

UNIVERSIDADE FEDERAL DE SÃO CARLOS – UFSCar
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA – CCET
DEPARTAMENTO DE ENGENHARIA MECÂNICA – DEMec
CURSO DE GRADUAÇÃO EM ENGENHARIA MECÂNICA

**Deteção de Defeitos em Impressões 3D com Visão Computacional e
Aprendizado Não Supervisionado**

Aluno:

Marcelo Batalha Hattori

Orientador:

Prof. Dr. Sidney Bruce Shiki



Fevereiro/2025



FUNDAÇÃO UNIVERSIDADE FEDERAL DE SÃO CARLOS

COORDENAÇÃO DO CURSO DE ENGENHARIA MECÂNICA - CCEMec/CCET

Rod. Washington Luís km 235 - SP-310, s/n - Bairro Monjolinho, São Carlos/SP, CEP 13565-905

Telefone: (16) 33519703 - <http://www.ufscar.br>

DP-TCC-FA nº 15/2025/CCEMec/CCET

Graduação: Defesa Pública de Trabalho de Conclusão de Curso

Folha Aprovação (GDP-TCC-FA)

FOLHA DE APROVAÇÃO

MARCELO BATALHA HATTORI

DETECÇÃO DE DEFEITOS EM IMPRESSÕES 3D COM VISÃO COMPUTACIONAL E APRENDIZADO NÃO SUPERVISIONADO

Trabalho de Conclusão de Curso

Universidade Federal de São Carlos – Campus São Carlos

São Carlos, 28 de fevereiro de 2025

ASSINATURAS E CIÊNCIAS

Cargo/Função	Nome Completo
Orientador	Sidney Bruce Shiki
Membro da Banca 1	Mario Godinho Junior
Membro da Banca 2	José Otávio Savazzi



Documento assinado eletronicamente por **Mario Godinho Junior, Professor(a) do Ensino Superior**, em 28/02/2025, às 09:41, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Sidney Bruce Shiki, Docente**, em 28/02/2025, às 09:42, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site <https://sei.ufscar.br/autenticacao>, informando o código verificador **1757053** e o código CRC **714213ED**.

Referência: Caso responda a este documento, indicar expressamente o Processo nº 23112.005201/2025-06

SEI nº 1757053

Modelo de Documento: Grad: Defesa TCC: Folha Aprovação, versão de 02/Agosto/2019

Documento assinado digitalmente



JOSE OTAVIO SAVAZZI

Data: 28/02/2025 11:21:12-0300

Verifique em <https://validar.it.gov.br>

Agradecimentos

Ao concluir esta jornada acadêmica, sinto-me profundamente grato por todos aqueles que contribuíram para minha trajetória. O caminho percorrido foi resultado de um conjunto de apoios e colaborações que moldaram minha formação pessoal e profissional.

Inicialmente, meus mais sinceros agradecimentos ao meu orientador, Prof. Dr. Sidney Bruce Shiki, cuja dedicação e expertise foram fundamentais para o desenvolvimento deste projeto. Sua orientação precisa e comprometimento foram essenciais para realização desta pesquisa, transformando um interesse de campo da minha parte em um trabalho significativo. A oportunidade de explorar um conteúdo de pesquisa alinhado com meus interesses foi um privilégio inestimável, pelo qual serei eternamente grato.

Aos meus pais e irmãos, minha mais profunda gratidão. Vocês foram meu porto seguro, minha referência e meu suporte incondicional. Nos momentos de dúvida e desafio, sempre estiveram presentes, oferecendo amparo emocional e perspectivas fundamentais para meu crescimento.

Aos professores do Departamento de Engenharia Mecânica (DEMec) da Universidade Federal de São Carlos, meu mais sincero reconhecimento. Cada aula e desafio contribuíram de maneira singular para minha formação, preparando-me como profissional e indivíduo crítico.

Aos meus amigos, que tornaram essa trajetória acadêmica especial, sendo nos momentos difíceis ou felizes, minha eterna gratidão. Um agradecimento especial ao Matheus Rocha, companheiro desde o início da graduação, que esteve presente em cada desafio sempre disposto a contribuir e participar independente da dificuldade.

Por fim, dedico um agradecimento especial a Victoria Sabadoto Brienze. Mais do que uma parceira, foi uma amiga, conselheira e apoio fundamental nesta jornada. Sua presença e contribuição foram essenciais para que eu pudesse atravessar os desafios deste período com resiliência e determinação.

Resumo

Garantir a qualidade de peças fabricadas por impressão 3D é um desafio crucial para a indústria. Inspeções convencionais, apesar de eficientes, costumam exigir equipamentos de alto custo, o que pode limitar sua aplicação em algumas empresas. Neste contexto, explorar alternativas mais acessíveis para detectar defeitos torna-se uma necessidade crescente. Este estudo explora o uso de tecnologias acessíveis e de baixo custo para a detecção de defeitos em peças impressas em 3D. Imagens dos componentes fabricados foram capturadas e analisadas utilizando algoritmos de análise, com o auxílio do OpenCV para detecção de bordas e segmentação de imagens. A análise identificou defeitos como bolhas, subextrusão e sobreextrusão. O aprendizado não supervisionado, utilizando o agrupamento Kmeans, possibilitou a classificação e caracterização dos defeitos com base em seus padrões visuais e distribuição espacial. Os resultados demonstram o potencial de integrar técnicas de visão computacional de baixo custo com aprendizado não supervisionado para aprimorar a inspeção na manufatura aditiva.

Palavras-chave: Impressão 3D, Detecção de Defeitos, OpenCV, Aprendizado de Máquina, Aprendizado Não Supervisionado, Manufatura Aditiva.

Abstract

The quality monitoring of 3D-printed components is essential to ensure product reliability and optimize manufacturing processes. Traditional inspection methods often rely on expensive equipment, making them unfeasible for many industries. This study explores the use of accessible and low-cost technologies for defect detection in 3D-printed parts. Images of the manufactured components were captured and analyzed using analytical algorithms, with the assistance of OpenCV for edge detection and image segmentation. The analysis identified defects such as bubbles, under-extrusion, and over-extrusion. Unsupervised learning, utilizing Kmeans clustering, enabled the classification and characterization of defects based on their visual patterns and spatial distribution. The results demonstrate the potential of integrating low-cost computer vision techniques with unsupervised learning to enhance inspection in additive manufacturing.

Keywords: 3D Printing, Defect Detection, OpenCV, Machine Learning, Unsupervised Learning, Additive Manufacturing.

Lista de Figuras

Figura 1 - Estudo realizado por Wang et al. (2024)	5
Figura 2 - Exemplificação de agrupamento do Kmeans	11
Figura 3 - Fluxograma utilizado para pré-processamento	12
Figura 4 - Aplicação da detecção de bordas de Canny para diferentes peças	13
Figura 5 - Visualização do cálculo de Local Binary Patterns (LBP)	15
Figura 6 - Representação do HOG	16
Figura 7 - Subsistemas da célula de manufatura aditiva robotizada.	23
Figura 8 - Imagem “Normal”	25
Figura 9 - Imagem “Bolhas”	25
Figura 10 - Imagem “Overextrusion40”	26
Figura 11 - Imagem “Overextrusion10”	27
Figura 12 - Imagem “Overextrusion”	27
Figura 13 - Exemplo de análise para RGB:	29
Figura 14 - Demonstração de como as features foram observadas	30
Figura 15 - Exemplo de uso do Método do Cotovelo	33
Figura 16 - Representação do Silhouette Score	35
Figura 17 - Fluxograma de funcionamento do algoritmo	36
Figura 18 - Representação das imagens sob a vista do OpenCV	38
Figura 19 - Média, desvio padrão e distribuição das features por grupo de imagem (Canny)	41
Figura 20 - Média, desvio padrão e distribuição das features por grupo de imagem (Mascara de pixels)	41
Figura 21 - Média, desvio padrão e distribuição das features por grupo de imagem (HOG)	42
Figura 22 - 100 features mais relevantes para diferenciação	43
Figura 23 - Distribuição dos p-values nas features	44
Figura 24 - p-values das features principais	45
Figura 25 - Método do cotovelo obtido para 4 clusters	46
Figura 26 - Silhouette Score obtido	47
Figura 27 - Resultado nos dados de treino	47
Figura 28 - Quantidade de arquivos por cluster	49
Figura 29 - Resultados no conjunto de validação	50
Figura 30 - Conjunto de dados completo (com PCA) e modelo clusterizado	52
Figura 31 - Conjunto de dados inteiro clusterizado	52
Figura 32 - Imagem de baixa qualidade para o modelo	53
Figura 33 - Clusterização do dataset estendido	54
Figura 34 - Distribuição de categorias por cluster	55

Lista de Siglas

CAD	ComputerAided Design (Desenho Assistido por Computador)
LBP	Local Binary Patterns
RGB	RedGreenBlue (VermelhoVerdeAzul)
PCA	Principal Component analysis
HOG	Histogram of Oriented Gradients
EDA	Exploratory Data Analysis
DBSCAN	Density-Based Spatial Clustering of Applications with Noise

Lista de Símbolos

Σ	Matriz de covariância dos dados
Q	Matriz de autovetores (direções principais)
Λ	Matriz diagonal contendo os autovalores
W	Matriz dos autovetores utilizados na transformação
Z	Matriz transformada no espaço dos componentes principais
X	Matriz de dados original
μ	Média dos valores da variável
σ	Desvio padrão da variável
k	Número de clusters
C	Conjunto de pontos pertencentes ao cluster
\bar{x}	Centróide do cluster
d	Distância euclidiana entre um ponto e o centroide
s	Silhouette Score de um ponto
a	Distância média do ponto aos outros pontos do mesmo cluster
b	Distância média do ponto ao cluster mais próximo
α	Proporção

Sumário

1. INTRODUÇÃO.....	1
2. REVISÃO BIBLIOGRÁFICA.....	4
2.1. Detecção de Defeitos em Imagens.....	4
2.2. Clusterização e Agrupamento de Anomalias.....	6
2.3. Extração e Análise de Características.....	7
2.4. Importância do Processamento Automático.....	7
2.5. Aprendizado não supervisionado.....	8
2.6. Técnicas de Visão Computacional para Detecção de Defeitos.....	10
2.6.1. Pré-processamento de Imagens.....	11
2.6.2. Detecção de Bordas com Algoritmo de Canny.....	12
2.6.3. Local Binary Patterns (LBP).....	13
2.6.4. Histogram of Oriented Gradients (HOG).....	15
2.6.5. Segmentação por Intensidade de Pixels.....	16
2.7. Análise de Componentes Principais (PCA) e Normalização com Scaler.....	17
2.7.1. Análise de Componentes Principais (PCA).....	17
2.7.2. Normalização com Scaler.....	19
2.8. Relevância Prática.....	20
2.9. Objetivos.....	20
2.9.1. Objetivo Geral.....	20
2.9.2. Objetivos Específicos.....	21
3. MATERIAIS E MÉTODOS.....	22
3.1. Descrição da célula de manufatura aditiva robotizada (CMAR) e imagens utilizadas.....	22
3.2. Descrição das imagens utilizadas.....	24
3.3. Análise exploratória de dados.....	28
3.4. Implementação do código, métricas e avaliação do modelo.....	31
3.4.1. Algoritmo Implementado.....	31
3.4.2. Métricas para Avaliação do Modelo.....	32
3.4.2.1. Elbow Method.....	32
3.4.2.2. Silhouette Score.....	34
3.4.3. Avaliação do Modelo.....	36
4. RESULTADOS E DISCUSSÃO.....	37
4.1. Análise Exploratória de Dados.....	37
4.1.1. Pré-processamento, Observações Visuais e Distribuição de Features.....	37
4.1.2. Análise das Médias e Desvio Padrão das Features.....	39
4.1.3. Análise do HOG.....	42
4.2. Modelo de Validação e Resultados.....	45
4.2.1. Determinação do Número de Clusters.....	45

4.2.2. Caracterização dos Clusters.....	48
4.3. Métodos de Validação.....	51
5. SUGESTÕES PARA TRABALHOS FUTUROS.....	56
6. CONCLUSÃO.....	58
Referências.....	59

Declaração do uso de Inteligência Artificial Generativa

Durante a elaboração deste trabalho, foram utilizados modelos de Inteligência Artificial generativa para aprimorar sua qualidade e fundamentação. Especificamente, utilizei o ChatGPT para correção de textos e auxílio na estruturação textual, o My Stylus AI para a busca de artigos científicos que poderiam enriquecer os conceitos base do estudo e por fim o Claude AI para sugestões de algoritmos passíveis de implementação. Ressalto que todas as ferramentas foram utilizadas como suporte, e que as ideias e pontos de vista expressos neste trabalho são integralmente de minha autoria, não refletindo opiniões ou posicionamentos das IAs mencionadas.

1. INTRODUÇÃO

A indústria está em constante evolução, e a busca por processos mais eficientes e econômicos se tornou uma necessidade. Dentro desse cenário, a impressão 3D vem ganhando espaço como uma alternativa inovadora e flexível, capaz de complementar ou até mesmo substituir métodos tradicionais de fabricação. Esse avanço está diretamente ligado à sua crescente adoção em ambientes industriais automatizados, onde a busca por precisão e otimização é constante (Wong e Hernandez, 2012).

A impressão 3D funciona a partir de modelos tridimensionais digitais gerados em softwares de CAD (*Computer-Aided Design*). Esses modelos servem como base para a fabricação camada por camada do objeto final (Volpato, 2017). Essa abordagem oferece maior liberdade geométrica na concepção de peças, possibilitando a criação de estruturas complexas. Muitas dessas geometrias seriam inviáveis ou economicamente inviáveis em métodos tradicionais de manufatura subtrativa. Além disso, a manufatura aditiva contribui para uma significativa redução de desperdício de matéria-prima, tornando-se uma alternativa mais sustentável e econômica.

Na década de 1990, a manufatura aditiva passou a ser classificada com base nos tipos de materiais empregados — líquidos, pós ou sólidos — e nas técnicas utilizadas para a construção das camadas, assim como na pesquisa de Gao et al., 2015. A ASTM (*American Society for Testing and Materials*) define essas tecnologias em diferentes categorias. A extrusão de materiais, por exemplo, consiste na deposição de camadas sucessivas de material fundido, enquanto a fusão seletiva a laser utiliza um feixe de alta energia para derreter seletivamente pós metálicos ou poliméricos. Na fotopolimerização em cuba, uma resina líquida é solidificada por meio de luz ultravioleta. Já o jato de material funciona de maneira semelhante a uma impressora a jato de tinta, mas utiliza materiais que se solidificam ao entrar em contato com a superfície. Por fim, a deposição de energia dirigida emprega um feixe de energia para fundir diretamente pó ou fio metálico sobre a peça em construção. Essas técnicas, apesar de distintas, compartilham a necessidade de monitoramento constante para garantir a qualidade das peças

produzidas, uma vez que falhas em qualquer etapa podem comprometer a integridade estrutural e funcional do produto final.

Para assegurar a qualidade final das peças produzidas, é fundamental que cada etapa do processo de impressão 3D seja monitorada cuidadosamente. Pequenas falhas durante a impressão podem comprometer não apenas a estética, mas também a funcionalidade da peça. O monitoramento contínuo possibilita a detecção de defeitos como desvios dimensionais, falhas de aderência entre camadas ou imperfeições na superfície, permitindo correções em tempo real e minimizando desperdícios (Paraskevaïdis, Karayannis e Koumoulos, 2020).

A aplicação de visão computacional e aprendizado de máquina na inspeção de peças impressas em 3D tem se tornado cada vez mais comum. Essas técnicas permitem identificar falhas automaticamente, analisando padrões nas imagens capturadas durante o processo de fabricação, sem precisar de uma base de dados previamente rotulada (Era et al., 2023). No presente trabalho, foi utilizado o algoritmo de *Kmeans* para agrupar os defeitos detectados com base em suas características. A biblioteca *OpenCV* foi utilizada para capturar e processar as imagens, identificando regiões de interesse e criando máscaras de defeitos. Além disso, a biblioteca *Matplotlib* foi empregada para exibir os resultados visualmente, destacando as áreas com possíveis falhas e facilitando a análise.

O sistema desenvolvido se mostrou eficaz para automatizar a detecção de defeitos em peças impressas em 3D. Ele reduz a dependência de inspeções manuais, tornando o controle de qualidade mais preciso e compatível com os padrões da Indústria 4.0.

No estudo de Wang et al. (2021), os pesquisadores exploraram o uso de técnicas de aprendizado não supervisionado em imagens de superfícies metálicas, buscando agrupar defeitos com características similares, como rugosidade, bolhas e rachaduras. A análise foi fundamentada na extração de descritores geométricos e texturais, que possibilitaram a detecção automatizada de falhas não perceptíveis a olho nu. Essa abordagem demonstrou ser eficaz para sistemas de inspeção automatizada, destacando a importância da combinação de visão computacional e aprendizado de máquina na melhoria dos processos industriais.

A metodologia desta pesquisa visa contribuir para o desenvolvimento de um sistema automatizado e eficiente de monitoramento da qualidade na impressão 3D, capaz de reduzir a dependência de inspeção manual, aumentar a precisão na detecção de falhas e garantir maior consistência na produção de peças, alinhando-se às demandas da Indústria 4.0 por processos cada vez mais inteligentes e autônomos.

2. REVISÃO BIBLIOGRÁFICA

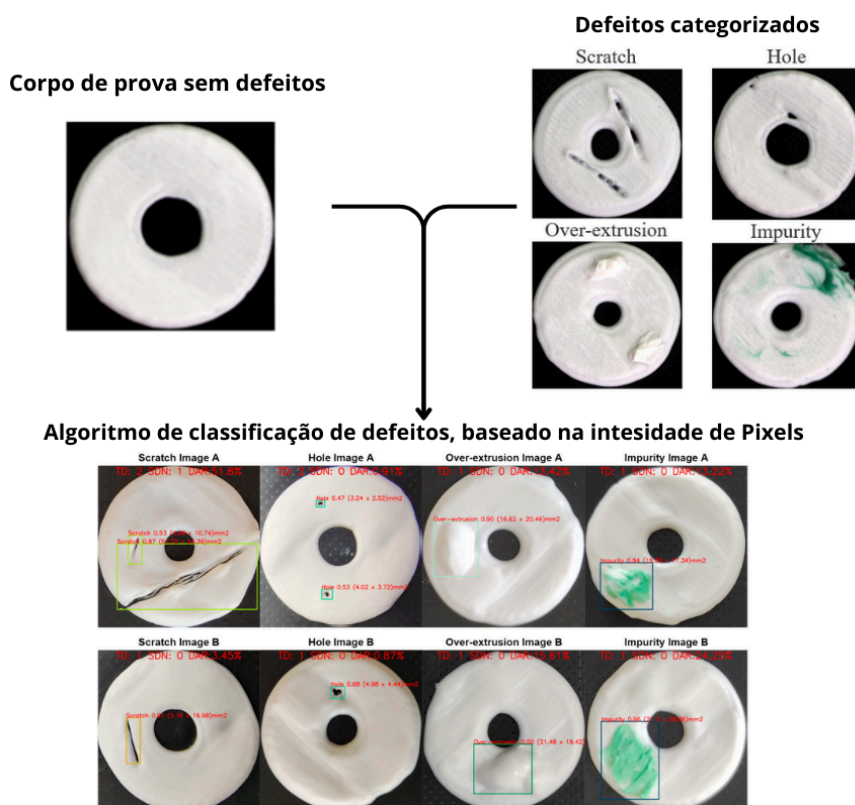
A detecção de defeitos em superfícies industriais e manufatura aditiva tem recebido atenção crescente na literatura, impulsionada pela necessidade de automação e inspeção de qualidade em ambientes de produção modernos. Esta seção revisa abordagens recentes e inovações no uso de técnicas de visão computacional e aprendizado de máquina para identificação de anomalias e padrões em superfícies complexas

2.1. Detecção de Defeitos em Imagens

A análise de imagens para identificar defeitos é um campo que combina visão computacional, processamento digital e aprendizado de máquina. O objetivo é encontrar padrões ou irregularidades que possam comprometer a qualidade do produto final. Na manufatura aditiva, defeitos como bolhas, subextrusão e sobreextrusão são desafiantes de identificar devido à sua natureza irregular, podendo comprometer a integridade das peças fabricadas.

Defeitos como bolhas ou vazios, frequentemente encontrados em processos de manufatura aditiva, são particularmente desafiantes de identificar devido à sua natureza irregular. Estudos recentes como o de Wang et al. (2024) demonstram que técnicas de segmentação baseadas em intensidade de pixels, aliadas a descritores texturais, podem detectar essas anomalias com precisão superior a 95%, na Figura 1 é possível observar como o experimento dos autores foi realizado. Bolhas, por exemplo, são frequentemente caracterizadas por regiões com intensidades abruptamente diferentes das áreas vizinhas, o que permite sua segmentação e análise detalhada. Em paralelo, os vazios são identificáveis por padrões texturais característicos, frequentemente capturados por métodos como o *Local Binary Patterns* (LBP).

Figura 1 - Estudos sobre padrões de defeitos



Fonte: Wang et al. (2024)

Além disso, defeitos superficiais, como rachaduras ou arranhões, apresentam um comportamento diferente em termos de textura e gradientes, o que pode ser explorado por algoritmos de detecção de bordas, como o Canny. Isso é corroborado por Li et al. (2023), que aplicaram técnicas híbridas de detecção de bordas e segmentação em peças metálicas para capturar e classificar rachaduras de forma automática e precisa.

Técnicas baseadas em detecção de bordas, como o algoritmo de Canny, são amplamente empregadas para identificar contornos significativos em imagens. O método de Canny segue um processo em quatro etapas: suavização da imagem para reduzir ruídos, cálculo de gradientes para detectar mudanças de intensidade, supressão não-máxima para refinar as bordas e limiarização com histerese para selecionar bordas fortes e fracas conectadas (Canny, 1986). Essa técnica é eficaz na identificação de bordas claras em imagens monocromáticas e é amplamente utilizada em sistemas de inspeção automática (Gonzalez e Woods, 2018).

Além disso, técnicas de segmentação por intensidade de pixels, como a limiarização global e adaptativa, são aplicadas para isolar regiões de interesse em uma imagem. A limiarização global utiliza um único limiar fixo para dividir a imagem em duas regiões (fundo e objeto), enquanto a limiarização adaptativa calcula limiares locais para lidar com variações de iluminação (Otsu, 1979). Essas abordagens permitem a criação de máscaras binárias que destacam regiões defeituosas, fornecendo uma base para a extração de características quantitativas.

A detecção de defeitos em imagens não apenas melhora a confiabilidade da análise visual, mas também automatiza tarefas que seriam demoradas e suscetíveis a erros humanos. Combinando técnicas de processamento, como a detecção de bordas e a segmentação, é possível capturar e isolar características de defeitos que são difíceis de identificar em inspeções manuais. Essa abordagem é especialmente relevante na manufatura aditiva, onde as peças frequentemente apresentam geometrias complexas e texturas que tornam a inspeção visual desafiadora.

2.2. Clusterização e Agrupamento de Anomalias

A clusterização é uma técnica de aprendizado não supervisionado utilizada para organizar dados em grupos baseados em similaridades. No contexto de detecção de defeitos, essa abordagem é empregada para agrupar regiões defeituosas com características similares, como tamanho, forma e intensidade de textura. O algoritmo de *Kmeans* é amplamente utilizado para esse fim devido à sua simplicidade e eficácia na separação de *clusters* em dados estruturados (Macqueen, 1967).

Chen et al. (2023) introduziram recentemente uma variante do *Kmeans* que incorpora informações espaciais e texturais, alcançando uma melhoria de 40% na precisão da identificação de defeitos em superfícies complexas. Esta abordagem tem se mostrado particularmente eficaz na manufatura aditiva, onde diferentes tipos de defeitos podem apresentar características visuais similares.

A clusterização também é vantajosa para a detecção de padrões subjacentes nos dados, como a frequência de falhas em regiões específicas ou a identificação de classes de defeitos recorrentes. Estudos recentes, como os de Zhu et al. (2021), demonstraram a

eficácia da clusterização em sistemas de inspeção automática, aumentando a precisão na classificação de defeitos.

2.3. Extração e Análise de Características

A extração de características é uma etapa essencial na detecção de defeitos, pois transforma as imagens em representações numéricas que podem ser analisadas por algoritmos de aprendizado de máquina. No contexto do presente trabalho, técnicas como o LBP e o *Histogram of Oriented Gradients* (HOG) foram utilizadas para capturar informações texturais e geométricas das regiões defeituosas.

O LBP, introduzido por Ojala et al. (2002), é um descritor de textura baseado na comparação da intensidade de um pixel central com seus vizinhos, gerando um código binário que representa os padrões locais da textura. Esse descritor é robusto e computacionalmente eficiente, sendo amplamente utilizado para identificar padrões texturais associados a defeitos superficiais. Por outro lado, o *Histogram of Oriented Gradients* HOG é utilizado para capturar informações geométricas relacionadas à orientação de gradientes em regiões da imagem, permitindo a identificação de contornos e formas de defeitos (Dalal e Triggs, 2005).

Além desses descritores, outras características como área, perímetro e razão entre largura e altura dos defeitos foram calculadas a partir das máscaras binárias geradas na etapa de segmentação. Essas informações foram utilizadas como entrada para os algoritmos de aprendizado de máquina, permitindo uma classificação mais precisa e fundamentada nos padrões detectados.

2.4. Importância do Processamento Automático

O processamento automático de imagens em sistemas de inspeção oferece várias vantagens em relação à inspeção manual. Além de reduzir a subjetividade e os erros humanos, os sistemas automatizados permitem uma análise rápida e consistente, mesmo em cenários de alta demanda industrial. Estudos recentes, como os de Thompson et al. (2023), demonstram que soluções baseadas em visão computacional e aprendizado de máquina não supervisionado podem reduzir significativamente os custos operacionais, aumentando a eficiência produtiva e garantindo a qualidade final das peças fabricadas.

Essa automação é especialmente vantajosa no setor de manufatura aditiva, onde as peças apresentam alta variabilidade em seus formatos e superfícies. O uso de sistemas automatizados elimina a dependência de inspeções manuais demoradas e propensas a erros, garantindo maior precisão e uniformidade. A combinação de técnicas como segmentação, extração de características e algoritmos de aprendizado não supervisionado não apenas acelera o processo de análise, mas também proporciona insights detalhados sobre os padrões de defeitos. Isso permite a tomada de decisões mais assertivas na cadeia de produção, garantindo maior controle de qualidade e eficiência no processo industrial.

2.5. Aprendizado não supervisionado

O *KMeans* é amplamente utilizado em aprendizado não supervisionado para agrupar dados com base em similaridades. Na detecção de defeitos em impressão 3D, ele permite agrupar falhas conforme características como tamanho, forma e posição. Introduzido por MacQueen em 1967, o *Kmeans* busca particionar os dados em *k clusters* distintos, minimizando a variância intra-cluster e maximizando a variância inter-cluster. Este algoritmo é iterativo, começando pela inicialização aleatória de *k* centróides, seguida pelo cálculo das distâncias entre os pontos de dados e os centróides, e pela atualização das posições desses centróides até a convergência. O *Kmeans* tem como vantagens a simplicidade computacional, a eficiência para grandes volumes de dados e a capacidade de produzir agrupamentos bem definidos quando os dados apresentam estruturas claras.

A escolha deste algoritmo é embasada em sua habilidade de lidar com vetores de características numéricas, que podem ser derivados de técnicas de processamento de imagem. No caso dos defeitos analisados, vetores como áreas dos contornos, coordenadas centrais e intensidade média das regiões defeituosas servem como entrada para o modelo.

Entretanto, para que o *Kmeans* seja eficaz, é crucial que os dados sejam adequadamente pré-processados. No estudo apresentado, ferramentas da biblioteca *OpenCV* desempenharam um papel fundamental no tratamento inicial das imagens. Técnicas como a detecção de bordas, por meio do algoritmo de Canny, e a extração

destes contornos com funções disponibilizadas pelo *OpenCv*, foram utilizadas para segmentar os defeitos e transformar as imagens em informações estruturadas. A aplicação de limiares binários garantiu que as regiões de interesse fossem destacadas, reduzindo ruídos e permitindo uma análise mais robusta.

A relevância do *Kmeans* na análise de imagens está na simplicidade do modelo matemático, que possibilita rápida implementação e ajuste. Contudo, limitações como a sensibilidade ao número de *clusters* k e a dependência de inicializações aleatórias podem afetar os resultados. Estudos como os de Jain (2010) discutem estratégias para mitigar essas limitações, incluindo a execução de múltiplas inicializações e o uso de métricas como o coeficiente de silhueta para determinar o valor ótimo de k . Além disso, metodologias complementares, como o uso de aprendizado supervisionado após a clusterização, podem ser exploradas para rotular automaticamente os *clusters* com base em padrões conhecidos.

A principal equação que descreve o *Kmeans* envolve a minimização da distância entre os pontos de dados e os centróides dos *clusters*. A equação 1 de otimização do *Kmeans* é a seguinte:

$$J = \sum_{i=1}^m \sum_{k=1}^K \{c_i = k\} \|x_i - \mu_k\|^2 \quad (1)$$

Onde: J é a soma das distâncias quadradas dentro de todos os *clusters*, k é o número de *clusters*.

x_i são os pontos de dados.

c_i É o conjunto de pontos de dados no *cluster* i .

μ_k é o centróide do *cluster* i , $\|x_i - \mu_k\|^2$ é a distância quadrática entre o ponto de dado x_i e o centróide μ_k .

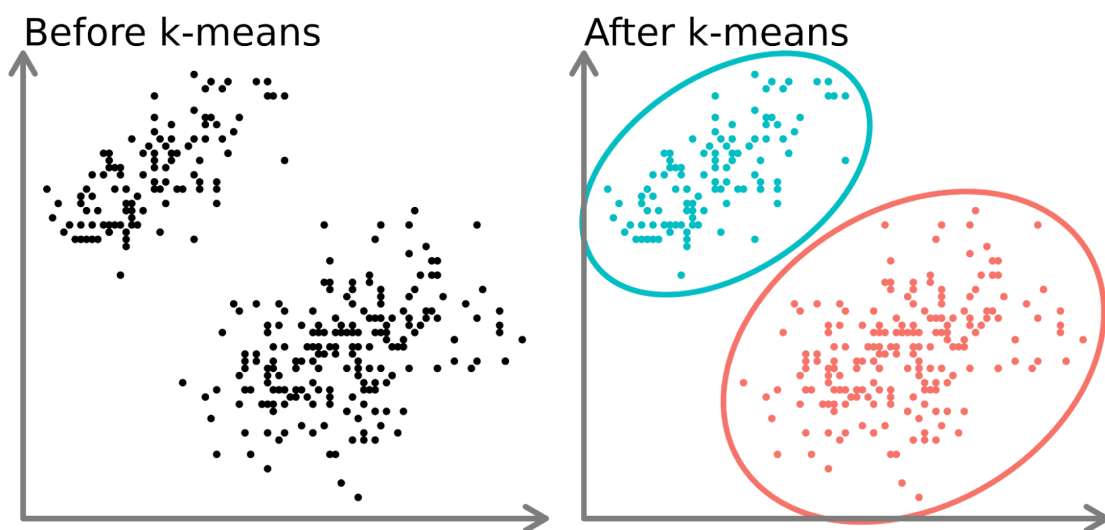
O uso de clusterização para análise de defeitos em manufatura aditiva representa uma interseção interessante entre a visão computacional e o aprendizado de máquina. Segundo Sun et al. (2021), sistemas automatizados baseados em aprendizado não supervisionado têm o potencial de reduzir significativamente os custos de inspeção e melhorar a consistência na identificação de anomalias, principalmente em processos

onde a inspeção manual seria impraticável devido à escala ou à complexidade das peças produzidas.

Embora o *KMeans* tenha sido escolhido pela sua eficiência e simplicidade, alternativas como o *DBSCAN* (*Density-Based Spatial Clustering of Applications with Noise*) poderiam lidar melhor com ruídos e dados não estruturados. No entanto, devido à distribuição das *features* e à necessidade de definir *clusters* bem separados, o *KMeans* mostrou-se mais adequado para esta aplicação.

Em resumo, a escolha do *Kmeans* e das técnicas de visão computacional reflete um alinhamento entre simplicidade algorítmica e eficácia prática, permitindo uma análise adequada de defeitos em impressão 3D e contribuindo para avanços na automação de processos industriais. Na Figura 2 é possível observar uma exemplificação de como o algoritmo de agrupamento *Kmeans* funciona na prática.

Figura 2 - Exemplificação de agrupamento do *Kmeans*



Fonte: DataCamp (2023)

2.6. Técnicas de Visão Computacional para Detecção de Defeitos

A visão computacional oferece uma ampla gama de técnicas para analisar imagens e identificar defeitos em superfícies, incluindo pré-processamento, extração de características e segmentação. Esses métodos permitem processar imagens de forma

eficiente, mesmo em cenários com alta complexidade visual, como na análise de peças impressas em 3D (Kovalskyi, 2024).

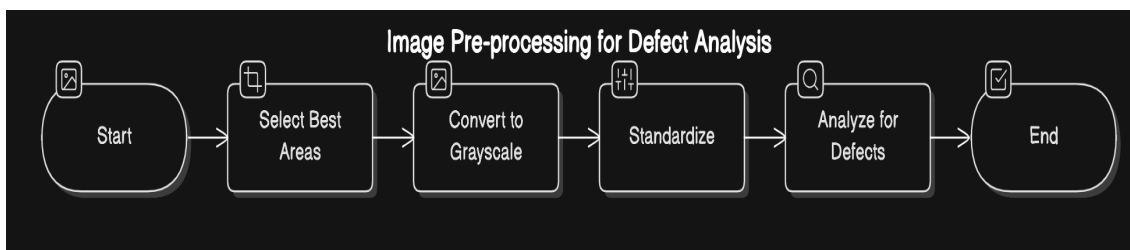
2.6.1. Pré-processamento de Imagens

O pré-processamento é uma etapa crucial que melhora a qualidade da imagem, reduzindo ruídos e destacando características relevantes para as etapas posteriores de análise. Entre as técnicas mais utilizadas destacam-se:

- Suavização com Filtros Gaussianos: Reduz os detalhes de alta frequência, como ruídos e texturas finas, utilizando um filtro Gaussiano. Gonzalez e Woods (2018) destacam que a suavização preserva bordas principais enquanto elimina interferências indesejadas.
- Conversão para Escala de Cinza: Simplifica o processamento, reduzindo a imagem para um único canal de intensidade. Essa técnica é implementada transformando a imagem em tons de cinza. Essas etapas criam uma base sólida para métodos mais complexos de análise, garantindo resultados mais robustos na detecção de padrões e defeitos.
- Redimensionamento de imagem: Ajuda o algoritmo a focar apenas na imagem de forma que o ambiente ao redor não seja analisado da mesma maneira que a peça.

Na Figura 3 é possível observar um fluxograma de funcionamento do Pré-processamento que foi utilizado e elaborado por este projeto:

Figura 3 - Fluxograma utilizado para pré-processamento



Fonte: Próprio Autor

2.6.2. Detecção de Bordas com Algoritmo de Canny

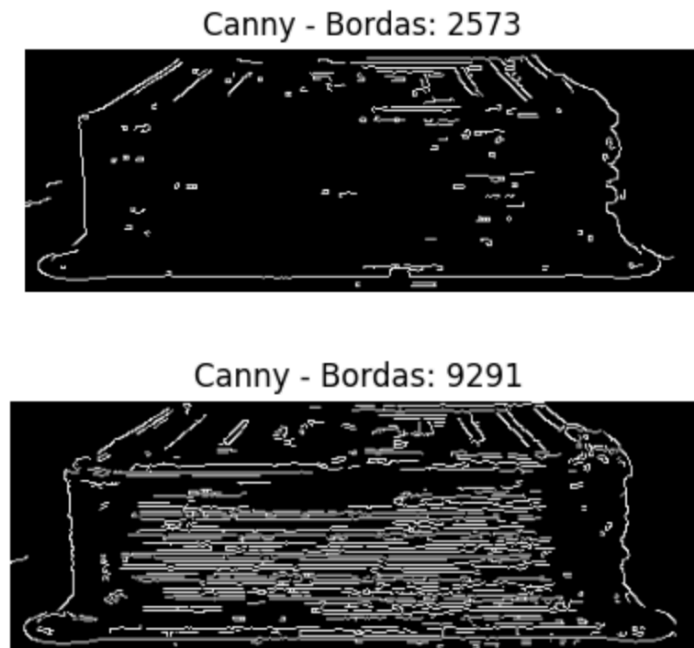
O algoritmo de Canny (Canny, 1986) é amplamente utilizado para detectar bordas, sendo eficaz em destacar contornos de objetos e defeitos. Ele é composto pelas seguintes etapas:

- Suavização Gaussiana: Reduz ruídos para evitar falsos positivos.
- Cálculo dos Gradientes: Utiliza filtros de Sobel para calcular as direções e magnitudes dos gradientes de intensidade em cada pixel (Sobel, 2022).
- Supressão Não-Máxima: Refinar as bordas, mantendo apenas os pixels com gradiente máximo ao longo das direções calculadas.
- Limiarização com Histerese: Identifica bordas fortes e conecta bordas fracas relacionadas.

Trabalhos como os de Suzuki e Abe (1985) discutem como a hierarquia de contornos pode ser explorada para diferenciar defeitos superficiais de imperfeições estruturais mais profundas.

No caso de peças impressas em 3D, bordas bem definidas facilitam a identificação de anomalias, como bolhas ou falhas de extrusão. O *OpenCV* oferece a funcionalidade de aplicar o algoritmo de detecção de bordas de Canny, permitindo identificar padrões e verificar se os contornos extraídos condizem com as bordas esperadas. Essa técnica possibilita ajustar parâmetros como os limiares de detecção, garantindo maior precisão na diferenciação entre pixels do contorno e do fundo. Na Figura 4, são apresentados exemplos dessa aplicação em duas imagens distintas, destacando a quantidade de bordas detectadas e demonstrando a versatilidade dessa abordagem na análise de contornos.

Figura 4 - Aplicação da detecção de bordas de Canny para diferentes peças



Fonte: Próprio Autor

2.6.3. *Local Binary Patterns* (LBP)

O *Local Binary Patterns* (LBP) é um descritor de textura amplamente utilizado em visão computacional devido à sua simplicidade e eficácia na análise de padrões locais em imagens. Ele foi originalmente proposto por Ojala et al. (2002) como uma técnica para codificar a textura de uma região de interesse, comparando a intensidade de cada pixel central com a de seus vizinhos em uma janela definida. Essa comparação gera um código binário que descreve a variação de intensidade no entorno do pixel central, sendo particularmente eficaz para capturar padrões texturais, como rugosidades ou defeitos de superfícies. A equação 2 demonstra como calcular o LBP em um pixel central (x_c, y_c) :

$$LBP(x_c, y_c) = \sum_{n=0}^{P-1} 2^n \cdot s(i_n - i_c) \quad (2)$$

Onde $s(x) = 1$ se $x \geq 0$, ou $s(x) = 0$ se $x < 0$.

x_c, y_c : Coordenadas do pixel central no qual o LBP está sendo calculado.

P : Número de vizinhos ao redor do pixel central.

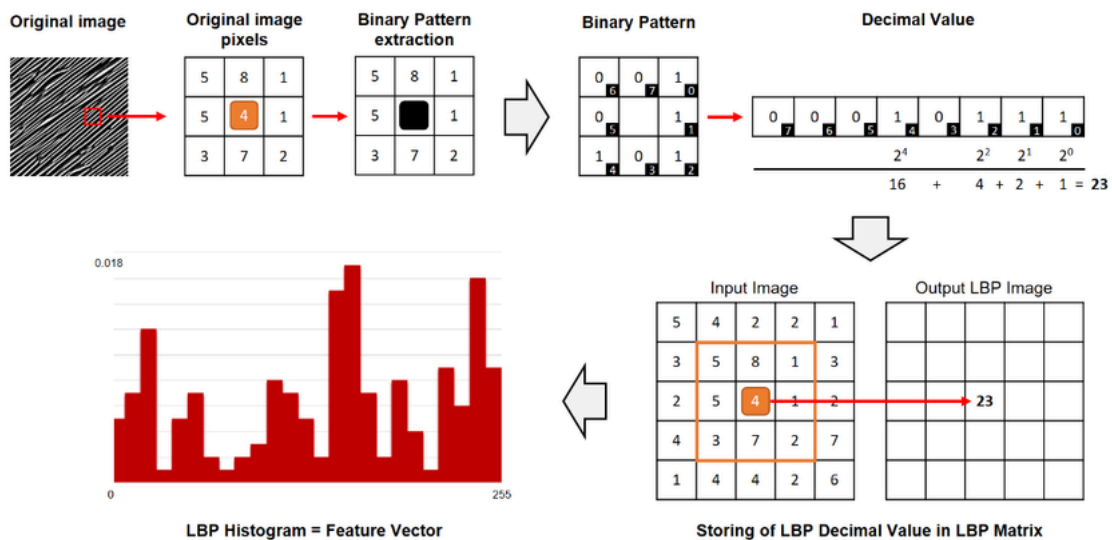
i_c : Intensidade do pixel central na imagem em escala de cinza.

i_n : Intensidade do n -ésimo pixel vizinho ao redor do pixel central.

$s(i_n - i_c)$: Função de limiarização que verifica se o vizinho é maior ou menor que o pixel central:

A Figura 5 demonstra a representação de como o LBP funciona com uma imagem prática.

Figura 5 - Visualização do cálculo de *Local Binary Patterns* (LBP)



Fonte: ResearchGate (2020).

Após o cálculo do LBP em todos os pixels de uma imagem, a textura global é representada por meio de um histograma que contabiliza a frequência dos padrões binários gerados. Esse histograma atua como um descritor robusto de textura, sendo amplamente utilizado como entrada para algoritmos de aprendizado de máquina ou clusterização. A simplicidade do LBP o torna adequado para aplicações em que rapidez e eficiência computacional são essenciais, como na detecção de defeitos em peças impressas em 3D. No contexto deste estudo, o LBP mostrou-se eficaz para destacar diferenças texturais entre regiões normais e defeituosas, como rugosidades ou bolhas.

2.6.4. Histogram of Oriented Gradients (HOG)

Os autores Dalal e Triggs (2005) introduziram um método simples e eficaz para separar características para detecção de objetos, que é normalmente usado para a identificação de objetos em reconhecimento de padrões e visão computacional. HOG é um método de detecção de objetos que conta as ocorrências de orientação de gradiente em porções localizadas de uma imagem. HOG usa variações de intensidade local e, portanto, é menos sensível a mudanças de iluminação e pequenas distorções geométricas do que métodos baseados em recursos de pixels brutos.

A extração de características do HOG refere-se a uma sequência de etapas: a imagem é processada primeiro para gerar uma escala de cinza antes de calcular os gradientes de intensidade horizontal e vertical. Esses gradientes são agrupados em células, e histogramas de gradientes orientados são construídos. As células são normalizadas em blocos sobrepostos para tornar a representação menos sensível à iluminação e contraste. Portanto, o vetor de características gerado carrega informações estruturais que permitem o reconhecimento de padrões na imagem.

Matematicamente, os gradientes da imagem são obtidos utilizando derivadas diferenciais de acordo com as equações 3 e 4:

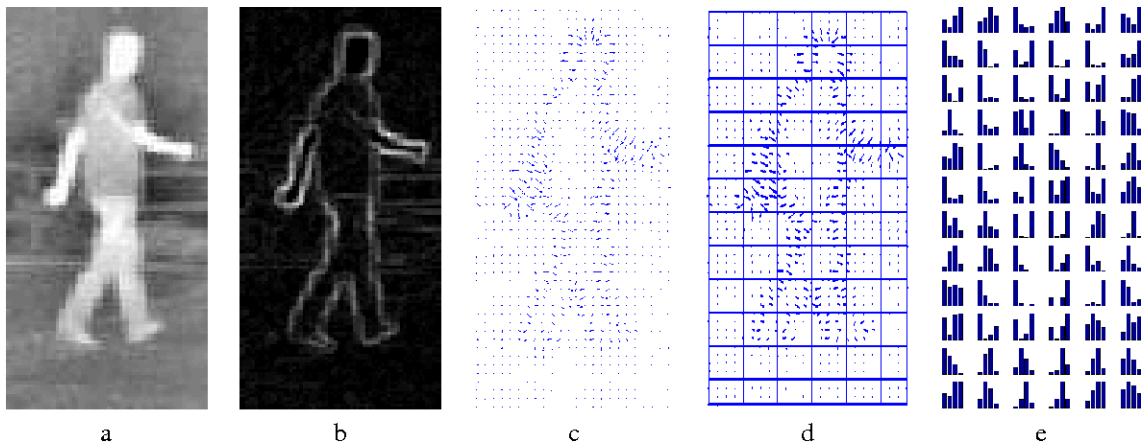
$$G_x = I(x + 1, y) - I(x - 1, y) \quad G_y = I(x, y + 1) - I(x, y - 1) \quad (3)$$

Onde, G_x e G_y representam os gradientes horizontais e verticais da Imagem I, respectivamente. A Magnitude M e a orientação θ do gradiente são então calculadas como:

$$M = \sqrt{G_x^2 + G_y^2} \quad \theta = \arctan\left(\frac{G_x}{G_y}\right) \quad (4)$$

Na Figura 6 é possível observar como o HOG pode ser representado e analisado tanto qualitativa quanto quantitativamente.

Figura 6 - Representação do HOG



Fonte: Semantic Scholar 2007

Desde sua introdução, o HOG tem sido comparado a outras técnicas de extração de características, como SIFT (*Scale-Invariant Feature Transform*) e SURF (*Speeded Up Robust Features*). O HOG é caracterizado por apresentar melhor desempenho para tarefas de detecção de objetos estáticos, devido à sua robustez a ruído e variações de iluminação. No entanto, técnicas baseadas em redes neurais profundas, como CNNs (*Convolutional Neural Networks*), têm se tornado alternativas viáveis devido à capacidade de aprendizado de características mais complexas.

A utilização do HOG na inspeção e análise de defeitos em manufatura aditiva tem ganhado atenção na literatura recente como visto no trabalho de Caggiano, Alessandra et al. (2025). A impressão 3D pode apresentar defeitos estruturais como delaminação, porosidade e irregularidades superficiais. O HOG, quando combinado com técnicas de aprendizado de máquina, como *KMeans* e *Support Vector Machines* (SVMs), tem sido empregado para detectar e classificar esses defeitos com alta precisão.

2.6.5. Segmentação por Intensidade de Pixels

A segmentação de imagens é uma técnica fundamental em visão computacional, cujo objetivo é dividir a imagem em regiões homogêneas, permitindo isolar áreas de interesse. No contexto da detecção de defeitos em peças impressas em 3D, a segmentação baseada em intensidade de pixels é amplamente utilizada devido à sua simplicidade e eficiência. Essa abordagem utiliza informações de intensidade (valores

de pixel) para identificar e separar regiões específicas que possam conter defeitos, como bolhas ou áreas de subextrusão. A segmentação por intensidade é particularmente adequada para imagens com variações sutis, mas significativas, em tons de cinza.

Essa técnica produz máscaras binárias, que destacam as regiões de interesse, separando-as do fundo. Essas máscaras são essenciais para etapas subsequentes, como extração de características e detecção de contornos. As máscaras permitem isolar áreas defeituosas e fornecer métricas quantitativas, como área e forma dos defeitos.

Na análise de peças impressas em 3D, a segmentação por intensidade de pixels foi aplicada para identificar falhas como bolhas, sobreextrusão e subextrusão. Estudos como os de Rao et al. (2021) demonstram que a limiarização adaptativa é particularmente eficaz para segmentar defeitos em peças com padrões de iluminação irregulares, enquanto a limiarização global é eficiente em casos de iluminação controlada e uniforme.

2.7. Análise de Componentes Principais (PCA) e Normalização com Scaler

2.7.1. Análise de Componentes Principais (PCA)

A Análise de Componentes Principais (PCA, do inglês *Principal Component Analysis*) é uma técnica estatística amplamente utilizada em problemas de alto volume de dados para simplificar sua representação, reduzindo a dimensionalidade ao mesmo tempo que preserva a maior parte da variância presente nos dados. Essa técnica é fundamental em projetos que lidam com dados complexos e multivariados, como a análise de imagens industriais.

O PCA opera transformando o espaço original dos dados em um novo conjunto de eixos (componentes principais), que são ordenados de acordo com a variância que representam. Matematicamente e computacionalmente, o PCA encontra os autovalores e autovetores da matriz de covariância dos dados, com os autovalores representando a variância explicada por cada componente principal. O processo é descrito matematicamente nas equações 5 e 6 pela decomposição da matriz de covariância :

$$\Sigma = Q\Lambda Q^T \quad (5)$$

Σ : Matriz de covariância dos dados.

Q : Matriz de autovetores (direções principais).

Λ : Matriz diagonal contendo os autovalores (variância explicada por cada componente principal).

Os dados projetados nos componentes principais são calculados como:

$$Z = X \cdot W \quad (6)$$

X : Matriz de dados original.

W : Matriz dos autovetores.

Z : Matriz transformada no espaço dos componentes principais.

A redução de dimensionalidade proporcionada pelo PCA tem diversas vantagens, incluindo:

Facilidade de visualização: Com a redução para duas ou três dimensões, é possível interpretar visualmente os padrões em dados que inicialmente possuem dezenas ou centenas de variáveis.

Redução de redundâncias: O PCA elimina correlações redundantes entre as variáveis, condensando a informação em um conjunto menor de dimensões.

Melhoria na eficiência computacional: Processar dados em dimensões reduzidas diminui significativamente o custo computacional de etapas subsequentes, como a clusterização ou classificação.

Estudos como o de Jolliffe (2016) demonstram que o PCA é uma ferramenta eficaz para encontrar padrões ocultos em conjuntos de dados complexos, especialmente quando o objetivo é identificar as dimensões mais relevantes para a análise. Em projetos relacionados à manufatura aditiva, o PCA poderia ser aplicado para analisar características como texturas, bordas e intensidade de pixels, facilitando a identificação de padrões de defeitos em superfícies industriais.

2.7.2. Normalização com *Scaler*

A normalização é um passo essencial em análises de dados que envolvem variáveis com escalas distintas. Em muitos casos, como na análise de imagens, diferentes características extraídas podem variar em ordens de magnitude, como intensidades de pixels e contagens de bordas. Se essas variáveis não forem normalizadas, as que possuem maiores magnitudes podem dominar as análises e enviesar os resultados.

O *Standard Scaler* é uma técnica de normalização que transforma as variáveis para que possuam média zero e desvio padrão igual a um. A fórmula para essa transformação é dada na equação 7:

$$X' = \frac{X - \mu}{\sigma} \quad (7)$$

Onde:

X: Valor original.

μ : Média da variável.

σ : Desvio padrão da variável.

X' é o Valor normalizado ou *Z-score*.

A normalização com o *Standard Scaler* é amplamente recomendada em análises que envolvem o PCA, pois garante que todas as variáveis sejam tratadas de maneira equitativa. O PCA calcula os componentes principais com base na variância dos dados, e sem a normalização, variáveis com maiores magnitudes poderiam influenciar desproporcionalmente os resultados.

Estudos como os de Géron (2019) mostram que a aplicação do *Standard Scaler* é essencial em cenários de aprendizado de máquina e análise de dados multivariados, especialmente quando há a necessidade de combinar variáveis heterogêneas em uma única análise. No contexto deste projeto, o uso do *Standard Scaler* antes do PCA poderia assegurar que todas as características extraídas das imagens, como intensidade de pixels, textura e contornos, sejam tratadas de forma balanceada, permitindo uma análise mais precisa e imparcial.

2.8. Relevância Prática

A aplicação da *OpenCV* na análise de defeitos em impressão 3D combina eficiência computacional com flexibilidade. Essa biblioteca de código aberto se destaca pela sua ampla gama de funções, como filtragem de imagens, detecção de bordas, segmentação e extração de características, que podem ser facilmente integradas em *pipelines* automatizados. A capacidade de processar imagens em tempo real, identificar padrões complexos e integrar esses resultados em sistemas de inspeção reduz significativamente o tempo e o custo de detecção de falhas em ambientes industriais. Além disso, a modularidade da *OpenCV* permite que desenvolvedores adaptem soluções específicas às necessidades de diferentes processos de produção.

A aplicação de algoritmos de detecção de bordas, como o Canny, facilita a identificação de contornos em superfícies irregulares, enquanto técnicas de segmentação por intensidade de pixels permitem isolar regiões de interesse, como bolhas, rachaduras ou sobreextrusão. Esses métodos são particularmente úteis para inspecionar peças com padrões texturais variados, garantindo maior precisão na identificação de anomalias.

Adicionalmente, o uso da *OpenCV* em conjunto com descritores de textura, como o HOG, amplia a capacidade de análise ao permitir a classificação de defeitos com base em características visuais. Essa integração torna possível não apenas detectar falhas, mas também categorizá-las, fornecendo informações valiosas para ajustes nos processos de fabricação.

Em síntese, a *OpenCV* é uma ferramenta essencial na detecção de defeitos em impressão 3D, fornecendo uma base robusta para análise visual e facilitando a implementação de sistemas inteligentes em ambientes produtivos. Sua eficiência, aliada à possibilidade de personalização, oferece soluções escaláveis e de alto desempenho, contribuindo diretamente para a melhoria da qualidade e redução de custos na indústria.

2.9. Objetivos

2.9.1. Objetivo Geral

Este estudo busca desenvolver um sistema para detectar e classificar defeitos em peças impressas em 3D. Serão empregadas técnicas de processamento de imagens e

aprendizado de máquina não supervisionado para automatizar a análise e aprimorar a detecção das falhas.

2.9.2. Objetivos Específicos

- Realizar uma revisão da literatura sobre o uso de aprendizado não supervisionado e técnicas de visão computacional para detecção de defeitos em processos de manufatura aditiva;
- Implementar uma análise exploratória de dados com intuito de extrair *features* e *insights* para elaboração de um algoritmo de clusterização das imagens;
- Desenvolver um algoritmo baseado em aprendizado não supervisionado para análise de imagens de peças fabricadas por impressão 3D, com ênfase na identificação e agrupamento de anomalias;
- Validar o algoritmo utilizando os grupos de dados obtidos das peças fabricadas, utilizando métodos de avaliação de modelos de clusterização, além disso usando métodos holísticos manuais para observar a relação dos grupos.

3. MATERIAIS E MÉTODOS

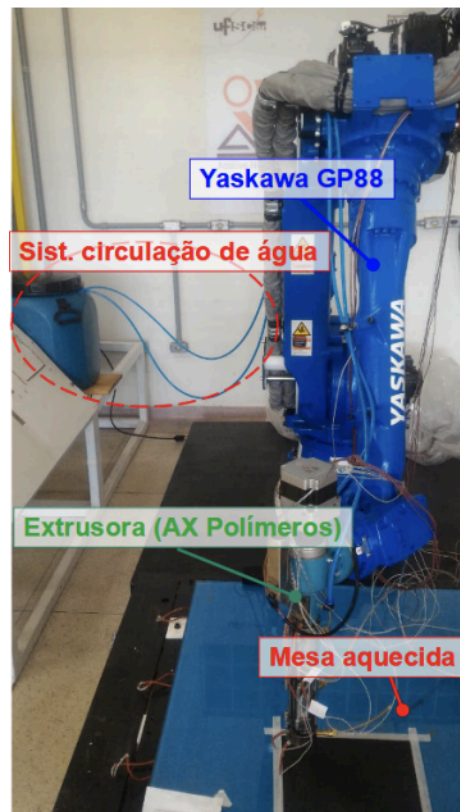
3.1. Descrição da célula de manufatura aditiva robotizada (CMAR) e imagens utilizadas

Para este estudo, foram analisadas imagens capturadas em um ambiente de manufatura aditiva robotizada (CMAR). Os experimentos foram conduzidos conforme descrito por Pulquerio, Barbosa e Shiki (2024). Essas imagens serviram como base para testar e validar o modelo proposto. A CMAR é composta por um manipulador robótico Yaskawa GP88, com capacidade de carga de 88 kg e 6 eixos de movimentação, responsável pelo manuseio de uma extrusora mono-rosca da fabricante AX Polímeros. A extrusora opera com rotação de até 60 RPM e é alimentada por pellets poliméricos.

Para garantir a qualidade da impressão, a célula conta com uma mesa aquecida de 1 m², composta por quatro chapas de aquecimento de 2 kW cada, que promovem maior aderência e uniformidade térmica da peça. Além disso, o sistema possui um circuito de resfriamento com bomba e reservatório de 50 litros para controle térmico da extrusora. A Figura 7 ilustra os subsistemas da CMAR.

(Importante: No presente estudo, não foram realizados experimentos diretamente na CMAR, mas sim utilizadas imagens capturadas nesse ambiente experimental em um trabalho de Iniciação Científica anterior.)

Figura 7 - Subsistemas da célula de manufatura aditiva robotizada.



Fonte:Próprio Autor

As peças foram modeladas no software SIEMENS NX, onde foram definidas suas geometrias tridimensionais. O fatiamento dos modelos foi realizado com o software Ultimaker Cura, que gerou o G-Code necessário para a impressão. Para permitir a integração com o manipulador robótico, esse código passou por um processo de conversão na plataforma RoboDK, gerando os comandos apropriados para a movimentação do robô durante a fabricação da peça.

As imagens utilizadas neste estudo foram adquiridas em um sistema de captura já implementado na pesquisa de Iniciação Científica. Esse sistema utilizou uma câmera Raspberry Pi 5MP (sensor OV5647, abertura f/1.8), acoplada a um computador Raspberry Pi 3 B+, que rodava um script desenvolvido em Python e OpenCV para captura das imagens.

O conjunto de imagens analisado abrangeu diferentes condições de superfície das peças fabricadas, incluindo casos de extrusão adequada, sobreextrusão e presença de

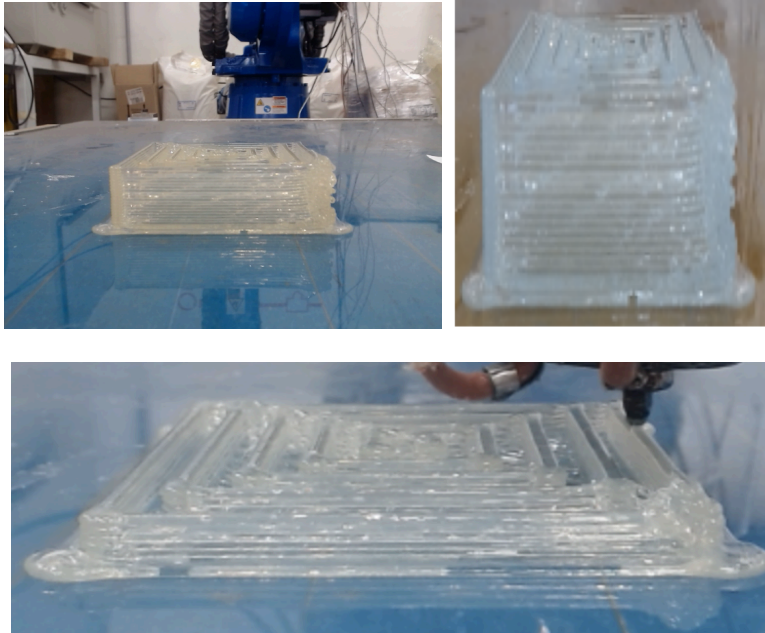
bolhas, permitindo a implementação e teste das técnicas de processamento e clusterização propostas neste trabalho.

3.2. Descrição das imagens utilizadas

O conjunto de dados utilizado neste trabalho foi composto por cinco grupos principais, representando diferentes condições de superfícies industriais fabricadas por meio de manufatura aditiva. Cada corpo de prova foi gerado de forma criteriosa para abranger variações de padrões texturais e defeitos, permitindo uma análise. No total foram utilizadas 72 imagens selecionadas de cada um dos 5 grupos, estas 72 imagens descrevem bem os grupos ao qual estão inclusas facilitando o algoritmo captar os padrões de agrupamento para posteriormente aplicar em um modelo de validação, após a realização da modelagem utilizaram-se 455 imagens (todas as imagens do repositório) para avaliar a classificação do modelo. A seguir, descrevem-se as características de cada imagem de acordo com nome:

A Figura 8 apresenta uma superfície ideal, sem falhas aparentes, com textura uniforme. Essa imagem foi utilizada como referência para definir os padrões desejáveis de manufatura, servindo como base para comparações com outras categorias de superfícies analisadas. Para facilitar a análise e a detecção de defeitos, a imagem foi previamente cortada (entre as coordenadas 200 a 350 e 120 a 535 pixels) e redimensionada para 100x100 pixels. Esse processo garante uniformidade nas dimensões das imagens, otimizando a visualização e a consistência no tratamento posterior. O procedimento foi repetido para as Figuras 9, 10, 11 e 12

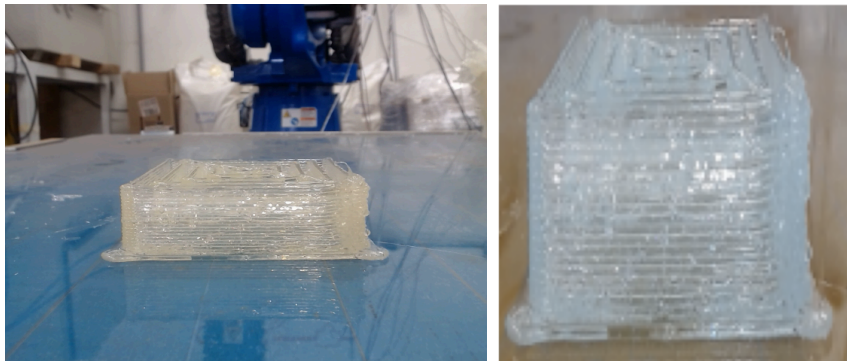
Figura 8 - Imagem “Normal”

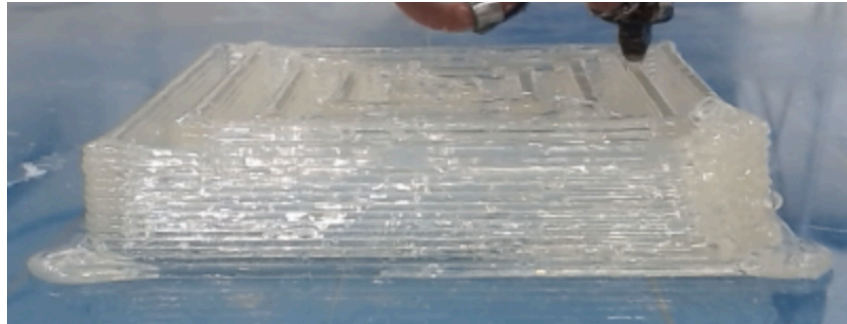


Fonte: Próprio Autor

A Figura 9 apresenta irregularidades na forma de bolhas visíveis. Essas características texturais marcantes facilitaram a segmentação por intensidade de pixels e permitiram identificar padrões específicos associados a defeitos localizados.

Figura 9 - Imagem “Bolhas”

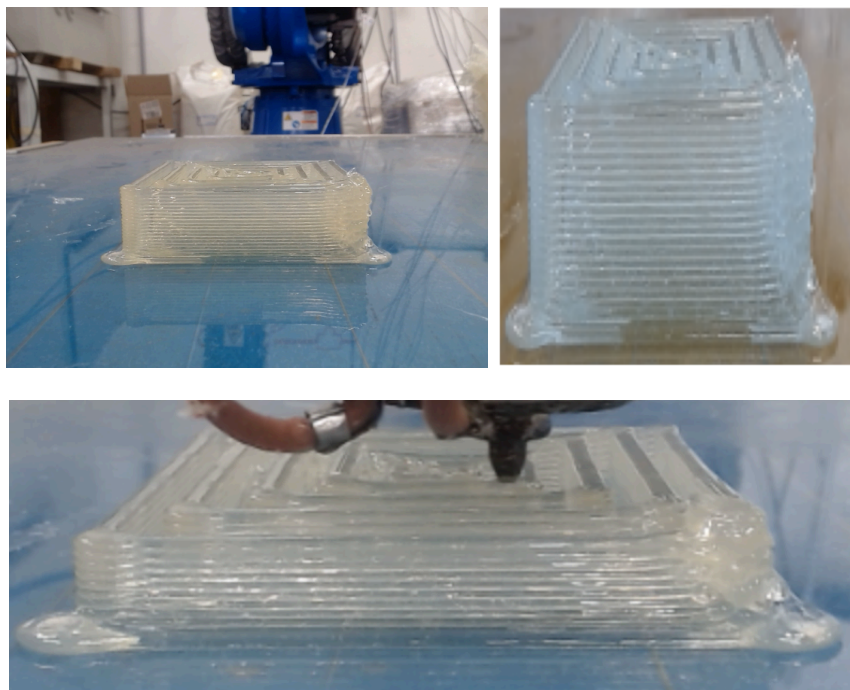




Fonte: Próprio Autor

A Figura 10 apresenta problemas de sobreextrusão em níveis baixos, com pequenos excessos de material nas camadas, e moderados, onde saliências mais evidentes comprometem a regularidade da superfície.

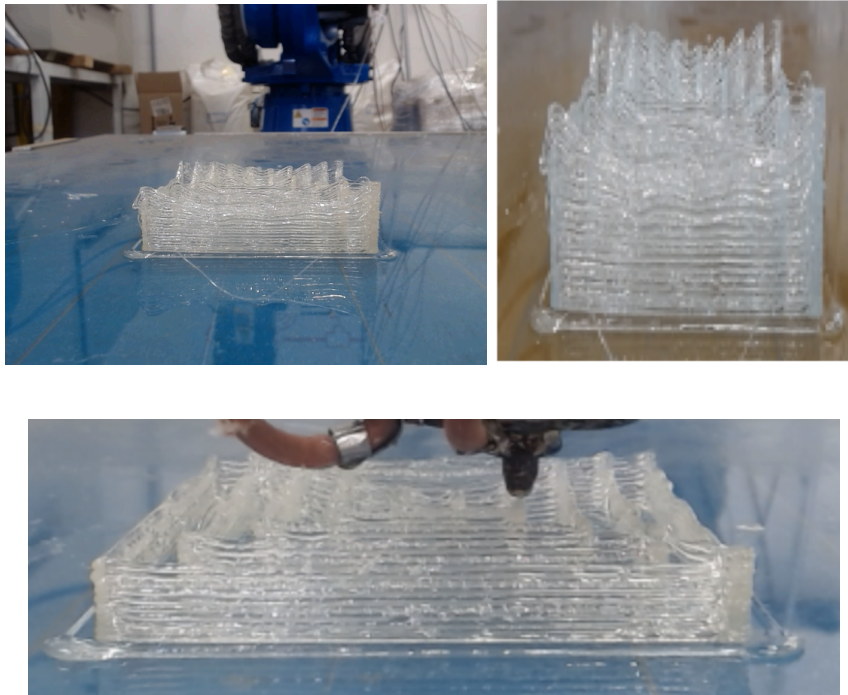
Figura 10 - Imagem “Overextrusion40”



Fonte: Próprio Autor

A Figura 11 contém problemas significativos de sobreextrusão, com bordas acentuadas e a formação de texturas irregulares. Este tipo de imagem foi particularmente útil para treinar o modelo a identificar defeitos críticos na superfície, garantindo maior precisão na classificação.

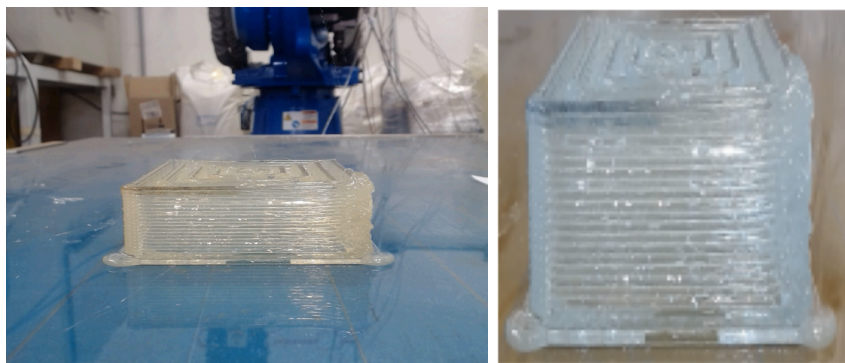
Figura 11 - Imagem “Overextrusion10”

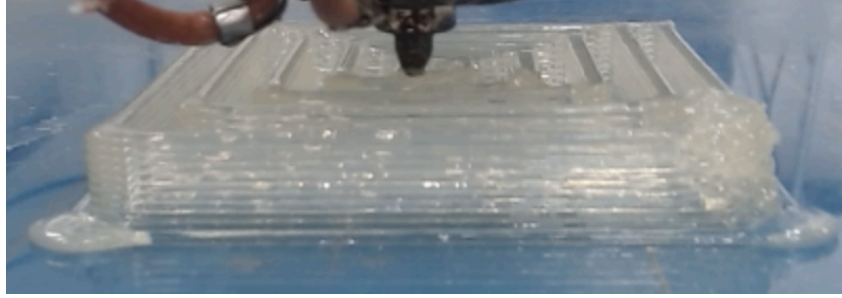


Fonte: Próprio Autor

Por fim, a Figura 12 reúne características combinadas, incluindo bolhas e áreas com sobreextrusão. Essa composição tornou-se ideal para testar a capacidade do *pipeline* em lidar com padrões mistos e mais complexos, proporcionando um cenário desafiador para validação dos modelos.

Figura 12 - Imagem “Overextrusion”





Fonte: Próprio Autor

Essas imagens foram submetidas a um processo de pré-processamento, incluindo normalização e redimensionamento, para assegurar consistência no *pipeline* de análise. Além disso, a diversidade textural presente no conjunto foi amplificada por meio de *data augmentation*, gerando variações que enriqueceram o conjunto de dados e garantiram maior robustez nos modelos desenvolvidos.

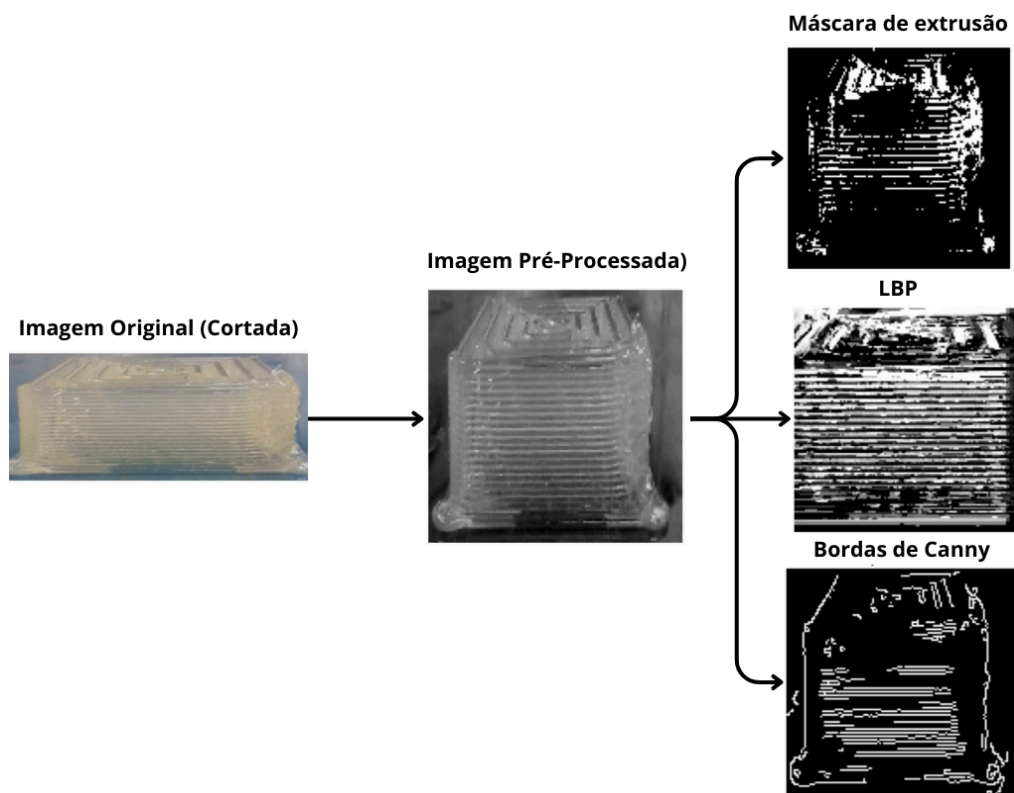
3.3. Análise exploratória de dados

Antes da aplicação dos métodos de clusterização e aumento de dados, foi realizada uma análise exploratória para entender as características das imagens e suas respectivas *features*. Essa etapa é fundamental para avaliar a distribuição dos atributos, identificar padrões iniciais e detectar possíveis anomalias nos dados.

Na análise das imagens originais, foram calculadas métricas estatísticas como médias, variâncias e desvios-padrão para os atributos derivados das *features*. Entre as métricas analisadas estavam a média das intensidades das cores nos canais RGB, que fornece uma visão geral sobre a iluminação e tonalidade predominante nas imagens, bem como informações sobre a uniformidade de suas texturas.

A análise visual das imagens, como ilustrado na Figura 14, é crucial em algoritmos de clusterização, pois permite compreender como as diferentes técnicas de pré-processamento (como máscaras de extrusão, LBP e detecção de bordas de Canny) extraem características específicas das imagens originais. Essa etapa é fundamental para identificar padrões relevantes e avaliar se as features extraídas destacam efetivamente as diferenças entre os tipos de imagens ou defeitos analisados. Essa compreensão qualitativa auxilia no ajuste de parâmetros e na validação do desempenho do modelo, garantindo que a clusterização seja baseada em informações significativas e visualmente interpretáveis.

Figura 14 - Demonstração de como as *features* foram observadas



Fonte: Próprio Autor

Essa investigação inicial forneceu uma base sólida para a definição dos parâmetros das etapas subsequentes, como a clusterização e a extração de características. Os resultados da análise exploratória permitiram adaptar os métodos às características específicas das imagens estudadas, garantindo maior precisão e robustez no *pipeline* de análise.

3.4. Implementação do código, métricas e avaliação do modelo

Nesta etapa, foi desenvolvido um algoritmo computacional responsável pela clusterização não supervisionada de imagens visando a detecção de defeitos em peças produzidas por manufatura aditiva. Além disso, duas métricas foram estabelecidas para avaliar a qualidade dos agrupamentos obtidos.

3.4.1. Algoritmo Implementado

O algoritmo em Python foi desenvolvido a partir das seguintes bibliotecas:

- *skimage*: responsável pela implementação do LBP e Hog;
- *OpenCV*: utilizada para leitura, redimensionamento, pré-processamento das imagens, implementação da detecção de bordas de Canny e também para quantização dos *overextrusions* e também transformação para escala de cinza;
- *Pandas*, *Seaborn* e *Matplotlib*: utilizadas para manipulação de dados e visualização dos resultados;
- *Scikit-Image*: responsável pela implementação dos descritores de *features* como HOG e LBP;
- *Scikit-Learn*: encarregada dos algoritmos de aprendizado de máquina como PCA e *Kmeans*, além de métricas de avaliação como Silhouette Score;
- *Scipy*: Responsável pelo cálculo do *Z-score* durante a etapa de análise de dados;
- *Numpy*: utilizada para realizar operações matemáticas em matrizes;

O funcionamento do algoritmo pode ser descrito pelas seguintes etapas:

1. Definição dos parâmetros de configuração, como o uso ou não de determinados descritores (HOG, LBP, Canny), seus valores específicos, dimensões para redimensionamento das imagens, número de componentes do PCA e quantidade de *clusters* do *Kmeans*.
2. Implementação das funções para extração de *features* das imagens de acordo com os parâmetros especificados. Cada imagem é processada por

uma função, que aplica os descritores selecionados e retorna um vetor de características.

3. Divisão do conjunto total de imagens nos subconjuntos de treino, validação e teste para posterior avaliação do modelo.
4. Extração das *features* das imagens de treino, seguida de escalonamento e redução de dimensionalidade com PCA.
5. Aplicação do algoritmo de clusterização *Kmeans* nos dados transformados para obtenção dos agrupamentos.
6. Visualização dos resultados por meio de gráficos de dispersão e gráficos de barras empilhadas, evidenciando a distribuição das imagens por *cluster* e por nome base dos arquivos.
7. Validação da qualidade dos agrupamentos obtidos utilizando o método do cotovelo (do inglês, *Elbow Method*) e a métrica *Silhouette Score*.

3.4.2. Métricas para Avaliação do Modelo

Para avaliar o desempenho do modelo de clusterização não supervisionada, foram utilizadas duas métricas principais: o *Elbow Method* e o *Silhouette Score*.

3.4.2.1. *Elbow Method*

O *Elbow Method* é uma técnica amplamente utilizada para determinar o número ideal de *clusters* em um conjunto de dados, sendo particularmente aplicado em algoritmos de clusterização como o *Kmeans*. Ele baseia-se na premissa de que, ao aumentar o número de *clusters*, a inércia (definida como a soma das distâncias ao quadrado entre cada ponto e o centro de seu *cluster*) diminui. Entretanto, após um certo ponto, o ganho obtido pela adição de novos *clusters* torna-se marginal, caracterizando os retornos decrescentes. Na equação de número 6 é mostrado como a inércia é calculada:

$$Inércia = \sum_{k=1}^K \sum_{i \in C_k} ||x_i - u_k||^2 \quad (7)$$

Onde,

k : Representa um cluster específico (de 1 até K, que é o número total de clusters).

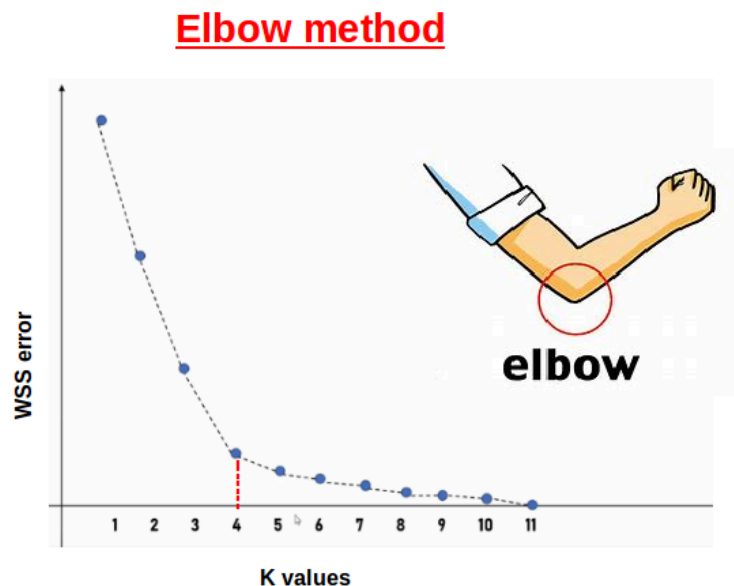
C_k : É o conjunto de pontos que pertencem ao cluster k .

x_i : É um ponto de dados específicos que pertence ao cluster k .

u_k : É a média das coordenadas de todos os pontos dentro do cluster k .

A aplicação do método do cotovelo envolve o cálculo da inércia para diferentes valores de número de *clusters*. Esses valores são então ilustrados em um gráfico que relaciona à inércia. A curva resultante geralmente apresenta uma queda acentuada nos primeiros valores de K (número de *clusters*), seguida por uma estabilização progressiva. O "cotovelo" da curva representa o ponto de inflexão onde a redução na inércia se torna menos significativa. Este ponto é considerado o número ideal de *clusters*, pois equilibra a simplicidade do modelo e a qualidade do agrupamento.

Figura 15 - Exemplo de uso do Método do Cotovelo



Fonte: Medium (2023)

Essa abordagem é altamente visual e heurística, permitindo que analistas interpretem os resultados com base na geometria da curva. No entanto, é importante notar que o método do cotovelo pode nem sempre fornecer um ponto de inflexão claro, especialmente em dados de alta dimensionalidade ou com distribuição complexa. Nesses casos, técnicas complementares, como o *Silhouette Score*, podem ser utilizadas para corroborar a escolha do número de *clusters*.

Em resumo, o método do cotovelo oferece uma maneira intuitiva e prática de selecionar , sendo uma ferramenta essencial para a validação de um modelo em cenários de clusterização.

3.4.2.2. *Silhouette Score*

O *Silhouette Score* avalia a separação entre clusters, medindo o grau de similaridade de cada ponto com os demais do mesmo grupo em comparação com outros *clusters*. Para cada ponto, calcula-se:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (8)$$

Onde,

$s(i)$: é o *Silhouette Score*;

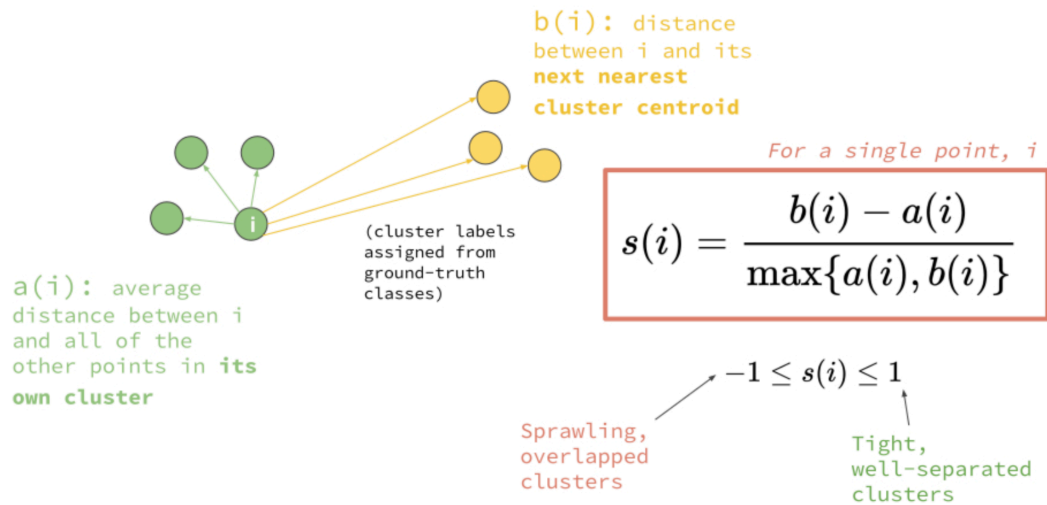
$a(i)$: é a distância média do ponto i a todos os outros pontos do mesmo *cluster*;

$b(i)$: [e a distância média do ponto i ao *cluster* mais próximo que não seja o seu;

$\max\{a(i), b(i)\}$: denota o valor máximo entre $a(i)$ e $b(i)$;

Essa métrica varia de -1 a 1, sendo que valores mais próximos de 1 indicam que a amostra está bem atribuída ao seu *cluster*, valores próximos de 0 sugerem que a amostra está no limite entre dois *clusters* e valores negativos indicam que a amostra provavelmente foi atribuída ao *cluster* errado. Na Figura 16, é possível observar uma representação da equação do cálculo do *Silhouette Score*.

Figura 16 - Representação do *Silhouette Score*



Fonte: Medium (2021)

A média dos *Silhouette Scores* de todas as amostras fornece uma medida geral da qualidade da clusterização. Quanto maior o valor médio, melhor a separação e coesão dos *clusters* encontrados.

Baseado na publicação seminal de Rousseeuw (1987), o *Silhouette Score* apresenta uma escala interpretativa que qualifica a qualidade dos agrupamentos. Valores próximos a 1 indicam uma estrutura de *clustering* robusta, onde os pontos dentro de cada *cluster* são altamente similares entre si e distintivamente diferentes dos pontos em outros *clusters*. Em contrapartida, valores próximos a 0 sugerem uma sobreposição significativa ou definição imprecisa dos clusters, enquanto valores negativos podem indicar uma potencial atribuição incorreta de pontos aos agrupamentos. Convencionalmente, escores acima de 0,7 são considerados excelentes, entre 0,5 e 0,7 são interpretados como bons, e valores entre 0,3 e 0,5 representam agrupamentos razoáveis, fornecendo assim um critério quantitativo para avaliar a coesão e separação dos *clusters* (Rousseeuw, 1987).

3.4.3. Avaliação do Modelo

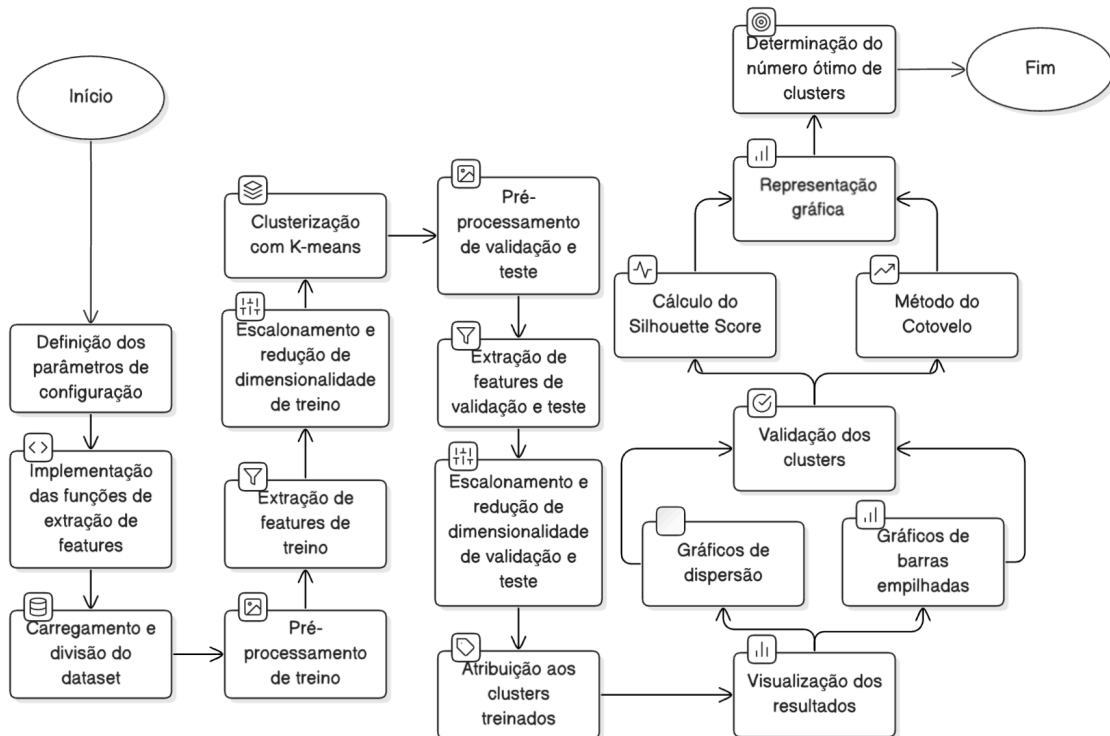
Para avaliar o modelo de clusterização, foi utilizada a, que aplica o método do cotovelo e calcula o *Silhouette Score* para diferentes valores de K. Os resultados são representados em gráficos para análise visual.

Além disso, a distribuição dos arquivos por *cluster* e por nome base é visualizada por meio de gráficos de dispersão e de barras empilhadas para os conjuntos de treino, validação e teste. Isso permite verificar a consistência dos agrupamentos encontrados em diferentes partições dos dados.

Por fim, o número ótimo de *clusters* é determinado com base no valor máximo do *Silhouette Score* médio, buscando obter agrupamentos bem definidos e separados.

O fluxograma presente na Figura 17 resume as etapas da implementação do código e avaliação do modelo descritas nesta seção.

Figura 17 - Fluxograma de funcionamento do algoritmo



Fonte: Próprio Autor

4. RESULTADOS E DISCUSSÃO

Os resultados da análise das imagens trouxeram insights valiosos sobre os padrões de defeitos presentes nas peças. A aplicação do modelo de clusterização permitiu separar diferentes tipos de falhas, facilitando a identificação de irregularidades na impressão. Esta seção apresenta uma discussão detalhada baseada nas visualizações, métricas e interpretações obtidas, estruturada em subtópicos para maior clareza.

4.1. Análise Exploratória de Dados

A Análise Exploratória de Dados (EDA, do inglês *Exploratory Data Analysis*) têm um papel fundamental na identificação de padrões, tendências e possíveis anomalias nas imagens analisadas. A partir da extração de *features* como detecção de bordas (Canny), análise de textura (LBP), intensidade de extrusão e componentes principais (PCA), foram geradas métricas estatísticas que permitiram uma melhor compreensão da estrutura dos dados.

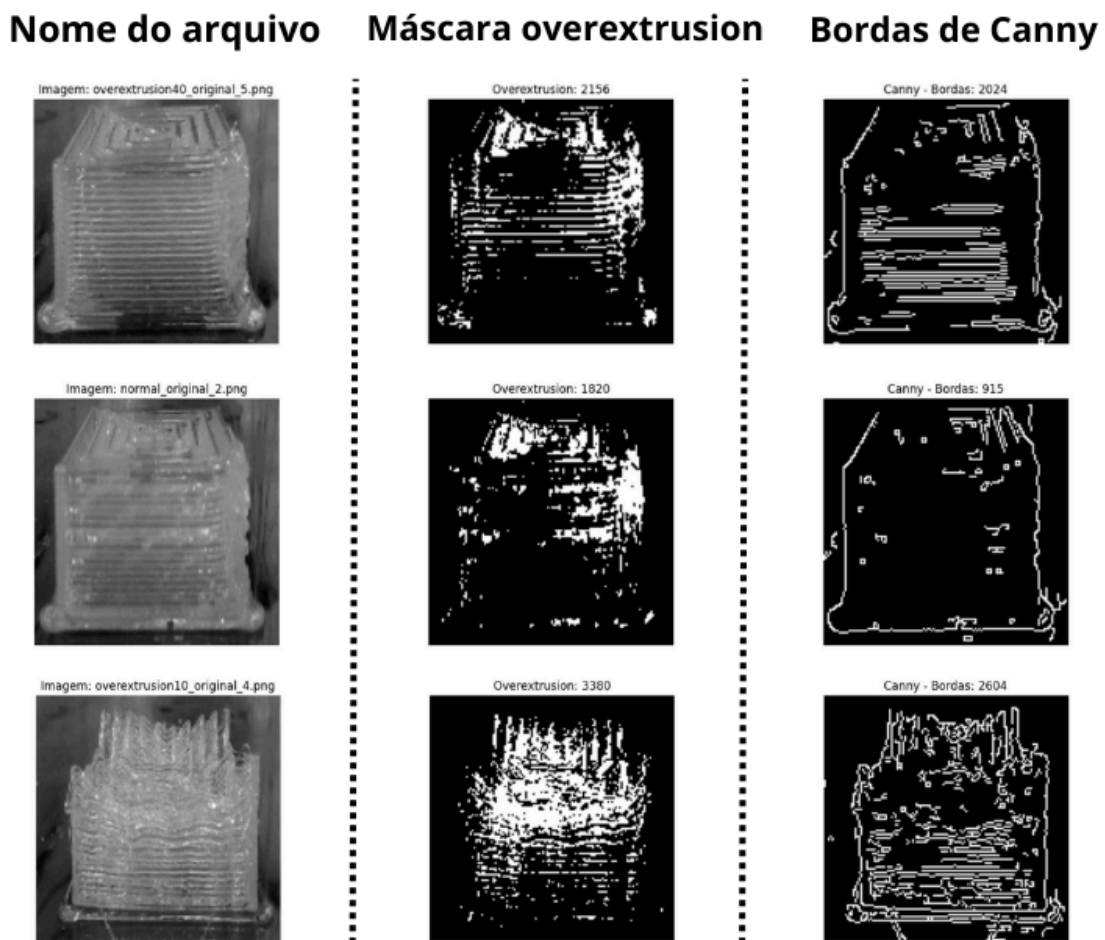
4.1.1. Pré-processamento, Observações Visuais e Distribuição de *Features*

O pré-processamento das imagens incluiu a conversão para escala de cinza, segmentação e realce das bordas, visando facilitar a análise visual e a extração de características. A Figura 18 apresenta um conjunto de imagens processadas, onde é possível observar a segmentação realizada por técnicas de *OpenCV* e a extração de bordas por Canny. Essas transformações permitiram destacar diferenças estruturais entre superfícies normais e defeituosas.

A observação direta dessas imagens evidenciou que defeitos como bolhas e sobreextrusão possuem padrões distintos que podem ser melhor identificados por métodos de segmentação. A segmentação baseada na intensidade de pixels destacou regiões críticas, enquanto a detecção de bordas evidenciou falhas estruturais mais acentuadas. Essa análise visual foi fundamental para confirmar que as *features* extraídas capturavam informações relevantes para a clusterização e diferenciação das superfícies.

A Figura 18 apresenta a análise visual das imagens utilizando *OpenCV*, destacando como as técnicas de pré-processamento, segmentação e detecção de bordas realçam padrões distintos entre superfícies normais e defeituosas. É possível observar que regiões com defeitos, como bolhas e sobreextrusão, apresentam contrastes mais acentuados e contornos irregulares, facilitando sua identificação. Essa análise visual foi essencial para validar a eficácia das features extraídas e fundamentar a escolha dos parâmetros utilizados na clusterização..

Figura 18 - Representação das imagens sob a vista do *OpenCV*



Fonte: Próprio Autor

4.1.2. Análise das Médias e Desvio Padrão das Features

Para compreender a distribuição das características extraídas, foram analisadas as médias e os desvios padrão das principais features utilizadas na clusterização. Essas métricas foram calculadas para todas as features extraídas, incluindo bordas de Canny, a máscara de extrusão e o HOG, permitindo avaliar a estabilidade das características e sua capacidade de diferenciar superfícies normais e defeituosas.

As imagens foram classificadas em diferentes condições, cada uma representando um estado específico da superfície analisada. Essas classes refletem defeitos comuns encontrados na impressão 3D e ajudam a avaliar como as features extraídas podem diferenciar entre superfícies normais e defeituosas.

A classe “Normal” representa superfícies sem defeitos aparentes, servindo como referência para comparação, enquanto a classe “Bolhas” inclui superfícies com pequenas irregularidades ou cavidades que alteram a textura e a detecção de bordas. Já a “Overextrusion”, dividida em níveis de acordo com a rotação de rosca pela qual foram gerados Overextrusion (sendo este a rotação de rosca padrão utilizada como 30), “Overextrusion10” e “Overextrusion40”, caracteriza-se pelo excesso de material depositado, resultando em padrões irregulares de bordas, texturas mais rugosas e maior detecção na máscara de extrusão. Essas diferenças impactam diretamente as features extraídas, como Canny Edges, que identifica discontinuidades, Extrusion Mask, que mede o acúmulo de material, e HOG, que analisa a textura em detalhes dos filamentos e também das superfícies, permitindo uma distinção quantitativa entre os diferentes tipos de defeitos.

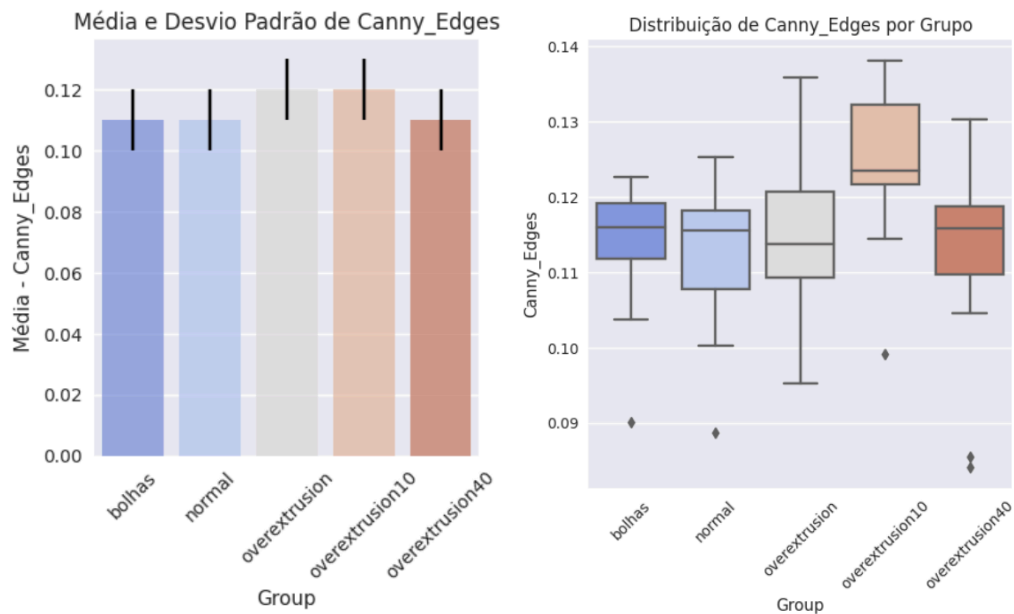
A análise das médias revelou que imagens contendo sobreextrusão apresentaram valores de *features* significativamente mais elevados em comparação às imagens normais, confirmando uma maior presença de variações estruturais capturadas pelos métodos de segmentação e detecção de bordas. No entanto, a magnitude dessas diferenças variou dependendo da técnica utilizada para extração das features. A Figura 19 (Canny) mostrou que as bordas nas imagens com sobreextrusão apresentaram maior intensidade e variabilidade, enquanto as normais mantiveram um padrão mais homogêneo. Já a Figura 20 (máscara de extrusão) indicou que as áreas de extrusão

excessiva possuíam uma distribuição menos uniforme de intensidade, reforçando a utilidade dessa abordagem para diferenciar superfícies defeituosas. Por fim, a Figura 21 (HOG) aparenta não ter impacto para a classificação do algoritmo, mas visto que o HOG utiliza uma análise mais profunda dos pixels da cada imagem ele será analisado de forma individual para entender sua correlação com os grupos.

O desvio padrão das *features* também demonstrou um papel fundamental na diferenciação das classes. Enquanto imagens normais exibiram um comportamento mais consistente, superfícies com defeitos apresentaram maiores variações. Em particular, as imagens classificadas como "Overextrusion10" exibiram o maior desvio padrão entre todas as categorias, evidenciando a inconsistência desse tipo de falha. Já as imagens "Overextrusion40" tiveram uma distribuição intermediária, reforçando que esses defeitos podem ocorrer em intensidades distintas e serem reconhecidos com base nos padrões de suas *features*.

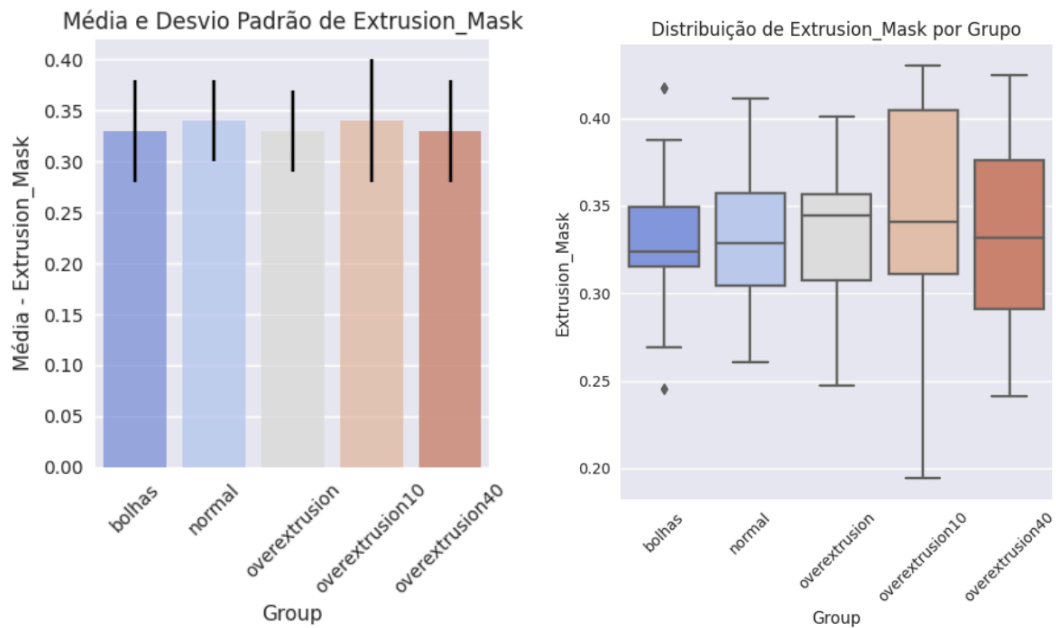
As Figuras 19, 20 e 21 ilustram os valores médios e os desvios padrões das principais *features* extraídas, validando a eficácia das técnicas utilizadas para a diferenciação dos grupos de imagens. Esses resultados confirmam que a escolha das características foi adequada para capturar as diferenças entre os tipos de superfície analisados, reforçando a importância da análise estatística na seleção das melhores *features* para o modelo de clusterização.

Figura 19 - Média, desvio padrão e distribuição das features por grupo de imagem (Canny)



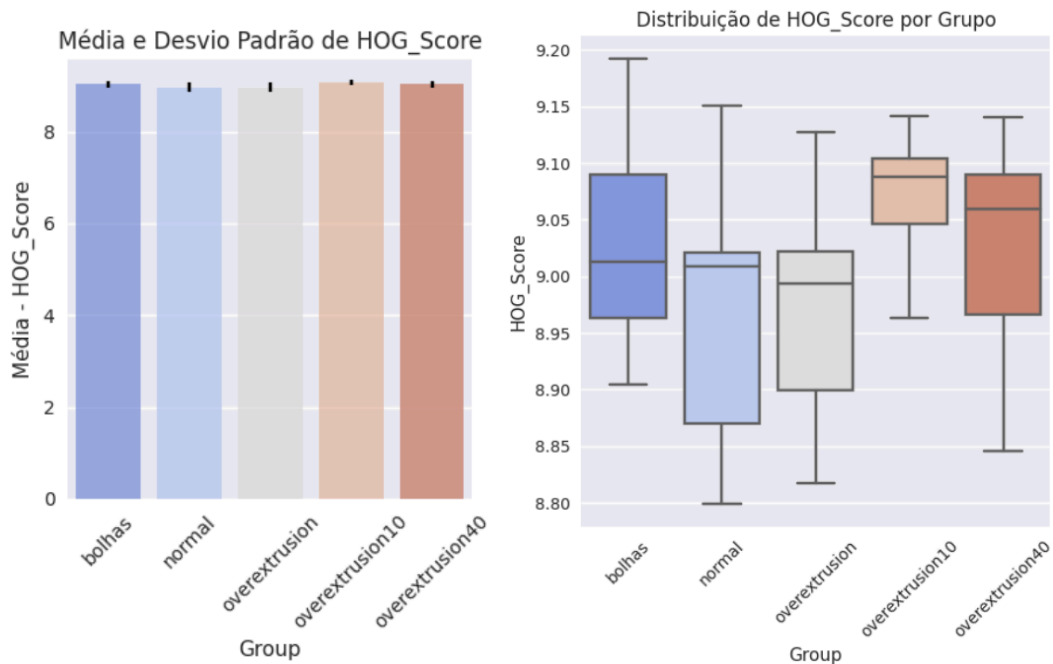
Fonte: Próprio Autor

Figura 20 - Média, desvio padrão e distribuição das features por grupo de imagem (Mascara de pixels)



Fonte: Próprio Autor

Figura 21 - Média, desvio padrão e distribuição das features por grupo de imagem (HOG)



Fonte: Próprio Autor

Apesar do baixo desvio padrão da feature HOG, indicando pouca dispersão dos valores ao longo do conjunto de dados na Figura 22, essa métrica demonstrou ser a mais relevante para a diferenciação das classes. A análise dos agrupamentos revelou que o HOG apresentou a maior correlação entre os grupos, capturando com maior precisão as diferenças estruturais entre as imagens. Essa característica permitiu uma separação mais consistente e eficiente das categorias de defeitos, tornando-se o principal componente para a clusterização. Dessa forma, mesmo com uma variabilidade relativamente baixa, o HOG destacou padrões que foram fundamentais para o modelo preditivo, reforçando sua importância na segmentação dos grupos de defeitos que será destacado na seção 5.1.3.

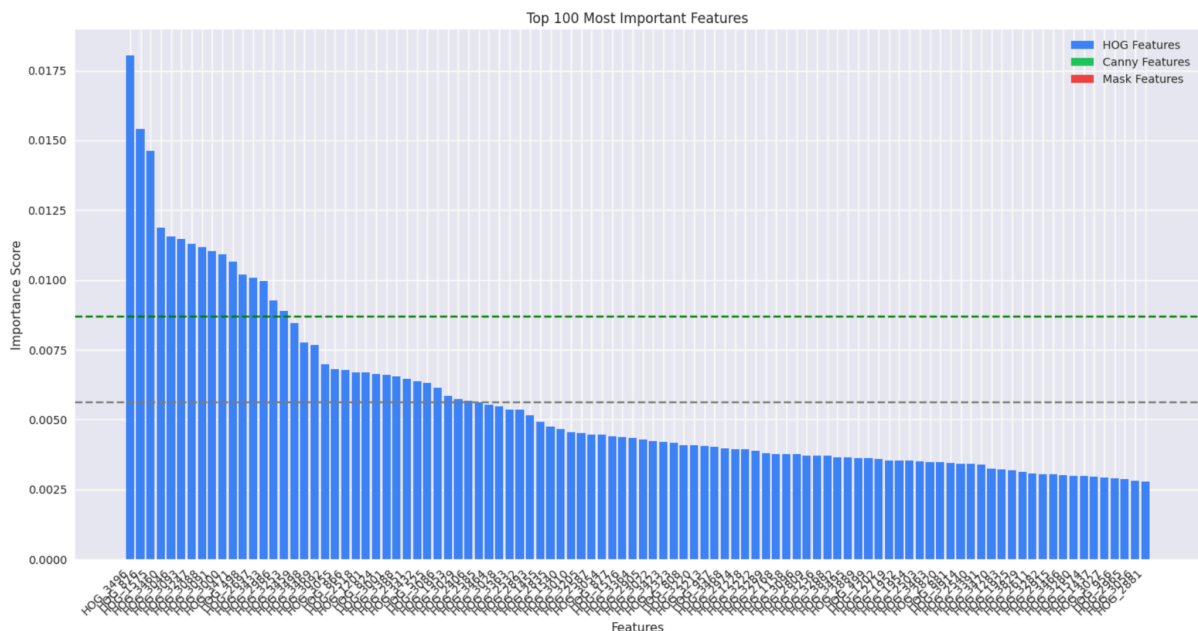
4.1.3. Análise do HOG

A análise das features revelou a importância do algoritmo HOG em comparação com outras técnicas, como Canny e análise de intensidade de pixels. O HOG é amplamente utilizado como um descritor robusto para detecção de objetos estáticos,

pois captura eficientemente padrões de gradientes locais e suas distribuições. Enquanto o Canny se concentra apenas na detecção de bordas e a análise de intensidade de pixels considera apenas valores brutos de luminosidade, o HOG fornece uma representação mais detalhada das texturas e variações estruturais presentes nas imagens.

Para avaliar a contribuição de cada feature na clusterização, foi utilizada uma abordagem baseada em feature importance, empregando um modelo de classificação Random Forest para identificar as features mais discriminativas. A análise obtida demonstra a predominância das features derivadas do HOG, que apresentaram os maiores escores de importância, como pode ser observado na Figura 22. Esse resultado reforça que o HOG foi a feature mais relevante para a separação dos grupos, pois conseguiu capturar de forma consistente as diferenças estruturais entre as classes analisadas. A robustez do HOG para representar padrões de textura e gradientes fez com que ele se destacasse como o componente fundamental na clusterização, garantindo uma segmentação mais eficiente e precisa.

Figura 22 - 100 features mais relevantes para diferenciação



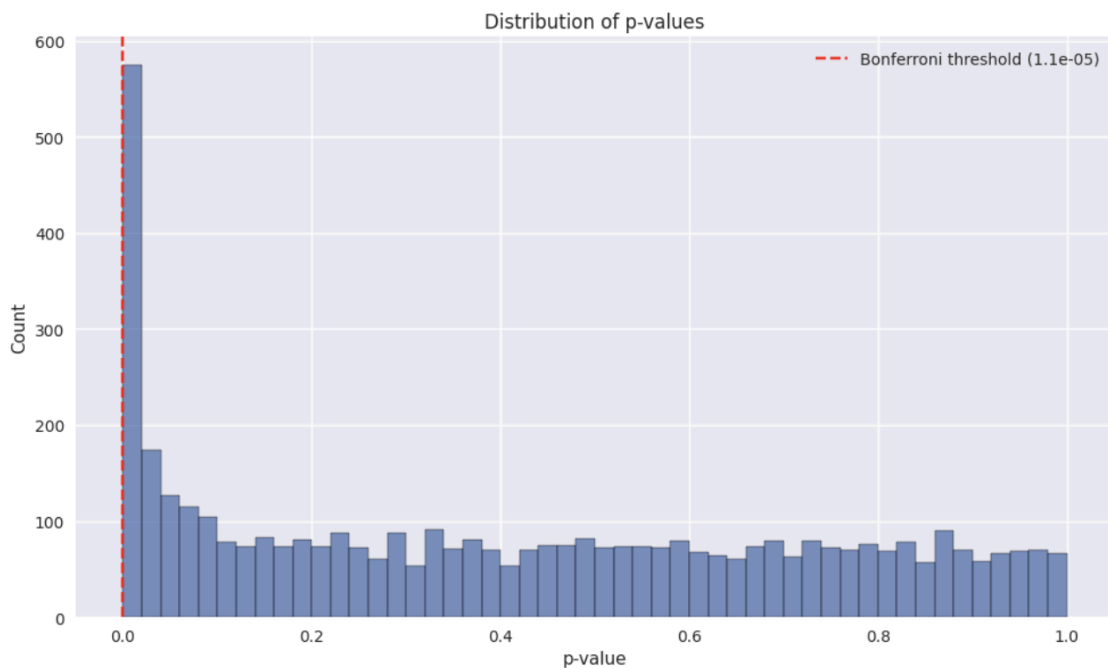
Fonte: Próprio Autor

A análise de variância (ANOVA) foi utilizada para verificar se as features extraídas pelo descritor HOG possuem capacidade significativa de diferenciar entre as

classes de defeitos em impressão 3D. Como podemos observar na Figura 23, a distribuição dos p-values mostra uma concentração expressiva de features com valores próximos a zero, significativamente abaixo do limiar de Bonferroni ($1.1e^{-5}$), que é um critério conservador para controle de falsos positivos em múltiplos testes estatísticos.

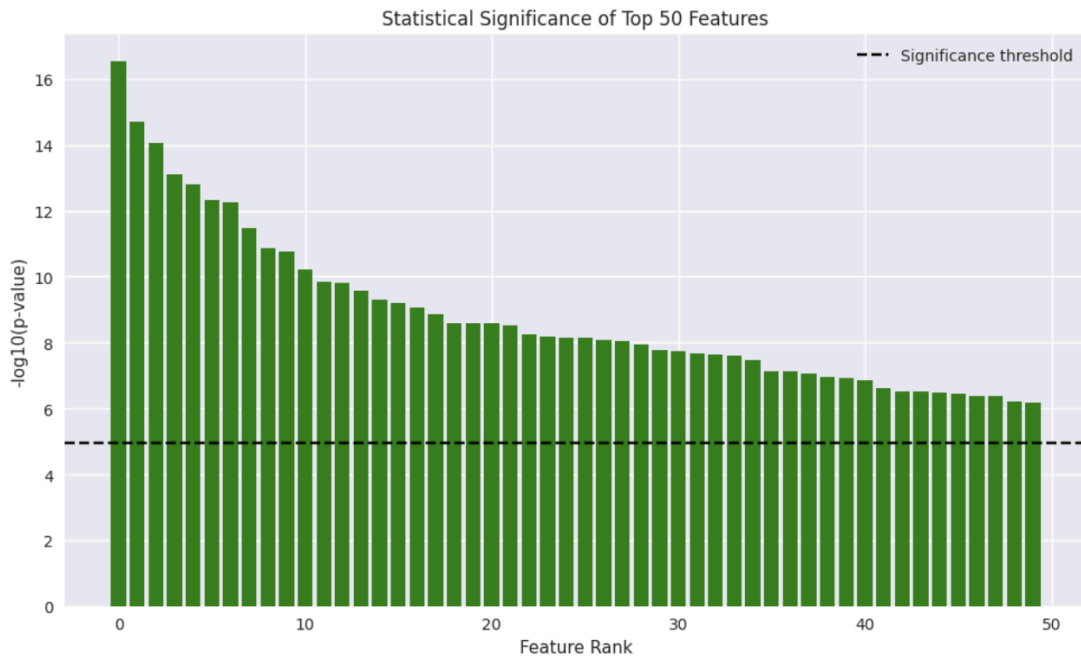
Para ter outro critério de validação da significância das features selecionadas, a Figura 24 apresenta o $-\log_{10}$ dos p-values para as 50 features mais relevantes, onde todas superam o limiar de significância estatística (representado pela linha tracejada). Os valores variam de aproximadamente 16 a 6 na escala logarítmica, correspondendo a p-values da ordem de 10^{-16} a 10^{-6} , substancialmente menores que o nível de significância convencional de 0.05. Estes resultados fornecem forte evidência estatística de que as features HOG selecionadas possuem real poder discriminativo entre as diferentes classes de defeitos, não sendo resultado de variações aleatórias nos dados. Tal validação estatística é crucial para garantir a robustez e confiabilidade do método proposto para identificação de defeitos em impressão 3D.

Figura 23 - Distribuição dos *p-values* nas *features*



Fonte: Próprio Autor

Figura 24 - *p-values* das *features* principais



Fonte: Próprio autor

4.2. Modelo de Validação e Resultados

4.2.1. Determinação do Número de *Clusters*

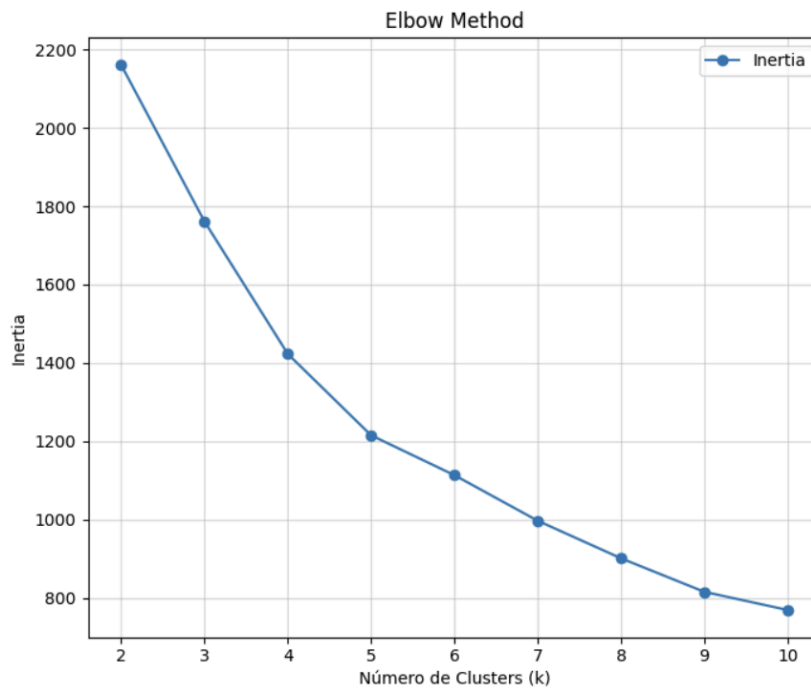
A escolha do número de clusters foi feita com base em hipóteses formuladas a partir da análise visual e qualitativa das imagens. Durante a EDA, foi observada a seguinte distribuição de características:

- Overextrusion10: Este grupo apresentou má formação perceptível a olho nu, sendo esperado que estivesse isolado em um único *cluster* devido à sua natureza distinta.
- Normal: Como não possuía defeitos aparentes, foi hipotetizado que este grupo também formaria um *cluster* isolado, representando superfícies ideais.
- Bolha e Overextrusion: Este tipo de defeito foi claramente visível, com texturas distintas que sugeriram a formação de um *cluster* específico.
- Overextrusion40: Embora apresentasse problemas de sobreextrusão, esses defeitos não eram visíveis a olho nu e foram identificados apenas durante a

EDA, indicando um padrão mais sutil que poderia ser agrupado de forma diferenciada.

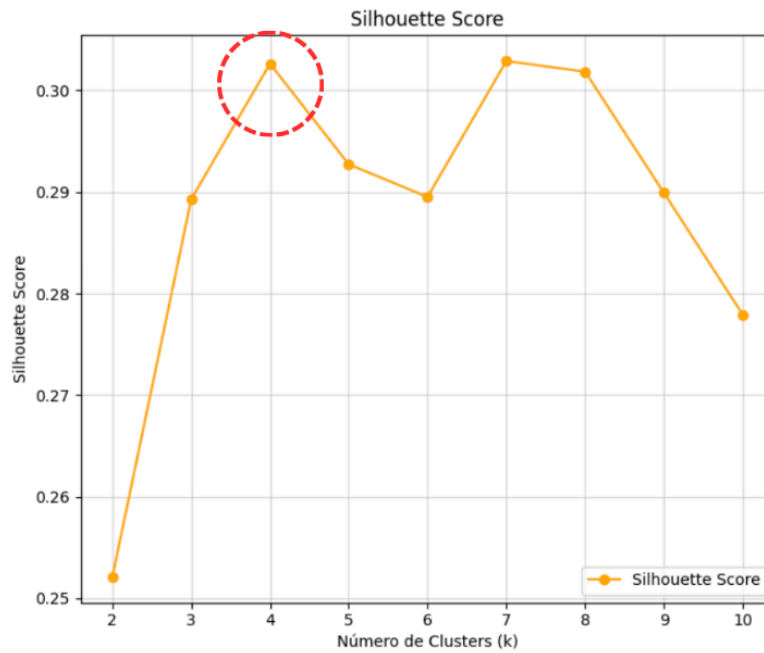
A decisão de utilizar quatro *clusters* foi validada quantitativamente por meio do *Elbow Method* e do *Silhouette Score*. A Figura 25 apresenta os resultados do *Elbow Method*, onde a inflexão no gráfico sugere o valor entre 4 e 5 *clusters*. Já a Figura 26 exibe o resultado obtido no *Silhouette Score*.

Figura 25 - Método do cotovelo obtido para 4 *clusters*



Fonte: Próprio Autor

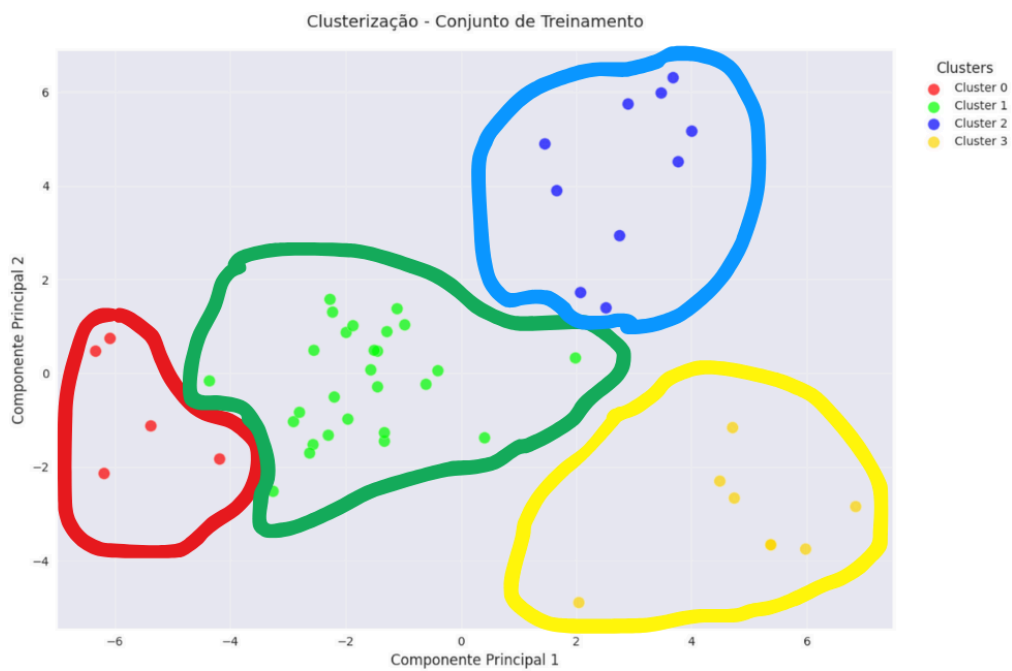
Figura 26 - *Silhouette Score* obtido



Fonte: Próprio Autor

A análise dos *Scatter Plots* no espaço PCA (Figura 27) reforça a separação entre os *clusters* e confirma a robustez das hipóteses formuladas durante a inspeção visual.

Figura 27 - Resultado nos dados de treino



Fonte: Próprio Autor

4.2.2. Caracterização dos *Clusters*

Os *clusters* formados apresentaram as seguintes características detalhadas:

Cluster 0: Inclui imagens com sobreextrusão severa (*overextrusion10*), marcadas por bordas acentuadas e texturas irregulares, tendo a superfície mais afetada pela sobreextrusão dentro dos grupos.

Cluster 1: Apresenta imagens com defeitos de sobreextrusão de todos os tipos tanto alto quanto baixo grau, correspondendo a defeitos em alguns casos não aparente (visualmente), mas ainda assim prejudicando a peça.

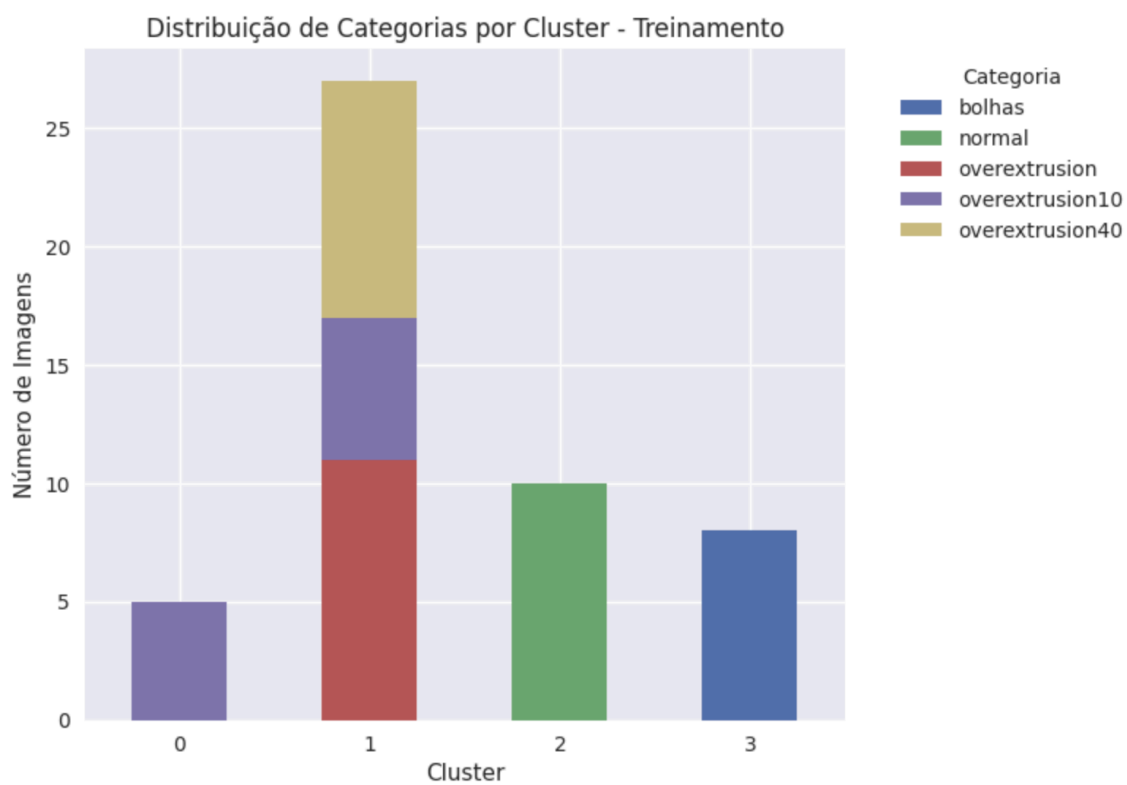
Cluster 2: Representa imagens sem defeitos aparentes, correspondendo a superfícies normais.

Cluster 3: Contém imagens com bolhas principalmente e outras irregularidades, caracterizando superfícies com alta variação na textura.

Os clusters formados através da análise HOG revelam padrões interessantes na distribuição das diferentes categorias de extrusão. O cluster 1 concentra a maior diversidade de casos, incluindo *overextrusion*, *overextrusion10* e *overextrusion40*, sugerindo que o HOG detectou padrões de gradientes similares entre diferentes níveis de sobreextrusão. O cluster 2 é predominantemente composto por peças normais, indicando que o HOG foi eficaz em identificar o padrão de gradientes característicos de peças sem defeitos. O cluster 3 agrupa especificamente as peças com bolhas, demonstrando que os gradientes orientados capturaram eficientemente as bordas circulares e texturas específicas deste tipo de defeito. O cluster 0 contém exclusivamente casos de *overextrusion10*, sugerindo que este nível específico de sobreextrusão produz um padrão de gradientes distintivos. Esta distribuição demonstra que a técnica HOG foi eficaz em capturar as variações sutis nas bordas e texturas que caracterizam cada tipo de defeito de extrusão (Para o grupo de treino).

A Figura 28 apresenta a distribuição dos *clusters* no conjunto de treino, ilustrando a relação entre os tipos de defeito e os grupos formados.

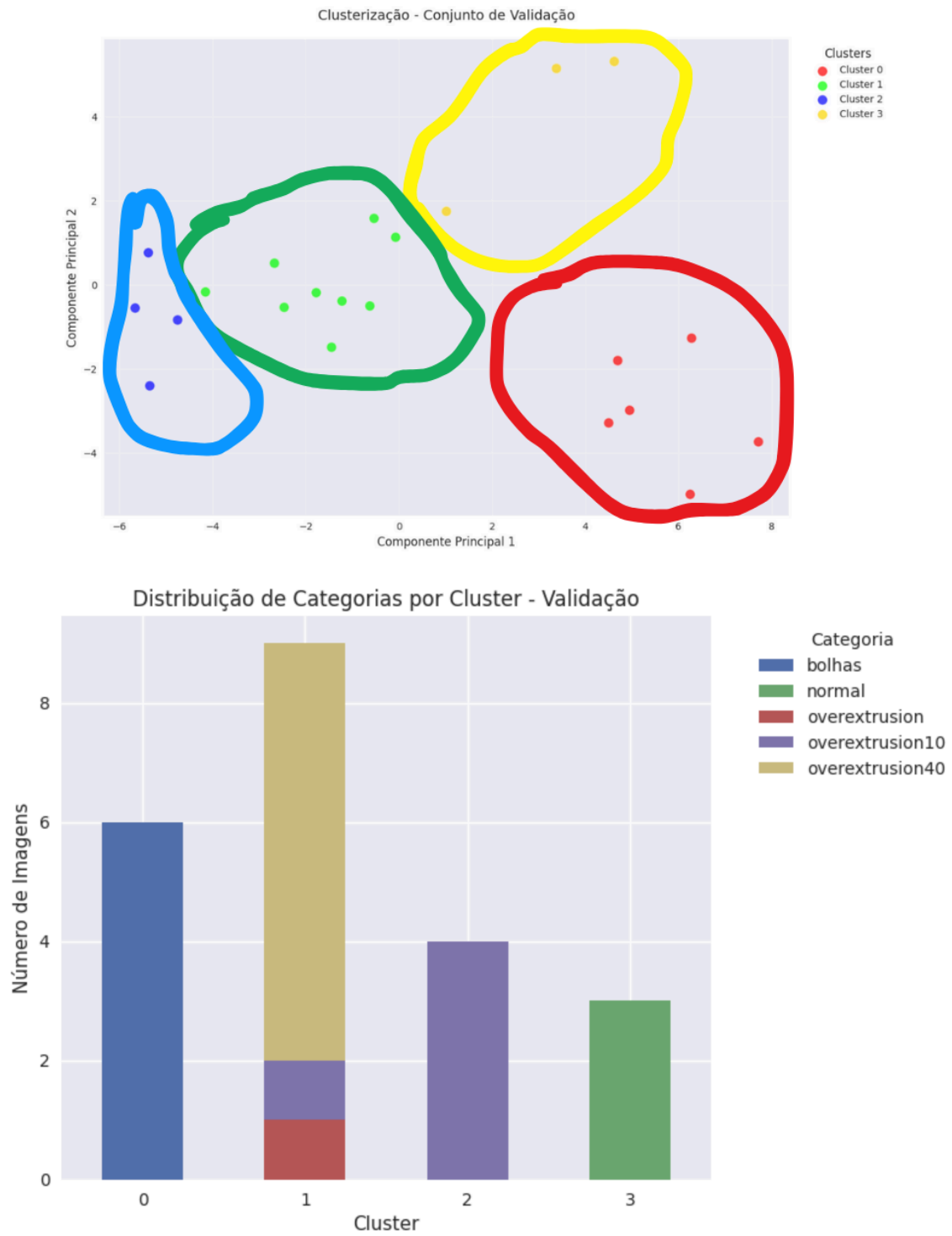
Figura 28 - Quantidade de imagens por *cluster*



Fonte: Próprio Autor

Da mesma forma, a Figura 29 exibe a distribuição dos *clusters* no conjunto de validação, permitindo verificar a consistência dos grupos em diferentes subconjuntos.

Figura 29 - Resultados no conjunto de validação



Fonte: Próprio Autor

4.3. Métodos de Validação

Para garantir a confiabilidade dos clusters formados, aplicamos múltiplas estratégias de validação:

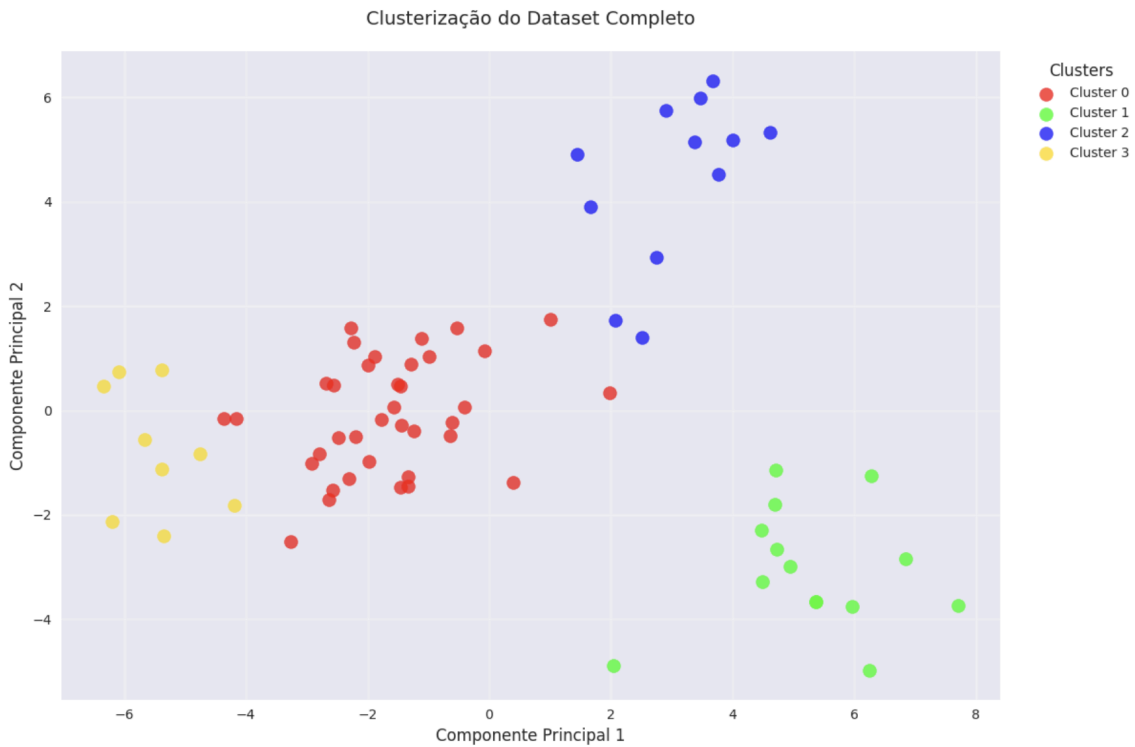
- Validação quantitativa
 - O *Elbow Method* (Figura 25) indicou entre 4 a 5 *clusters* como as melhores opções.
 - O *Silhouette Score* (Figura 26) mostrou valores entre 0.3 a 0.5, confirmando uma separação razoável dos grupos.
- Validação visual
 - Os *Scatter Plots* (Figuras 27 e 29) demonstram que os *clusters* estão separados de forma satisfatória no espaço PCA.
 - O gráfico de distribuição por *cluster* (Figura 28) revela uma distribuição lógica e coesa dos grupos.
- Validação qualitativa
 - Testes com diferentes números de *clusters* indicaram que valores superiores a 4 resultam em fragmentação desnecessária, dividindo grupos de forma inconsistente.
 - Para $k = 7$, os *clusters* foram fragmentados de acordo com tipos de padrões não reconhecidos, sem um benefício claro na segmentação. Potencialmente classificando apenas os defeitos ou formato da imagem.
 - As Figuras 30 e 31 apresentam dois plots globais, com todos os subconjuntos (treino e validação), demonstrando que a segmentação se mantém coerente.

Figura 30 - Conjunto de dados completo (com PCA) e modelo clusterizado



Fonte: Próprio Autor

Figura 31 - Conjunto de dados inteiro clusterizado



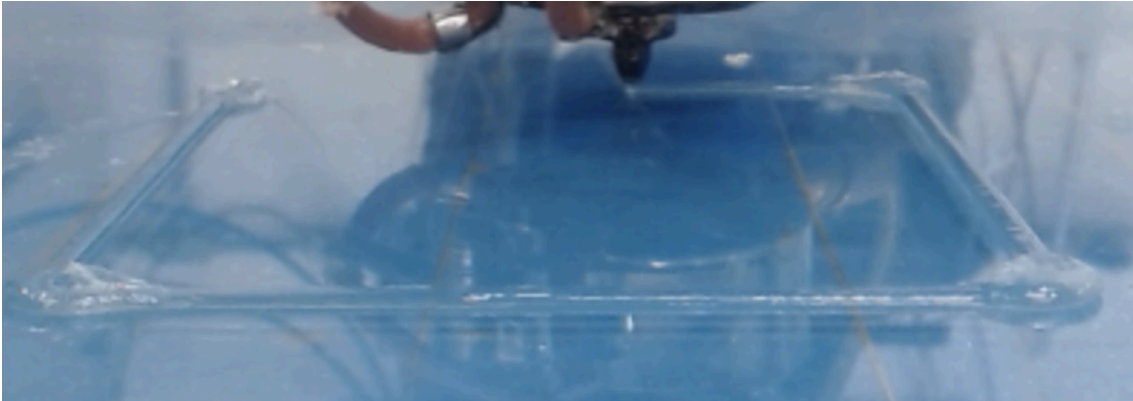
Fonte: Próprio Autor

- Validação com conjunto completo de imagens

A validação com o dataset expandido de aproximadamente 450 imagens proporciona insights importantes sobre a robustez do modelo HOG inicialmente

treinado com 72 imagens selecionadas. É crucial destacar que a seleção inicial mais restrita foi necessária para garantir a qualidade dos dados de treinamento, excluindo imagens que continham elementos que poderiam interferir na análise, como a presença do bico da extrusora na imagem ou peças ainda em processo de impressão, conforme exemplificado na Figura 32.

Figura 32 - Imagem de baixa qualidade para o modelo



Fonte: Próprio autor

Mesmo com essa expansão significativa para um conjunto de dados menos controlado, o modelo demonstrou uma capacidade consistente de diferenciação entre as categorias, como evidenciado pelos padrões de agrupamento nos gráficos de dispersão e na distribuição dos clusters. A clara separação espacial mantida entre peças normais (*cluster 2*) e diferentes tipos de defeitos sugere que as características HOG são suficientemente robustas para identificar padrões relevantes mesmo em um conjunto de dados mais heterogêneo e desafiador.

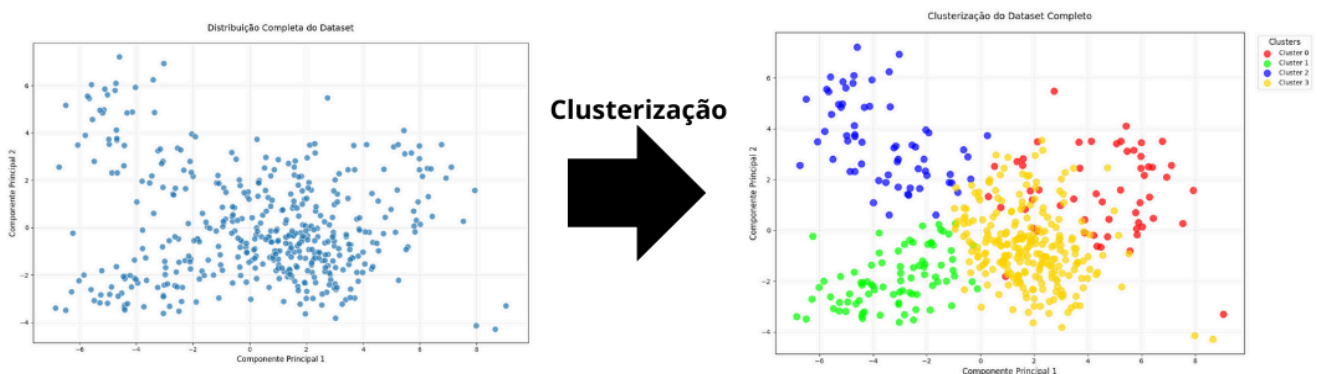
Esta validação para dataset expandido indica que, apesar do uso inicial de uma seleção das imagens para treinamento, o modelo desenvolvido possui uma generalização satisfatória, sendo capaz de processar e classificar corretamente mesmo imagens que não passaram pelo mesmo nível de curadoria inicial.

- O cluster 0 (vermelho) apresenta uma concentração de imagens com *overextrusion10* e *overextrusion40*, indicando que o HOG identificou padrões similares entre estes níveis de sobreextrusão, ocupando principalmente a região direita do espaço das componentes principais.

- O *cluster* 1 agrupou majoritariamente imagens com bolhas, demonstrando que o HOG foi eficaz em capturar os padrões circulares e texturas características deste tipo de defeito, localizando-se na região inferior esquerda do gráfico.
- O cluster 2 concentrou as peças normais na região superior esquerda do gráfico, evidenciando uma clara separação espacial e demonstrando a capacidade do HOG em identificar os padrões de gradientes característicos de peças sem defeitos.
- O *cluster* 3 apresenta uma mistura interessante de categorias (bolhas, normal e overextrusion), porém é crucial notar que esta mistura ocorre principalmente com imagens capturadas no início do processo de extrusão, quando os objetos ainda estão em formação. Isto explica a aparente "confusão" do modelo, pois estas imagens compartilham características visuais similares por estarem em estágios iniciais de impressão, especialmente nos casos de overextrusion¹⁰ e no começo do processo de extrusão.
- A distribuição espacial dos clusters no gráfico de dispersão mostra uma transição gradual entre as categorias, com separações mais definidas para peças normais e com bolhas, enquanto os diferentes níveis de overextrusion e peças em estágio inicial de impressão apresentam algumas sobreposições, o que é coerente com a natureza progressiva destes processos, porém não seria definido como o ideal.

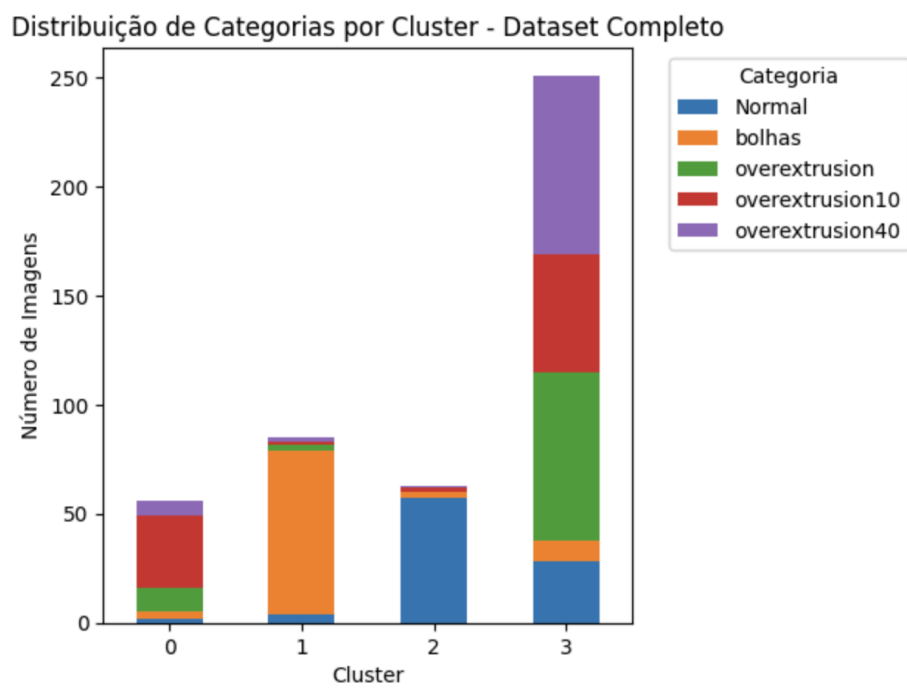
Nas Figuras 33 e 34 é possível observar os resultados no conjunto estendido de dados.

Figura 33 - Clusterização do dataset estendido



Fonte: Próprio Autor

Figura 34 - Distribuição de categorias por cluster



Fonte: Próprio Autor

5. SUGESTÕES PARA TRABALHOS FUTUROS

Este trabalho teve como objetivo investigar a aplicação de técnicas de visão computacional e aprendizado de máquina na detecção de defeitos em superfícies industriais, com foco especial na manufatura aditiva. A partir da utilização do algoritmo HOG para extração de características, combinado com técnicas de redução de dimensionalidade como PCA e clusterização com *Kmeans*, foi possível desenvolver um pipeline para identificação e classificação de defeitos.

Os resultados obtidos demonstraram pontos importantes tanto de sucesso quanto de limitações da abordagem escolhida. Entre os aspectos positivos, destaca-se a eficácia do modelo em identificar corretamente padrões de peças sem defeitos e aquelas com presença de bolhas, além da capacidade de classificar adequadamente os casos mais críticos de overextrusion (especialmente overextrusion10), que são particularmente prejudiciais para a qualidade final das peças.

Por outro lado, algumas limitações importantes foram identificadas. Os métodos de detecção de bordas Canny e análise de intensidade de pixels, inicialmente propostos como features relevantes, não demonstraram contribuição significativa para a classificação das imagens. Além disso, o modelo apresentou dificuldades na classificação de imagens capturadas no início do processo de impressão, um aspecto crítico considerando que a detecção precoce de defeitos seria ideal para minimizar desperdícios de material e tempo.

A aplicação do *Standard Scaler* e do PCA foi fundamental para normalizar os dados e reduzir a dimensionalidade, facilitando a análise e visualização dos clusters. O processo de seleção cuidadosa das imagens para treinamento, excluindo aquelas com elementos que poderiam interferir na análise (como o bico da extrusora), mostrou-se uma decisão metodológica importante, embora tenha limitado a capacidade do modelo em lidar com imagens em diferentes estágios do processo de impressão.

Como perspectivas futuras, sugere-se a exploração de técnicas supervisionadas, como redes neurais convolucionais (CNNs), que poderiam ser mais eficazes na classificação de imagens em diferentes estágios do processo de impressão. Além disso,

seria interessante desenvolver um sistema que pudesse identificar e classificar defeitos desde os estágios iniciais da impressão, possivelmente incorporando informações temporais do processo. A integração com sistemas embarcados para inspeção em tempo real também permanece como um objetivo relevante para aplicações práticas.

6. CONCLUSÃO

Uma contribuição significativa deste trabalho foi a demonstração de que, mesmo com técnicas relativamente simples de visão computacional, é possível criar um sistema eficaz para detecção de certos tipos de defeitos em manufatura aditiva. As limitações identificadas fornecem direcionamentos valiosos para futuras pesquisas na área, especialmente no que tange à detecção precoce de defeitos e à robustez do sistema em diferentes condições de operação.

Referências

BRADSKI, G.; KAEHLER, A. Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media, 2008.

CAGGIANO, Alessandra et al. Machine learning-based image processing for on-line defect recognition in additive manufacturing. *CIRP Annals*, v. 68, n. 1, p. 451–454, 2019. Disponível em: <https://doi.org/10.1016/j.cirp.2019.03.021>. Acesso em: 18 fev. 2025.

CANNY, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6), p. 679-698, 1986. Disponível em: <https://doi.org/10.1109/TPAMI.1986.4767851>.

CHEN, S.; JOHNSON, M. Enhanced K-means clustering for defect detection in complex surfaces. *Journal of Intelligent Manufacturing*, v. 34, n. 5, p. 781-795, 2023.

DALAL, Navneet; TRIGGS, Bill. Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). IEEE, 2005. p. 886-893.

ERA, I. Z.; AHMED, I.; LIU, Z.; DAS, S. An unsupervised approach towards promptable defect segmentation in laser-based additive manufacturing by Segment Anything. arXiv preprint, arXiv:2312.04063, 2023.

GAO, Wei et al. The status, challenges, and future of additive manufacturing in engineering. *Computer-Aided Design*, v. 69, p. 65-89, 2015.

GÉRON, A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. 2. ed. Sebastopol: O'Reilly Media, 2019.

GONZALEZ, R. C.; WOODS, R. E. Digital Image Processing. 4. ed. New York: Pearson, 2018.

JAIN, Anil K. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, v. 31, n. 8, p. 651-666, 2010.

JOLLIFFE, I. T. *Principal Component Analysis*. 2. ed. New York: Springer, 2016.

KAUR, P.; SHARMA, R. Clustering approaches for defect detection in manufacturing systems using K-Means and PCA. *International Journal of Advanced Manufacturing Technology*, v. 89, n. 4, p. 112–124, 2022.

KOVALSKYI, Semen; KOVAL, Vasyl. Comparison of image processing techniques for defect detection, 2024.

MACQUEEN, J. Some methods for classification and analysis of multivariate observations. In: *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, 1967.

OJALA, Timo; PIETIKAINEN, Matti; MAENPAA, Topi. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24, n. 7, p. 971-987, 2002.

OTSU, N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 9, n. 1, p. 62-66, 1979.

PARASKEVOUDIS, Konstantinos; KARAYANNIS, Panagiotis; KOUMOULOS, Elias P. Real-time 3D printing remote defect detection (stringing) with computer vision and artificial intelligence. *Processes*, v. 8, n. 11, p. 1464, 2020.

PULQUERIO, Eduardo Costa; BARBOSA, Gustavo Franco; SHIKI, Sidney Bruce. Robotic additive manufacturing system: development of suitable range of process parameters for 3D printing of a large-sized object in PLA polymer. *Progress in Additive Manufacturing*, Springer, 2024. p. 1-12.

RAO, P. K.; LIU, J.; ROBERSON, D.; KONG, Z. J.; WILLIAMS, C. Online real-time quality monitoring in additive manufacturing processes using heterogeneous sensors. *Journal of Manufacturing Science and Engineering*, v. 137, n. 6, p. 061007, 2021.

ROUSSEUW, Peter J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, v. 20, p. 53-65, 1987.

SOBEL, Irwin et al. Sobel-feldman operator. Preprint at <https://www.researchgate.net/profile/Irwin-Sobel/publication/285159837>. Accessed, v. 20, 2022.

SONG, Ruixian; MA, Jialing. Visualization of calculation of Local Binary Pattern (LBP): An example region of the image. ResearchGate, 2020. Disponível em: https://www.researchgate.net/figure/visualization-of-calculation-of-Local-Binary-Pattern-LBP-An-example-region-of-the_fig1_341341744. Acesso em: 26 jan. 2025.

SUZUKI, S.; ABE, K. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, v. 30, n. 1, p. 32-46, 1985. Disponível em: [https://doi.org/10.1016/0734-189X\(85\)90023-3](https://doi.org/10.1016/0734-189X(85)90023-3).

THOMPSON, R.; ANDERSON, K. Cost-benefit analysis of automated inspection systems in aerospace manufacturing. *Journal of Manufacturing Systems*, v. 66, p. 156-169, 2023.

VERBEECK, N.; CAPRIOLI, R. M.; VAN DE PLAS, R. Unsupervised machine learning for exploratory data analysis in imaging mass spectrometry. *Mass Spectrometry Reviews*, v. 39, n. 3, p. 245-291, 2020. Disponível em: <https://doi.org/10.1002/mas.21689>.

VOLPATO, Neri. *Manufatura aditiva: tecnologias e aplicações da impressão 3D*. Editora Blucher, 2017.

WANG, Y. et al. Advances in vision-based defect detection systems for additive manufacturing. *Additive Manufacturing Journal*, v. 36, p. 202–214, 2022.

WANG, Y.; ZHANG, J.; LIU, H. Real-time defect detection in additive manufacturing using deep learning approaches. *Additive Manufacturing*, v. 72, p. 102-115, 2024.

WONG, Kaufui V.; HERNANDEZ, Aldo. A review of additive manufacturing. *International Scholarly Research Notices*, v. 2012, 2012.

ZHANG, T. et al. Real-time edge detection systems in additive manufacturing using OpenCV. *Industrial Applications of Vision Systems*, v. 5, p. 98–110, 2023.

ZHU, Zhenhua; GERMAN, Stephanie; BRILAKIS, I. Detection of large-scale concrete columns for automated bridge inspection. *Automation in Construction*, v. 19, n. 8, p. 1047-1055, 2010.