

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas / Faculdade de Engenharia
Programa de Pós-Graduação em Modelagem Computacional

Cássio Danelon de Almeida

**Quantificação Automática de Microestruturas em Aços via Redes Neurais
Convolucionais**

Juiz de Fora

2020

Cássio Danelon de Almeida

**Quantificação Automática de Microestruturas em Aços via Redes Neurais
Convolucionais**

Dissertação apresentada ao Programa de Pós-Graduação em Modelagem Computacional da Universidade Federal de Juiz de Fora, na área de concentração em Sistemas Computacionais Aplicados, como requisito parcial para obtenção do título de Mestre em Modelagem Computacional.

Orientador: D.Sc. Leonardo Goliatt da Fonseca

Coorientador: D.Sc. Moisés Luiz Lagares Júnior

Juiz de Fora

2020

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Danelon, Cássio.

Quantificação Automática de Microestruturas em Aços via Redes Neurais
Convolucionais / **Cássio Danelon de Almeida**. – 2020.

61 f. : il.

Orientador: D.Sc. Leonardo Goliatt da Fonseca

Coorientador: D.Sc. Moisés Luiz Lagares Júnior

Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Instituto
de Ciências Exatas / Faculdade de Engenharia. Programa de Pós-Graduação
em Modelagem Computacional, 2020.

1. Quantificação de microestruturas. 2. Aprendizado de Máquina. 3.
Redes Neurais Convolucionais. I. Goliatt, Leonardo, orient. II. Lagares Jr.,
Moisés Luiz, coorient. III Título.

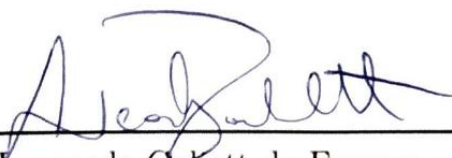
Cássio Danelon de Almeida

**Quantificação Automática de Microestruturas em Aços via Redes Neurais
Convolucionais**

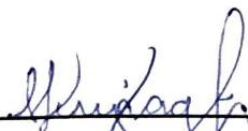
Dissertação apresentada ao Programa de Pós-Graduação em Modelagem Computacional da Universidade Federal de Juiz de Fora, na área de concentração em Sistemas Computacionais Aplicados, como requisito parcial para obtenção do título de Mestre em Modelagem Computacional.

Aprovado em: 04/03/2020

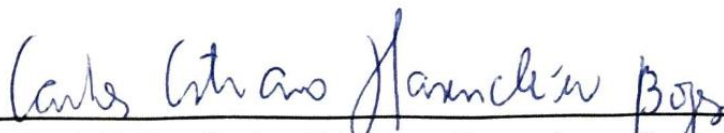
BANCA EXAMINADORA



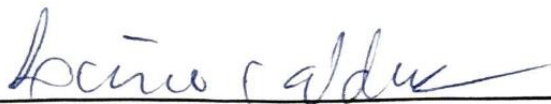
Prof. D.Sc. Leonardo Gohatt da Fonseca - Orientador
Universidade Federal de Juiz de Fora



Prof. D.Sc. Moisés Luiz Lagares Júnior - Coorientador
Universidade Federal de Juiz de Fora



Prof. D.Sc. Carlos Cristiano Hasenclever Borges
Universidade Federal de Juiz de Fora



Prof. D.Sc. Lecino Caldeira
Instituto Federal do Sudeste de Minas Gerais

AGRADECIMENTOS

Agradeço aos meus pais Paulo Roberto e Rita de Cássia, externo todo meu amor e gratidão. Talvez não existam palavras suficientes e significativas que me permitam agradecer vocês devidamente. Porém, tenham a certeza de que o suporte que vocês me ofereceram ao longo de toda minha formação foi para mim de valor inestimável. Agradeço também aos meus irmãos Eduardo e Elias, que sempre estiveram ao meu lado e me auxiliaram muito a completar mais esta etapa.

Externo também meus sinceros agradecimentos a todos que contribuíram direta ou indiretamente para a conclusão desta monografia; ao professor Leonardo Goliatt, pela atenciosa orientação e os grandes ensinamentos; ao coorientador Moisés Lagares pela importante ajuda no trabalho; aos professores Carlos Cristiano e Lecino Caldeira, pelas valiosas sugestões e contribuições feitas ao trabalho. Agradeço também ao meu amigo Thales Tozatto que me auxiliou muito no desafio desta dissertação.

Por fim, agradeço a todos meus amigos e funcionários do PGMG, à UFJF e à CAPES pelo apoio financeiro durante este período. Muito obrigado!

"Lembre-se de olhar para o alto, para as estrelas, e não para baixo, para os seus pés. Tente entender o que você vê e se pergunte o que faz o universo existir. Seja curioso. E por mais difícil que a vida possa parecer, sempre há algo que você possa fazer e obter sucesso. O importante é não desistir!"

Stephen Hawking

RESUMO

A análise microestrutural de um material permite a completa caracterização de suas propriedades mecânicas. Assim, o desempenho de um componente mecânico depende fortemente da identificação e quantificação de seus constituintes microestruturais. Atualmente, este processo ainda é feito manualmente por especialistas, tornando-o lento, muito trabalhoso e ineficiente. Estima-se que um especialista experiente leve em média 15 minutos por imagem para realizar a identificação e quantificação adequada de microconstituintes. Portanto, uma ferramenta computacional poderia ajudar bastante a melhorar o desempenho nesta tarefa. No entanto, uma vez que uma microestrutura pode ser uma combinação de diferentes fases ou constituintes com subestruturas complexas, sua quantificação automática pode ser muito difícil e, como resultado, há poucos trabalhos anteriores lidando com este problema. Redes Neurais Convolucionais são promissoras para este tipo de aplicação, já que recentemente esse tipo de rede tem alcançado grande performance em aplicações complexas de visão computacional. Neste trabalho, propomos uma quantificação automática de constituintes microestruturais de aço de baixo carbono via Redes Neurais Convolucionais. Nosso conjunto de dados consiste em 210 micrografias de aço de baixo carbono, e essa quantidade de imagens foi aumentada através de técnicas de aumento dos dados, resultando em um total de 720 amostras para treinamento. Com relação às arquiteturas de rede, foi utilizado as redes AlexNet e VGG16 treinadas do zero, e VGG19, Xception e InceptionV3 todas pré-treinadas. Os resultados mostraram que as CNNs podem quantificar microestruturas de forma muito eficaz.

Palavras-chave: Quantificação de microestruturas. Aprendizado de máquina. Redes Neurais Convolucionais.

ABSTRACT

The microstructural analysis of a material allows the complete characterization of its mechanical properties. Thus, the performance of a mechanical component depends heavily on the identification and quantification of its microstructural constituents. Currently, this process is still done mostly manually by experts, making it slow, very labor-intensive and inefficient. It is estimated that an experienced expert takes 15 minutes per image to perform the proper identification and quantification of microconstituents. Therefore, a computational tool could greatly assist to improve the performance in this task. However, since a microstructure can be a combination of different phases or constituents with complex substructures, their automatic quantification can be very hard and, as a result, there are few previous works dealing with this problem. Convolutional Neural Networks are promising for this type of application since recently this type of network has achieved great performance in complex applications of computational vision. In this work, we propose an automatic quantification of microstructural constituents of low carbon steel via Convolutional Neural Networks. Our dataset consists of 210 micrographs of low carbon steel, and this amount of images was increased through data augmentation techniques, resulting in a total of 720 samples for training. With regard to network architectures, we used AlexNet and VGG16 both trained from scratch, and three pre-trained models: VGG19, InceptionV3, and Xception. The results showed that CNNs can quantify microstructures very effectively.

Keywords: Microstructures quantification. Machine learning. Convolutional Neural Networks.

LISTA DE ILUSTRAÇÕES

Figura 1 – Micrografia de cordão de solda mostrando diferentes microconstituintes, segundo [31]	21
Figura 2 – Micrografia de cordão de solda do conjunto de dados analisado neste trabalho	21
Figura 3 – Fluxo de um algoritmo baseado em regras	22
Figura 4 – Fluxo de um algoritmo de aprendizado	23
Figura 5 – Áreas de inteligência computacional	23
Figura 6 – Gradiente de imagem demonstrando pixels variando de 0 (preto) até 255 (branco)	26
Figura 7 – Escala colorida de pixels RGB	27
Figura 8 – Diferença entre como nós percebemos uma imagem e como o computador a percebe	28
Figura 9 – Comparação entre métodos tradicionais de machine learning (a) e deep learning com redes neurais convolucionais (b) para análise e classificação de imagens	29
Figura 10 – Unidade básica de uma RNA	30
Figura 11 – Uma típica arquitetura de uma Rede Neural Artificial do tipo <i>multilayer feedforward</i> totalmente conectada	30
Figura 12 – Uma típica arquitetura de uma Rede Neural Convolucional	32
Figura 13 – Operação de convolução por filtros	32
Figura 14 – Operação de compressão de imagens via Max-Pooling	33
Figura 15 – Função Softmax	34
Figura 16 – O modelo AlexNet	39
Figura 17 – O modelo VGG16	39
Figura 18 – O modelo VGG19	40
Figura 19 – O módulo Inception	41
Figura 20 – O modelo Xception	42
Figura 21 – Exemplo de micrografia analisada neste trabalho. Microconstituintes: 36% PF(G), 3% PF(I), 16% AF, 34% FS(NA) e 11%FS(A)	43
Figura 22 – Imagem quadruplicada via técnicas de espelhamento	44
Figura 23 – A técnica de validação cruzada K-Fold	45
Figura 24 – Erro ao longo do treinamento por modelo	48
Figura 25 – R^2 score ao longo do treinamento por modelo	49
Figura 26 – Erro médio em cada iteração por modelo	50
Figura 27 – R^2 score médio em cada iteração por modelo	51
Figura 28 – Erro médio de cada modelo por microconstituente	52
Figura 29 – Quantificação manual e quantificação prevista	53
Figura 30 – Quantificação manual e quantificação prevista	53

Figura 31 – Quantificação manual e quantificação prevista	53
Figura 32 – Quantificação manual e quantificação prevista	53
Figura 33 – Quantificação manual e quantificação prevista	54
Figura 34 – Quantificação manual e quantificação prevista	54
Figura 35 – Quantificação manual e quantificação prevista	55
Figura 36 – Quantificação manual e quantificação prevista	55

LISTA DE TABELAS

Tabela 1 – Quantificação microestrutural manual de algumas imagens do conjunto de dados	43
Tabela 2 – Comparação dos resultados entre os modelos	49

LISTA DE ABREVIATURAS E SIGLAS

ASTM	Sociedade Americana para Testese Materiais
CNN	<i>Convolutional Neural Network</i>
PF (G)	Ferrita Primária Granular
PF (I)	Ferrita Primária Intragranular
FS (A)	Ferrita de secundafase alinhada
FS (NA)	Ferrita de segunda fase não alinhada
AF	Ferrita Acicular
IA	Inteligência Artificial
ML	<i>Machine Learning</i>
DL	<i>Deep Learning</i>
OPF	<i>Optimum-Path Forest</i>
SVM	<i>Support Vector Machine</i>
GLCM	<i>Gray-Level Co-Occurrence Matrix</i>
LBP	<i>Local Binary Patterns</i>
RNA	Redes Neurais Artificiais
SGD	<i>Stochastic Gradient Descent</i>
GD	<i>Gradient Descent</i>
MSE	<i>Mean Squared Error</i>
RMSE	<i>Root Mean Squared Error</i>
MAE	<i>Mean Absolute Error</i>

SUMÁRIO

1	INTRODUÇÃO	19
1.1	A IMPORTÂNCIA DA ANÁLISE MICROESTRUTURAL	19
1.2	INTELIGÊNCIA ARTIFICIAL COMO UMA ALTERNATIVA VIÁVEL	21
1.3	JUSTIFICATIVA	24
1.4	OBJETIVOS	24
1.5	ESTRUTURA DO TRABALHO	25
2	REVISÃO DA LITERATURA	26
2.1	FUDAMENTAÇÃO TEÓRICA	26
2.1.1	Análise computacional de imagens	26
2.1.2	Redes Neurais Artificiais	29
2.1.3	Redes Neurais Convolucionais	31
2.2	TRABALHOS RELACIONADOS	34
2.2.1	Extração de Atributos + Algoritmos de <i>Machine Learning</i> . .	35
2.2.2	Redes Neurais Convolucionais para análise microestrutural . .	36
3	MATERIAIS E MÉTODOS	38
3.1	TRANSFERÊNCIA DE APRENDIZADO	38
3.2	ALEXNET	38
3.3	VGG16	39
3.4	VGG19	40
3.5	INCEPTIONV3	40
3.6	XCEPTION	41
3.7	CONJUNTO DE DADOS	42
3.8	PREPARAÇÃO DOA DADOS	42
3.9	DETALHES DA IMPLEMENTAÇÃO	44
3.9.1	Validação Cruzada	44
3.9.2	Treinamento	45
3.9.3	Métricas de Avaliação	45
3.9.4	Otimizadores	46
4	RESULTADOS E DISCUSSÃO	48
5	CONCLUSÕES	56
5.1	CONSIDERAÇÕES FINAIS	56
5.2	TRABALHOS FUTUROS	57

REFERÊNCIAS 58

1 INTRODUÇÃO

1.1 A IMPORTÂNCIA DA ANÁLISE MICROESTRUTURAL

Microestruturas são preenchimentos de espaços tridimensionais e distribuições de fases não regulares e não aleatórias [16]. Elas influenciam praticamente todos os aspectos do comportamento dos materiais e são os elementos mais suscetíveis ao controle na hierarquia estrutural dos materiais [16].

As propriedades mecânicas de um material são determinadas essencialmente por suas microestruturas [6]. Essas microestruturas armazenam a gênese de um material e determinam todas as suas propriedades físicas e químicas [5]. Se a microestrutura consistir em mais de uma fase, as propriedades do material dependem fortemente do tipo e distribuição das respectivas fases. Portanto, é crucial determinar o tipo e a quantidade das diferentes fases, a fim de se avaliar a relação estrutura-propriedade do material [18]. Assim, a identificação e quantificação de constituintes microestruturais de um material é um aspecto crítico em engenharia.

Uma área da indústria em que é essencial o procedimento de contagem de microconstituintes em micrografias, é a soldagem. A soldagem está presente quase todos lugares, com aplicações importantes em diversas áreas, tais como automotiva, aeroespacial, naval, informática e construção civil. Independente do tipo e da aplicação da soldagem, é sempre importante se analisar as propriedades mecânicas do cordão de solda obtido, e isso se dá pela análise e quantificação microestrutural do material [2].

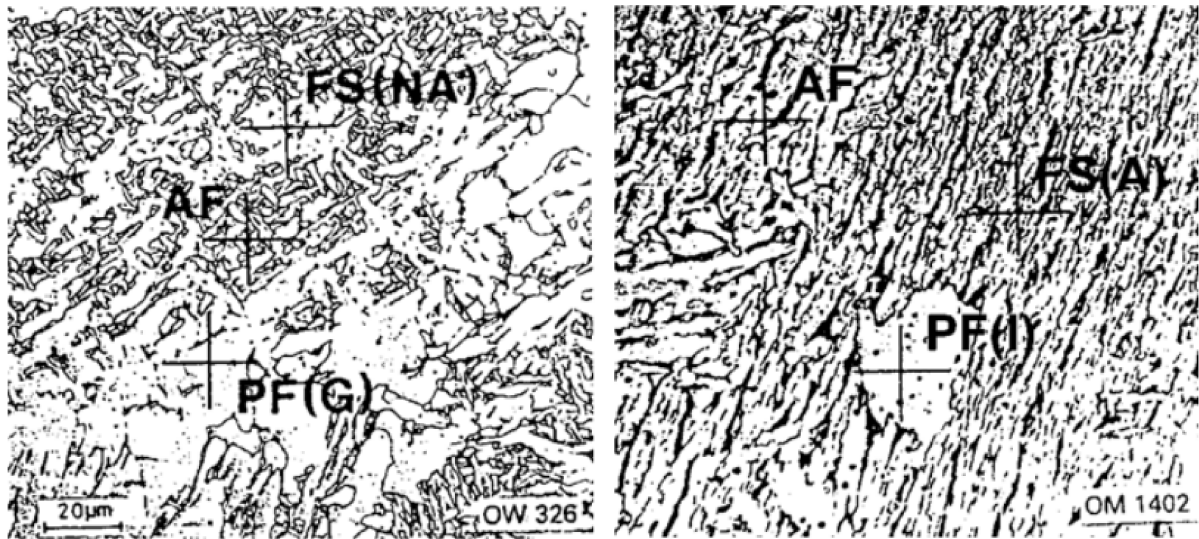
No entanto, até hoje, esta tarefa tem que ser feita manualmente por especialistas ou com o auxílio de dispositivos caros e ineficientes. A Sociedade Americana para Testes e Materiais (ASTM) produz industrialmente padrões reconhecidos para a quantificação microestrutural de forma manual. A norma ASTM E562 estabelece padrão para a determinação das frações volumétricas de microconstituintes via o Método da Contagem Manual de Pontos [4]. No entanto, a implementação desses padrões é muito trabalhosa e muito propensa a erros. Estima-se que um especialista experiente, implementando estes padrões, leve em média 15 minutos por imagem para realizar a identificação e quantificação adequada de microestruturas [9].

Neste trabalho, Redes Neurais Convolucionais, (do inglês, *Convolutional Neural Networks* (CNNs)) foram aplicadas para quantificar cinco microestruturas muito comuns em zonas fundidas por soldagem: Ferrita Primária Granular (PF (G)), Ferrita Primária Intragranular (PF (I)), Ferrita de segunda fase alinhada (FS (A)), Ferrita de segunda fase não alinhada (FS (NA)) e Ferrita Acicular (AF), seguindo a terminologia de [31]. Cada uma dessas microestruturas possui uma morfologia própria e indica uma característica no material, como especificado abaixo [29].

- Ferrita Primária de Contorno de Grão PF(G): Grãos de ferrita usualmente poligonais, localizados dentro dos grãos austeníticos prévios e mais três vezes maiores do que os grãos ou lâminas adjacentes de ferrita. Apresenta-se em forma de veios claros e lisos, delineando o contorno de grão colunar da austenita conferindo-lhe uma forma alongada. De um modo geral, altas quantidades de PF(G) são indesejáveis em soldas que devem apresentar elevada resistência à fratura frágil, uma vez que entre os seus grãos podem ser observados constituintes ricos em carbono e em impurezas [29] [10].
- Ferrita Poligonal Intragranular PF(I): Apresenta Veios de grãos poligonais associados com os contornos austeníticos prévios. É mais comum em soldas com baixa velocidade de resfriamento e com baixo teor de elementos de liga [29] [10].
- Ferrita com Segunda Fase Alinhada FS(A): Aparece na forma de grãos grosseiros e paralelos, crescendo sempre ao longo de um plano bem definido, formando duas ou mais placas de ferrita paralelas. Essa morfologia juntamente com a presença de filmes nç de constituintes ricos em carbono e frágeis em seus contornos, fazem com que a FS(A) seja pouco desejada na zona fundida de soldas que devam apresentar um certo grau de tenacidade [29] [10].
- Ferrita com Segunda Fase não Alinhada FS(NA): Ferrita envolvendo completamente ou microfases aproximadamente equiaxiais ou distribuídas aleatoriamente ou lâminas isoladas de AF.
- Ferrita Acicular (AF): A Ferrita Acicular pode ser considerada uma microestrutura caótica, ela é caracterizada por grãos pequenos em um formato agulhado, quando vistos em duas dimensões. Os grãos, na verdade tridimensionais, tem a forma de uma fina camada lenticular. Devido a essa ordenação caótica, ela possui propriedades mecânicas superiores, especialmente a resistência. Por isso, ela é amplamente reconhecida por ser uma microestrutura desejável [6]

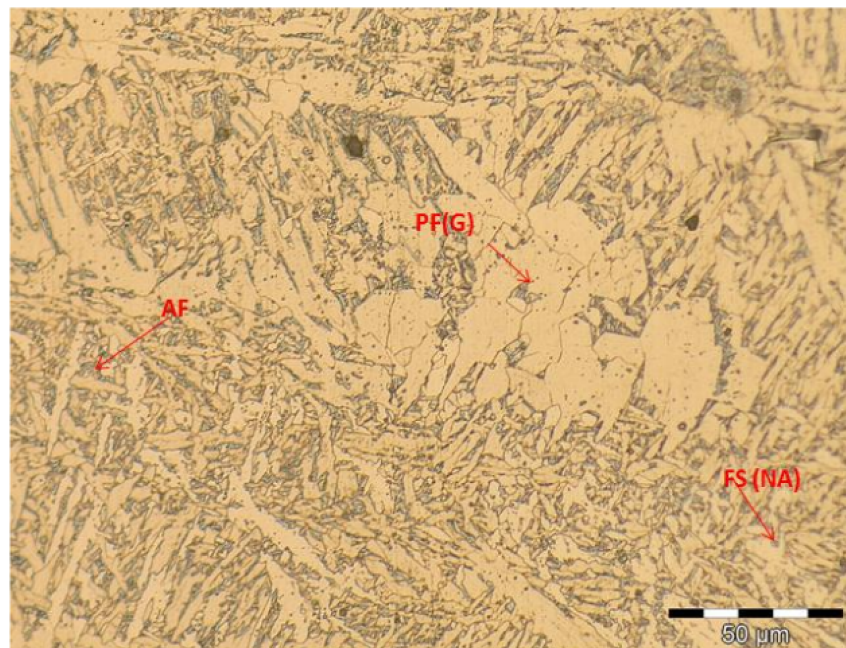
A Figura 1 a seguir, apresenta exemplos de micrografias de aço com as respectivas morfologias de microestruturas presentes no material. Já a Figura 2 apresenta uma das micrografias analisadas neste trabalho, suas respectivas morfologias microestruturais também são destacadas. Em ambas imagens, é possível notar que a identificação das diferentes regiões microestruturais não é fácil.

Figura 1 – Micrografia de cordão de solda mostrando diferentes microconstituintes, segundo [31]



Fonte: Extraído de [29].

Figura 2 – Micrografia de cordão de solda do conjunto de dados analisado neste trabalho



Fonte: Extraído de [10]

1.2 INTELIGÊNCIA ARTIFICIAL COMO UMA ALTERNATIVA VIÁVEL

Apesar do que muitos pensam, o tema Inteligência Artificial (IA) não é tão recente. Quando Alan Turing perguntou no seu artigo [45]: "As máquinas podem pensar?", muitos na época podem ter o achado futurista demais, mas nota-se que este tema já estava em

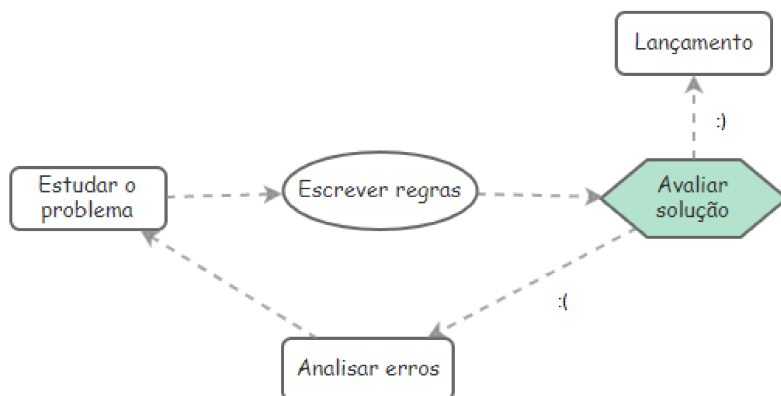
discussão. Pouco tempo depois o modelo Perceptron, que é a base de uma neural, foi proposto por [35]. Entretanto, nesta época não se tinha duas coisas essenciais para o desenvolvimento de sistemas inteligentes: dados e poder computacional. Logo, o tema inteligência artificial ficou no esquecimento por longas duas décadas, período conhecido como Primeiro Inverno da IA.

Somente na década de 1980 que o tema IA voltou a ficar quente, com o surgimento de diversas publicações importantes, como a de [36], que comprovaram a relevância e assertividade das redes neurais. Porém, a disponibilidade de dados ainda não era tão grande, tampouco o poder computacional. Sendo assim, a IA só foi voltar à tona de vez na década de 2000, na era da *Big Data*, em que a disponibilidade de dados nunca foi tão grande. Além disso, o poder computacional está crescendo exponencialmente ao longo dos anos. Isto fez com que a IA começasse a fazer parte do nosso dia a dia, com aplicações em diversos campos da vida humana.

As técnicas de inteligência artificial já se mostraram muito eficazes em diversas áreas, como medicina, advocacia e engenharia, superando muitas vezes resultados obtidos com operadores humanos [39] [40]. Além disso, na maioria das vezes estas ferramentas conseguem entregar os resultados de forma muito mais rápida.

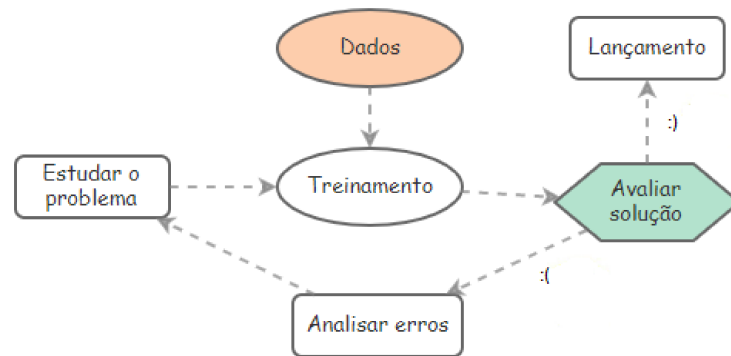
A base da inteligência artificial são as técnicas de Aprendizado de Máquina (do inglês, *Machine Learning* (ML)) e de Aprendizado Profundo (do inglês, *Deep Learning* (DL)). *Machine Learning* é a ciência (e arte) da programação de algoritmos para que eles possam aprender com dados, elas se diferem dos algoritmos tradicionais baseados em regras porque estes não possuem a capacidade de aprender e evoluir, como é possível observar nas Figuras 3 e 4. *Deep Learning* pode ser vista como uma subárea das áreas de *Machine Learning* (ML) e *Artificial Intelligence* (AI), Figura 5 abaixo. Em DL os algoritmos de aprendizado são redes neurais com muitas camadas (profundas).

Figura 3 – Fluxo de um algoritmo baseado em regras



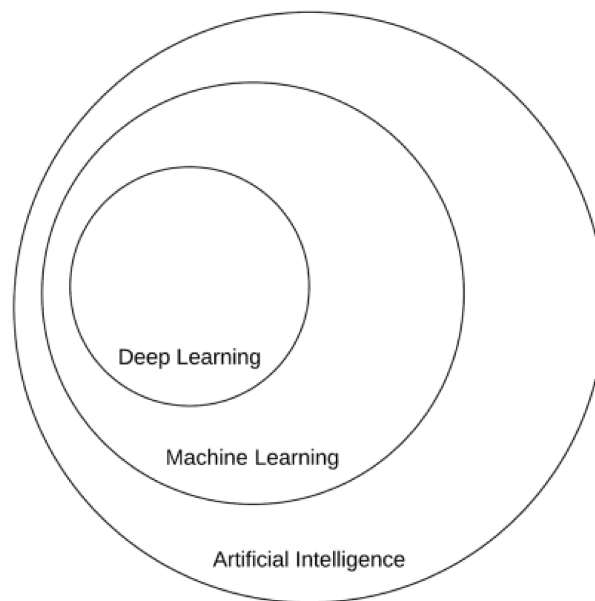
Fonte: Elaborada pelo autor (2020).

Figura 4 – Fluxo de um algoritmo de aprendizado



Fonte: Elaborada pelo autor (2020).

Figura 5 – Áreas de inteligência computacional



Fonte: Extraído de [34].

A indústria como um todo vem passando por uma enorme transformação. Cada dia que passa mais métodos computacionais são empregados na análise e manufatura de produtos e isso faz parte da chamada Indústria 4.0. Diversas atividades manuais e demoradas podem ser substituídas ou aprimoradas via técnicas de Inteligência Artificial. Entretanto, algumas dessas atividades ainda continuam a serem executadas de forma manual, como é o caso da quantificação microestrutural de materiais. Isto nos motiva a usar os métodos de DL para esta tarefa de quantificar automaticamente estes microconstituintes nos materiais.

Nos últimos anos, devido à sua notável precisão no reconhecimento e classificação

de padrões em imagens, as CNNs são consideradas o tipo de DL mais eficaz no tratamento de tarefas de visão computacional [5]. Elas se tornaram tão populares devido a sua capacidade de automaticamente criar e analisar atributos em imagens, através de operações de convolução, esses modelos retêm a informação da vizinhança, permitindo vincular atributos locais à um label associado, sem a necessidade de extrair atributos de forma manual [33]. No entanto, vale destacar que a aplicação das redes neurais convolucionais para quantificar os microconstituintes ainda foi pouco explorada. Neste projeto cinco diferentes arquiteturas de CNNs foram comparadas: AlexNet e VGG16 treinadas do zero, e VGG19, Xception e InceptionV3 todas pré-treinadas.

1.3 JUSTIFICATIVA

A análise e quantificação de microestruturas de materiais é um procedimento muito importante em engenharia, que ainda não foi automatizado [9]. Ao se estudar a viabilidade da aplicação de CNNs para este problema, a abertura de um caminho para otimizar o trabalho nesta área da indústria pode se tornar possível.

As CNNs são muito aplicadas em problemas de classificação e segmentação de imagens. Porém, a aplicação dessas redes para problemas envolvendo regressão ainda não foi tanto explorada. Especificamente, a aplicação destas redes para quantificar microconstituintes em micrografias é um caminho que ainda não foi muito desbravado na ciência e engenharia. Desta forma, com este estudo é possível abrir campo para mais pesquisas nesta área, além de apresentar mais uma aplicação desta poderosa ferramenta.

Ao estudar a aplicação de cinco das principais arquiteturas de CNNs, é possível se analisar o modelo que melhor se adequa para esta aplicação. Além disso, a comparação de modelos treinados do zero e pré-treinados é importante para a análise destas duas possibilidades. Isto pode servir de base para futuros trabalhos e pesquisas, ou até mesmo difusão desta solução para o mercado e a indústria.

1.4 OBJETIVOS

O presente trabalho tem como **objetivo geral** analisar a eficácia da aplicação de redes neurais convolucionais na tarefa de quantificação de microconstituintes em micrografias de aço.

Os **objetivos específicos** que se destacam no trabalho são:

- Verificar qual modelo de CNN performa melhor nesta aplicação;
- Avaliar a performance de CNNs em uma tarefa de regressão;

- Avaliar e comparar o desempenho de modelos pré-treinados com modelos treinados do zero.

1.5 ESTRUTURA DO TRABALHO

Este trabalho está dividido em seis capítulos: Introdução, Revisão da Literatura, Materiais e Métodos, Resultados e Discussão, e Conclusões e Trabalhos Futuros.

No **capítulo 2** de Revisão da Literatura, é apresentado o estado da arte envolvendo aplicações aprendizado de máquina e aprendizado profundo para análise de imagens, através da exposição dos principais trabalhos relacionados ao tema.

No **capítulo 3** de Materiais e Métodos é apresentado como é feito a análise computacional de imagens e como as redes neurais atuam no tratamento de dados. As CNNs, que é o principal método utilizado neste trabalho, são descritas de forma teórica minuciosa e todas as arquiteturas utilizadas são também apresentadas. Além disso, é descrito também as métricas utilizadas, os otimizadores, como o conjunto de dados foi apresentado e os demais detalhes da implementação.

No **capítulo 4** de Resultados e Discussões são apresentados os principais resultados de performance das redes, gráficos de comparação, os resultados obtidos para cada microestrutura separadamente, bem como algumas previsões para os conjuntos de teste.

No **capítulo 5** de Conclusões e Trabalhos Futuros é feita uma revisão do que foi abordado no trabalho, ressaltando as principais contribuições do projeto, além da análise dos resultados, verificação dos objetivos atingidos e sugestões para trabalhos futuros.

2 REVISÃO DA LITERATURA

Este capítulo possui duas seções principais: Fundamentação Teórica e Trabalhos Relacionados. Na primeira seção, é feita uma descrição teórica de três tópicos importantes para o trabalho: Análise computacional de imagens, Redes Neurais Artificiais e Redes Neurais Convolucionais. Na segunda seção de Trabalhos Relacionados, temos uma subseção de trabalhos de análise microestrutural de imagens envolvendo técnicas de extração de atributos combinadas com algoritmos de *machine learning* e uma subseção de trabalhos deste tema que utilizaram CNNs.

2.1 FUDAMENTAÇÃO TEÓRICA

Como o conjunto de dados utilizado neste trabalho é formado por imagens, comecemos esta seção apresentando como o computador analisa esse tipo de dado e quais são as principais técnicas de aprendizado empregadas para este fim. Como as CNNs são um tipo de rede neural artificial, a segunda subseção apresenta uma breve introdução sobre a lógica deste algoritmo de aprendizado. Encerrando esta seção, temos um tópico abrangente envolvendo CNNs. É apresentado a teoria por trás destas redes, bem como suas principais camadas.

2.1.1 Análise computacional de imagens

Enquanto nós vemos uma imagem como ela é, o computador vê a imagem como uma matriz de números. Isto ocorre porque toda imagem consiste de um conjunto de pixels. Normalmente, um pixel é considerado uma cor ou intensidade de luz que aparece em um dado lugar da imagem. Se pensarmos a imagem como uma malha, cada quadrado contém um único pixel. Em uma escala de cinza, cada pixel é um escalar entre 0 e 255, onde zero corresponde a "preto" e 255 a "branco"[34]. A Figura 6 abaixo demonstra como os pixels são agrupados em uma escala de cinza.

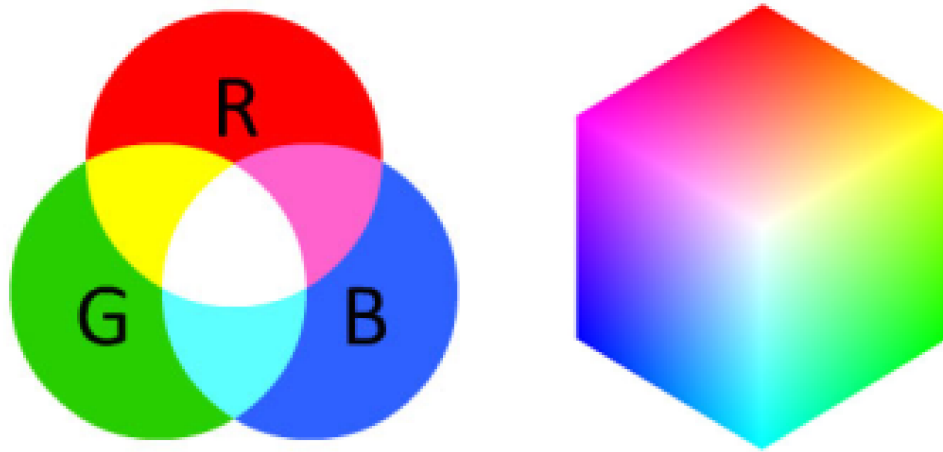
Figura 6 – Gradiente de imagem demonstrando pixels variando de 0 (preto) até 255 (branco)



Fonte: Extraído de [34].

Entretanto, pixels coloridos são representados normalmente em uma escala RGB. Os pixels na escala RGB são nada mais do que um valor escalar como na escala de cinza, a única diferença é que agora os pixels são representados por uma lista de três valores: um valor para o componente vermelho (R), um para o verde (G) e outro para o azul (B). Portanto, para definir uma cor na escala RGB, tudo que precisamos é definir a quantidade de Vermelho, Verde e Azul contida em um único pixel. Cada canal possui valores também em uma faixa de 0 a 255, onde 0 indica que não há representação e 255 demonstra representação total da cor [34]. A Figura 7 ilustra como é representada a escala RGB.

Figura 7 – Escala colorida de pixels RGB

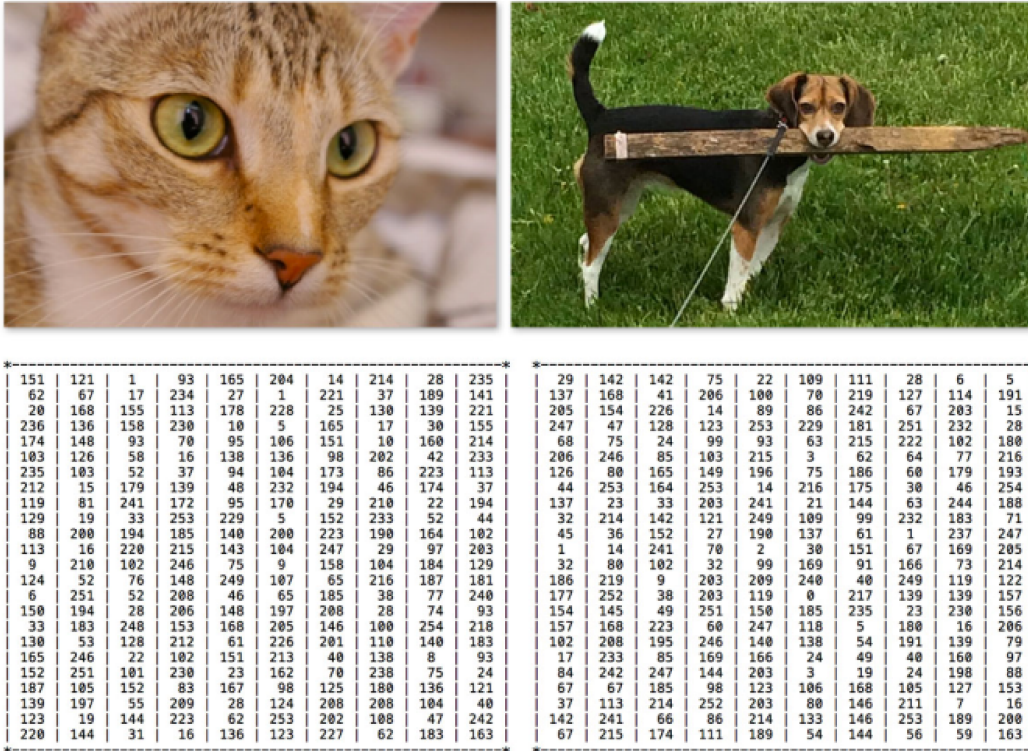


Fonte: Extraído de [34].

Para ilustrar a diferença como nós vemos uma imagem e como computador a vê, a Figura 8 abaixo apresenta duas imagens: uma de um gato e a outra de um cachorro, ambas com a representação matemática computacional abaixo. Nosso cérebro consegue ver claramente a diferença entre a imagem que contém o gato e a imagem que contém o cachorro. No entanto, tudo o que o computador identifica é uma grande matriz de números. A diferença entre como percebemos uma imagem e como a imagem é representada (uma matriz de números) é chamado de *gap semântico* [34].

A principal vantagem de se utilizar uma escala em cinza é que ela é menos custosa computacionalmente. Entretanto, como neste trabalho foi aplicado uma redução na resolução das imagens, a diferença do tempo de execução entre as imagens lidas na escala cinza e na escala colorida não foi tão grande. Logo, foi decidido se utilizar a escala RGB para a leitura das imagens neste trabalho. No entanto, devido ao padrão das imagens, a utilização da escala de cinza também seria tranquilamente viável.

Figura 8 – Diferença entre como nós percebemos uma imagem e como o computador a percebe



Fonte: Extraído de [34].

Para se trabalhar técnicas de aprendizado de máquina em um conjunto de dados formado por imagens, geralmente se utiliza duas abordagens, ou é utilizado uma técnica de extração de atributos combinada com uma algoritmo de *machine learning* ou se utiliza redes neurais convolucionais. Nas duas abordagens, a extração dos atributos é fundamental para diminuição da complexidade computacional e redução de *overfitting*.

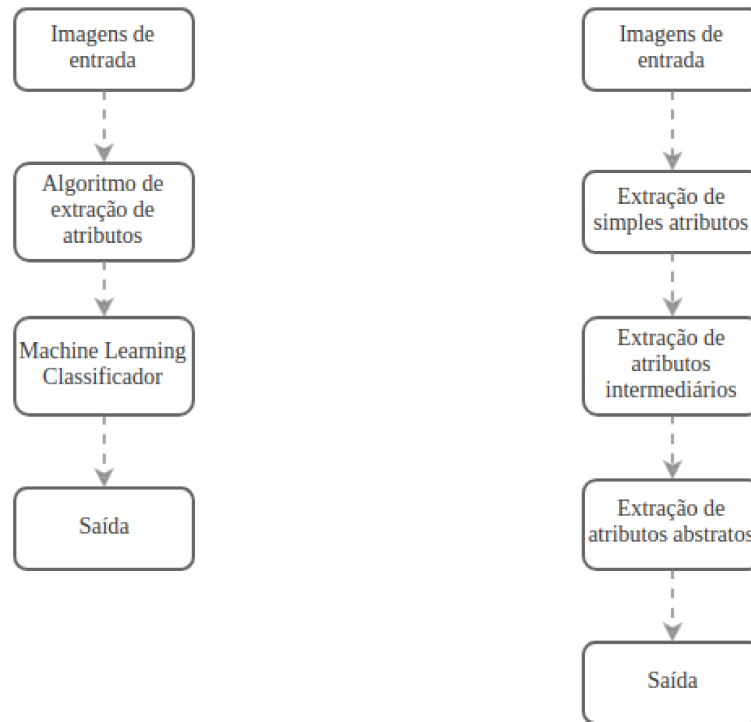
A Figura 9 apresenta o fluxo do processo nos dois diferentes casos. É possível perceber dois pontos com a análise das duas diferentes abordagens: o primeiro é que ao se utilizar as redes neurais convolucionais, uma gama maior de atributos é extraída; o segundo ponto é, que em CNNs, a classificação dos atributos é feita sem separação da fase de classificação [5].

A escolha do método depende muito do tipo de imagem contido no conjunto de dados, da quantidade de dados e o custo computacional que se quer investir. Para este trabalho foi escolhido a aplicação de deep learning via CNNs por conta da complexidade das imagens e pela viabilidade computacional.

a)

b)

Figura 9 – Comparação entre métodos tradicionais de machine learning (a) e deep learning com redes neurais convolucionais (b) para análise e classificação de imagens

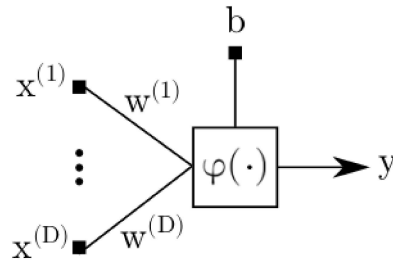


Fonte: Adaptado de [34].

2.1.2 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNAs) são modelos computacionais inspirados nas conexões nervosas do sistema nervoso animal [20]. De forma similar aos neurônios biológicos, esses modelos são formados por unidades interconectadas capazes de receber, processar e transmitir informações. Esses modelos são compostos basicamente por três elementos: dados de entrada, pesos de ponderação e funções de ativação. A Figura 10 abaixo apresenta a unidade básica de uma RNA, o neurônio artificial ou nó.

Figura 10 – Unidade básica de uma RNA

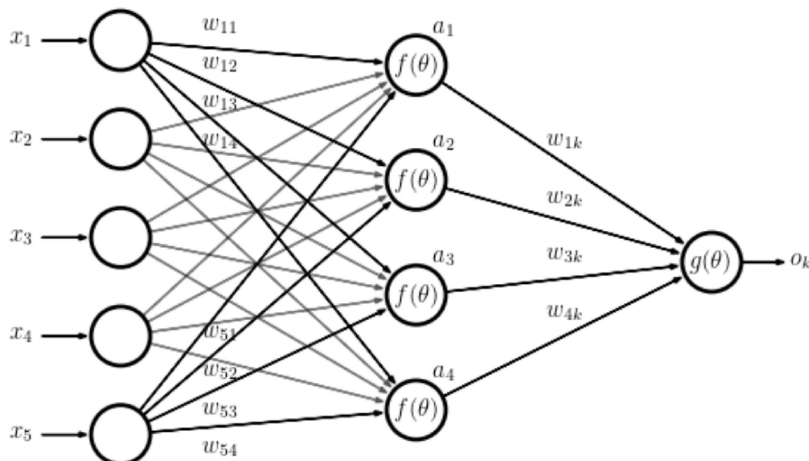


Fonte: Extraído de [30].

Onde $x^{(1)}, \dots, x^{(D)}$ correspondem aos valores de entrada provenientes de nós anteriores. Os pesos $w^{(1)}, \dots, w^{(D)}$ literalmente pesam cada valor de entrada, indicando a relevância de cada um. A função ϕ corresponde à função de ativação do nó, que tem como função ativar ou não um sinal de acordo com uma regra pré-estabelecida. O bias b corresponde a um valor limiar interno incluído como forma de generalização do modelo. Por fim, a saída y corresponde ao valor de saída do nó, que é transmitido aos nós subsequentes. Esta lógica que combina entradas, pesos e função de ativação foi proposta por [35] e é denominada *Perceptron*.

Quando expandimos o modelo, juntando vários dados de entrada, pesos e neurônios de ativação, obtemos uma configuração conhecida como *Multilayer Perceptron*. Diversas configurações podem ser obtidas por meio da organização espacial desses nós. Uma configuração típica é denominada *multilayer feedforward* totalmente conectada, Figura 11. Nesta rede a informação só move uma direção, para frente, dos nós de entrada, através dos nós intermediários, até os nós de saída [20].

Figura 11 – Uma típica arquitetura de uma Rede Neural Artificial do tipo *multilayer feedforward* totalmente conectada



Fonte: Adaptado de [34].

O valor obtido pelo nó intermediário a é dado por:

$$a_j = f \left(\sum_{i=1}^N w_{ij} x_i + b_j \right) \quad (2.1)$$

O valor obtido pelo nó o de saída da rede é dado por:

$$o_k = g \left(\sum_{j=1}^M w_{jk} a_j + b_k \right) \quad (2.2)$$

onde x_i são os valores de entrada da rede, w representa os pesos, b se refere aos valores de bias e f e g são as funções de ativação.

Como veremos na próxima subseção, uma das camadas de uma CNN é este tipo de configuração totalmente conectada representada na Figura 11. Desta forma, além de as CNNs utilizarem a lógica de aprendizado das RNAs, elas possuem até mesmo camadas de uma rede neural típica.

O **Treinamento de uma RNA** segue as seguintes etapas: A função de perda (\mathcal{L}) mede o quão distante as saídas da rede estão das saídas corretas. O problema de otimização é resolvido via técnica de descida do gradiente, que será abordado com mais rigor no capítulo 3. Então, é utilizada a lógica de *Back-Propagation* para propagar o gradiente da função de perda para as camadas anteriores [26]. Então, os pesos (w) se adaptam para diminuir a função de perda. O tamanho de passo desta atualização dos pesos é dado por uma taxa de aprendizado, representada por η na Equação 2.3, este parâmetro não tem unidade e é definido pelo usuário [26].

$$w_{ij}^k = w_{ij}^{k-1} - \eta \frac{\delta \mathcal{L}}{\delta w_{ij}} \quad (2.3)$$

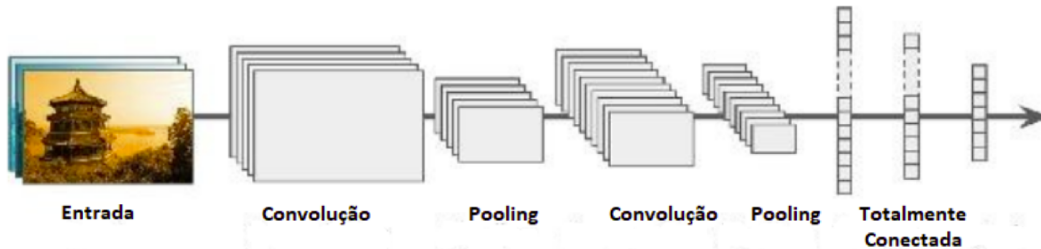
2.1.3 Redes Neurais Convolucionais

A CNN é um tipo de rede neural que tem como foco o reconhecimento de padrões visuais das imagens. A arquitetura desse tipo de rede neural é similar ao do padrão de conectividade dos neurônios do cérebro humano e sua organização teve como base o córtex visual [25]. Apesar das CNNs terem aparecido nos holofotes da comunidade científica somente na última década, seu conceito básico foi desenvolvido há quase 40 anos por [17]. No entanto, somente depois do aumento dos recursos computacionais, da disponibilidade de dados e da evolução dos algoritmos de aprendizado, é que os trabalhos utilizando este tipo de rede neural voltaram a ter uma atenção especial da ciência. Atualmente, as CNNs representam o estado da arte na área de visão computacional, estando presente em diversas aplicações complexas de reconhecimento, classificação e segmentação de imagens [33], [27], [39].

Como é possível observar na Figura 12, uma CNN padrão é formada por vários tipos repetidos de camadas empilhadas entre si [5]. Cada camada dessa possui um nome de

acordo com as operações que realiza. A seguir temos uma breve introdução das principais camadas e funções que compõem uma típica CNN.

Figura 12 – Uma típica arquitetura de uma Rede Neural Convolucional



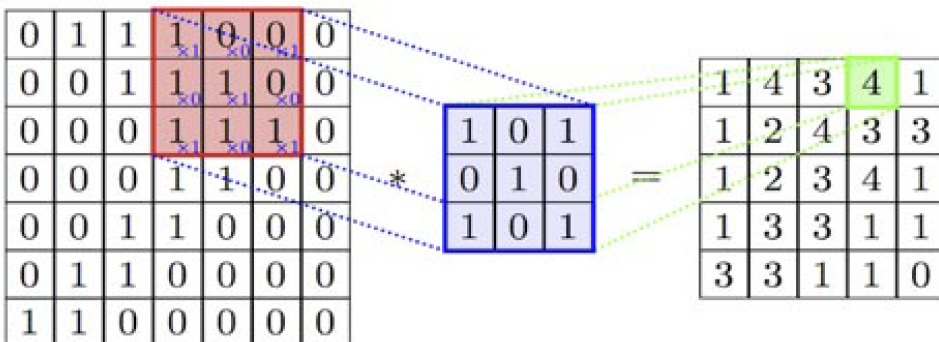
Fonte: Extraído de [19].

Camada de convolução é a primeira camada a extrair atributos de uma imagem de entrada. É composta por filtros de aprendizado, que são convoluídos com os dados de entrada, conforme mostrado na Equação 2.4 e na Figura 13

$$(h_k)_{ij} = (W_k * x)_{ij} + b_k \tag{2.4}$$

onde $k = 1, \dots, K$ é o índice do k -ésimo mapa de características na camada de convolução e (i, j) é o índice do neurônio s do k -ésimo mapa de características e x representa os dados de entrada. W_k and b_k são parâmetros treináveis (pesos) dos filtros lineares (kernel) e bias para neurônios no k -ésimo mapa de características, respectivamente. $(h_k)_{ij}$ é o valor da saída do neurônio no k -ésimo mapa de características com posição (i,j) [14], [5].

Figura 13 – Operação de convolução por filtros

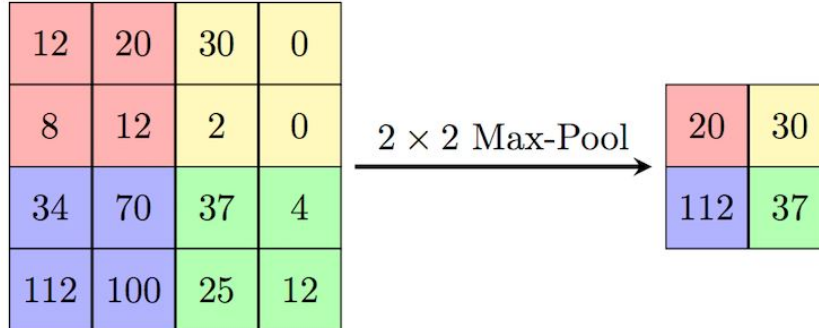


Fonte: Extraído de [38].

Camada Pooling reduz progressivamente o tamanho espacial da representação para diminuir o número de parâmetros e cálculos na rede, conforme é apresentado na Figura 14. As duas formas mais comuns são a *max-pooling* e a *global average pooling*, que utilizam valores máximos ou médios em cada sub-região dos dados de entrada [14], [28].

Neste trabalho se utilizou o tipo *max-pooling* para as arquiteturas AlexNet e VGG16 e *global average pooling* para as demais arquiteturas.

Figura 14 – Operação de compressão de imagens via Max-Pooling



Fonte: Extraído de [13]

Camada totalmente conectada é uma camada de rede neural clássica, conforme apresentado na Figura 11, em que cada nó está conectado a todos nós da camada anterior. Sua saída é uma combinação linear dos atributos da camada anterior, como mostrado na Equação:

$$y_k = \sum_l W_{kl}x_l + b_k \quad (2.5)$$

onde y_k representa o k -ésimo neurônio de saída e W_{kl} é o k -ésimo peso entre x_l e y_k [14], [5].

Função de Ativação aplica uma ativação não linear sobre um sinal de entrada. Geralmente segue uma camada de pooling ou totalmente conectada. Existem diferentes tipos de funções de ativação, tais como a tangente hiperbólica e a função sigmóide. No entanto, a função ReLU $relu(x) = \max(0, x)$ foi usada porque treina a rede neural mais rápido, sem uma penalidade significativa na precisão da generalização [14], [5].

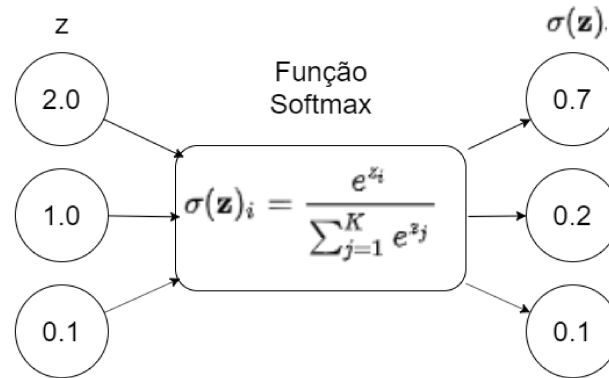
Camada de Classificação Softmax é a última camada da rede e calcula a probabilidade de uma classe do label dos dados de entrada. A camada de classificação mais utilizada em CNNs, e que também foi tida como a mais apropriada para este trabalho, é a função *Softmax*. Esta função aplica uma distribuição categórica de probabilidade baseada na função exponencial, como mostra a Equação 2.6, para uma k -ésima classe e uma entrada X [5].

$$P(y = j|X; W, b) = \frac{\exp^{X^T W_j}}{\sum_k^K \exp^{X^T W_j^T}} \quad (2.6)$$

Para ilustrar melhor essa função, a Figura 15 apresenta de forma esquemática como é o processo nesta camada. Os valores de entrada z são aplicados na equação 2.6 e

então se obtém as saídas em forma probabilística. Esta foi uma etapa fundamental neste trabalho. Especificamos a saída da função em 5 valores e foram estes valores que foram trabalhados na rede, até que se obtivesse valores próximos dos desejados para a nossa tarefa de quantificação.

Figura 15 – Função Softmax



Fonte: Elaborado pelo autor (2020).

Função de perda determina o desvio entre valores verdadeiros e os valores previstos, e é normalmente a camada final da rede neural. Como este trabalho pode ser considerado um problema de regressão. Foi aplicado neste trabalho o Erro Quadrático Médio (MSE) dado pela Equação 2.7, que é uma métrica de regressão muito utilizadas na camada loss [19].

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (2.7)$$

onde n é número de dados, Y é o valor verdadeiro e \hat{Y} é o valor previsto.

Camada Dropout é uma forma de regularização que tem como objetivo evitar overfitting ao desconectar randomicamente entradas da camada anterior para a próxima camada na arquitetura da rede [19]. Esta camada foi utilizada em todas arquiteturas implementadas neste trabalho.

O **Treinamento de CNNs** é feito, como visto na Figura 9, sem separação entre a etapa de extração dos atributos e a etapa de classificação. As demais etapas seguem o processo descrito na seção 2.1.2.

2.2 TRABALHOS RELACIONADOS

Na literatura é possível encontrar alguns trabalhos que visam a classificação automática e análise de microestruturas em micrografias. Os trabalhos encontrados que visam a identificação e quantificação de microconstituintes em imagens, na maioria das vezes utilizam técnicas de processamento digital de imagens [9], [47], [8]. Entretanto, são

raros os trabalhos que exploram a aplicação de *machine learning* e *deep learning* para a quantificação de microestruturas em micrografias. Os trabalhos relacionados encontrados utilizam ou redes neurais convolucionais ou a combinação de algoritmos de extração de atributos com algoritmos de machine learning.

2.2.1 Extração de Atributos + Algoritmos de *Machine Learning*

Algoritmos de extração de atributos combinados com modelos tradicionais de machine learning foram usados, por exemplo, por [32] para caracterizar e classificar três tipos de ferro fundido: nodular, cinzento e maleável. Três classificadores de machine learning foram utilizados para a execução desta tarefa: *Support Vector Machine* (SVM), *Optimum-Path Forest* (OPF) e o classificador Bayesiano, além do método de segmentação de imagens Otsu. Para extração dos atributos das imagens foi utilizado o método Filtro de Gabor. Pelos resultados, o classificador OPF apresentou uma performance geral superior, tanto em termos de acurácia quanto em tempo de execução.

É importante lembrar que, o conjunto de dados utilizado neste presente trabalho é composto por micrografias de cordões de solda. Técnicas de machine learning para a análise da qualidade de cordões de solda também foram empregadas por [46]. Entretanto, o autor utilizou macrografias de cordões de solda e o trabalho tinha objetivo de classificar se o cordão era bom ou ruim. Foram empregados os métodos de extração de atributos *Gray-Level Co-Occurrence Matrix* (GLCM), *Local Binary Patterns* (LBP) e os algoritmos de classificação SVM e KNN. Os melhores resultados foram obtidos com a combinação LBP e SVM.

A aplicação do método SVM é muito comum em tarefas de classificação. Ele também foi empregado por [18], em que diferentes microestruturas (martensita, perlita e bainita) de dois aços distintos foram classificadas por métodos de mineração de dados. Os resultados foram obtidos a partir de diferentes métodos de pré-processamento (bruto e transformado em log), diferentes métodos de divisão de dados (aleatório e amostral) e diferentes parâmetros C e gama do método SVM. Os melhores resultados atingiram 87% de acurácia.

Vale ressaltar que, as CNNs também podem ser utilizadas somente como algoritmos de extração de atributos, como no trabalho de [15]. Neste trabalho técnicas de aprendizado de máquina supervisionadas e não supervisionadas foram usadas para a identificação de microestruturas em aços de alto carbono tratados termicamente. Além da CNN, o método *Bag of Words* também foi utilizado para extrair os atributos das imagens, e o algoritmo de aprendizado SVM aplicado como classificador. Os resultados mostraram que as CNNs foram as melhores extratoras de características.

Um amplo trabalho envolvendo a comparação de métodos de extração de atributos e modelos de machine learning, foi desenvolvido por [12], em que microestruturas específicas

com diferentes magnitudes, tratamentos químicos e orientações foram automaticamente identificadas. Os principais métodos utilizados para a extração dos atributos foram: GLCM, LBP, *Visual Bag of Words* (VBOW) e algumas CNNs pré-treinadas. Duas tarefas de classificação foram desenvolvidas: classificação entre micrografias com e sem dendritos e classificação entre vistas de dendritos longitudinais e seções transversais. Os algoritmos de aprendizado SVM, KNN e Random Forests foram utilizados na tarefa de classificação. Os melhores resultados foram obtidos quando as redes neurais pré-treinadas foram usadas para extrair os atributos das imagens.

Um trabalho que também abordou técnicas de *machine learning* para tratar micrografias de microestruturas metalúrgicas, foi desenvolvido por [7]. Os autores demonstram uma abordagem de reconhecimento de padrões baseada no algoritmo de aprendizado *Random Forest* para automaticamente segmentar as microestruturas nas imagens. Segundo os autores, esta abordagem pode ajudar no tratamento de grandes volumes de imagem em pouco tempo e também permite a busca de novos aços com propriedades desejáveis.

2.2.2 Redes Neurais Convolucionais para análise microestrutural

Em [23], CNNs foram empregadas para correlacionar microestruturas experimentais com as suas respectivas condutividades iônicas. Os autores desenvolveram uma CNN própria parecida com a rede VGG, eles também utilizaram técnicas para aumentarem o número de imagens do conjunto de dados. Os resultados se mostraram superiores em performance comparados com métodos de extração de atributos tradicionais e também os autores concluíram que, ao contrário do que muitos pensam, não é necessário grande quantidade de imagem, dependendo de como estas estão preparadas.

Outra publicação que mostra o quando as CNNs podem superar os métodos tradicionais de extração de atributos foi desenvolvida por [37]. Neste trabalho, a performance de classificação das CNNs foram comparadas com outros métodos de extração manual em três diferentes conjunto de imagens: Defeitos em aço, defeitos em madeira e defeitos aço. Em todos os cenários os resultados obtidos pelas CNNs foram superiores.

Métodos de aprendizagem profunda foram usados também por [5] para classificar automaticamente microestruturas, tais como martensita, bainita e perlita. Técnicas de segmentação de imagens e de redução de dimensionalidade foram empregadas. Além disso, Diferentes arquiteturas de CNNs e também Máquinas de Vetor Suporte (SVM) foram empregados, mas os melhores resultados de acurácia atingindo a ordem de 93%, foram atingidos pelo modelo de CNN proposto pelos autores, chamado MVFCNN (*Max-voting Fully Convolutional Neural Network*).

CNNs também foram empregadas por [28] para classificar microestruturas. Micrografias de aço de alto teor de carbono foram divididas em sete classes diferentes de microestruturas, e quatro arquiteturas de CNNs foram utilizadas na tarefa de classificação:

LeNet, GoogleNet, VGG19 e ResNet. Todas as redes performaram satisfatoriamente e os melhores resultados em termos de acurácia foram obtidos pelo modelo LeNet.

Quando se trata de CNNs, um tema bastante relevante e que será abordado com mais detalhes no capítulo 3, é a Transferência de Aprendizado (do inglês, *Transfer Learning*). Na literatura é possível encontrar trabalhos significativos que utilizaram esta abordagem, como é o caso de [27]. Nesta obra, os autores utilizaram uma abordagem de *transfer learning* para o problema de reconstrução de microestruturas e a previsão de propriedades estruturais. As abordagens não se limitam à materiais específicos e se demonstraram vantajosas para o tratamento de microestruturas complexas.

Em [33] uma abordagem orientada a dados é implementada para mapear a microestrutura ao desempenho fotovoltaico usando redes neurais convolucionais. Os autores desenvolveram uma arquitetura CNN para classificar as morfologias em classes de performances. Eles compararam a performance do modelo proposto com os modelos VGG16 e ResNet 50. Os resultados mostraram que o modelo proposto é mais adequado, uma vez que ele é capaz de indentificar atributos críticos nas morfologias, o que ambos VGG16 e ResNet50 falharam em identificar consistentemente.

Em suma, visto a quantidade de trabalhos relacionadas, pode-se dizer que a abordagem de técnicas de inteligência artificial para a análise microestrutural de imagens é um assunto que está aquecido na ciência. No entanto, como já mencionado anteriormente, a aplicação de CNNs para a quantificação de microestruturas em micrografias ainda não foi muito trabalhada por outros autores, e isto serve de motivação e justificativa para este projeto.

3 MATERIAIS E MÉTODOS

Neste capítulo, descrevemos a abordagem proposta para quantificar microestruturas em imagens de aço. Começamos o capítulo introduzindo o método Transferência de Aprendizado, que foi aplicado em três das arquiteturas aplicadas neste projeto. Posteriormente, cada arquitetura utilizada é introduzida separadamente. Depois, temos duas seções relacionadas ao conjunto e preparação dos dados que foram utilizados. Por fim, a última seção traz os principais detalhes da implementação, apresentando as técnicas de treinamento, métricas e otimizadores utilizados.

3.1 TRANSFERÊNCIA DE APRENDIZADO

A técnica Transferência de Aprendizado (do inglês, *Transfer Learning*) é uma ferramenta de aprendizado de máquina baseada na transferência de parâmetros de uma rede neural treinada com um conjunto de dados para outro problema com outro conjunto de dados. Desta forma, a rede tem somente suas últimas camadas retreinadas para se adaptar ao problema proposto. Isto não só aumenta a velocidade de treino consideravelmente, como também requer muito menos dados de treinamento [19].

Neste projeto, esta técnica foi utilizada para os modelos VGG19, Xception e InceptionV3. Estes modelos já foram treinados no conjunto de dados *ImageNet* [1], que é um conjunto de mais de 1 milhão de imagens de mais mil categorias diferentes. Desta forma, estes modelos pré-treinados já são capazes de reconhecer diversos tipos de padrões diferentes em imagens, e, como não há a necessidade do treinamento de todas as camadas, eles podem chegar à solução final de forma mais rápida.

Nas subseções seguintes, as cinco arquiteturas CNNs aplicadas neste trabalho são descritas. Vale ressaltar que em todas as configurações, se utilizou a função de ativação ReLU nas camadas de convolução e a função *Softmax* na camada de classificação.

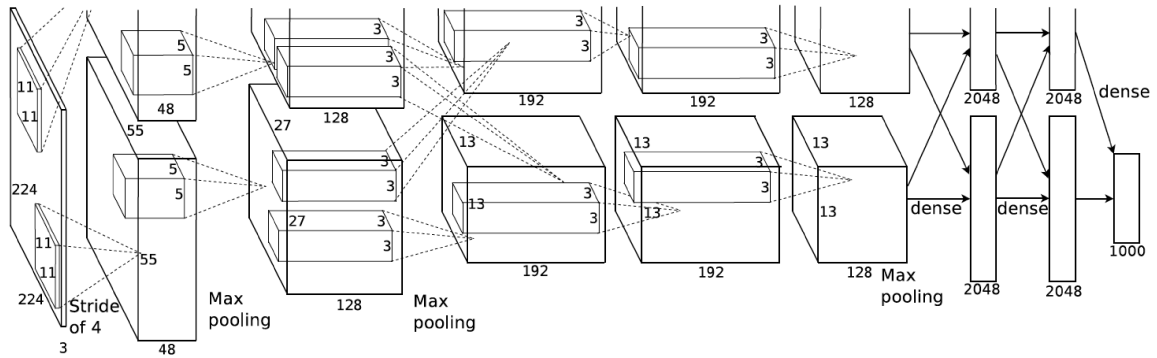
3.2 ALEXNET

AlexNet, proposta por [24], atingiu um erro de 15.3% no desafio de reconhecimento visual em larga escala ImageNet (ILSVRC), mais de 10.8 pontos percentuais abaixo do segundo colocado, sendo até hoje considerada um divisor de águas na área de visão computacional. O modelo é uma CNN profunda com mais de 60 milhões de parâmetros e 650.000 neurônios. A arquitetura contém oito camadas, como mostra a Figura 16 a seguir.

Pela figura, observa-se que as primeiras cinco camadas são do tipo convolucionais, sendo que algumas delas são seguidas por camadas max-pooling, e as três últimas são camadas totalmente conectadas. A não linearidade ReLU é aplicada após toda a convolução e as camadas totalmente conectadas. Além disso, a camada Dropout é aplicada antes da

primeira e da segunda camada totalmente conectada. É importante enfatizar, como já mencionado, que a AlexNet foi treinada do zero neste trabalho. Todas as configurações apresentadas na figura foram empregadas na implementação.

Figura 16 – O modelo AlexNet

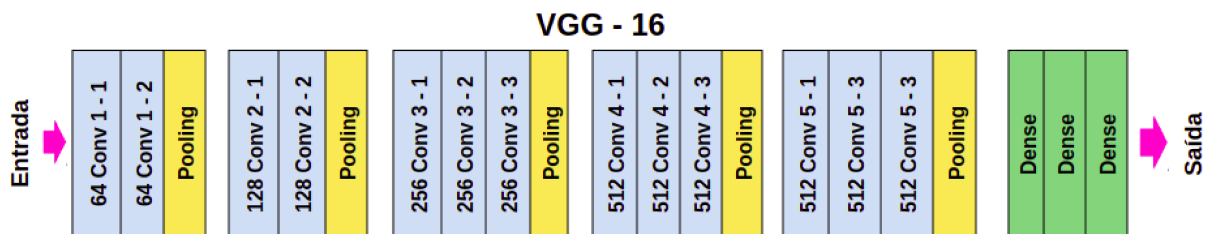


Fonte: Extraído de [24]

3.3 VGG16

VGG16, proposta por [41] em 2014, atingiu 92.7% no top-5 teste de acurácia no ImageNet. Este modelo faz uma melhora em relação ao modelo AlexNet ao substituir filtros grandes demais (11x11 e 5x5 na primeira e na segunda camada, respectivamente) por múltiplos filtros 3x3. ReLu e max-pooling são também aplicadas e as camadas totalmente conectadas, cada uma com 4.096 nós, são seguidas por um classificador *softmax*. A arquitetura possui 16 camadas, Figura 3.3, sendo 13 convolucionais e 3 totalmente conectadas.

Figura 17 – O modelo VGG16



Fonte: Elaborada pelo autor (2020).

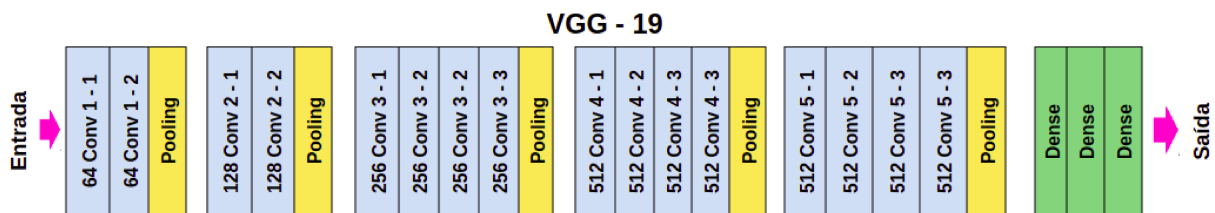
Uma grande desvantagem da VGG16 é que ela é muito lenta para se treinar [34]. Isso é devido a sua profundidade e o número de camadas totalmente conectadas, o que a torna uma rede muito pesada. Posteriormente na seção de resultados, será possível

notar este fato, pois esta rede foi a que apresentou o maior tempo de execução total. Vale ressaltar que neste trabalho a VGG16 também foi treinada do zero.

3.4 VGG19

VGG19 é outra versão da VGGnet, proposta por [41]. Foi a segunda colocada na competição ILSVRC 2014, alcançando um erro de 7.3%. É composta por 16 camadas convolucionais e três totalmente conectadas, totalizando 19 camadas no total. Da mesma forma da VGG16, a largura das camadas de convolução começa em 64 na primeira camada e depois aumenta em um fator de 2 após cada camada de max-pooling até atingir 512. As camadas convolucionais usam 3 x 3 filtros com um passo de 1 e preenchimento de 1 e uma ativação ReLU é realizada logo após cada convolução. As duas primeiras camadas totalmente conectadas possuem 4096 canais cada, e a terceira contém o número de classes previsíveis, conforme mostrado na Figura 18 abaixo.

Figura 18 – O modelo VGG19



Fonte: Elaborada pelo autor (2020).

A VGG19 foi treinada em mais de um milhão de imagens do conjunto de dados ImageNet. Conseqüentemente, a rede aprendeu representações ricas de atributos para uma ampla variedade de imagens. Além disso, como a configuração de peso da VGG19 está disponível ao público, foi possível aplicar a versão pré-treinada da VGG19 em nosso trabalho.

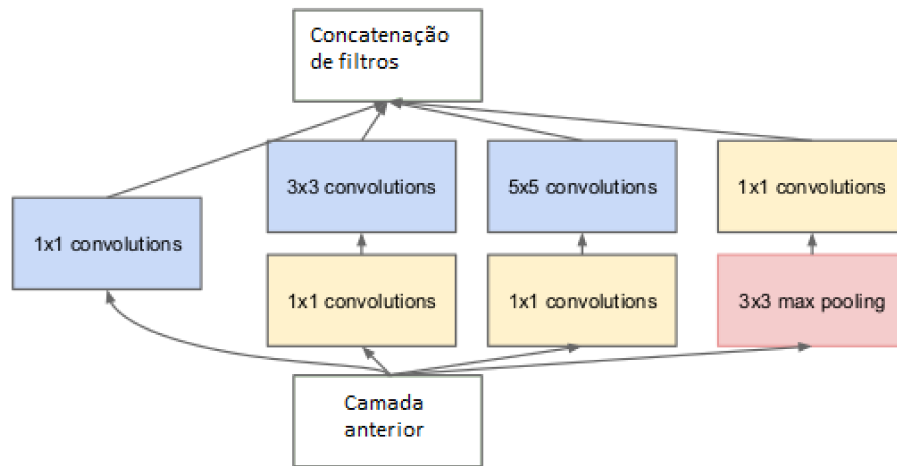
3.5 INCEPTIONV3

O módulo Inception, e conseqüentemente a arquitetura Inception, foi proposto por [43]. O objetivo do módulo Inception é atuar como "extrator de recursos de vários níveis", computando as convoluções 1 x 1, 3 x 3 e 5 x 5 dentro do mesmo módulo da rede, a saída desses filtros é então empilhada ao longo da dimensão do canal antes de ser alimentada na próxima camada da rede [34]. A figura 19 abaixo apresenta o módulo Inception.

Esta arquitetura possui 42 camadas e atingiu a segunda posição no ILSVRC 2015, com um erro médio de 3,58%. Além disso, o seu design permite que se tenha muito menos parâmetros a serem treinados. Desta forma, este modelo pode ser considerado muito mais eficiente do que os modelos VGGnet. Como sua configuração de peso também está

disponível ao público, também foi possível aplicar a transferência de aprendizado, usando a versão pré-treinada da InceptionV3 em nossa tarefa, em que novamente somente as últimas camadas da rede foram retreinadas para se adaptar ao problema proposto.

Figura 19 – O módulo Inception



Fonte: Extraído de [43].

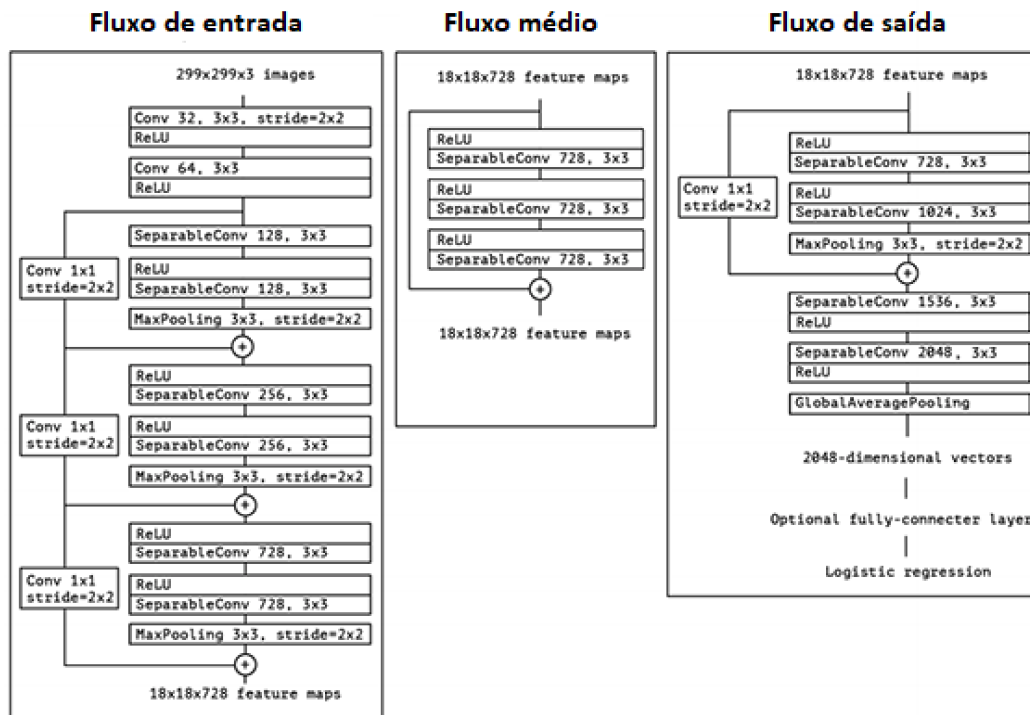
3.6 XCEPTION

Xception, proposta por [11], é considerada uma versão extrema da InceptionV3. A Xception possui 36 camadas convolucionais estruturadas em 14 módulos, todos com conexões residuais lineares ao seu redor, exceto o primeiro e o último módulo, como mostra a Figura 20 a seguir. É possível observar que os módulos *SeparableConvs* são tratados como Módulos Inception e são colocados em toda a arquitetura.

Como já mencionado, este é considerado uma versão evoluída da InceptionV3. Sua performance foi comparada no conjunto de dados *ImageNet* com outras arquiteturas, tais como a própria InceptionV3, ResNet e VGG16, e os resultados demonstraram sua superioridade em relação às demais [44]. No entanto, como este modelo possui uma complexidade considerável, seu desempenho não chega a ser exuberante em pequenos conjuntos dados, como o deste trabalho.

Como a Xception também foi treinada no conjunto de dados ImageNet ILSVRC e como sua configuração de peso também está disponível ao público, também foi possível aplicar a transferência de aprendizado. Sendo, desta forma, aplicada o seu modelo pré-treinado.

Figura 20 – O modelo Xception



Fonte: Extraído de [11].

3.7 CONJUNTO DE DADOS

O conjunto de dados deste trabalho foi obtido em [10]. Este conjunto contém 210 micrografias de aço 1020 em regiões de zonas fundidas por soldagem. As imagens foram obtidas por um microscópio Olympus GX5 e têm uma resolução de 2048 x 1532 pixels. Cada uma dessas micrografias foi quantificada manualmente em [10] seguindo as normas presentes em [4]. Desta forma, temos um problema de aprendizado de máquina supervisionado. A Tabela 1 apresenta as porcentagens volumétricas de algumas imagens do conjunto de dados, vale observar que os dados são composicionais, uma vez que sua soma sempre resulta em 100% [3]. A Figura 21 abaixo representa uma das micrografias com suas respectiva porcentagem microestrutural.

3.8 PREPARAÇÃO DO DADOS

Como as CNNs funcionam melhor com grandes conjuntos de dados, foi necessário a aplicação de técnicas para aumentar o número de imagens. As imagens do conjunto de dados foram espelhadas nos eixos x e y, aumentando em quatro vezes o número de imagens para treinamento. Outras técnicas para aumentar ainda mais o conjunto de dados, tais como rotação e inserção de ruídos nas imagens, também foram aplicadas. Porém, além do alto custo computacional, os resultados não apresentaram uma melhora significativa. A

Tabela 1 – Quantificação microestrutural manual de algumas imagens do conjunto de dados

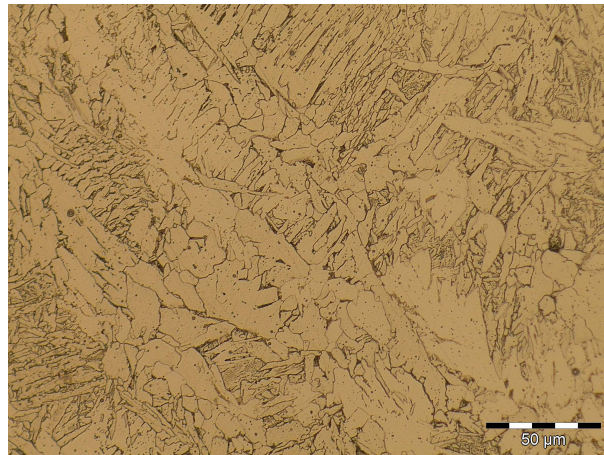
Imagem	PF (G)	PF (I)	AF	FS (NA)	FS(A)
1	26,04%	1,56%	10,94%	47,40%	14,06%
2	34,38%	4,17%	11,98%	39,06%	10,42%
3	33,85%	3,13%	20,31%	42,71%	1,00%
4	31,25%	4,17%	7,81%	47,40%	9,38%
5	33,33%	3,13%	18,23%	40,10%	5,21%
6	45,83%	3,65%	11,98%	37,50%	1,04%
7	41,67%	3,65%	16,67%	35,94%	2,08%
8	50,00%	4,17%	16,67%	23,44%	5,73%

Fonte: Elaborada pelo autor (2020).

Figura 22 apresenta um exemplo de uma imagem quadruplicada.

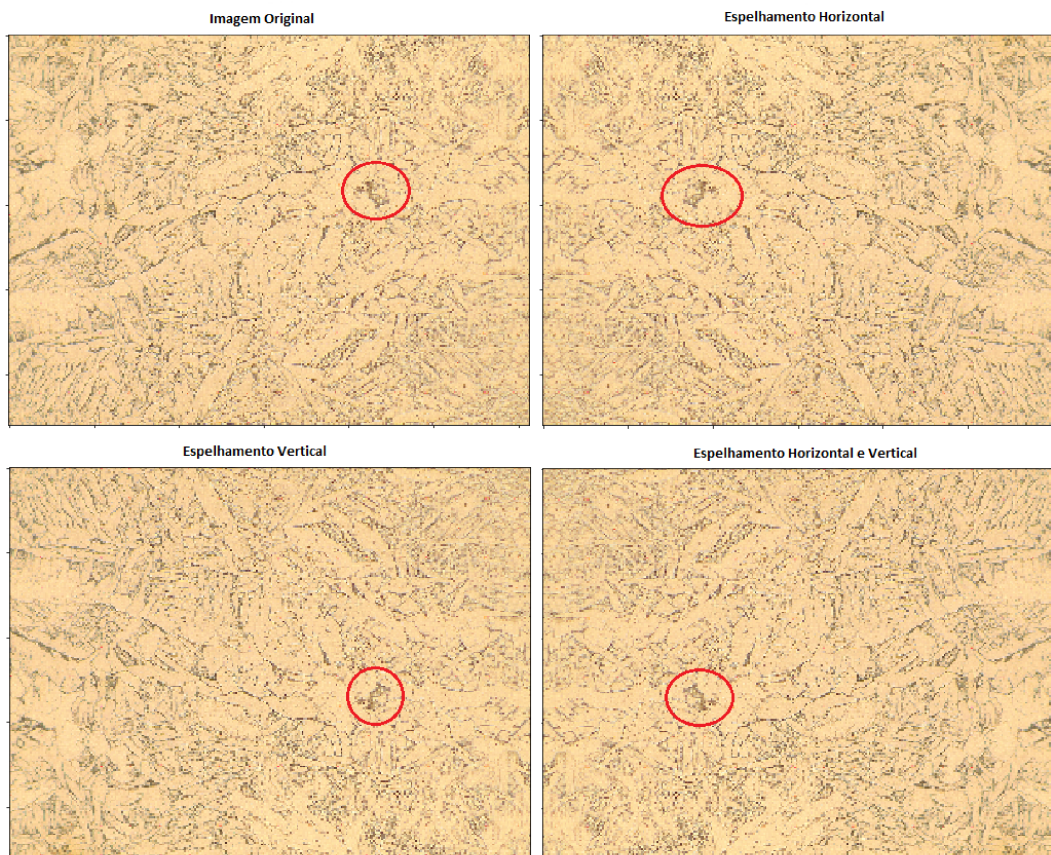
Deve-se ressaltar também, que a resolução original das imagens era muito alta e teria sido muito caro computacionalmente executar o programa com imagens tão grandes. Portanto, tivemos que escolher uma resolução mais baixa sem comprometer os resultados. A melhor opção encontrada, depois de se realizar vários testes com diferentes resoluções, foi aplicar um fator de redução de 0.15, reduzindo as imagens para 307x230, como é possível também se notar na Figura 22.

Figura 21 – Exemplo de micrografia analisada neste trabalho. Microconstituintes: 36% PF(G), 3% PF(I), 16% AF, 34% FS(NA) e 11%FS(A)



Fonte: Extraído de [10].

Figura 22 – Imagem quadruplicada via técnicas de espelhamento



Fonte: Elaborada pelo autor (2020).

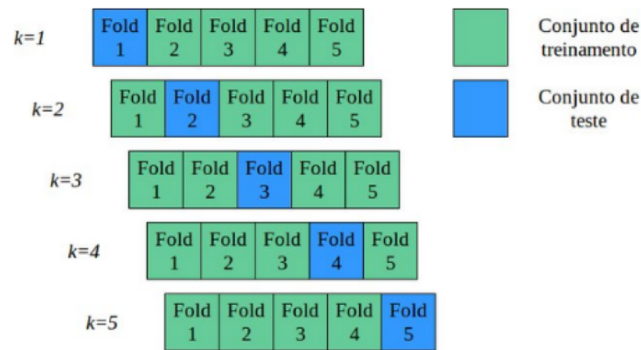
3.9 DETALHES DA IMPLEMENTAÇÃO

3.9.1 Validação Cruzada

Para se obter os resultados e realizar a comparação entre os modelos, foi utilizado o método da Validação Cruzada. Ao se utilizar este método é possível verificar de uma forma mais confiável o poder do modelo, uma vez que essa abordagem testa treina e testa todo o conjunto de dados [21].

A técnica de K-Fold é uma das técnicas mais utilizadas de validação cruzada. Esta técnica divide os dados disponíveis em K partes e utiliza algumas dessas partes para a treinar o modelo e outras partes diferentes para testá-lo [21]. No presente trabalho foi utilizado 5 folds para se validar de forma cruzada os modelos. A Figura 23 mostra esquematicamente um procedimento 5-Fold, onde a cada iteração temos um conjunto de treino e teste diferentes.

Figura 23 – A técnica de validação cruzada K-Fold



Fonte: Extraído de [42].

3.9.2 Treinamento

Das 210 imagens do conjunto de dados, 30 delas foram separadas para teste, restando 180 imagens que foram quadruplicadas pelas técnicas descritas na seção 3.3, obtendo ao final 720 imagens para treinamento. A linguagem de programação *Python* foi utilizada para implementar os códigos, os modelos de CNN foram construídos na biblioteca *Keras* e os dados foram tratados utilizando majoritariamente as bibliotecas *Numpy* e *Scikit-Learn*. Para treinar e testar os modelos, foi utilizada a NVIDIA Tesla T4 GPU (15079 MiB fornecida por *Google Colaboratory*). Cada modelo computacional foi executado 10 vezes em 100 épocas utilizando 5-folds de validação cruzada com dados aleatórios gerados por diferentes sementes aleatórias.

3.9.3 Métricas de Avaliação

O Erro Médio Absoluto (MAE do inglês, *Mean Absolute Error*) mede o valor médio absoluto dos erros em um conjunto de previsões, sem considerar sua direção. É a média na amostra de teste das diferenças absolutas entre previsão e observação real em que todas as diferenças individuais têm peso igual [19]. O MAE pode ser calculado pela equação 26 abaixo.

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - x_i|}{n} \quad (3.1)$$

onde n é número de amostras, y é o valor verdadeiro e x é o valor previsto.

A métrica RMSE (*Root Mean Squared Error*) é a raiz quadrada da variação dos resíduos. Ela aponta a adequação absoluta do modelo aos dados, ou seja, a distância entre pontos de dados observados e os valores previstos do modelo. Como uma métrica de erro padrão, valores mais baixos de *RMSE* indicam melhor ajuste e sendo uma boa medida

da precisão com que o modelo prevê a resposta [19].

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} (y_i - \hat{y}_i)^2} \quad (3.2)$$

onde N é número de dados, y é o valor verdadeiro e \hat{y} é o valor previsto.

R^2 -score, que é a proporção da variação na variável dependente que é previsível a partir da variável independente. Desta forma, o R^2 -score é uma métrica que avalia a adequação do modelo aos dados observados, e seus valores geralmente variam entre 0 e 1, sendo que um R^2 -score próximo de 1 geralmente indica uma boa adequação do modelo aos dados [21]. Sua representação matemática é dada pela equação 25 abaixo.

$$R^2 = 1 - \frac{\sum_i (y_i - \bar{y})^2}{\sum_i (f_i - \bar{y})^2} \quad (3.3)$$

onde y são os valores observados e f são os valores previstos pela rede.

3.9.4 Otimizadores

Em relação ao otimizador, foram empregados neste trabalho o otimizador ADAM e o otimizador SGD (Stochastic Gradient Descent). Para cada modelo foi estudado qual desses otimizadores seria mais apropriado.

O SGD é uma variação da Descida do Gradiente (GD) e é um dos métodos mais tradicionais em problemas de *machine learning*. Estes problemas sempre se consideram o problema de minimização de uma função objetivo que tem a forma de uma soma [19]:

$$Q(w) = \frac{1}{n} \sum_{i=1}^n Q_i(w) \quad (3.4)$$

Onde o parâmetro w que minimiza a função $Q(w)$ tem que ser estimado.

Quando usado pra minimizar a função acima, o método GD (*Gradient Descent*) padrão performaria as seguintes iterações [19]:

$$w = w - \eta \nabla Q(w) = w - \eta \sum_{i=1}^n \nabla Q_i(w) / n \quad (3.5)$$

Onde η é uma taxa de aprendizado do modelo.

Ao invés de performar cálculos no conjunto inteiro como é feito no GD, o que é redundante e ineficiente, o SGD só faz as computações em subconjuntos de dados selecionados aleatoriamente. Apesar de ser um método consideravelmente antigo, o SGD, por sua boa capacidade de generalização, é ainda considerado um dos algoritmos mais

importantes quando se trata de treinamento de CNNs. Entretanto, apesar do SGD ser um otimizador muito robusto e eficaz, ele apresenta algumas imperfeições, como a lentidão de sua convergência em algumas aplicações. [34].

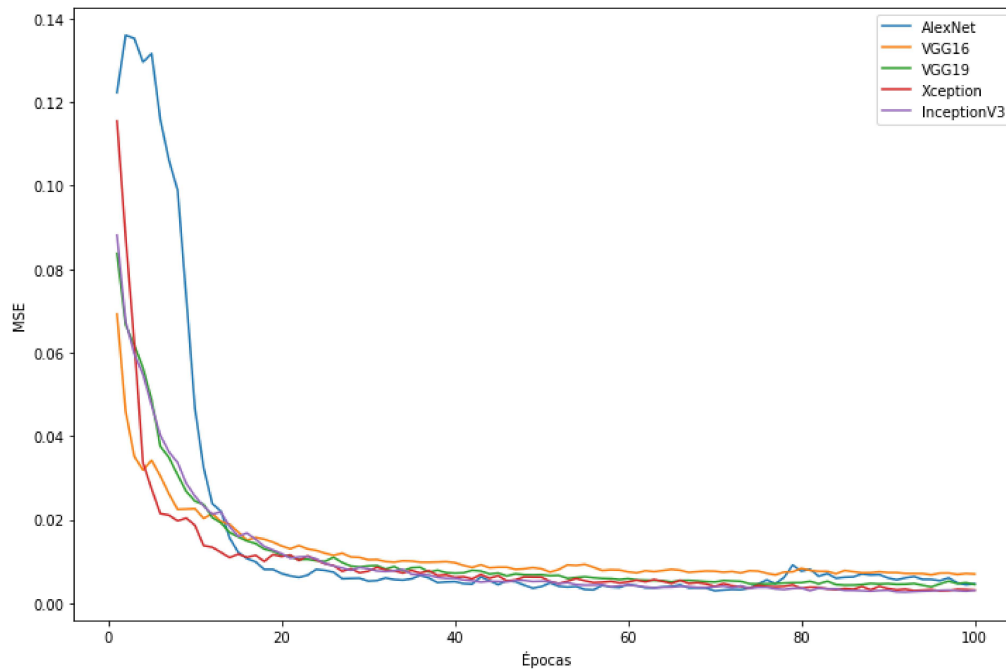
Para aprimorar estes pontos, algumas extensões do SGD foram criadas. O otimizador ADAM pode ser considerado uma dessas evoluções do SGD, apesar de performar pior em alguns casos. Este é um método de taxa de aprendizado adaptável, ou seja, calcula taxas de aprendizado individuais para diferentes parâmetros. Seu nome é derivado da estimativa do momento adaptável, pois no ADAM as estimativas do primeiro e do segundo momentos de gradiente são utilizadas para adaptar a taxa de aprendizado para cada peso da rede neural [22].

Após a realização de vários testes, a melhor combinação de parâmetros encontrada foi a seguinte: para o otimizador ADAM, foi aplicada uma taxa de aprendizado de **0.0001** e, no otimizador SGD, utilizou-se uma taxa de aprendizado de **0.01**, momento de **0.9** e decaimento de peso de 10^{-6} . Em relação aos modelos, somente a arquitetura VGG19 performou melhor com o otimizador SGD, os outros quatro modelos se adequaram melhor ao otimizador ADAM.

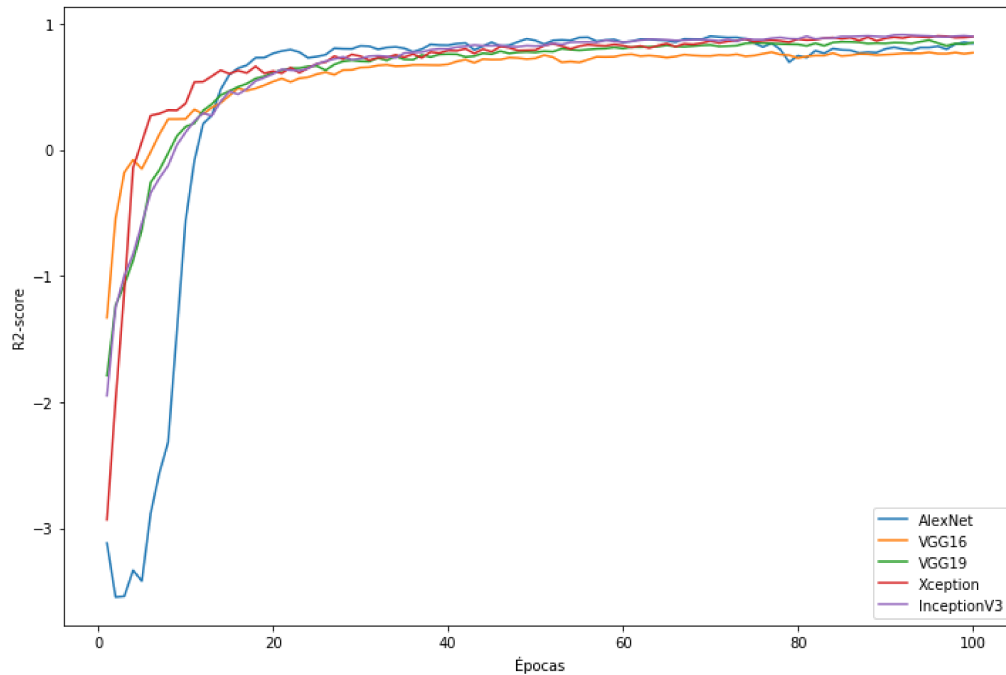
4 RESULTADOS E DISCUSSÃO

As Figuras 24 e 25 abaixo apresentam o erro (MSE) e o R^2 -score, respectivamente, ao longo do treinamento. Como é possível observar, todos os modelos convergiram e conseguiram executar a tarefa proposta.

Figura 24 – Erro ao longo do treinamento por modelo



Fonte: Elaborada pelo autor (2020).

Figura 25 – R^2 score ao longo do treinamento por modelo

Fonte: Elaborada pelo autor (2020).

A Tabela 2 contém os resultados, em termos de tempo de execução e performance, dos cinco modelos CNN, sendo que os resultados de performance são apresentados com os respectivos desvios padrão.

Pode-se notar que AlexNet foi o modelo mais rápido, levando aproximadamente 339 minutos para obter a solução completa, isso é explicado porque este é o modelo menos complexo entre os cinco estudados. Como esperado, a rede VGG16 demorou mais tempo, pois como foi abordado na seção 3.1.2 esta rede é muito pesada para se treinar. Verifica-se também, que a arquitetura VGG19 com otimizador SGD alcançou os melhores resultados em termos de performance.

Tabela 2 – Comparação dos resultados entre os modelos

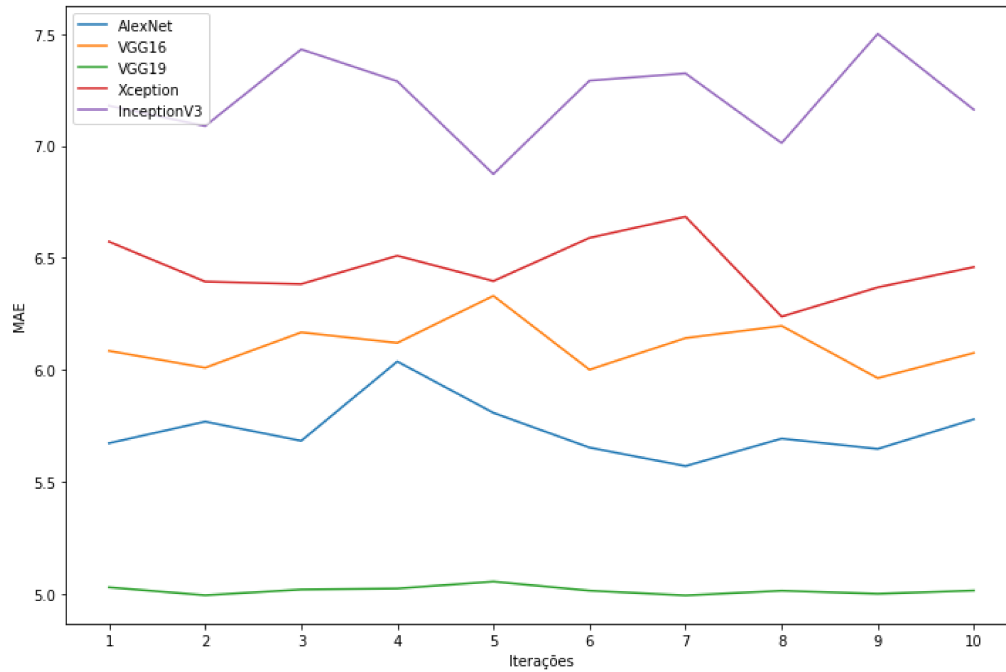
Arquitetura	Duração (min)	R2-score médio	MAE médio	RMSE médio
AlexNet	338,4	0,782 ± 0,009	5,731 ± 0,12	7,978 ± 0,17
VGG16	1508,4	0,758 ± 0,007	6,018 ± 0,10	8,411 ± 0,14
Xception	673,4	0,724 ± 0,013	6,459 ± 0,13	8,978 ± 0,21
VGG19	510,2	0,838 ± 0,001	5,014 ± 0,02	6,876 ± 0,03
Inception	506,8	0,659 ± 0,021	7,215 ± 0,181	9,973 ± 0,29

Fonte: Elaborada pelo autor (2020).

As Figuras 26 e 27 apresentam o erro médio (MAE) e o R^2 -score em cada uma das 10 iterações para cada um dos cinco modelos. É possível observar com mais clareza a

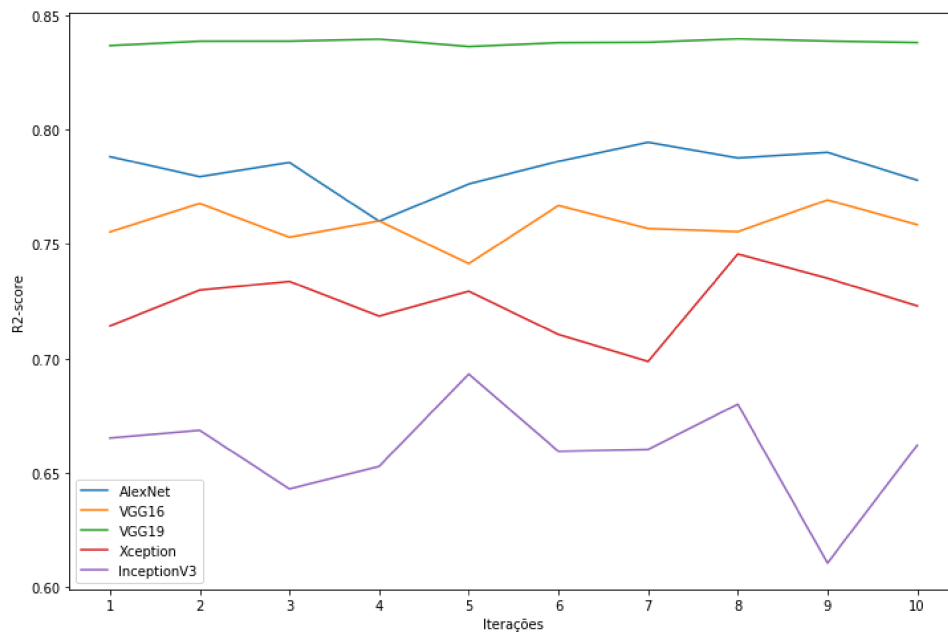
supremacia do modelo VGG19 para esta aplicação, uma vez que além de obter os melhores resultados na média, os mesmos não variaram consideravelmente ao longo das iterações.

Figura 26 – Erro médio em cada iteração por modelo



Fonte: Elaborada pelo autor (2020).

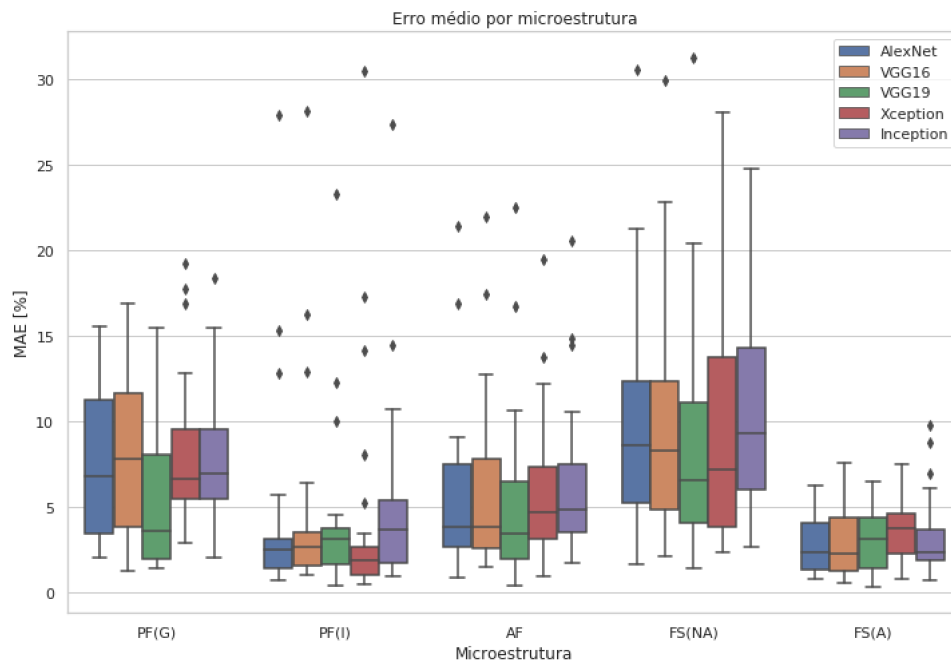
Figura 27 – R^2 score médio em cada iteração por modelo



Fonte: Elaborada pelo autor (2020).

A Figura 28 abaixo apresenta um boxplot do erro médio de cada modelo para os diferentes microconstituintes. No entanto, também é observável que o modelo erra mais em alguns microconstituintes do que em outros. Nossos modelos predizem melhor os microconstituintes que apresentam menor variabilidade entre os dados, como PF (I) e FS (A). Em contraste, microconstituintes como FS (NA) e PF (G) apresentam grande variabilidade entre os dados, o que torna a tarefa mais difícil

Figura 28 – Erro médio de cada modelo por microconstituente



Fonte: Elaborada pelo autor (2020).

A fim de mostrar como os modelos podem prever a porcentagem de microconstituintes com muita precisão, apresentamos as previsões do modelo VGG19 para oito imagens de teste. Estes são exibidos nas Figuras de 29 até 36. Cada micrografia é apresentada com seu respectivo rótulo original de porcentagens, as previsões do modelo e o erro absoluto médio. As porcentagens são apresentadas da seguinte forma: [%PF(G), %PF(I), %AF, %FS(NA), %FS(A)]. Analisando os resultados, é possível observar que o modelo pode atingir valores muito próximos aos valores de entrada. Isto nos mostra que o modelo entendeu a tarefa e sua aplicação pode ser considerada satisfatória.

Figura 29 – Quantificação manual:
 [49%,5.7%,12%,29.2%,4.2%]
 Quantificação prevista:
 [48.9%,5.5%,12.1%,28.9%,4.6%]
MAE: 0.22%

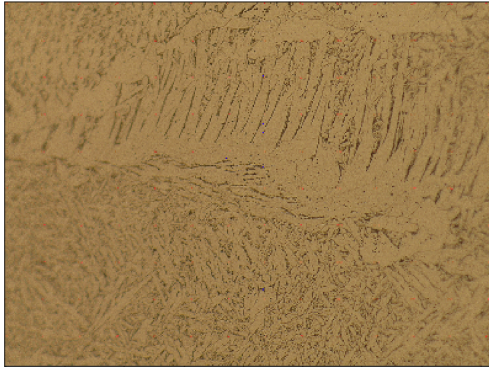
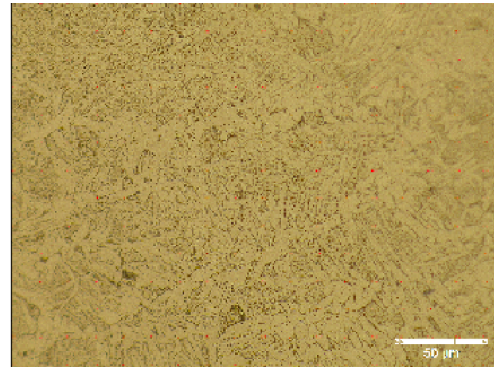


Figura 30 – Quantificação manual:
 [25%,5.7%,10.9%,45.8%,12.5%]
 Quantificação prevista:
 [33.7%,5.5%,12.3%,39.5%,9%]
MAE: 2.42%



Fonte: Elaborada pelo autor (2020).

Figura 31 – Quantificação manual:
 [22.4%,3.1%,19.3%,45.8%,9.4%]
 Quantificação prevista:
 [28.2%,4.9%,15.6%,44.6%,6.8%]
MAE: 4.26%

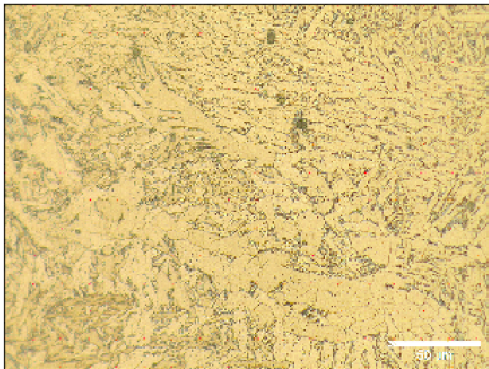
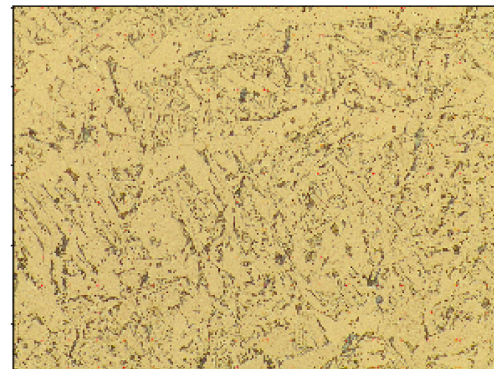


Figura 32 – Quantificação manual:
 [24.5%,2.6%,15.1%,54.2%,3.6%]
 Quantificação prevista:
 [25.7%,3.8%,15.3%,48.4%,6.8]
MAE: 2.32%



Fonte: Elaborada pelo autor (2020).

Figura 33 – Quantificação manual:
[18.8%,1.6%,26%,48.4%,5.2%]
Quantificação prevista:
[21.6%,3.2%,15.3%,54.5%,5.3%]
MAE: 4.26%

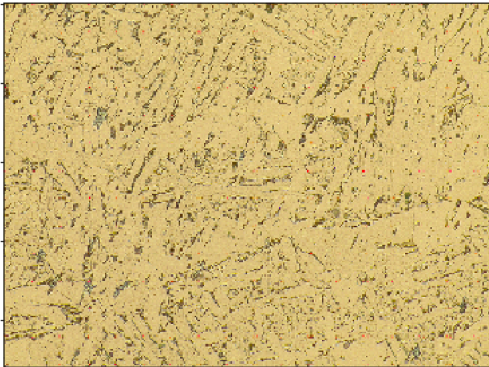
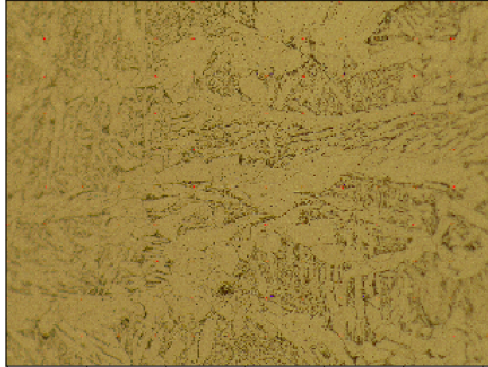


Figura 34 – Quantificação manual:
[31.5%,3.7%,13.9%,41.7%,9.3%]
Quantificação prevista:
[31.2%,7%,16.6%,39.1%,6.1%]
MAE: 2.42%



Fonte: Elaborada pelo autor (2020).

Figura 35 – Quantificação manual:
[45.3%,10.9%,11.5%,28.6%,3.6%]
Quantificação prevista:
[45.9%,7.5%,10.5%,31.4%,4.8%]
MAE: 1.78%

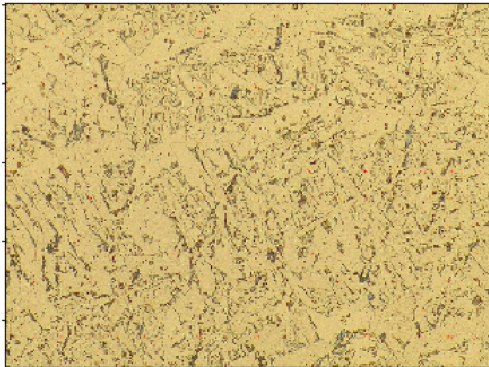
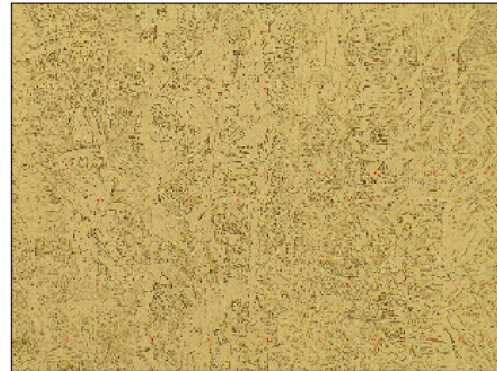


Figura 36 – Quantificação manual:
[31.2%,1%,11.5%,51%,5.2%]
Quantificação prevista:
[25.9%,5%,13.1%,50.6%,5.4]
MAE: 2.32%



Fonte: Elaborada pelo autor (2020).

5 CONCLUSÕES

5.1 CONSIDERAÇÕES FINAIS

Este trabalho investigou a viabilidade da aplicação de CNNs na quantificação microestrutural de micrografias de aço. Foram utilizados as configurações treinadas do zero: AlexNet e VGG16, e as configurações pré-treinadas: Xception, Inception e VGG19. O desempenho das cinco arquiteturas foi comparado sob métricas de erro, acurácia e tempo de execução. Além disso, buscando se obter resultados mais viáveis e satisfatórios, algumas técnicas foram empregadas para se aumentar a quantidade de dados e diminuir a resolução da imagens.

Pode-se concluir que o objetivo geral do trabalho foi totalmente atingido, uma vez que ficou comprovado que as CNNs podem ser empregadas nesta aplicação. Ao se analisar as Figuras 20 e 21, nota-se que todos os cinco modelos convergiram e aprenderam a realizar a tarefa. Além disso, a Tabela 1 e as Figuras 22 e 23 nos mostram que os valores de erro ficaram dentro do aceitável, visto a pouca quantidade de dados disponíveis.

Em relação ao objetivo de se verificar qual modelo de CNN performa melhor para esta aplicação. Temos que de acordo com os resultados obtidos e apresentados na Tabela 1 e nas Figuras 22, 23 e 24, o modelo VGG19 apresentou o melhor desempenho para esta aplicação. Este resultado não é surpreendente, tendo em vista que o modelo VGG19 é um modelo consideravelmente robusto e mais profundo do que a VGG16. Além disso, ao se utilizar sua versão pré-treinada, seu tempo de solução foi otimizado e, como esta versão foi treinada em conjunto de mais de 1 milhão de imagens, ele foi capaz de reconhecer e extrair os atributos das imagens de forma muito assertiva. Os modelos Xception e InceptionV3 são modelos mais complexos e precisam de quantidades de dados maiores para obter melhores performances. Já, o modelo AlexNet, apesar de ter obtido também resultados satisfatórios, é o modelo menos complexo entre os cinco e, portanto, não foi apto a superar o modelo VGG19.

O objetivo de se verificar a performance das CNNs em uma tarefa de regressão também foi atingido. A partir das previsões no conjunto de testes, Figuras 25 a 30, verificou-se que o modelo pode alcançar valores de saída muito próximos aos valores de entrada. É interessante observar também, que o modelo realmente aprendeu a tarefa e reconhece os padrões e atributos de cada imagem, uma vez que suas previsões não são parecidas de imagem pra imagem, pra cada micrografia é feita uma previsão específica e apropriada. Isto comprova a eficácia na aplicação de métodos de aprendizagem profunda nesta tarefa.

Em relação ao último objetivo proposto de se avaliar a performance de modelos pré-treinados e modelos treinados do zero, observou-se que em ambos os casos os modelos

convergir e atingiram resultados plausíveis. Entretanto, levando em consideração que o conjunto de dados disponível é pequeno e o modelo que performou melhor foi um modelo pré-treinado, pode-se concluir que modelos previamente treinados são mais adequados para esta aplicação com este conjunto de dados, tendo em vista as características abordadas na seção 3.1.

É importante enfatizar que a aplicação das CNNs realiza a quantificação dos microconstituintes automaticamente em poucos segundos, o que as torna muito atraentes para a aplicação destes modelos no mundo profissional, pois ao realizar manualmente a contagem das microestruturas, como é feito atualmente, leva-se muito mais tempo e o processo é muito propenso a erros. Desta forma, ao se obter uma maior quantidade de dados e uma infraestrutura computacional mais robusta, podemos aprimorar a performance dos modelos CNNs e com uma construção de uma interface para o usuário poderíamos expandir a ideia deste projeto, modelando uma ferramenta que atenda a indústria, universidades e laboratórios de pesquisa.

Em conclusão, visto que todos os objetivos estabelecidos foram atingidos, espera-se que este trabalho sirva de base para outros estudos na área de quantificação automática de microestruturas, e também como um modelo para a aplicação desta ferramenta computacional na indústria e em centros de pesquisa.

5.2 TRABALHOS FUTUROS

Como trabalhos futuros, é proposto a inclusão de técnicas de segmentação de imagens como abordado em [33], [5], [8]. Ao se utilizar essas ferramentas de segmentação, será possível se identificar e reconhecer cada morfologia separadamente dentro da imagem, fazendo com que o modelo fique mais evoluído.

É proposto também o treinamento dos modelos com o máximo de micrografias que for possível, podendo envolver micrografias de diferentes tipos de aço, com diferentes resoluções e obtidas de diferentes dispositivos microscópios. Adicionalmente, o treinamento feito com uma infraestrutura computacional maior poderá viabilizar mais testes nos modelos, mais execuções e variações.

Por fim, seria interessante transformar este projeto em um produto de fato, para que as indústrias e os laboratórios de pesquisa possam usufruir desse recurso computacional, e comecem a realizar esta tarefa de quantificação, que é tão importante e laboriosa, de forma automática e precisa.

REFERÊNCIAS

- [1] *ImageNet* - <http://www.image-net.org/>, 2020 (acesso em 01 de Abril, 2020).
- [2] A. J. Abson. Acicular ferrite and bainite in c-mn and low-alloy steel arc weld metals. *Science and Technology of Welding and Joining*, 2018.
- [3] J. Aitchison. A concise guide to compositional data analysis. *University of Glasgow*.
- [4] ASTM. *E562 Standard Test Method for Determining Volume Fraction by Systematic Manual Point Count*. American Society for Testing and Materials, 201.
- [5] S. M. Azimi, D. Britz, M. Engstler, M. Fritz, and F. Mücklich. Advanced steel microstructural classification by deep learning methods. *Nature - Scientific Reports*, 8:2128, 2018.
- [6] H. Bhadeshia and R. Honeycombe. *Steels: Microstructure and Properties*, volume 4. Elsevier - Butterworth-Heinemann, 2017.
- [7] D. S. Bulgarevich, S. Tsukamoto, T. Kasuya, M. Demura, and M. Watanabe. Pattern recognition with machine learning on optical microscopy images of typical metallurgical microstructures. *Nature - Scientific Reports*, 2018.
- [8] D. S. Bulgarevich, S. Tsukamoto, T. Kasuya, M. Demura, and M. Watanabe. Automatic steel labelling on certain microstructural constituents with image processing and machine learning tools. *Science and Technology of Advanced Materials*, 2019.
- [9] A. Campbell, P. Murray, E. Yakushina, S. Marshall, and W. Ion. New methods for automatic quantification of microstructural features using digital image processing. *Materials and Design*, 141:395–406, 2018.
- [10] G. Catão and M. Lagares. Análise comparativa do cordão de solda gerado pelos processos de soldagem fcaw e smaw na soldagem de aços de baixo teor de carbono. *Universidade Federal de Juiz de Fora*, 2016.
- [11] F. Chollet. Xception: Deep learning with depthwise separable convolutions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [12] A. Chowdhury, E. Kautz, B. Yener, and D. Lewis. Image driven machine learning methods for microstructure recognition. *Computational Materials Science*, 123:176–187, 2016.
- [13] ComputerScienceWiki. *Max-pooling / Pooling*, 2018 (acesso em 15 de Março, 2020).

- [14] C. Danelon, T. Tozatto, L. Goliatt, M. Lagares, G. Catão, and L. Caldeira. Quantificação automática de microestruturas em aços via redes neurais convolucionais. *XXII Encontro Nacional de Modelagem Computacional (ENMC)*, 2019.
- [15] B. L. DeCost, T. Francis, and E. A. Holm. Exploring the microstructure manifold: Image texture representations applied to ultrahigh carbon steel microstructures. *Acta Materialia*, 133:30–40, 2017.
- [16] R. DeHoff. Engineering of microstructures. *Materials Research*, 2:111–126, 1999.
- [17] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 1980.
- [18] J. Gola, D. Britz, T. Staudt, M. Winter, and F. M. Andreas Simon Schneider, Marc Ludovici. Advanced microstructure classification by data mining methods. *Computational Materials Science*, 148:324–335, 2018.
- [19] A. Géron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. O’Reilly Media, 2017.
- [20] S. Haykin. *Neural Networks and Learning Machines*, volume 3. Pearson Education, 2009.
- [21] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning: with Applications in R*, volume 1. Springer, 2013.
- [22] D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.
- [23] R. Kondo, S. Yamakawa, Y. Masuoka, S. Tajima, and R. Asahi. Microstructure recognition using convolutional neural networks for prediction of ionic conductivity in ceramics. *Acta Materialia*, 2017.
- [24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *NIPS’12 Proceedings of the 25th International Conference on Neural Information Processing Systems*, 1:1097–1105, 2012.
- [25] I. Kuzovkin, R. Vicente, M. Petton, J.-P. Lachaux, M. Baciú, P. Kahane, S. Rheims, J. R. Vidal, and J. Aru. Activations of deep convolutional neural networks are aligned with gamma band activity of human visual cortex. *Nature - Communications Biology*, 1, 2018.
- [26] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

- [27] X. Li, Y. Zhang, H. Zhao, C. Burkhart, L. C. Brinson, and W. Chen. A transfer learning approach for microstructure reconstruction and structure-property predictions. *Nature - Scientific Reports*, 2018.
- [28] G. Lorena, G. Robinson, P. Stefania, C. Pasquale, B. Fabiano, and M. Franco. Automatic microstructural classification with convolutional neural network. *Information and Communication Technologies of Ecuador (TIC.EC). TICEC 2018*, 884, 2018.
- [29] P. J. Modenesi. *Soldabilidade dos Aços Transformáveis*. Universidade Federal de Minas Gerais, 2012.
- [30] J. M. M. Neto. Modelos híbridos para construção de redes neurais artificiais via programação genética cartesiana. *Programa de Pós-Graduação em Modelagem Computacional da Universidade Federal de Juiz de Fora*, 2019.
- [31] I. I. of Welding—IIW. *Guide to the light microscope examination of ferritic steel weld metals*, volume IX. 1988.
- [32] J. P. Papa, R. Y. M. Nakamura, V. H. C. de Albuquerque, A. X. Falcão, and J. M. R. S. Tavares. Computer techniques towards the automatic characterization of graphite particles in metallographic images of industrial materials. *Expert Systems with Applications*, 2013.
- [33] B. S. S. Pokuri, S. Ghosal, A. Kokate, S. Sarkar, and B. Ganapathysubramanian. Interpretable deep learning for guided microstructure-property explorations in photovoltaics. *Nature - Computational Materials*, 95, 2019.
- [34] A. Rosebrock. *Deep Learning for Computer Vision with Python*, volume 1. PyImageSearch, 2017.
- [35] F. Rosenblatt. The perceptron—a perceiving and recognizing automaton. *Cornell Aeronautical Laboratory*, 1957.
- [36] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.
- [37] T. H. Ruoxu Ren and K. C. Tan. A generic deep-learning-based approach for automated surface inspection. *IEEE TRANSACTIONS ON CYBERNETICS*, 2017.
- [38] T. D. Science. *Demystifying Convolutional Neural Networks*, 2018 (acesso em 15 de Março, 2020).
- [39] L. Shen, L. R. Margolies, J. H. Rothstein, E. Fluder, R. McBride, and W. Sieh. Deep learning to improve breast cancer detection on screening mammography. *Nature - Scientific Reports*, 2019.

- [40] S. Shen, S. X. Han, D. R. Aberle, A. A. Bui, and W. Hsu. An interpretable deep hierarchical semantic convolutional neural network for lung nodule malignancy classification. *Expert Systems with Applications*, 128:84–95, 2019.
- [41] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations ICLR*, 2015.
- [42] N. Soares, L. Goliatt, R. Santis, and E. P. de Aguiar. Máquinas de aprendizado extremo para a predição do volume de clones de eucaliptos. *CILAMCE*, 2017.
- [43] C. Szegedy and et al. Going deeper with convolutions. *Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [44] S.-H. Tsang. Review: Xception — with depthwise separable convolution, better than inception-v3 (image classification), 2018 (acesso em 15 de Fevereiro, 2020).
- [45] A. M. Turing. Computing machinery and intelligence. *Mind*, 49, 1950.
- [46] P. Valentin, T. Kounalakis, and L. Nalpantidis. Weld classification using gray level co-occurrence matrix and local binary patterns. *IEEE*, 2018.
- [47] H. Zhao, A. Ho, A. Davis, A. Antonysamy, and P. Prangnell. Automated image mapping and quantification of microstructure heterogeneity in additive manufactured ti6al4v. *Materials Characterization*, 147:131–145, 2019.