em Rede SOM para Inspeção Visual Automática de Defeitos em Aços Laminados. Learning and Nonlinear Models, n. 7, p. 71, 2009.

Mahesh, Batta. (2020). Machine learning algorithms-a review. International Journal of Science and Research (IJSR), v. 9, n. 1, p. 381-386. https://www.researchgate.net/profile/Batta-Mahesh/publication/344717762_Machine_Learning_Algorithms_-_A_Review/links/5f8b2365299bf1b53e2d243a/Machine-Learning-Algorithms-A-Review.pdf?eid=5082902844932096t, acessado em 08/09/2023.

Mallat, Stéphane. (1999). A wavelet tour of signal processing. Elsevier.

Mcculloch, Warren S.; PITTS, Walter. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, v. 5, p. 115-133, 1943.

Medsker, Larry R.; Jain, L. C. (2001). Recurrent neural networks. Design and Applications, v. 5, n. 64-67, p. 2.

Meng, G. Z.; Zhang, C.; Cheng, Y. F. (2008). Effects of corrosion product deposit on the subsequent cathodic and anodic reactions of X-70 steel in near-neutral pH solution. Corrosion Science, v. 50, n. 11, p. 3116-3122.

Mintz, B.; Morrison, W. B. (1988). Influence of warm working and tempering on fissure formation. Materials science and technology, v. 4, n. 8, p. 719-731.

Mintz, B. (1980). Influence of grain boundaries on fissure formation during impact testing of ferritic stainless steels. *Metals Technology*, v. 7, n. 1, p. 127-129.

NEU (Northeastern University)
http://faculty.neu.edu.cn/yunhyan/NEU_surface_defect_database.html, acessado em 05/06/2023.

Pathak, Purnanand; Das, Ghanshyam; Jha, Sanjay Kumar. (2022). Influence of Roll Wear in Hot Rolling of Steel at Hot Strip Mills. In: *Recent Advances in Manufacturing Processes: Select Proceedings of RDMPMC 2020*. Springer Singapore, p. 153-169.

Phillip, Stefan. Design and implementation of a multi-class network architecture for hardware neural networks. 2008. Tese de Doutorado.

Roberts, William L. (1983). *Hot rolling of steel*. CRC Press.

Schwenk, Holger; Bengio, Yoshua. (2000). Boosting neural networks. *Neural computation*, v. 12, n. 8, p. 1869-1887.

Scardapane, Simone; Wang, Dianhui. (2017). Randomness in neural networks: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, v. 7, n. 2, p. e1200.

Sekine, Hiroshi; Tanaka, Tomo; Tamura, Imao. (1988). *Thermomechanical processing of high-strength low-alloy steels*. Butterworth-Heinemann.

Sharma, Vidushi; RAI, Sachin; DEV, Anurag. A comprehensive study of artificial neural networks. *International Journal of Advanced research in computer science and software engineering*, v. 2, n. 10, 2012.

SIGANOS, Dimitrios; STERGIOU, Christos. *Neural networks*. Google Search, 1996.

Singh, Pravendra; RAJ, Prem; NAMBOODIRI, Vinay P. EDS pooling layer. *Image and Vision Computing*, v. 98, p. 103923, 2020.

Smith, Leslie N. (2018). A disciplined approach to neural network hyper-parameters: Part 1--learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*.

SONG, Kechen; HU, Shaopeng; YAN, Yunhui. Automatic recognition of surface defects on hot-rolled steel strip using scattering convolution network. *J. Comput. Inf. Syst*, v. 10, n. 7, p. 3049-3055, 2014.

Su, Jiang; Zhang, Harry. (2006). A fast decision tree learning algorithm. In: *Aaai*, p. 500-505.

Tang, Bo; Chen, Li; Sun, Wei; Lin, Zhong-kang. (2023). Review of surface defect detection of steel products based on machine vision. *IET Image Processing*, v. 17, n. 2, p. 303-322.

Talab, Ahmed Mahgoub Ahmed et al. (2016). Detection crack in image using Otsu method and multiple filtering in image processing techniques. *Optik*, v. 127, n. 3, p. 1030-1033.

Taylor, Brian J. (Ed.). (2006). *Methods and procedures for the verification and validation of artificial neural networks*. Springer Science & Business Media.

Tu, Fengbin et al. (2017). Deep convolutional neural network architecture with reconfigurable computation patterns. *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, v. 25, n. 8, p. 2220-2233.

Twomey, Janet M. et al. (1997). Validation and verification. *Artificial neural networks for civil engineers: Fundamentals and applications*, p. 44-64.

Wang, Sun-Chong. (2003). *Artificial neural network. Interdisciplinary computing in java programming*, p. 81-100.

Wang, Wenyan et al. (2021). Surface defects classification of hot rolled strip based on improved convolutional neural network. *ISIJ International*, v. 61, n. 5, p. 1579-1583.

Weiss, Karl; KHOSHGOFTAAR, Taghi M.; WANG, DingDing. A survey of transfer learning. *Journal of Big data*, v. 3, n. 1, p. 1-40, 2016.

Wilamowski, Bogdan M. (2009). *Neural network architectures and learning algorithms*. *IEEE Industrial Electronics Magazine*, v. 3, n. 4, p. 56-63.

What is pooling? | CNN's #3 (2021), 1 video (1 min e 50). publicado pelo canal IntuitiveML. Disponível em: <https://www.youtube.com/watch?v=KKmCnwGzSv8>. Acesso em: 22 de setembro de 2023.

Woods, Kevin; BOWYER, Kevin W. Generating ROC curves for artificial neural networks. *IEEE Transactions on medical imaging*, v. 16, n. 3, p. 329-337, 1997.

WU, Yu-chen; FENG, Jun-wen. Development and application of artificial neural network. *Wireless Personal Communications*, v. 102, p. 1645-1656, 2018.

Yin, C, Zhu, Y, Fei, J, & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*, v. 5, p. 21954-21961.

Zhang, Z, & Zhang, Y. (2019). Automatic defect detection in steel plates using deep learning. *IEEE Access*, 7, 104669-104678.

Zhao, YuFeng; SUN, XiaoLei; YANG, JiaXing. Automatic recognition of surface defects of hot rolled strip steel based on deep parallel attention convolution neural network. *Materials Letters*, v. 353, p. 135313, 2023.

Zhang, Zhilu; Sabuncu, Mert. (2018). Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, v. 31.

Zhang, Zijun. (2018). Improved adam optimizer for deep neural networks. In: 2018 IEEE/ACM 26th international symposium on quality of service (IWQoS). IEEE, p. 1-2.

Zhu, Y, Zhang, Y, & Yin, C. (2017). A review on machine learning algorithms in wireless sensor networks. *Computers, Materials & Continua*, 54(2), 245–268.