

DAS Departamento de Automação e Sistemas
CTC **Centro Tecnológico**
UFSC Universidade Federal de Santa Catarina

Digimetal: medição de metalografia

*Relatório submetido à Universidade Federal de Santa Catarina
como requisito para a aprovação da disciplina:
DAS 5511: Projeto de Fim de Curso*

Gabriel Paiva de Oliveira

Florianópolis, Novembro de 2019

Digimetal: medição de metalografia

Gabriel Paiva de Oliveira

Esta monografia foi julgada no contexto da disciplina

DAS 5511: Projeto de Fim de Curso

e aprovada na sua forma final pelo

Curso de Engenharia de Controle e Automação

Prof. Marcelo Ricardo Stemmer

Banca Examinadora:

Adriano de Souza Pinto Pereira
Orientador na Empresa

Prof. Ricardo Marcelo Stemmer
Orientador no Curso

Prof. Ricardo J. Rabelo
Responsável pela disciplina

Prof. Eduardo Rauh Müller, Avaliador

Victor Lopes, Debatedor

Maria Eduarda Bastos Lübke, Debatedora

Agradecimentos

Gostaria de agradecer e dedicar este trabalho às seguintes pessoas:

Meus pais, pela zelosa atenção e cuidados constantes.

Minha irmã, o meu anjo mais velho, e a seu companheiro, meu parceiro André.

Minha namorada, Naymi pela companhia sempre carinhosa.

Meus professores, exemplos que levarei para vida, em especial ao meu orientador Marcelo Stemmer.

Meus amigos de longa data, que me inspiram e aconselham sempre que podem.

Meus amigos mais recentes, que dividem problemas e cervejas sempre querem.

Meus colegas de faculdade, pela luz no fim do túnel que foram nos momentos mais assustadores.

À todos os outros que de alguma forma participaram dessa jornada.

E, finalmente, eu gostaria de agradecer à UFSC, lugar onde vivi longos e deliciosos anos, permeados de cultura, política, diversão e diversidade.

*"Por definição há cor,
Por definição há doce,
Por definição há amargo,
Mas na realidade há átomos e espaço."*

— Demócrito 460 A.C.

Resumo

Este trabalho versa sobre o desenvolvimento de um ambiente visual para avaliação de imagens de metalografia através de visão computacional. As imagens para análise são derivadas do processo de Deposição Metálica a Laser voltado para manufatura aditiva. Usando o método de desenvolvimento de software Prototipação Evolucionária, o processamento de imagem criado conta com correção de iluminação, contraste, filtro de ruídos e texturas. Para a segmentação das três áreas de interesse criamos uma função que age por seleção de cor e ao final há um *feedback* visual para confirmação ou correção dos parâmetros. No fim, é exibido um relatório de resultado dos testes realizados com três usuários do Laboratório de Mecânica de Precisão da UFSC com três materiais diferentes: ferro, inonel 625 e NiCrBSi. A partir desses resultados, entende-se que o sistema de visão computacional desenvolvido poderá vir a ser treinável por redes neurais, aplicado na web e se aproximar do software que visa substituir, o ImageJ.

Palavras-chave: LMD; Metalografia; Visão Computacional; Visão de Máquina; Feedback visual; Desenvolvimento de software; Segmentação por cor; Área de cordão.

Abstract

This paper is about the development of a visual environment for the evaluation of metallography images using computer vision. The images are made with Laser Metal Deposition process for additive manufacturing. The software development method used is Evolutionary Prototyping. Image processing has correction for illumination, contrast and noise and texture filters. Segmentation of three areas of interest by color selection with visual feedback. In the end shows a report of the test executed with three users from the Precision Mechanics Laboratory (LMP), at UFSC, using three different materials: iron, Inconel 625 and NiCrBSi. From these results, it has come to an understanding that the computer vision system developed can be trainable through neural networks, applied on the web and look closer to the software it intends to substitute, ImageJ.

Keywords: LMD; Metallography; Computer Vision; Machine Vision; Visual Feedback; Software Development; Color Segmentation; Clad Area.

Lista de ilustrações

Figura 1 – imagem de metalografia de um cordão de Inconel 625 produzido no LMP.	20
Figura 2 – Alturas, larguras e áreas selecionadas através do software ImageJ.	21
Figura 3 – Montagem (a) e esquema ótico de um laser com alimentação coaxial de pó (b).	25
Figura 4 – Peça pronta para visualização em dispositivo ótico.	26
Figura 5 – Microscópio ótico com iluminadores bilaterais do LMP.	27
Figura 6 – Taça feita por manufatura aditiva a Laser no LMP.	27
Figura 7 – Imagem ampliada a ponto de mostrar o valor de cada pixel da imagem, nesta área foi desenhada uma linha preta (intensidade igual a zero) de dois pixels de largura.	28
Figura 8 – Infográfico que mostra as influências no projeto de um sistema de visão.	29
Figura 9 – Cronograma criado em agosto para o andamento do trabalho.	34
Figura 10 – Curvas após uma transformação gamma, note que com gamma igual a 1 a luz de saída é a mesma da de entrada.	36
Figura 11 – Imagem processada pelo filtro de ruído com <code>k_size = 13</code>	36
Figura 12 – Gráfico aproximado de como funciona o filtro de textura	37
Figura 13 – Imagem processada pelo filtro de textura com <code>cor = 25</code> e <code>espaço = 30</code>	37
Figura 14 – Barras de ajustes para os parâmetros do programa.	42
Figura 15 – Histogramas que representam as mudanças feitas na imagem pela função <i>isola</i>	43
Figura 16 – Imagem típica na entrada da função <i>calcula_imagem</i>	45
Figura 17 – Imagem original a esquerda e filtrada a direita, este é o feedback visual para o usuário.	46
Figura 18 – A esquerda temos a imagem original e a direita a imagem filtrada. Na parte superior a instrução para a função Correção Gamma.	47
Figura 19 – Gráfico gerado a partir do arquivo CSV dos contornos da imagem 1 In.	48
Figura 20 – Diagrama de fluxo de informação.	51
Figura 21 – A) 1 In; B) 2 Fe; C) 1 NiCrBSi; D) 4 NiCrBSi. A imagem "A" serve de imagem modelo enquanto "B", "C" e "D" são exemplos de imagens problemáticas para o Digimetal.	56
Figura 22 – Diferença entre as médias de tempo de um programa e outro.	58
Figura 23 – Tempo de cada usuário, por imagem no software Digimetal.	59
Figura 24 – Tempo de cada usuário, por imagem no software ImageJ.	59
Figura 25 – Forma básica do perceptron: valores de entrada são multiplicados por pesos e então somados, se o resultado passar pelo limiar a saída é positiva, se não passar é negativa.	65

Figura 26 – Perceptron de múltiples capas. 66

Lista de tabelas

Tabela 1 – Exemplo de Tabela de Saída desajada	23
Tabela 2 – Dimensões medidas pelo Digimetal e ImageJ.	39
Tabela 3 – Imagens usadas nos testes e seus nomes.	40
Tabela 4 – Tabela de variáveis.	49
Tabela 5 – Diferença entre os erros de medidas lineares e de medidas de áreas.	52
Tabela 6 – Tabela comparativa de erros relativos por usuários para medidas de largura do cordão, usando a média das imagens.	52
Tabela 7 – Tabela comparativa de erro relativo entre as medidas de cordão por imagem, usando a média dos usuários.	53
Tabela 8 – Tabela comparativa de erros relativos por material, média de todos os usuários e imagens.	54
Tabela 9 – Em verde as medições que foram bem realizadas seguindo os critérios estabelecidos.	55
Tabela 10 – Tabela com quantidades acertadas por cada usuário.	55
Tabela 11 – Tabela comparativa entre os erros das áreas por usuário, média de todas as imagens.	57

Lista de abreviaturas e siglas

AZTA - Area da Zona Termicamente Afetada

DED - Deposição Direta de Energia (do inglês *Direct Energy Deposition*)

HAZ - Área da Zona Termicamente Afetada (do inglês *Heated Affected Zone*)

LMD - Deposição Metálica a Laser (do inglês *Laser Metal Deposition*)

LMP - Laboratório de Mecânica de Precisão

RGB - Vermelho, Verde e Azul (do inglês *Red Green and Blue*)

UFSC - Universidade Federal de Santa Catarina

PI - Proporcional e Integral

PID - Proporcional Integral Derivativo

Sumário

1	INTRODUÇÃO	19
1.1	Contexto da problemática	19
1.1.1	Sobre o laboratório	20
1.1.2	Objetivos	21
1.1.3	Objetivos específicos	22
2	REVISÃO BIBLIOGRÁFICA	25
2.1	Deposição metálica a Laser	25
2.1.1	Metalografia	26
2.2	Visão Computacional	27
2.2.1	<i>Visão Computacional e de Máquina: Teoria, Algoritmos, Práticas</i>	28
2.3	Estado da arte em medição por imagem de cordões de solda	29
3	ABORDAGEM PROPOSTA	33
3.1	Prototipação Evolucionária	33
3.2	Efeitos de Imagem	35
3.2.1	Brilho e Contraste	35
3.2.2	Correção Gamma	35
3.2.3	Filtro de Ruídos	35
3.2.4	Filtro de Textura	36
3.2.5	Escala de cinza	37
3.2.6	Filtro de Limiarização	38
3.2.7	Achar contorno	38
3.2.8	Criar polígono	38
3.3	Validação do método	38
4	IMPLEMENTAÇÃO E RESULTADOS	41
4.1	Ferramentas criadas	41
4.1.1	Função <i>Main</i>	41
4.1.2	<i>Tracks</i>	42
4.1.3	Função <i>isola</i>	43
4.1.4	Linha média automática	44
4.1.5	Linha média manual	44
4.1.6	Retângulo alto e baixo	44
4.1.7	Função <i>calculo_imagem</i>	45
4.1.8	Exclui Apen de Azta	46

4.1.9	Visualização das imagens	46
4.1.10	Contorno superficial	47
4.2	Arquitetura do programa	47
4.2.1	Inicialização do programa	48
4.2.2	Avaliando Imagens	49
4.2.3	Finalização	50
4.3	Validação do programa	51
4.3.1	Medição de largura do cordão	51
4.3.2	Diferença entre os dados	52
4.3.3	Diferença entre materiais	53
4.3.4	Imagens bem avaliadas	54
4.3.5	Áreas de Interesse	56
4.3.6	Tempo: comparação entre softwares	57
4.3.7	Tempo: comparação entre usuários	58
5	CONCLUSÃO E PERSPECTIVAS	61
5.1	Sugestões	63
5.1.1	Seleção de área e preenchimento pelo método de pontos	63
5.1.2	Segmentação por textura	64
5.1.3	Redes Neurais	64
5.1.4	Aplicação na Web	65
	REFERÊNCIAS	67

1 Introdução

A visão humana é, provavelmente, o sentido ao qual mais prestamos atenção durante a vida e é também um dos mais propícios a nos iludir: chamamos esses efeitos de ilusões de óptica. As imagens são valiosas para todas as culturas humanas conhecidas, desde as pinturas rupestres até as artes digitais como os atuais memes, ou ainda desde fotografias em preto e branco até imagens geradas por satélites que nos permitiram ver a silhueta real de um buraco-negro, em 2019. [1] A nossa visão é uma composição da imagem física projetada na retina com o processamento posterior que se passa no nosso cérebro, fornecendo uma interpretação final que nos permite analisar situações e tomar decisões.

Uma rotina de visão computacional tem um mecanismo análogo; há um sistema de aquisição, digitalização, armazenamento, e processamento de imagem. Assim é feita a análise desta imagem, gerando entendimento sobre a ela, ou seja, transformando-a em descrições do ambiente que se comunique com outros processos e que determinem uma ação apropriada, e desta forma produzindo informação na forma de dados ou de decisões lógicas. Pode-se visualizar esse entendimento como a extração de informações simbólicas de uma imagem com auxílio de modelos geométricos, físicos, estatísticos e de aprendizado [2].

1.1 Contexto da problemática

O objeto de estudo deste trabalho é a melhora do processo de medição de metalografia do Laboratório de Mecânica de Precisão (LMP) por meio da utilização de processamento de imagem e visão computacional. Para isto, foi desenvolvido um software que faça a medição de imagens de metalografia, determinando áreas e dimensões do cordão de Deposição Metálica a Laser (do inglês, *Laser Metal Deposition - LMD*). Esse estudo é conduzido no LMP sempre que se testa novos parâmetros, materiais ou técnicas de operação do Laser, porém acredita-se que esse processo pode ser melhorado. As imagens representam cortes transversais de linhas de solda, caracterizando-as pelos valores de suas áreas, larguras e alturas. Existem 3 áreas de interesse, e dentro destas áreas ainda há 4 dimensões a serem determinadas.

Na Figura 1 pode-se visualizar uma imagem com uma área preta no centro e uma mancha esbranquiçada na sua parte inferior. A área preta no centro é o material adicionado via Laser, a mancha esbranquiçada é a parte do substrato que reagiu à temperatura do processo. Para revestir o material e protegê-lo da oxidação é adicionada baquelite preta em cima da peça.

A área de deposição (*Clad Area*), em preto, está localizada acima do substrato,

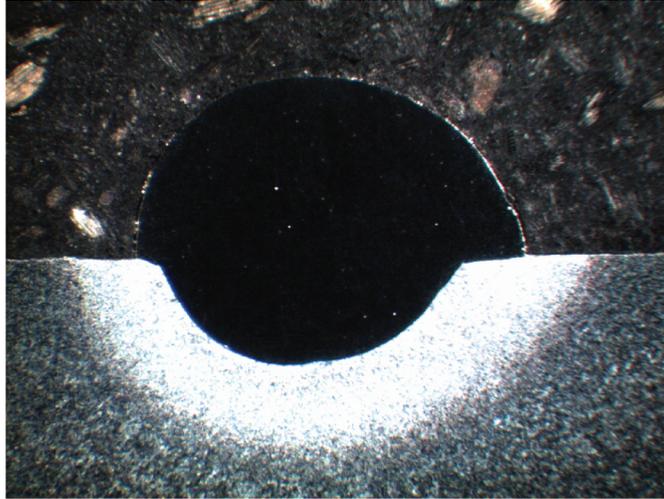


Figura 1 – imagem de metalografia de um cordão de Inconel 625 produzido no LMP.

e a área de penetração (*Melt Pool Area*), também em preto, fica na parte do substrato (inferior) e conectada a área de deposição. Por fim, temos a Zona Termicamente Afetada (do inglês, *Heated Affected Zone - HAZ*), esta é a área mais clara logo abaixo da área de penetração. Também é desejado que o programa informe:

- largura máxima da área de deposição;
- largura do cordão na superfície (largura na altura de 5%);
- altura da área de deposição; e
- altura da área de penetração.

Além dessas medidas, também é possível determinar o formato exato do contorno superficial do cordão para comparar com outros estudos de previsão de geometria de cordões.

1.1.1 Sobre o laboratório

Localizado dentro do Bloco B do Departamento de Engenharia Mecânica da UFSC, o Laboratório de Mecânica de Precisão (LMP) se destaca como um dos laboratórios com maior projeção nacional e internacional dentro da UFSC. O laboratório trabalha com manufatura aditiva, que tem por diferencial ser feita a laser, um processo que abrange muitas possibilidades, mas também aposta na inovação de novas tecnologias para continuar se destacando na sua área.

O papel que o autor exerce no LMP como estagiário é muito gratificante, o seu objetivo é substituir um processo de medição manual por um de medição automatizada por software. Dessa forma o produto deste trabalho é poupar que outros membros do

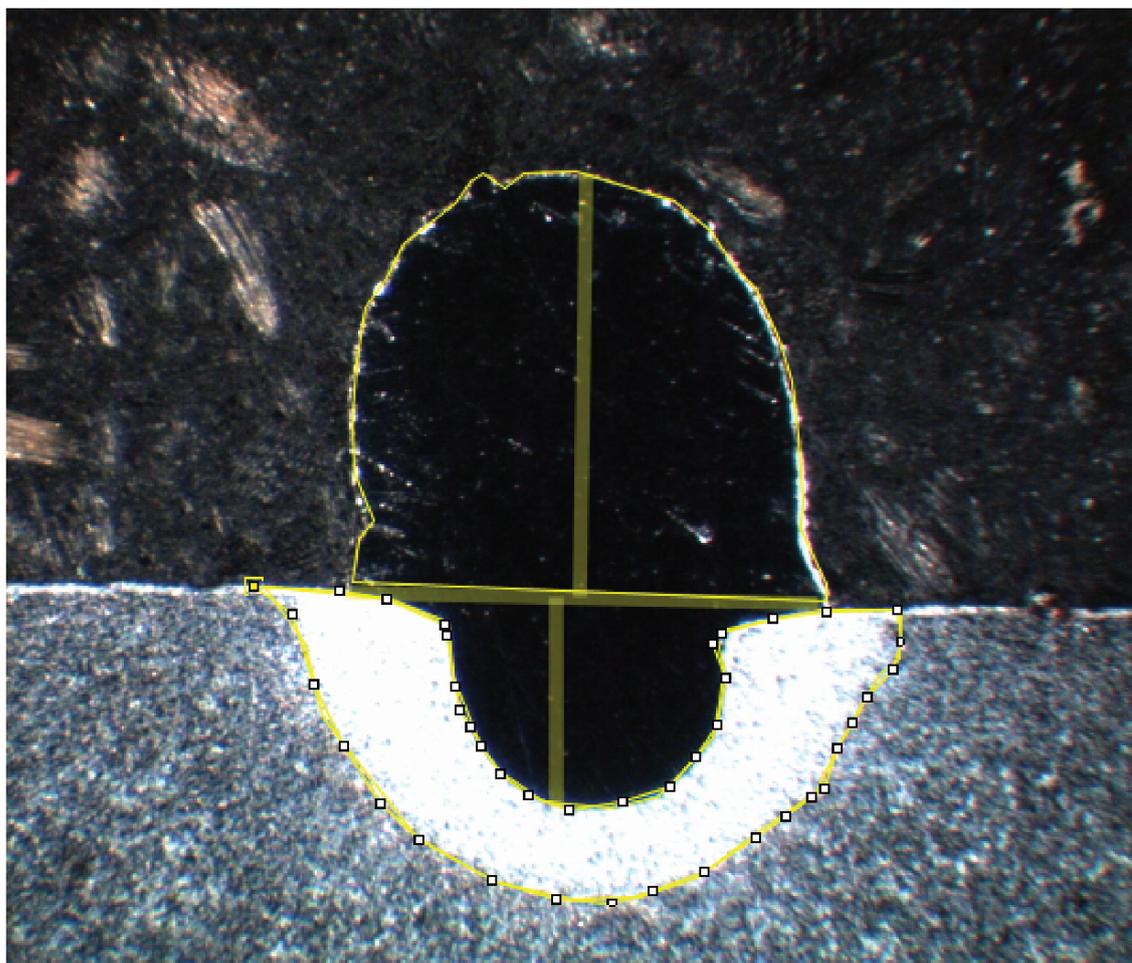


Figura 2 – Alturas, larguras e áreas selecionadas através do software ImageJ.

laboratório executem esforços repetitivos, de baixa gratificação e com pouco valor científico, que é a medição de peças de metalografia.

1.1.2 Objetivos

Validar a hipótese de que usando um software de visão computacional pode-se obter melhores resultados e menor dependência do usuário. E para isso foi, produzir um software intuitivo, que possua uma curva de aprendizado rápida e que seja o mais invariante ao usuário quanto possível, ou seja, que consiga medir e exportar as medidas e o contorno superficial de uma imagem de metalografia com resultados e que estes sejam semelhantes ao de outros usuários.

A produção do algoritmo abordado neste relatório foi um pedido de membros do laboratório que já trabalham com as peças de metalografia e com a medição assistida por computador. É desejado que o software Digimetal seja um substituto para o processo de medição atual, que pode ser visto na Figura 2.

São requisitos comuns de um software de visão computacional:

1. o algoritmo deve funcionar com a informação ideal;
2. o algoritmo deve funcionar com informação com até $x\%$ de defeito;
3. o algoritmo deve funcionar com p pixels de acurácia;
4. o algoritmo deve operar dentro de s segundos;
5. o algoritmo pode ser treinável;
6. o algoritmo deve ser implementado com uma taxa de falha inferior a y por 100 imagens.
7. o custo do material para implementar o algoritmo deve custar menos do que z R\$.

[3]

Para o trabalho atual foram especificados objetivos de tempo e acurácia já vistos no capítulo anterior. O custo de implementação é zero pois não foi necessário fazer nenhuma aquisição de material, somente foram feitas sugestões quanto as características de contraste e iluminação farta e equilibrada bilateralmente. O algoritmo pode ser adaptado para se tornar treinável por Redes Neurais e Algoritmo Genético, ver sugestões. Não foram estabelecidos critérios de informação ideal ou defeituosa. Procura-se saber neste presente trabalho quais as limitações do software desenvolvido.

1.1.3 Objetivos específicos

Para validar a hipótese o produto proposto deve possuir, para a maior parte das áreas avaliadas:

- medição mais rápida;
- qualidade nas medidas um erro mínimo, referente ao ImageJ, de no máximo:
 - 15% para área de deposição;
 - 30% para zona termicamente afetada; e
 - 25% para área de penetração.
- exportação dos pontos (x,y) que caracterizam o contorno superficial da linha de solda; e
- saída de dados em forma de tabela, tal como a tabela de exemplo.

O ImageJ não oferece nenhuma garantia de acurácia ou informação sobre erro de medição, ainda assim ele é utilizado no LMP e usaremos a sua medida como valor de referência.

A tabela de saída de dados serve para analisar os dados obtidos e pode ser usado para comparar o software desenvolvido com o que deve ser substituído. Pode-se ver na Tabela 1.1.3 que a primeira linha possui o nome das colunas onde, da esquerda para direita temos: arquivo, área de cordão, poça de penetração, Zona Termicamente Afetada, largura de 5%, altura, largura máxima e altura da poça de penetração.

file	Clad Area	Penetration	HAZ	width	height	max width	M.P. height
A2	690.54	160.54	0.0	125.35	50.66	91.99	95.36
A3	605.34	150.92	511.00	71.65	153.65	150.45	665.9

Tabela 1 – Exemplo de Tabela de Saída desajada

2 Revisão Bibliográfica

Este Capítulo proporciona uma fundamentação teórica acerca da problemática abordada no presente trabalho. Aqui é possível entender os conceitos de LMD (ou DED, de forma mais genérica) e como técnicas de visão computacional podem ser úteis para verificação da qualidade do processo metalúrgico. Por fim, foi feita uma pesquisa acerca de artigos que versam sobre o uso de software para interpretação de imagens de cordões de LMD.

2.1 Deposição metálica a Laser

O LMP tem como seu carro-chefe a técnica de LMD (Deposição Metálica a Laser, do inglês, *Laser Metal Deposition*) para estudo de materiais e técnicas de manufatura aditiva. O equipamento do laboratório possui diversos parâmetros de ajustes e pode atuar em diversos tipos de materiais. Cada combinação dos parâmetros pode formar um cordão (*clad bead*, do inglês) com geometria diferente, então, entender essa geometria significa entender a deposição metálica feita.

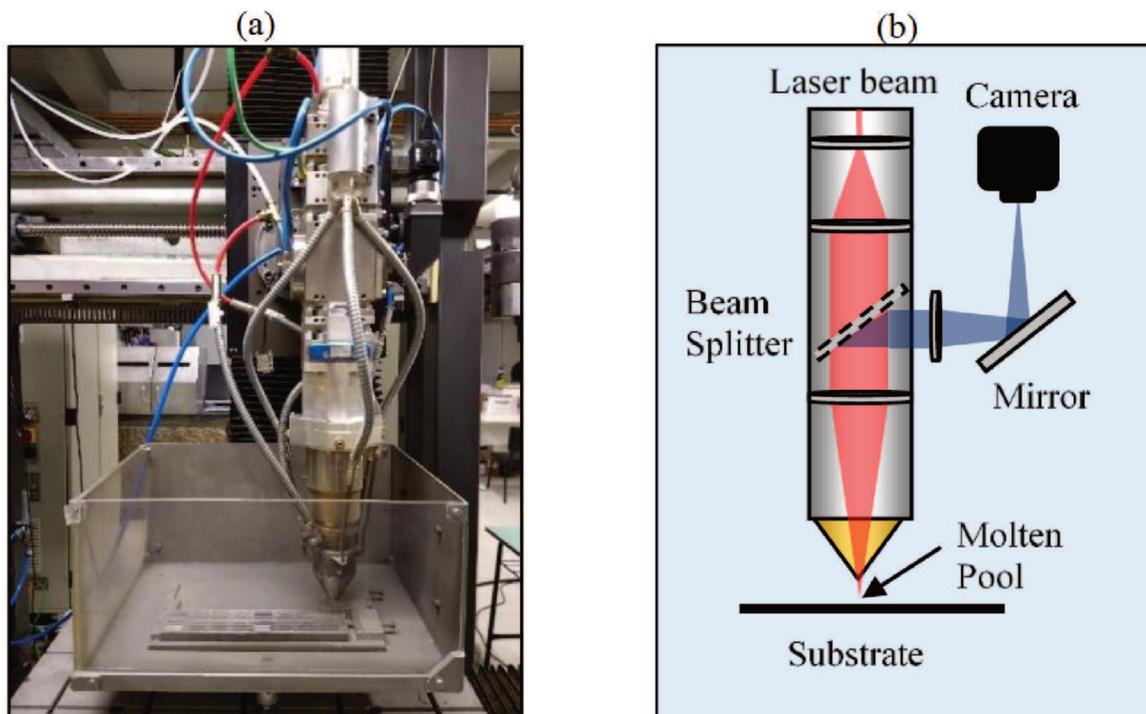


Figura 3 – Montagem (a) e esquema ótico de um laser com alimentação coaxial de pó (b).

Fonte: Gonçalves [4]

2.1.1 Metalografia

O estudo de cordões depende da medição de peças-teste, de forma que cada peça deve passar por um processo de preparação para que o material seja protegido e suas medidas permaneçam inalteradas.

O processo contempla uma sequência de fases, de forma que cada fase tem um objetivo específico, como apresentado abaixo: - Corte: exposição transversal; - Polimento: uso de lixa; - Tratamento químico: ataque com ácido; e - Revestimento: embutimento de baquelite preta.

Na Figura 4 podemos ver um exemplo de uma peça que já passou por esses processos de preparação. Esse estudo de cordões faz parte da rotina científica do laboratório.



Figura 4 – Peça pronta para visualização em dispositivo ótico.

Para que as peças possam ser analisadas, ainda é necessário fotografá-las em um mecanismo de amplificação óptica, como pode ser visto na Figura 5, e só então transferidas para um computador, onde se pode efetuar as medições e então converter os dados de píxel para micrômetro.

O resultado desses estudos com cordões isolados de clad possibilita que o processo seja repetido, adicionando cordão em cima de cordão, e então ir dando forma a peças mais complexas, através de manufatura aditiva de metal em pó. A Figura 6 apresenta uma peça feita com Aço Inoxidável ASTM 316L no LMP.



Figura 5 – Microscópio óptico com iluminadores bilaterais do LMP.



Figura 6 – Taça feita por manufatura aditiva a Laser no LMP.

2.2 Visão Computacional

Visão Computacional é o conjunto de técnicas computacionais de processamento e interpretação de imagens, neste trabalho foi utilizada a biblioteca OpenCV para Python [5]. Esse campo de estudo da informática trata as imagens como matrizes bidimensionais em que cada elemento é chamado de píxel (do inglês *pixel*, aglutinação de *picture* com *element*) e possui informação sobre a cor do ponto da matriz em formato de vetor [5]. Se a imagem

possui cores, o elemento terá mais de um valor numérico. Caso seja codificado em RGB, um valor será para a intensidade da cor vermelha, outro verde e o terceiro para azul. Para uma imagem em tons de cinza, os píxeis possuem somente uma informação, referente à intensidade de cinza que o ponto representa na imagem, como pode ser visto na Figura 7.

93	77	69	55	49	47	47	46	46	46	46	46
98	91	71	65	61	48	47	47	46	46	46	46
99	95	89	70	64	53	48	47	47	46	46	46
103	100	98	86	70	63	63	52	48	47	47	46
103	103	103	97	86	72	63	61	55	50	48	48
103	103	103	103	99	91	81	72	64	64	64	64
102	102	102	103	103	101	97	93	89	89	89	93
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
250	250	248	248	248	248	249	245	250	247	250	250
250	250	250	250	250	250	250	250	250	250	250	250

Figura 7 – Imagem ampliada a ponto de mostrar o valor de cada pixel da imagem, nesta área foi desenhada uma linha preta (intensidade igual a zero) de dois píxeis de largura.

Para alguns autores, o sistema de visão de computacional começa na aquisição da imagem, passa pelo processamento da imagem, posteriormente pela segmentação de área de interesse e finalmente a medição, que permite alguma tomada de decisão por parte do programa [6]. O sistema de aquisição de imagem do LMP está sujeito a problemas comuns de imagem, então o estudo destes se fez necessário, para Davies [3] esses problemas são:

1. ruído;
2. oclusão do fundo da imagem ou da imagem principal;
3. objeto defeituoso ou com trincas;
4. distorções ou perspectivas ópticas;
5. luz não uniforme (e suas consequências);
6. efeitos da luz dispersa.

2.2.1 Visão Computacional e de Máquina: Teoria, Algoritmos, Práticas

Em seu livro o professor Roy Davies [3], da *University of London*, ajuda a pesquisadores e praticantes de visão computacional a entender conceitos, métodos e práticas relevantes de visão computacional. Já tendo escrito centenas de artigos no assunto, o

autor ensina diversas possibilidades disponíveis no mundo da visão computacional, desde processamento simples de imagens, até técnicas de ajuste de sistemas de aprendizado de máquina, da forma correta de fazer a aquisição de imagens e também em como criar um sistema de visão computacional em tempo real. Na Figura 8 estão expostos os temas abordados no livro e quais áreas estão envolvidas neles. Este livro foi consultado e lido sempre que haviam dúvidas quanto a melhor abordagem para um determinado problema.

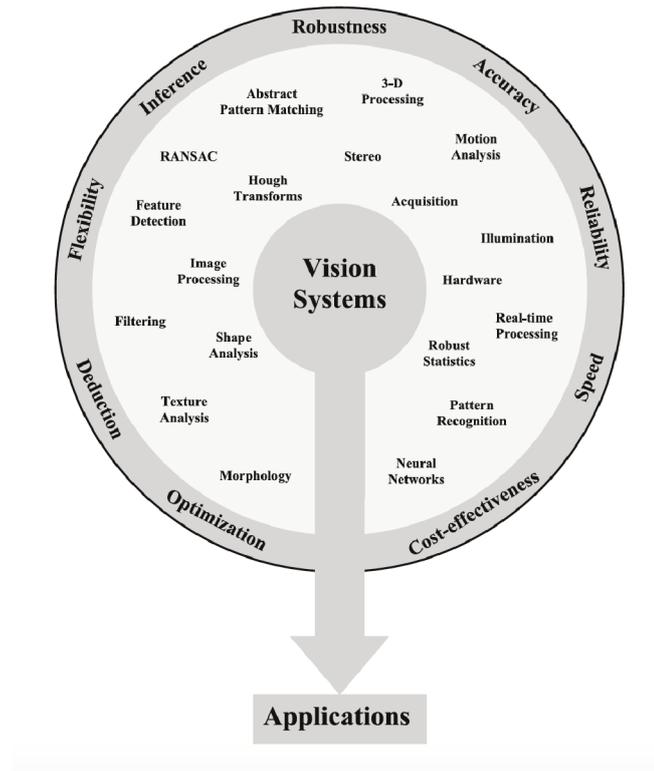


Figura 8 – Infográfico que mostra as influências no projeto de um sistema de visão.

Fonte: [3]

2.3 Estado da arte em medição por imagem de cordões de solda

Para saber quais tecnologias do Estado da Arte estão sendo desenvolvidas na área de visão computacional para aplicação em imagens de metalografia, foram feitas pesquisas em bancos de dados científicos, como o Google Scholar e Scopus, além de contar com sugestões diretamente de professores do Departamento de Automação e Sistemas (DAS), da UFSC, e ter acesso a todo acervo científico do Laboratório de Mecânica e Precisão.

Na dissertação *A Convolutional Neural Network Approach On Bead Geometry Estimation for a Laser Cladding System*, a autora Denise Gonçalves [4] propõe um método que estima dimensões de um cordão para o processo de LMD. Através da biblioteca Keras para inteligência artificial, a autora desenvolve redes neurais convolucionais que são

capazes de estimar a geometria do cordão com alta acurácia. Esta dissertação teve grande influência no presente trabalho pois ambos foram produzidos no Laboratório de Mecânica e Precisão da UFSC no ano de 2019.

Para a série de trabalhos intitulada *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, por [Hu e Kovacevic](#) [7], foram utilizadas duas câmeras CCD (*charge-coupled device*) para capturar ondas do espectro infra-vermelho por um ângulo vertical, pegando somente as banda de ondas relativas à temperatura. Combinando uma imagem do espectro e luz visível com a imagem do espectro infravermelho (IR do inglês, *infrared*), com um filtro de limiarização de cinza, pode-se visualizar o contorno da poça fundida e contar a quantidade de pixel dentro da poça. Com essa informação alimenta-se um controlador PI (proporcional com ação integradora) pode-se controlar o tamanho da poça fundida ao variar a potência do laser. O uso de filtro de limiarização também foi utilizado no Digimetal, uma vez que este fitro permite selecionar cores, ou no caso do trabalho em análise, radiação do espectro IR.

Semelhantemente ao trabalho citado acima, em *Camera Based Feedback Control of the Laser Cladding Process* foi utilizada uma câmera para aquisição de ondas relativas à temperatura do objeto, desta vez usando uma câmera CMOS (*complementary metal-oxide semiconductor*). Então a imagem adquirida passava por processamento para suavização, limiarização, adaptação para uma elipse, extração das informações da elipse e então conversão de pixel para mm e mm². Com essas informações [Hofman, Lange e Meijer](#) [8] utilizaram um controlador digital PID (proporcional integral derivativo), a fim manter a largura da poça fundida constante e garantir a menor diluição do material no processo, por meio da variação da potência do laser.

O trabalho *Development of an image-based feature tracking algorithm for realtime clad height detection*, conduzido inicialmente por [Toyserkani e Khajepour](#) [9], utilizou uma câmera CCD posicionada num ângulo raso fixo com uma lâmpada para obter uma imagem lateral da poça fundida. Essa disposição obriga que o sentido do cordão seja sempre o mesmo. Para binarizar a imagem foi utilizada uma técnica fuzzy criada pelos autores, e então aplicado um controlador digital PID que consegue determinar a altura do cordão, através de variação na potência do laser, enquanto rejeita perturbações do processo.

Na continuação desse trabalho (*An image-based feature tracking algorithm for real-time measurement of clad height*), os autores [Iravani-Tabrizipour e Toyserkani](#) [10] desenvolveram um sistema com três câmeras CCD focando no centro da poça fundida. Esse novo arranjo proporcionou independência na direção do cordão a ser formado. [11]

Já [Barua, Sparks e Liou](#) [12], em um trabalho mais moderno chamado de *Development of low-cost imaging system for laser metal deposition processes*, valeram-se da teoria de corpos negros, que afirma que um corpo negro não emite radiação ultravioleta. Os autores iluminaram a poça fundida com esse tipo de radiação e filtraram a imagem resultante

no espectro ultravioleta, obtendo a forma da poça fundida. Para obter a imagem foi feito um processamento de imagem utilizando a biblioteca OpenCV, para visão computacional.

A partir da análise da bibliografia estudada, o autor do presente trabalho começa a identificar campos com potencial para maior utilização de técnicas de visão computacional e o quanto esse campo se relaciona diretamente com as áreas de Controle e de Automação. Ainda há muito campo de atuação que não aproveita as potencialidades dessa técnica. Os materiais de referência base foram de importância essencial para as soluções que serão abordadas nos capítulos vindouros, visto que ofereceram exemplos e teorias que ajudaram no desenvolvimento do novo produto.

3 Abordagem Proposta

Neste capítulo explica-se o método de desenvolvimento do software, as funções existentes de processamento de imagem, medições das dimensões e das áreas de interesse. A estruturação dos testes e a explicação da forma como eles foram realizados para validar do produto são o assunto final deste capítulo.

3.1 Prototipação Evolucionária

Este método clássico de desenvolvimento propõe o uso de protótipos para ajudar na comunicação entre o desenvolvedor e o cliente, principalmente na elaboração da arquitetura, da interface e da definição dos requisitos do sistema. [13] No caso o cliente era representado por Adriano Pereira, supervisor do autor no laboratório.

Wazlawick [13] explica em seu livro de Engenharia de Software que no processo de prototipação evolucionária cria-se um protótipo inicial, a fim de validar algum objetivo específico. Este protótipo é apresentado ao cliente, para que faça comentários e considerações. A avaliação do cliente é usada para realimentar o processo, podendo gerar melhorias protótipo ou insumos para as fases seguintes, passando por sucessivas evoluções até que se aproxime ou atinja os objetivos iniciais. Um protótipo inicial pode ser de dois tipos:

1. Descartável (*throw-away*): são usados unicamente para estudar aspectos do sistema e aumentar a compreensão sobre ele. Após cumprir essa função o protótipo é descartado.
2. Pedra fundamental (*cornerstone*): esse protótipo também é utilizado para aumentar a compreensão sobre o sistema ou estudá-lo. Porém o protótipo fará parte do programa final, ou seja, ele vai evoluindo até que se torne um sistema que atenda os requisitos e que possa ser entregue.

Para a produção desse programa foram feitos protótipos do tipo pedra fundamental. Um primeiro protótipo, com objetivo de testar filtros, foi o *Filtros Em Sequência*, que consistia de um conjunto de 10 efeitos de imagem que eram utilizados sequencialmente, modificando continuamente a imagem original através de uma *trackbar*. Desta forma, foi possível estudar quais filtros eram relevantes e/ou necessários, além de avaliar quais as configurações de ordem e magnitude dos filtros de suavização geravam melhores resultados na remoção dos ruídos, levando em conta a qualidade das linhas divisórias da imagem processada. Este protótipo serviu de base para o desenvolvimento da função *Tracks*, posteriormente.

Outro protótipo desenvolvido foi o *FiltraMedeDesenha*, um programa que possuía parâmetros de filtros modificáveis em código, executava uma sequência de filtros e segmentava a imagem de forma semelhante a usada na função *Main* do *Digimetal*, e por fim, escrevia as medidas calculadas e mostrava a imagem com os contornos desenhados. A partir deste protótipo foi desenvolvida a função *Main* deste presente trabalho.

Posteriormente, esses dois protótipos foram fundidos para que, em uma janela, fossem exibidas várias *trackbars* para configurar os parâmetros dos filtros utilizados na rotina *FiltraMedeDesenha* no final. Este protótipo recebeu novas funcionalidades como:

- selecionar a linha média com cliques;
- exportar medidas em arquivo .xlsx;
- exportar contorno superficial do cordão em arquivo .csv;
- mostrar instrução sobre a *trackbar* modificada na imagem e;
- exibir a imagem original ao lado da imagem resultante, com os contornos desenhados na original.

Deve-se fazer a ressalva que este modelo não é bom em relação a previsão de tempo de desenvolvimento. Porém durante o processo de produção deste trabalho foi feito um cronograma com quais funções deveriam estar funcionando dentro do prazo. Este cronograma pode ser visto na Figura 9.

LMP LASER	Cronograma de Atividades - 2019.2 - Gabriel Paiva de Oliveira															
	Atividades - Laboratório de Mecânica de Precisão - Laser															
	Agosto				Setembro				Outubro				Novembro			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Função nova para correção de iluminação Gamma	■	■														
Escolha dos filtros mais relevantes dentre os já desenvolvidos durante o estágio		■														
Criar uma janela com várias trackbars		■	■													
Concatenar duas imagens para serem exibidas na mesma janela			■	■												
Criar textos de instrução para o usuário (toast)			■	■												
Juntar as funções em um script separado do programa principal	■			■												
Exportar dados medidos para CSV							■									
Estudar formato do contornos em OpenCV							■	■								
Exportar contornos para CSV							■	■								
Gerar um executável do programa								■	■							
Ilustrar a linha lógica do programa								■	■							
Teste do programa por usuário final (funcionários do LMP da área de engenharia de materiais)								■	■	■						
Repensar a lógica da função de detectar Linha Média									■	■						
Rotina de detectar Linha Média alternativa, através de clique										■	■					
Implementar modificações sugeridas por usuário final											■	■				
Testes com novo banco de dados para validar o projeto												■	■			
Elaboração de Documentação de Software e Manual de Uso													■	■	■	■

Figura 9 – Cronograma criado em agosto para o andamento do trabalho.

3.2 Efeitos de Imagem

Esta seção explica as funções de efeitos de imagens, disponíveis na biblioteca OpenCV, que foram utilizadas no Digimetal, bem como o encadeamento de estados pelo qual passa a imagem original: desde os pré-filtros até as funções de identificação de contorno.

3.2.1 Brilho e Contraste

O brilho e o contraste de uma imagem são expressos pelo valor de intensidade de cinza que cada píxel assume de acordo com a Equação:

$$g(i, j) = \alpha.f(i, j) + \beta$$

A equação acima representa a luminosidade "g" em cada píxel na linha "i" e coluna "j", sendo "f" a cor inicial que é multiplicada pelo contraste e somada ao brilho, representados por alfa e beta, respectivamente [5].

3.2.2 Correção Gamma

A função matemática que descreve a função Correção Gamma é a lei de potência:

$$s = c.r^y$$

onde "s" é a luminosidade final, "r" é a luminosidade inicial, e "c" e "y" são constantes positivas.

Através dessa relação nós podemos mapear os valores escuros em vários claros, assim como valores claros em poucos claros, para um gamma inferior a 1, como podemos ver na Figura 10 o valor na entrada correspondente a L/4 é mapeado para, aproximadamente, 3L/4 quando gamma possui valor . Este efeito nos permite corrigir a iluminação da imagem, clareando ou escurecendo como um todo, tanto no brilho quanto no contraste, simultaneamente [14].

3.2.3 Filtro de Ruídos

A função que aplica o filtro de mediana em OpenCV é *cv.medianBlur*, criando uma matriz de dimensão *k_size* x *k_size* para operação, onde o píxel sendo operado é o central. O píxel operado recebe o valor da mediana dos valores dos píxeis pertencentes a matriz de abertura.

Apesar de bom para remoção de ruídos esse filtro apresenta a características de diminuir o contraste de contornos e bordas. Podemos observar o seu efeito na Figura 11. Cada canal de cor (vermelho, verde e azul) é processado independentemente [5].

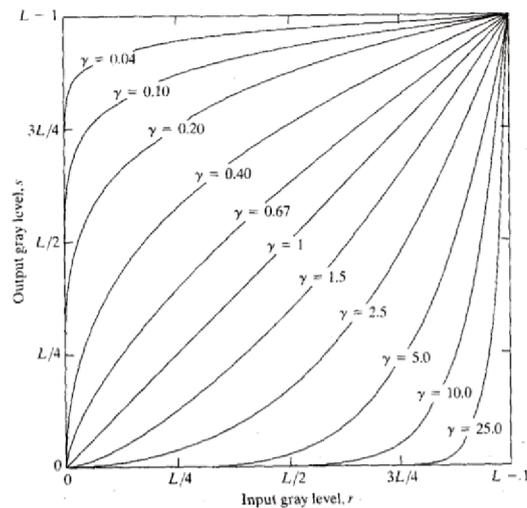


Figura 10 – Curvas após uma transformação gamma, note que com gamma igual a 1 a luz de saída é a mesma da de entrada.

Fonte: [14]

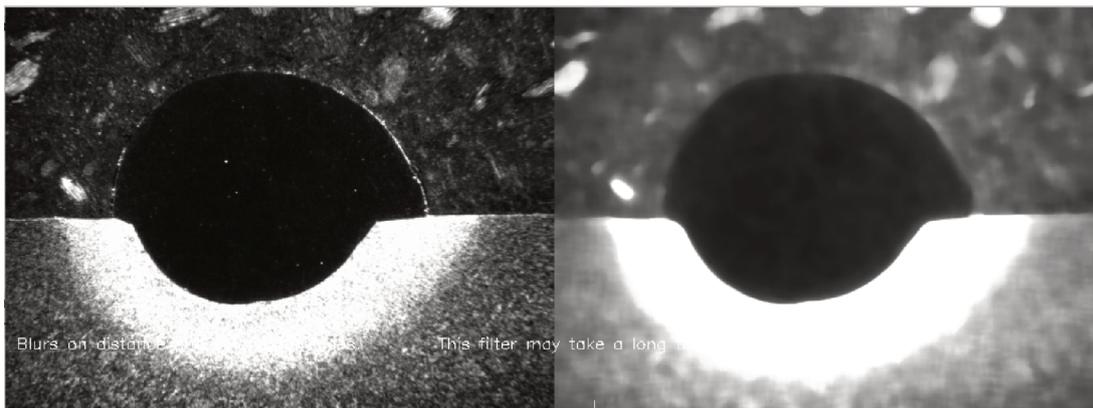


Figura 11 – Imagem processada pelo filtro de ruído com $k_size = 13$.

3.2.4 Filtro de Textura

Qualquer ponto x no espaço RGB converge a um ponto local máximo adjacente a x . Se trocarmos todos os pontos da imagem para o seu máximo local, temos uma suavização da imagem e uma mudança da média de cor desses pontos. Na Figura 12, a linha verde mostra a suavização da imagem.

O Filtro em questão se chama *cv.pyrMeanShiftFiltering* e atua na informação referente à distância dos píxeis, não só na cor, para que píxeis distantes não sejam afetados. Assim, pode-se corrigir os dois parâmetros na imagem, de forma que a diferença de espaço seja mais relevante que a diferença de cor entre dois píxeis [15]. Na Figura 13 pode-se observar um exemplo do resultado obtido pela aplicação do Filtro de Textura.

Então podemos acertar esses dois parâmetros na imagem, fíemos de forma que a

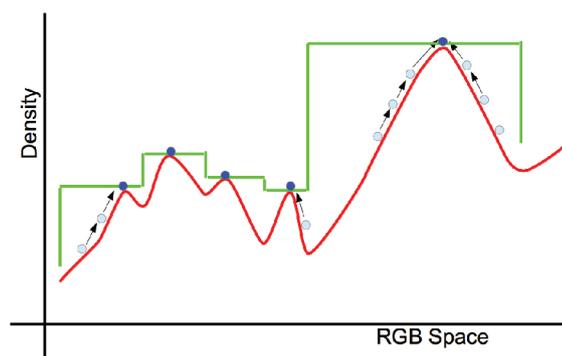


Figura 12 – Gráfico aproximado de como funciona o filtro de textura

Fonte: [15]

diferença de espaço seja mais forte que a de cor, na Figura podemos ver um exemplo desse filtro .

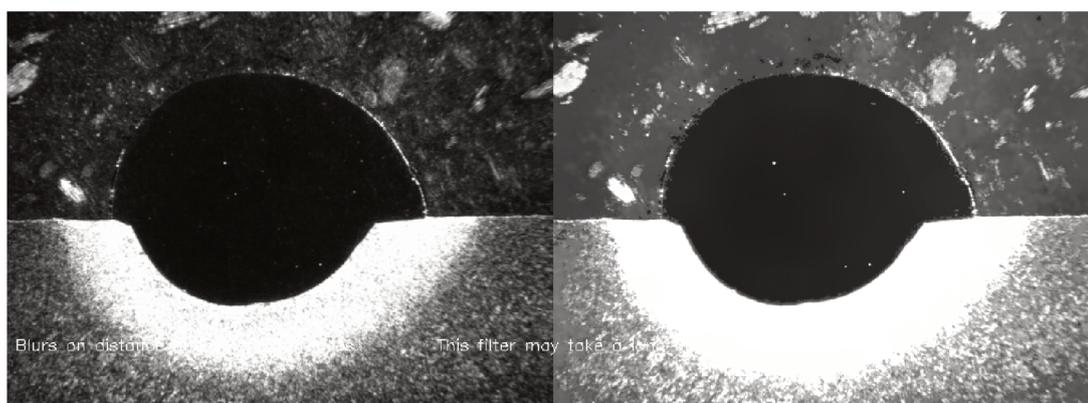


Figura 13 – Imagem processada pelo filtro de textura com cor = 25 e espaço = 30.

3.2.5 Escala de cinza

Transformar uma imagem colorida em escala de cinza é uma operação matemática complexa, pois faz uma transformação de um espaço de 3 dimensões para o de 1 dimensão evitando perder informação, somando os valores de R, G e B de cada imagem de acordo com a equação:

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B$$

Onde R, G, B e Y são dados de intensidade de cor, entre 0 e 255 [5].

3.2.6 Filtro de Limiarização

Esse tipo de filtro recebe um determinado valor limiar como parâmetro, podendo agir de duas formas diferentes em relação a um píxel. Caso o valor calculado do píxel seja superior ao valor limiar, ele permanece inalterado. Caso o valor calculado do píxel seja inferior ou igual ao limiar, o píxel é alterado para 0, ou seja, transformado em preto. O método usado foi `cv.threshold_tozero` [5].

3.2.7 Achar contorno

A função (`cv.findContours`) detecta todos os contornos brancos em uma imagem de entrada com fundo preto. A saída dessa função é uma lista de pontos representando um polígono pertencente a imagem e uma lista que relaciona a posição relativa desses contornos, chamada de hierarquia [5].

3.2.8 Criar polígono

Essa função não pertence ao pacote OpenCV, mas sim ao Numpy. Ela funciona recebendo um conjunto de pontos para a criação de matrizes de contornos. Desta forma, podem-se criar polígonos, para então inserí-los nas imagens. Primeiro, deve-se escolher os pontos e, utilizando o método `numpy.array`, incluí-los em um array. Em seguida, usando a função `cv.fillPoly`, é possível desenhar o polígono na devida posição sobre a imagem.

3.3 Validação do método

Para validar o programa foram convidados três colaboradores do Laboratório de Mecânica e Precisão que já trabalharam ou trabalham com a criação e medição de imagens de metalografia. Essas pessoas possuem conhecimento técnico prévio no que se refere a área de engenharia de materiais, porém não são programadores e desconhecem os efeitos e aplicações de visão computacional.

A fim de comparar o novo processo com o antigo, os usuários farão a medição de 12 imagens de metalografia tanto no software desenvolvido neste trabalho, o Digimetal, quanto no software que permanece vigente no LMP, o ImageJ. Cada usuário possui uma familiaridade diferente com cada software. O usuário 1, o mestrando Adriano Pereira, já possui muita prática com o ImageJ e foi quem participou como cliente do software Digimetal, especificando os requisitos e participando de todas as reuniões de protótipos. Já o usuário 2, Matheus Ferreira é quem faz as medições com o ImageJ para o laboratório atualmente, possuindo uma leve familiaridade com este programa e tendo sido instruído e treinado rapidamente na utilização do programa Digimetal, tendo acompanhado, com menor participação, da evolução do sistema. Por fim, o usuário 3, Rafael Petters, utilizou

algumas vezes o ImageJ e nunca tinha visto a interface do Digimetal, sendo brevemente instruído das funcionalidades dele no dia do teste.

Desta forma foram medidas, além do tempo para cada imagem, 4 dimensões lineares, dispostas na Tabela 2.

Medida	Simbologia
Largura de 5% do cordão (μm)	5% width
Largura máxima do cordão (μm)	max width
Altura do cordão (μm)	Height
Altura da área de diluição aparente (μm)	Depth
Área de deposição (μm^2)	Clad Area
Área da Zona Termicamente Afetada (μm^2)	AZTA
Área de diluição aparente (μm^2)	Penetration

Tabela 2 – Dimensões medidas pelo Digimetal e ImageJ.

O conjunto de imagens teste foi escolhido pelo usuário 1, procurando colocar o maior número possível de situações complicadoras, gerando um grupo de imagens variadas. Um dos objetivos do teste é comparar o tempo de medição, então enquanto os usuários mediam com o ImageJ o autor cronometrou o tempo demorado para medir a imagem e digitar os valores obtidos na tabela, o Digimetal já faz essa medição de tempo e exporta os valores automaticamente.

Foram testados três diferentes materiais: uma liga de Níquel, Cromo, Boro e Silício, o Inconel 625 e o Ferro. Esses materiais são muito utilizados pelo LMP: A liga NiCrBSi ajuda na resistência e abrasão e é utilizada no revestimento de peças; o Inconel 625 é utilizado no revestimento de peças que precisam de maior resistência à corrosão; o Ferro é utilizado, principalmente, em testes de parametrização do laser, por ser mais barato. Para cada material foram selecionadas quatro imagens e a sua nomenclatura original foi modificada para esse trabalho, como pode ser visto na Tabela 3. Foi mantida a ordem usada na sequência de testes, sendo primeiro feito a medição no ImageJ e depois no Digimetal.

Filename	Referência
Fe - 2018.07.04 - A26	1 Fe
Fe - 2018.07.04 - A35	2 Fe
Fe - 2018.07.04 - A41	3 Fe
Fe - 2018.07.04 - A48	4 Fe
Inconel 625 - 2018.02.27 - A5	1 In
Inconel 625 - 2018.02.27 - A9	2 In
Inconel 625 - 2018.02.27 - A27	3 In
Inconel 625 - 2018.02.27 - A46	4 In
NiCrBSi - 2018.05.10 - A5 - 10X	1 NiCrBSi
NiCrBSi - 2018.05.10 - A9 - 12X	2 NiCrBSi
NiCrBSi - 2018.05.10 - A29 10x	3 NiCrBSi
NiCrBSi - 2018.05.10 - A35 10x	4 NiCrBSi

Tabela 3 – Imagens usadas nos testes e seus nomes.

4 Implementação e Resultados

Este Capítulo explica os pilares erguidos para sustentar o projeto Digimetal. Esses algoritmos foram criados a partir de experimentação com imagens de metalografia e usando os softwares pedra fundamentais de forma encapsulada.

Primeiramente será detalhado o que faz cada função, sua entrada, sua saída, onde é chamada e porque é necessária. Em seguida, será exposta a divisão de etapas que orientou o desenvolvimento do programa, bem como serão explicadas decisões de arquitetura. Por fim, a seção de Validação explicitará os resultados obtidos e comentários sobre o desempenho do Digimetal em comparação com o ImageJ.

4.1 Ferramentas criadas

Tendo em vista a utilização futura do software desenvolvido, bem como questões relacionadas à sua manutenção e evolução, foram seguidas algumas premissas e boas práticas de engenharia, arquitetura e desenvolvimento de software.

A modularização e encapsulamento de métodos é mister para que um código fique mais compreensível para a leitura e, principalmente, facilita a edição, manutenção e evolução das lógicas desenvolvidas. Logo, foi necessário extrair métodos sempre que havia uma parte do código que tinha uma função isolada ou que se repetia.

4.1.1 Função *Main*

Esta a função central do programa, inteiramente programado com técnicas de Visão Computacional, que realiza chamadas para os métodos de Filtragem, Segmentação, Medição e Visualização, a partir as configurações dos parâmetros realizadas pelo usuário por meio das *trackbars*, ou barras de ajuste.

Essa função é chamada pela função *Tracks*, responsável por implementar as barras de ajuste dos parâmetros. Tais valores são as variáveis globais utilizadas na função *Main*. Essas variáveis já possuem um valor inicial, possibilitando uma primeira chamada dessa função antes do usuário modificar alguma barra de ajuste.

Esta função foi desenvolvida a partir da evolução do protótipo *FiltraMedeDesenha*, e realiza as 4 etapas a seguir:

1. Filtragem: correção da iluminação (gamma e contraste), aplicação dos filtros de ruído e textura e transformação da imagem em escala de cinzas;

2. Segmentação: medição das cores de fundo e aplicação da função *isola*.
3. Medição: funções que detectam linha média, contornos e informam suas medidas;
4. Visualização das imagens: organização do resultado das etapas anteriores para análise pelo usuário.

4.1.2 Tracks

Ao executar o programa, o usuário deve realizar configuração inicial do usuário que consiste em informar o diretório em que estão localizadas as imagens que ele deseja examinar e qual a taxa de conversão de pixel para μm . Caso o usuário aperte "Enter" sem digitar nenhuma dessas informações os valores padrão serão usados, o diretório escolhido será o que está executando o programa e a taxa de conversão de pixel para micrômetro será 1, um valor arbitrário.

Na Figura 14 podemos ver as 7 variáveis que podem ser modificadas. As cores de Deposição, Penetração e da Zona Termicamente Afetada (HAZ em inglês) são a entrada da função *isola* que será explicada em seguida.

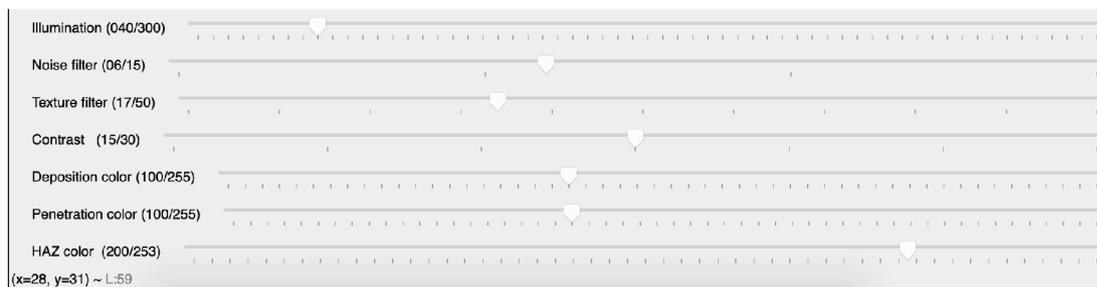


Figura 14 – Barras de ajustes para os parâmetros do programa.

Cada vez que se modifica uma dessas barras de ajuste, uma função *callback* é chamada com o valor selecionado como parâmetro de entrada. Essas funções foram programadas para modificar o valor do parâmetro global, o valor da variável *trackbars*, indicando qual barra foi a última a ser modificada e chamando a função *Main*.

É nessa função que o usuário passará a maior parte do tempo, enquanto analisa o resultado da iteração anterior e planeja qual parâmetro ele deverá modificar para próxima iteração nessa imagem. Além de modificar os parâmetros, essa função oferece algumas possibilidades de interação ao usuário:

- Tecla "ESC": usuário indica que a imagem não pode ser medida adequadamente;
- Tecla "q": sair do programa, salvando as medidas e contornos anteriores; e
- Barra de espaço: usuário indica que a imagem foi medida, encerrando o processamento da imagem atual e passando para a próxima imagem.

4.1.3 Função isola

Começamos pela função mais importante do desenvolvimento, que se comporta de forma análoga a um filtro passa-faixa do espectro escala de cinza. A função *isola* recebe como parâmetro uma imagem, uma cor e uma *flag*, que pode ser verdadeira ou falsa, chamada "escura". Essa função aplica um filtro limiar *cv.thres_tozero*, uma inversão de imagem (*cv.bitwise_not*) e outro filtro limiar *thresh_tozero* com o valor de cor invertido e adicionado 16 unidades, para que forme uma banda passante de cor. Por fim, a flag "escura" determina se haverá mais uma inversão da imagem ou não, antes de retornar a imagem filtrada.

Na Figura 15 podem ser observados três histogramas, nos quais o eixo horizontal informa as cores dos píxeis e o eixo vertical informa quantos píxeis tem cada cor. (Foi utilizada a biblioteca *matplotlib* para plotar os gráficos de histograma.) Na Figura 15.a temos o histograma da imagem original, com píxeis em várias cores. Na Figura 15.b, após uma filtragem, sem nenhum píxel de cor 87 para baixo. E a Figura 15.c representa a imagem ao final do processo, sem píxel nas cores acima de 103. A imagem analisada por este histograma foi usada nos testes. A janela de 16 píxeis foi determinada empiricamente.

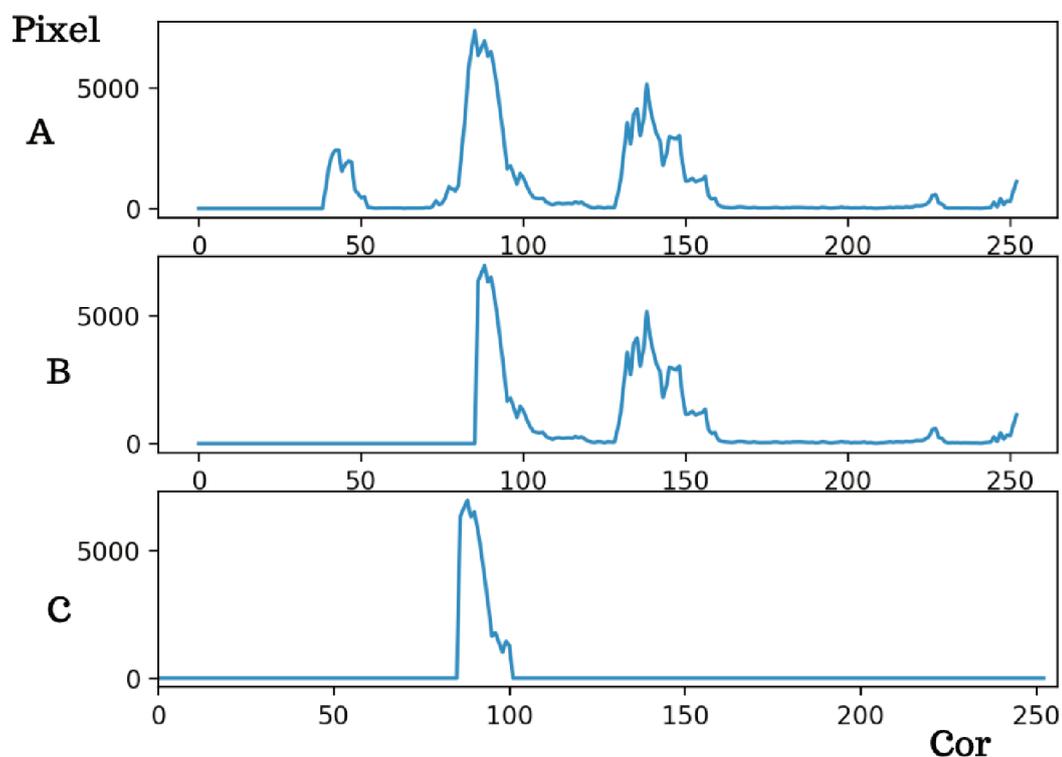


Figura 15 – Histogramas que representam as mudanças feitas na imagem pela função *isola*.

4.1.4 Linha média automática

Essa função descreve a linha que define o substrato, dividindo as áreas de deposição das áreas de penetração e da zona termicamente afetada.

Processa-se a imagem fazendo cortes em faixas longitudinais e detectando a altura que o substrato tem nessas faixas, por altura entende-se a medida do ponto mais inferior de um contorno até o mais superior. Em seguida, constrói-se uma lista com os valores dessas alturas e através da função *numpy.polyfit*, obtem-se uma aproximação de linha de primeiro grau que descreve o substrato do material.

A área central do substrato é desconsiderada, devido à deformação causada pelo processo de LMD. Desta forma, somente os dois quartos externos da imagem são considerados na captura de faixas longitudinais, sendo o primeiro quarto o mais a esquerda e o último, o mais a direita.

4.1.5 Linha média manual

Através de dois cliques com o botão esquerdo o usuário pode definir a sua linha média. Em cada clique aparecerá um L, de forma que seu vértice indica o local clicado. Esses pontos indicam a altura do substrato com informações de "y0" e "yf" (altura inicial e final, respectivamente), e sendo pareados com os valores "x0" igual a 0 e "xf" que é a largura máxima da imagem. Para maior precisão os cliques devem ser feitos próximos à borda lateral da imagem.

Caso o primeiro clique seja mais a direita que o segundo, o programa inverte a ordem e define "yf" para o primeiro clique e "y0" para o segundo. O "L" é desenhado para indicar onde foi efetuado o clique do usuário, com um "L" na imagem o usuário só fez um clique, mantendo a linha média automática, a nova linha média só aparecerá quando o segundo clique (e o segundo "L") for efetuado.

4.1.6 Retângulo alto e baixo

A partir das alturas da linha média ("y0" e "yf") e dos vértices superiores da imagem define-se um quadrilátero a ser preenchido de preto, referente à área a ser excluída da imagem. A referência, o ponto (0,0), é no canto superior esquerdo.

Os pontos do retângulo superior são: (0, y0), (largura_img, yf), (largura_img, 0), (0, 0).

Para o retângulo inferior são: (0, y0), (largura_img, yf), (largura_img, altura_img), (0, altura_img).

Onde a largura_img e altura_img são a largura e altura da imagem.

4.1.7 Função `calculo_imagem`

Nessa etapa do processamento, a imagem analisada será um fundo escuro com áreas brancas, como podemos ver na Figura 16. Estas áreas correspondem aos contornos que, após as filtrações, estivessem dentro de um determinado intervalo de cor.



Figura 16 – Imagem típica na entrada da função `calculo_imagem`.

Assim, usando a função `cv.findContours`, podemos detectar todas essas áreas brancas, elencá-las em uma lista e reorganizar a lista por área, de forma decrescente. Selecionando o primeiro elemento desta lista obteremos a maior área da imagem com a cor desejada, ou seja, somente a área de interesse que foi descrita pelos parâmetros selecionados pelo usuário.

Essa função tem efeitos adicionais, desenhando o contorno selecionado na imagem original, apresentando um *feedback* visual ao usuário.

O retorno da função são as medidas da área selecionada. Usando as funções `cv.boundingRect` e `cv.contourArea`, podemos inferir o ponto inicial (x,y), a largura e a altura da área, bem como a área total do contorno. Na Figura 17 é possível observar uma medição feita através do Digimetal, a figura é dividida em duas imagens. Na imagem da esquerda pode-se observar quatro elementos:

- uma linha branca grossa, indicando o contorno da área de deposição;
- uma linha branca fina, representando o contorno da área de penetração;
- uma linha preta fina, indicando o contorno da zona termicamente afetada.
- Na imagem da direita há uma linha preta grossa, correspondente à linha média do usuário.

Não foram utilizadas cores nas linhas pois a imagem está codificada em escalas de cinza.

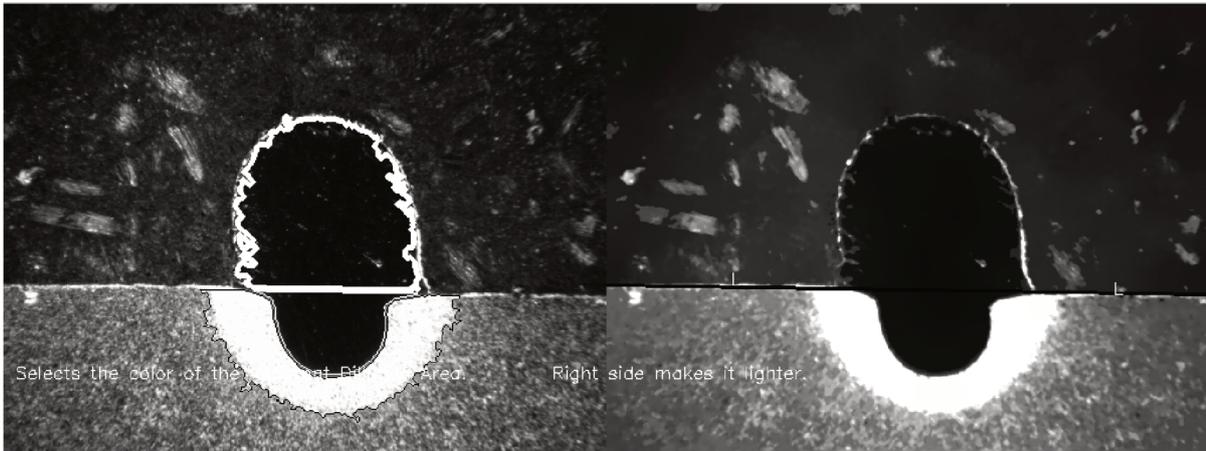


Figura 17 – Imagem original a esquerda e filtrada a direita, este é o feedback visual para o usuário.

4.1.8 Exclui Apen de Azta

Uma das últimas características adicionadas a esse programa é a separação entre as áreas que ficam na parte inferior da imagem, a área de penetração e a zona termicamente afetada.

A área selecionada pela cor de penetração será adicionada, na cor preta, na imagem que selecionará a área da zona termicamente afetada, evitando que essas áreas possuam pontos em comum, já que não podem ocupar o mesmo lugar.

4.1.9 Visualização das imagens

Ao final do processamento da imagem, o programa deve fornecer meios para o usuário avaliar se o processamento ficou adequado. A solução desenvolvida nesse trabalho é a visualização da imagem original ao lado da imagem filtrada.

Na imagem filtrada é necessário colocar a linha média para que o usuário avalie se a detecção automática de linha média foi adequada ou se ele precisa clicar para defini-la. Na imagem original, tal linha não é desenhada, para não ser confundida com outras linhas desenhadas, que são as 3 áreas de interesse. Desta forma, o usuário pode verificar se a detecção feita pelo Digimetal está próxima das áreas originais.

Por fim, de acordo com a barra de ajuste que foi modificada por último, são escritas duas frases curtas que instruem o usuário como utilizar o programa, como pode ser visto na Figura 18.

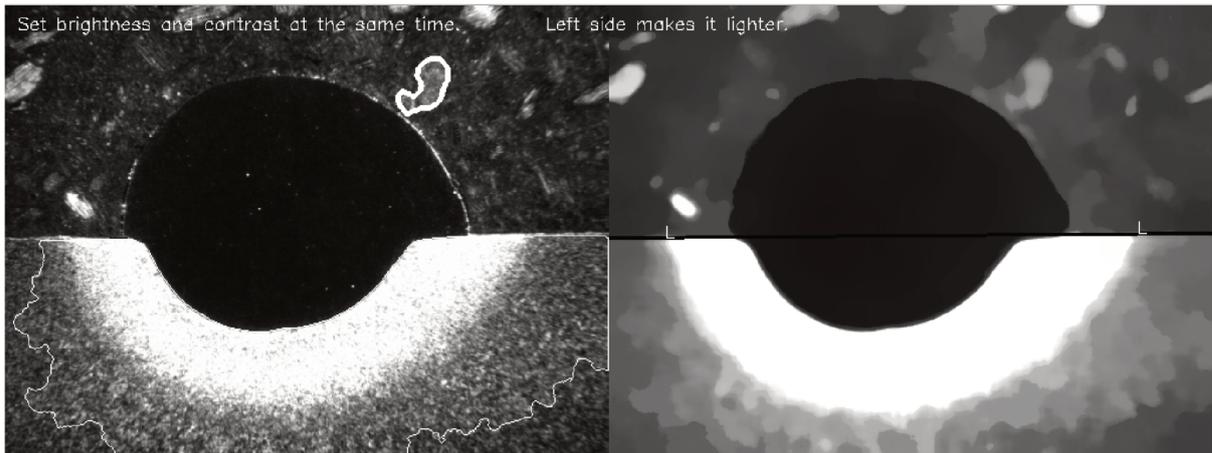


Figura 18 – A esquerda temos a imagem original e a direita a imagem filtrada. Na parte superior a instrução para a função Correção Gamma.

4.1.10 Contorno superficial

Um dos requisitos para o programa era que este exportasse o contorno superficial do cordão de solda. A solução para essa questão foi criar um quadro do mesmo tamanho da imagem original, desenhar o contorno da imagem da Área de Deposição neste quadro e usar a descrição da linha média (tanto automática, quanto manual) para construir um retângulo na parte inferior, onde estaria o substrato.

Esta nova imagem é a entrada de uma chamada da função *cv.findContour*, para que tenhamos a descrição desse grande contorno que é o retângulo substituindo o substrato unido a Área de Deposição. Quando o usuário termina de avaliar uma imagem, a função *exporta_contorno* é chamada e os pontos indexados na matriz de contorno são adicionados a um objeto *Dataframe* do pacote *pandas*, exceto os pontos com "x" igual a 0, "x" igual ao comprimento máximo e com "y" abaixo da reta média. Podemos ver a descrição da imagem Inconel A5 na Figura 19.

4.2 Arquitetura do programa

Para o desenvolvimento do programa Digimetal foi necessário que uma estrutura lógica fosse definida. A escolha foi de criar um laço de processamento de imagem que deverá ser interrompido quando o usuário interpretar que chegou a uma solução correta ou que não há solução do programa para a imagem em questão. Enquanto uma das alternativas não é alcançada, o usuário deve continuar modificando os parâmetros do software como as barras de ajuste ou clicando na imagem para determinar a linha média.

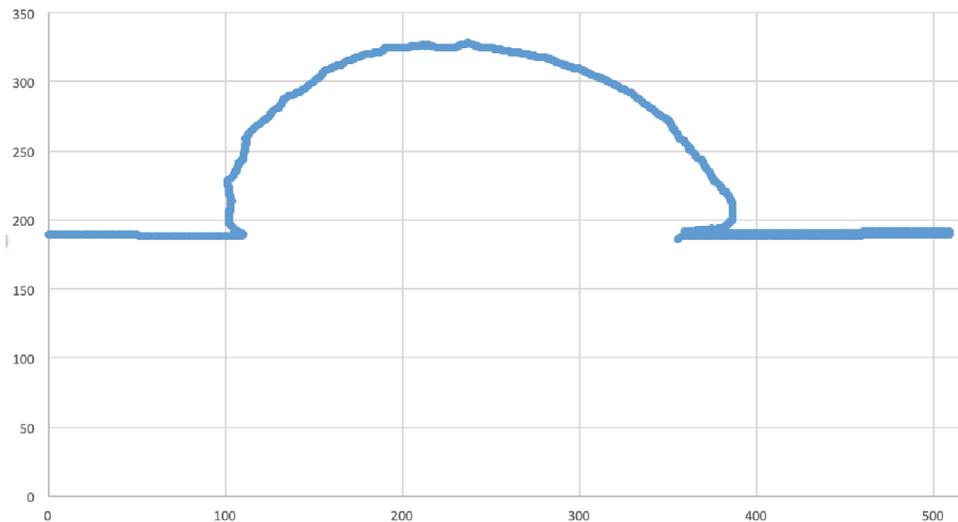


Figura 19 – Gráfico gerado a partir do arquivo CSV dos contornos da imagem 1 In.

4.2.1 Inicialização do programa

Como o Digimetal foi escrito em Python 3 e com o auxílio do pacote OpenCV, o código começa definindo os pacotes que o Python deverá usar durante o processamento:

- openCV: Visão Computacional em código aberto.
- numpy: Computação numérica para python.
- statistics : Funções estatísticas para dados numéricos.
- matplotlib: Gráficos 2D em Python.
- glob: Encontra nomes em listas de arquivos seguindo um padrão.
- pandas: Estruturas de dados de alta performance e fáceis de utilizar, com ferramentas de análise e exportação de dados para Python.
- time: Pacote popular de Python que provê funções para trabalhar e representar a passagem de tempo.

Com as bibliotecas importadas, o programa define as constantes e inicializa as variáveis que serão usadas. Boa parte dessas variáveis são globais e podemos vê-las na Tabela 4.

Antes de começar a usar o Digimetal para avaliar as imagens o usuário deve fazer a configuração inicial do usuário. Esta etapa acontece no prompt de comando, ou terminal do computador, e é o momento em que é inserido o diretório com as imagens a serem avaliadas e com a taxa de conversão de pixel para micrômetros. Após essa segunda informação ser enviado pelo usuário, o Python abrirá uma janela para interagir com ele.

Tipo	Número	Descrição
Parâmetros de ajuste	7	São a entrada do usuário para operar na imagem.
Dados da imagem	4	Identificam a imagem como: índice, nome e posição da linha média.
Lista previsão	7	Armazena valores antigos dos parâmetros e determina os futuros.
Valores iniciais	7	Valor inicial de uma <i>trackbar</i> calculado a partir do item acima.
Dados exportados	8	Saída do processo, as 7 dimensões e o tempo da medição das imagens.
Listas exportação	9	Armazena valores e nomes das imagens para exportação dos dados.
Estados do sistema	4	Diretório das imagens, redução, conversão de pixel e cliques dados.

Tabela 4 – Tabela de variáveis.

4.2.2 Avaliando Imagens

O programa desenvolvido realiza uma pesquisa no diretório informado, buscando por todas as imagens com extensão ".tif", organiza-as por ordem alfabética e seleciona a primeira imagem para ser medida. Neste momento, serão aplicados os filtros e a segmentação com os parâmetros iniciais do programa e o usuário vai fazer o primeiro julgamento de qualidade da medição, com base nas imagens mostradas.

A partir daí não há uma sequência bem definida de ações, pois fica a critério do usuário a forma de tratamento da imagem. Entretanto, aconselha-se começar a avaliação pela reta média. Caso a detecção automática não esteja boa, o usuário pode efetuar dois cliques, gerando dois pontos e definindo a reta média de acordo com a sua visão. Alternativamente, pode-se modificar os quatro valores de processamento de imagens (os dois referentes a iluminação e os dois filtros, o de ruído e o de textura) para que a detecção automática de linha melhore.

Uma vez que o usuário julgue que a filtragem está adequada, ou seja, que as 3 áreas em questão estejam com uma divisão bem definida, ele pode seguir para os outros três parâmetros, referente a fase de segmentação. Cada barra de segmentação pode assumir um valor entre 0 e 255, que corresponde a cor aproximada, em escala de cinza, que a área possui, uma vez que os filtros foram aplicados. Caso isso não seja possível, o usuário pode voltar a mexer nas barras de ajuste dos parâmetros da filtragem, recomeçando o processo.

Uma vez que o usuário decidiu que aquela imagem está com os parâmetros adequados, ou seja, que as áreas que aparecem na imagem estão bem representadas pelos contornos desenhados sobre elas, o usuário confirma apertando a "barra de espaço". Caso ele julgue que não há uma boa maneira de ajustar o programa para que essa imagem possa ser medida, o usuário cancela apertando "ESC".

Dependendo da ação do usuário, o sistema Digimetal armazenará a imagem e os dados de saída de forma diferente. No caso de uma imagem bem mensurada, o sistema salva a imagem com os dados de saída. Caso contrário, salvará a imagem com os dados NaN (do inglês, *Not a Number*), que em programação indica que o dado não representa um número válido.

Quando o usuário tiver passado por todas as imagens do diretório escolhendo entre "barra de espaço" e "ESC", o programa exportará os dados de saída e fechar. Existe ainda uma última possibilidade de interação do usuário, no caso dele desejar interromper as medidas sem perder o trabalho feito até então. Neste caso ele pode apertar "Q" e aguardar a finalização. Neste caso, a imagem atual não será avaliada mas as avaliações de todas as anteriores serão exportadas.

4.2.3 Finalização

Ao fim do programa, todas as informações desejadas devem ser exportadas para que não sejam perdidas. Assim, a lista com nome das imagens, as listas com os dados das medições e o tempo de medição de cada imagem são salvos em uma linha de uma instância do objeto Dataframe, da biblioteca *pandas*. Esse Dataframe será exportado para o formato .xlsx, ou seja, será uma tabela com os dados numéricos de todas as imagens. Para cada imagem que tenha sido corretamente identificada será criada uma outra instância de Dataframe, na qual cada linha será composta por um par de coordenadas (x, y) que descreve o contorno superficial da imagem. O formato de exportação será .csv. Esses arquivos serão criados no mesmo diretório especificado durante configuração inicial. O nome e o local dos arquivos são exibidos no terminal do sistema operacional.

A Figura 20 exibe o fluxo de informação que ocorre em cada iteração do programa, começando com $i = 0$, sendo "i" o índice da imagem. Tendo a lista de todas as imagens disponíveis no diretório, a função envia a primeira imagem para passar pelo processo da função *Main*. Então, a imagem passa pelos Filtros, na seguinte ordem: correção gamma, ajuste de contraste, filtro de ruído e filtro de textura.

Nesse ponto é aplicada a Segmentação, chama-se a função "retângulo alto e baixo" que diferencia a imagem em três, e então aplica-se a função "isola" com a cor da sua respectiva área, "cor dep" para a área de deposição, "Adep", "pen" é para área de penetração e "cor azta" para a área da zona termicamente afetada.

A função "calcula_imagem" efetua a Medição das áreas e dos contornos, para então a etapa da Visualização, em que estes contornos são desenhados na imagem original e esta é unida com a filtrada. Em seguida, ambas são exibidas para avaliação visual do usuário.

O programa fica esperando alguma interação do usuário, podendo esta ser um reajuste, simbolizado pelo símbolo de uma seta curva apontando para o seu início, uma confirmação de que a imagem foi bem avaliada ou que foi mal avaliada, simbolizados pelo mão verde e vermelha, respectivamente, então o contorno será exportado. Assim o programa segue para próxima imagem, isso foi simbolizado pela expressão de programação " $i = i + 1$ ", indicando que o índice é incrementado em uma unidade.

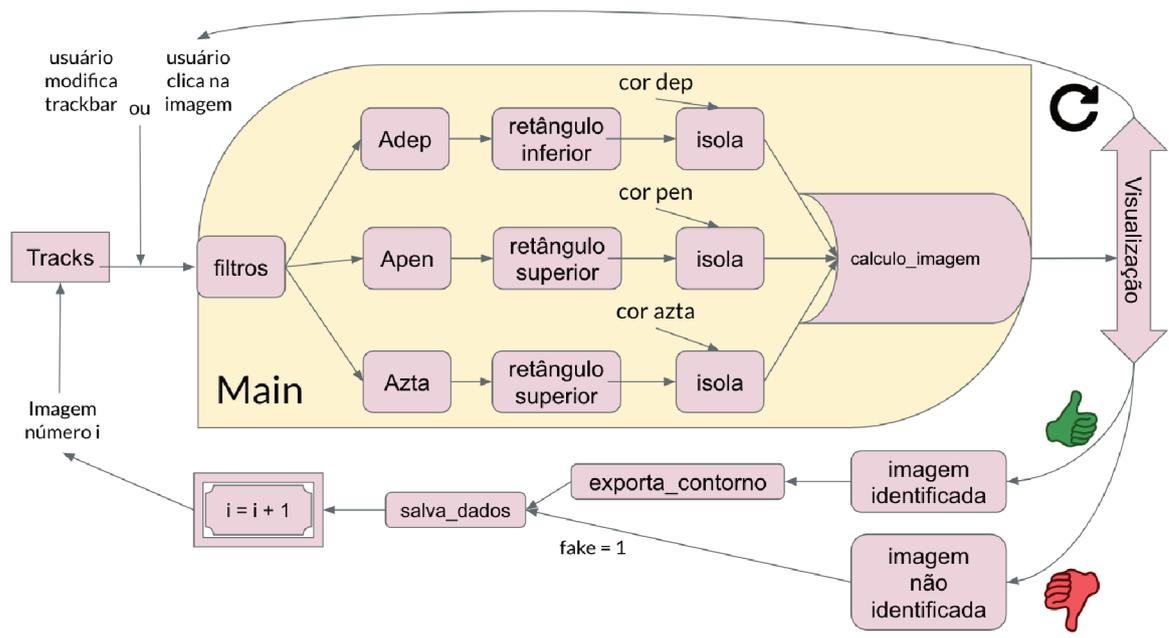


Figura 20 – Diagrama de fluxo de informação.

4.3 Validação do programa

Nesta Seção são mostradas as tabelas e são avaliados os desempenhos dos usuários perante o programa, sua curva de aprendizado e a fidedignidade dos contornos obtidos. Essas são as análises que puderam ser deduzidas através dos resultados dos testes. Elas dizem respeito aos métodos implementados, às imagens, aos materiais, aos usuários e ao programa usado como referência, o ImageJ.

4.3.1 Medição de largura do cordão

A medida linear mais importante da medição é a largura do cordão de solda. A sua medida no ImageJ é aproximada, pois nem sempre está claro o momento em que a linha de fato começa, já no Digimetal há duas medidas que tentam descrever essa dimensão: a largura de 5% e a largura máxima. A largura de 5% mede a distância no eixo "x", o ponto de medida é localizado acima da linha média, a uma distância da linha média é equivalente a 5% da altura do cordão. Como o eixo "x" não está perfeitamente alinhado com a linha média, pode haver alguma distorção nessa medida, porém não foi considerado que seja um valor significativo, visto a inclinação ser sempre pequena. A sugestão de evitar essa inclinação foi feita para os colaboradores do laboratório.

A largura máxima é a largura do retângulo que abrange o contorno do cordão. Nem sempre a largura máxima corresponde a largura do cordão, pois os filtros e as cores originais da imagem podem criar uma área próxima a linha média com a cor muito próxima da deposição do material.

Como ambas medidas apresentam possíveis falhas, foram implementadas ambas técnicas, a fim de descobrir qual apresentaria menos erros e numa proporção menor. Na Tabela 5 podemos ver o erro médio para todos os usuários e imagens. Já na Tabela 6 é possível observar que, para quase todos os usuários, a melhor medida foi a largura de 5%, exceto para o usuário 2 (onde foi observada a menor diferença entre as medidas), para o qual a medida mais precisa foi a largura máxima.

Medida	Erro médio
Clad width at surface	9,42%
Clad Height	7,95%
Melt Pool Depth	24,02%
Clad Area	8,85%
Apparent HAZ	28,66%
Apparent Dillution Area	20,71%
Média	16,42%

Tabela 5 – Diferença entre os erros de medidas lineares e de medidas de áreas.

Usuário	Largura de 5%	Largura máxima
U1	7,15%	22,18%
U2	10,64%	8,48%
U3	10,47%	15,30%
média	9,42%	15,32%

Tabela 6 – Tabela comparativa de erros relativos por usuários para medidas de largura do cordão, usando a média das imagens.

Por fim a Tabela 7 mostra o erro médio por imagem para todos os usuário, mostrando que na maior parte das imagens a diferença a favor da largura de 5% é grande, enquanto as que favorecem a largura máxima são um pouco mais precisas. Desta forma, para as próximas análises quando se falar de erro médio, não estará incluído a medida de largura máxima.

4.3.2 Diferença entre os dados

A saída do Digimetal é composta por 8 dados, os sete mostrados na Tabela 5 e o tempo que levou para medir a imagem. Porém podemos classificar esses sete dados a partir de dois critérios:

Tipo de medida:

- Linha: são as medidas de altura e largura, que por possuírem somente uma dimensão apresentam menos chance de erro de medição. Erro médio: 13.80%.
- Área: são as medidas principais, as três áreas apresentam, em média, erros maiores, o que é esperado. Erro médio: 19.40%.

Filename	Largura de 5%	Largura máxima
1 Fe - 2018.07.04 - A26	25,92%	41,13%
2 Fe - 2018.07.04 - A35	6,51%	20,47%
3 Fe - 2018.07.04 - A41	4,10%	10,67%
4 Fe - 2018.07.04 - A48	7,37%	14,18%
1 Inconel 625 - 2018.02.27 - A5	1,86%	1,29%
2 Inconel 625 - 2018.02.27 - A9	5,15%	35,25%
3 Inconel 625 - 2018.02.27 - A27	9,90%	8,91%
4 Inconel 625 - 2018.02.27 - A46	17,20%	17,20%
1 NiCrBSi - 2018.05.10 - A5	NaN	NaN
2 NiCrBSi - 2018.05.10 - A9	11,04%	7,73%
3 NiCrBSi - 2018.05.10 - A29	7,10%	4,54%
4 NiCrBSi - 2018.05.10 - A35	7,46%	7,15%

Tabela 7 – Tabela comparativa de erro relativo entre as medidas de cordão por imagem, usando a média dos usuários.

Tipo de área:

- Cordão: Esses dados são os únicos que estão na parte superior da imagem, acima da linha média. Dessa forma é mais fácil de detectar o cordão na imagem e seus erros tendem a ser menores. Erro médio (largura de 5%, altura e área): 10.38%.
- Penetração: É a área desse contorno e a sua altura, está na parte debaixo da imagem e pode ser confundido com a outra área que está nessa parte da imagem. Erro médio (altura e área): 22.36%.
- Zona Térmicamente Afetada: Essa área não é bem definida, sendo necessária a interpretação da imagem por parte do usuário pois é uma área que é afetada pela temperatura do processo a ponto de modificar a estrutura do material. Somente a área dessa área é relevante e seu erro é o maior, como já esperado. Erro médio (área): 28,66%.

A análise dos erros lineares demonstra que há uma aproximação boa dos valores medidos pelo software ImageJ, assim como a área de cordão mas diferentemente das outras áreas, isso indica que a precisão linear e de área podem possuir erros semelhantes e que as medidas da parte de baixo da imagem possuem menor precisão.

4.3.3 Diferença entre materiais

Nos testes foram utilizados três grupos de quatro imagens, sendo que cada grupo referia-se a imagens de um material diferente. Esses materiais foram escolhidos por serem os mais utilizados para testes ou aplicações dos colaboradores do laboratório.

O ferro é um material muito barato então costuma ser utilizado quando se deseja entender o processo a laser, mas também pode ser utilizado para estudar o material, que possui diversas aplicações. Este material, porém, foi aplicado sobre ferro, logo a sua diluição acontece facilmente, o que causa uma mistura onde é difícil de distinguir visualmente a área de Penetração (*Apparent Dillution Area*) e a Área da Zona Termicamente Afetada (*Apparent Heated Affected Zone*). Os erros desse material são os maiores.

O Inconel 625 é um material largamente utilizado em todo o mundo, especialmente no revestimento de materiais. Essa superliga de Níquel possui ótimas propriedades de resistência a corrosão e por isso seu preço é elevado. O Inconel apresenta as melhores imagens para detecção, visto que possui pouca textura, e o material usado para substrato fica muito claro quando afetado pela temperatura do processo, contrastando bem com a cor da liga, que é muito escura.

Já a liga de NiCrBSi é um material que também é considerado uma superliga, com propriedades de proteção à abrasão nos materiais que reveste e, assim como o Inconel 625, possui um valor alto. Esse material apresenta um bom contraste nas áreas abaixo da linha média, porém a textura e cor são muito semelhantes ao material de revestimento, a baquelite preta (um polímero de condensação), e a baixa qualidade das imagens de metalografia aumentaram a magnitude do erro. Esse problema poderia ser facilmente resolvido caso a cor da baquelite fosse vermelha.

Esses erros foram resumidos na Tabela 8, demonstrando que o Ferro é um material que o Digimetal tem grande dificuldade de medir. Entretanto tanto para Inconel como para o NiCrBSi, o Digimetal consegue efetuar a medição com a qualidade requisitada pelo cliente.

Material	Erro médio
Ferro	24,13%
Inconel 625	10,48%
NiCrBSi	15,30%

Tabela 8 – Tabela comparativa de erros relativos por material, média de todos os usuários e imagens.

4.3.4 Imagens bem avaliadas

Nesta seção será avaliado o desempenho no software na questão de quantas áreas e imagens ele conseguiu fazer a medição de forma adequada. Os critérios estabelecidos para caracterizar uma medição adequada foram os mesmos dos requisitos do sistema: para a área de Clad é aceito um erro inferior a 15%, para a Zona Termicamente Afetada (HAZ) um erro inferior a 30% e para a área de Penetração o erro máximo é 25%. Com esses parâmetros o podemos observar na Tabela 9 que, das 34 áreas (seriam 36, porém as peças

"1 Fe" e "1 In" não possuem área de Penetração) que foram testadas, obtivemos sucesso em 24 e falha em 10. Um índice de acerto de 70.58%.

File	Clad Area	HAZ	Penetration
1 Fe	10,51%	36,12%	0%
2 Fe	10,89%	NaN	NaN
3 Fe	7,40%	53,84%	NaN
4 Fe	6,92%	75,02%	4,13%
1 In	4,16%	16,97%	7,17%
2 In	7,96%	17,81%	0%
3 In	11,71%	10,48%	10,31%
4 In	14,17%	32,82%	16,18%
1 NiCrBSi	NaN	23,03%	4,27%
2 NiCrBSi	13,31%	15,00%	16,72%
3 NiCrBSi	3,99%	9,64%	66,95%
4 NiCrBSi	6,36%	33,15%	39,99%

Tabela 9 – Em verde as medições que foram bem realizadas seguindo os critérios estabelecidos.

Analisando as imagens e considerando que foram medidas corretamente somente as que se encaixaram nos três requisitos ao mesmo tempo. Dessa forma os resultados são menos satisfatórios e somente um terço das imagens foram identificadas pela maior parte dos usuários, como pode ser visto na Tabela 10. Porém, o número de imagens inteiramente bem avaliadas é menor, para o usuário 1 é 6, para o 2 é 5 e para o usuário 3 é somente 4.

Usuário	Clad Area	Apparent HAZ	Penetration	Totalmente
U1	9/12	8/12	8/12	5/12
U2	7/12	9/12	8/12	4/12
U3	10/12	7/12	8/12	4/12
Média	11/12	6/12	8/12	4/12

Tabela 10 – Tabela com quantidades acertadas por cada usuário.

Podemos ver na Figura 21 algumas das imagens testadas. A primeira imagem 21.A é a que obteve o menor erro para todos os usuário, sendo considerada a imagem mais fácil de medir com o Digimetal. Já as outras imagens são as que tiveram problemas na identificação de algumas das áreas. Na imagem 21.B vemos como o ferro pode se dissolver, de forma que a cor da área de penetração fique muito próxima a Zona Termicamente Afetada. A figura 21.C é um exemplo de uma imagem com boa nitidez nas áreas, porém apresenta um enquadramento que dificulta a medição, pois os filtros podem reduzir significativamente algumas áreas e aumentar outras, causando maior erro quando comparada com uma imagem em que as áreas de interesse ocupem uma maior parte da imagem. O fato de haver um outro cordão na parte de cima dessa imagem não atrapalhou a medição com nenhum usuário. Já na figura 21.D a área de Clad, após a filtragem, não apresenta definição de

borda suficiente para que seja apropriadamente medida. Isso se deve à fraca iluminação da peça e ao uso de baquelite preto como material de revestimento plástico. As imagens da Figura 21 representam a área de operação pois mostram onde a medição funciona e onde não, principalmente as limitações do sistema proposto neste trabalho em que a segmentação é feita por cor em escala de cinza.

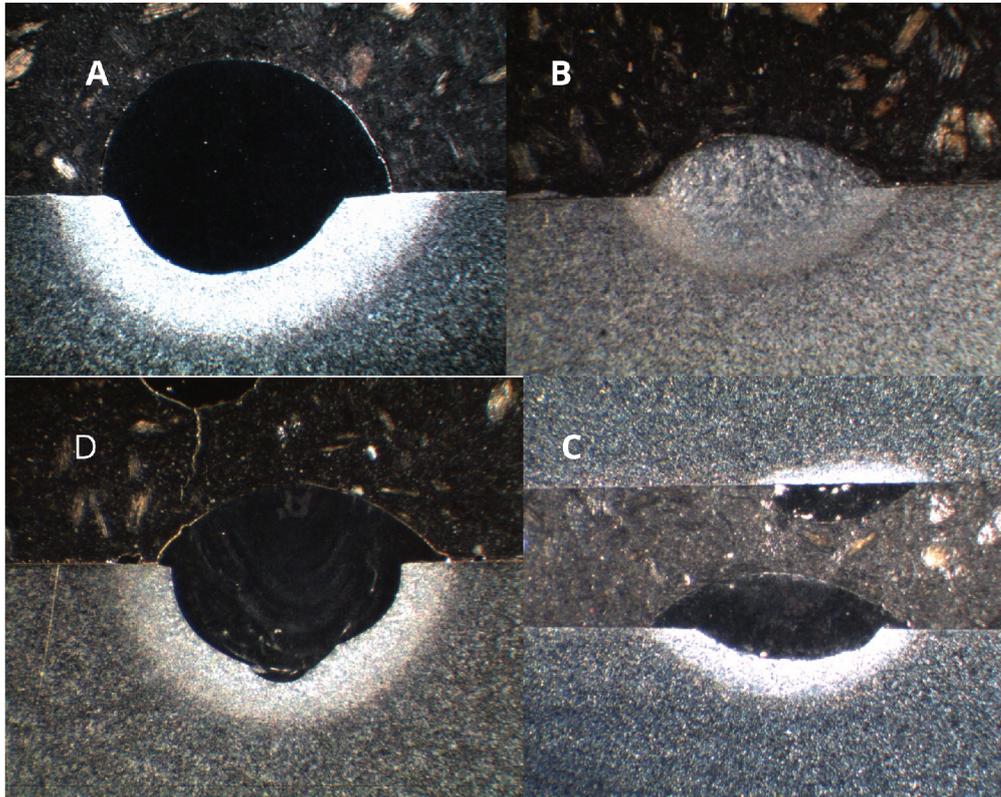


Figura 21 – A) 1 In; B) 2 Fe; C) 1 NiCrBSi; D) 4 NiCrBSi. A imagem "A" serve de imagem modelo enquanto "B", "C" e "D" são exemplos de imagens problemáticas para o Digimetal.

4.3.5 Áreas de Interesse

Cada área que desejamos medir possui sua própria característica e, da mesma forma, possui um erro atrelado, com os resultados obtidos pelas áreas desses contornos podemos entender melhor essas diferenças.

Olhando a tabela comparativa do erro entre as áreas, a Tabela 11, pode-se observar que a Zona Termicamente Afetada é mais difícil de medir do que as outras duas áreas, pois não é uma região bem definida, gerando dúvida até para visão humana onde deve ser considerado como termicamente afetada. Tendo em vista que há o erro na parte superior da imagem por alguma diluição da área de penetração que se confunde com esta área, e na parte inferior, por nem sempre ter o degrade de cor bem definido.

A área de Clad, por ser a única na parte superior da imagem, só precisa se diferenciar da cor da baquelite de revestimento, o que normalmente é possível. Esta área é a de maior importância para os testes pois é a que melhor descreve o resultado obtido numericamente.

Já a área de Penetração possui a facilidade de ter a cor semelhante a área de Clad, porém divide o espaço inferior da imagem com a Zona Termicamente Afetada, condição que pode causar interferência entre as suas cores na região limítrofe das duas áreas como consequência do uso de filtros de Ruídos e Textura. Como essa área possui uma medição mais confiável, sua área é removida na análise do contorno da Zona Termicamente Afetada, melhorando significativamente a sua detecção.

Usuário	Clad Area	Apparent HAZ	Apparent Dillution Area
U1	7,75%	28,11%	22,39%
U2	11,60%	31,54%	23,53%
U3	7,20%	26,34%	16,22%
média	8,85%	28,66%	20,71%

Tabela 11 – Tabela comparativa entre os erros das áreas por usuário, média de todas as imagens.

4.3.6 Tempo: comparação entre softwares

A Figura 22 é um gráfico de pontos que relaciona a média de tempo dos três usuários nos dois programas, sendo que em vermelho estão os pontos referentes ao tempo médio para medição de cada imagem no ImageJ e, em azul, os pontos das mesmas imagens, mas no Digimetal. Percebe-se que uma imagem que seja mais demorada para medir em um programa não necessariamente significa que demorará no outro, o que é esperado, é esperado uma vez que os métodos de medição adotados são diferentes. Também podemos observar na Figura 22 que a maior parte dos pontos referentes ao Digimetal se encontram entre 50 e 100 segundos, enquanto os pontos do ImageJ residem na faixa de 100 a 200 segundos. Visualmente pode-se perceber que a linha azul é inferior em todos os pontos exceto em um caso (onde um usuário insistiu em tentar achar uma área que estava difícil por mais tempo, os outros usuários também falharam nessa tarefa, porém desistiram mais rapidamente).

O tempo médio de uma imagem no ImageJ é de 147.64 segundos e o do Digimetal 88.09s, seria tentador afirmar que o método aqui proposto é, aproximadamente, um minuto mais rápido que o ImageJ, mas levando em consideração que o Digimetal efetuou 30 das 34 (88.24%) medidas de área, o tempo do ImageJ poderia ser corrigido para 130.27 segundos, logo a diferença média entre os métodos de medição seria de 42.18 segundos.

Uma possível justificativa para explicar essa grande diferença entre os programas pode estar no fato de o ImageJ não exportar os dados e depender do usuário para fazer

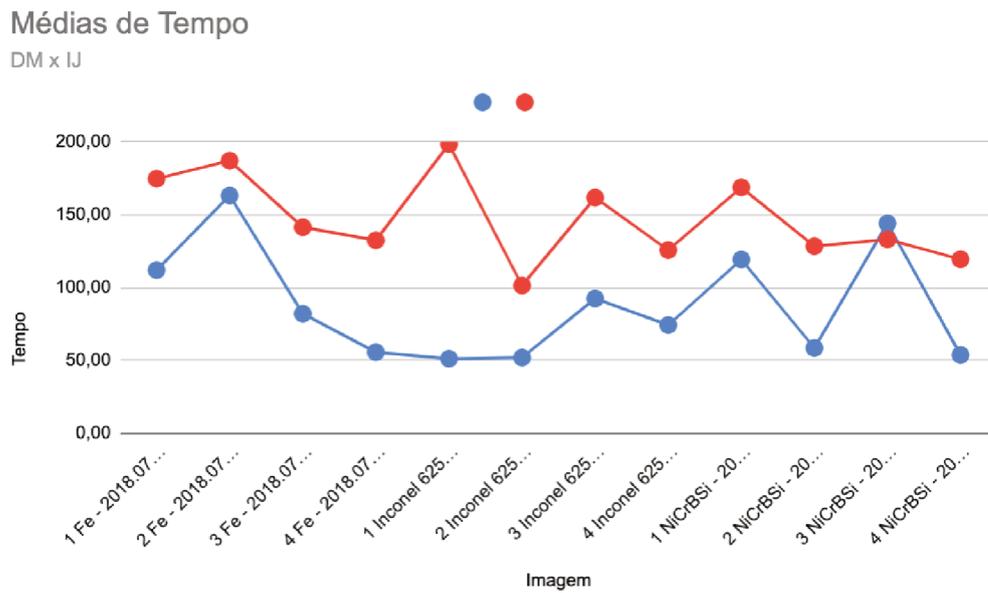


Figura 22 – Diferença entre as médias de tempo de um programa e outro.

isso manualmente, esse tempo foi considerado nos testes. Porém outra questão a se levar em consideração é a rotina de abrir a imagem após a medição da anterior que é automática no Digimetal mas não no ImageJ, esse tempo não foi levado em consideração nos testes.

4.3.7 Tempo: comparação entre usuários

Para o fim deste capítulo deseja-se estudar a diferença que tivemos para cada usuário no quesito de tempo. A sequência das imagens corresponde à sequência em que as imagens foram avaliadas, possibilitando analisar se houve alguma familiarização com o uso do software e também entender como a diferença de experiência prévia e conhecimento sobre os softwares podem influenciar no desempenho do usuário.

Na Figura 23 pode-se observar o tempo medido para a análise de cada imagem pelo usuário 1 nos pontos em azul, o usuário 2 com pontos em vermelho e o usuário 3 com os pontos verdes. Os usuários 1 e 2 tiveram um ponto onde teve maior demora enquanto o usuário 3 teve dois, o que é de se esperar, visto que o terceiro tinha menos experiência. Porém para a maioria das imagens o tempo que eles precisaram para efetuar a medição foi aproximadamente o mesmo. Percebe-se alguma evolução para os três usuários, de forma mais clara para o usuário 1 e 3, visto que as primeiras medições são em torno de 30% mais lentas.

Analogamente, a Figura 24 mostra os pontos (de mesmas cores e usuários) dos tempo de cada usuário no software ImageJ. A curva de aprendizado fica clara somente para o usuário 3 que era pouco familiarizado, o que ocasionou grande demora nas primeiras medidas, se aproximando (mas nunca superando) os outros usuários no final da seção de

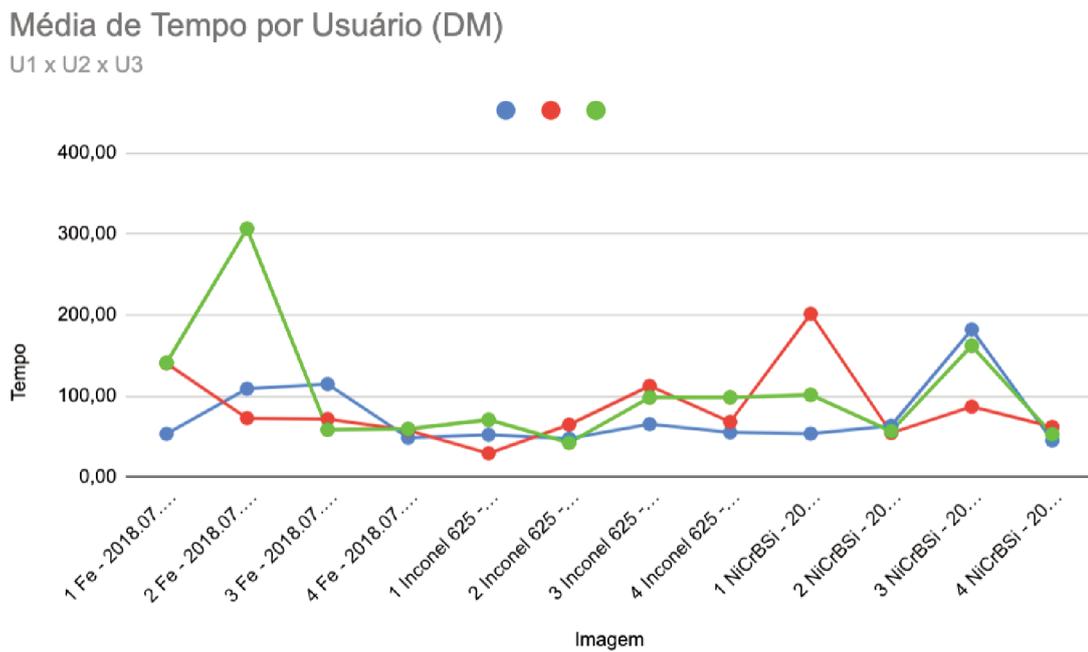


Figura 23 – Tempo de cada usuário, por imagem no software Digimetal.

testes.

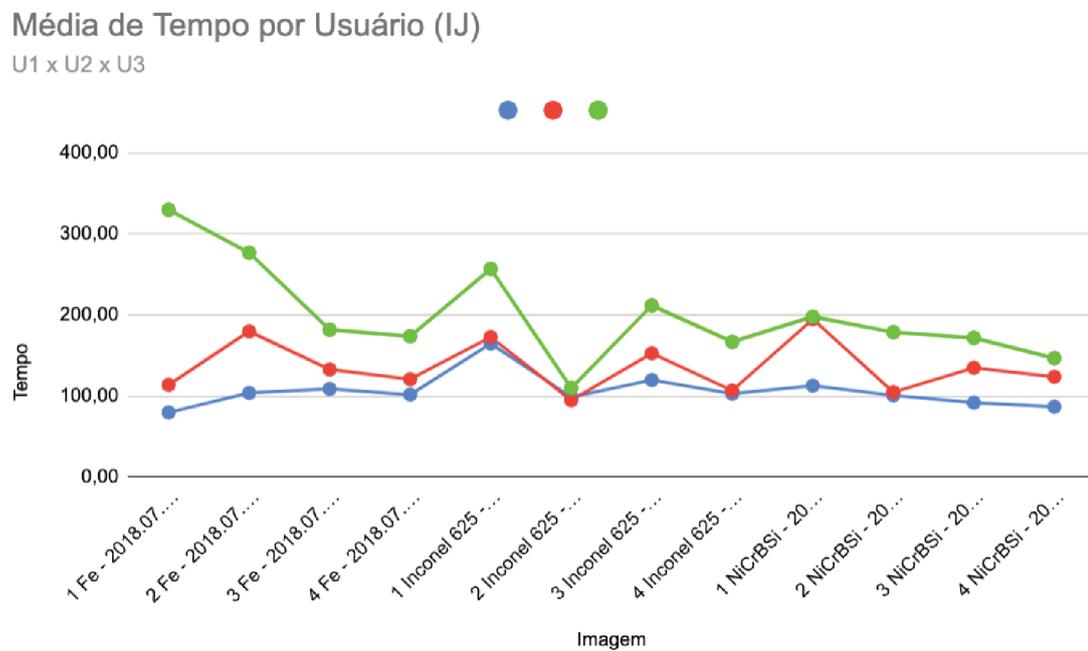


Figura 24 – Tempo de cada usuário, por imagem no software ImageJ.

5 Conclusão e Perspectivas

O trabalho aqui apresentado teve início em janeiro de 2019 com a introdução de visão computacional e o desafio de implementar uma solução nesse sentido para melhorar as técnicas de medidas do Laboratório de Mecânica e Precisão. O objetivo dos colaboradores do laboratório era poder substituir o programa atual por um que oferecesse melhor qualidade de trabalho e resultados mais confiáveis, para que o trabalho deles pudesse ser potencializado. Ao longo do ano foram inúmeras reuniões e acompanhamentos sobre os avanços no projeto, sempre recebido com muito entusiasmo por parte da equipe do LMP. Chega-se ao fim da segunda etapa desse processo com um programa concluído e com o desafio vencido.

O problema proposto por Adriano Pereira foi o de criar uma solução com Visão Computacional para realizar medições em imagens de metalografia do laboratório, uma vez que o programa usado causava insatisfação. O desenvolvimento durou dois semestres e concluiu a primeira etapa no fim do primeiro semestre de trabalho. Até então possuindo as características das imagens descritas e o conhecimento de filtros e efeitos de imagem para começar um programa com filtragem e visualização do resultado. Mas as etapas de segmentação, medição e exportação ainda não estavam resolvidas.

O processo foi caracterizado pelo desenvolvimento de conhecimentos e habilidades valiosos para qualquer engenheiro em formação. O LMP possui vasto material sobre os seus processos e uma equipe disposta em ajudar e a compartilhar experiências. Assim, foi possível ter acesso a informações específicas do processo de Deposição Metálica a Laser necessárias para a condução do trabalho.

Conduzir as evoluções de um projeto tão complexo foi um exercício de abstrair e encapsular processos e organização lógica. Assim foram criadas novas funções, executados muitos testes e, com base em outras experiências, os protótipos foram se superando até que a medição estivesse acessível e precisa.

Qualquer sistema de visão computacional começa pela aquisição da imagem e problemas nessa etapa ocasionarão dificuldades para o resto do sistema, como explica Davies [3]:

"Ao implementar uma sistema de visão nada é mais importante do que a aquisição de imagem. Qualquer deficiência nas imagens originais pode causar grandes problemas na análise e interpretação de imagem. Um exemplo óbvio é o da falta de detalhe devido ao insuficiente contraste ou ao foco ruim da câmera: isso pode gerar, no melhor dos casos, que as dimensões dos objetos

não sejam precisamente mensuráveis a partir das imagens, e no pior dos casos o objeto sequer será reconhecível, então o propósito da visão não pode ser comprida."

Todavia, no projeto aqui apresentado o autor arcou com a responsabilidade de lidar com as imagens já adquiridas. Essa característica se demonstrou ser o maior obstáculo desse método, então uma melhoria na preparação das imagens é mister, pois os seguintes problemas foram observados:

- as cores do material de revestimento se assemelhavam com a cor do material de adição;
- o enquadramento da imagem não garantia que a maior parte da imagem estivesse mostrando as áreas de interesse e não o revestimento ou substrato;
- a iluminação não incidia igualmente nos dois lados da peça, tanto em angulação quanto intensidade;
- baixa iluminação.

Esses problemas podem ser mitigados com uma adequada iluminação pois quanto maior a iluminação (desde que não ocasione saturação de branco em alguma área de interesse), maior a quantidade de informações da imagem e melhor a definição das bordas. Da mesma forma, o bom enquadramento sugere erros proporcionalmente menores e o uso de um material plástico de revestimento de outra cor facilita a identificação do cordão.

Apesar disso, o resultado esperado foi obtido, pois a grande maioria das áreas pode ser detectada com a acurácia requerida, justificando a qualidade das medidas que são exportadas ao fim da medição. Os contornos superficiais do cordão retratam as respectivas geometrias com fidelidade, e como foi requisitado, exportados em formato .CSV. O aprendizado foi progressivo com o uso, como se espera de um programa intuitivo. A experiência de utilizar o Digimetal se mostrou mais célere e prazerosa que o antigo sistema. Porém, o programa ficou mais sensível ao usuário do que o ImageJ, ou seja, as medidas entre os usuários apresentaram menores variações no ImageJ do que as medidas no Digimetal, porém não haviam dados estatísticos fortes, já que foram três usuários rodando o teste uma vez em cada programa.

O material Inconel 625 atingiu os melhores resultados e o Ferro os piores, já a liga de NiCrBSi apresentou resultados intermediários. A partir dos testes realizados foi possível entender aonde estão as limitações do produto e aonde este pode melhorar para obter resultados semelhantes em outros materiais.

A maior vantagem do novo método é a gratificação gerada ao executar a medição. No Digimetal o usuário se sente entretido em ajustar iluminação, filtros e procurar cores,

principalmente por saber que a parte repetitiva do trabalho está automatizada, que são as partes de digitar manualmente os dados medidos em uma tabela e de fechar a imagem atual e abrir a seguinte. Um erro em qualquer uma dessas partes poderia gerar graves erros ou, em caso de perceber o erro, ocasionar retrabalho por parte do colaborador responsável pela medição. O Digimetal é um programa que oferece a tranquilidade para o usuário de que o seu esforço não será desperdiçado.

Secundariamente, também se pode entender o ganho no tempo das medições como uma vantagem, uma vez que disponibiliza a mão-de-obra que antes ficava ocupada por mais tempo, podendo causar estresse do colaborador e prejudicar outra atividade do laboratório, acadêmica ou pessoal. Isso pode ser percebido pelo entusiasmo que os usuários participantes do teste demonstraram.

Além dessas contribuições diretas ao ambiente de trabalho, a área de Visão Computacional agrega imenso valor em qualquer ambiente pois é uma área do conhecimento fortemente relacionada com o Estado da Arte no tangente aos processos de manufatura aditiva como o LMD usado no no laboratório, que agora possui uma aplicação para auxiliar nos estudos e também para servir de objeto de estudo. Já existem outros projetos envolvendo este tópico que poderá se beneficiar do estudo aqui apresentado e da documentação do código fonte.

5.1 Sugestões

Por fim, iremos abordar quais opções de desenvolvimento foram deixadas de lado por não estar compreendido no objetivo atual do projeto, mas podem aumentar substancialmente a qualidade do produto. As sugestões aqui variam desde adaptações já planejadas no início do processo, como a implementação de um método de medida semelhante ao do ImageJ até ideias que surgiram na revisão bibliográfica ou então a partir de conversas empolgadas com colegas de trabalho.

5.1.1 Seleção de área e preenchimento pelo método de pontos

A ferramenta que era utilizada no LMP para medição de imagens de metalografia era o programa ImageJ. O método utilizado era a criação de um polígono a partir de pontos informados pelo usuário com cliques do cursor. Essa abordagem pode ser facilmente adicionada no sistema do Digimetal. Sugere-se aqui adicionar uma funcionalidade que permita ao usuário alternar o método de medição quando alguma área não estiver sendo corretamente mensurada pelo método atual do Digimetal.

Desta forma, a segmentação por cores funcionará como método padrão, contando com a precisão e celeridade atuais, porém ao se deparar com uma área em que não há contraste ou diferença de cor suficiente para ter uma boa medição, o usuário poderá

digitar uma letra que represente uma área e, então, adicionar ponto com cliques que serão representados por letras L e, quando o usuário clicar novamente no ponto inicial, será criada uma área e ela será medida e representada como se fosse uma medição normal pelo programa. Assim, acredita-se que poderá ser atingido um compromisso entre o tempo, a praticidade da medida por segmentação de cor e a precisão do clique e do olhar humano, sem ter que recorrer ao trabalho manual de abertura de arquivos e digitação manual de dados.

5.1.2 Segmentação por textura

Como as imagens de metalografia são capazes de mostrar a textura criada pelo contorno dos grãos e suas cores, e que um material deve ter uma textura ou cor diferente de outro, a função *isola* poderia ter sido programada para segmentar por textura.

Essa abordagem pode ser feita de forma a substituir a função *isola* atual ou somente a fim de melhorar. De acordo com a metodologia usada, seriam testadas, em diferentes imagens, as três formas de segmentar as áreas e, partir de uma análise visual, poderá ser decidido qual o melhor método. No caso essa análise fosse insuficiente, seriam feitas comparações com as medições manuais do ImageJ.

5.1.3 Redes Neurais

Modeladas com base no entendimento do funcionamento de um neurônio humano, as redes neurais são um sistema de aprendizado de máquina. Este tipo de sistema é adequado para a identificação de padrões em análise de imagens e já foi utilizado no LMP para a previsão de medidas de cordão de deposição metálica. Essa tecnologia já foi estudada, conforme apresentado em [Gonçalves \[4\]](#), e o desenvolvimento deste trabalho ocorreu no início do desenvolvimento do projeto Digimetal, influenciando fortemente o autor.

[Davies \[3\]](#) afirma que um simples Perceptron (entidade básica de uma Rede Neural) é capaz de "identificar objetos em transportadoras e fazendo julgamento de sua qualidade, lendo etiquetas e códigos, verificando assinaturas, conferindo impressões digitais e assim por diante". Esse processo é possível através do processo de pegar as entradas do sistema (características da imagem, no nosso exemplo), multiplicar por um peso, somar e então analisar se a saída de uma determinada decisão deve ser ativada ou desativada, através de um valor limiar. O modelo do Perceptron básico pode ser visto na [Figura 25](#). Este modelo de aprendizagem requer treinamento.

Porém, um simples Perceptron não caracteriza uma rede neural, para isso é necessária a composição de camadas de Perceptrons, onde as saídas de uma camada de Perceptron é a entrada da outra camada. Essa arquitetura pode ser vista na [Figura 26](#).

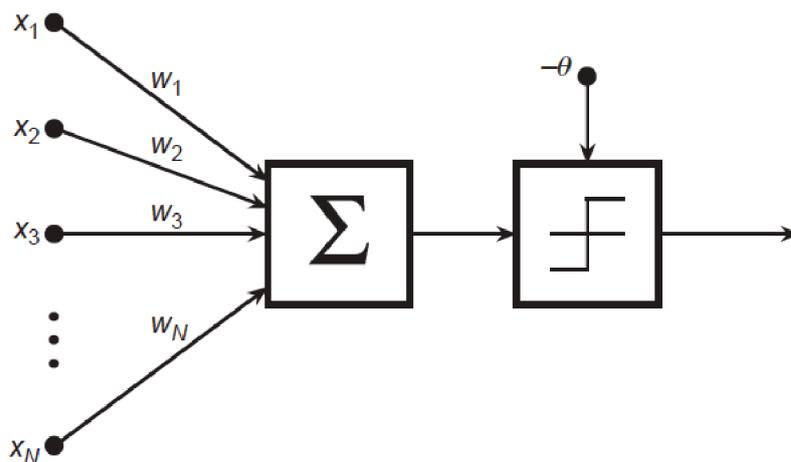


Figura 25 – Forma básica do perceptron: valores de entrada são multiplicados por pesos e então somados, se o resultado passar pelo limiar a saída é positiva, se não passar é negativa.

Fonte: [3]

Assim, pode-se considerar um sistema que a emula o comportamento do cérebro humano.

Deve-se tomar um cuidado especial com o fenômeno chamado de *overfitting*, que é o sobre-ajuste de um sistema para determinada imagem ou tipo de imagem. Para isso, deve-se treinar o sistema com imagem de materiais diferentes.

5.1.4 Aplicação na Web

Soluções para esse tipo de problema podem ser utilizadas em diversos laboratórios que trabalham com manufatura aditiva de cordões metálicos, para o estudo das estruturas formadas. Pensando em expandir a influência no meio científico, compartilhar conhecimento e aplicar essa tecnologia para inovação, o LMP tem planos para colocar esse programa a disposição para uso na web.

Caso seja implementada, essa sugestão permitirá que o usuário que acesse o sítio eletrônico do laboratório possa enviar as suas imagens de metalografia e ajustar as barras dos parâmetros no seu navegador para executar as medições. Uma vez que ele tenha finalizado o seu trabalho, os arquivos de saída são enviados para o usuário por e-mail ou através de download direto, tanto os contornos em .CSV quanto os dados das dimensões medidas em tabela .xlsx.

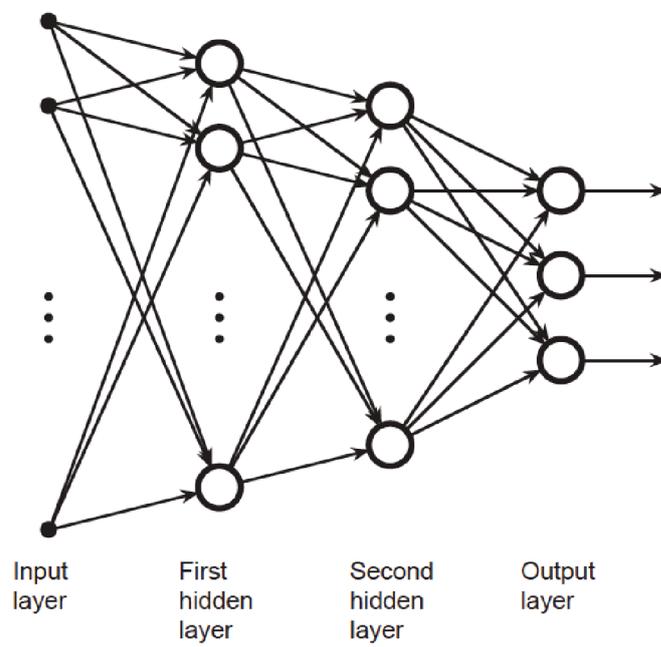


Figura 26 – Perceptron de múltiplas camadas.

Fonte: [3]

Referências

- 1 LUTZ, O. *How Scientists Captured the First Image of a Black Hole*. 2019. Access date:. Disponível em: <<https://www.jpl.nasa.gov/edu/news/2019/4/19/how-scientists-captured-the-first-image-of-a-black-hole/>>. Citado na página 19.
- 2 FORSYTH, D. A.; PONCE, J. *Computer vision: a modern approach*. [S.l.]: Prentice Hall Professional Technical Reference, 2002. Citado na página 19.
- 3 DAVIES, E. R. *Computer vision: principles, algorithms, applications, learning*. [S.l.]: Academic Press, 2017. Citado 7 vezes nas páginas 22, 28, 29, 61, 64, 65 e 66.
- 4 GONÇALVES, D. A. A convolutional neural network approach on bead geometry estimation for a laser cladding system. Florianópolis, SC., 2019. Citado 3 vezes nas páginas 25, 29 e 64.
- 5 OpenCV team. *OpenCV tutorials*. 2019. Access date: 1 jun. 2019. Disponível em: <<https://opencv.org/>>. Citado 4 vezes nas páginas 27, 35, 37 e 38.
- 6 KLETTE, R. *Concise computer vision*. [S.l.]: Springer, 2014. Citado na página 28.
- 7 HU, D.; KOVACEVIC, R. Modelling and measuring the thermal behaviour of the molten pool in closed-loop controlled laser-based additive manufacturing. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, Sage Publications Sage UK: London, England, v. 217, n. 4, p. 441–452, 2003. Citado na página 30.
- 8 HOFMAN, J.; LANGE, D. D.; MEIJER, J. Camera based feedback control of the laser cladding process. In: LIA. *International Congress on Applications of Lasers & Electro-Optics*. [S.l.], 2006. v. 2006, n. 1, p. 1404. Citado na página 30.
- 9 TOYSERKANI, E.; KHAJEPOUR, A. A mechatronics approach to laser powder deposition process. *Mechatronics*, Elsevier, v. 16, n. 10, p. 631–641, 2006. Citado na página 30.
- 10 IRAVANI-TABRIZIPOUR, M.; TOYSERKANI, E. An image-based feature tracking algorithm for real-time measurement of clad height. *Machine Vision and Applications*, Springer, v. 18, n. 6, p. 343–354, 2007. Citado na página 30.
- 11 IRAVANI-TABRIZIPOUR, M.; ASSELIN, M.; TOYSERKANI, E. Development of an image-based feature tracking algorithm for real-time clad height detection. *IFAC Proceedings Volumes*, Elsevier, v. 39, n. 16, p. 914–920, 2006. Citado na página 30.
- 12 BARUA, S.; SPARKS, T.; LIOU, F. Development of low-cost imaging system for laser metal deposition processes. *Rapid Prototyping Journal*, Emerald Group Publishing Limited, v. 17, n. 3, p. 203–210, 2011. Citado na página 30.
- 13 WAZLAWICK, R. *Engenharia de software: conceitos e práticas*. [S.l.]: Elsevier Editora Ltda., 2019. Citado na página 33.

-
- 14 SINGH, A. K. *Power Law (Gamma) Transformations*. 2019. Access date: NOVEMBER 11, 2019. Disponível em: <<https://theailearner.com/2019/01/26/power-law-gamma-transformations/>>. Citado 2 vezes nas páginas 35 e 36.
- 15 KUMADA seiya. *Mean Shift Filtering Practice by OpenCV*. 2013. Access date: NOVEMBER 11, 2019. Disponível em: <<http://seiya-kumada.blogspot.com/2013/05/mean-shift-filtering-practice-by-opencv.html>>. Citado 2 vezes nas páginas 36 e 37.