

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA - CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL MECÂNICA
CURSO DE BACHARELADO EM ENGENHARIA MECATRÔNICA**

GABRIEL FURTADO PAVANATI

**APLICAÇÃO DE MACHINE LEARNING EM VISÃO COMPUTACIONAL PARA
DETECÇÃO DE FALHAS EM OBJETOS**

FLORIANÓPOLIS, 2026.

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA - CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL MECÂNICA
CURSO DE BACHARELADO EM ENGENHARIA MECATRÔNICA**

GABRIEL FURTADO PAVANATI

**APLICAÇÃO DE MACHINE LEARNING EM VISÃO COMPUTACIONAL PARA
DETECÇÃO DE FALHAS EM OBJETOS**

Trabalho de Conclusão de Curso submetido
ao Instituto Federal de Educação, Ciência
e Tecnologia de Santa Catarina como parte
dos requisitos para obtenção do título de
Engenheiro Mecatrônico.

Orientador:
Prof. Dr. Delcino Picinin Júnior

FLORIANÓPOLIS, 2026.

Ficha de identificação da obra elaborada pelo autor.

Pavanati, Gabriel Furtado
**Aplicação de Machine Learning em Visão Computacional
para Detecção de Falhas em Objetos / Gabriel Furtado Pavanati,
orientação de Delcino Picinin Júnior. - Florianópolis,
SC, 2026.**
79 p.

**Trabalho de Conclusão de Curso (TCC) - Instituto Federal
de Santa Catarina, Câmpus Florianópolis. Bacharelado
em Engenharia Mecatrônica. Departamento
Acadêmico de Metal Mecânica.
Inclui Referências.**

**1. Visão computacional. 2. Aprendizado de máquina.
3. Processamento de imagens. 4. Classificação de imagens.
I. Picinin Júnior, Delcino. II. Instituto Federal
de Santa Catarina. III. Aplicação de Machine Learning
em Visão Computacional para Detecção de Falhas em
Objetos.**

APLICAÇÃO DE MACHINE LEARNING EM VISÃO COMPUTACIONAL PARA DETECÇÃO DE FALHAS EM OBJETOS

GABRIEL FURTADO PAVANATI

Este trabalho foi julgado adequado para obtenção do título de Bacharel em Engenharia Mecatrônica e aprovado na sua forma final pela banca examinadora do Curso de Bacharelado em Engenharia Mecatrônica do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis, 27 de Julho, 2026.

Banca Examinadora:

Delcino Picinin Júnior, Prof. Dr. (orientador)

Maurício Edgar Stivanello, Prof. Dr.

Raimundo Ricardo Matos da Cunha, Prof. Dr.

AGRADECIMENTOS

Aos meus pais, por me proporcionarem as melhores condições possíveis ao longo da vida, oferecendo apoio e oportunidades que foram fundamentais para minha formação pessoal e acadêmica.

À minha namorada Pietra, por me incentivar diariamente a evoluir 1% a cada dia, sempre oferecendo apoio, motivação e palavras de encorajamento ao longo desta jornada.

Aos colegas de curso, pelo apoio nos momentos difíceis, pela parceria e pela amizade construída ao longo de toda a graduação.

Aos professores do IFSC, pela dedicação, pelos ensinamentos transmitidos e por contribuírem de forma significativa para minha formação acadêmica e pessoal.

A beleza do aprendizado é que ninguém pode roubá-lo de você.
- B. B. King

RESUMO

Este trabalho propõe o desenvolvimento de um sistema de inspeção visual automatizada baseado em técnicas de *machine learning* aplicadas à visão computacional, com o objetivo de classificar em tempo real amostras de tampas como intactas ou violadas. A motivação para a adoção dessa abordagem reside na dificuldade de se desenvolver algoritmos tradicionais suficientemente robustos para lidar com as variações presentes nas amostras, tais como reflexos metálicos, irregularidades superficiais e variações de ângulo de captura. O sistema foi desenvolvido utilizando o *software NI Vision Builder AI* (VBAI), no qual foram implementadas etapas de pré-processamento de imagem, binarização e classificação por aprendizado de máquina supervisionado. Para o treinamento do classificador, foram coletadas 200 imagens, igualmente distribuídas entre as duas classes, e empregada a técnica de validação cruzada com 10 *folds*, garantindo maior confiabilidade na avaliação do desempenho do modelo. A fim de identificar a combinação mais eficiente, foram testadas todas as combinações possíveis entre os três algoritmos de classificação disponíveis, sendo eles o *Nearest Neighbor*, o *K-Nearest Neighbor* e o *Minimum Mean Distance*, e as três métricas de distância: euclidiana, máxima e de *Manhattan*. O melhor resultado obtido atingiu uma taxa média de acertos de 71%, valor que, embora indique a viabilidade da abordagem, não se mostrou suficiente para aplicação em ambiente produtivo real. A principal limitação identificada foi a impossibilidade de manipulação manual do vetor de características dentro do *software* utilizado, fator determinante para a precisão de classificadores baseados em *machine learning*. Como perspectivas futuras, sugere-se a utilização de ferramentas que permitam maior controle sobre o vetor de características, ou adoção de abordagens mais sofisticadas, como o *deep learning*, paradigma no qual o próprio sistema determina automaticamente as características mais relevantes de forma iterativa.

Palavras-chave: Visão computacional. Aprendizado de máquina. Processamento de imagens. Classificação de imagens.

ABSTRACT

This work proposes the development of an automated visual inspection system based on machine learning techniques applied to computer vision, with the objective of classifying samples of lids as intact or tampered in real time. The motivation for adopting this approach lies in the difficulty of developing traditional algorithms sufficiently robust to handle the variations present in the samples, such as metallic reflections, surface irregularities, and variations in capture angle. The system was developed using the NI Vision Builder AI (VBAI) software, in which image preprocessing, binarization, and supervised machine learning classification steps were implemented. For classifier training, 200 images were collected, equally distributed between the two classes, and the cross-validation technique with 10 folds was employed, ensuring greater reliability in the evaluation of the model's performance. In order to identify the most efficient combination, all possible combinations among the three available classification algorithms were tested, namely Nearest Neighbor, K-Nearest Neighbor, and Minimum Mean Distance, along with the three distance metrics: Euclidean, maximum, and Manhattan. The best result achieved an average accuracy rate of 71%, a value that, although indicating the viability of the approach, proved insufficient for application in a real production environment. The main limitation identified was the inability to manually manipulate the feature vector within the software used, a determining factor for the precision of machine learning-based classifiers. As future perspectives, the use of tools that allow greater control over the feature vector is suggested, as well as the adoption of more sophisticated approaches, such as deep learning, a paradigm in which the system itself automatically determines the most relevant features in an iterative manner.

Keywords: Computer vision. Machine learning. Image processing. Image classification.

LISTA DE FIGURAS

Figura 1 – Imagem RGB convertida para tons de cinza	18
Figura 2 – Imagem binarizada	19
Figura 3 – Processo de captura de imagem	21
Figura 4 – Filtro para redução de ruído	22
Figura 5 – Imagem ilustrativa de campo de visão	27
Figura 6 – Diagrama ilustrativo de distância focal	29
Figura 7 – Fluxograma do processo de treinamento e classificação de amostras	31
Figura 8 – Sistema montado	38
Figura 9 – Câmera Basler acA1300-60gc	39
Figura 10 – Software NI Vision Builder AI	40
Figura 11 – Exemplar de tampa intacta	41
Figura 12 – Exemplar de tampa violada	42
Figura 13 – Mesa rotativa	42
Figura 14 – Interface gráfica de amostragem	44
Figura 15 – Câmera NI-1776C	45
Figura 16 – Interface para sinais de entrada e saída	46
Figura 17 – Diagrama interno da interface	47
Figura 18 – Interface com tampa superior removida	47
Figura 19 – Interface montada	48
Figura 20 – Etapa de aquisição de imagem	50
Figura 21 – Etapa de conversão para tons de cinza	50
Figura 22 – Etapa de aplicação de filtro gaussiano	51
Figura 23 – Etapa de extração de contorno circular	52
Figura 24 – Etapa de ajuste de brilho	52
Figura 25 – Detalhes do ajuste de brilho	53
Figura 26 – Etapa de classificação da amostra	55
Figura 27 – Configurações do classificador	55
Figura 28 – Calculadora lógica	56
Figura 29 – Etapa de exportação de imagem	57
Figura 30 – Atualização do status global	58
Figura 31 – Delimitador de <i>string</i>	59
Figura 32 – Envio de dados via TCP	60
Figura 33 – Aplicação de delay na aquisição	61
Figura 34 – Algoritmo <i>K-Nearest Neighbor</i> e distância euclideana	64
Figura 35 – Algoritmo <i>K-Nearest Neighbor</i> e distância soma	64
Figura 36 – Algoritmo <i>K-Nearest Neighbor</i> e distância máxima	65
Figura 37 – Algoritmo <i>Nearest Neighbor</i> e distância euclideana	66

Figura 38 – Algoritmo <i>Nearest Neighbor</i> e distância soma	66
Figura 39 – Algoritmo <i>Nearest Neighbor</i> e distância máxima	67
Figura 40 – Algoritmo <i>Minimum Mean Distance</i> e distância euclideana	68
Figura 41 – Algoritmo <i>Minimum Mean Distance</i> e distância soma	68
Figura 42 – Algoritmo <i>Minimum Mean Distance</i> e distância máxima	69
Figura 43 – Cálculo de velocidade linear da mesa	75
Figura 44 – Cálculo de tempo máximo de exposição	76
Figura 45 – Tabela-verdade pass/fail	77
Figura 46 – Especificações Técnicas da câmera Basler acA1300-60gc	79

LISTA DE ABREVIATURAS E SIGLAS

CIELAB	CIE 1976 L^* , a^* , b^* color space
CIELUV	CIE 1976 L^* , u^* , v^* color space
CSI	Controle e Supervisão Inteligente
FOV	Field of View
GigE	Gigabit Ethernet
IFSC	Instituto Federal de Santa Catarina
IP	Internet Protocol
ITU-R	International Telecommunication Union – Radiocommunication Sector
LED	Light Emitting Diode
NI	National Instruments
RGB	Red Green Blue
TCP	Transmission Control Protocol
VBAI	Vision Builder for Automated Inspection
sRGB	Standard Red Green Blue

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Justificativa	14
1.2	Definição do Problema	15
1.3	Objetivo Geral	15
1.4	Objetivos Específicos	15
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Visão Computacional	17
2.1.1	Conversão entre Plano de Cor RGB para Escala de Cinza	17
2.1.2	Binarização	19
2.1.3	Processo de Captura de Imagem	20
2.1.4	Filtros para Redução de Ruído	22
2.1.4.1	<i>Filtros Lineares</i>	23
2.1.4.2	<i>Filtros Não Lineares</i>	23
2.1.4.3	<i>Filtragem em Imagens Coloridas</i>	23
2.1.5	Parâmetros de Aquisição de Imagem	24
2.1.5.1	<i>Zoom</i>	24
2.1.5.2	<i>Tempo de Exposição</i>	25
2.1.5.3	<i>Ganho</i>	26
2.1.5.4	<i>Campo de Visão</i>	27
2.1.5.5	<i>Abertura do Diafragma</i>	28
2.1.5.6	<i>Foco e Distância Focal</i>	28
2.2	<i>Machine Learning</i> Supervisionado	30
2.2.1	Vetor de Características	32
2.2.2	Algoritmos de Classificação	33
2.2.3	Métricas de Distância	35
2.2.4	Validação Cruzada	36
3	DESCRIÇÃO DO SISTEMA PROPOSTO	38
3.1	Câmera Industrial	39
3.2	Software	40
3.3	Amostra	41
3.4	Mesa Rotativa	42
3.5	Interface Gráfica de Amostragem	43
3.6	Consideração do Sistema	44
4	METODOLOGIA	49
5	APRESENTAÇÃO DOS RESULTADOS	63
5.1	Análise e discussão dos resultados	63
6	CONSIDERAÇÕES FINAIS	71
6.1	Sugestões para trabalhos futuros	71
	REFERÊNCIAS	72
	APÊNDICES	74
	APÊNDICE A – CÁLCULO DE VELOCIDADE LINEAR DA MESA	75
	APÊNDICE B – CÁLCULO DE TEMPO MÁXIMO DE EXPOSIÇÃO	76

APÊNDICE C – TABELA-VERDADE PASS/FAIL	77
ANEXOS	78
ANEXO A – ESPECIFICAÇÕES CÂMERA BASLER ACA1300-60GC	79

1 INTRODUÇÃO

Em um cenário de constante evolução tecnológica, a indústria busca continuamente inovações que permitam produzir com maior segurança, qualidade, eficiência e menor custo. Nesse contexto, a visão computacional tem sido amplamente adotada para otimizar processos, possibilitando a realização de inferências complexas que, até então, dependiam exclusivamente da atuação humana. Como estratégia para enfrentar problemas de maior complexidade, o campo de *machine learning* vem sendo cada vez mais explorado, especialmente em situações nas quais o processamento convencional de imagens, por si só, não se mostra suficiente.

A visão humana destaca-se como o sentido mais sofisticado, razão pela qual as imagens exercem papel central na percepção do mundo. Entretanto, enquanto o ser humano está restrito à faixa visível do espectro eletromagnético, sistemas de formação de imagens são capazes de explorar praticamente todo esse espectro, desde raios gama até ondas de rádio, além de processar dados provenientes de fontes não convencionalmente associadas a imagens, como ultrassom, microscopia eletrônica e imagens geradas por computador. Nesse contexto, o processamento digital de imagens configura-se como uma área ampla e com múltiplas aplicações. Paralelamente, campos como a visão computacional buscam reproduzir, por meio de computadores, capacidades da visão humana incluindo aprendizado, inferência e tomada de decisão a partir de informações visuais. Diante disso, observa-se que o processamento digital de imagens exerce influência significativa em praticamente todos os domínios técnicos contemporâneos (GONZALEZ; WOODS, 2018).

O uso de técnicas de *machine learning* torna-se particularmente relevante em problemas caracterizados por alta complexidade ou pela necessidade de adaptação a diferentes condições ao longo do tempo. Diferentemente de programas tradicionais, que executam instruções previamente definidas e permanecem inalterados após sua implementação, algoritmos de aprendizado de máquina são capazes de aprender a partir de dados e experiências, permitindo lidar com tarefas difíceis de serem explicitamente programadas, como reconhecimento de imagens, reconhecimento de voz e outras atividades associadas ao comportamento humano. Além disso, tais métodos possibilitam maior flexibilidade em cenários nos quais os requisitos ou padrões de uso variam entre usuários ou ao longo do tempo (DATTA; DAVIM, 2022).

Diante disso, o trabalho proposto consiste em utilizar técnicas de visão computacional associadas ao *machine learning* para a detecção automática de falhas em tampas de latas de atum, por meio da aquisição de imagens utilizando uma câmera industrial. A partir dessas imagens, serão aplicados métodos de processamento ao objeto, permitindo que um algoritmo de aprendizado de máquina realize a classificação

das tampas quanto à sua integridade.

1.1 Justificativa

A inspeção de qualidade em processos industriais é uma etapa fundamental para garantir a conformidade dos produtos, a segurança do consumidor e a eficiência da produção. No setor alimentício, esse controle torna-se ainda mais crítico, uma vez que falhas no processo de vedação ou integridade de embalagens podem comprometer a conservação do alimento e representar riscos sanitários. No caso específico de latas de atum, a integridade da tampa é um fator essencial para assegurar que o produto permaneça devidamente selado até o momento do consumo.

Tradicionalmente, a verificação da integridade das embalagens é realizada por meio de inspeção visual humana ou por sistemas automatizados baseados em regras fixas de processamento digital de imagens, como limiarização, detecção de bordas, operações morfológicas e análise de histogramas. Essas técnicas funcionam bem em condições controladas, mas apresentam limitações quando aplicadas a superfícies metálicas, como as tampas de latas. Isso ocorre porque esse tipo de superfície costuma apresentar reflexos luminosos, variações de textura e pequenas deformações decorrentes do processo produtivo, fatores que dificultam a definição de critérios rígidos de detecção. Como essas abordagens dependem de parâmetros fixos, como limiares de intensidade ou padrões de borda pré-estabelecidos, pequenas variações nas condições da imagem podem ser suficientes para reduzir sua precisão, comprometendo a confiabilidade de técnicas puramente determinísticas.

Nesse contexto, o uso de técnicas de visão computacional associadas ao *machine learning* surge como uma alternativa promissora para aumentar a robustez e a confiabilidade da inspeção automática. Diferentemente de métodos baseados exclusivamente em regras previamente definidas, algoritmos de aprendizado de máquina são capazes de identificar padrões relevantes diretamente a partir de conjuntos de dados, permitindo maior adaptação a variações presentes nas imagens analisadas.

Além de contribuir para o aumento da precisão na detecção de falhas, a aplicação de técnicas de *machine learning* em sistemas de inspeção visual também favorece a automação de processos industriais, reduzindo a dependência de inspeção manual, aumentando a repetibilidade das análises e possibilitando maior escalabilidade em linhas de produção de alta velocidade. Dessa forma, o desenvolvimento de um sistema capaz de classificar automaticamente tampas de latas de atum quanto à sua integridade apresenta relevância tanto do ponto de vista acadêmico quanto industrial, contribuindo para o avanço de aplicações práticas de visão computacional no contexto da inspeção de qualidade.

1.2 Definição do Problema

A detecção automática de falhas em tampas metálicas de latas de atum constitui um problema relevante no contexto da inspeção visual automatizada. O objetivo central consiste em determinar, a partir de imagens capturadas do produto, se a tampa encontra-se em condição íntegra ou se apresenta indícios de violação ou deformação que possam comprometer a vedação da embalagem.

Entretanto, a identificação dessas condições por meio de técnicas tradicionais de processamento digital de imagens pode tornar-se complexa devido a diversos fatores inerentes ao processo de captura e às características físicas do objeto analisado. Superfícies metálicas tendem a gerar reflexos luminosos intensos, variações de brilho e irregularidades aparentes de textura, o que pode dificultar a aplicação de métodos baseados exclusivamente em limiares, segmentação simples ou extração de características rígidas.

Além disso, pequenas variações no posicionamento do objeto, nas condições de iluminação e nas propriedades da superfície podem produzir alterações significativas na aparência das imagens, mesmo quando o objeto encontra-se em estado normal de operação. Dessa forma, a definição de regras determinísticas capazes de separar de maneira confiável as classes de interesse torna-se um desafio considerável.

Diante desse cenário, torna-se pertinente investigar o uso de algoritmos baseados em *machine learning* para auxiliar na tarefa de classificação das tampas, permitindo que o sistema aprenda padrões discriminantes diretamente a partir de exemplos previamente rotulados. Assim, o problema abordado neste trabalho consiste em desenvolver e avaliar um método de visão computacional capaz de classificar automaticamente imagens de tampas de latas de atum em duas categorias: intacta ou violada, utilizando técnicas de aprendizado de máquina que permitam lidar com as variabilidades presentes nas imagens capturadas.

1.3 Objetivo Geral

Desenvolver um algoritmo baseado em técnicas de visão computacional e *machine learning* capaz de classificar objetos em uma esteira como intactos ou violados, por meio de imagens capturadas por uma câmera industrial.

1.4 Objetivos Específicos

Dado o objetivo geral, foram traçados os seguintes objetivos específicos:

- a) Projetar o sistema de aquisição de imagens;
- b) Construir um sistema supervisor com banco de dados;

- c) Implementar técnicas de pré-processamento;
- d) Treinar e validar o classificador;
- e) Avaliar o desempenho do sistema.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão abordados os principais conceitos teóricos relacionados ao aprendizado de máquina e à visão computacional. Inicialmente, serão apresentados temas como aprendizado de máquina supervisionado, técnicas de validação cruzada, algoritmos de classificação e métricas de distância, que são fundamentais para a construção e avaliação de modelos preditivos. Em seguida, serão explorados tópicos de visão computacional, incluindo os parâmetros de aquisição de imagem, como zoom, tempo de exposição, foco, abertura de diafragma e ganho, além do processo de captura de imagens, técnicas de filtragem para redução de ruído, métodos de binarização e conversão entre diferentes planos de cores, com o objetivo de estabelecer uma base sólida para a compreensão e aplicação dessas abordagens.

2.1 Visão Computacional

A visão computacional pode ser compreendida como uma área que busca desenvolver métodos capazes de interpretar e reconstruir informações do mundo real a partir de imagens. Enquanto o sistema visual humano é capaz de perceber, de forma aparentemente simples, a estrutura tridimensional de cenas, identificar objetos e interpretar características como forma, iluminação e profundidade, a reprodução desse processo por sistemas computacionais apresenta elevada complexidade. Isso ocorre porque a visão computacional está associada à resolução de um problema inverso, no qual se tenta inferir propriedades do mundo tridimensional com base em projeções bidimensionais, frequentemente contendo informações incompletas ou ambíguas.

Nesse contexto, torna-se necessário o uso de modelos matemáticos e físicos que descrevem a formação da imagem, incluindo aspectos relacionados à propagação da luz, interação com superfícies e processo de aquisição por sensores. Além disso, abordagens estatísticas e técnicas de aprendizado de máquina são frequentemente empregadas para lidar com incertezas e ambiguidades inerentes ao processo. Apesar dos avanços significativos, ainda existem limitações na capacidade de sistemas computacionais em interpretar imagens com o mesmo nível de compreensão e generalização apresentado por humanos, evidenciando a complexidade intrínseca da área. Ainda assim, a visão computacional possui ampla aplicação prática, sendo utilizada em diferentes domínios, como reconhecimento de caracteres, inspeção industrial, veículos autônomos e análise de imagens médicas (SZELISKI, 2022).

2.1.1 Conversão entre Plano de Cor RGB para Escala de Cinza

A representação de imagens em cor no espaço RGB é amplamente utilizada em sistemas digitais devido à sua compatibilidade com dispositivos eletrônicos, como

monitores e câmeras. No entanto, em muitos processos de tratamento e análise de imagens, a conversão para uma representação monocromática (escala de cinza) é necessária para simplificar a análise ou reduzir a complexidade computacional.

A conversão de imagens coloridas do espaço RGB para tons de cinza não deve ser realizada por uma simples média aritmética dos valores das componentes R (vermelho), G (verde) e B (azul), pois o olho humano possui sensibilidades distintas a cada uma dessas cores. Conforme modelos fundamentados na percepção visual humana e na colorimetria, a luminância, ou intensidade percebida da luz, pode ser representada como uma combinação linear ponderada das componentes RGB, atribuindo pesos que refletem a sensibilidade relativa do sistema visual às diferentes faixas espectrais. A Figura 1 ilustra a conversão de uma imagem RGB para tons de cinza.

Figura 1 – Imagem RGB convertida para tons de cinza



Fonte: Do autor (2026)

A fórmula utilizada para a obtenção da intensidade em escala de cinza I é expressa por:

$$I = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \quad (1)$$

onde R , G e B correspondem às intensidades das componentes vermelha, verde e azul, respectivamente (ITU-R, 2011). A maior ponderação atribuída à componente verde justifica-se pela maior sensibilidade do olho humano a essa faixa do espectro, seguida pelo vermelho e, por fim, pelo azul, que contribui menos para a percepção de luminância.

Esse modelo está embasado em processos da colorimetria que explicam como o sistema visual humano processa a luz em termos de luminância e cromaticidade, sendo amplamente utilizado em aplicações práticas de processamento digital de

imagens. A transformação linear ponderada é parte fundamental da conversão do espaço de cores RGB para espaços baseados em luminância, essenciais em técnicas de compressão, transmissão e *enhancement* de imagens (PRATT, 2001).

Portanto, para preservar adequadamente a percepção visual do brilho e evitar distorções causadas por métodos inadequados, a conversão para escala de cinza deve seguir a combinação ponderada explicitada na fórmula acima.

2.1.2 Binarização

Documentos digitalizados geralmente apresentam uma combinação de elementos, como textos, desenhos e regiões gráficas, caracterizando-se como conteúdos heterogêneos. Em diversas aplicações, torna-se necessário extrair ou aprimorar essas informações, sendo comum a conversão da imagem para uma representação binária. Nesse contexto, a binarização consiste em um processo fundamental no processamento de imagens, no qual uma imagem em escala de cinza é convertida em duas classes distintas de pixels, normalmente associadas ao fundo e ao objeto de interesse. Essa separação é realizada por meio de um valor de limiar, o que torna a técnica de limiarização uma abordagem eficiente para segmentação, com aplicações que incluem análise de documentos, processamento de mapas, detecção de alvos e inspeção de qualidade (FIRDOUSI; PARVEEN, 2014).

A Figura 2 mostra algumas imagens binarizadas com diferentes limiares aplicados.

Figura 2 – Imagem binarizada



Fonte: Adaptado de (FIRDOUSI; PARVEEN, 2014)

Na abordagem de limiarização local adaptativa, o valor de limiar deixa de ser único para toda a imagem e passa a ser definido individualmente para cada pixel, sendo representado por uma função $T(x, y)$. Esse limiar é obtido a partir de características estatísticas da vizinhança do pixel, considerando uma região local de dimensão $w \times w$,

e pode envolver medidas como variância, amplitude ou ajustes de superfície. Dada uma imagem em escala de cinza representada por $I(x, y)$, a classificação dos pixels na imagem binarizada $b(x, y)$ é realizada com base na comparação entre a intensidade local e o limiar calculado, conforme a seguinte relação:

$$b(x, y) = \begin{cases} 1, & \text{se } I(x, y) > T(x, y) \\ 0, & \text{caso contrário} \end{cases} \quad (2)$$

Dessa forma, cada pixel é atribuído a uma das duas classes de acordo com seu contexto local, o que torna essa técnica mais robusta em situações onde há variações de iluminação ou presença de ruído, em comparação com métodos de limiarização global (FIRDOUSI; PARVEEN, 2014).

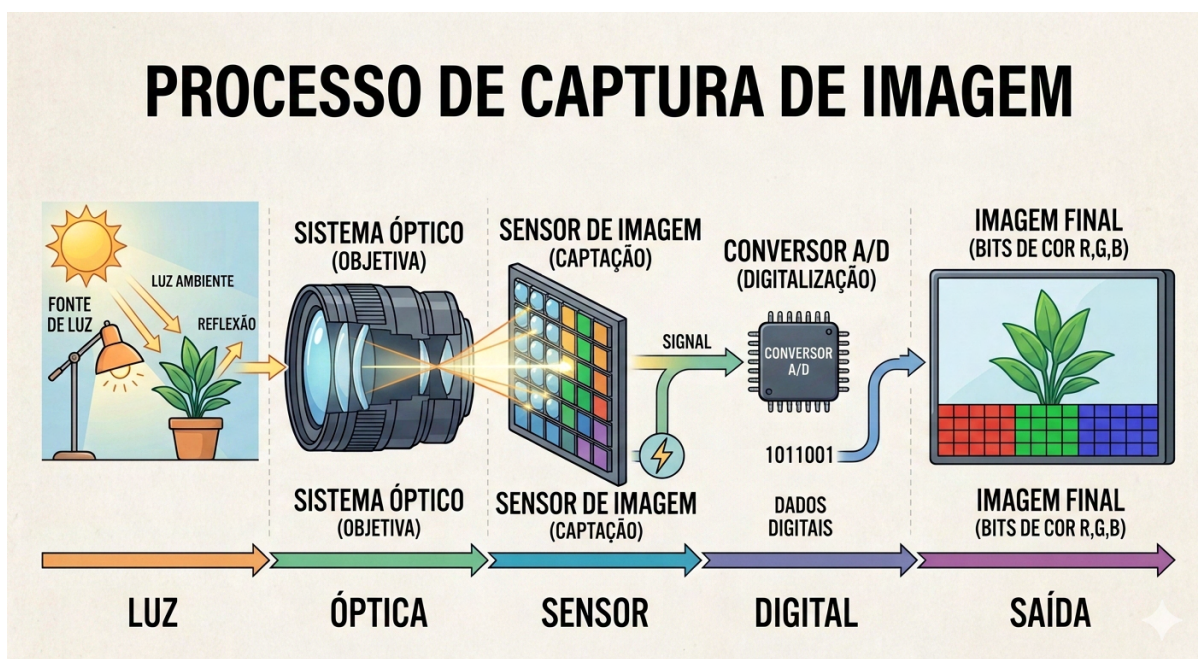
2.1.3 Processo de Captura de Imagem

O processo de captura de imagens em câmeras digitais envolve uma sequência de etapas físicas e eletrônicas que convertem a luz proveniente de uma cena em uma representação numérica discreta (GONZALEZ; WOODS, 2018). A Figura 3 apresenta o processo completo de captura de uma imagem. Inicialmente, a luz refletida pelos objetos na cena é coletada pelo sistema óptico da câmera, que geralmente inclui lentes responsáveis por focar e projetar a imagem num plano de captura. Esse plano consiste em uma grade bidimensional de pequenos detectores fotossensíveis que registram a intensidade da luz em pontos discretos, formando a imagem digital (FORSYTH; PONCE, 2012).

Cada elemento sensor em um arranjo (pixel) responde à energia incidente transformando-a em uma tensão proporcional à intensidade da luz recebida. Essa tensão é resultado da conversão da energia luminosa pela combinação do poder elétrico de entrada e do material sensível ao tipo de energia detectada. A saída do sensor é então digitalizada para formar a imagem digital. A quantidade de sinal gerada está relacionada à energia que incide sobre cada elemento sensor durante o processo de aquisição da imagem (GONZALEZ; WOODS, 2018).

A formação de imagens coloridas em sensores baseia-se em capturar a luz em diferentes comprimentos de onda, geralmente através do uso de filtros que modulam a resposta dos receptores para componentes de cor específicas. A combinação dessas respostas permite a reconstrução da informação cromática da cena capturada, apesar das limitações práticas na sensibilidade e sobreposição espectral dos sensores (FORSYTH; PONCE, 2012).

Figura 3 – Processo de captura de imagem



Fonte: Do Autor (2026)

Após a aquisição da imagem, o sinal contínuo gerado pelos sensores é convertido em formato digital por meio dos processos de amostragem e quantização, que consistem em discretizar as coordenadas espaciais e os níveis de intensidade, respectivamente. A profundidade de bits da imagem está relacionada ao número de níveis de intensidade quantizados, sendo comum o uso de 8 bits (256 níveis), embora algumas aplicações utilizem mais níveis, como 16 bits, para melhor representação. A imagem digital resultante é representada como uma função bidimensional discreta, $f(x, y)$, onde cada par de coordenadas corresponde a um pixel contendo o valor da intensidade luminosa para imagens monocromáticas. Para imagens coloridas, cada pixel normalmente armazena múltiplos valores correspondentes aos componentes de cor (GONZALEZ; WOODS, 2018).

Formalmente, uma imagem digital pode ser descrita como uma função discreta:

$$f(x, y) \rightarrow Z \quad (3)$$

onde x e y representam as coordenadas espaciais discretizadas e o valor associado corresponde à intensidade quantizada. No caso de imagens coloridas, essa representação é estendida para múltiplos canais:

$$f(x, y) = [R(x, y), G(x, y), B(x, y)] \quad (4)$$

Além disso, o processo de digitalização envolve duas etapas fundamentais: a amostragem espacial, que define a resolução da imagem (número de pixels), e a quantização, que define a precisão dos níveis de intensidade. Imperfeições nesse processo, como ruído eletrônico, limitações do sensor e distorções ópticas, influenciam diretamente a qualidade da imagem capturada, sendo aspectos essenciais a serem considerados em aplicações de visão computacional e processamento digital de imagens (BOVIK, 2000).

Dessa forma, a captura de imagens não é apenas uma simples aquisição visual, mas um processo complexo de transdução físico-eletrônica e discretização matemática, que estabelece a base para todas as etapas subsequentes de processamento e análise.

2.1.4 Filtros para Redução de Ruído

Filtros desempenham um papel fundamental no processamento digital de imagens, permitindo a manipulação dos valores dos pixels para diversas finalidades, como suavização, realce e remoção de ruídos. Diferentemente das operações ponto a ponto, em que o valor de cada pixel resultante depende unicamente do pixel correspondente na imagem original, os filtros utilizam uma região ao redor do pixel para calcular o novo valor, considerando múltiplos pixels simultaneamente (BURGER; BURGE, 2016).

A Figura 4 ilustra uma aplicação de filtro de ruído, onde a imagem da esquerda representa a imagem original, enquanto a imagem da direita recebe o filtro.

Figura 4 – Filtro para redução de ruído



2.1.4.1 Filtros Lineares

Os filtros lineares são caracterizados pela aplicação de uma combinação linear dos valores de pixels de uma vizinhança, usando um conjunto de pesos conhecidos como máscara ou kernel. Formalmente, um filtro linear H aplicado a uma imagem discreta $I(u, v)$ pode ser expresso pela convolução bidimensional

$$I'(u, v) = (I * H)(u, v) = \sum_{(i,j) \in R_H} I(u - i, v - j) \cdot H(i, j), \quad (5)$$

onde R_H indica a região da máscara, e $I'(u, v)$ é o valor do pixel filtrado na posição (u, v) (BURGER; BURGE, 2016).

Um dos filtros lineares mais importantes é o filtro gaussiano, que utiliza uma função gaussiana bidimensional como peso da máscara:

$$H_{G,\sigma}(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right). \quad (6)$$

Este filtro realiza uma suavização isotrópica, com maior peso dado aos pixels próximos ao centro da máscara, e é especialmente eficaz para redução de ruídos enquanto preserva detalhamento estrutural moderado da imagem. Além disso, a propriedade de separabilidade da gaussiana permite sua implementação eficiente como a aplicação consecutiva de filtros unidimensionais em cada direção.

Outro filtro linear importante é o filtro "box", que calcula a média simples dos pixels em uma região, porém apresenta um comportamento inadequado no domínio da frequência, introduzindo artefatos indesejados, como oscilações ("ringing"), devido ao seu corte abrupto no espaço da máscara (BURGER; BURGE, 2016).

2.1.4.2 Filtros Não Lineares

Filtros não lineares são aplicados quando as características do ruído ou dos detalhes da imagem exigem operações que não sejam combinações lineares dos pixels. Um exemplo clássico é o filtro de mediana, que substitui cada pixel pelo valor mediano dos pixels na vizinhança, sendo particularmente eficiente para remover ruído sal e pimenta sem desfocar as bordas (BURGER; BURGE, 2016).

2.1.4.3 Filtragem em Imagens Coloridas

A aplicação de filtros em imagens coloridas é comumente realizada separadamente em cada canal de cor, como nos canais R, G e B. Embora simples, essa abordagem pode gerar artefatos de cor, especialmente quando o espaço de cor é não

linear, como o sRGB. Para garantir maior fidelidade, o processamento deve considerar espaços lineares (e.g., RGB linear) ou espaços perceptualmente uniformes, como CIELAB ou CIELUV, que são mais adequados para operações baseadas em distância e interpolação de cores.

Essa escolha afeta substancialmente o produto final do filtro, pois o espaço de cor influencia a forma como a suavização ou realce são percebidos visualmente. Por exemplo, filtros gaussianos aplicados diretamente em sRGB podem introduzir bandas escuras em regiões com mudança abrupta de saturação, enquanto sua aplicação em RGB linear ou espaços perceptuais melhora a qualidade visual da suavização (BURGER; BURGE, 2016).

2.1.5 Parâmetros de Aquisição de Imagem

A forma como uma imagem é capturada está diretamente relacionada a um conjunto de parâmetros que controlam o comportamento da câmera durante a aquisição. Esses parâmetros determinam como a luz proveniente da cena será registrada pelo sensor e, conseqüentemente, influenciam características importantes da imagem, como brilho, nitidez e enquadramento. Assim, antes de qualquer etapa de processamento, é importante compreender como essas variáveis interferem no resultado final obtido.

Entre os principais parâmetros envolvidos nesse processo, destacam-se o tempo de exposição, o zoom, o foco, o ganho, o campo de visão e a abertura do diafragma. Cada um deles atua de maneira distinta na formação da imagem, seja controlando a quantidade de luz capturada, ajustando a região observada ou influenciando o nível de detalhamento da cena. Na prática, esses parâmetros não atuam de forma isolada, mas sim de maneira conjunta, exigindo ajustes que considerem as condições específicas de cada aplicação.

Dessa forma, torna-se pertinente apresentar esses parâmetros de maneira individual, destacando suas principais características e efeitos.

2.1.5.1 Zoom

O zoom em câmeras fotográficas está relacionado à capacidade do sistema óptico de variar a distância focal, permitindo alterar o campo de visão e o tamanho da imagem projetada no sensor ou filme. Lentes com distâncias focais mais curtas, chamadas de grande angular, proporcionam um campo de visão mais amplo, capturando uma área maior da cena. Por outro lado, lentes com distância focal longa, denominadas teleobjetivas, focam em ângulos estreitos, ampliando objetos distantes. A variação contínua entre esses extremos é realizada por conjuntos de lentes que se movem para ajustar a distância focal do sistema, permitindo capturar imagens com diferentes campos angulares sem a necessidade de troca de lente física. O ajuste do

foco acompanha a mudança na posição da lente para manter a nitidez da imagem no sensor. Lentes zoom modernas são projetadas para manter uma boa qualidade de imagem com grande abertura relativa, possibilitando exposições rápidas adequadas para fotografia em diferentes condições luminosas. O campo angular é um parâmetro crucial para definir o zoom, pois ele determina a amplitude da cena capturada pela câmera. Assim, o zoom é uma ferramenta essencial na fotografia, oferecendo versatilidade para enquadrar objetos em diversas distâncias e composições, facilitada pela evolução técnica dos sistemas ópticos (HECHT, 2017).

2.1.5.2 Tempo de Exposição

O tempo de exposição determina o intervalo durante o qual o sensor permanece exposto à luz para capturar uma imagem. À medida que a luz incide sobre o sensor, ela libera elétrons, acumulando uma carga proporcional à intensidade da iluminação e à duração da exposição. Portanto, um tempo de exposição mais longo permite que o sensor acumule mais carga, resultando em uma imagem mais clara, enquanto um tempo muito curto pode gerar imagens escuras e com baixo contraste, prejudicando a distinção de detalhes.

Para aplicações industriais, especialmente na inspeção de objetos em movimento, o controle adequado do tempo de exposição é crucial para evitar imagens desfocadas. O desfoque ocorre quando o objeto desloca-se significativamente durante o período em que a exposição está ativa. Assim, é necessário calcular um tempo máximo de exposição que permita captar imagens nítidas, considerando a velocidade do objeto e a resolução espacial da câmera.

O tempo máximo de exposição pode ser calculado conforme a seguinte fórmula:

$$E_{\max} = \frac{FOV_{\text{horizontal}} \times B}{(V_{\text{horizontal}}) \times (N_{\text{horizontal}})} \quad (7)$$

Onde $E_{\max}$ representa o tempo máximo de exposição, $FOV_{\text{horizontal}}$ corresponde ao campo de visão na direção do movimento, $V_{\text{horizontal}}$ é a velocidade do objeto projetada nessa mesma direção e $N_{\text{horizontal}}$ é o número de pixels do sensor ao longo desse eixo. O termo B representa o borramento admissível, expresso em pixels, ou seja, o deslocamento máximo tolerado da imagem no plano do sensor durante a aquisição (CHOUINARD, 2018).

O borramento por movimento pode ser modelado como o deslocamento da imagem no sensor durante o tempo de exposição. Considerando a velocidade relativa do objeto e os parâmetros geométricos do sistema de imageamento, é possível estimar o tempo máximo de exposição admissível de forma a limitar o borramento.

Na prática, adota-se frequentemente $B = 1$ pixel como limite para evitar degradação perceptível da imagem, garantindo que o deslocamento durante a exposição não ultrapasse a resolução de um único pixel.

Além disso, em ambientes com iluminação variável ou presença de fontes de luz oscilantes, como lâmpadas fluorescentes que piscam em frequências detectáveis pela câmera, é recomendada a utilização de fontes de iluminação dedicadas para garantir a consistência das imagens adquiridas. O tempo de exposição também pode ser sincronizado com fontes externas de iluminação através de gatilhos (*trigger*), que acionam a captura da imagem em momentos precisos, permitindo o uso de luzes estroboscópicas para evitar os efeitos de variações ambientais (NI, 2011).

Portanto, o tempo de exposição deve ser cuidadosamente configurado em função das condições de iluminação, velocidade dos objetos e requisitos de processamento da imagem, de modo a otimizar o contraste e a nitidez das imagens capturadas para aplicação em visão computacional e inspeção automatizada (NI, 2011).

2.1.5.3 Ganho

O ganho corresponde a um multiplicador aplicado ao sinal analógico antes da digitalização, influenciando diretamente o nível de brilho dos pixels na imagem. A aplicação de um ganho baixo resulta em valores de pixel mais homogêneos, o que pode limitar a distinção entre diferentes regiões da imagem. Por outro lado, níveis intermediários de ganho aumentam a variação nos valores dos pixels, aprimorando o contraste e facilitando a identificação de detalhes visuais. Contudo, a ampliação excessiva do ganho pode causar saturação da imagem em áreas mais claras, onde os pixels atingem o valor máximo e, conseqüentemente, perde-se a capacidade de diferenciar nuances nessas regiões.

Além disso, o ganho atua sobre o sinal e o ruído de maneira equivalente, de modo que sua elevação também propicia aumento do ruído presente na imagem. Dessa forma, o aumento do ganho deve ser utilizado com cautela, preferencialmente em situações onde não seja possível incrementar a iluminação do ambiente. Para aplicações que demandem imagens claras e de alta qualidade, é mais eficaz a utilização de fontes de iluminação adicionais do que o aumento do ganho puro e simples.

Por fim, o ganho está intimamente relacionado com outros parâmetros da câmera, como o tempo de exposição e a qualidade da iluminação, que devem ser balanceados para assegurar imagens com contraste e brilho adequados, otimizando o desempenho dos algoritmos de processamento de imagem e a precisão da inspeção visual (NI, 2011).

2.1.5.4 Campo de Visão

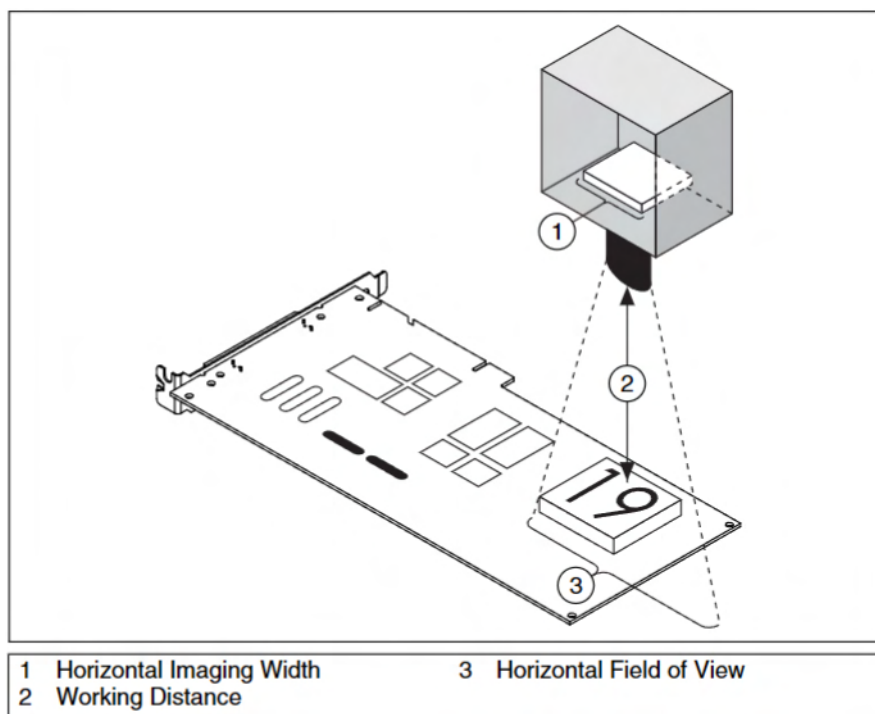
O campo de visão (*Field of View* - FOV) é a área do objeto a ser inspecionado que será capturada pela câmera. É fundamental garantir que o campo de visão abranja completamente o elemento a ser analisado, pois isso influencia diretamente a eficiência e precisão da inspeção visual.

A definição do campo de visão depende de alguns parâmetros que envolvem o sensor de imagem e a posição física da câmera. Para calcular o FOV horizontal ou vertical, pode-se utilizar a seguinte relação:

$$FOV = \frac{\text{Passo do Pixel} \times \text{Número de Pixels Ativos} \times \text{Distância de Trabalho}}{\text{Distância Focal}} \quad (8)$$

onde “Passo do Pixel” representa a distância entre o centro de pixels adjacentes no sensor (sendo os pixels quadrados), “Número de Pixels Ativos” corresponde à quantidade de pixels ativos na direção considerada, “Distância de Trabalho” é a distância entre a lente da câmera e o objeto a ser inspecionado, e “Distância Focal” se refere à distância focal da lente, que determina o grau de convergência ou divergência da luz. A Figura 5 ilustra esse conceito usando uma câmera e uma amostra.

Figura 5 – Imagem ilustrativa de campo de visão



Fonte: National Instruments (2011)

Este cálculo auxilia na seleção apropriada de componentes, incluindo o ajuste na distância de trabalho, escolha da lente e resolução do sensor, para estabelecer um campo de visão adequado ao processo de inspeção.

Além disso, a escolha correta do campo de visão influencia o nível de detalhe da imagem e a velocidade de processamento, já que um campo de visão maior pode reduzir a resolução por área, enquanto um campo de visão menor pode aumentar a precisão, porém restringindo o espaço visualizado (NI, 2011).

2.1.5.5 Abertura do Diafragma

A abertura do diafragma, também conhecida como abertura relativa ou número f , é um dos elementos fundamentais no controle da exposição e da qualidade da imagem fotográfica. Ela é definida como a relação entre a distância focal da lente e o diâmetro da pupila de entrada (equivalente ao diâmetro efetivo do diafragma). Essa relação é expressa numericamente pelo número f (por exemplo, $f/2$, $f/5.6$), indicando que uma abertura maior corresponde a um número f menor, permitindo a entrada de mais luz no sistema óptico.

A abertura afeta diretamente a quantidade de luz que incide sobre o sensor ou filme, sendo que uma abertura maior resulta em maior iluminação da imagem, facilitando exposições em ambientes com pouca luz. Além disso, o ajuste do diafragma influencia a profundidade de campo, ou seja, a faixa de distância dentro da cena que permanece em foco nítido. Aberturas menores (números f maiores) proporcionam maior profundidade de campo, ampliando a área de nitidez, enquanto aberturas maiores reduzem essa faixa, isolando o foco em um plano específico e desfocando o fundo e a frente da imagem.

O mecanismo do diafragma geralmente é composto por lâminas que formam uma abertura circular ajustável (íris), que podem ser controladas manualmente ou automaticamente. Em câmeras modernas, sistemas automáticos permitem ajuste eletrônico da abertura, sincronizando com mecanismos de foco e exposição para otimizar o controle da imagem (JACOBSON *et al.*, 2000).

2.1.5.6 Foco e Distância Focal

O conceito de foco em ótica refere-se ao ponto onde os raios de luz paralelos incidentes em um sistema óptico se convergem após a refração ou reflexão. No caso particular dos espelhos esféricos, o ponto focal está relacionado ao raio de curvatura da superfície refletora. Quando os raios de luz chegam paralelos ao eixo principal, a distância entre o espelho e a posição onde os raios convergem define a distância focal f , que corresponde à metade do raio de curvatura r da superfície, representada pela expressão:

$$f = \frac{r}{2} \quad (9)$$

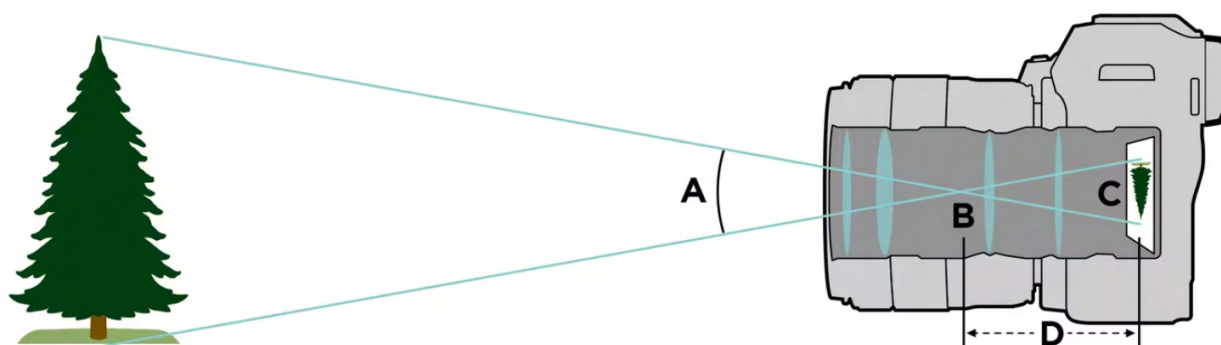
Essa relação está explicitada a partir da análise das propriedades geométricas dos espelhos esféricos e é fundamental para a compreensão do comportamento dos

sistemas ópticos refletivos (FOWLES, 1975).

Em uma lente simples, composta por apenas um elemento, isto é, uma única peça de vidro óptico responsável por refratar a luz, a distância focal é a distância, em milímetros, entre o plano focal e o centro do elemento quando a lente está focada no infinito. Em uma câmera analógica, o plano focal é o filme; em uma câmera digital, é a superfície sensível à luz do sensor (CANON, 2026).

As lentes modernas são muito mais complexas do que uma lente de elemento único, mas ainda possuem um centro óptico conhecido como ponto nodal. Esse é o ponto pelo qual todos os raios de luz passam, convergindo para um ponto em seu caminho até o sensor. A distância focal é a distância entre o plano focal e o ponto nodal da lente (CANON, 2026).

Figura 6 – Diagrama ilustrativo de distância focal



Fonte: Canon (2026)

A Figura 6 mostra que a distância focal de uma lente é a distância (D) entre o plano do sensor (C) e o centro óptico ou ponto nodal (B) da lente, determinando o ângulo de visão (A) da lente.

Para lentes delgadas, a distância focal relaciona-se com os índices de refração do material da lente e o meio externo, bem como os raios de curvatura das duas superfícies da lente. A equação que relaciona o objeto, a imagem e a distância focal f para paraxialmente pequenos ângulos é dada pela fórmula de lentes delgadas:

$$\frac{1}{f} = (n - 1) \left(\frac{1}{r_1} - \frac{1}{r_2} \right) \quad (10)$$

onde n é o índice de refração relativo da lente em relação ao meio, e r_1 e r_2 são os raios de curvatura das faces convexa e côncava da lente, respectivamente (FOWLES, 1975).

A geometria de raios é fundamental para a determinação da formação de imagens, focalização e propriedades da distância focal em sistemas ópticos. Para uma correta interpretação dos fenômenos óticos, são adotadas convenções de sinais para distâncias e raios de curvatura. Especificamente, o raio de curvatura de uma superfície

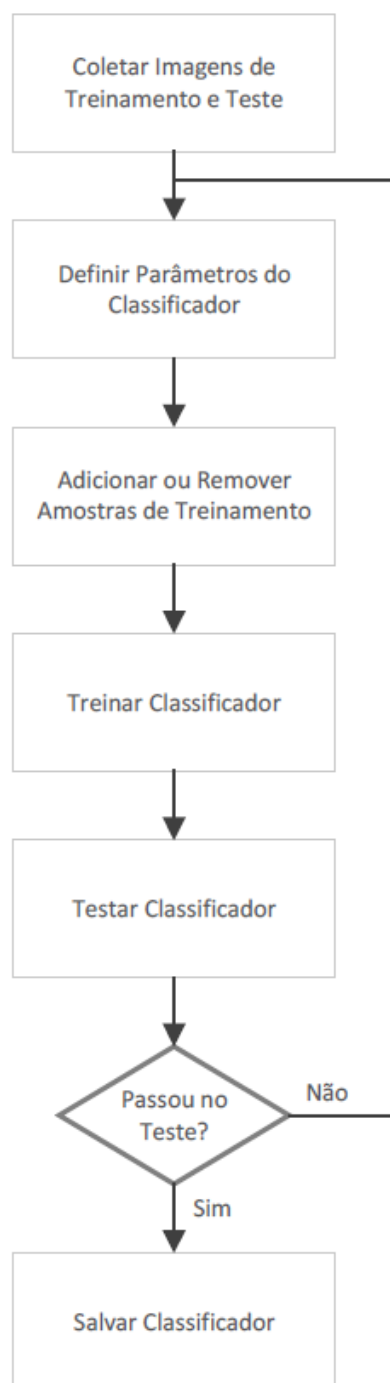
é considerado positivo quando a superfície é côncava voltada para a direção da luz incidente, e negativo quando é convexa nesse mesmo sentido.

2.2 *Machine Learning* Supervisionado

O aprendizado de máquina supervisionado é uma das abordagens centrais para tarefas de classificação e regressão em visão computacional. Nesse paradigma, o modelo é treinado a partir de um conjunto de dados rotulado, no qual cada exemplo de entrada é associado à sua respectiva saída ou classe desejada (SZELISKI, 2022).

O aprendizado de máquina é uma área que permite que programas aprendam automaticamente a partir de dados, oferecendo uma alternativa prática à programação manual, e tem sido amplamente utilizado em diversas aplicações, como filtros de spam, sistemas de recomendação e detecção de fraude. No aprendizado supervisionado, especificamente os classificadores são sistemas que recebem vetores compostos por valores de características, podendo ser discretas ou contínuas, para prever uma classe discreta, como, por exemplo, identificar uma mensagem como “spam” ou “não spam” (DOMINGOS, 2012).

O processo de treinamento e classificação de amostras pode ser definido como mostra a Figura 7:

Figura 7 – Fluxograma do processo de treinamento e classificação de amostras

Fonte: National Instruments (2014)

Inicialmente, são coletadas imagens de treinamento e teste. Em seguida, realiza-se a definição dos parâmetros do classificador, onde são configuradas as características e ajustes do modelo. Depois disso, ocorre a etapa de adição ou remoção de amostras de treinamento, permitindo ajustar o conjunto de dados conforme necessário. Com os dados preparados, o próximo passo é treinar o classificador, seguido da etapa de teste, na qual o desempenho do modelo é avaliado com dados de teste (NI, 2014).

Após o teste, há uma decisão: “Passou no teste?”. Caso não, o processo retorna para etapas anteriores, permitindo novos ajustes nos parâmetros e/ou nas amostras, repetindo o ciclo de treinamento e teste. Caso sim, o classificador é considerado adequado e segue para a etapa final de salvamento (NI, 2014).

2.2.1 Vetor de Características

Para classificar um objeto ou evento, é necessário medir suas propriedades ou características representativas. A título de exemplo, para classificar um caractere impresso, pode ser útil conhecer sua área e seu perímetro, mensurar sua compacidade por meio da razão entre sua área e o quadrado de seu perímetro, ou ainda avaliar o grau de simetria do objeto em relação a um eixo horizontal, comparando a área de sua metade superior com a de sua metade inferior (DUDA, 1997).

O modelo clássico de reconhecimento de padrões é composto por dois elementos principais: o extrator de características e o classificador. O extrator de características processa os dados brutos e determina os valores numéricos de um conjunto de d características, denominadas $x_1, x_2, \dots, x_d$. O classificador, por sua vez, recebe esse conjunto de valores e o atribui a uma das c categorias possíveis.

Quando é possível mensurar um conjunto fixo de d características para qualquer objeto que se deseja classificar, o conjunto dessas medições pode ser organizado na forma de um vetor de características $\mathbf{x}$, definido matematicamente como um vetor coluna de dimensão d , conforme expresso na Equação 11:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \quad (11)$$

Esse vetor pode ser interpretado geometricamente como um ponto em um espaço de características d -dimensional, de modo que cada objeto inspecionado é representado abstratamente como um ponto nesse espaço. Essa representação permite que algoritmos de classificação operem sobre grandezas numéricas bem definidas, facilitando a separação entre classes distintas.

O projeto do extrator de características é fortemente dependente do problema em questão. O extrator ideal seria aquele capaz de produzir o mesmo vetor de características para todos os padrões pertencentes a uma mesma classe, e vetores distintos para padrões de classes diferentes. Na prática, entradas diferentes fornecidas ao extrator produzirão inevitavelmente vetores de características ligeiramente distintos, sendo desejável que a variabilidade dentro de uma mesma classe seja pequena em

relação à variabilidade entre classes distintas. Dado um conjunto de características adequadamente definido, a tarefa do classificador consiste em utilizar essas informações para distinguir os padrões e atribuir cada objeto à sua respectiva classe.

2.2.2 Algoritmos de Classificação

No contexto de aprendizado de máquina supervisionado, os algoritmos de classificação desempenham um papel fundamental na construção de sistemas capazes de atribuir categorias a padrões baseando-se em exemplos previamente rotulados. A aprendizagem supervisionada envolve a disponibilidade de dados de treinamento nos quais cada padrão é associado a uma categoria específica. Esta abordagem requer o desenvolvimento de técnicas que sejam suficientemente poderosas para aprender a solução do problema e estáveis diante das variações dos parâmetros (DUDA; HART; STORK, 2001).

Além disso, a seleção adequada do modelo é crucial para evitar o *overfitting*, um problema em que modelos excessivamente complexos classificam perfeitamente os dados de treinamento mas apresentam baixo desempenho em dados novos, devido à captura de ruídos e variações específicas dos dados de treino. Portanto, é necessário encontrar um equilíbrio na complexidade do modelo, assegurando que ele seja suficientemente expressivo para discriminar as categorias relevantes sem superajustar os dados disponíveis (DUDA; HART; STORK, 2001).

Existem diversos algoritmos de classificação no contexto do aprendizado de máquina supervisionado, cada um com características e aplicações específicas. Dentre os mais conhecidos, destacam-se o *Nearest Neighbor*, o *K-Nearest Neighbor* e o *Minimum Mean Distance*, amplamente utilizados em tarefas de reconhecimento de padrões e tomada de decisão baseada em similaridade.

Na classificação pelo método do vizinho mais próximo (*Nearest Neighbor*), a distância entre um vetor de características X , cuja classe é desconhecida, e uma classe C_j é definida como a menor distância entre X e as amostras que representam essa classe, conforme expresso por

$$d(X, C_j) = \min_i d(X, X_i^j), \quad (12)$$

em que $d(X, X_i^j)$ representa a distância entre X e a i -ésima amostra da classe C_j . A regra de decisão consiste em atribuir o padrão X à classe cujo representante esteja mais próximo, isto é,

$$X \in C_j \quad \text{se} \quad d(X, C_j) = \min_i d(X, C_i). \quad (13)$$

Esse método é amplamente reconhecido por sua simplicidade e caráter intuitivo, apresentando bom desempenho quando existem vetores representativos adequados para cada classe. Além disso, em cenários nos quais uma classe é representada por múltiplas amostras distribuídas em diferentes regiões do espaço de características, o classificador do vizinho mais próximo tende a ser mais eficaz do que abordagens baseadas na distância média mínima (NI, 2014).

Na classificação pelo método dos k -vizinhos mais próximos (*K-Nearest Neighbor*), um vetor de características de entrada X é atribuído a uma classe C_j com base em um mecanismo de votação. Nesse processo, o classificador identifica os k vizinhos mais próximos de X dentre todas as classes disponíveis, e a classe atribuída corresponde àquela que apresenta a maioria dos votos entre esses k vizinhos.

Em aplicações reais, padrões atípicos (outliers), frequentemente causados por ruído, podem levar a classificações incorretas quando se utiliza o método do vizinho mais próximo. Nesse contexto, o classificador k -vizinhos mais próximos apresenta maior robustez ao ruído, uma vez que a decisão é baseada em múltiplas amostras vizinhas, reduzindo a influência de pontos isolados. Por exemplo, para um vetor de entrada X , a escolha de $k = 1$ pode resultar em uma determinada classe, enquanto $k = 3$ pode levar a uma classificação diferente, refletindo a influência do conjunto de vizinhos considerados (NI, 2014).

No classificador baseado na distância média mínima (*Minimum Mean Distance*), considera-se que cada classe C_j é representada por um conjunto de n_j vetores de características $\{X_1^j, X_2^j, \dots, X_{n_j}^j\}$, todos rotulados como pertencentes à classe j . O centro da classe, também chamado de média, é definido como

$$M_j = \frac{1}{n_j} \sum_{i=1}^{n_j} X_i^j, \quad (14)$$

representando o vetor médio das amostras da classe. Na fase de classificação, um vetor de entrada X , de classe desconhecida, é atribuído à classe cujo centro esteja mais próximo, de acordo com a regra

$$X \in C_j \quad \text{se} \quad d(X, M_j) = \min_i d(X, M_i), \quad (15)$$

em que $d(X, M_j)$ corresponde à função de distância definida conforme a métrica adotada durante a fase de treinamento.

Esse método é particularmente eficaz em cenários nos quais há pouca variabilidade nos padrões de características ou quando os dados apresentam baixo nível de ruído, fazendo com que as amostras de cada classe se concentrem em torno

de seus respectivos centros. Nessas condições, o classificador de distância média mínima apresenta boa eficiência computacional, uma vez que, durante a classificação, é necessário calcular apenas as distâncias entre o vetor de entrada e os centros das classes, em vez de considerar todas as amostras representativas (NI, 2014).

2.2.3 Métricas de Distância

As métricas de distância são fundamentais para quantificar a dissimilaridade entre objetos em espaços de múltiplas dimensões, sendo extensamente utilizadas em técnicas como agrupamento e classificação. Uma classe importante de dissimilaridades são as distâncias, que possuem propriedades específicas, tais como positividade, simetria e a desigualdade triangular, que lhes conferem utilidade teórica e prática.

As distâncias possuem propriedades que garantem consistência para análise:

Não-negatividade: $d(x, y) \geq 0$ (16)

Identidade: $d(x, x) = 0$, e $d(x, y) = 0$ se, e somente se, $x = y$ (17)

Simetria: $d(x, y) = d(y, x)$ (18)

Desigualdade triangular: $d(x, y) \leq d(x, z) + d(z, y)$ (19)

Estas propriedades permitem o uso dessas distâncias em algoritmos de classificação para *machine learning* supervisionado (HAN; KAMBER; PEI, 2012).

A distância Euclidiana é a mais conhecida e utilizada, caracterizando-se pelo cálculo da raiz quadrada da soma dos quadrados das diferenças entre as componentes correspondentes dos vetores dos objetos. Formalmente, para dois pontos x e y em um espaço n -dimensional, a distância Euclidiana é definida pela seguinte fórmula:

$$d(x, y) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2} \quad (20)$$

Esta métrica corresponde à menor distância “em linha reta” entre os pontos e é amplamente empregada quando se deseja medir similaridade em espaços contínuos e métricos (TAN; STEINBACH; KUMAR, 2006).

A distância de Manhattan, também conhecida como distância do táxi ou *city block*, é um caso particular da distância de Minkowski para o parâmetro $r=1$. Ela é calculada somando-se os valores absolutos das diferenças entre os atributos correspondentes:

$$d(x, y) = \sum_{k=1}^n |x_k - y_k| \quad (21)$$

Essa medida reflete um cenário onde o movimento entre os pontos ocorre ao longo de eixos ortogonais, como num tabuleiro urbano, e pode ser menos sensível a discrepâncias extremas em apenas uma dimensão (TAN; STEINBACH; KUMAR, 2006).

Por sua vez, a distância de Chebyshev corresponde à distância do máximo, ou distância L_∞ , que toma o valor máximo entre as diferenças absolutas dos atributos:

$$d(x, y) = \max_{1 \leq k \leq n} |x_k - y_k| \quad (22)$$

Essa métrica avalia a maior discrepância entre as componentes dos objetos, sendo relevante em situações onde um atributo pode dominar a consideração de dissimilaridade (TAN; STEINBACH; KUMAR, 2006).

De maneira geral, essas distâncias são exemplos da família das distâncias de Minkowski, que generaliza diversas métricas através do parâmetro r e permite escolher a forma mais adequada às características dos dados e ao problema tratado.

$$d(x, y) = \left(\sum_{k=1}^n |x_k - y_k|^r \right)^{\frac{1}{r}} \quad (23)$$

A escolha entre elas depende da natureza dos atributos, do comportamento desejado da métrica, e considerações práticas de aplicação (TAN; STEINBACH; KUMAR, 2006).

2.2.4 Validação Cruzada

A validação cruzada é uma técnica fundamental para a avaliação e ajuste de classificadores em aprendizado de máquina, uma vez que ajuda a mitigar o problema do *overfitting*, que ocorre quando um modelo aprende peculiaridades ou ruídos dos dados de treinamento, apresentando excelente desempenho nesses dados, mas baixo desempenho em dados novos. A validação cruzada consiste em dividir aleatoriamente o conjunto de dados de treinamento em vários subconjuntos (por exemplo, dez), reservando cada um deles para teste enquanto o modelo é treinado nos demais. Dessa

forma, é possível testar o classificador em dados não utilizados no treinamento e calcular a média dos resultados obtidos nos diferentes subconjuntos, proporcionando uma estimativa mais robusta da performance do modelo. Essa prática é especialmente relevante porque, sem uma separação rigorosa entre os dados de treino e teste, há o risco de que o classificador apresente um desempenho ilusório, excelente nos dados conhecidos, mas inadequado para dados novos (DOMINGOS, 2012).

No entanto, deve-se tomar cuidado para não usar a validação cruzada excessivamente na escolha dos parâmetros do modelo, pois isso pode levar a um ajuste exagerado da validação, comprometendo sua eficácia como método preventivo de *overfitting* (DOMINGOS, 2012).

Além disso, na formação dos subconjuntos, é comum manter aproximadamente a mesma proporção entre as classes em cada partição, o que contribui para uma avaliação mais representativa do modelo. A avaliação do desempenho do classificador pode ser expressa por meio da acurácia, conforme a Equação 24.

$$\text{Acurácia} = \frac{\text{Número de amostras classificadas corretamente}}{\text{Número total de amostras}} \times 100 \quad (24)$$

Como a divisão dos dados é realizada de forma aleatória, é esperado que ocorram pequenas variações nos resultados entre diferentes execuções do processo. Essas variações podem ser utilizadas como indicativo da estabilidade do classificador, sendo que oscilações reduzidas tendem a indicar maior robustez, enquanto variações mais expressivas podem sugerir instabilidade no modelo (NI, 2014).

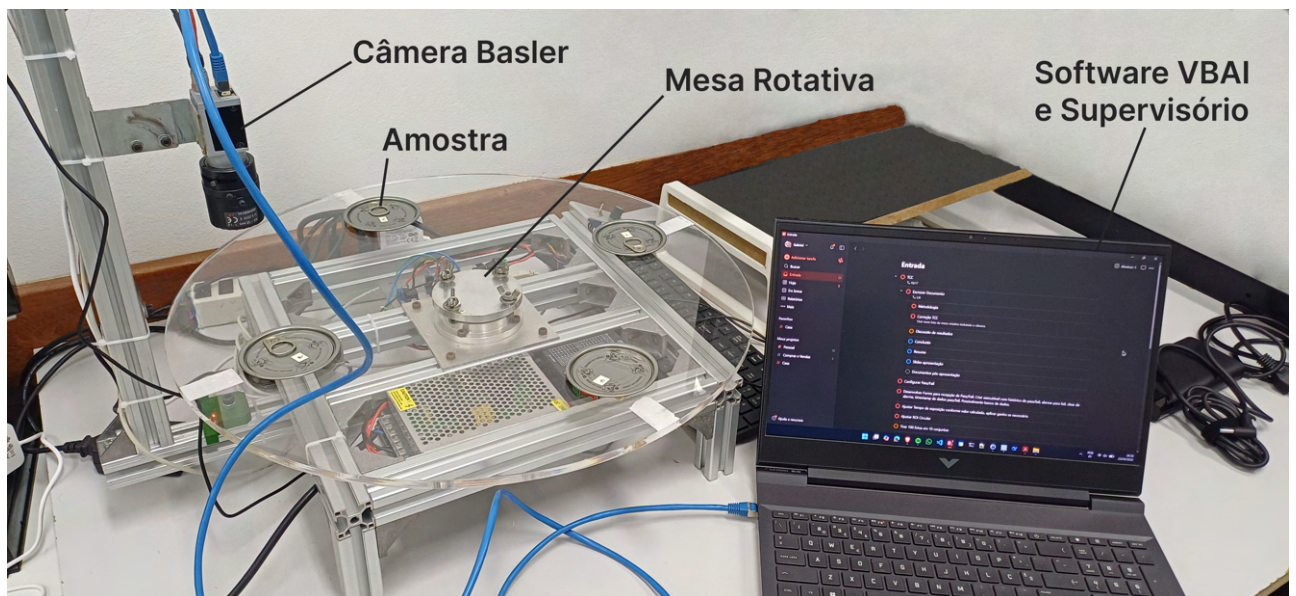
3 DESCRIÇÃO DO SISTEMA PROPOSTO

O sistema, conforme é apresentado na Figura 8, é composto por uma mesa rotativa responsável por transportar as amostras em movimento circular. Essas amostras podem estar em duas condições: intactas ou violadas. Acima da mesa, encontra-se uma câmera industrial posicionada de forma a capturar imagens das amostras à medida que elas passam pelo campo de visão.

As imagens adquiridas pela câmera são enviadas para um computador, onde são processadas pelo *software NI Vision Builder for Automated Inspection*. Nesse ambiente, é realizado o pré-processamento das imagens, seguido da aplicação de um algoritmo de aprendizado de máquina, cujo objetivo é classificar cada amostra como intacta ou violada.

Após a conclusão do processamento no *NI Vision Builder for Automated Inspection*, tanto a imagem capturada quanto as informações resultantes da análise, como a classificação atribuída à amostra e eventuais parâmetros extraídos, são encaminhadas para uma interface gráfica. Essa interface está integrada a um banco de dados, cuja finalidade é armazenar e organizar os registros das amostras processadas, possibilitando rastreabilidade, consulta histórica e análise posterior dos resultados obtidos.

Figura 8 – Sistema montado

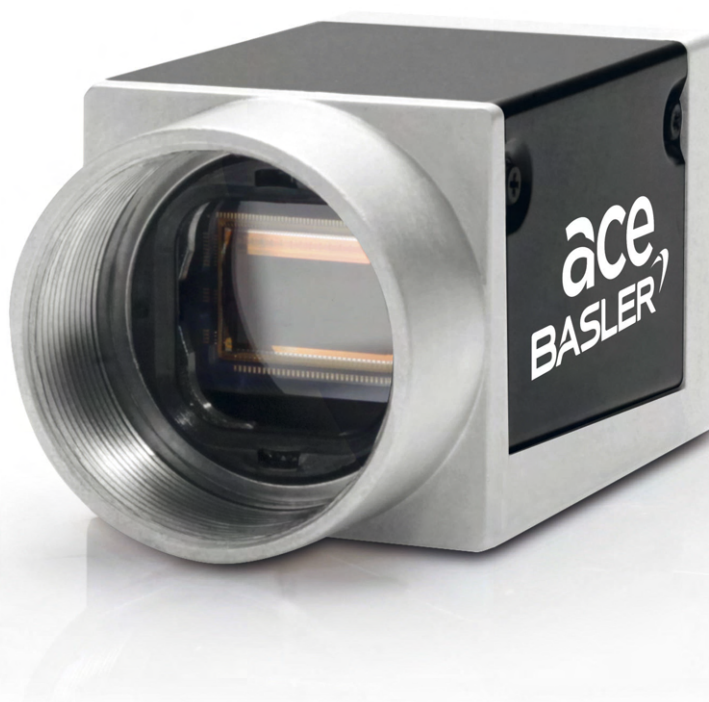


Fonte: Do Autor (2026)

3.1 Câmera Industrial

A câmera utilizada neste projeto é a Basler acA1300-60gc, um modelo de câmera industrial desenvolvido pela Basler AG. Trata-se de uma câmera projetada para aplicações de visão computacional, na qual a aquisição de imagens é realizada de forma contínua e sincronizada com o processo analisado. A Figura 9 mostra a câmera utilizada no projeto.

Figura 9 – Câmera Basler acA1300-60gc



Fonte: (ALTAVISION, 2026)

Diferentemente de câmeras do tipo *smart camera*, esse modelo não possui capacidade de processamento embarcado, sendo necessário o uso de um computador externo para o tratamento das imagens. Nesse contexto, a câmera atua como um dispositivo dedicado à aquisição de imagens, enviando os dados capturados para um sistema de processamento, onde são executados os algoritmos de inspeção e classificação.

A comunicação com o computador é realizada por meio de interface Gigabit Ethernet (GigE), amplamente utilizada em aplicações industriais devido à sua robustez e facilidade de integração. Essa característica permite a transmissão eficiente das imagens, contribuindo para o funcionamento do sistema em tempo real.

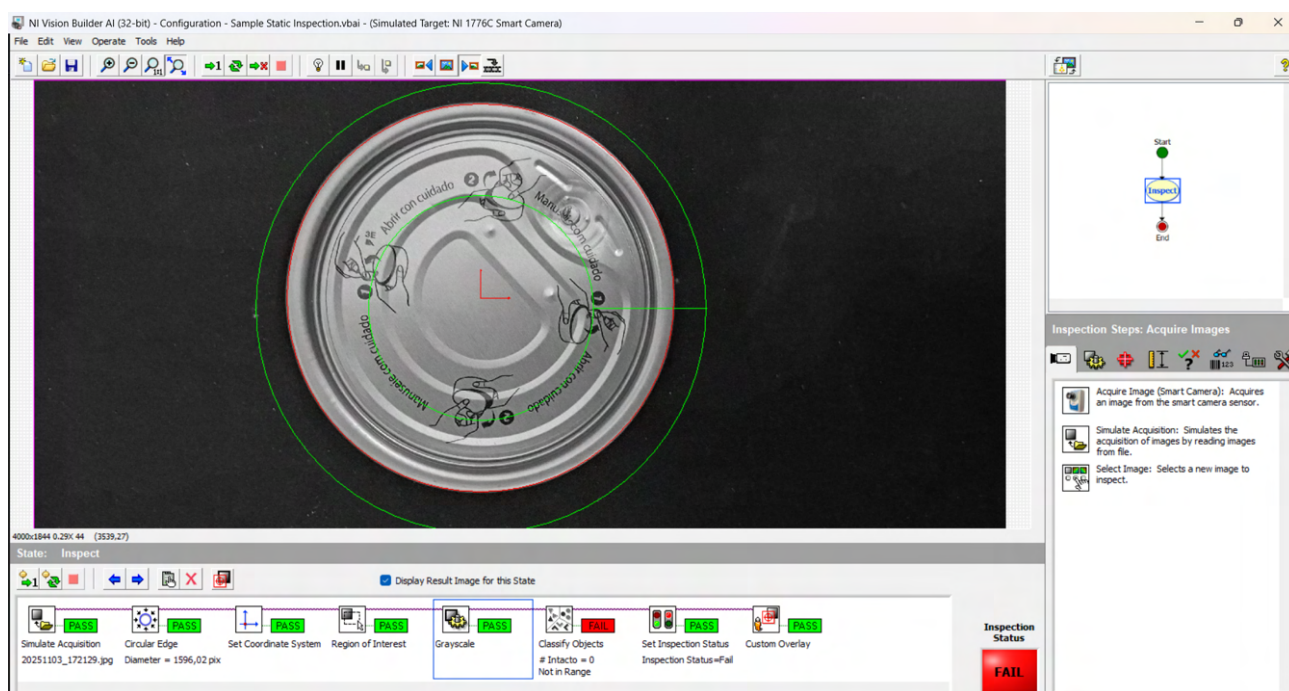
Além disso, a câmera pode operar em diferentes modos de aquisição, como disparo contínuo ou por meio de sinais de *trigger*, o que possibilita sua sincronização com eventos externos, como a passagem das amostras pela região de inspeção. Essa flexibilidade é essencial para garantir que as imagens sejam capturadas no momento adequado do processo.

O Anexo A apresenta as especificações técnicas da câmera, incluindo parâmetros relevantes para os cálculos realizados neste projeto.

3.2 Software

O *software* utilizado neste trabalho é o NI Vision Builder AI 2026, uma ferramenta proprietária da *National Instruments* com integração à câmera Basler acA1300-60gc. Conforme é ilustrada na Figura 10, esse ambiente opera por meio de uma estrutura sequencial baseada em etapas, na qual cada bloco representa uma operação específica, como aplicação de técnicas de pré-processamento, recorte de imagens, análise de padrões geométricos e execução de algoritmos de aprendizado de máquina, entre outras funcionalidades.

Figura 10 – Software NI Vision Builder AI



Fonte: Do Autor (2026)

Cada etapa do processamento permite a configuração de critérios de validação do tipo *pass/fail*, definidos de acordo com os requisitos específicos do projeto. Dessa forma, é possível estabelecer limites ou condições para determinados parâmetros, de modo que, ao serem atendidos, a etapa seja considerada aprovada (*pass*),

ou, caso contrário, reprovada (*fail*). Ao final da execução do fluxo completo, o sistema consolida os resultados de todas as etapas, gerando um status global de *pass/fail*, que representa a condição final da análise realizada.

3.3 Amostra

As amostras utilizadas nesse trabalho consistem em tampas de lata de atum, dividindo-se entre duas classes: tampa intacta e violada. Um exemplar de tampa intacta pode ser vista na Figura 11:

Figura 11 – Exemplar de tampa intacta



Fonte: Do Autor (2026)

A outra classe de amostra consiste em causar uma avaria proposital na tampa, de forma a ter um conjunto de amostras violadas para o projeto. Um exemplar de tampa violada pode ser vista na Figura 12:

Figura 12 – Exemplar de tampa violada

Fonte: Do Autor (2026)

Salienta-se que o critério que define a avaria é a amostra não estar lacrada.

3.4 Mesa Rotativa

Como meio de movimentação das amostras, foi empregada no projeto uma mesa rotativa acionada por motor de passo, permitindo controle preciso de posição e ajuste de velocidade conforme a necessidade experimental. A mesa foi disponibilizada para uso no laboratório CSI do IFSC, Câmpus Florianópolis. Adicionalmente, o sistema conta com um driver de motor de passo responsável pelo acionamento e controle de corrente do motor, bem como um microcontrolador Arduino, utilizado para geração e gerenciamento dos sinais de controle. A Figura 13 ilustra a mesa em questão:

Figura 13 – Mesa rotativa

Fonte: Do Autor (2026)

3.5 Interface Gráfica de Amostragem

A interface gráfica de supervisão foi desenvolvida em C# utilizando a plataforma *Windows Forms* com o ambiente de desenvolvimento *Microsoft Visual Studio*. O objetivo da interface é receber, processar, armazenar e exibir os resultados gerados pelo software *NI Vision Builder AI* (VBAI), permitindo ao operador acompanhar em tempo real o resultado de cada inspeção realizada pelo sistema de visão computacional.

A comunicação entre o VBAI e a interface gráfica foi estabelecida por meio do protocolo TCP/IP. Ao término de cada ciclo de inspeção, o VBAI transmite à interface os dados resultantes da análise realizada, que são então processados e armazenados pela aplicação.

Os dados recebidos a cada inspeção são armazenados em um banco de dados local, contendo as informações relevantes do resultado da análise e a imagem capturada durante o processo. Os registros são mantidos de forma persistente, permitindo a consulta histórica das inspeções realizadas mesmo após o encerramento da aplicação.

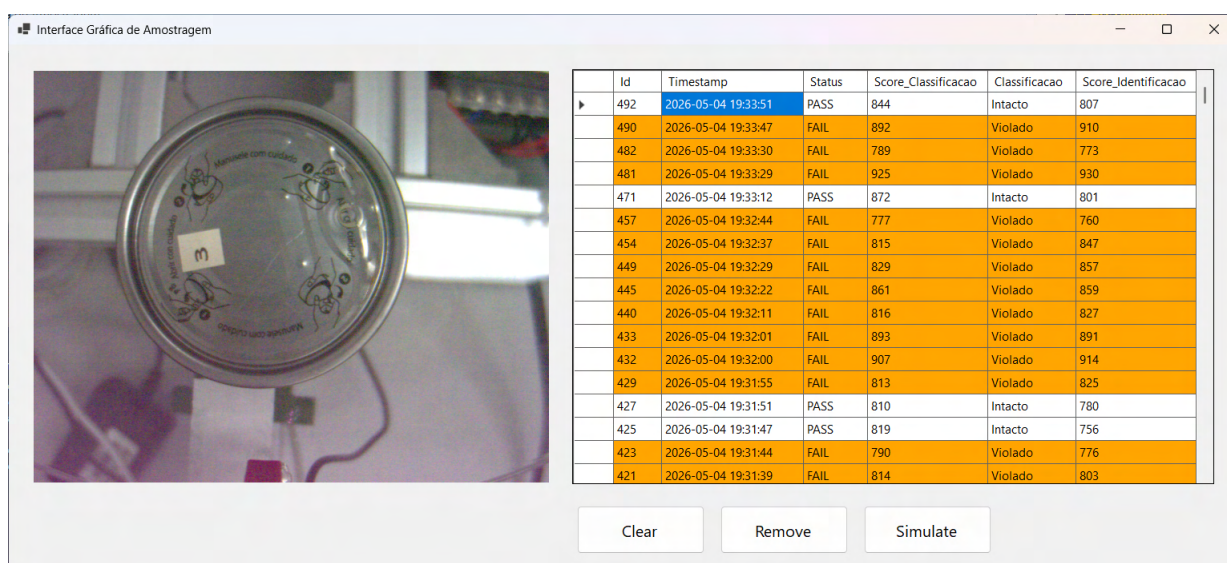
A interface gráfica foi desenvolvida com o objetivo de ser objetiva e funcional, fornecendo ao operador as informações necessárias para acompanhar o processo de inspeção em tempo real. A janela principal é composta por uma tabela que lista todos os registros de inspeção armazenados e por um componente de exibição de imagem, que apresenta a fotografia capturada durante a inspeção correspondente ao registro selecionado pelo operador na tabela.

A interface implementa um sistema de alerta visual para notificar o operador sobre inspeções que requerem atenção. Registros com resultado negativo são automaticamente sinalizados na tabela por meio de uma animação visual, na qual a linha correspondente alterna sua cor de fundo em intervalos regulares. O estado de alerta é mantido mesmo após o encerramento e a reabertura da aplicação, sendo desativado somente mediante confirmação explícita do operador.

A interface disponibiliza ao operador três funcionalidades principais. A primeira permite confirmar um registro selecionado, encerrando o alerta visual associado a ele. A segunda permite excluir permanentemente um registro do banco de dados, mediante confirmação prévia para evitar exclusões acidentais. A terceira funcionalidade, destinada a testes e validação do sistema, permite simular o recebimento de uma inspeção sem a necessidade de acionar o VBAI, verificando assim o funcionamento completo da interface de forma independente.

A Figura 14 ilustra a interface desenvolvida, exibindo as colunas de informações provenientes do VBAI que são persistidas em uma base de dados.

Figura 14 – Interface gráfica de amostragem



Fonte: Do Autor (2026)

3.6 Consideração do Sistema

A etapa de aquisição de imagens do projeto foi conduzida inicialmente com a câmera inteligente NI-1776C, ilustrada na Figura 15. Trata-se de uma câmera inteligente colorida de alta performance para visão computacional, fabricada pela *National Instruments* e equipada com um processador Intel Atom de 1,6 GHz.

Diferentemente de uma câmera industrial convencional, que apenas captura e transmite imagens a um computador hospedeiro para processamento externo, uma câmera inteligente realiza todas essas operações diretamente no próprio dispositivo, incluindo aquisição, processamento e tomada de decisões com base nos resultados da inspeção. Essa característica torna a NI-1776C especialmente adequada para aplicações de inspeção automatizada, eliminando a necessidade de um sistema de processamento externo dedicado.

Ao longo do desenvolvimento do projeto, foram identificados problemas no hardware da câmera NI-1776C que inviabilizaram a continuidade de sua utilização. A origem exata da falha não pôde ser determinada com precisão. Em razão dessa limitação, optou-se pela substituição da câmera pelo modelo da Basler, permitindo assim a continuidade do desenvolvimento do projeto.

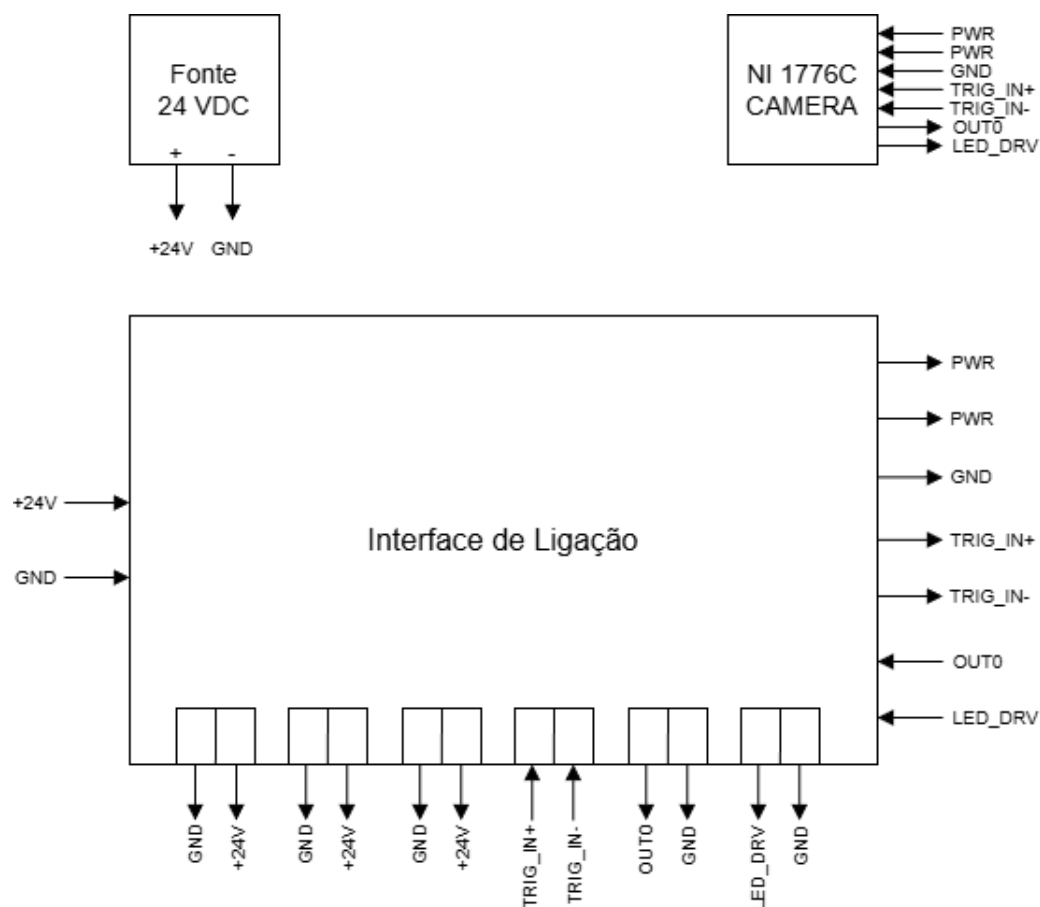
Figura 15 – Câmera NI-1776C

© Artisan Technology Group

Fonte: (ARTISAN TECHNOLOGY GROUP, 2026)

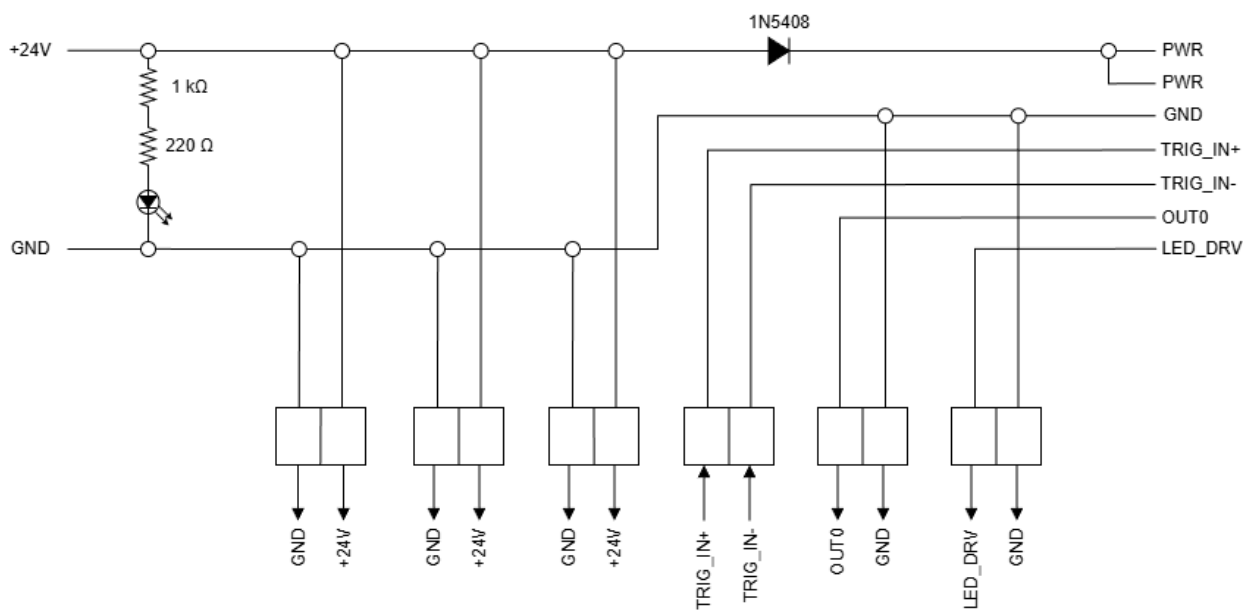
Em conjunto com a câmera NI-1776C, foi desenvolvido um circuito eletrônico dedicado à alimentação e ao roteamento dos sinais de entrada e saída do dispositivo, uma vez que o cabo original responsável por essa função não estava disponível. A câmera NI-1776C opera com tensão de alimentação de 24V, e o circuito desenvolvido foi responsável por fornecer essa tensão de forma adequada, além de realizar o condicionamento e o direcionamento dos sinais digitais entre a câmera e os demais elementos do sistema.

A Figura 16 mostra o esquemático projetado para fazer o roteamento dos sinais e alimentação da câmera.

Figura 16 – Interface para sinais de entrada e saída

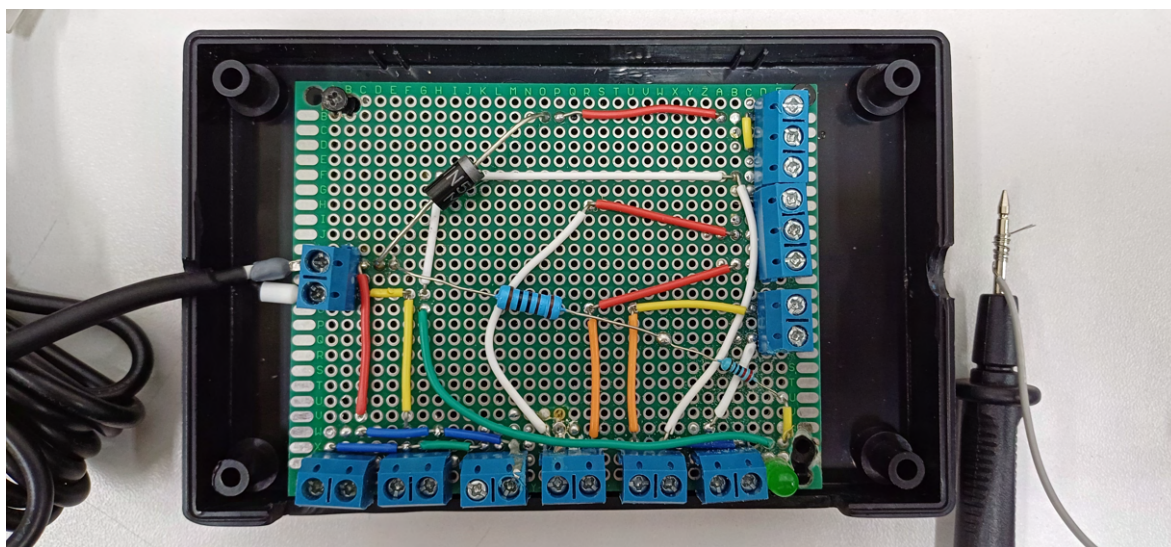
Fonte: Do Autor (2026)

A Figura 17 apresenta os componentes internos da interface. De modo geral, a interface realiza o roteamento dos sinais entre os bornes e a câmera NI-1776C, em ambas as direções. Além disso, foi incorporado um LED para facilitar a visualização da presença de alimentação no circuito. Por fim, adicionou-se um diodo 1N5408 como mecanismo de proteção contra inversão de polaridade na alimentação, garantindo a integridade da câmera.

Figura 17 – Diagrama interno da interface

Fonte: Do Autor (2026)

As Figuras 18 e 19 apresentam a versão final da interface, com todos os componentes devidamente montados e pronta para operação.

Figura 18 – Interface com tampa superior removida

Fonte: Do Autor (2026)

Figura 19 – Interface montada



Fonte: Do Autor (2026)

4 METODOLOGIA

Antes de iniciar a aquisição de imagens, é necessário calcular o tempo máximo de exposição da câmera, tendo em vista que as amostras estarão em movimento durante o processo de inspeção, e a utilização de um tempo de exposição inadequado pode resultar em borrão de movimento nas imagens capturadas, comprometendo a qualidade da análise.

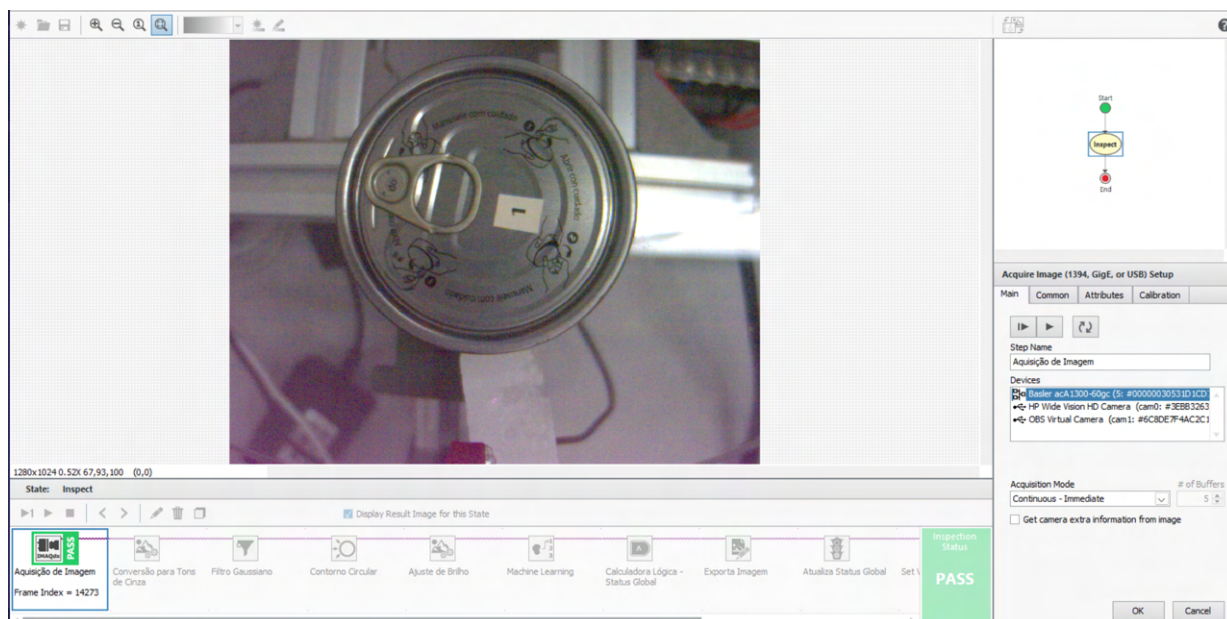
A equação (7), apresentada na seção 2.1.5.2, define a formulação utilizada para esse cálculo. Os Apêndices A e B ilustram, respectivamente, os cálculos da velocidade linear da mesa transportadora e do tempo máximo de exposição obtido a partir dessa velocidade.

O parâmetro B da equação (7), cuja explicação foi atribuída na seção 2.1.5.2, foi definido com valor igual a 2, com o objetivo de evitar com que as imagens se tornassem excessivamente escuras, dispensando assim o uso de ganho de imagem, recurso que, embora aumente o brilho da imagem, introduz ruído indesejado ao sinal capturado. Vale ressaltar que o valor ideal para esse parâmetro seria 1, o qual minimiza ao máximo a distorção causada pelo movimento, porém implicaria em uma imagem mais escura, exigindo compensação por outros meios.

Após a determinação do tempo máximo de exposição, deu-se início à primeira etapa de operações no *software Vision Builder AI*, correspondente à aquisição de imagem. Nos parâmetros dessa etapa, foi configurado o valor de 747 microssegundos para o tempo de exposição, obtido por meio do cálculo apresentado no Apêndice B, sem a adição de ganho de imagem.

A Figura 20 ilustra a o primeiro bloco de operação no *Vision Builder AI*, contendo a aquisição de imagens por meio da câmera Basler.

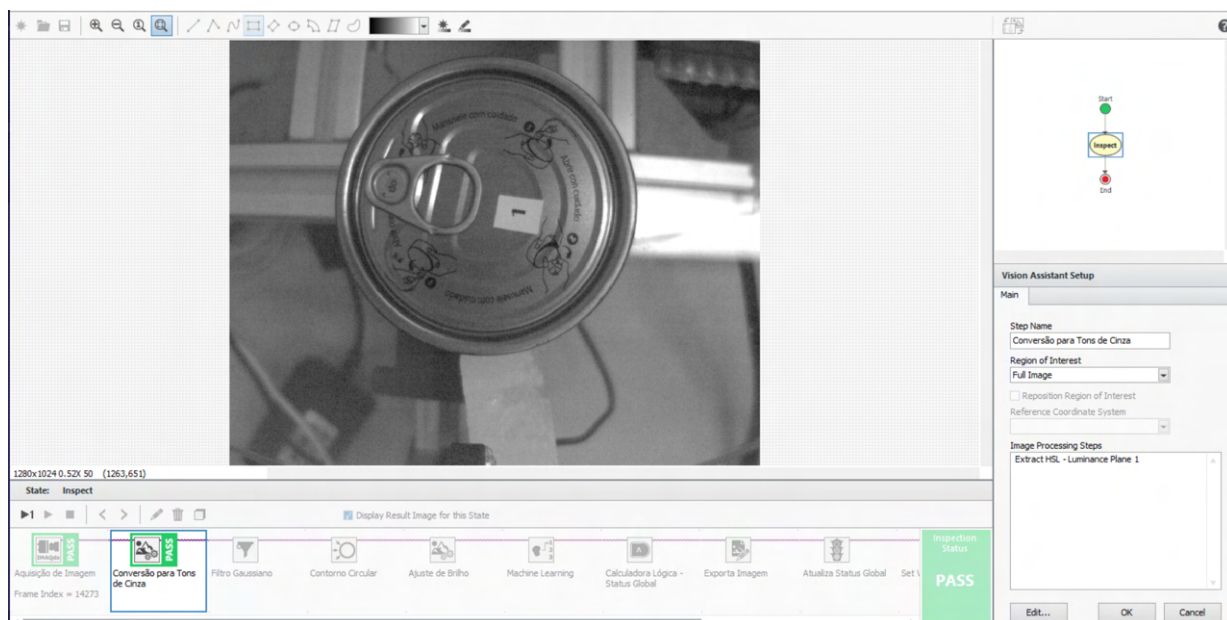
Figura 20 – Etapa de aquisição de imagem



Fonte: Do Autor (2026)

Após a aquisição de imagem, realizou-se a conversão da imagem colorida para tons de cinza, conforme a Figura 21, por meio da extração do plano de luminância no espaço de cores HSL.

Figura 21 – Etapa de conversão para tons de cinza

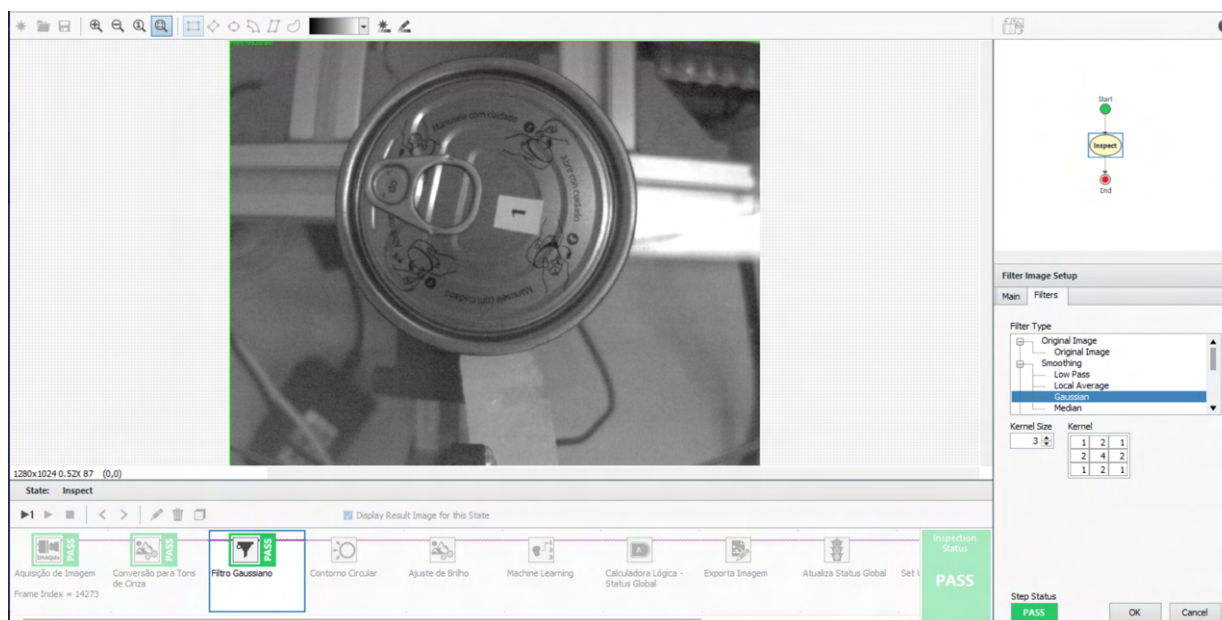


Fonte: Do Autor (2026)

Subsequentemente, aplicou-se um filtro gaussiano à imagem com o objetivo de suavizá-la, reduzindo a presença de ruídos que poderiam prejudicar as etapas

posteriores de processamento. A Figura 22 ilustra o tamanho do *kernel* utilizado, bem como a composição do próprio *kernel* aplicado durante essa etapa.

Figura 22 – Etapa de aplicação de filtro gaussiano

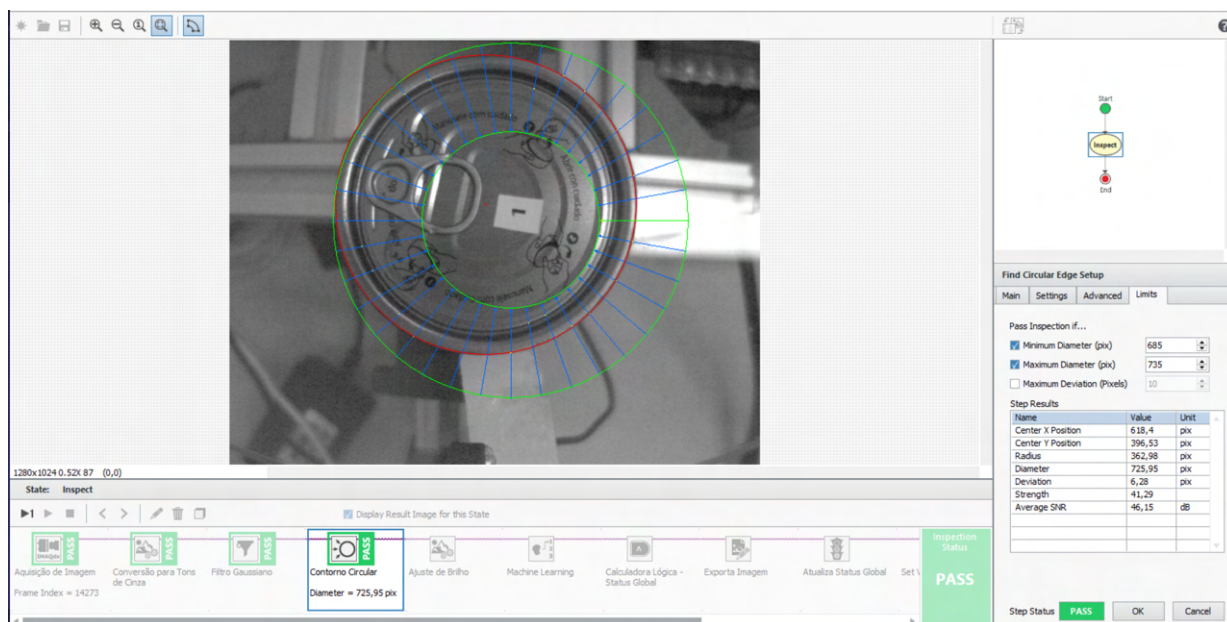


Fonte: Do Autor (2026)

Na etapa seguinte, foi aplicado um detector de contorno circular com o intuito de identificar a presença de uma amostra no campo de visão da câmera. Tendo em vista que a amostra possui formato circular, essa etapa atua como um sensor de presença virtual, sinalizando se há ou não uma amostra na região de interesse da imagem no momento da inspeção.

A Figura 23 ilustra a aplicação do detector de contorno circular em verde, juntamente com a amostra detectada em vermelho.

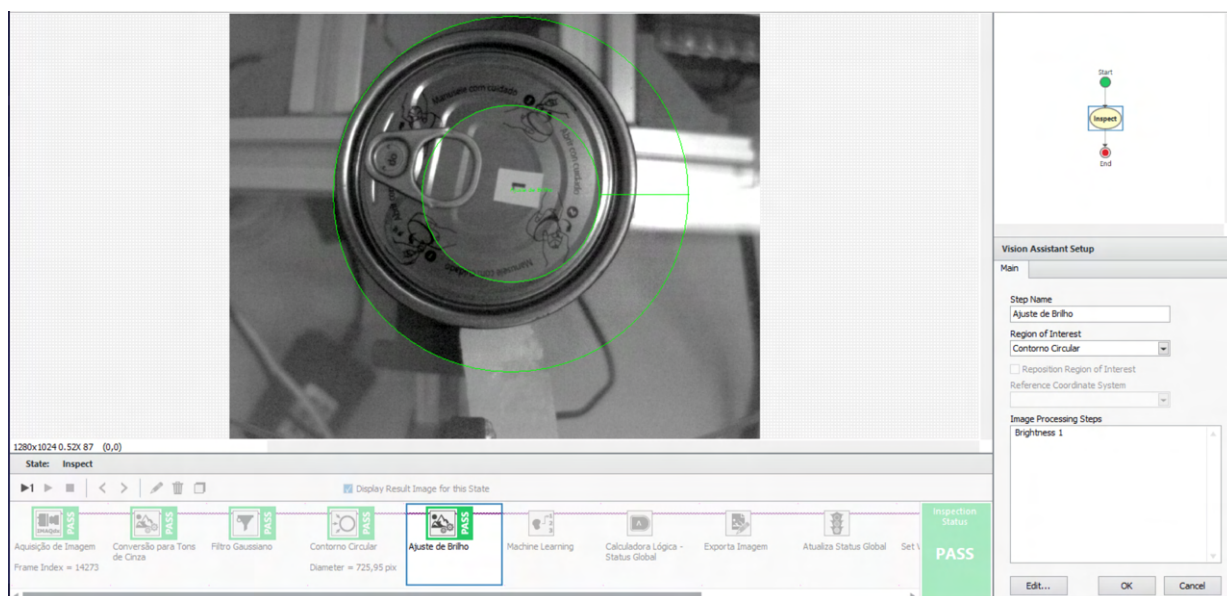
Figura 23 – Etapa de extração de contorno circular



Fonte: Do Autor (2026)

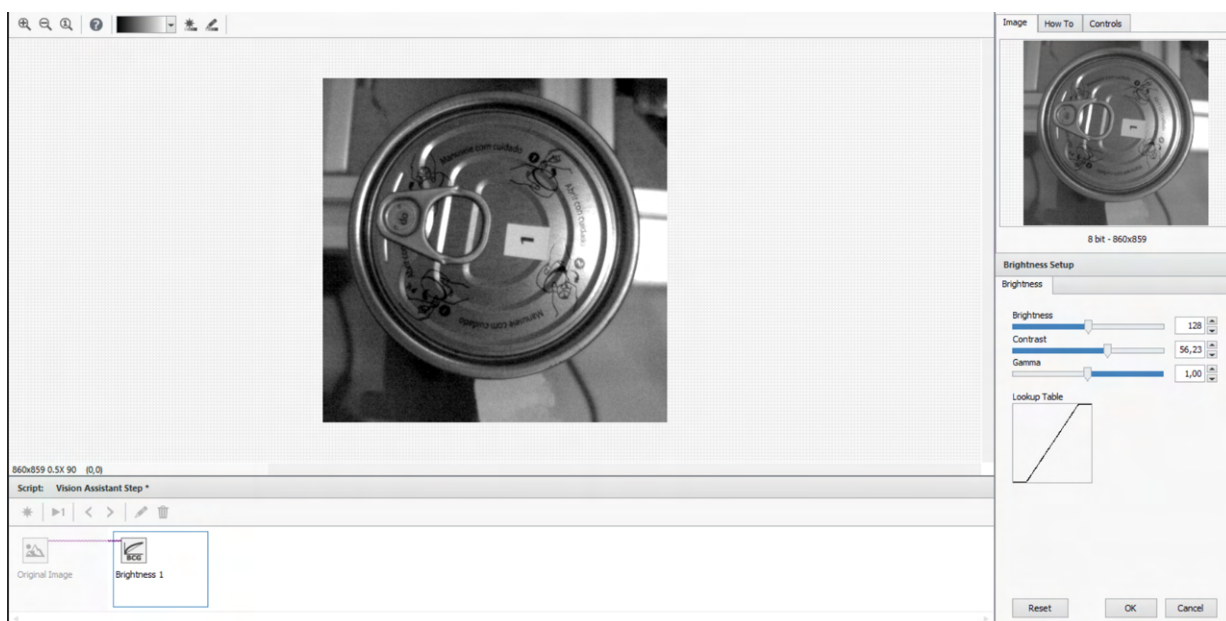
Em seguida, realizou-se o ajuste de brilho e contraste da imagem, com o objetivo de padronizar as condições de iluminação entre as amostras inspecionadas e realçar as características visuais relevantes para as etapas posteriores de análise. A Figura 24 apresenta a etapa configurada no *software* VBAI, enquanto a Figura 25 detalha os parâmetros de ajuste de brilho e contraste utilizados.

Figura 24 – Etapa de ajuste de brilho



Fonte: Do Autor (2026)

Figura 25 – Detalhes do ajuste de brilho



Fonte: Do Autor (2026)

Na etapa de classificação, utilizando aprendizado de máquina supervisionado, foram criadas duas classes distintas para o treinamento do classificador: intacta e violada. Para o treinamento, foi empregada a técnica de validação cruzada, na qual as 200 imagens capturadas foram divididas em 10 subconjuntos, denominados *folds*, cada um contendo o mesmo número de amostras das classes intacta e violada. A validação cruzada oferece a vantagem de utilizar todas as amostras disponíveis tanto para treinamento quanto para validação, de forma alternada entre os *folds*, reduzindo o risco de sobreajuste do modelo e proporcionando uma estimativa mais confiável e representativa do desempenho real do classificador em relação a uma simples divisão estática entre conjuntos de treino e teste.

Esta etapa aplica uma binarização na imagem, realizando a análise em uma porção específica da amostra. O limiar de binarização foi ajustado de forma que a região da tampa fosse segmentada de maneira satisfatória tanto no estado intacto quanto no estado violado, permitindo que o algoritmo identificasse as diferenças morfológicas entre as duas condições.

O *software* oferece três algoritmos de classificação disponíveis para uso: o *Nearest Neighbor*, o *K-Nearest Neighbor* e o *Minimum Mean Distance*. Adicionalmente, permite a utilização de três métricas de distância: distância euclidiana, distância máxima e distância de *Manhattan*. Com o objetivo de identificar a combinação de melhor desempenho, foi realizado um treinamento sistemático utilizando todas as combinações possíveis entre algoritmos e métricas de distância, totalizando nove configurações avaliadas.

Ao aplicar uma amostra de inspeção ao classificador treinado, o bloco retorna a classe inferida juntamente com dois indicadores de desempenho: o *score* de classificação e o *score* de identificação. O *score* de classificação indica o grau com que a classe atribuída representa a amostra de entrada em comparação com as demais classes, sendo definido conforme a Equação 25:

$$\text{Score de Classificação} = \left(1 - \frac{d_1}{d_2}\right) \times 1000 \quad (25)$$

onde d_1 é a distância até a classe mais próxima e d_2 é a distância até a segunda classe mais próxima.

Intuitivamente, representa o quanto a classe atribuída é melhor que as outras classes concorrentes.

O *score* de identificação, por sua vez, indica a similaridade entre a amostra de entrada e a classe atribuída, sendo definido conforme a Equação 26:

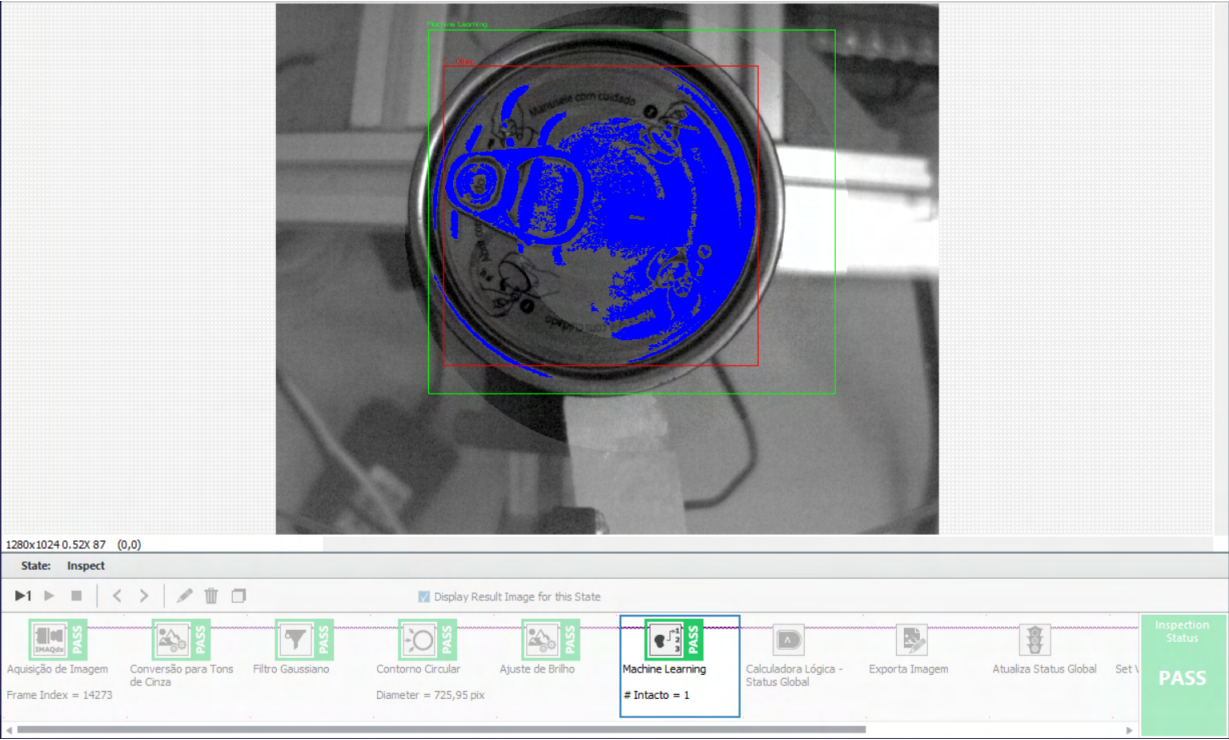
$$\text{Score de Identificação} = (1 - d) \times 1000 \quad (26)$$

onde d é a distância normalizada entre o vetor de entrada e a classe atribuída, calculada como a razão entre a distância da amostra à sua classe e o fator de normalização, definido como a distância máxima entre classes.

Ambos os *scores* fornecem, portanto, informações complementares sobre a qualidade da classificação realizada. No presente projeto, foram consideradas confiáveis apenas as classificações cujos *scores* de identificação e classificação fossem superiores a 750, garantindo que somente resultados com elevado grau de certeza fossem utilizados nas etapas subsequentes do sistema de inspeção.

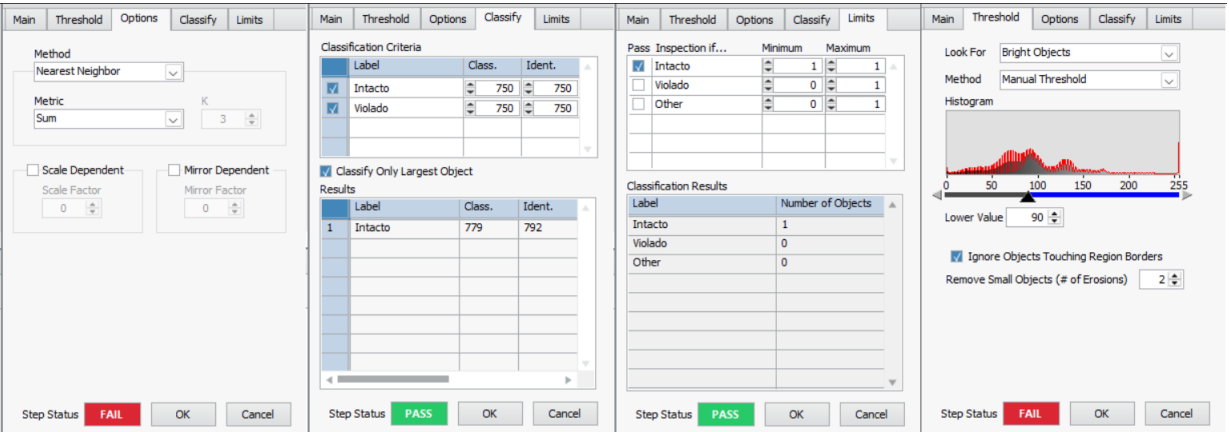
As Figuras 26 e 27 mostram, respectivamente, o bloco de processamento de *machine learning* com a amostra binarizada e as janelas de configurações.

Figura 26 – Etapa de classificação da amostra



Fonte: Do Autor (2026)

Figura 27 – Configurações do classificador



Fonte: Do Autor (2026)

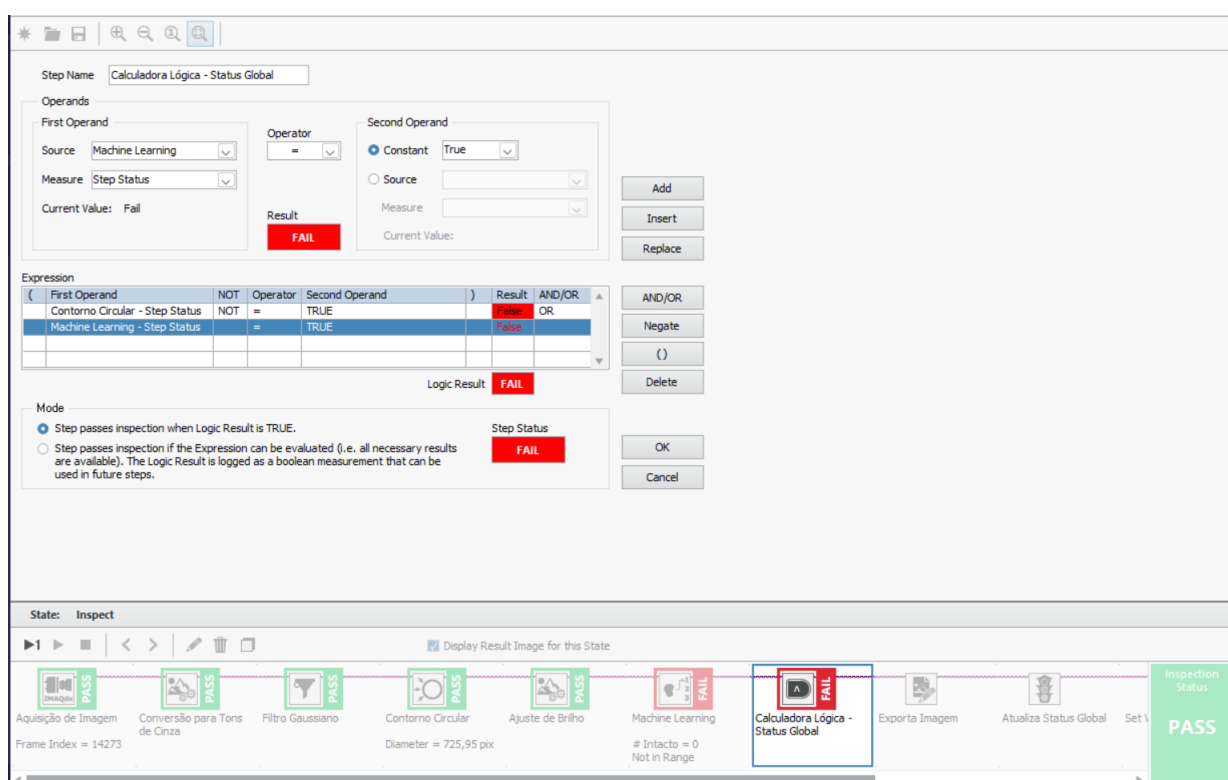
A etapa subsequente consiste em uma calculadora lógica, responsável por determinar o status global da inspeção por meio de álgebra booleana. A lógica implementada considera duas variáveis de entrada: a presença de amostra no campo de visão da câmera, detectada pelo detector de contorno circular, e o resultado da classificação da amostra.

O Apêndice C apresenta a tabela verdade correspondente, contemplando todas as combinações possíveis dessas variáveis de entrada e suas respectivas saídas.

De acordo com a tabela verdade, quando a entrada “Amostra presente no Frame” é igual a 0, ou seja, não há amostra no frame, a saída “Status da Inspeção” assume o valor 1, sinalizando que a inspeção está sem problemas, uma vez que não há nada a ser processado nesse momento. Já quando a entrada “Amostra presente no Frame” é igual a 1 e a entrada “Amostra intacta” é igual a 0, ou seja, a amostra está violada, a saída assume o valor 0, indicando falha na inspeção. Por fim, quando ambas as entradas estão em 1, significa que há uma amostra presente no frame e que ela está intacta, resultando em um status de inspeção sem falhas.

A equação booleana resultante dessa análise foi então aplicada à calculadora lógica configurada nessa etapa, conforme a Figura 28.

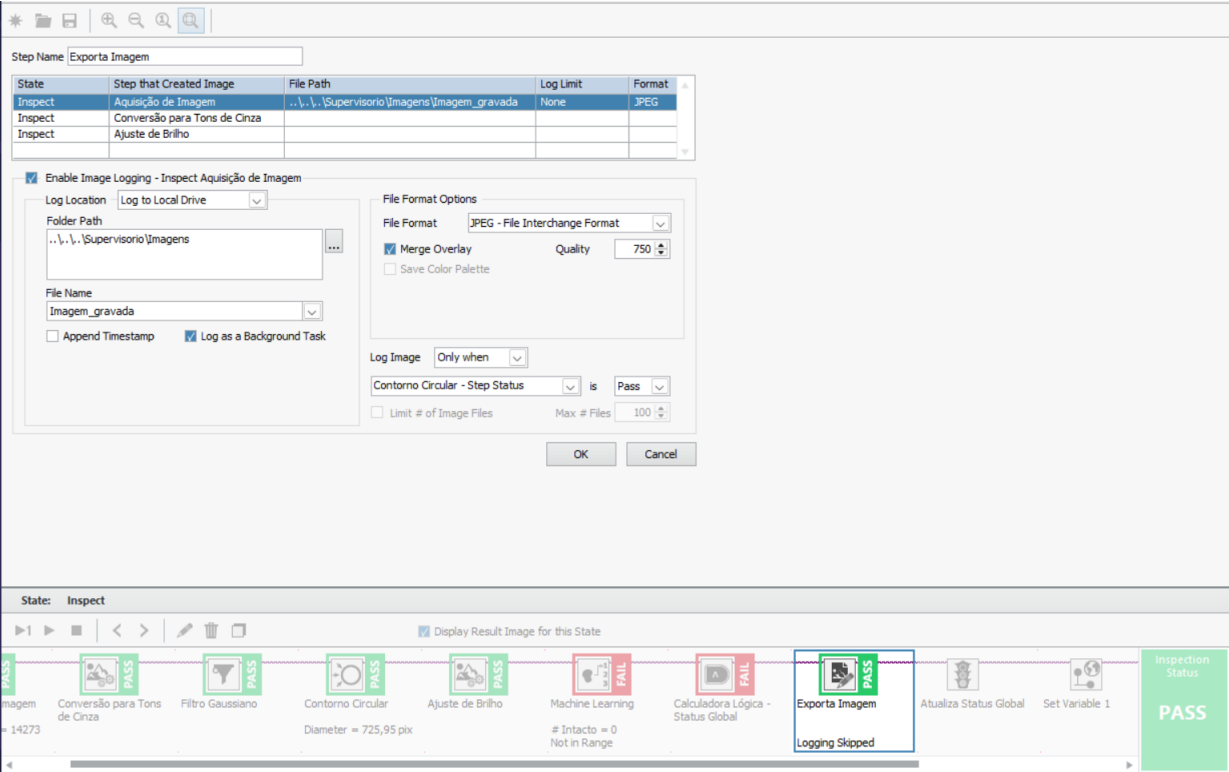
Figura 28 – Calculadora lógica



Fonte: Do Autor (2026)

Logo após, realizou-se a exportação da imagem capturada para o sistema supervisorio de amostragem, ilustrada na Figura 29. Nessa etapa, a imagem é armazenada em uma pasta específica no computador, da qual o sistema supervisorio a recupera para exibição e armazenamento no banco de dados. Vale ressaltar que a exportação da imagem ocorre somente quando a presença de amostra é detectada no campo de visão da câmera, evitando o armazenamento desnecessário de imagens sem amostra.

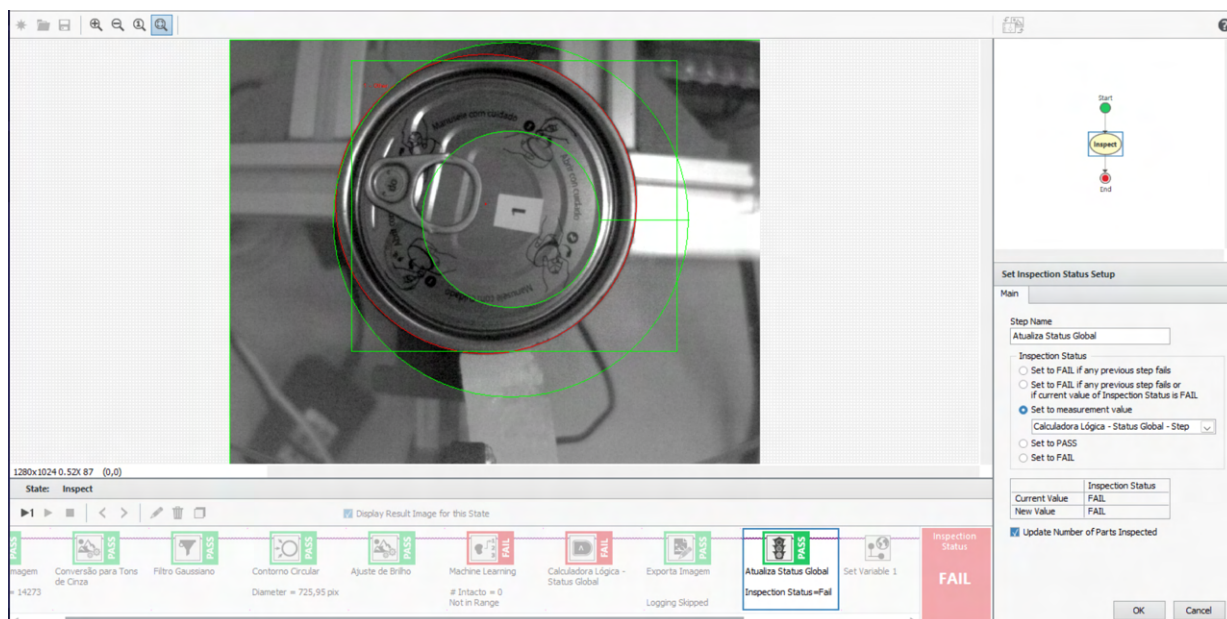
Figura 29 – Etapa de exportação de imagem



Fonte: Do Autor (2026)

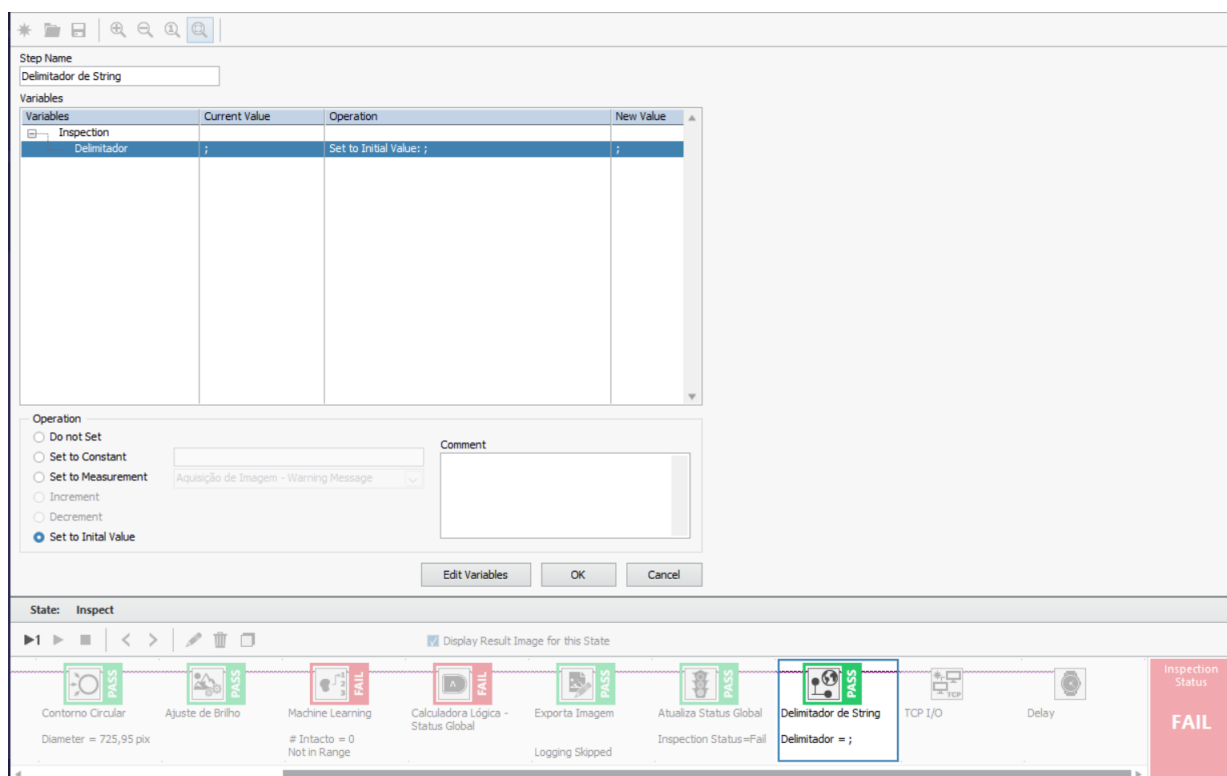
Adiante, foi realizada uma atualização do status global da inspeção. Para isso, o resultado obtido pela calculadora lógica foi diretamente atribuído a esse status, de forma que ele representa um reflexo da tabela verdade apresentada no Apêndice C. Dessa forma, o status assume o valor *PASS* quando não há amostra presente no campo de visão da câmera, ou quando há amostra e esta é classificada como intacta. Em caso contrário, ou seja, quando há amostra presente e esta é classificada como violada, o status é indicado como *FAIL*. A Figura 30 ilustra o bloco de processamento dessa etapa no VBAI.

Figura 30 – Atualização do status global



Fonte: Do Autor (2026)

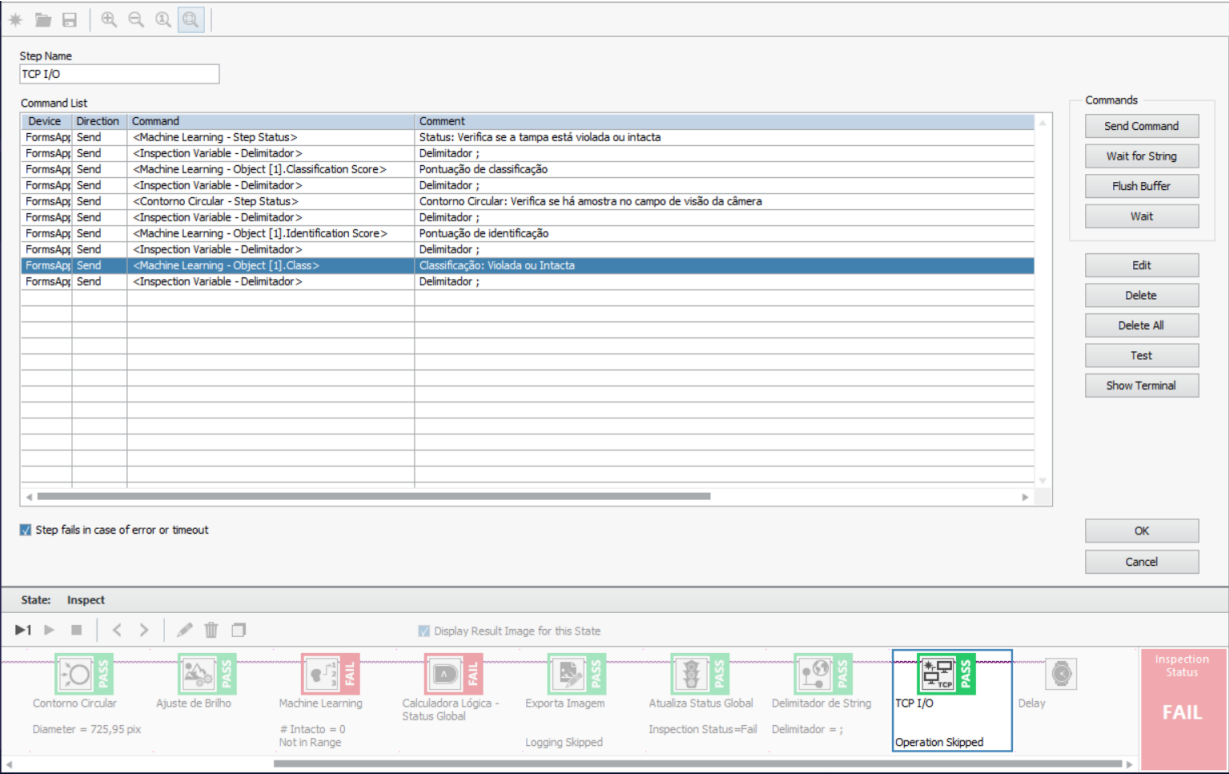
Posteriormente, foi criada uma variável denominada Delimitador, contendo o caractere ponto e vírgula, a qual é utilizada como separador entre os valores transmitidos ao sistema supervisorio na etapa seguinte de comunicação TCP. A configuração dessa etapa é mostrada pela Figura 31.

Figura 31 – Delimitador de *string*

Fonte: Do Autor (2026)

O bloco seguinte de operações é responsável pelo envio dos dados resultantes da inspeção ao sistema supervisorio por meio do protocolo TCP, conforme a Figura 32. Os dados transmitidos são, na ordem de envio: o status global da inspeção, a pontuação de classificação, o status de presença de amostra obtido pelo detector de contorno circular, a pontuação de identificação e a classificação textual da amostra, que indica se a mesma é intacta ou violada. Cada dado é separado pelo caractere delimitador definido na etapa anterior, de forma a facilitar a decodificação e o processamento correto das informações no lado do sistema supervisorio.

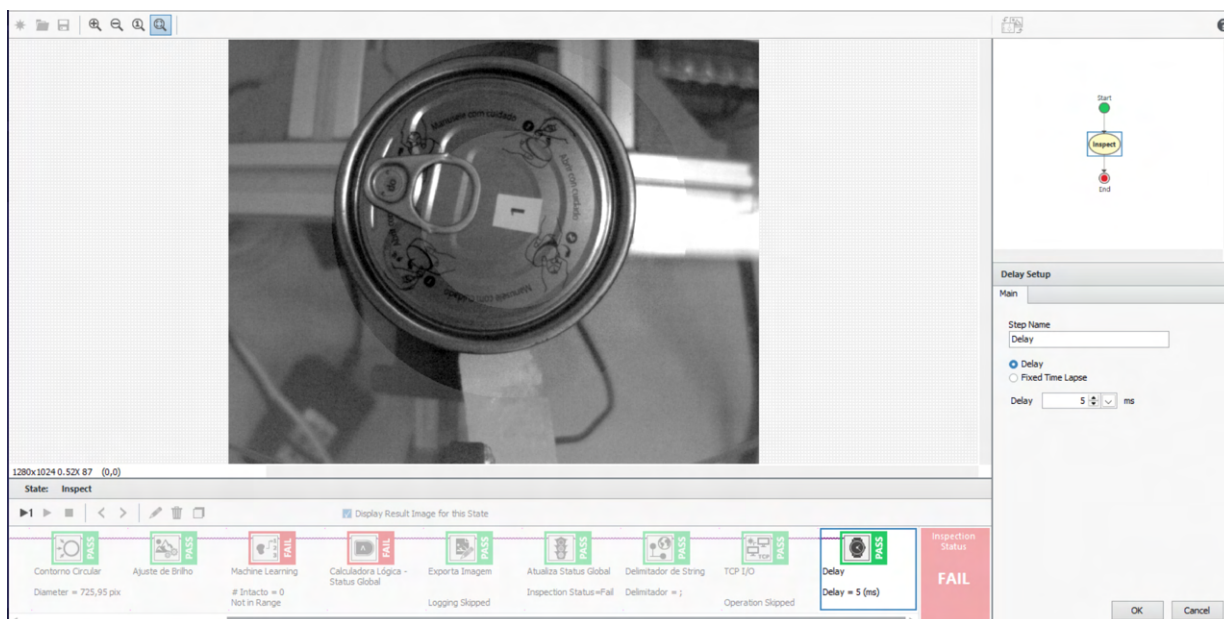
Figura 32 – Envio de dados via TCP



Fonte: Do Autor (2026)

Por fim, foi adicionado um atraso de 5 milissegundos entre os ciclos de aquisição de imagem, com o objetivo de evitar a captura de imagens duplicadas de uma mesma amostra, garantindo que cada ciclo de inspeção corresponda a uma amostra distinta presente no campo de visão da câmera. Essa configuração é apresentada pela Figura 33.

Figura 33 – Aplicação de delay na aquisição



Fonte: Do Autor (2026)

Adicionalmente ao sistema de inspeção, foi implementado um sistema supervisor responsável por receber os dados transmitidos pelo VBAI via protocolo TCP e armazená-los de forma persistente em um banco de dados local. A cada ciclo de inspeção cujo resultado de presença de amostra seja positivo, um novo registro é inserido no banco de dados contendo a data e hora da inspeção, o status global, as pontuações de classificação e identificação, a classificação textual da amostra e a imagem capturada durante o processo.

A interface gráfica do sistema supervisor, ilustrada na Figura 14, exibe os registros armazenados em uma tabela, permitindo ao operador acompanhar o histórico de inspeções realizadas. Ao selecionar um registro na tabela, a imagem correspondente à inspeção é recuperada do banco de dados e exibida em um componente dedicado na interface, possibilitando a análise visual da amostra inspecionada.

O sistema conta com um mecanismo de alarme visual destinado a notificar o operador sobre inspeções com resultado negativo. Registros cujo status seja igual a *FAIL*, indicando a detecção de uma amostra violada, são sinalizados na tabela por meio de uma animação de alternância de cor, na qual a linha correspondente pisca entre as cores amarelo e laranja em intervalos regulares. O estado de alarme é persistido no banco de dados, de forma que registros não confirmados retomam a sinalização visual automaticamente ao reiniciar a aplicação.

Para a interação do operador com o sistema, a interface disponibiliza três funcionalidades por meio de botões dedicados. O botão *Remove* permite excluir permanentemente um registro selecionado do banco de dados, sendo útil em situações em

que a aquisição de imagem tenha sido realizada de forma inadequada ou em que a classificação tenha sido executada de maneira incorreta. O botão *Clear* desabilita o alarme visual do registro selecionado, sinalizando ao sistema que o operador tomou ciência do resultado da inspeção. Por fim, o botão *Simulate* foi desenvolvido com finalidade de depuração, removendo todos os registros existentes no banco de dados e inserindo um registro pré-preenchido com status *FAIL*, permitindo verificar o funcionamento completo do sistema supervisor de forma independente do VBAI.

5 APRESENTAÇÃO DOS RESULTADOS

O presente capítulo expõe os resultados obtidos ao longo do desenvolvimento deste trabalho, cujo objeto central é a aplicação de técnicas de machine learning aliadas à visão computacional para a detecção automática de falhas em objetos. A motivação para a adoção dessa abordagem reside na dificuldade inerente ao desenvolvimento de algoritmos tradicionais, sem aprendizado de máquina, que sejam suficientemente robustos para lidar com as variações naturais presentes nas amostras analisadas, tais como reflexos metálicos, irregularidades superficiais e variações de ângulo de captura.

Na sequência, serão apresentados os resultados obtidos por meio de experimentos sistemáticos, nos quais diferentes algoritmos e métricas de distância foram testados em conjunto com o objetivo de identificar a combinação mais eficiente.

5.1 Análise e discussão dos resultados

Os resultados a seguir são apresentados em uma tabela que indica, para cada execução, o *fold* utilizado como conjunto de validação e a respectiva acurácia obtida pelo classificador. Vale destacar que, quando a coluna *Fold* indica "*Fold 4*", por exemplo, isso indica que o *Fold 4* foi reservado para validação, enquanto os 9 *folds* restantes foram empregados no treinamento do modelo.

Os primeiros resultados apresentados referem-se aos testes conduzidos com o algoritmo *K-Nearest Neighbor*. Dentre as diferentes métricas de distância avaliadas, o melhor desempenho alcançou uma acurácia média de 71%, valor que não se mostrou plenamente satisfatório. As Figuras 34, 35 e 36 ilustram as tabelas de resultados utilizando o algoritmo *K-Nearest Neighbor* com as distâncias euclidiana, soma e máxima, respectivamente.

Figura 34 – Algoritmo *K-Nearest Neighbor* e distância euclidiana

K-Nearest Neighbor + Distância Euclidiana	
Fold	Acurácia
Fold 1	85%
Fold 2	85%
Fold 3	65%
Fold 4	75%
Fold 5	60%
Fold 6	60%
Fold 7	70%
Fold 8	70%
Fold 9	75%
Fold 10	65%
Média	71%

Fonte: Do Autor (2026)

Figura 35 – Algoritmo *K-Nearest Neighbor* e distância soma

K-Nearest Neighbor + Distância Soma	
Fold	Acurácia
Fold 1	75%
Fold 2	85%
Fold 3	60%
Fold 4	75%
Fold 5	65%
Fold 6	50%
Fold 7	65%
Fold 8	65%
Fold 9	65%
Fold 10	70%
Média	68%

Fonte: Do Autor (2026)

Figura 36 – Algoritmo *K-Nearest Neighbor* e distância máxima

K-Nearest Neighbor + Distância Máxima	
Fold	Acurácia
Fold 1	80%
Fold 2	80%
Fold 3	60%
Fold 4	65%
Fold 5	50%
Fold 6	50%
Fold 7	65%
Fold 8	60%
Fold 9	65%
Fold 10	60%
Média	64%

Fonte: Do Autor (2026)

A série de resultados seguinte refere-se aos testes conduzidos com o algoritmo *Nearest Neighbor* e suas diferentes métricas de distância, conforme ilustram as Figuras 37, 38 e 39. Ao contrário do *K-Nearest Neighbor*, este algoritmo demonstrou desempenho consideravelmente inferior, atingindo uma acurácia média de até 22%.

Figura 37 – Algoritmo *Nearest Neighbor* e distância euclidiana

Nearest Neighbor + Distância Euclidiana	
Fold	Acurácia
Fold 1	25%
Fold 2	25%
Fold 3	40%
Fold 4	20%
Fold 5	15%
Fold 6	10%
Fold 7	20%
Fold 8	30%
Fold 9	15%
Fold 10	25%
Média	23%

Fonte: Do Autor (2026)

Figura 38 – Algoritmo *Nearest Neighbor* e distância soma

Nearest Neighbor + Distância Soma	
Fold	Acurácia
Fold 1	25%
Fold 2	25%
Fold 3	40%
Fold 4	25%
Fold 5	15%
Fold 6	15%
Fold 7	15%
Fold 8	30%
Fold 9	15%
Fold 10	25%
Média	23%

Fonte: Do Autor (2026)

Figura 39 – Algoritmo *Nearest Neighbor* e distância máxima

Nearest Neighbor + Distância Máxima	
Fold	Acurácia
Fold 1	25%
Fold 2	25%
Fold 3	35%
Fold 4	20%
Fold 5	20%
Fold 6	10%
Fold 7	15%
Fold 8	30%
Fold 9	15%
Fold 10	25%
Média	22%

Fonte: Do Autor (2026)

Por fim, foram conduzidos testes com o algoritmo *Minimum Mean Distance* e suas diferentes combinações de métricas de distância, conforme as Figuras 40, 41 e 42. Dentre todos os algoritmos avaliados, este apresentou o pior desempenho, atingindo uma acurácia média de até 10%.

Figura 40 – Algoritmo *Minimum Mean Distance* e distância euclideana

Minimum Mean Distance + Distância Euclideana	
Fold	Acurácia
Fold 1	15%
Fold 2	10%
Fold 3	10%
Fold 4	10%
Fold 5	25%
Fold 6	20%
Fold 7	5%
Fold 8	20%
Fold 9	10%
Fold 10	10%
Média	14%

Fonte: Do Autor (2026)

Figura 41 – Algoritmo *Minimum Mean Distance* e distância soma

Minimum Mean Distance + Distância Soma	
Fold	Acurácia
Fold 1	10%
Fold 2	10%
Fold 3	5%
Fold 4	5%
Fold 5	20%
Fold 6	15%
Fold 7	5%
Fold 8	15%
Fold 9	5%
Fold 10	5%
Média	10%

Fonte: Do Autor (2026)

Figura 42 – Algoritmo *Minimum Mean Distance* e distância máxima

Minimum Mean Distance + Distância Máxima	
Fold	Acurácia
Fold 1	25%
Fold 2	20%
Fold 3	25%
Fold 4	15%
Fold 5	30%
Fold 6	30%
Fold 7	5%
Fold 8	35%
Fold 9	25%
Fold 10	10%
Média	22%

Fonte: Do Autor (2026)

Dentre todos os algoritmos avaliados, o *K-Nearest Neighbor* com distância euclidiana demonstrou os melhores resultados para este projeto. No entanto, o desempenho obtido ainda deixa a desejar, uma vez que a acurácia média alcançada permanece muito abaixo do esperado para uma aplicação prática no contexto industrial.

Ressalta-se que o *software Vision Builder AI* não permite selecionar manualmente as características que compõem o vetor de características. Isso implica que as características extraídas por padrão pelo *software* não possibilitam uma diferenciação adequada entre tampas intactas e violadas. Com uma seleção criteriosa das características que efetivamente discriminam os dois estados da tampa, os vetores resultantes tendem a se tornar mais coesos e separáveis entre si, viabilizando uma classificação mais precisa e confiável.

De acordo com o manual do *software VBAI*, as características pré-definidas no vetor são mencionadas da seguinte maneira:

- Característica 1 descreve a circularidade da amostra.
- Característica 2 descreve o grau de alongamento da amostra.
- Característica 3 representa a convexidade do formato da amostra.
- Característica 4 fornece uma descrição mais detalhada da convexidade do formato da amostra.

- Característica 5 é utilizada para a discriminação de amostras com furos.
- Característica 6 é utilizada para uma discriminação mais detalhada de amostras com furos.
- Característica 7 representa a dispersão espacial da amostra.
- Característica 8 representa a esbeltez da amostra.

6 CONSIDERAÇÕES FINAIS

O presente trabalho propôs o desenvolvimento de um sistema de classificação baseado em *machine learning* aplicado à visão computacional, com o objetivo de identificar em tempo real amostras intactas ou violadas. A proposta inicial demonstrou-se promissora, evidenciando a viabilidade de se obter um algoritmo capaz de realizar tal classificação de maneira automatizada. Contudo, mesmo diante do melhor resultado alcançado, com uma taxa média geral de acertos de 71%, a solução não se mostrou plenamente satisfatória para aplicação prática. Tal limitação está atrelada, sobretudo, à restrição imposta pelo *software* utilizado, que não permitiu a manipulação manual do vetor de características, fator de fundamental importância para a precisão e o desempenho de classificadores baseados em *machine learning*. Dessa forma, conclui-se que, embora os resultados obtidos indiquem um caminho promissor, avanços metodológicos e tecnológicos, como os discutidos nas perspectivas de trabalhos futuros, são necessários para que a solução atinja um nível de acurácia adequado às exigências de um ambiente produtivo real.

6.1 Sugestões para trabalhos futuros

Como perspectivas para trabalhos futuros, recomenda-se a utilização de ferramentas que possibilitem a seleção manual das características empregadas no classificador, como o *LabVIEW*. Adicionalmente, vislumbra-se a adoção de abordagens mais sofisticadas, como o *deep learning*, paradigma no qual o próprio sistema é capaz de identificar automaticamente as características mais relevantes e avaliá-las de forma iterativa, buscando a combinação que proporcione o melhor desempenho na detecção de falhas.

REFERÊNCIAS

- ALTAVISION. **Basler acA1300-60gc Datasheet**. 2026. Disponível em: <http://www.altavision.com.br/Datasheets/Basler/acA1300-60gc.html>. Acesso em: 22 abr 2026. 39, 79
- ARTISAN TECHNOLOGY GROUP. **NI-1776C Value Smart Camera for Embedded Vision**. 2026. Disponível em: <https://www.artisanng.com/TestMeasurement/89480-7/National-Instruments-NI-1776C-Value-Smart-Camera-for-Embedded-Vision>. Acesso em: 30 mar 2026. 45
- BOVIK, A. **Handbook of Image and Video Processing**. [S.l.]: Academic Press, 2000. 22
- BURGER, W.; BURGE, M. J. **Digital Image Processing: An Algorithmic Introduction Using Java**. 2. ed. [S.l.]: Springer, 2016. 22, 23, 24
- CANON. **Understanding focal length in photography**. 2026. Disponível em: <https://www.canon-europe.com/pro/infobank/understanding-focal-length/>. Acesso em: 25 mar 2026. 29
- CHOUINARD, J. **Imaging Basics - Calculating Exposure Time for Machine Vision Cameras**. 2018. Disponível em: <https://www.1stvision.com/machine-vision-solutions/2018/06/industrial-camera-exposure-time.html>. Acesso em: 20 abr 2026. 25
- DATTA, S.; DAVIM, J. P. **Machine Learning in Industry**. [S.l.]: Springer Nature Switzerland AG, 2022. 13
- DOMINGOS, P. **A Few Useful Things to Know About Machine Learning**. *Communications of the ACM*, v. 55, n. 10, 2012. 30, 37
- DUDA, R. O. **Pattern Recognition for HCI**. 1997. Department of Electrical Engineering, San Jose State University. Disponível em: https://www.cs.princeton.edu/courses/archive/fall08/cos436/Duda/PR_home.htm. Acesso em: 07 mai 2026. 32
- DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern Classification**. 2. ed. [S.l.]: John Wiley and Sons, 2001. 33
- FIRDOUSI, R.; PARVEEN, S. **Local Thresholding Techniques in Image Binarization**. *International Journal Of Engineering And Computer Science*, v. 3, 2014. 19, 20
- FORSYTH, D. A.; PONCE, J. **Computer Vision: A Modern Approach**. 2. ed. [S.l.]: Pearson Education, 2012. 20
- FOWLES, G. R. **Introduction to Modern Optics**. 2. ed. [S.l.]: Dover Publications, Inc., 1975. 29
- GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. 4. ed. [S.l.]: Pearson Education Limited, 2018. 13, 20, 21
- HAN, J.; KAMBER, M.; PEI, J. **Data Mining: Concepts and Techniques**. 3. ed. [S.l.]: Elsevier Inc., 2012. 35

- HECHT, E. **Optics**. 5. ed. [S.l.]: Pearson Education, 2017. 25
- ITU-R. **Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios**. [S.l.], 2011. 18
- JACOBSON, R. E. *et al.* **The Manual of Photography: Photographic and digital imaging**. 9. ed. [S.l.]: Focal Press, 2000. 28
- NI. **NI Vision: NI 177x Smart Camera User Manual**. [S.l.], 2011. Manual do Usuário. 26, 28
- NI. **NI Vision Assistant Help**. [S.l.], 2014. Documentação de ajuda do software, acessada via menu Help. 31, 32, 34, 35, 37
- PRATT, W. K. **Digital Image Processing**. 3. ed. [S.l.]: John Wiley and Sons, 2001. 19
- SZELISKI, R. **Computer Vision: Algorithms and Applications**. 2. ed. [S.l.]: Springer, 2022. 17, 30
- TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introduction to Data Mining**. 1. ed. [S.l.]: Pearson Education, 2006. 35, 36

APÊNDICES

APÊNDICE A – CÁLCULO DE VELOCIDADE LINEAR DA MESA

Figura 43 – Cálculo de velocidade linear da mesa

$R = 275$ $R = 275$	' Raio da mesa em mm
$t = 5.4$ $t = 5.4$	' Tempo em que uma amostra completa uma revolução na mesa
$w = (2 \cdot \pi) / t$ $w = 1.163552835$	' Velocidade angular da mesa em radianos por segundo
$v = w \cdot R$ $v = 319.977029532$	' Velocidade linear da mesa na extremidade, em mm/s

Fonte: Do Autor (2026)

APÊNDICE B – CÁLCULO DE TEMPO MÁXIMO DE EXPOSIÇÃO

Figura 44 – Cálculo de tempo máximo de exposição

B = 2 B = 2	' Borramento em Pixels
FOVh = 153 FOVh = 153	' FOV Horizontal em mm
Vh = 319.977 Vh = 319.977	' Velocidade Linear da amostra na Direção Horizontal em mm/s
Nh = 1280 Nh = 1280	' Quantidade de Pixels do Sensor na Direção Horizontal
Emax = (B * FOVh)/(Vh * Nh) * (10 ^ 6) Emax = 747.124012038	' Tempo Máximo de Exposição em us

Fonte: Do Autor (2026)

APÊNDICE C – TABELA-VERDADE PASS/FAIL

Figura 45 – Tabela-verdade pass/fail

Amostra presente no Frame (A)	Amostra Intacta (B)	Status Pass no Final da Inspeção (S)
0	0	1
0	1	1
1	0	0
1	1	1

$$S = A \cdot B + \bar{A} \cdot B + \bar{A} \cdot \bar{B}$$

$$S = B \cdot (A + \bar{A}) + \bar{A} \cdot \bar{B}$$

$$S = B + \bar{A} \cdot \bar{B}$$

$$S = B + \bar{A}$$

Fonte: Do Autor (2026)

ANEXOS

ANEXO A – ESPECIFICAÇÕES CÂMERA BASLER ACA1300-60GC

Figura 46 – Especificações Técnicas da câmera Basler acA1300-60gc

Resolução (H x V pixels)	1280 x 1024 Pixels
Tamanho do Sensor	1 / 1.8"
Tipo	CMOS Colorido
Shutter	Global Shutter
Tamanho do Pixel horizontal/vertical	5.3 µm x 5.3 µm
Taxa de Captura	60 FPS
Interface	Gigabit
Profundidade de Pixel	12 bits
Controle de Exposição	• Programável
Sincronização	• Trigger Externo • Conexão Ethernet • Free-Run
Tamanho (C x L x A)mm	42.0 x 29.0 x 29.0
Montagem	• C-Mount • CS-Mount
Entrada Digital	1
Saída Digital	1
Alimentação	PoE or 12 VDC
Consumo	2.0 W / 2.6 W (PoE)
Peso (typical)	90 g
Sensor	EV76C560

Fonte: (ALTAVISION, 2026)