

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Lucas Meurer Lameira

Sistema de Inspeção Visual de Placas de Circuito Impresso

Porto Alegre

2020

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Lucas Meurer Lameira

**Sistema de Inspeção Visual de Placas de Circuito
Impresso**

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como requisito parcial para Graduação em Engenharia Elétrica.

Orientador: Prof. Dr. Tiago Roberto Balen

Porto Alegre

2020

*Este trabalho é dedicado à minha família
que com muito esforço e carinho
permitiram que eu chegasse onde estou.*

Agradecimentos

Agradeço à Universidade Federal do Rio Grande do Sul, à Escola de Engenharia e ao Departamento de Engenharia Elétrica pela possibilidade de realizar minha formação. Aos professores pelo ensinamento e conhecimento transmitido. À Ecole CentraleSupélec que me possibilitou realizar meu Duplo Diploma. Ao professor Tiago que me orientou neste trabalho e durante minha jornada de Iniciação Científica.

Aos meus amigos e colegas de barra, à galera da Resistência, à turma do apartamento BD2D e à AlterVision minha profunda gratidão por todos os momentos que vivenciamos juntos.

Principalmente, eu quero agradecer à minha família pelo suporte dado durante essa jornada. Em especial, agradeço minha mãe Bernadete e ao Luis, ao meu pai Luciano e à Sandra e à minha irmã Vivian. Um agradecimento em especial também para minha avó Vera que me abrigou em Porto Alegre e me ajudou no início desse processo. Agradeço também à minha parceira Camila, que está ao meu lado em todas ocasiões.

Resumo

A inspeção de qualidade de produtos na Indústria Brasileira ainda é manual ou pouco automatizada. Através da automatização dessa tarefa, é possível aumentar a produtividade e a qualidade do produto final. Com esse intuito, este trabalho visa desenvolver uma solução de inspeção automática visual de placas de circuito impresso. Através de imagens capturadas de placas de circuito impresso e de técnicas de processamento digital de imagens, é proposto o desenvolvimento de um modelo de aprendizagem de máquina capaz de classificar determinados componentes ou a ausência deles em placas de circuito impresso SMD (*Surface-Mount Device*). Além da classificação, o sistema é capaz de inspecionar o posicionamento do componente, gerando um alerta para componentes rotacionados e deslocados. As SVMs (Máquinas de Vetores de Suporte) consistem na técnica de aprendizado de máquina utilizada na implementação do classificador. O modelo recebe como entrada um vetor de características que representa a forma, textura e cor das imagens dos componentes. As características de forma e textura são obtidas pelo HOG (Histograma de Gradientes Orientados) e a cor é caracterizada pelo histograma da imagem no espaço HSV (*Hue, Saturation, Value*). O modelo de classificação implementado alcançou uma acurácia de 98,7% nas imagens testadas.

Palavras-chaves: Visão Computacional, Processamento Digital de Imagens, Inspeção Automática Visual, Placa de Circuito Impresso.

Abstract

The quality inspection of final products in the Brazilian Industry is still manual or very little automated. Through the automation of this task, it is possible to increase productivity and product quality. To this end, this work aims to develop a printed circuit board automated inspection system. Therefore, a machine learning model capable of classifying SMD (Surface-Mount Device) components is proposed. In addition to the image classification, the system is able to inspect the component's position, generating an alert for rotated and displaced components. SVMs (Support Vector Machines) are the machine learning technique used in the implementation of the classifier. The model receives as input a feature vector that contains information describing the shape, texture and color of the images. The shape and texture features are computed using HOG (Histogram of Oriented Gradients) and the color descriptor is created by calculating the histogram of the image in the HSV (Hue, Saturation and Value) color space. The model achieved an accuracy of 98.7% on the test images.

Key-words: Computer Vision, Digital Image Processing, Automatic Optical Inspection, Printed Circuit Board.

Lista de ilustrações

| | |
|--|----|
| Figura 1 – Exemplo de placa com trilhas em bom estado. | 15 |
| Figura 2 – Exemplo de placa com trilhas defeituosas. | 16 |
| Figura 3 – Exemplo de falhas no processo de soldagem. | 16 |
| Figura 4 – Técnicas de iluminação. | 18 |
| Figura 5 – Máquina de inspeção <i>offline</i> | 19 |
| Figura 6 – Exemplo de falhas detectadas pela máquina HJ-760. | 20 |
| Figura 7 – Partes da máquina HJ-760. | 20 |
| Figura 8 – Linha de montagem com máquina de inspeção <i>inline</i> | 21 |
| Figura 9 – Representação do espaço de cores RGB. | 22 |
| Figura 10 – Modelo do espaço de cores HSV. | 23 |
| Figura 11 – Representação do histogramas da imagem escura, clara, baixo contraste, alto contraste. | 24 |
| Figura 13 – Operações morfológicas de erosão e dilatação. | 25 |
| Figura 12 – Representação do funcionamento do método de Otsu. | 25 |
| Figura 14 – Operações morfológicas de abertura e fechamento. | 26 |
| Figura 15 – Processo de extração das características do HOG: a imagem é dividida em células de dimensões $N \times N$ pixels. A orientação de todos os pixels é computada e acumulada em um histograma de orientações de M -bins. Finalmente, as células dos histogramas são agrupadas e o vetor final é construído. Neste exemplo a imagem é dividida em 4×4 células com 8 bins representando as orientações. | 28 |
| Figura 16 – Separação das classes através de hiperplanos. | 30 |
| Figura 17 – SVMs com margens rígidas. | 31 |
| Figura 18 – SVMs com margens suaves. | 31 |
| Figura 19 – Fluxograma de inspeção da solução proposta. | 35 |
| Figura 20 – Fluxograma do classificador proposto. | 36 |
| Figura 21 – Amostra de imagem para cada classe de classificação. | 36 |
| Figura 22 – Posicionamentos possíveis do componente na PCI. | 37 |
| Figura 23 – Configuração para a aquisição de imagens da base de dados FICS-PCB. | 39 |
| Figura 24 – Interface da aplicação. | 40 |
| Figura 25 – Recorte dos componentes. | 41 |
| Figura 26 – Aplicação dos gradientes no resistor SMD. | 43 |
| Figura 27 – Aplicação dos gradientes no capacitor cerâmico SMD. | 43 |
| Figura 28 – Aplicação dos gradientes no capacitor eletrolítico. | 43 |
| Figura 29 – Aplicação dos gradientes no circuito integrado. | 43 |
| Figura 30 – Aplicação dos gradientes na falta de componente. | 44 |

| | |
|---|----|
| Figura 31 – Divisão da imagem em células de 16 x 16 pixels. | 44 |
| Figura 32 – HOG do resistor SMD. | 44 |
| Figura 33 – HOG do capacitor cerâmico SMD. | 45 |
| Figura 34 – HOG do capacitor eletrolítico. | 45 |
| Figura 35 – HOG do circuito integrado. | 45 |
| Figura 36 – HOG da ausência de componente. | 45 |
| Figura 37 – Imagem HSV do resistor SMD. | 46 |
| Figura 38 – Imagem HSV do capacitor cerâmico SMD. | 46 |
| Figura 39 – Imagem HSV do capacitor eletrolítico. | 46 |
| Figura 40 – Imagem HSV do circuito integrado. | 47 |
| Figura 41 – Imagem HSV da ausência de componente. | 47 |
| Figura 42 – Segmentação do componente - caso 1. | 48 |
| Figura 43 – Segmentação do componente - caso 2. | 48 |
| Figura 44 – Fluxograma de inspeção da posição. | 49 |
| Figura 45 – Simulação de componente rotacionado. | 50 |
| Figura 46 – Simulação de componente linearmente deslocado. | 50 |
| Figura 47 – Curva de treinamento do modelo de classificação. | 51 |
| Figura 48 – Curva de escalabilidade do modelo de classificação. | 52 |
| Figura 49 – Curva do desempenho do modelo de classificação. | 52 |
| Figura 50 – Matriz de confusão do modelo de classificação. | 53 |
| Figura 51 – Inspeção do componente não rotacionado. | 54 |
| Figura 52 – Inspeção do componente rotacionado. | 54 |
| Figura 53 – Inspeção do componente id=1 da Tabela 3. | 55 |
| Figura 54 – Inspeção do componente não deslocado. | 55 |
| Figura 55 – Inspeção do componente deslocado. | 57 |

Lista de tabelas

| | |
|---|----|
| Tabela 1 – Descrição do hardware utilizado no desenvolvimento do trabalho. . . . | 39 |
| Tabela 2 – Métricas de avaliação do modelo. | 53 |
| Tabela 3 – Resultados de inspeção do deslocamento angular. | 56 |
| Tabela 4 – Resultados de inspeção do deslocamento linear. | 58 |
| Tabela 5 – Resultados de classificação nos componentes deslocados e rotacionados. | 59 |

Lista de abreviaturas e siglas

| | |
|--------|---|
| PCI | Placa de Circuito Impresso |
| ROI | Região de Interesse |
| AM | Aprendizado de Máquina |
| SVM | Máquinas de Vetores de Suporte |
| HOG | Histograma de Gradientes Orientados |
| RNA | Redes Neurais Artificiais |
| AOI | Inspeção Automática Visual |
| CDD | <i>Charge-Coupled Device</i> |
| SMD | <i>Surface-Mount Device</i> |
| SMT | <i>Surface-Mount Technology</i> |
| THT | <i>Through-Hole Technology</i> |
| SPI | <i>Solder Paste Inspection</i> |
| RGB | <i>Red, Green and Blue</i> |
| HSV | <i>Hue, Saturation and Value</i> |
| HSI | <i>Hue, Saturation and Intensity</i> |
| MLP | <i>Multilayer Perceptron</i> |
| DWT | <i>Discrete Wavelet Transform</i> |
| LQV | <i>Learning Vector Quantization</i> |
| UFRGS | Universidade Federal do Rio Grande do Sul |
| LASCAR | Laboratório de Sistemas de Controle, Automação e Robótica |

Sumário

| | | |
|------------|--|-----------|
| 1 | INTRODUÇÃO | 12 |
| 2 | FUNDAMENTAÇÃO TEÓRICA | 14 |
| 2.1 | Placa de circuito impresso | 14 |
| 2.1.1 | Tecnologias de PCIs | 14 |
| 2.2 | Métodos de Inspeção de PCIs: Inspeção Automática Visual | 16 |
| 2.3 | Iluminação | 18 |
| 2.4 | Máquinas AOI | 19 |
| 2.5 | Técnicas de processamento de imagens | 21 |
| 2.5.1 | Espaço de cores RGB | 21 |
| 2.5.2 | Espaço de cores HSV | 22 |
| 2.5.3 | Histograma de imagens | 22 |
| 2.5.4 | Limiarização (<i>Thresholding</i>) | 23 |
| 2.5.5 | Transformações Morfológicas Básicas | 25 |
| 2.5.6 | HOG (<i>Histogram of Oriented Gradients</i>) | 26 |
| 2.5.6.1 | Normalização global da imagem | 26 |
| 2.5.6.2 | Cálculo do gradiente nas direções x e y | 26 |
| 2.5.6.3 | Cálculo dos histogramas dos gradientes | 27 |
| 2.5.6.4 | Normalização dos gradientes | 28 |
| 2.5.6.5 | Formação de um vetor provindo dos histogramas dos gradientes | 28 |
| 2.6 | Técnica de aprendizado de máquina SVM | 28 |
| 2.7 | Trabalhos Relacionados | 32 |
| 2.7.1 | Placa nua | 32 |
| 2.7.2 | Placa montada | 32 |
| 3 | SOLUÇÃO PROPOSTA | 34 |
| 3.1 | Configuração inicial através de um referencial | 34 |
| 3.2 | Inspeção dos componentes | 34 |
| 3.2.1 | Classificação de componentes | 36 |
| 3.2.2 | Inspeção da posição do componente | 37 |
| 4 | METODOLOGIA EXPERIMENTAL | 38 |
| 4.1 | Base de dados | 38 |
| 4.2 | Linguagem de programação e bibliotecas | 38 |
| 4.3 | Descrição do Hardware | 39 |
| 4.4 | Interface Gráfica | 40 |

| | | |
|------------|--|-----------|
| 4.5 | Aquisição das coordenadas dos componentes | 40 |
| 4.6 | Classificador | 41 |
| 4.6.1 | Histogram of Oriented Gradients | 42 |
| 4.6.2 | Descritor de cores no espaço HSV | 45 |
| 4.7 | Posicionamento | 47 |
| 4.7.1 | Deslocamento angular do componente | 48 |
| 4.7.2 | Deslocamento linear do componente | 50 |
| 5 | RESULTADOS E DISCUSSÕES | 51 |
| 5.1 | Classificador | 51 |
| 5.2 | Posicionamento | 54 |
| 6 | CONSIDERAÇÕES FINAIS | 60 |
| | REFERÊNCIAS | 62 |

1 Introdução

Na indústria, o custo de envio de um produto com falhas ao cliente pode acarretar perdas enormes. Sem contar no impacto negativo na imagem da marca. Com clientes cada vez mais exigentes, automatizar a inspeção visual tornou-se uma necessidade em diversos setores. Além disso, o mercado de visão computacional já vale 11,9 bilhões de dólares globalmente e há uma previsão de crescimento para 17,4 bilhões dólares até 2023 (MARKETSANDMARKETS, 2018). No Brasil, esse segmento é principalmente impulsionado pelo setor automotivo e farmacêutico. Só o setor automotivo já é responsável por 22% do PIB brasileiro com um faturamento anual de 59 bilhões de dólares (MDIC, 2017).

Na Universidade Federal do Rio Grande do Sul, o laboratório LASCAR (Laboratório de Sistemas de Controle, Automação e Robótica) adquiriu uma máquina de inspeção automática visual de placas de circuito impresso, nomeada AOI HJ-760. A função desse equipamento é detectar defeitos na fabricação de PCIs (Placa de Circuito Impresso), de forma a substituir a inspeção visual humana. Uma vez que, geralmente esta inspeção é realizada por operadores humanos na indústria, estando sujeita a falhas pelo cansaço e pela dificuldade visual atrelada ao tamanho reduzido dos componentes eletrônicos a serem inspecionados. Desta forma, inicialmente, este Projeto de Diplomação teria como objetivo colocar esse equipamento para funcionar, o testando com diferentes PCIs, analisando e discutindo os resultados obtidos. Com o entendimento do funcionamento do equipamento, um tutorial detalhado do funcionamento da máquina seria escrito, de tal forma que ela seria posteriormente utilizada por bolsistas e funcionários do laboratório. Entretanto, com o avanço da pandemia de COVID-19, foram restringidos os acessos aos laboratórios e optou-se pela definição de um novo objetivo de trabalho.

Desta forma, o objetivo do trabalho aqui apresentado consiste em implementar um *software* de inspeção de componentes de placas de PCIs SMD (*Surface-Mount Device*). Para isto, é proposto um processo de inspeção que consiste em um classificador de componentes, cuja função é classificar cada componente de uma placa de teste, comparando o resultado obtido ao devido componente de uma placa de referência. A classificação de imagens é realizada através do algoritmo de aprendizado de máquina SVM (Máquinas de Vetores de Suporte). O classificador SVM recebe como entrada um vetor de características que descreve a forma, textura e cor dos componentes. As características de forma e textura foram extraídas com a ajuda do algoritmo HOG (Histograma de Gradientes Orientados) e o descritor de cor é caracterizado pelo histograma da imagem no domínio de cores HSV (*Hue, Saturation and Value*).

Além do classificador, deseja-se que o aplicativo seja capaz de medir o posicionamento linear e angular dos componentes em relação a referência. Essa etapa é realizada através da segmentação do componente analisado, onde o processo de segmentação é realizado por técnicas de binarização de imagens, operações morfológicas e detecção de contornos.

Este trabalho é organizado da seguinte forma. No Capítulo 2 é apresentado o embasamento teórico utilizado para a realização do trabalho. No Capítulo 3 é detalhado o funcionamento da solução proposta. No Capítulo 4 são apresentados os materiais e as ferramentas utilizadas para a implementação da solução. No Capítulo 5, por sua vez, são discutidos os resultados obtidos no trabalho e, finalmente, no Capítulo 6 são apresentadas as conclusões e considerações finais.

2 Fundamentação Teórica

Neste capítulo são apresentados os principais conceitos teóricos utilizados no presente trabalho. Primeiramente, é introduzido a funcionalidade de PCIs. Logo após, são apresentadas as principais tecnologias de fabricação de PCIs e os possíveis erros que podem ocorrer durante o processo de produção. Então, os principais métodos de inspeção são citados, e o método de inspeção automática visual é discutido. Algumas técnicas de processamento digital de imagens e aprendizado de máquina são revisadas, e por fim, os trabalhos relacionados disponíveis na literatura são expostos.

2.1 Placa de circuito impresso

Uma PCI consiste em uma placa fina de material dielétrico que suporta os componentes de um circuito e os interliga por trilhas de condução, geralmente feitas de cobre, em uma ou ambas faces da placa (FUSETTN, 1999).

O uso de PCIs tem sido uma técnica vastamente utilizada para realizar a interconexão de componentes eletrônicos durante décadas. Como este trabalho objetiva desenvolver um protótipo de inspeção de PCIs, nesta seção são apresentadas as principais tecnologias de fabricação, assim como os possíveis erros que podem ocorrer durante este processo.

2.1.1 Tecnologias de PCIs

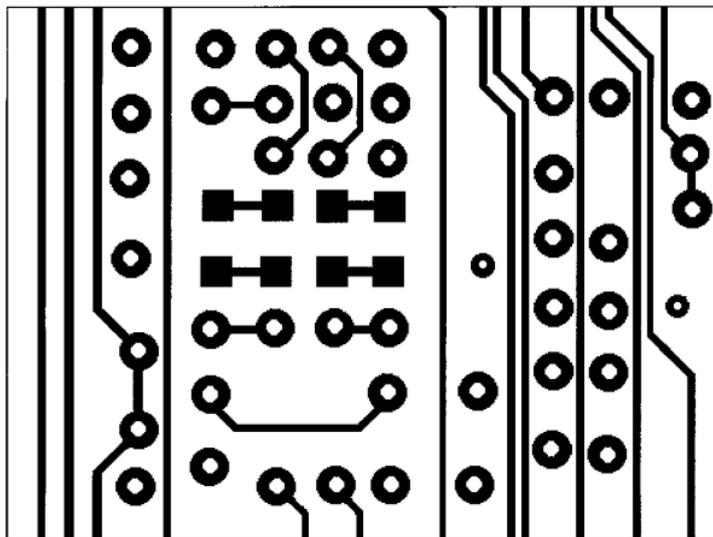
Existem dois tipos de tecnologias relacionadas à fixação dos componentes na placa:

- THT - *Through-Hole Technology*.
- SMT - *Surface-Mount Technology*.

A tecnologia THT consiste em componentes que possuem fios como terminais condutores que atravessam a placa através de furos. Para garantir a conectividade elétrica, esses componentes são soldados na outra face da placa. Já na tecnologia SMT, os componentes não possuem fios como terminais e a placa não é furada. Os componentes são posicionados em cima da ilha e então soldados. A tecnologia SMT possibilita uma alta densidade de componentes e suporta um alto desempenho. No entanto, sua manufatura e seus reparos são mais difíceis. Devido a este fato, torna-se necessário alto investimento em equipamentos automatizados para o processo de manufatura, inspeção e reparos (FUSETTN, 1999).

As PCIs são convencionalmente classificadas em três diferentes grupos (COOMBS; HOLDEN, 2016): placa de face simples, placa de face dupla e placa multicamadas. A

Figura 1 – Exemplo de placa com trilhas em bom estado.



Fonte – Retirado de (MOGANTI et al., 1996).

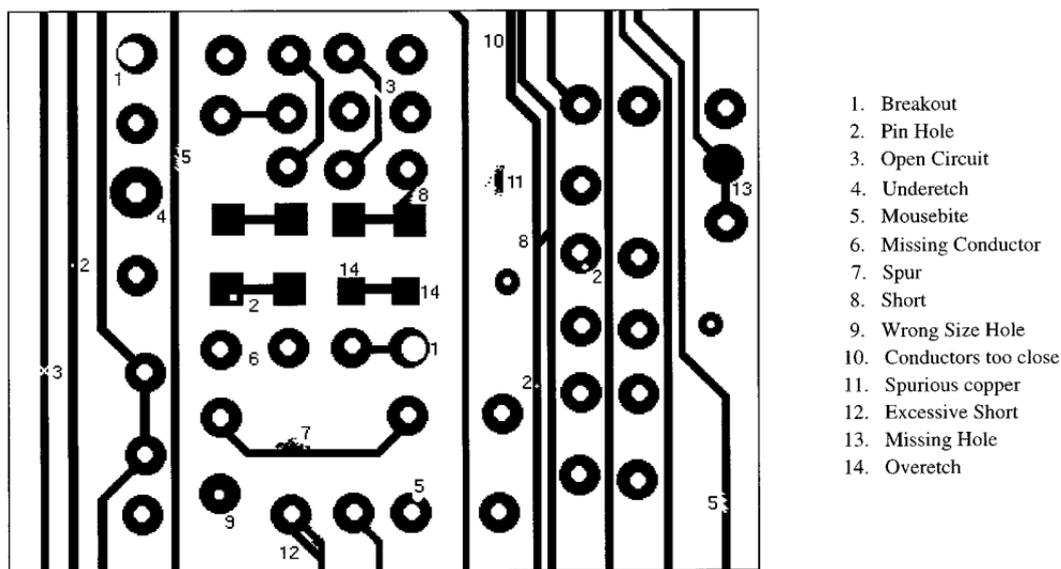
placa de face simples possui numa das faces as trilhas condutoras e na outra face os seus componentes eletrônicos. Já a placa de face dupla possui trilhas condutoras em ambas as faces e os componentes podem ser soldados em qualquer uma das faces. As placas multicamadas possuem várias camadas de trilhas, sendo duas delas as faces da placa, e as camadas intermediárias se alternam entre camadas com trilhas e camadas com material isolante.

É possível dividir os potenciais defeitos em placas de circuito impresso em três grupos: defeitos nos componentes ou na placa, defeitos na inserção dos componentes e defeitos no processo de soldagem (DORO, 2004).

Os defeitos nos componentes ou na placa ocorrem durante o processo de fabricação dos mesmos, má qualidade dos componentes ou das placas, pelo armazenamento, transporte ou manuseio incorreto. Defeitos como curto-circuito, condutor interrompido, excesso ou falta de cobre nas trilhas são alguns exemplos de defeitos que podem ser encontrados (MOGANTI et al., 1996), (DORO, 2004). Na Figura 1 é apresentado as trilhas de uma placa em boas condições. Na Figura 2, por sua vez, é apresentado um exemplo de placa onde 14 possíveis defeitos são expostos, como curto-circuito, circuito aberto falta de furos, entre outros.

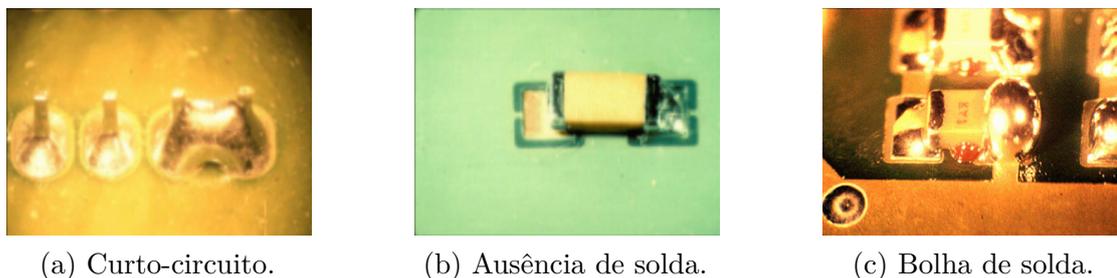
Durante o processo de inserção de componentes nas PCIs, podem ocorrer alguns erros causados tanto por operadores realizando a tarefa de forma manual quanto por processos automatizados devido a máquinas mal reguladas ou programadas. Dessa forma, no processo de inserção podem ocorrer defeitos de falta de componente, componente errado ou desalinhado (DORO, 2004). O sistema de inspeção proposto neste trabalho visa detectar

Figura 2 – Exemplo de placa com trilhas defeituosas.



Fonte – Retirado de (MOGANTI et al., 1996).

Figura 3 – Exemplo de falhas no processo de soldagem.



Fonte – Imagens retiradas de (ARAUJO, 2020).

principalmente esses defeitos citados.

No processo de soldagem, diversos defeitos podem ocorrer. Alguns desses defeitos são apresentados na Figura 3. Na Figura 3a é possível verificar que a solda está causando um curto-circuito. Na Figura 3b nota-se que o componente foi soldado apenas em um dos lados. Na Figura 3c verifica-se um excesso de solda, onde uma bolha de solda é gerada.

2.2 Métodos de Inspeção de PCIs: Inspeção Automática Visual

(MOGANTI et al., 1996) descreve as principais técnicas de inspeção de PCIs, onde diversos algoritmos de inspeção automática de PCIs são analisados. Os métodos são separados em elétricos e não elétricos. Sendo os elétricos classificados em com contato ou sem contato. Enquanto os métodos não elétricos, por sua vez, são classificados em cinco classes diferentes, conforme a tecnologia empregada: inspeção automática visual, inspeção

por imagens de raios x, laminografia, inspeção por imagens de ultrassom e inspeção por imagens térmicas (MOGANTI et al., 1996).

Este trabalho é focado no método de inspeção automática visual, amplamente conhecido como AOI (*Automated Optical Inspection*). Sistemas de AOI utilizam câmeras para captar imagens e, através de algoritmos de processamento digital de imagens, são capazes de identificar falhas em PCIs. A aquisição da imagem pode ser realizada por diversos tipos de câmeras, onde a maioria delas é baseada em sensores CDD (*charge-coupled device*). Uma das condições mais importantes que o sistema de câmeras deve atender é possuir uma resolução suficientemente adequada para a sua aplicação.

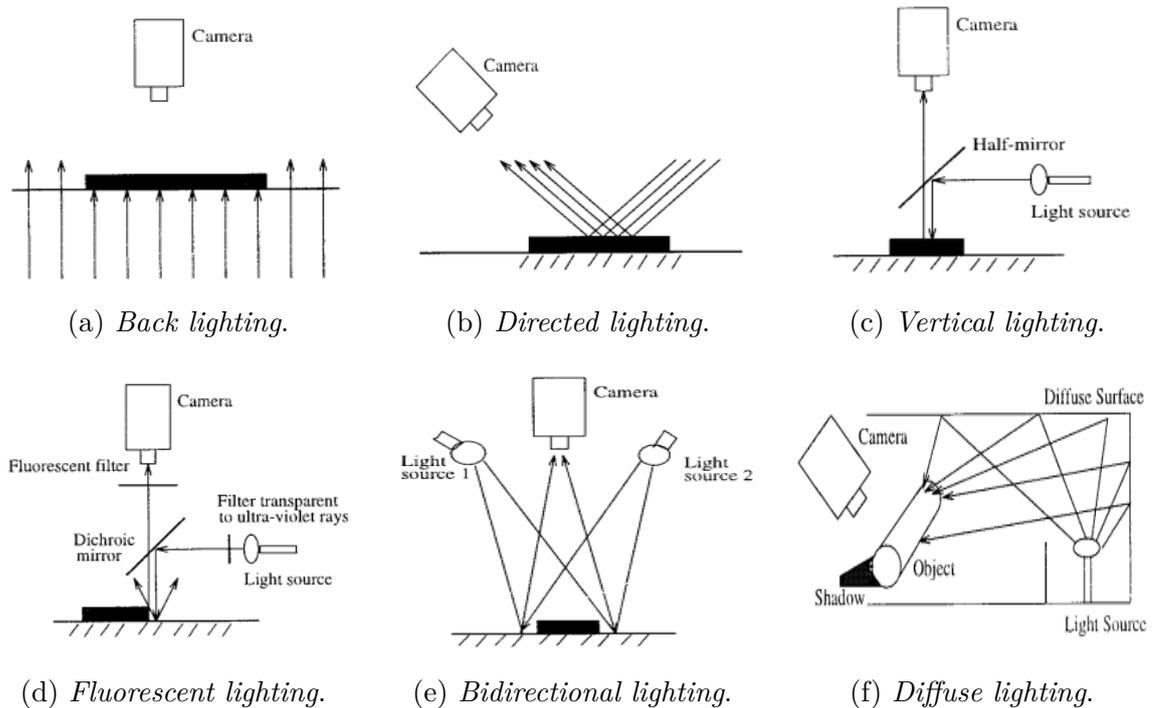
O processo de inspeção de PCIs THT e SMT pode ser dividido em: inspeção de placa nua e inspeção de placa montada (SZYMANSKI, 2014). Em relação ao *software* de inspeção, categoriza-se os algoritmos em duas categorias: inspeção baseada em imagem e inspeção baseada em algoritmo (PEALLAT et al., 2018).

A inspeção de placa nua consiste na inspeção da placa sem os componentes, ocorrendo antes da soldagem dos mesmos. Neste processo são inspecionadas as trilhas para verificar se há curto-circuito ou circuito aberto, assim como o posicionamento e a falta de furos e ilhas. Nas Figuras 1 e 2, apresentadas anteriormente, é possível verificar os defeitos que são inspecionados neste processo. A inspeção de placa montada consiste na inspeção da placa com os componentes já soldados, ou seja, é a última etapa da cadeia de produção, onde a placa já está pronta. Neste processo é inspecionado os componentes, verificando se há algum faltando, se estão na posição correta e se foram inseridos os componentes certos (SZYMANSKI, 2014).

As tecnologias baseadas em imagem são projetadas para realizar a verificação do componente por meio de comparação. Para isto, compara-se a imagem capturada com uma coleção de imagens anteriormente adquiridas, onde constam imagens de acerto e erro dos componentes. As diversas técnicas de comparação variam desde algoritmos básicos até redes neurais, mas a ideia é sempre a mesma, comparar a imagem atual com as imagens armazenadas no banco de dados (PEALLAT et al., 2018).

Através de padrões matemáticos e geometria, a inspeção baseada em algoritmo emprega técnicas de reconhecimento de padrões para encontrar os componentes nas imagens. Em vez de comparar uma imagem com outra, a tecnologia usa um padrão definido (geometria ou esqueleto do objeto) para inspecionar os componentes. Esse tipo de tecnologia oferece uma melhor precisão porque não há necessidade de ficar comparando os pixels das imagens, processo que pode ser bastante lento. Ademais, o sistema fica mais robusto e menos dependente de alterações no ambiente, como alterações nas cores de componentes e placas, ou alteração na luminosidade do local (PEALLAT et al., 2018).

Figura 4 – Técnicas de iluminação.



Fonte – (MOGANTI et al., 1996).

2.3 Iluminação

A iluminação do local consiste em parte fundamental do processo de AOI. Um ambiente mal iluminado dificulta consideravelmente o desenvolvimento de algoritmos de processamento de imagem. Utilizando uma iluminação adequada, é possível realizar uma inspeção mais precisa e também diminuir a complexidade dos algoritmos de processamento.

A Figura 4 apresenta algumas técnicas de iluminação descritas na literatura (MOGANTI et al., 1996). Na Figura 4a é aplicado a técnica de *back lighting*, onde gera-se silhuetas escuras contra um fundo brilhante. Esta técnica é usada para medição de objetos e detecção de bordas, pois apresenta uma boa precisão. A Figura 4b apresenta a técnica *directed lighting*, essa é a técnica de iluminação mais usada. Consiste em iluminar a PCI com um ângulo em que a reflexão da luz encontra diretamente a câmera. Na Figura 4c está sendo aplicada a técnica *vertical lighting*, na Figura 4d a técnica *fluorescent lighting* e na Figura 4e a técnica *bidirectional lighting*. Nessas técnicas a iluminação é aplicada no eixo da placa. A iluminação no eixo é adequada para objetos planos, onde aprimora-se suas texturas e ângulos. Por fim, na Figura 4f é aplicado a *diffuse lighting*, que é usada em aplicações onde luzes em multi direções são necessárias. Este tipo de iluminação suprime sombras indesejadas de objetos espaciais e ajuda a equilibrar o brilho do ambiente (MOGANTI et al., 1996), (KUNTE, 2018).

Figura 5 – Máquina de inspeção *offline*.

Fonte – Adquirido de (ALLPOINT, 2020).

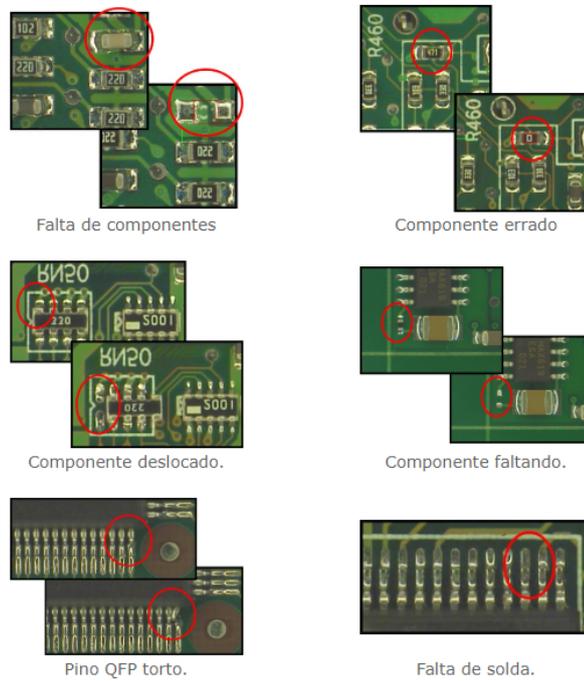
2.4 Máquinas AOI

Existem dois tipos de máquinas AOI: *offline* e *inline*. As máquinas *offline* são máquinas desenvolvidas para funcionar fora de uma linha de produção, onde é preciso um operador humano ou um braço robótico para inserir as PCIs na máquina. A Figura 5 apresenta um exemplo de máquina AOI *offline*. Essa máquina consiste em uma máquina AOI HJ-760 e foi adquirida e instalada no laboratório LASCAR da UFRGS.

A máquina HJ-760 é capaz de detectar diversos defeitos como: falta de componentes, componentes errados, deslocados, faltando, pinos tortos e falta de solda. Na Figura 6 é exemplificado alguns defeitos que podem ser detectados com a AOI HJ-760. A máquina funciona com um sistema de iluminação sofisticado e, através do *software HJ-760*, que roda em plataforma *Windows*, os algoritmos inteligentes de detecção de falhas da máquina realizam a inspeção. As partes das máquinas são apresentadas na Figura 7.

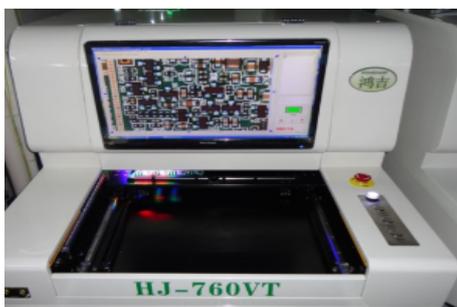
Nos sistemas *inline*, a máquina AOI está acoplada à linha de produção. No trabalho de (MELO, 2013) é apresentado a linha de montagem SMT do LabElectron, um laboratório-fábrica de montagem de PCIs, conforme é apresentado na Figura 8. O processo de produção começa na Printer, onde é aplicado a pasta de solda responsável por fixar os componentes SMD às placas. Em seguida a solda aplicada é inspecionada pela SPI (*Solder Paste Inspection*). As máquinas de inserção, tanto a automática quanto a manual, são responsáveis por posicionar os componentes na placa e o forno de refusão solda efetivamente estes componentes. Então, no final da linha de produção, uma máquina AOI *inline* realiza a inspeção da placa.

Figura 6 – Exemplo de falhas detectadas pela máquina HJ-760.



Fonte – Adquirido de (ALLPOINT, 2020).

Figura 7 – Partes da máquina HJ-760.



(a) Visualização frontal.



(b) Encaixe das PCIs..

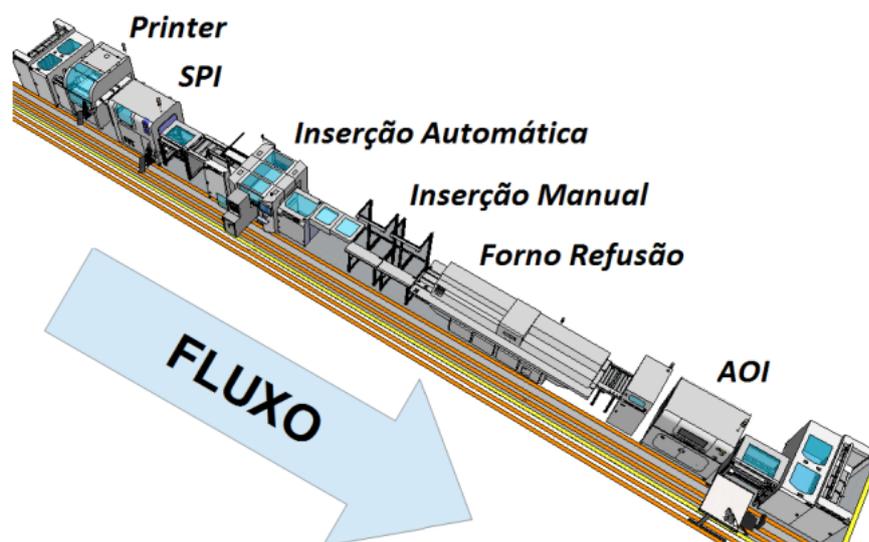


(c) Câmera de alta qualidade.



(d) Visualização das PCIs no *software*.

Fonte – Adquirido de (ALLPOINT, 2020).

Figura 8 – Linha de montagem com máquina de inspeção *inline*.

Fonte – Retirado de (MELO, 2013).

2.5 Técnicas de processamento de imagens

Com a evolução do poder computacional e das tecnologias de câmeras, cada vez mais cresce a possibilidade de aplicações de processamento digital de imagens nas mais diversas áreas. Nesta seção é apresentado alguns conceitos de processamento de imagens, desde a definição do pixel até métodos mais complexos envolvendo algoritmos inteligentes.

Uma imagem pode ser definida com uma função $f(x, y)$, onde x e y são coordenadas espaciais. A amplitude de f em qualquer ponto de (x, y) é chamada de intensidade de cinza da imagem nesse ponto. Quando os valores de x , y e da amplitude f são finitos e discretos, representam uma imagem digital. A imagem digital é representada por uma quantidade de números finitos, cada um com um valor e localização, esse elemento é denominado pixel. (GONZALEZ; WOODS, 2009).

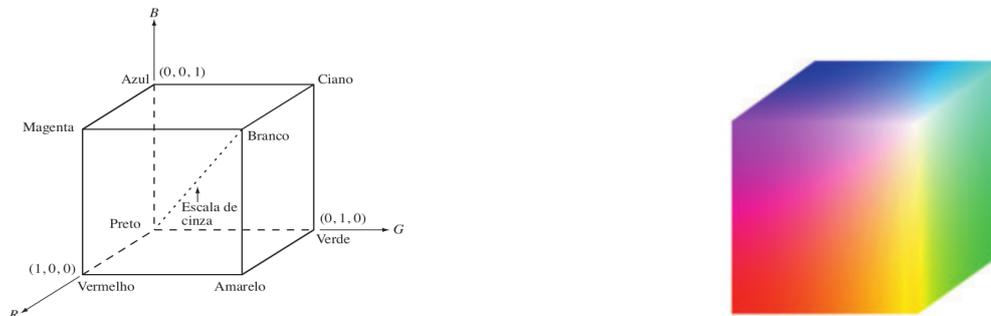
Uma imagem colorida pode ser representada de diversas formas. Existem inúmeros modelos de cores, conhecidos como espaço de cores. Na Seção 2.5.1 e Seção 2.5.2 são apresentados os dois modelos amplamente utilizados em Visão Computacional.

2.5.1 Espaço de cores RGB

Uma imagem representada no espaço de cor RGB (*red, green, blue*) possui três canais, um para cada cor primária, onde as cores primárias são vermelho, verde e azul. Na Figura 9a é apresentado o cubo representando o sistema de cores RGB. As cores primárias estão localizadas em três vértices. Ciano, magenta e amarelo são cores secundárias e estão representadas nos outros três vértices. A cor preta localiza-se na origem e os pontos da

linha diagonal apresentam os valores de cinza, desde o preto localizado no ponto $(0, 0, 0)$ até o branco no ponto $(1, 1, 1)$. A Figura 9b apresenta um cubo de cores RGB de 24 bits, onde cada canal da imagem possui 8 bits. (GONZALEZ; WOODS, 2009).

Figura 9 – Representação do espaço de cores RGB.



(a) Esquema do cubo de cores RGB.

(b) Exemplo de um cubo de cores 24 bits.

Fonte – Retirado de (GONZALEZ; WOODS, 2009).

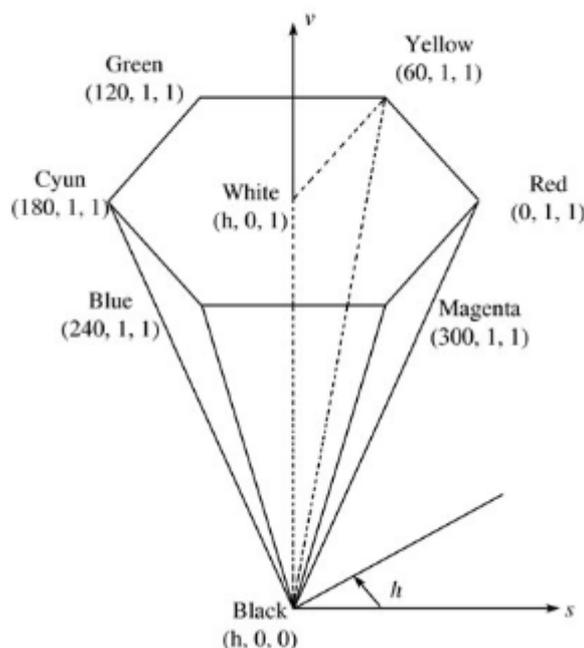
2.5.2 Espaço de cores HSV

O modelo de cores HSV (*hue, saturation, value*) é uma ferramenta bastante usada em algoritmos de processamento digital de imagens que se baseiam em características de cores. O modelo HSV é constituído por três canais matiz (H), saturação (S) e luminância (V). A Figura 10 representa graficamente o modelo HSV. Este modelo consiste em uma pirâmide hexagonal derivada do cubo RGB, onde os vértices do hexágono correspondem às cores primárias e secundárias RGB. A luminância varia de 0 (cor preta) a 1, onde as intensidade das cores são máximas. A saturação varia de 0 a 1 e representa a razão entre a pureza de um matiz e sua pureza máxima com $S = 1$ (PEDRINI, 2020).

2.5.3 Histograma de imagens

O histograma de uma imagem em escala de cinza é uma função discreta que corresponde à distribuição dos níveis de cinza. A normalização do histograma representa uma distribuição discreta de probabilidade (MARTINS, 2020). Além da informação estatística fornecida pelo histograma, é uma técnica bastante utilizada em procedimentos de classificação e segmentação de imagens. Devido ao baixo custo computacional para serem calculados, tornam-se uma ferramenta amplamente utilizada em aplicações de Visão Computacional em tempo real. A Figura 11 demonstra a aplicação do histograma em imagens em escala de cinza. Nota-se que na imagem escura, os componentes do histograma estão localizados no lado inferior da escala de intensidade. Já na imagem clara, os componentes estão localizados no lado superior (GONZALEZ; WOODS, 2009). Se a imagem digital for representada por 8 bits, em um histograma de 256 bins (quantidade de

Figura 10 – Modelo do espaço de cores HSV.



Fonte – Retirado de (LEI et al., 2012).

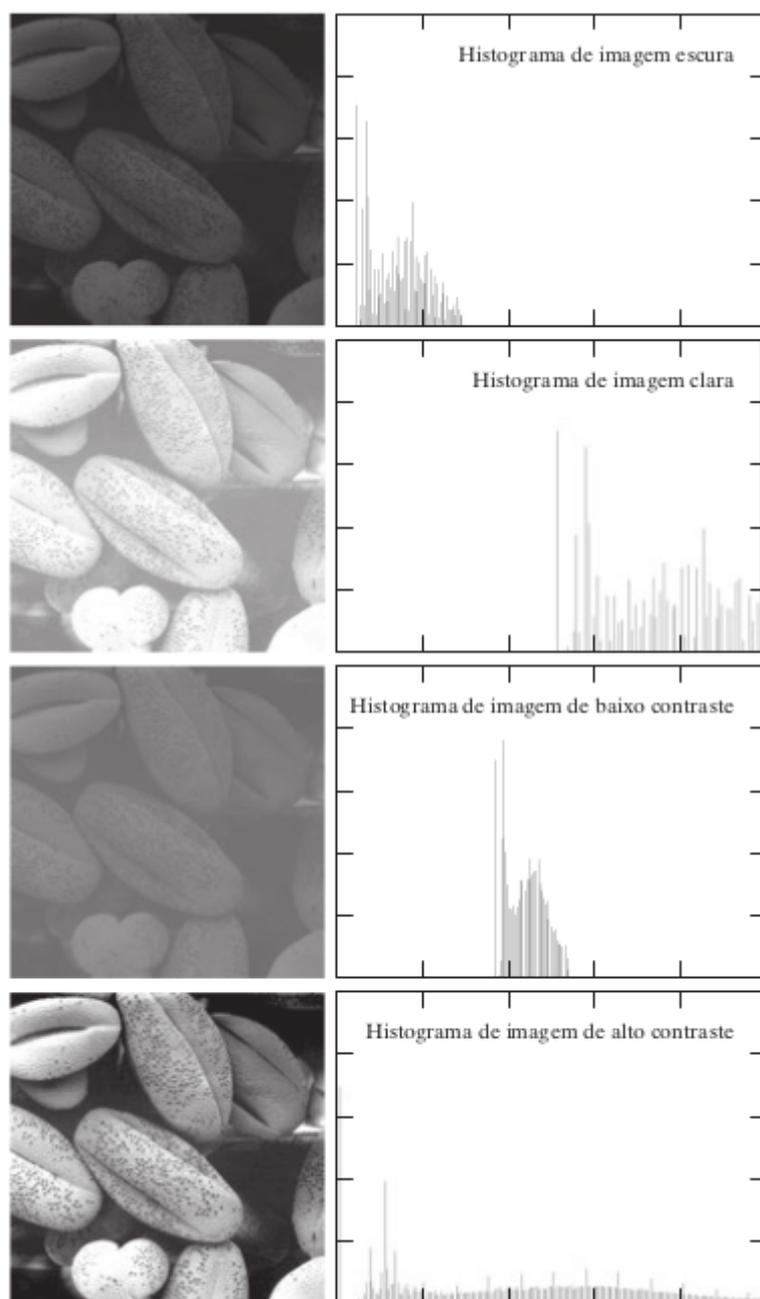
barras do histograma), a escala do eixo horizontal seria de 0 a 255, onde o 0 representa a cor preta e 255 a cor branca. A Figura 11 demonstra a forma do histograma em imagens de baixo e alto contraste. Na imagem de alto contraste, as componentes do histograma estão distribuídas em toda a faixa do eixo de horizontal, enquanto que na imagem de baixo contraste o histograma é bem estreito.

2.5.4 Limiarização (*Thresholding*)

A técnica de limiarização objetiva dividir a imagem em duas classes, denominadas fundo e objeto. Como o processo de limiarização gera uma imagem binária, esse processo é também denominado binarização (FILHO; NETO, 1999). Considerando uma imagem $f(x, y)$, a forma mais intuitiva de separar a imagem é selecionar um limiar T , e separar os pontos da imagem em duas classes, uma com $f(x, y) \leq T$ e a outra com $f(x, y) > T$.

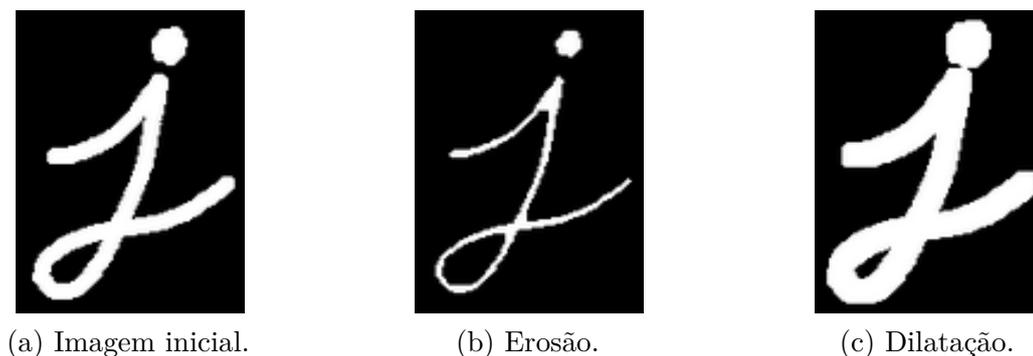
Um método mais sofisticado para a realização da limiarização é descrito em (OTSU, 1979). Essa técnica, conhecida como método de Otsu, objetiva maximizar a variância entre as classes para encontrar um limiar ótimo. Na Figura 12 é demonstrado o funcionamento do método de Otsu. A imagem da esquerda da Figura 12 representa a imagem original, e o seu histograma é representado na imagem central, onde a linha vermelha vertical representa o limiar ótimo. Os pontos de menor intensidade em relação ao ponto ótimo são representados com a cor preta na imagem de saída, e os pontos com maior intensidade em relação ao ponto ótimo são representados com a cor branca na imagem de saída binária.

Figura 11 – Representação do histogramas da imagem escura, clara, baixo contraste, alto contraste.



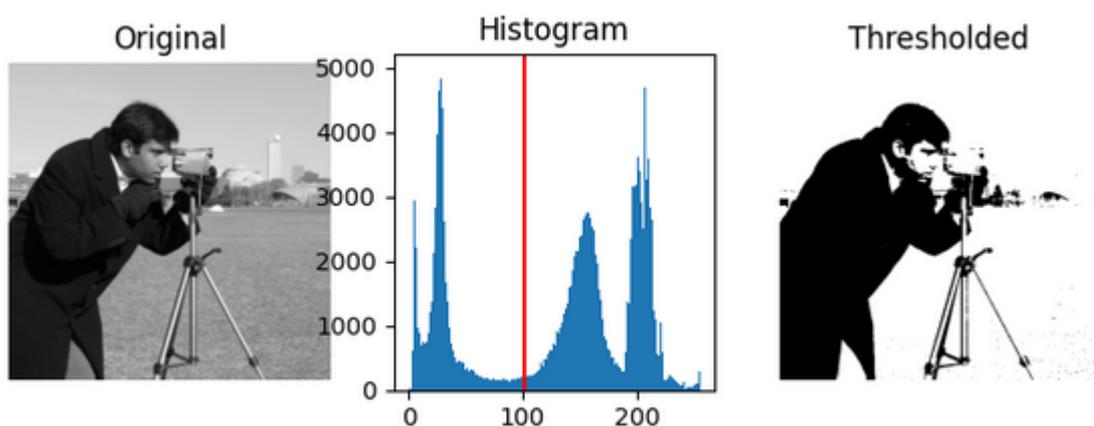
Fonte – Retirado de (GONZALEZ; WOODS, 2009)

Figura 13 – Operações morfológicas de erosão e dilatação.



Fonte – (OPENCV, 2020).

Figura 12 – Representação do funcionamento do método de Otsu.



Fonte – Retirado de (WALT et al., 2014).

2.5.5 Transformações Morfológicas Básicas

As transformações morfológicas são algumas operações fundamentais em processamento de imagens. Geralmente, essas operações são aplicadas em imagens binárias. Esse processo é baseado na forma da imagem, onde aplica-se um elemento estruturante *kernel* que define como a operação ocorre. As operações morfológicas básicas são: erosão, dilatação, abertura e fechamento. A Figura 13 apresenta exemplos da aplicação das operações de erosão e dilatação. Na Figura 13a é representado a imagem binária inicial. Após a aplicação da transformação de erosão é obtido a imagem resultante da Figura 13b. Por outro lado, a Figura 13c apresenta o resultado da operação de dilatação.

O processo de abertura, consiste nada mais que uma erosão seguida de uma dilatação. A abertura é bastante usada para a remoção de ruídos nas imagens, como demonstrado na Figura 14a. O processo de fechamento consiste em uma dilatação seguida por uma erosão. É útil para preencher os buracos dentro dos objetos, como exemplificado

Figura 14 – Operações morfológicas de abertura e fechamento.



Fonte – (OPENCV, 2020).

na Figura 14b.

2.5.6 HOG (*Histogram of Oriented Gradients*)

HOG é um extrator de características frequentemente utilizado em aplicações de reconhecimento de padrões. O descritor HOG foi proposto por (DALAL; TRIGGS, 2005) com o objetivo de detectar pessoas, e desde então o seu uso se expandiu para diversas aplicações, principalmente para quantificar e representar forma e textura. Conforme (WALT et al., 2014), o procedimento para o cálculo do HOG é dividido em cinco etapas:

1. Normalização global da imagem (etapa opcional).
2. Cálculo do gradiente nas direções x e y.
3. Cálculo dos histogramas dos gradientes.
4. Normalização dos gradientes.
5. Formação de um vetor provindo dos histogramas dos gradientes.

2.5.6.1 Normalização global da imagem

A primeira etapa visa reduzir a influencia de efeitos de iluminação. (DALAL; TRIGGS, 2005) cita a normalização gama, aplicada em imagens nos espaços de cores RGB, LAB e em nível de cinza. Entretanto, os próprios autores comentam que essa normalização proporciona uma modesta melhora no desempenho do algoritmo. Sendo assim, em algumas implementações do descritor HOG essa etapa não é realizada.

2.5.6.2 Cálculo do gradiente nas direções x e y

O segundo passo consiste em calcular o gradiente nas direções x e y. De acordo com (GONZALEZ; WOODS, 2009), o gradiente consiste em um vetor que aponta na direção

da máxima taxa de variação de f nas coordenadas (x, y) . (ZENZO, 1986) demonstra a aplicação do gradiente em imagens RGB. Considerando \mathbf{r} , \mathbf{g} e \mathbf{b} vetores unitários associados aos eixos R, G e B do espaço de cores RGB, defini-se os vetores \mathbf{u} e \mathbf{v} pelas Equações 2.1 e 2.2 respectivamente.

$$\mathbf{u} = \frac{\partial R}{\partial x} \mathbf{r} + \frac{\partial G}{\partial x} \mathbf{g} + \frac{\partial B}{\partial x} \mathbf{b} \quad (2.1)$$

$$\mathbf{v} = \frac{\partial R}{\partial y} \mathbf{r} + \frac{\partial G}{\partial y} \mathbf{g} + \frac{\partial B}{\partial y} \mathbf{b} \quad (2.2)$$

Os vetores \mathbf{u} e \mathbf{v} são funções do espaço de coordenadas x, y . Logo, é demonstrado em (ZENZO, 1986) que a direção da taxa da máxima de variação é dada pelo ângulo da Equação 2.3.

$$\theta(x, y) = \frac{1}{2} \arctan \frac{2g_{xx}}{g_{yy} - g_{xx}} \quad (2.3)$$

Onde,

$$g_{xx} = \left| \frac{\partial R}{\partial x} \right|^2 + \left| \frac{\partial G}{\partial x} \right|^2 + \left| \frac{\partial B}{\partial x} \right|^2 \quad (2.4)$$

$$g_{yy} = \left| \frac{\partial R}{\partial y} \right|^2 + \left| \frac{\partial G}{\partial y} \right|^2 + \left| \frac{\partial B}{\partial y} \right|^2 \quad (2.5)$$

$$g_{xy} = \frac{\partial R}{\partial x} \frac{\partial R}{\partial y} + \frac{\partial G}{\partial x} \frac{\partial G}{\partial y} + \frac{\partial B}{\partial x} \frac{\partial B}{\partial y} \quad (2.6)$$

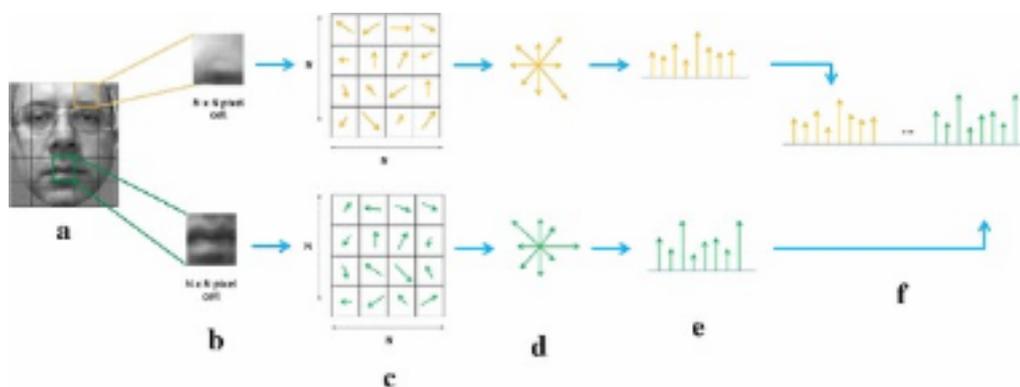
Finalmente, segundo (GONZALEZ; WOODS, 2009), o valor da taxa de variação em (x, y) para valores de θ no intervalo semiaberto $[0, \pi)$, na direção de $\theta(x, y)$ é dado pela Equação 2.7.

$$F_{\theta}(x, y) = \left\{ \frac{1}{2} \left| \frac{(g_{xx} + g_{yy}) + (g_{xx} - g_{yy}) \cos 2\theta(x, y) + 2g_{xy} \sin 2\theta(x, y)}{\cos 2\theta(x, y) + 2g_{xy} \sin 2\theta(x, y)} \right| \right\}^2 \quad (2.7)$$

2.5.6.3 Cálculo dos histogramas dos gradientes

Nesta etapa a imagem é dividida em células de dimensões de $N \times N$ pixels. Por exemplo, em uma imagem de dimensões 128×128 pixels, com células de 4×4 pixels, a imagem é dividida em 32×32 células, contabilizando 1024 células no total. Para cada célula constrói-se o histograma das orientações dos gradientes, calculando o valor da magnitude do gradiente e sua orientação. Para construir o histograma, é preciso definir o número de orientações. O número de orientações define o número de bins do histograma.

Figura 15 – Processo de extração das características do HOG: a imagem é dividida em células de dimensões $N \times N$ pixels. A orientação de todos os pixels é computada e acumulada em um histograma de orientações de M -bins. Finalmente, as células dos histogramas são agrupadas e o vetor final é construído. Neste exemplo a imagem é dividida em 4×4 células com 8 bins representando as orientações.



Fonte – (CARCAGNÌ et al., 2015).

2.5.6.4 Normalização dos gradientes

Na quarta etapa é realizado a normalização dos histogramas. A normalização melhora a invariância a iluminação. Neste processo, as células são agrupadas em blocos, e cada bloco é normalizado separadamente. Os descritores de bloco normalizados são referidos como descritores do Histograma de Gradientes Orientados (HOG).

2.5.6.5 Formação de um vetor provindo dos histogramas dos gradientes

O último passo consiste em coletar os descritores de cada bloco e agrupá-los em um vetor. Esse vetor representa o descritor final do processo. A Figura 15 demonstra o funcionamento de cada etapa do processo de extração do descritor HOG.

2.6 Técnica de aprendizado de máquina SVM

Aprendizado de Máquina (AM) é um domínio de Inteligência Artificial (IA) que visa melhorar o desempenho de tarefas por meio de experiências. Devido a essa capacidade de aprender, os métodos de AM vêm sendo utilizados em diversas tarefas de reconhecimento de padrões. Na prática de processamento de imagens, esses padrões são geralmente agrupados em um vetor que representa certas características da imagem, como forma, cor e textura. Esse vetor é denominado descritor.

Na literatura, o aprendizado é comumente dividido em dois grupos de acordo com sua interação com os dados. No aprendizado supervisionado, a aprendizagem necessita de um supervisor externo que indique e ensine a resposta certa. Essa técnica é amplamente

utilizada em aplicações de reconhecimento e classificação de imagens. Nesse caso, um conjunto de imagens devidamente anotadas é utilizado para treinar um algoritmo de classificação de objeto. Após treinado, o algoritmo é capaz de induzir a classe desse objeto. No aprendizado não-supervisionado, a aprendizagem é feita sem a ajuda da supervisão externa. Em tarefas de aprendizado não-supervisionado, o objetivo é explorar, descrever e descobrir similaridades entre os dados (FACELI et al., 2011) (SHALEV-SHWARTS; BEN-DAVID, 2014).

Neste trabalho, o algoritmo SVM de aprendizado supervisionado é abordado. As Máquinas de Vetores de Suporte (SVMs) consistem em um método de aprendizado embasado na Teoria de Aprendizagem Estatística. Essa técnica de AM é amplamente aplicada, obtendo-se ótimos resultados, em problemas de classificação, regressão e detecção de *outliers*. A origem das SVMs provém de diversos trabalhos teóricos de estatística desenvolvidos durante anos por diferentes autores, ganhando uma maior notabilidade no trabalho de (VAPNIK, 1995). As SVMs se desenvolveram na ordem reversa do desenvolvimento das Redes Neurais Artificiais (RNAs). Enquanto as RNAs evoluíram de aplicações e experimentos para então receberem uma formulação mais teórica, as SVMs tiveram o caminho inverso, evoluíram de uma base teórica para depois serem aplicadas em experimentos (LORENA; CARVALHO, 2007), (KECMAN, 2005), (STEINWART, 2014).

As SVMs lineares buscam separar os dados de cada classe através de um hiperplano. Seja um conjunto de pontos x e y de duas classes distintas, a Figura 16 demonstra um exemplo de possíveis hiperplanos que podem separar os pontos das duas classes.

A equação de um hiperplano é dado pela Equação 2.8. Nota-se que é possível obter um número infinito de hiperplanos.

$$f(x) = w \cdot x + b = 0 \quad (2.8)$$

Busca-se então maximizar a margem de separação dos dados. Logo, procura-se resolver o problema de otimização, (LORENA; CARVALHO, 2007), (CERVANTES et al., 2020):

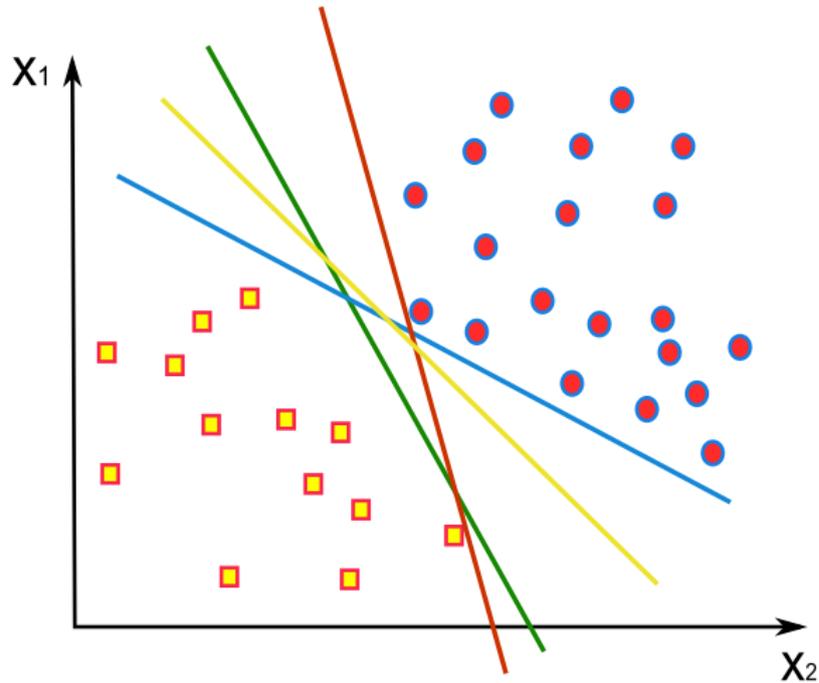
$$\min_{w,b} \frac{1}{2} \|w\|^2 \quad (2.9)$$

Com as seguintes restrições:

$$y_i(w \cdot x_i + b) - 1 \geq 0, \forall_i = 1, \dots, n \quad (2.10)$$

Esse problema de otimização pode ser resolvido através da técnica matemática de multiplicadores de Lagrange. A Figura 17 apresenta as representações geométricas dos hiperplanos $H1 = w \cdot x + b = 1$ e $H2 = w \cdot x + b = -1$, e com um hiperplano ótimo

Figura 16 – Separação das classes através de hiperplanos.



Fonte – (CERVANTES et al., 2020).

$H = w \cdot x + b = 0$. Esse é o caso de dados que são linearmente separáveis, onde as SVMs são conhecidas como SVMs lineares com margens rígidas.

Entretanto, na prática dificilmente os dados são linearmente separáveis. Logo, adiciona-se uma variável de folga ξ_i . Neste novo caso, denominado de SVMs lineares com margens suaves, procura-se resolver o problema de otimização, (LORENA; CARVALHO, 2007):

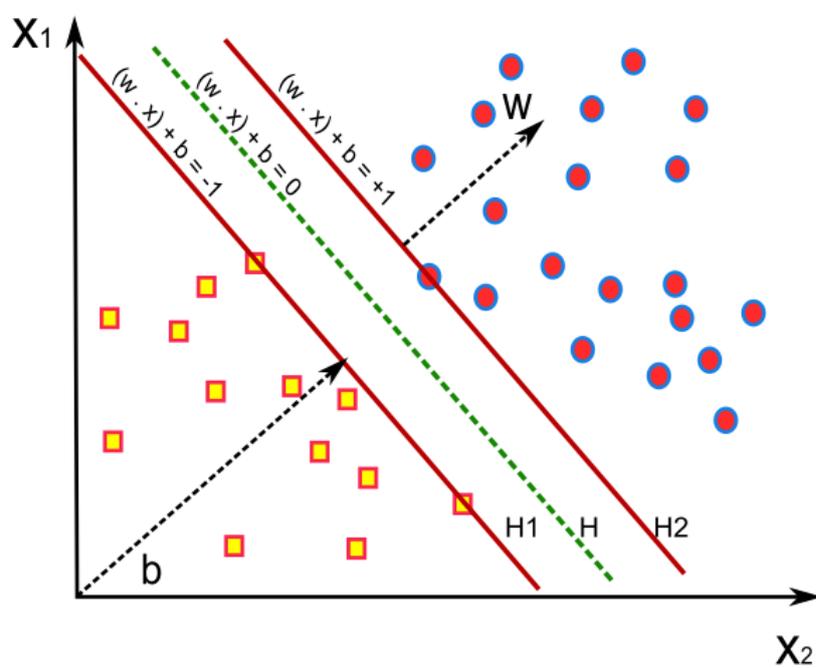
$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \left(\sum_{i=1}^n \xi_i \right) \quad (2.11)$$

Com as seguintes restrições:

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \xi_i \geq 0, \forall_i = 1, \dots, n \quad (2.12)$$

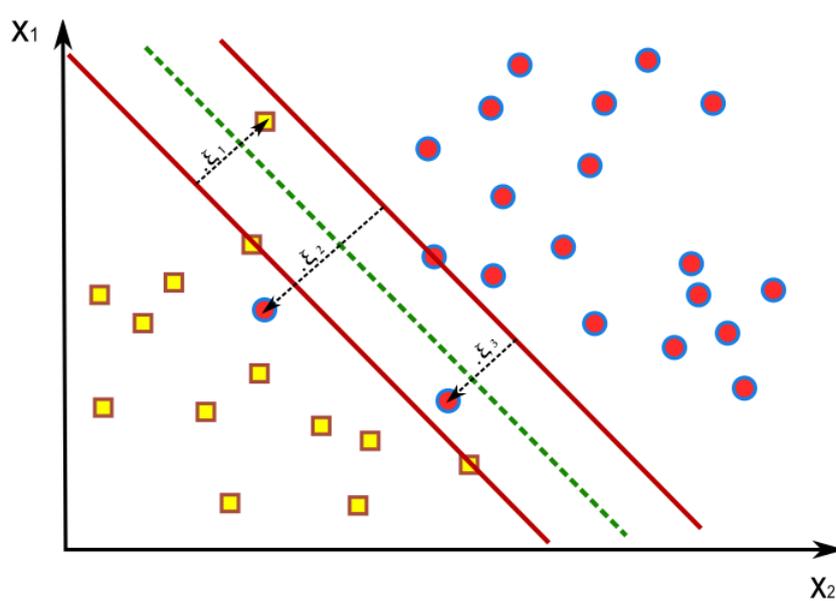
Na Figura 18 é apresentado a separação dos pontos de duas classes pelas SVMs com margens suaves. Observa-se na Figura 18 que alguns pontos estão localizados fora da região de sua classe. No caso da Figura 18, seria preciso utilizar um *kernel* não linear para separar todos os pontos das classes.

Figura 17 – SVMs com margens rígidas.



Fonte – (CERVANTES et al., 2020).

Figura 18 – SVMs com margens suaves.



Fonte – (CERVANTES et al., 2020).

2.7 Trabalhos Relacionados

Nesta seção, primeiramente, é apresentada uma breve discussão de trabalhos da literatura sobre inspeção de placa nua. Logo após, os trabalhos de inspeção de placa montada são discutidos. Como este trabalho visa ao desenvolvimento de um protótipo de inspeção de placas montadas, os trabalhos relacionados a esse tópico receberam uma maior ênfase na discussão.

2.7.1 Placa nua

No estudo realizado por (TATIBANA; LOTUFO, 1997), um algoritmo de inspeção de trilhas é proposto. Com esse método, é possível detectar curto circuito e circuito aberto nas placas. O algoritmo consiste em analisar se dois furos conectados por trilhas na placa de referência também estão conectados na placa de teste. Em (LETA; FELICIANO; MARTINS, 2008), o algoritmo de (TATIBANA; LOTUFO, 1997) foi aprimorado. Dessa forma, o sistema detecta diversos defeitos como falta de trilhas, curto-circuito, circuito aberto e falta de furos. Já nos trabalhos de (IBRAHIM; AL-ATTAS; ASPAR, 2002) e (SANTOYO et al., 2007), a detecção dos defeitos em placas nua é realizada por algoritmos baseados na transformada de wavelet.

Mais recentemente em (TANG et al., 2019), foi publicada uma base de dados contendo 1500 pares de imagens, onde cada par consiste em uma imagem de referência sem defeito e uma imagem de teste contendo os defeitos mais comumente encontrados em PCIs: circuito aberto, curto-circuito, falhas na trilha e nos furos. Com a ajuda dessa base de dado, foi possível comparar diversas técnicas de detecção de defeitos, onde comparou-se o método desenvolvido com algoritmos baseados em morfologia matemática (PUTERA et al., 2010), e com modelos de detecção de objetos baseados em *deep learning* como Yolo (*You Only Look Once*), Fast R-CNN e SSD (*Single Shot Multibox Detector*).

2.7.2 Placa montada

No trabalho de (HAN; MAZOUZ; SARAVANAN, 1993), através de uma rede neural de três camadas e com a ajuda do algoritmo *back propagation*, foi possível realizar a inspeção em placas com componentes faltando, componentes desalinhados e componentes no lugar errado. A entrada da rede neural consiste em uma imagem binária, onde um valor de *threshold* constante foi usado para realizar a binarização. Os dois neurônios de saída possuem valor de ativação 0 ou 1, o que indica se a placa possui falhas ou não.

No trabalho de (LIN; SU, 2006), o processo de inspeção de PCIs é realizado em duas etapas. O objetivo do estudo foi desenvolver uma estratégia para inspecionar mais rapidamente placas de circuito impresso. Na primeira etapa, algumas características das imagens são extraídas, como: quantidade de pixels brancos, histograma, coeficiente de

correlação, índice regional e índice de contraste alto. Em seguida, é realizado um teste rápido de comparação de índices, onde as imagens que passam o teste são classificadas como normais (sem defeito). As imagens que não passam no teste, são então classificadas na segunda etapa por um processo mais robusto. Essa segunda etapa consiste em um rede neural artificial que classifica a imagem em: componente normal, componente invertido, componente ausente ou componente deslocado.

Em (ACCIANI; BRUNETTI; FORNARELLI, 2006), um sistema AOI para inspecionar juntas de solda foi desenvolvido. O processo consiste em três etapas: uma etapa de pré-processamento, onde extrai-se a ROI da imagem; na segunda etapa são extraídas determinadas características da ROI; e na última etapa é realizado a classificação através de uma RNA. No pré-processamento, é realizado três etapas de segmentação, primeiramente para se obter a imagem da placa, posteriormente para se obter a imagem do componente, e finalmente se obtém a imagem da junta de solda a ser testada. Na segunda etapa é extraído dois tipos de características: característica geométrica, onde é definido um vetor composto pela área, perímetro, compacidade, altura e larguras; e a *wavelet*, onde o vetor é formado pelos coeficientes extraídos da aplicação da transformada discreta de *wavelet* (DWT). Na última etapa é realizada a classificação, onde compara-se diferentes métodos de classificação: uma rede neural LQV (*Learning Vector Quantization*); uma rede neural MLP (*Multilayer Perceptron*); e um classificador KNN (*K Nearest Neighbor*). Os algoritmos classificam as juntas de solda em cinco parâmetros: excesso de solda, excesso de solda porém aceitável, solda correta, pouca solda porém aceitável, solda insuficiente.

Em (WU et al., 2010), um algoritmo de inspeção da posição de componentes em PCIs foi desenvolvido obtendo a ROI do componente aplicando o método de segmentação adaptativa proposto no trabalho. além do algoritmo de inspeção de posicionamento do componentes, os três canais de cores RGB do corpo dos componentes são extraídos e um classificador de Bayes inspeciona se o corpo do componente fixado na PCI está errado.

No trabalho de (YOUN; LEE; PARK, 2014) uma rede neural de classificação de componentes SMD foi proposta. A rede neural recebe como dados de entrada característica extraídas dos histogramas das imagens no espaço de cor HSI (*hue, saturation e intensity*), e características das bordas obtidas pelo detector de bordas de Canny.

No trabalho de (MELLO; STEMMER, 2015), a ausência de componente é inspecionada através de características dos histogramas e da técnica de KNN. O contorno dos componentes são obtidos aplicando uma técnica de segmentação de Watershed em conjunto com o detector de bordas de Canny. Então, são calculados os 6 primeiros descritores de Fourier que são utilizados como entrada de uma rede neural de classificação de componentes. Os contornos também são utilizados para comparar a posição do componente comparando com uma placa de referência.

3 Solução Proposta

Neste capítulo é apresentado o funcionamento do sistema de inspeção proposto. O processo de inspeção proposto é dividido em duas etapas. A primeira etapa consiste em configurar o programa através de um referencial. Esse referencial pode ser tanto uma PCI de referência quanto o desenho elétrico da placa. A segunda etapa consiste em aplicar os algoritmos de processamento de imagens e aprendizado de máquina nas imagens das PCIs a serem analisadas. Nessa segunda etapa, inicialmente, através de um classificador de componentes, verifica-se em cada posição pré-determinada da placa se o componente ali soldado corresponde ao esperado. Se a classe de saída do classificador da placa testada não for a mesma que a classe do componente da placa de referência, setado na configuração inicial, o sistema gera um alerta ao operador. Quando esse alerta é acionado, na interface gráfica da aplicação é exibido a imagem do componente da placa de referência e a imagem do componente da placa de teste correspondente, para fins de comparação, assim o operador pode verificar visualmente qual foi o erro encontrado. Caso o componente passe a etapa de classificação de maneira positiva, é então verificado se o componente não está deslocado de sua posição de referência. Nessa etapa, é inspecionado a posição linear e angular do componente. Caso o componente esteja deslocado ou desalinhado, é gerado um alerta ao operador, como na etapa de classificação.

3.1 Configuração inicial através de um referencial

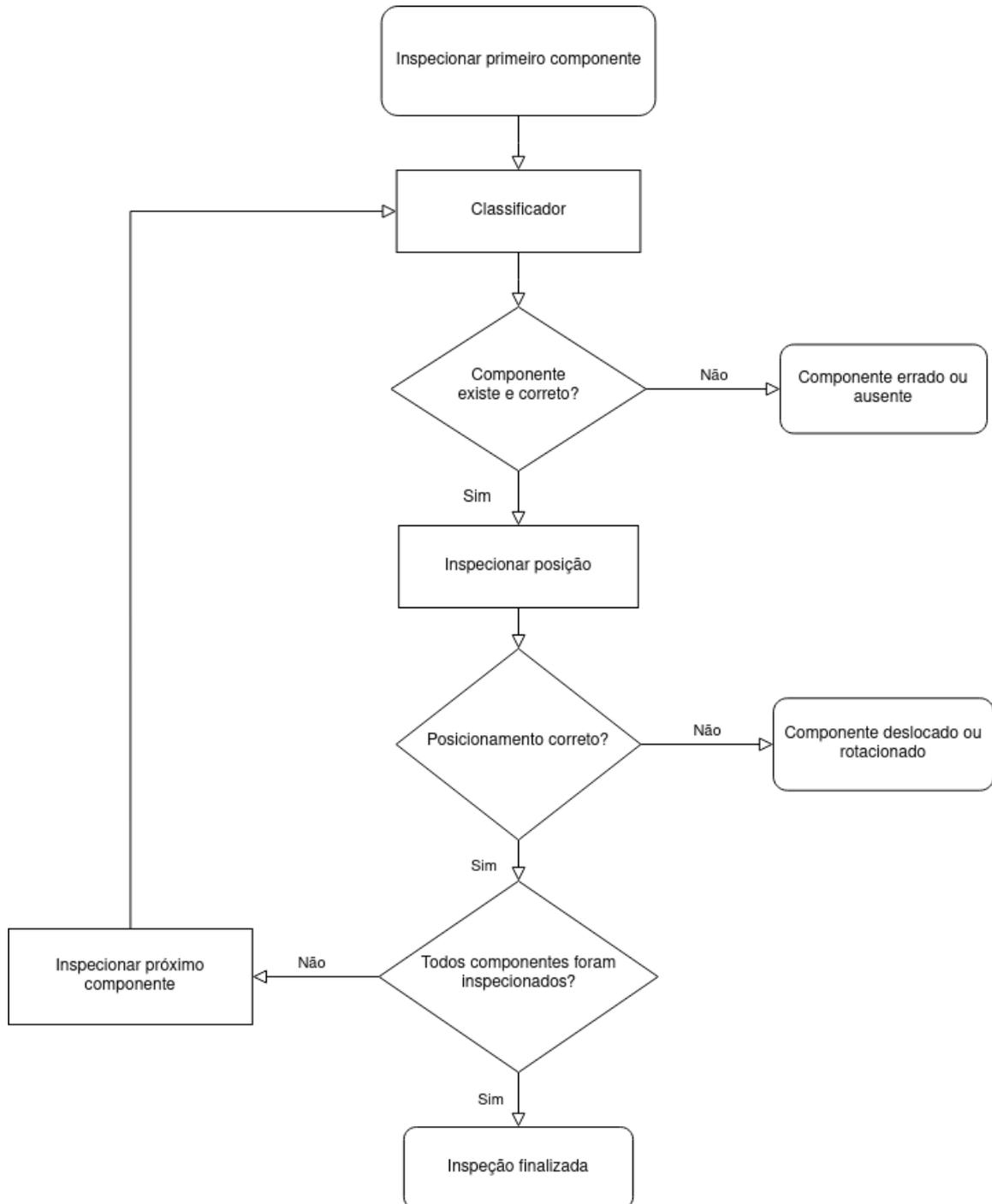
A partir de um referencial, é extraído a posição dos componentes os quais deseja-se inspecionar. Essa extração pode ser feita de duas formas, através de uma placa de referência (*golden board*), ou integrando diretamente com um *software* de *design* de PCIs, o que diminui o tempo de *setup*. Idealmente, essa etapa de configuração do sistema é executada apenas um vez para cada lote de placas de circuito impresso que se planeja fabricar.

3.2 Inspeção dos componentes

Após configurar o sistema com a placa de referência, inicia-se então o processo de inspeção. Sendo assim, as próximas placas a serem fabricadas são inspecionadas. A inspeção proposta consiste em analisar individualmente a região de interesse de cada componente da PCI. Para isso, primeiramente é aplicada a técnica de classificação de imagem para classificar o componente, e verifica-se então se o componente coincide com o esperado, o qual foi definido na fase de *setup* com a ajuda da referência. O classificador também deve indicar a falta de componente, erro que pode ocorrer durante o processo de

fabricação de PCIs. Se o tipo de componente estiver correto, o próximo passo é inspecionar a posição do mesmo. Nessa etapa são analisadas a inclinação e o deslocamento linear do componente. Essas etapas de classificação e inspeção do alinhamento são aplicadas para todos os componentes definidos na parte de configuração inicial. A Figura 19 demonstra o fluxograma de inspeção do sistema proposto.

Figura 19 – Fluxograma de inspeção da solução proposta.



3.2.1 Classificação de componentes

A classificação de componentes em PCIs é um processo desafiador por diversos motivos. A similaridade visual de resistores SMD e capacitores SMD é um deles. Outro desafio se encontra na dificuldade da preparação da base de dados de imagens, dificuldade essa devida ao desbalanceamento do número de cada componente encontrado em placas de circuito impresso. Geralmente, a quantidade de resistores em uma PCI é muito maior comparando aos demais componentes.

O classificador proposto neste trabalho consiste em um modelo inteligente capaz de classificar os principais componentes encontrados nas PCIs SMD, sendo assim, propõe-se desenvolver um classificador capaz de classificar componentes em cinco classes, sendo elas: resistor SMD, capacitor cerâmico SMD, capacitor eletrolítico, circuito integrado e ausência de componente. Escolheu-se as classes de resistores, capacitores e circuitos integrados pois são as classes de componentes mais encontradas em PCIs. A Figura 20 demonstra o fluxograma contendo as etapas de funcionamento e as classes de classificação propostas, e a Figura 21 apresenta uma amostra de imagem para cada classe de classificação.

Figura 20 – Fluxograma do classificador proposto.

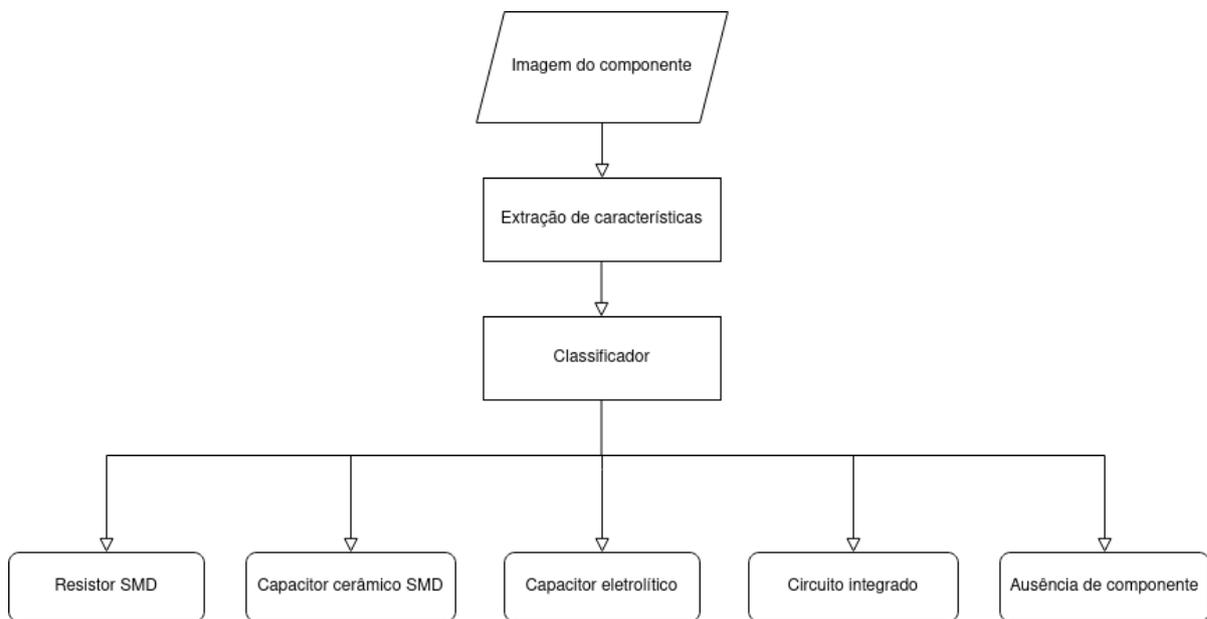
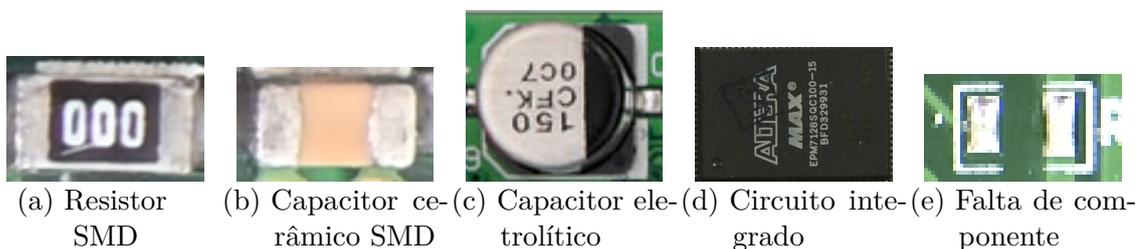


Figura 21 – Amostra de imagem para cada classe de classificação.



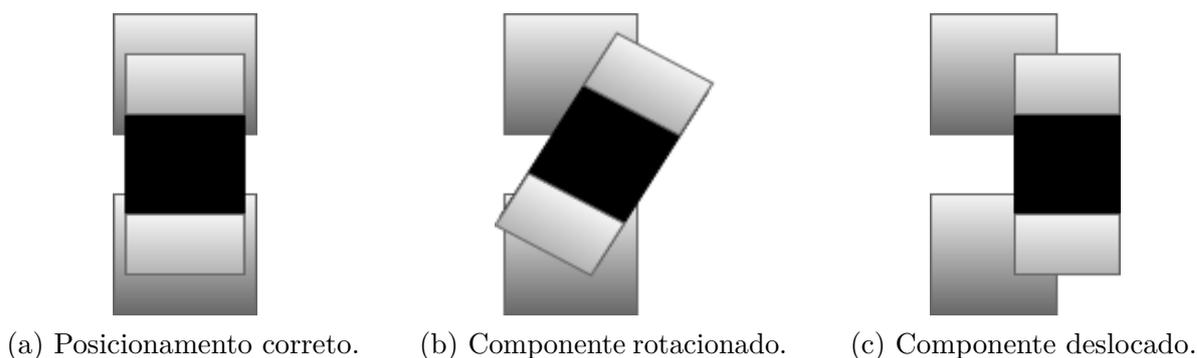
Nota-se no fluxograma da Figura 20, que há um bloco destinado a extração de características da imagem. Nessa etapa são aplicados dois algoritmos com o objetivo de caracterizar a forma, textura e cor dos componentes. A caracterização da forma e textura é feita pelo descritor HOG, apresentado teoricamente na Seção 2.5.6. Por outro lado, o descritor de cor é obtido pelo cálculo do histograma da imagem no domínio HSV. O referencial teórico de histograma e do domínio de cores HSV são apresentados na Seção 2.5.3 e Seção 2.5.2 respectivamente. A forma de aplicação desses métodos é explicada mais detalhadamente na Seção 4.6.1 e Seção 4.6.2, para HOG e HSV respectivamente.

3.2.2 Inspeção da posição do componente

Se o componente inspecionado da PCI de teste passar pelo teste do classificador, aplica-se então os algoritmos de inspeção da posição linear e angular. A inspeção da posição consiste em duas checagens, mede-se tanto o ângulo de inclinação do componente quanto o deslocamento linear de seu centro de massa em relação ao centro da ROI, ou seja, mede-se o deslocamento linear do centroide do componente de teste em relação ao centroide do componente de referência.

A Figura 22 apresenta exemplos de possíveis posicionamentos de componentes na PCI. Na Figura 22a é demonstrado um componente considerado correto em termos de posição. Nas Figuras 22b e 22c, por sua vez, são apresentados exemplos de componentes defeituosos em relação ao seu posicionamento. Na Figura 22b é demonstrado um exemplo de um componente rotacionado e na Figura 22c é apresentado um exemplo de componente deslocado linearmente.

Figura 22 – Posicionamentos possíveis do componente na PCI.



Nesta etapa se desenvolveu uma forma de analisar a posição de componentes do tipo resistor SMD e capacitor cerâmico SMD, pois são os dois componentes mais utilizados nas PCIs e os mais suscetíveis a erros. Os detalhes de implementação dos métodos de inspeção da posição são apresentados e discutidos com mais detalhes na Seção 4.7.

4 Metodologia Experimental

Nesta seção é apresentado a base de dados utilizada para desenvolver este trabalho, a descrição detalhada de cada etapa de inspeção do projeto proposto, e os algoritmos de processamento digital de imagens e aprendizado de máquina empregados.

4.1 Base de dados

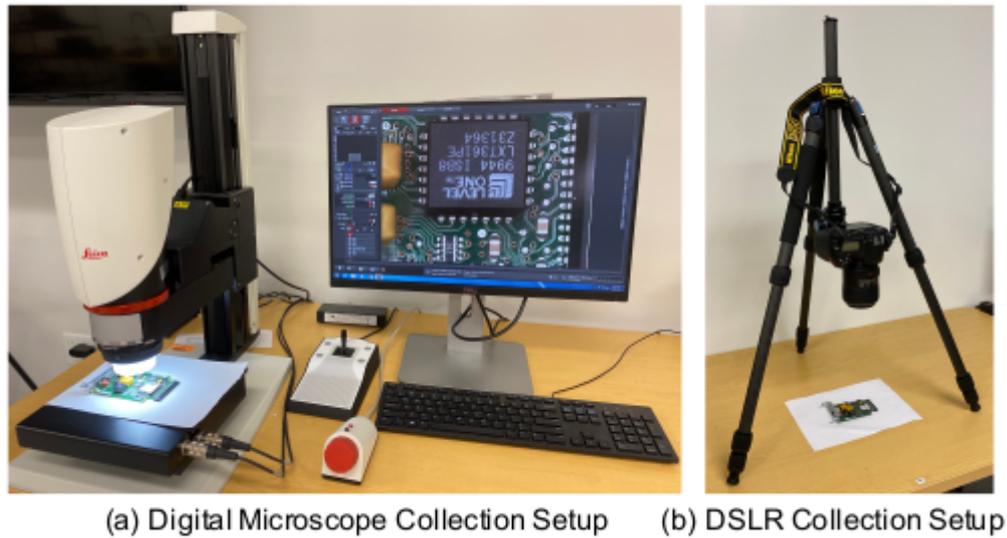
A inspeção óptica automatizada de placas de circuito impresso é um processo de bastante interesse comercial e cada vez mais empregado na indústria. Entretanto, até o momento, o número de banco de imagens publicados e de livre acesso é limitado. Essa limitação dificulta o avanço da pesquisa nessa área, pois inviabiliza comparar a performance dos algoritmos de inspeção. Muitos dos artigos publicados na área, também, tiveram seus algoritmos testados em banco de dados privados, o que dificulta a reprodutibilidade dos experimentos. Felizmente, nos últimos anos, alguns trabalhos foram publicados com intuito de atender essa necessidade. Os trabalhos de (HUANG; WEI, 2018) e (TANG et al., 2019) são publicações de banco de imagens placas nuas. Como discutido na seção 2, as imagens de PCIs nuas são capturas com o objetivo de inspecionar principalmente as trilhas das placas. Já os trabalhos de (PRAMERDORFER; KAMPEL, 2015), (KUO et al., 2019), (MAHALINGAM; GAY; RICANEK, 2019) e (LU et al., 2020) apresentam imagens de placas com componentes soldados.

Neste trabalho utilizou-se principalmente as imagens publicas em (LU et al., 2020). Esta base de dados possui 31 PCIs contendo 2.971 CIs, 36.112 capacitores, 32.766 resistores, 1.163 indutores, 1.349 transistores e 1.604 diodos. As imagens foram obtidas de duas maneiras. Primeiramente obteve-se as imagens através de um Microscópio Digital, e posteriormente foram obtidas imagens das mesmas PCIs com uma câmera DSLR. A Figura 23 demonstra a configuração montada para a realizar a captura das imagens.

4.2 Linguagem de programação e bibliotecas

A solução proposta foi desenvolvida em linguagem de programação Python. Python é uma linguagem de alto nível e de código aberto, o que impulsionou o desenvolvimento de diversas bibliotecas e pacotes *open source*. As técnicas de processamento de imagens foram implementadas através das bibliotecas OpenCV e scikit-image. OpenCV (*Open Source Computer Vision*) é uma biblioteca escrita em C/C++ que fornece um grande ferramental para o desenvolvimento de aplicações em Visão Computacional. Scikit-image é um biblioteca escrita principalmente em Python, e com alguns algoritmos escritos

Figura 23 – Configuração para a aquisição de imagens da base de dados FICS-PCB.



Fonte – (LU et al., 2020).

em Cython para melhorar o desempenho, que disponibiliza uma coleção de algoritmos implementados que facilitam no desenvolvimento de aplicações na área de processamento de imagens. Em conjunto com essas ferramentas, utiliza-se o pacote NumPy, uma poderosa ferramenta de cálculo numérico, que permite realizar operações em vetores e matrizes de forma eficiente, facilitando assim realizar operações e manipular imagens. O modelo de aprendizado de máquina que realiza a classificação dos componentes foi implementado com a ajuda do módulo de código aberto scikit-learn, que facilita a implementação de diversos algoritmos de classificação, regressão e agrupamento (*clustering*). A interface gráfica da aplicação foi desenvolvida com a ajuda do pacote tkinter, que possibilita criar aplicativos com botões em Python, com o intuito de facilitar o manuseio do usuário. Para exibir as imagens na interface gráfica utilizou-se a biblioteca Pillow em conjunto com tkinter.

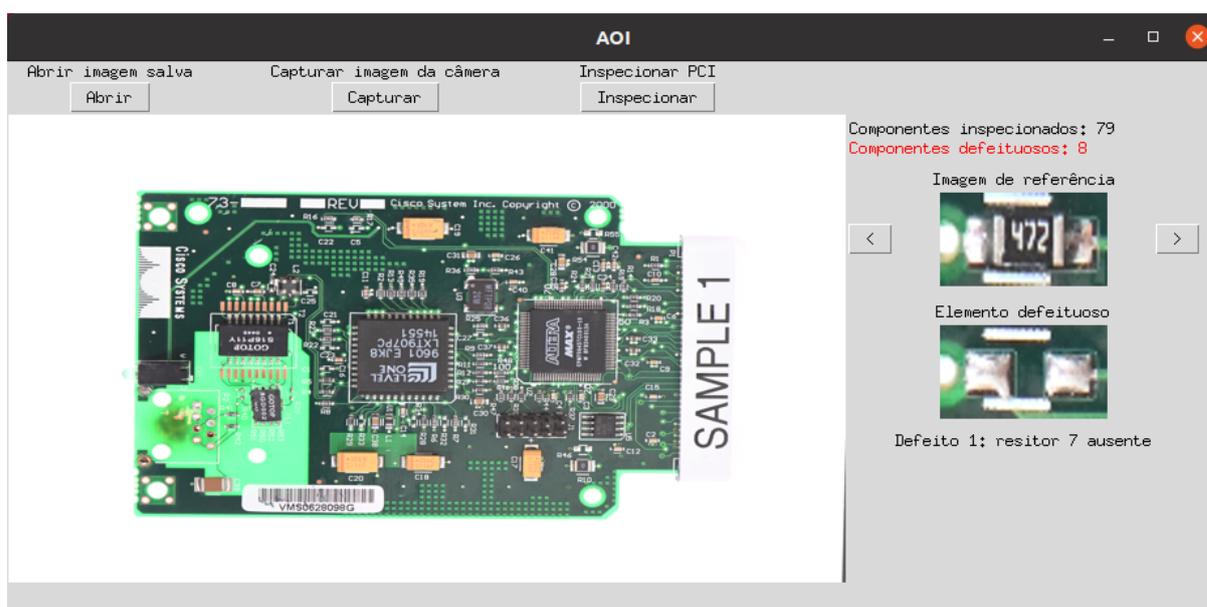
4.3 Descrição do Hardware

Todo o desenvolvimento da aplicação e experimentos realizados foram implementados em um computador pessoal contendo as características apresentados na Tabela 1.

Tabela 1 – Descrição do hardware utilizado no desenvolvimento do trabalho.

| | |
|----------------|---|
| Processador | Intel® Core™ i7-5500U CPU @ 2.40GHz x 4 |
| Memória RAM | 16 Gb |
| Placa de vídeo | Mesa Intel® HD Graphics 5500 (BDW GT2) |
| Disco rígido | 1 TB |

Figura 24 – Interface da aplicação.



Fonte – A imagem da PCI foi extraída da base de imagens publicada em (LU et al., 2020).

4.4 Interface Gráfica

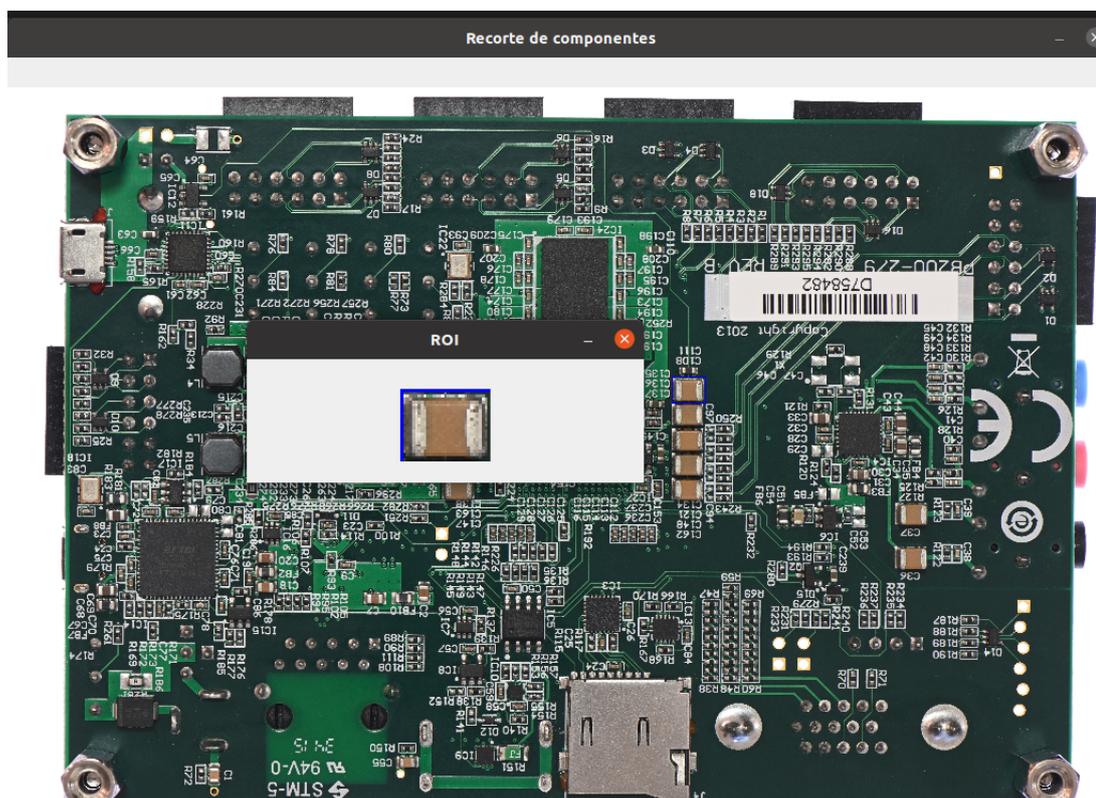
Como apresentado na Seção 4.4, a interface gráfica da aplicação foi desenvolvida utilizando o pacote tkinter. A interface permite ao usuário, por meio de botões, escolher uma imagem de PCI salva na máquina ou capturar uma imagem de uma nova placa. O botão Inspeccionar quando pressionado, inicializa a rotina de inspeção, aplicando o classificador e a verificação do posicionamento para todos os componentes inseridos no arquivo .csv da placa correspondente. Ao final da inspeção, é exibido na janela o número total de componentes inspeccionados e a quantidade de defeitos encontrados. Quando um defeito é encontrado, é exibido na janela principal do aplicativo a imagem do componente defeituoso, assim como a imagem de referência, para que o operador possa visualizar o defeito encontrado, de forma a ajudá-lo na tomada de decisões para assegurar que o erro não volte a ocorrer. A janela principal da interface gráfica é apresentada na Figura 24.

4.5 Aquisição das coordenadas dos componentes

Neste trabalho foi desenvolvido um *software* para extrair as coordenadas dos componentes da PCI de referência. Com a imagem da placa de referência capturada, o operador da máquina faz o recorte dos componentes com a ajuda do *mouse* e seleciona o nome do componente correspondente. Esses dados são salvos em um arquivo .csv contendo as coordenadas $xmin$, $ymin$, $xmax$, $ymax$ dos componentes na imagem. A Figura 25 apresenta um exemplo de recorte de um capacitor cerâmico SMD. A janela intitulada “Recorte de componentes” da Figura 25, apresenta uma imagem de PCI capturada e

exibida na interface gráfica da aplicação desenvolvida. A janela intitulada “ROI” da Figura 25 exibe o recorte desse componente. O operador então seleciona a classe do componente correspondente, capacitor cerâmico no caso da Figura 25, e a aplicação salva suas coordenadas em um arquivo *.csv*.

Figura 25 – Recorte dos componentes.



Fonte – A imagem da PCI foi extraída da base de imagens publicada em (LU et al., 2020).

4.6 Classificador

Para o desenvolvimento do classificador de componentes, considerou-se inicialmente implementar um classificador utilizando uma rede neural convolucional. Entretanto, para treinar um modelo de classificação de imagens através de uma rede neural artificial e obter resultados satisfatórios, precisa-se ter uma quantidade considerável de imagens de cada classe de componente. Ademais, o tempo de treinamento e inferência em um computador pessoal com as características descritas na Seção 4.3 pode ser bastante elevado. Desse modo, optou-se por implementar o método de aprendizado de máquina SVM apresentado teoricamente na Seção 2.6. As SVMs são conhecidas por proporcionar bons resultados mesmo quando a quantidade de dados para treinamento não é tão elevada, e os tempos de treinamento e inferência com um *kernel* linear é bem menor comparados aos tempos de treinamento de redes neurais.

Para o treinamento do classificador, extraiu-se características de forma, textura e cor das imagens, criando um vetor contendo as características desejadas. Sendo assim, o modelo de AM foi treinado com os vetores de características de cada imagem. Para caracterizar forma e textura, implementou-se o descritor HOG apresentado na Seção 2.5.6. As características de cor por sua vez são extraídas com a ajuda do histograma das imagens no espaço de cores HSV. O embasamento teórico da técnica de histograma e do domínio de cores HSV são apresentados na Seção 2.5.3 e Seção 2.5.2 respectivamente. Para a obtenção do descritor HOG, as imagens dos componentes foram redimensionadas para uma resolução de 96 x 96 pixels. As imagens então foram divididas em 36 células (6 x 6) de 16 x 16 pixels cada. Então, obteve-se os histogramas contendo 8 bins para cada célula, totalizando assim um vetor final de dimensão $36 \times 8 = 288$ para o HOG. Para o descritor de cor, primeiramente converteu-se a imagem RGB do componente para o espaço HSV. Calculou-se então o histograma de 8 bins para cada canal de cor, totalizando um vetor com $8^3 = 512$ valores. Os dois vetores obtidos foram normalizados e unidos para formar o descritor final que é usado como entrada do modelo de classificação.

A base de dados de treinamento e teste do modelo de classificação construída consiste em 400 imagens de cada classe. Foram estipuladas cinco classes de classificação: resistor SMD, capacitor cerâmico SMD, capacitor eletrolítico, circuito integrado e ausência de componente, totalizando 2000 imagens.

4.6.1 Histogram of Oriented Gradients

O primeiro descritor utilizado para treinar o classificador foi o HOG. As etapas para o cálculo do HOG são descritas na Seção 2.5.6. Seguindo os passos estipulados no embasamento teórico, primeiramente realiza-se a normalização *gamma*. No segundo passo, calcula-se os gradientes nas direções x e y. Os gradientes de primeira ordem podem ser calculados aplicando os filtro de convolução $[-1, 0, 1]$ e $[-1, 0, 1]^T$. Então, calcula-se a magnitude e orientação dos gradientes, conforme apresentado na Equação 4.1 e Equação 4.2 respectivamente.

$$g = \sqrt{g_x^2 + g_y^2} \quad (4.1)$$

$$\theta = \arctan \frac{g_y}{g_x} \quad (4.2)$$

Nas Figuras 26, 27, 28, 29 e 30 são apresentados as imagens resultantes da aplicação dos gradientes em x e y, e também a imagem resultante da magnitude do gradiente para uma amostra de cada componente.

Figura 26 – Aplicação dos gradientes no resistor SMD.

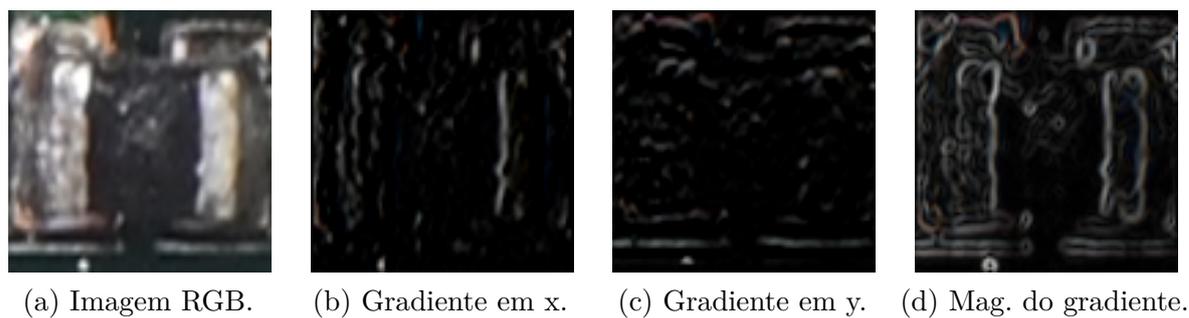


Figura 27 – Aplicação dos gradientes no capacitor cerâmico SMD.

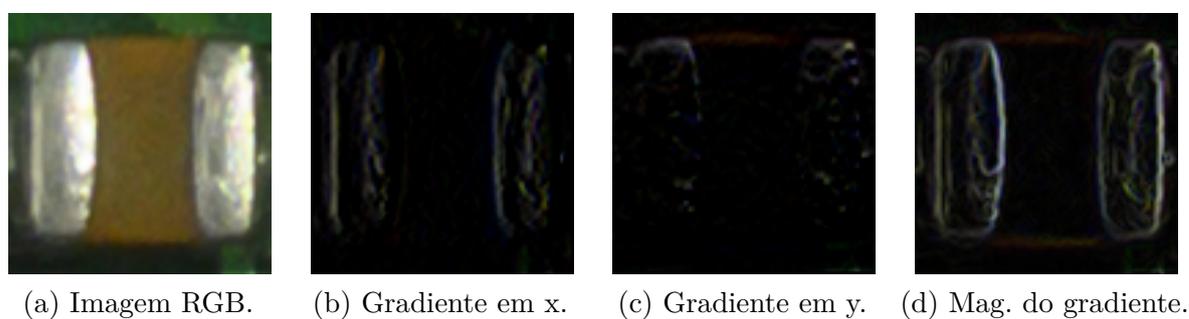


Figura 28 – Aplicação dos gradientes no capacitor eletrolítico.

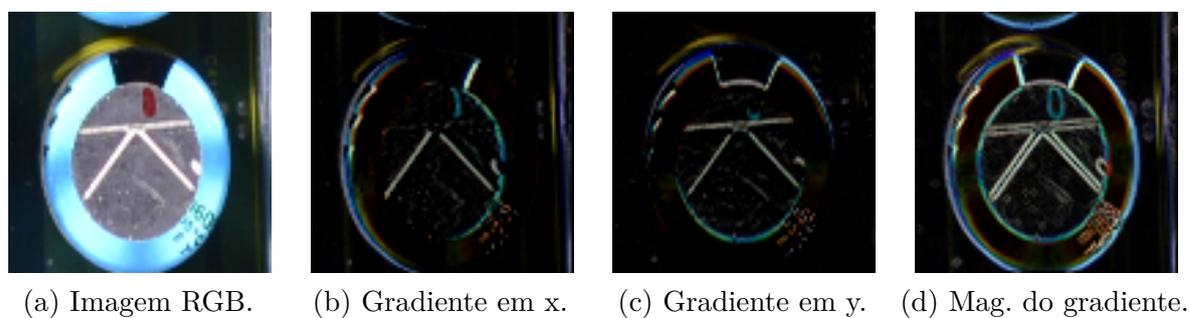


Figura 29 – Aplicação dos gradientes no circuito integrado.

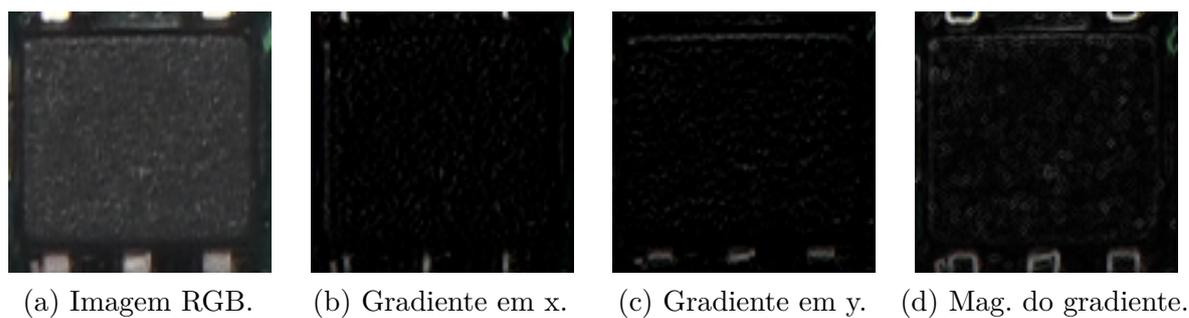
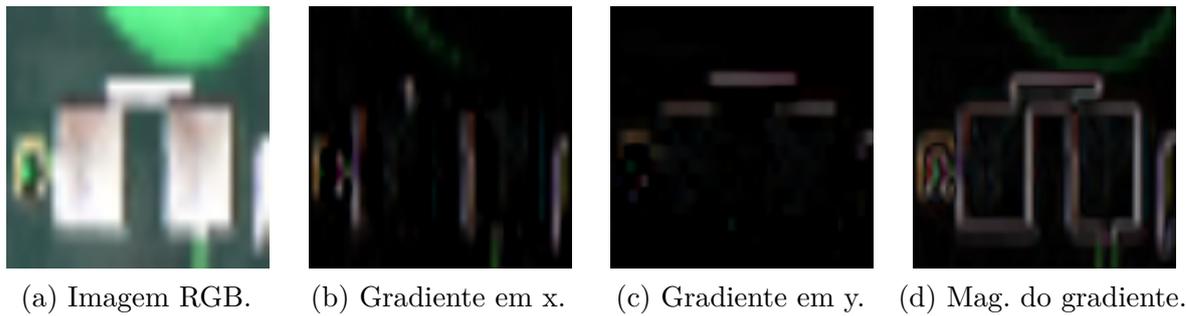
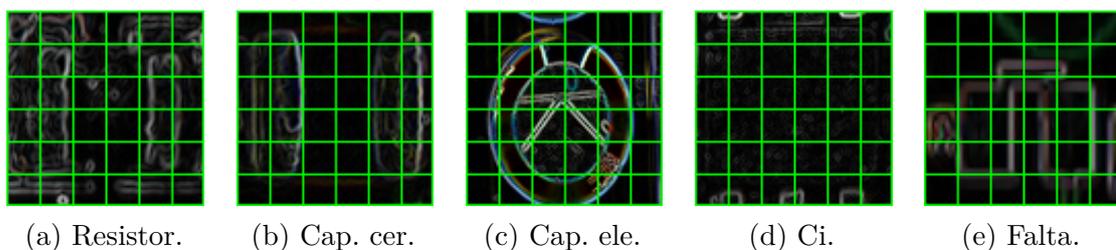


Figura 30 – Aplicação dos gradientes na falta de componente.



O terceiro passo para o cálculo do HOG consiste em dividir a imagem em células de $n \times n$ pixels. Posteriormente, para cada uma dessas células é calculado o histograma dos gradientes. Para o cálculo do histograma, foi utilizado 8 orientações, ou seja, um histograma com 8 bins. Na Figura 31 é demonstrado a divisão das imagens em células de 16×16 pixels. As imagens de entrada foram redimensionadas para uma resolução de 96×96 pixels. Logo, a imagem dividida contém 6×6 células. No próximo passo, as células são agrupadas em blocos e é realizado a normalização dos histogramas. O último passo consiste em agrupar os histogramas dos gradientes em forma vetorial. Esse vetor contém as características de forma e textura desejadas para servir como entrada do modelo de classificação. Como cada histograma foi computado considerando 8 bins, e a imagem foi dividida em 36 células, obteve-se um vetor de dimensão $36 \times 8 = 288$.

Figura 31 – Divisão da imagem em células de 16×16 pixels.

Nas Figuras 32, 33, 34, 35 e 36 são exibidos os resultados da aplicação dos histogramas de gradientes orientados para cada classe de componente.

Figura 32 – HOG do resistor SMD.

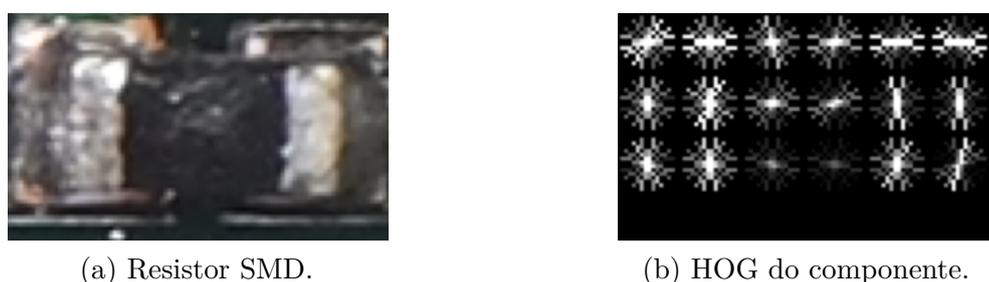
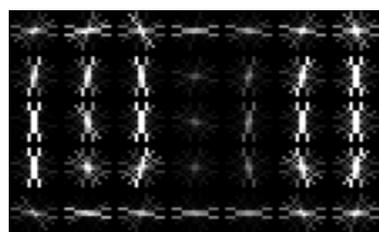


Figura 33 – HOG do capacitor cerâmico SMD.

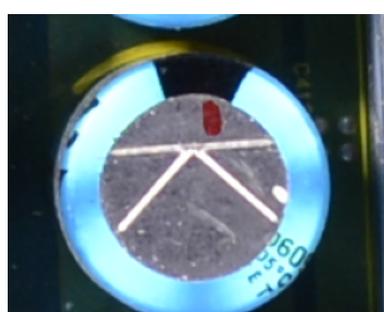


(a) Capacitor cerâmico SMD.

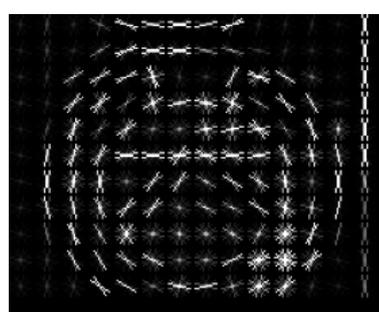


(b) HOG do componente.

Figura 34 – HOG do capacitor eletrolítico.



(a) Capacitor eletrolítico.

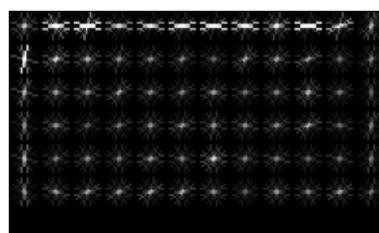


(b) HOG do componente.

Figura 35 – HOG do circuito integrado.



(a) Circuito integrado.

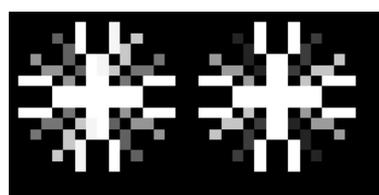


(b) HOG do componente.

Figura 36 – HOG da ausência de componente.



(a) Ausência de componente.



(b) HOG do componente.

4.6.2 Descritor de cores no espaço HSV

As características de cor dos componentes foram obtidas através do cálculo do histograma das imagens no espaço de cores HSV. Primeiramente, realiza-se a transformação

das imagens RGB para HSV. As Figuras 37, 38, 39, 40 e 41 apresentam as imagens resultantes da transformação das imagens do espaço RGB para HSV. O vetor de características foi obtido pela representação do histograma com 8 bins da imagem para cada canal do espaço HSV, obtendo um vetor de dimensão $8^3 = 512$. Observa-se que o descritor de cor é fundamental para distinguir os componentes das classes resistor e capacitor cerâmico, pois analisando o HOG dos mesmos nas Figuras 32b e 33b, pode-se notar que os resultados obtidos são muito similares, o que tende a dificultar a classificação pelo método das SVMs lineares. Porém, é possível diferenciar os capacitor cerâmicos SMD dos resistores SMD pela cor do seu interior. Nota-se nas imagens RGB das Figuras 37a e 38a que a cor do resistor SMD tende ao preto, e do capacitor cerâmico para o marrom. Essa diferença é mais perceptível no espaço de cores HSV, como apresentado nas Figuras 37b e 38b.

Figura 37 – Imagem HSV do resistor SMD.

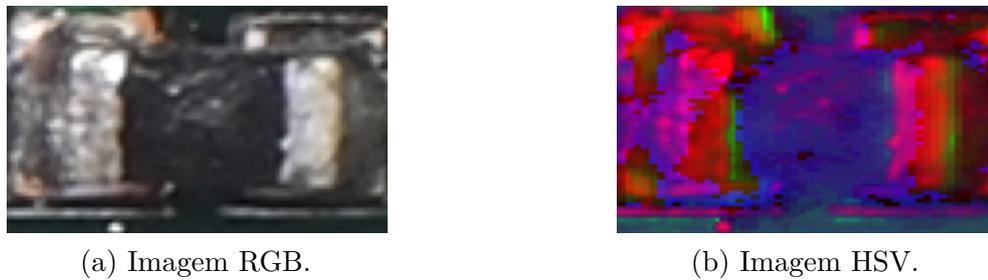


Figura 38 – Imagem HSV do capacitor cerâmico SMD.



Figura 39 – Imagem HSV do capacitor eletrolítico.

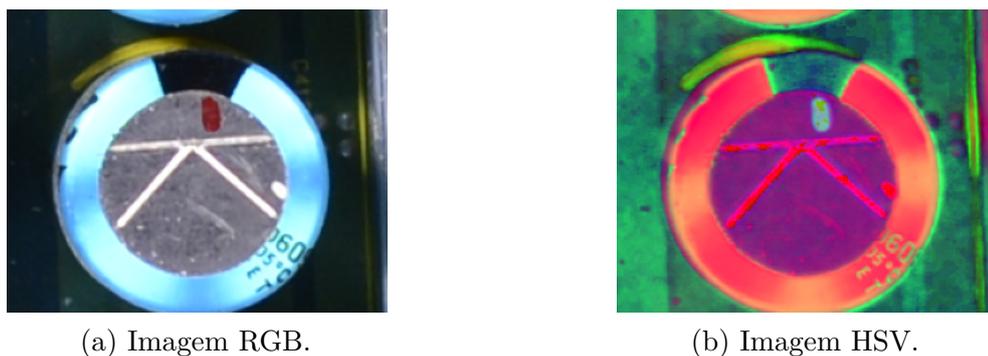


Figura 40 – Imagem HSV do circuito integrado.

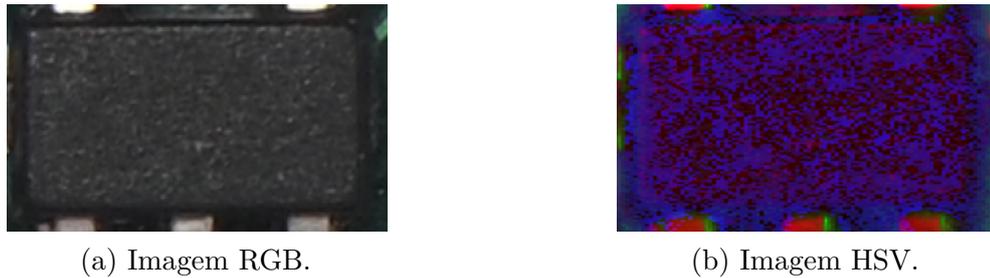
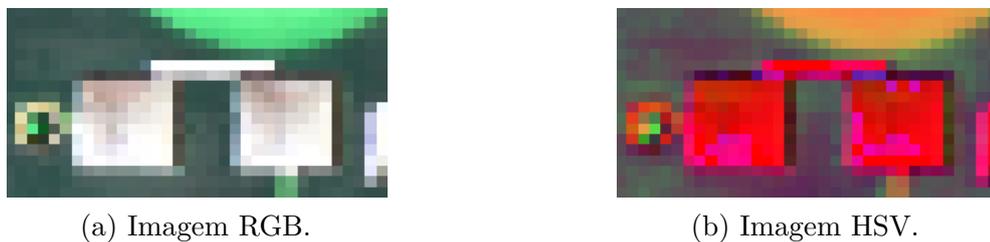


Figura 41 – Imagem HSV da ausência de componente.



4.7 Posicionamento

Para obter a posição do componente na PCI, primeiro é necessário realizar uma etapa de pré-processamento com o intuito de segmentar o componente. Primeiramente, amplia-se a janela de visualização do componente, selecionando uma imagem com 40 pixels a mais nas direções x e y. Nessa nova imagem, aplica-se a transformação para escala de cinza. Na imagem em escala de cinza, é aplicado a limiarização de Otsu, discutida na Seção 2.5.4. São realizadas então operações morfológicas de erosão, fechamento e abertura para eliminar os ruídos. Posteriormente analisa-se o número dos *blobs* encontrado com uma área maior que a determinada. Nessa etapa, dois casos são possíveis. No caso 1, um único *blob* é detectado. No caso 2, dois *blobs* de menor área são detectados. Para o caso 2, é necessário unir os dois *blobs* gerando um único *blob*. Por fim, procura-se desenhar um retângulo de menor área possível em torno do *blob* encontrado. Na Figura 42 é apresentado as imagens resultantes de cada operação aplicada para a segmentação do caso 1. Na Figura 43 é demonstrado as imagens resultantes para a segmentação do caso 2.

As Figuras 42a e 43a apresentam as imagens de entrada no espaço RGB. Em seguida, as Figuras 42b e 43b demonstram os resultados da conversão da imagem para a escala de cinza. As Figuras 42c e 43c apresentam os resultados da binarização Otsu. Nas Figuras 42d e 43d são apresentados os resultados das imagens binárias após a aplicação das operações morfológicas de erosão, fechamento e abertura. No caso 1, Figura 42d, um *blob* de maior área é encontrado. Então, desenha-se um retângulo de menor área possível em torno do *blob*, a Figura 42e demonstra o resultado dessa operação. No caso 2, Figura

Figura 42 – Segmentação do componente - caso 1.

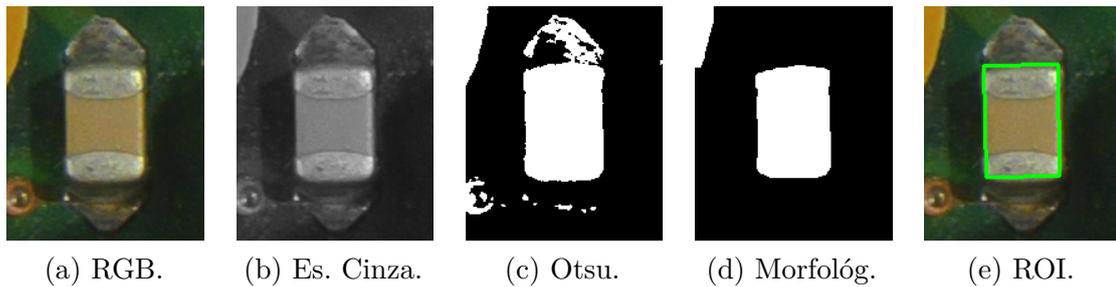
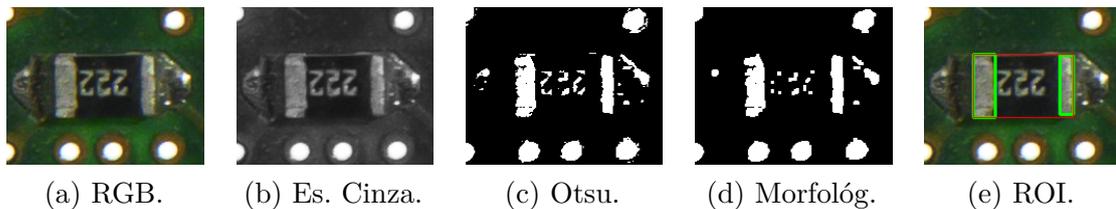


Figura 43 – Segmentação do componente - caso 2.



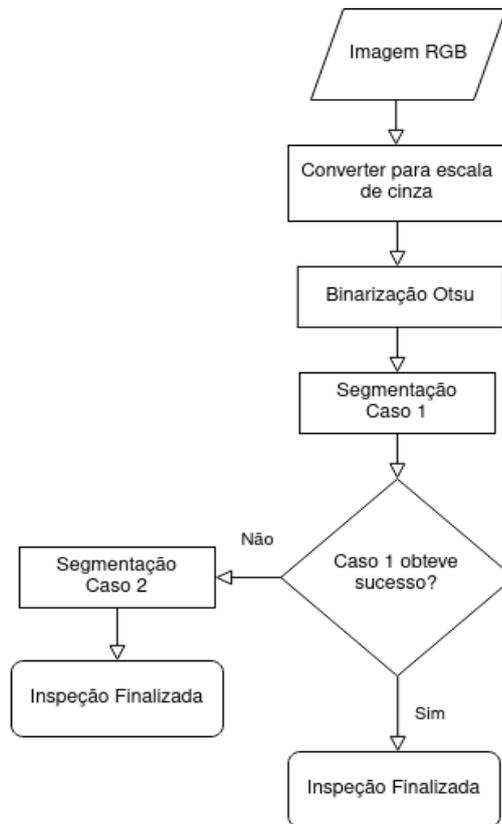
43d, é escolhido os dois *blobs* de maior área detectados. Na Figura 43e é demonstrado em verde os dois retângulos desenhados em torno dos *blobs* escolhidos, e em vermelho o retângulo resultante da união dos dois retângulos precedentes.

Na Figura 44 é apresentado o fluxograma da inspeção da posição. A Figura 44a apresenta o fluxograma principal, onde primeiramente tenta-se realizar a segmentação de componentes do caso 1. O fluxograma da segmentação de caso 1 é apresentado na Figura 44b. Se após a aplicação das operações morfológicas não é encontrado nenhum contorno com área maior que 10% da área da imagem do componente analisado, executa-se então a segmentação de caso 2. O fluxograma da segmentação para componentes de caso 2 é apresentado na Figura 44c, onde os 2 contornos de maior área encontrados são unidos para formar a ROI do componente. Foram utilizados *kernels* retangulares em todas as operações morfológicas.

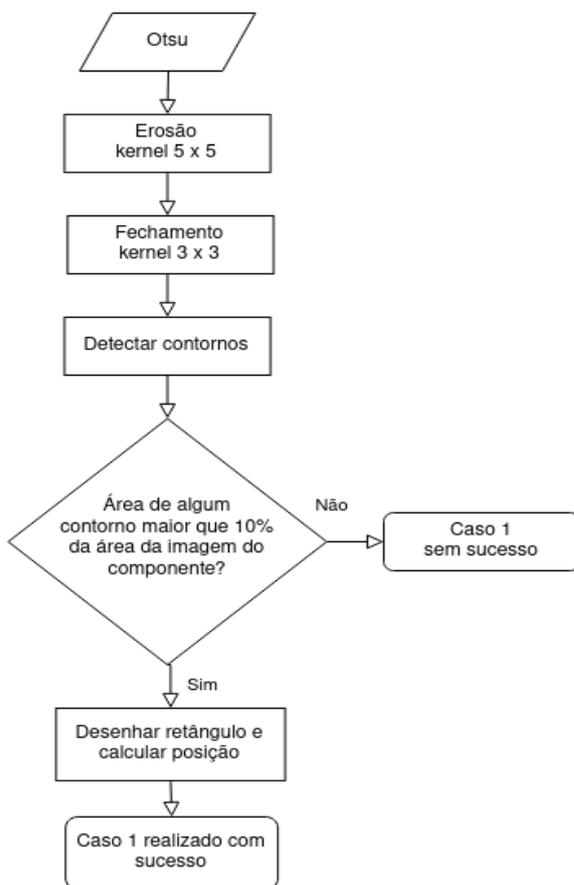
4.7.1 Deslocamento angular do componente

Para testar a inspeção do deslocamento angular dos componentes, foram simuladas 40 imagens defeituosas de capacitores cerâmicos e resistores. Nessas imagens, rotacionou-se o componente de um ângulo de 20 graus. Na Figura 45 é demonstrado um exemplo de defeito de componente rotacionado simulado. A Figura 45a demonstra o componente original, considerado correto. Na Figura 45b é apresentado a simulação do defeito, onde rotacionou-se o componente em um ângulo de 20 graus. Para encontrar o ângulo, realiza-se os passos de pré-processamento discutidos anteriormente, e desenha-se o retângulo de menor área possível ao redor do *blob* encontrado. Com as coordenadas do retângulo, torna-se trivial encontrar o ângulo de rotação. Os resultados obtidos das medições do

Figura 44 – Fluxograma de inspeção da posição.



(a) Processo principal.



(b) Segmentação caso 1.



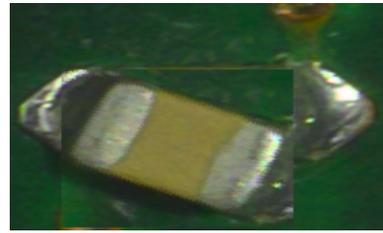
(c) Segmentação caso 2.

deslocamento angular dos componentes são apresentados na Seção 5.2.

Figura 45 – Simulação de componente rotacionado.



(a) Componente original.



(b) Componente rotacionado.

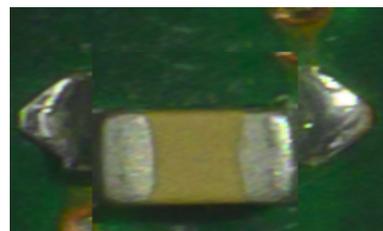
4.7.2 Deslocamento linear do componente

Da mesma forma, para testar a inspeção do deslocamento linear dos componentes, foram simuladas 40 imagens defeituosas de capacitores cerâmicos e resistores. Nessas imagens, deslocou-se a posição do componente em 20 pixels. Na Figura 46 é demonstrado um exemplo de simulação de componente deslocado. A Figura 46a demonstra o componente original, considerado correto. Na Figura 46b é apresentado a simulação do defeito, onde deslocou-se o componente em 20 pixels. Para encontrar o deslocamento, realiza-se os passos de pré-processamento discutidos anteriormente, e desenha-se o retângulo de menor área possível ao redor do *blob* encontrado. Calcula-se o centroide do componente assim como o centroide da imagem analisada. Com os centroides obtidos, calcula-se a distância da reta que une os dois centroides. Os resultados obtidos das medições do deslocamento linear dos componentes são apresentados na Seção 5.2.

Figura 46 – Simulação de componente linearmente deslocado.



(a) Componente original.



(b) Componente deslocado.

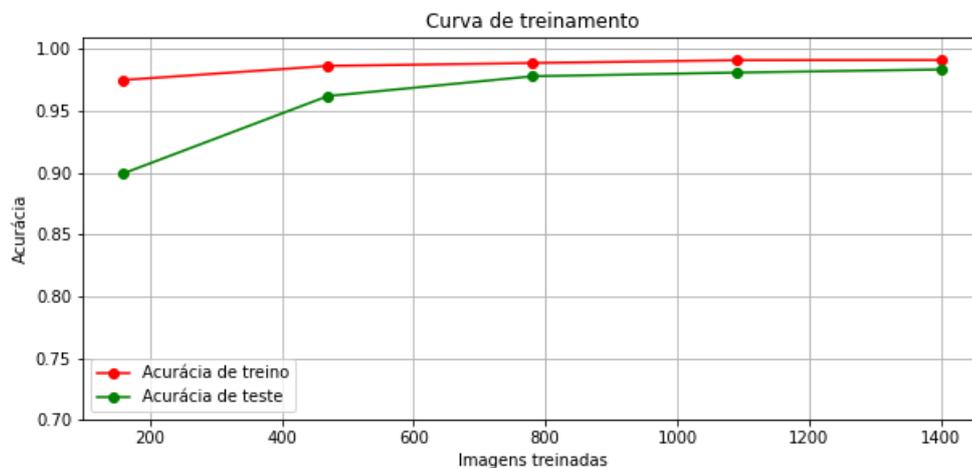
5 Resultados e Discussões

Neste capítulo são apresentados e discutidos os resultados obtidos do treinamento e dos testes de validação do modelo de classificação, assim como os resultados da inspeção da posição.

5.1 Classificador

A curva de acurácia por número de imagens treinadas e testadas do modelo de classificação por SVM linear é apresentado na Figura 47. Observa-se que a acurácia do modelo cresce com a quantidade de imagens treinadas. Esse crescimento é evidente até 900 imagens testadas, após esse valor a acurácia do modelo se mantém constante. O banco de imagens consiste em 2000 imagens no total, sendo 400 para cada classe. A base de dados é dividida em 30% para teste e 70% para treinamento. A acurácia de treinamento final obtida foi de 0,991 e a acurácia de teste de de 0,987.

Figura 47 – Curva de treinamento do modelo de classificação.



A curva do tempo em segundos gasto para treinar as SVMs é apresentada na Figura 48. Esse treinamento foi realizado em um computador pessoal, com características descritas em Seção 4.3. Nota-se que o modelo é treinado muito rapidamente, levando apenas 1,5 segundos para completar a tarefa. Entretanto, observa-se que nesse tempo não está contido a etapa de extração de característica das imagens. Na Figura 49 é apresentado a curva de acurácia por tempo de treinamento do modelo.

Figura 48 – Curva de escalabilidade do modelo de classificação.

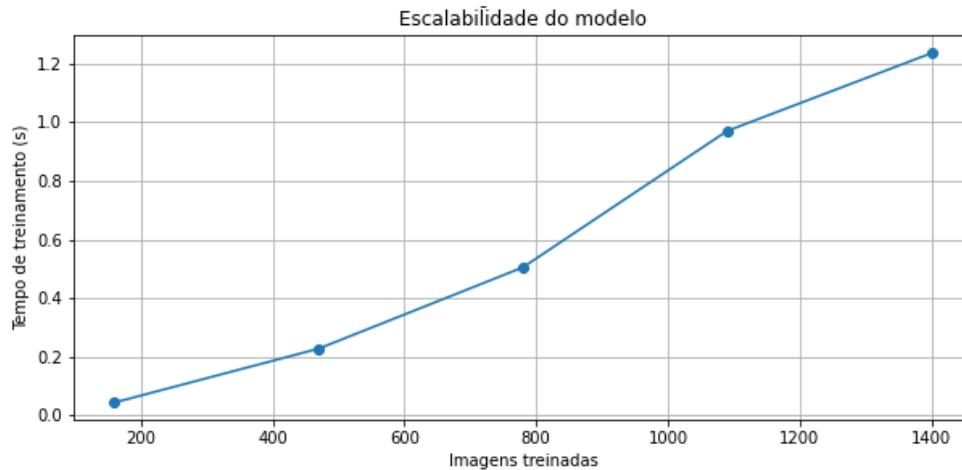
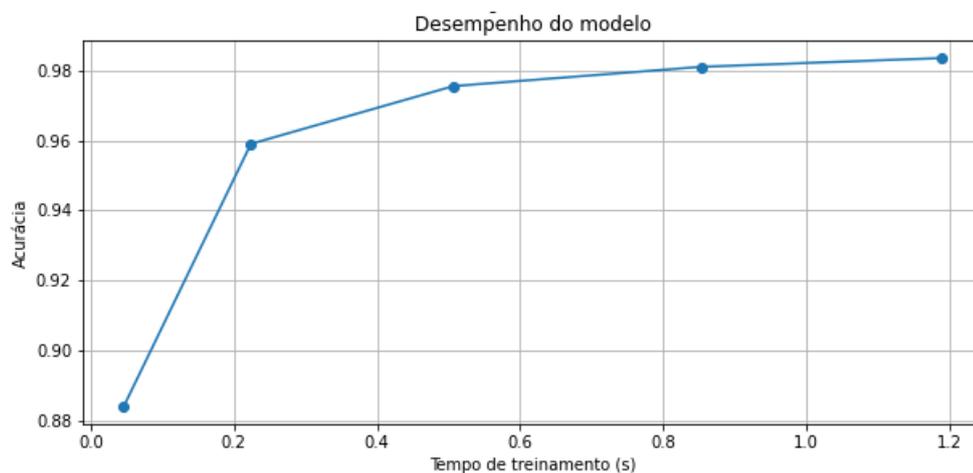


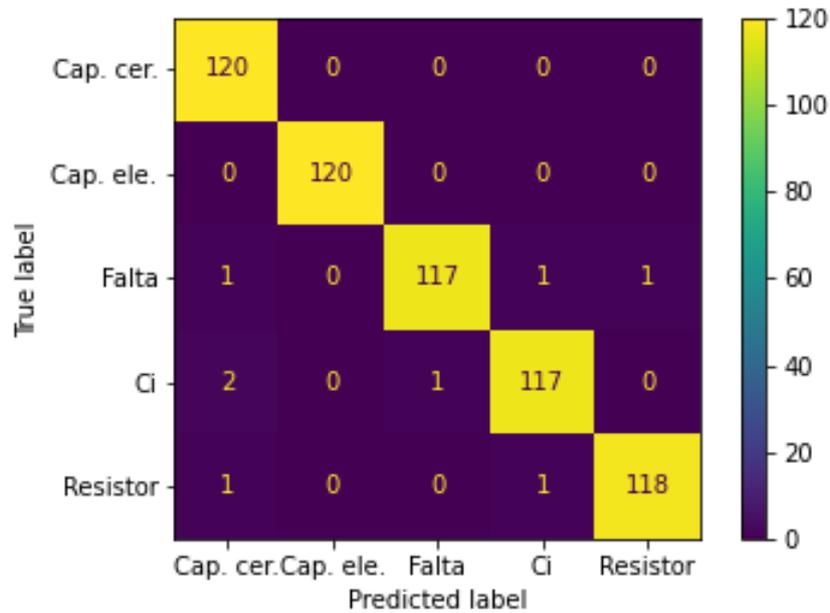
Figura 49 – Curva do desempenho do modelo de classificação.



Na Figura 50 é apresentado a matriz de confusão do classificador. Nota-se que o modelo classificou corretamente todos os 120 capacitores cerâmicos SMD e os 120 capacitores eletrolíticos. Entretanto, a classe de capacitores cerâmicos SMD obteve 5 falsos positivos. As classes com mais erros são a ausência de componentes e os circuitos integrado, com 3 erros e 117 acertos. Através da matriz de confusão, pode-se calcular a acurácia do sistema pela Equação 5.1, onde VP representa os verdadeiros positivos, e VN verdadeiros negativos.

$$Acurácia = \frac{VP + VN}{Total} = \frac{120 + 120 + 117 + 117 + 118}{5 \times 120} = 0,987 \quad (5.1)$$

Figura 50 – Matriz de confusão do modelo de classificação.



Para cada classe do componente, calcula-se também a precisão, revocação e F1 Score, conforme Equações 5.2, 5.3 e 5.4, onde VP representa os verdadeiros positivos, VN verdadeiros negativos, FP falsos positivos e FN falsos negativos.

$$Precisão = \frac{VP}{VP + FP} \quad (5.2)$$

$$Revocação = \frac{VP}{VP + FN} \quad (5.3)$$

$$F1score = \frac{2 \times precisão \times revocação}{precisão + revocação} \quad (5.4)$$

Os resultados obtidos aplicando as definições de precisão, revocação e f1-score são apresentados na Tabela 2.

Tabela 2 – Métricas de avaliação do modelo.

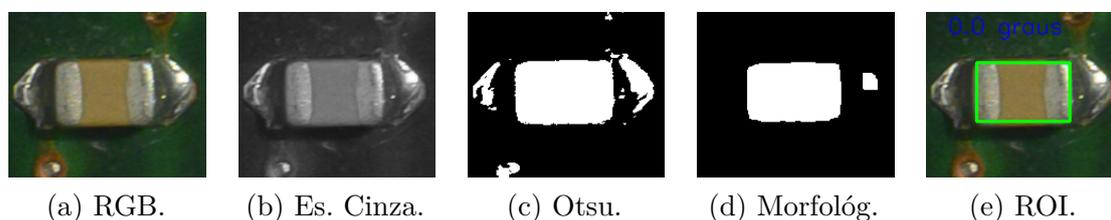
| Classe | Precisão | Revocação | F1-score |
|------------------------|----------|-----------|----------|
| Capacitor Cerâmico | 0,97 | 1,00 | 0,98 |
| Capacitor eletrolítico | 1,00 | 1,00 | 1,00 |
| Ausência de componente | 0,99 | 0,97 | 0,98 |
| Circuito Integrado | 0,98 | 0,97 | 0,98 |
| Resistor | 0,99 | 0,98 | 0,99 |

A acurácia do modelo indica a performance geral, levando em conta todos acertos e dividindo esse valor pelo total de imagens. Essa métrica representa o quanto o modelo classificou corretamente. A precisão, é uma métrica que recebe apenas os verdadeiros positivos no nominador. Essa métrica indica a capacidade de evitar falsos positivos. A revocação representa a frequência em que o modelo encontra as imagens de uma classe. Finalmente, a métrica f1-score calcula a média harmônica entre precisão e revocação representando a qualidade geral do modelo. Essa métrica é importante em conjunto de dados que possuem classes desproporcionais, como no caso das PCIs, onde o número de resistores em uma placa é muito maior que dos demais componentes.

5.2 Posicionamento

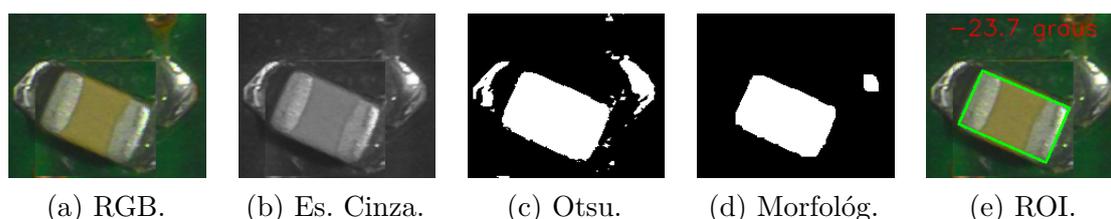
O algoritmo de medição do deslocamento angular foi aplicado em 80 imagens de capacitores cerâmicos e resistores, sendo 40 imagens originais da base de dados, e as demais 40 imagens com defeitos simulados, onde rotacionou-se os componentes em um ângulo de 20 graus. A Figura 51 apresenta a medição de um componente não rotacionado, onde encontrou-se um ângulo de 0 graus, como demonstrado na Figura 51e.

Figura 51 – Inspeção do componente não rotacionado.



Na Figura 52 é demonstrado a aplicação do algoritmo em um componente rotacionado. O resultado da medição é apresentado na Figura 52e, onde, nesse caso, mediu-se um ângulo de rotação de 23,7 graus.

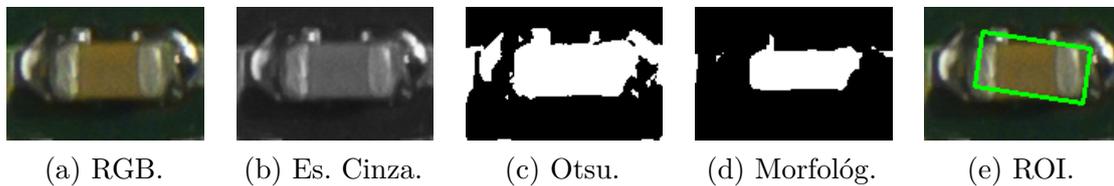
Figura 52 – Inspeção do componente rotacionado.



Na Tabela 3 é apresentado os resultados obtidos das medições nos componentes não rotacionados e rotacionados. Observa-se que algumas medições divergiram do valor esperado. Essa divergência deve-se incorreta segmentação do componente. A Figura 53

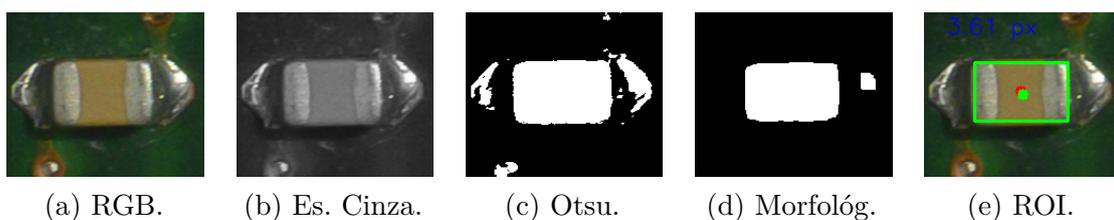
demonstra a segmentação do componente não rotacionado de $id=1$ da Tabela 3, onde mediu-se um ângulo de $8,58$. Na Figura 53a observa-se que há reflexão de luz na solda. Esse efeito causado pela iluminação faz com que na binarização Otsu, Figura 53c, apareça *blobs* e ruídos não desejados. Mesmo após as operações morfológicas, Figura 53d, observa-se um ruído na parte superior direita do *blob* encontrado. Esse ruído em alguns componentes causa a inexatidão de algumas medições. Esses ruídos podem ser mitigados capturando as imagens em um ambiente com maior controle da iluminação, que é o caso da máquina AOI HJ-760 apresentada na Seção 2.4. Entretanto, nota-se pela Tabela 3 que o sistema de inspeção já estaria funcional. Ao setar um valor máximo de rotação aceitável de 17 graus, por exemplo, todos os componentes rotacionados de 20 graus seriam detectados.

Figura 53 – Inspeção do componente $id=1$ da Tabela 3.



Igualmente, o algoritmo de medição do deslocamento linear foi aplicado em 80 imagens de capacitores cerâmicos e resistores, sendo 40 imagens originais da base de dados, e as demais 40 imagens com defeitos simulados, onde deslocou-se os componentes em 20 pixels. A Figura 54 apresenta a medição de um componente não deslocado, onde encontrou-se um deslocamento de $3,61$ pixels, como demonstrado na Figura 54e.

Figura 54 – Inspeção do componente não deslocado.



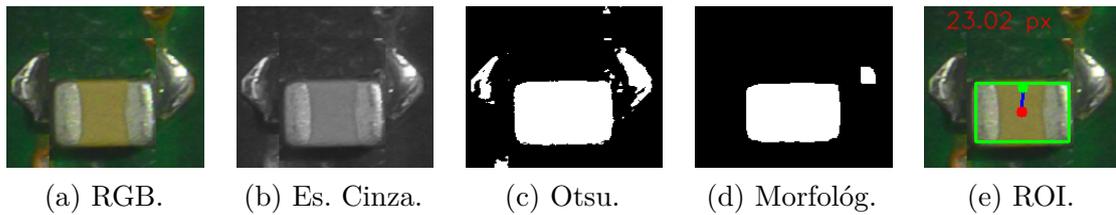
Na Figura 55 é demonstrado a aplicação do algoritmo em um componente linearmente deslocado. O resultado da medição é apresentado na Figura 55e, onde, nesse caso, mediu-se um deslocamento de $23,02$ pixels.

Na Tabela 4 é apresentado os resultados obtidos das medições do deslocamento linear nos componentes nos componentes. Observa-se que da mesma forma da medição do deslocamento angular, algumas medições divergiram do valor esperado devido à iluminação no sistema de captura do sistema. Entretanto, nota-se pela Tabela 4 que o sistema de inspeção já estaria funcional. Ao setar um valor máximo de deslocamento para 18 pixels, por exemplo, todos os componentes deslocados de 20 pixels seriam detectados.

Tabela 3 – Resultados de inspeção do deslocamento angular.

| Componente | Posição correta | Rotacionado 20 graus |
|---------------|-----------------|----------------------|
| Id | Ângulo medido | |
| 1 | 8,58 | 18,80 |
| 2 | 0,00 | 19,83 |
| 3 | 1,93 | 22,44 |
| 4 | 0,80 | 19,51 |
| 5 | 8,97 | 19,27 |
| 6 | 0,00 | 20,32 |
| 7 | 1,12 | 21,80 |
| 8 | 0,82 | 20,06 |
| 9 | 0,00 | 17,65 |
| 10 | 0,13 | 21,35 |
| 11 | 0,00 | 18,43 |
| 12 | 0,00 | 18,86 |
| 13 | 0,00 | 17,40 |
| 14 | 3,22 | 18,18 |
| 15 | 0,00 | 18,06 |
| 16 | 0,90 | 20,45 |
| 17 | 0,00 | 20,32 |
| 18 | 0,00 | 18,80 |
| 19 | 8,13 | 25,45 |
| 20 | 0,80 | 20,19 |
| 21 | 8,58 | 18,80 |
| 22 | 0,97 | 18,30 |
| 23 | 0,80 | 19,65 |
| 24 | 0,00 | 20,32 |
| 25 | 0,00 | 16,86 |
| 26 | 0,00 | 18,80 |
| 27 | 0,00 | 19,83 |
| 28 | 0,00 | 18,43 |
| 29 | 0,82 | 20,06 |
| 30 | 7,77 | 25,82 |
| 31 | 0,90 | 20,45 |
| 32 | 0,00 | 17,65 |
| 33 | 0,00 | 20,32 |
| 34 | 0,00 | 17,40 |
| 35 | 0,40 | 18,51 |
| 36 | 0,13 | 20,35 |
| 37 | 2,31 | 22,44 |
| 38 | 1,12 | 21,80 |
| 39 | 4,09 | 18,18 |
| 40 | 0,00 | 18,06 |
| Média | 1,58 | 19,73 |
| Desvio Padrão | 2,77 | 1,95 |

Figura 55 – Inspeção do componente deslocado.



Posteriormente, aplicou-se o algoritmo completo de inspeção (classificador + medição do posicionamento) nos 40 componentes corretamente posicionados, nas 40 imagens com os componentes deslocados de 20 pixels e também nas 40 imagens com os componentes rotacionados de 20 graus. A Tabela 5 apresenta os resultados do classificador para esse experimento. Observa-se na Tabela 5 que para os componentes corretamente posicionados, todos os componentes foram corretamente classificados e estão em conformidade com a referência. Já para os componentes defeituosos, observa-se que o processo de classificação classificou de forma errada diversos componentes. Esse resultado é esperado, pois o modelo foi treinado apenas com imagens de componentes bem posicionados. Para o funcionamento geral do sistema de inspeção de PCIs, essa classificação errada de componentes defeituosos é boa, pois o alerta de detecção de defeito aconteceria já na etapa de classificação, tornando o sistema de inspeção mais robusto.

Tabela 4 – Resultados de inspeção do deslocamento linear.

| Componente | Posição correta | Deslocado 20 px |
|---------------|----------------------------|-----------------|
| Id | Deslocamento linear medido | |
| 1 | 1,81 | 21,46 |
| 2 | 3,62 | 21,04 |
| 3 | 2,81 | 21,68 |
| 4 | 0,00 | 25,53 |
| 5 | 5,00 | 20,40 |
| 6 | 2,40 | 19,16 |
| 7 | 0,43 | 21,41 |
| 8 | 3,71 | 23,17 |
| 9 | 1,22 | 21,16 |
| 10 | 3,61 | 24,08 |
| 11 | 5,10 | 21,59 |
| 12 | 8,49 | 28,64 |
| 13 | 4,47 | 23,35 |
| 14 | 2,18 | 20,07 |
| 15 | 2,24 | 19,10 |
| 16 | 2,24 | 21,10 |
| 17 | 5,83 | 25,18 |
| 18 | 4,47 | 23,35 |
| 19 | 10,05 | 21,47 |
| 20 | 6,08 | 22,80 |
| 21 | 2,54 | 20,68 |
| 22 | 6,32 | 18,68 |
| 23 | 4,12 | 20,40 |
| 24 | 3,66 | 19,31 |
| 25 | 2,07 | 21,48 |
| 26 | 3,61 | 22,20 |
| 27 | 2,25 | 18,47 |
| 28 | 6,08 | 21,84 |
| 29 | 2,47 | 22,08 |
| 30 | 8,54 | 18,79 |
| 31 | 1,00 | 20,02 |
| 32 | 1,94 | 19,48 |
| 33 | 2,24 | 22,02 |
| 34 | 3,61 | 22,20 |
| 35 | 3,00 | 26,41 |
| 36 | 3,16 | 22,20 |
| 37 | 2,21 | 20,00 |
| 38 | 1,22 | 21,61 |
| 39 | 2,00 | 22,00 |
| 40 | 3,61 | 18,11 |
| Média | 3,54 | 21,59 |
| Desvio Padrão | 2.20 | 2.21 |

Tabela 5 – Resultados de classificação nos componentes deslocados e rotacionados.

| Componente | Referência | Posicionamento correto | Deslocado 20 px | Rotacionado 20 graus |
|---------------|--------------------|------------------------|--------------------|------------------------|
| 1 | capacitor cerâmico | capacitor cerâmico | capacitor cerâmico | capacitor cerâmico |
| 2 | capacitor cerâmico | capacitor cerâmico | capacitor cerâmico | capacitor eletrolítico |
| 3 | capacitor cerâmico | capacitor cerâmico | capacitor cerâmico | capacitor eletrolítico |
| 4 | capacitor cerâmico | capacitor cerâmico | componente ausente | capacitor eletrolítico |
| 5 | capacitor cerâmico | capacitor cerâmico | capacitor cerâmico | capacitor cerâmico |
| 6 | capacitor cerâmico | capacitor cerâmico | capacitor cerâmico | capacitor cerâmico |
| 7 | resistor | resistor | resistor | resistor |
| 8 | capacitor cerâmico | capacitor cerâmico | resistor | capacitor cerâmico |
| 9 | capacitor cerâmico | capacitor cerâmico | resistor | resistor |
| 10 | capacitor cerâmico | capacitor cerâmico | resistor | resistor |
| 11 | capacitor cerâmico | capacitor cerâmico | capacitor cerâmico | resistor |
| 12 | capacitor cerâmico | capacitor cerâmico | resistor | capacitor cerâmico |
| 13 | resistor | resistor | ci | resistor |
| 14 | resistor | resistor | resistor | resistor |
| 15 | resistor | resistor | resistor | resistor |
| 16 | resistor | resistor | ci | resistor |
| 17 | resistor | resistor | resistor | resistor |
| 18 | resistor | resistor | ci | capacitor cerâmico |
| 19 | resistor | resistor | resistor | resistor |
| 20 | resistor | resistor | resistor | resistor |
| 21 | capacitor cerâmico | capacitor cerâmico | capacitor cerâmico | capacitor cerâmico |
| 22 | capacitor cerâmico | capacitor cerâmico | resistor | capacitor cerâmico |
| 23 | resistor | resistor | resistor | resistor |
| 24 | capacitor cerâmico | capacitor cerâmico | capacitor cerâmico | capacitor cerâmico |
| 25 | capacitor cerâmico | capacitor cerâmico | resistor | capacitor eletrolítico |
| 26 | resistor | resistor | ci | capacitor cerâmico |
| 27 | capacitor cerâmico | capacitor cerâmico | capacitor cerâmico | capacitor eletrolítico |
| 28 | capacitor cerâmico | capacitor cerâmico | resistor | resistor |
| 29 | capacitor cerâmico | capacitor cerâmico | resistor | resistor |
| 30 | resistor | resistor | resistor | resistor |
| 31 | resistor | resistor | ci | resistor |
| 32 | capacitor cerâmico | capacitor cerâmico | capacitor cerâmico | capacitor cerâmico |
| 33 | resistor | resistor | resistor | resistor |
| 34 | resistor | resistor | ci | resistor |
| 35 | capacitor cerâmico | capacitor cerâmico | capacitor cerâmico | capacitor eletrolítico |
| 36 | capacitor cerâmico | capacitor cerâmico | resistor | capacitor cerâmico |
| 37 | capacitor cerâmico | capacitor cerâmico | capacitor cerâmico | capacitor eletrolítico |
| 38 | resistor | resistor | resistor | resistor |
| 39 | resistor | resistor | resistor | resistor |
| 40 | resistor | resistor | resistor | resistor |
| Classificação | Correta | 40 | 24 | 26 |
| | Incorreta | 0 | 16 | 14 |

6 Considerações Finais

A detecção de defeitos nos produtos é uma tarefa de extrema importância no processo industrial, especialmente a inspeção de qualidade visual. Em muitas indústrias no Brasil, esse processo é realizado manualmente por inspetores humanos. Entretanto, essa inspeção não é muito confiável e está sujeita a falhas devido à fadiga e perda de atenção do inspetor. Por essa razão, máquinas de inspeção automática visual estão cada vez mais sendo empregadas nos processos industriais. Logo, neste trabalho desenvolvido como Projeto de Diplomação do curso de Engenharia Elétrica foi proposto um sistema de inspeção de componentes de placas de circuito impresso.

O classificador de componentes desenvolvido através das características de forma e textura, obtidas pelo descritor HOG, e das características de cor, obtidas pelo histograma da imagem no espaço de cores HSV, obteve uma acurácia de 98,7% nas imagens testadas. Esse modelo foi desenvolvido utilizando a técnica de aprendizado de máquina SVM linear. O modelo implementado é capaz de classificar componentes em 5 classes: resistor SMD, capacitor cerâmico SMD, capacitor eletrolítico, circuito integrado e ausência de componente. Essa etapa de classificação é importante para detectar falhas comuns no processo de fabricação de PCIs, como a falta de um componente que não foi soldado na placa, ou quando é fixado à placa um componente errado que pertence a outra classe. A comparação destes resultados com os resultados dos trabalhos relacionados se torna difícil pois os algoritmos foram testados em imagens capturadas em um *setup* diferente, tanto em termos da câmera quanto em termos de iluminação.

Posteriormente, foi implementado uma etapa de inspeção do posicionamento de componentes pertencentes às classes capacitor cerâmico e resistor. Nessa etapa, algumas medições do deslocamento do componente foram inexatas devido à dependência de uma correta iluminação para que binarização e segmentação do componente funcione corretamente. Contudo, esse processo pode ser melhorado através de um sistema de aquisição de imagens nos moldes da máquina AOI HJ-760 apresentada na Seção 2.4, onde a aquisição das imagens é realizada no interior da máquina em um ambiente com iluminação bem controlada.

Para futuros trabalhos, sugere-se aumentar o número de classes do classificador, adicionando imagens de outros componentes como transistores, indutores e diodos. Possivelmente, seja necessário adicionar mais algum descritor de característica ao se adicionar novas classes. Além disso, também seria interessante expandir a análise de posicionamento, de forma que abranja todas as classes de componentes. Outra perspectiva, que agregaria muito ao sistema desenvolvido, seria implementar algoritmos para a inspeção das juntas

de solda.

Referências

- ACCIANI, G.; BRUNETTI, G.; FORNARELLI, G. Application of neural networks in optical inspection and classification of solder joints in surface mount technology. *IEEE Transactions on Industrial Informatics*, v. 2, p. 200–209, 08 2006. Disponível em: <<https://ieeexplore.ieee.org/document/1668079>>. Citado na página 33.
- ALLPOINT. *Máquinas e Serviços para SMT / PTH / TESTES - AllPoint*. 2020. Disponível em: <<https://www.allpoint.com.br/>>. Acesso em: 19/05/2020. Citado 2 vezes nas páginas 19 e 20.
- ARAUJO, K. *Wave Soldering Defects - Through-Hole and Surface Mount PCB Assembly*. 2020. Disponível em: <<http://www.epectec.com/pcb/wave-soldering-defects/>>. Acesso em: 05/01/2020. Citado na página 16.
- CARCAGNÌ, P. et al. Facial expression recognition and histograms of oriented gradients: a comprehensive study. *SpringerPlus*, v. 4, 10 2015. Citado na página 28.
- CERVANTES, J. et al. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, v. 408, p. 189–215, 09 2020. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0925231220307153>>. Citado 3 vezes nas páginas 29, 30 e 31.
- COOMBS, C. F.; HOLDEN, H. T. *Printed circuits handbook*. [S.l.]: New York McGraw-Hill Education, 2016. Citado na página 14.
- DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005. Citado na página 26.
- DORO, M. M. *Sistemática para Implantação da Garantia da Qualidade em Empresas Montadoras de Placas de Circuito Impresso*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, 2004. Citado na página 15.
- FACELI, K. et al. *Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina*. [S.l.: s.n.], 2011. ISBN 978-85-216-1880-5. Citado na página 29.
- FILHO, O. M.; NETO, H. V. *Processamento digital de imagens*. [S.l.]: Rio De Janeiro, Brasport, 1999. Citado na página 23.
- FUSETTN. *Printed Circuit Board and Surface Mount Design*. 1999. Disponível em: <<http://www.fuse-network.com/fuse/training/pcb/pcb.pdf>>. Acesso em: 01/05/2020. Citado na página 14.
- GONZALEZ, R. C.; WOODS, R. E. *Processamento de imagens digitais*. São Paulo: Pearson Education, 2009. ISBN 978-85-8143-586-2. Citado 5 vezes nas páginas 21, 22, 24, 26 e 27.
- HAN, C.; MAZOUZ, K.; SARAVANAN, N. A printed circuit board inspection system using artificial neural network. *1993 (25th) Southeastern Symposium on System Theory*,

1993. Disponível em: <<https://ieeexplore.ieee.org/document/522778>>. Citado na página 32.
- HUANG, W.; WEI, W. A pcb dataset for defects detection and classification. v. 14, 2018. Disponível em: <<https://arxiv.org/pdf/1901.08204.pdf>>. Citado na página 38.
- IBRAHIM, Z.; AL-ATTAS, S.; ASPAR, Z. *Analysis of the wavelet-based image difference algorithm for PCB inspection*. 2002. 2108–2113 vol.4 p. Disponível em: <<https://ieeexplore.ieee.org/document/1195721>>. Citado na página 32.
- KECMAN, V. *Support Vector Machines – An Introduction*. unknown, 2005. Disponível em: <https://www.researchgate.net/publication/226743605_Support_Vector_Machines_-_An_Introduction>. Citado na página 29.
- KUNTE, B. O. *Automatic Optical Inspection of Printed Circuit Boards*. Tese (Doutorado) — Czech Technical University in Prague, 2018. Citado na página 18.
- KUO, C.-W. et al. Data-efficient graph embedding learning for pcb component detection. In: IEEE. *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. [S.l.], 2019. Citado na página 38.
- LEI, T. et al. Vector morphological operators in hsv color space. *Science China Information Sciences*, v. 56, p. 1–12, 01 2012. Citado na página 23.
- LETA, F.; FELICIANO, F.; MARTINS, F. *Computer Vision System for Printed Circuit Board Inspection*. 2008. Disponível em: <<https://pdfs.semanticscholar.org/9b3e/d0eb98a8fd77b3835e9b3bc50ffcfedaa9f.pdf>>. Citado na página 32.
- LIN, S.-c.; SU, C.-h. *A Visual Inspection System for Surface Mounted Devices on Printed Circuit Board*. 2006. 1–4 p. Disponível em: <<https://ieeexplore.ieee.org/document/4017796>>. Citado na página 32.
- LORENA, A. C.; CARVALHO, A. C. P. L. F. D. Uma introdução às support vector machines. *Revista de Informática Teórica e Aplicada*, v. 14, p. 43–67, 12 2007. Citado 2 vezes nas páginas 29 e 30.
- LU, H. et al. *FICS-PCB: A Multi-Modal Image Dataset for Automated Printed Circuit Board Visual Inspection*. 2020. Disponível em: <<https://eprint.iacr.org/2020/366.pdf>>. Citado 4 vezes nas páginas 38, 39, 40 e 41.
- MAHALINGAM, G.; GAY, K. M.; RICANEK, K. *PCB-METAL: A PCB Image Dataset for Advanced Computer Vision Machine Learning Component Analysis*. 2019. 1–5 p. Disponível em: <<https://ieeexplore.ieee.org/document/8757928?denied=>>. Citado na página 38.
- MARKETSANDMARKETS. *Computer Vision Market | Global Market Forecast to 2023*. 2018. Disponível em: <<https://www.marketsandmarkets.com/Market-Reports/computer-vision-market-186494767.html>>. Acesso em: 01/09/2019. Citado na página 12.
- MARTINS, S. *Introdução ao Processamento Digital de Imagem*. 2020. Instituto de Computação UNICAMP. Disponível em: <https://www.ic.unicamp.br/~helio/disciplinas/MC920/aula_cores.pdf>. Acesso em: 01/09/2020. Citado na página 22.

- MDIC. *Setor Automotivo*. 2017. Disponível em: <<http://www.mdic.gov.br/index.php/competitividade-industrial/setor-automotivo>>. Acesso em: 16/12/2019. Citado na página 12.
- MELLO, A. R. d.; STEMMER, M. R. *Inspecting surface mounted devices using k nearest neighbor and Multilayer Perceptron*. 2015. 950–955 p. Disponível em: <<https://ieeexplore.ieee.org/document/7281599>>. Citado na página 33.
- MELO, D. F. F. de. *Desenvolvimento de Máquina de Inspeção Automática de Placas de Circuito Impresso para Produção em Pequenas Séries*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, 2013. Citado 2 vezes nas páginas 19 e 21.
- MOGANTI, M. et al. Automatic pcb inspection algorithms: A survey. *Computer Vision and Image Understanding*, v. 63, p. 287–313, 03 1996. Citado 4 vezes nas páginas 15, 16, 17 e 18.
- OPENCV. *OpenCV: Morphological Transformations*. 2020. Disponível em: <https://docs.opencv.org/4.4.0/d9/d61/tutorial_py_morphological_ops.html>. Acesso em: 01/09/2020. Citado 2 vezes nas páginas 25 e 26.
- OTSU, N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 9, p. 62–66, 01 1979. Citado na página 23.
- PEALLAT, J.-M. et al. *What is really inside your AOI?* 2018. Disponível em: <<http://amtest.bg/press/AOI/What-is-really-inside-your-AOI.pdf>>. Acesso em: 18/05/2020. Citado na página 17.
- PEDRINI, H. *Introdução ao Processamento Digital de Imagem MC920 / MO443*. 2020. Instituto de Computação UNICAMP. Disponível em: <https://www.ic.unicamp.br/~helio/disciplinas/MC920/aula_cores.pdf>. Acesso em: 01/09/2020. Citado na página 22.
- PRAMERDORFER, C.; KAMPEL, M. *A Dataset for Computer-Vision-Based PCB Analysis*. 2015. Disponível em: <<http://www.mva-org.jp/Proceedings/2015USB/papers/10-38.pdf>>. Citado na página 38.
- PUTERA, I. et al. *Printed Circuit Board Defect Detection Using Mathematical Morphology and MATLAB Image Processing Tools*. 2010. Disponível em: <http://www.pgembeddedsystems.com/securelogin/upload/project/IEEE/2/PGM_32/PGM_32.pdf>. Citado na página 32.
- SANTOYO, J. et al. *PCB Inspection Using Image Processing and Wavelet Transform*. unknown, 2007. Disponível em: <https://www.researchgate.net/publication/220887488_PCB_Inspection_Using_Image_Processing_and_Wavelet_Transform>. Citado na página 32.
- SHALEV-SHWARTS, S.; BEN-DAVID, S. *Understanding Machine Learning From Theory to Algorithms*. [S.l.: s.n.], 2014. ISBN 978-1-107-05713-5. Citado na página 29.
- STEINWART, I. *Support vector machines*. [S.l.]: Springer, 2014. Citado na página 29.
- SZYMANSKI, C. *Desenvolvimento de Técnicas de Processamento Digital de Imagens para Inspeção de Placas de Circuito Impresso produzidas em pequenas séries*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, 2014. Citado na página 17.

TANG, S. et al. Online pcb defect detector on a new pcb defect dataset. *arXiv:1902.06197 [cs]*, 02 2019. Disponível em: <<https://arxiv.org/abs/1902.06197>>. Citado 2 vezes nas páginas 32 e 38.

TATIBANA, M.; LOTUFO, R. d. A. Novel automatic pcb inspection technique based on connectivity. *Proceedings X Brazilian Symposium on Computer Graphics and Image Processing*, 1997. Disponível em: <<https://ieeexplore.ieee.org/document/625174>>. Citado na página 32.

VAPNIK, V. N. *The nature of statistical learning theory*. [S.l.]: Springer, Cop, 1995. Citado na página 29.

WALT, S. van der et al. scikit-image: image processing in python. *PeerJ*, v. 2, p. e453, 06 2014. Citado 2 vezes nas páginas 25 e 26.

WU, H. et al. *Automated visual inspection of surface mounted chip components*. 2010. 1789–1794 p. Disponível em: <<https://ieeexplore.ieee.org/abstract/document/5588029>>. Citado na página 33.

YOUN, S.; LEE, Y.; PARK, T. *Automatic classification of SMD packages using neural network*. 2014. 790–795 p. Disponível em: <<https://ieeexplore.ieee.org/document/7028139>>. Citado na página 33.

ZENZO, S. D. A note on the gradient of a multi-image. *Computer Vision, Graphics, and Image Processing*, v. 33, p. 116–125, 01 1986. Citado na página 27.