

# Análise Comparativa de Redes Leves e Arquiteturas de Hardware Embarcado para Inspeção Automatizada de PCBs

Fabricio Roskowski  
Universidade do Vale do Itajaí, Brasil  
fabricio.8614571@edu.univali.br

Felipe Viel  
Universidade do Vale do Itajaí, Brasil  
viel@univali.br

Miguel Bertozzin  
Universidade do Vale do Itajaí, Brasil  
bertozzin@edu.univali.br

Douglas Rossi de Melo  
Universidade do Vale do Itajaí, Brasil  
drm@univali.br

## Abstract

This review analyzes the evolution of automatic PCB inspection techniques, highlighting the shift from classical image-processing methods to Deep Learning-based solutions. The study shows that lightweight architectures equipped with attention mechanisms significantly improve micro-defect detection while preserving the performance required for industrial environments. The hardware analysis indicates that reconfigurable platforms such as FPGAs outperform CPUs and embedded GPUs by delivering deterministic, real-time inference with superior efficiency. Finally, the work points to promising future directions, such as self-supervised learning, explainable AI, and synthetic data generation, that are expected to drive the next generation of inspection systems toward accurate “Zero-Defect” manufacturing.

## Keywords

PCB Defect Detection, Edge Computing, Automated Optical Inspection.

## 1 Introdução

Placas de circuito impresso, conhecidas internacionalmente como *printed circuit boards* (PCBs), são amplamente utilizadas para interconectar componentes eletrônicos em aplicações que vão de eletrodomésticos a sistemas aeroespaciais. Como são constituídas por materiais laminados, tipicamente de fibra de vidro e resina epóxi, qualquer imperfeição no processo de fabricação pode resultar em falhas críticas no produto final [1]. A crescente demanda do setor, aliada à tendência de miniaturização e ao aumento da densidade de componentes, tem ampliado significativamente a complexidade na identificação de defeitos ao longo das linhas de produção, que passam a exigir processos cada vez mais rápidos e eficientes [2].

Relatos de fabricantes indicam que, em linhas de produção de grande escala, mesmo taxas de defeito da ordem de 0,01% podem gerar perdas relevantes: em um lote de 10 000 PCBs, por exemplo, esse índice corresponde a 100 placas defeituosas e pode resultar em cerca de US\$ 5 000 de prejuízo direto em custos de fabricação, sem considerar retrabalho, sucata e impacto na reputação da empresa [3].

Historicamente, a inspeção de PCBs era realizada predominantemente manualmente, com base no julgamento visual do operador. Esse método apresenta caráter subjetivo, sujeito a variações entre avaliadores, além de ser lento e altamente suscetível a falhas, especialmente em períodos prolongados de trabalho, quando o cansaço compromete a acurácia da análise [4].

Para superar essas limitações, foram introduzidos sistemas de inspeção óptica automatizada, baseados em técnicas clássicas de processamento de imagens. Esses métodos, como a subtração de imagens, não exigiam grande capacidade computacional e podiam ser implementados de forma eficiente em hardware, tornando-se soluções viáveis [5].

Com o avanço tecnológico e o surgimento das redes neurais convolucionais, esse cenário passou por uma mudança de paradigma. Modelos passaram a extrair automaticamente características complexas das imagens, alcançando desempenho superior ao de algoritmos baseados em regras para identificação e classificação [6]. Na literatura, observa-se que a evolução tecnológica dos métodos de detecção de objetos consolidou-se em duas categorias principais. A primeira abrange os detectores de dois estágios, como o Faster R-CNN, que priorizam a precisão na identificação de defeitos. A segunda reúne os detectores de estágio único, exemplificados pelo YOLO (You Only Look Once), que buscam um equilíbrio entre precisão e velocidade de inferência, tornando-os adequados para aplicações em tempo real [7].

Diante da diversidade de abordagens e da rápida evolução dos métodos de detecção de defeitos em PCBs, torna-se necessário organizar esta revisão de modo a contextualizar os fundamentos teóricos, os critérios metodológicos e as principais contribuições da literatura recente.

Este artigo está organizado em seis seções. A Seção 2 evidencia a evolução que passa do processamento de imagens clássico aos algoritmos contemporâneos de Deep Learning baseados em mecanismos de atenção. A Seção 3 descreve os procedimentos adotados na condução da revisão e os critérios de seleção dos trabalhos analisados. A Seção 4 reúne a análise do estado da arte. A seção 5 apresenta uma comparação entre métodos, plataformas e abordagens de aprendizado. Por fim, a Seção 6 reúne as conclusões do estudo e aponta direções para pesquisas futuras.

## 2 Fundamentação Teórica

Esta seção apresenta os fundamentos teóricos que sustentam as abordagens de inspeção automática de PCBs. Inicialmente, descrevem-se os métodos clássicos baseados em operações lógicas, na subtração de imagens e em técnicas morfológicas, destacando suas limitações operacionais em ambientes industriais. Em seguida, introduzimos os detectores de dois estágios, nos quais a detecção ocorre em duas fases distintas: uma primeira etapa dedicada à geração de propostas de regiões (Region Proposal Networks), seguida da classificação e do refinamento dos candidatos. Esse

paradigma, exemplificado pelo Faster R-CNN, prioriza a máxima precisão na localização e classificação de defeitos [8]. Avançamos, então, para os detectores de estágio único, que realizam detecção e classificação em uma única passagem pela rede, eliminando a etapa de geração de propostas e permitindo uma inferência substancialmente mais rápida. Entre esses, destaca-se a família YOLO (You Only Look Once), cuja filosofia consiste em transformar o problema de detecção em uma tarefa de regressão direta sobre caixas delimitadoras e probabilidades de classe [1]. A seção também contempla desafios estruturais do domínio, como a detecção de microdefeitos e a escassez de dados rotulados, bem como técnicas modernas envolvendo Few-Shot Learning, aprendizado semissupervisionado e métodos de detecção de anomalias. Por fim, apresentam-se os principais avanços relacionados à implementação em hardware, estabelecendo o pano de fundo necessário às análises comparativas e às discussões desenvolvidas nas seções subsequentes.

## 2.1 Inspeção de Placas de Circuito Impresso

Placas de circuito impresso (PCBs) são constituídas por trilhas condutoras e ilhas de solda gravadas sobre um substrato isolante, que são responsáveis pela interconexão elétrica entre os componentes eletrônicos. Quando ainda não possuem componentes montados, são denominadas *PCBs nuas*; após a inserção e soldagem de dispositivos discretos e circuitos integrados, passam a ser chamadas de PCBs montadas ou assemblies (PCBA) [9, 10]. A Figura 1 apresenta as principais etapas do processo de fabricação de PCBs, bem como os trabalhos associados a cada etapa.

Ao longo do processo de fabricação (impressão das trilhas, perfuração, metalização, montagem e soldagem), podem surgir diferentes categorias de defeitos:

- Defeitos fatais, que impedem o funcionamento do circuito, como curto-circuito (Figura 2d) entre trilhas, circuito aberto (Figura 2c) e furos ausentes (Figura 2a) [9];
- Defeitos potenciais, que não necessariamente causam falha imediata, mas reduzem a confiabilidade e a vida útil, por exemplo, mordidas de ratos (Figura 2b), esporas de cobre (Figura 2e) e cobre espúrio (Figura 2f), frequentemente associados a sobras ou falhas parciais na etapa de corrosão [10, 11];
- Defeitos de soldagem, comuns em PCBs montadas, incluindo ponte de solda, solda insuficiente ou excessiva e desalinhamento de componentes, frequentemente relacionados a falhas no processo de montagem [10, 12].

Para apoiar a pesquisa e a avaliação de algoritmos, alguns trabalhos disponibilizam conjuntos de dados com tipos padronizados de defeitos. O conjunto de dados DeepPCB, por exemplo, contempla seis classes típicas (missing hole, mouse bite, open circuit, short, spur and spurious copper), sintetizadas a partir de imagens de referência, o que reflete tanto a diversidade de defeitos quanto a limitação de dados reais disponíveis na indústria [11]. Outros estudos ampliam esse cenário com imagens de PCBs montadas e de defeitos de soldagem [10].

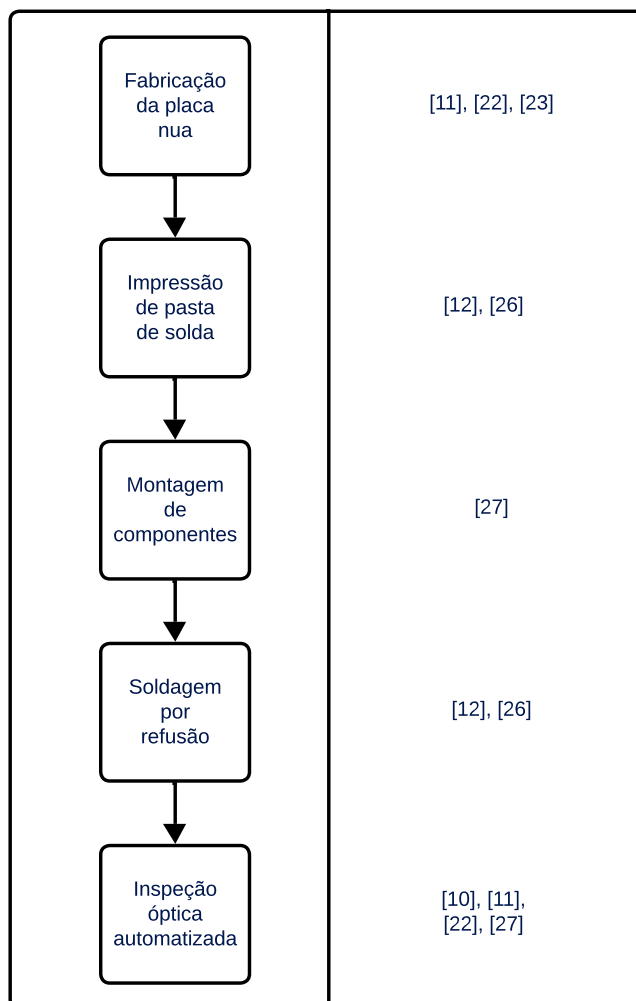
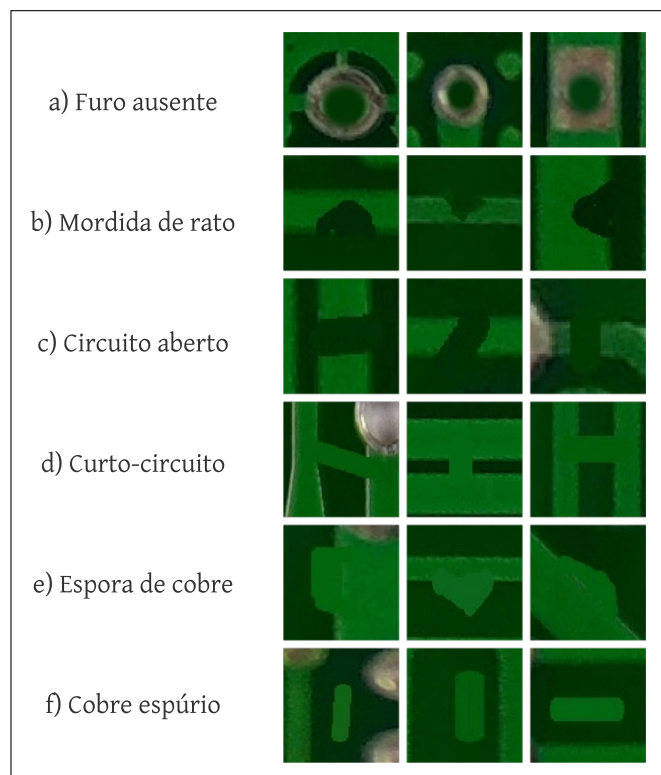


Figura 1: Etapas principais da fabricação em ordem de execução de placas de circuito impresso e respectivos trabalhos associados

A inspeção pode ser feita manualmente (lupa, microscópio), por meio de testes elétricos ou de técnicas ópticas automatizadas. Métodos manuais são simples, porém sujeitos à fadiga do inspetor, à variabilidade subjetiva e ao elevado custo de mão de obra. Já os testes elétricos e a inspeção a laser exigem equipamentos caros e não fornecem, isoladamente, informações detalhadas sobre a morfologia do defeito [9]. Nesse contexto, a *inspeção óptica automatizada* (IOA), baseada na aquisição de imagens e no processamento digital, tornou-se uma das abordagens mais utilizadas na indústria [10].

## 2.2 Abordagem de processamento de dados

Sistemas de IOA para PCBs podem ser divididos, de forma geral, em três grupos: métodos baseados em referência, métodos sem referência e métodos híbridos [9, 10]. A seguir, descrevem-se essas abordagens de forma resumida:



**Figura 2: Principais tipos de defeitos em placas de circuito impresso.**

- Métodos baseados em referência: uma imagem padrão de uma PCB sem defeitos é previamente obtida e registrada. A placa inspecionada é alinhada geometricamente à referência (por exemplo, por meio de pontos de interesse e transformações de registro) e, em seguida, comparada pixel a pixel, usualmente após a conversão para escala de cinza e a binarização adaptativa [5]. Operações como a subtração de imagens, a limiarização global ou local e a morfologia matemática são empregadas para realçar regiões discrepantes, que posteriormente são analisadas e classificadas conforme sua forma e conectividade [12]. Embora conceitualmente simples e eficientes em cenários controlados, esses métodos são sensíveis a desalinhamentos, variações de iluminação e diferenças de escala, o que exige que a etapa de aquisição de imagens seja controlada [9].
- Métodos sem referência: nesses métodos, busca-se verificar se trilhas, *pads* e furos estão de acordo com as regras de projeto, sem depender de uma imagem padrão. Estratégias típicas incluem segmentação cobre/não-cobre, análise de histogramas de cor e verificação de restrições geométricas (largura mínima de trilha, espaçamento, diâmetro de furo), o que reduz problemas de registro, mas pode apresentar limitações na detecção de defeitos mais complexos ou de grande extensão [9, 10].

- Métodos híbridos: combinam informações geométricas com comparação à referência, buscando maior robustez ao custo de maior complexidade computacional [10].

Com o aumento da resolução das imagens e da complexidade das placas, algoritmos puramente determinísticos tornaram-se insuficientes para lidar com ruídos de processo, contaminações e pequenas variações de fabricação. Nesse contexto, surgem as abordagens baseadas em *aprendizado profundo*, em especial as redes neurais convolucionais (CNNs). Trabalhos recentes mostram que modelos de detecção de objetos, como Faster R-CNN, SSD e a família YOLO, podem aprender características discriminativas de defeitos diretamente dos dados, superando algoritmos tradicionais em precisão e generalização [1, 10, 13–15].

Os detectores de duas etapas, representados por Faster R-CNN, geram inicialmente regiões de interesse (*region proposals*) e, em seguida, classificam e refinam as *bounding boxes* por meio de uma CNN. Em geral, oferecem maior acurácia, porém, com maior tempo de processamento, o que limita seu uso em inspeção totalmente em tempo real [8, 16]. Já os métodos de uma etapa, como SSD e YOLO, formulam a detecção como um problema de regressão direta de classes e coordenadas, realizando a previsão em uma propagação direta, o que proporciona altas taxas de quadros por segundo (QPS), com eventual perda de precisão em defeitos muito pequenos [1, 13–15].

Mais recentemente, redes baseadas em *Transformers* têm sido exploradas para classificação e detecção em PCBs, aproveitando mecanismos de self-attention para modelar dependências de longo alcance na imagem. O modelo Local Patch Vision Transformer (LPVit), por exemplo, aplica uma arquitetura inspirada em Vision Transformers a tarefas de classificação de placas e de detecção de defeitos, alcançando desempenho de estado da arte em conjuntos de dados como o DeepPCB [17]. Paralelamente, estudos discutem desafios típicos de dados industriais, como escassez de amostras, desbalanceamento entre classes e presença de defeitos do tipo objetos pequenos, motivando o uso de técnicas de aprendizado com poucas amostras (*few-shot learning*), semi-supervisionado e auto-supervisionado [18–21].

Dessa forma, a literatura recente converge para uma visão em que a inspeção de PCBs combina modelagem de defeitos (fatais, potenciais e de soldagem), aquisição de imagens controlada e técnicas de processamento de dados que vão de algoritmos estáticos a redes profundas otimizadas para diferentes plataformas de hardware (unidades centrais de processamento - CPUs, unidades de processamento gráfico - GPUs e arranjo de portas programáveis em campo - FPGAs), de modo a conciliar precisão, custo computacional e requisitos de tempo real dos ambientes industriais [9, 10, 22, 23].

### 3 Metodologia de pesquisa

Esta revisão foi conduzida com o objetivo de analisar o estado da arte na detecção de defeitos em placas de circuito impresso, com foco na transição tecnológica de algoritmos clássicos para redes neurais profundas e na sua implementação em hardware de borda. A seleção dos trabalhos seguiu critérios qualitativos e quantitativos para garantir uma cobertura abrangente da evolução histórica e das tendências futuras.

A compilação bibliográfica priorizou trabalhos publicados entre 2011 e 2025, abrangendo desde a consolidação dos métodos de processamento digital de imagens até as mais recentes arquiteturas de aprendizado profundo. As palavras-chave utilizadas para a filtragem incluíram combinações de termos como “PCB Defect Detection”, “Deep Learning” e “Automated Optical Inspection”.

As buscas bibliográficas foram realizadas em bases de dados científicas, como o IEEE Xplore, ScienceDirect, MDPI e SpingerLink. A fim de garantir a abrangência e a relevância dos trabalhos, as estratégias de busca foram estruturadas com base em combinações booleanas de palavras-chave relacionadas à inspeção automatizada de placas de circuito impresso. Entre as expressões de busca destacam-se: (“PCB” OR “Printed Circuit Board”) AND (“Defect Detection” OR “AOI”), (“FPGA” OR “Edge Computing”) AND (“PCB Inspection”), para localizar artigos completos, revisões e estudos experimentais relevantes. Foram selecionados artigos que apresentam métricas quantitativas de desempenho e detalhamento da arquitetura de hardware ou de rede. Os critérios de exclusão eliminaram estudos restritos a métodos de testes exclusivamente elétricos, sem abordagem de inspeção óptica, bem como trabalhos sem clara validação experimental.

## 4 Estado-da-Arte

Este capítulo apresenta uma revisão dos trabalhos na área de inspeção automática de PCBs, estruturando a análise em três eixos centrais: estratégias para lidar com a escassez de dados, métricas de desempenho, requisitos de operação em tempo real e implementações em hardware. Inicialmente, a Seção 4.1 discute abordagens desenvolvidas para mitigar a limitação de datasets rotulados, condição comum na indústria, na qual os defeitos reais ocorrem com baixa incidência.

Na Seção 4.2, a revisão concentra-se na avaliação de desempenho dos modelos, introduzindo os principais indicadores utilizados na literatura, incluindo mAP, IoU, QPS e latência, bem como métricas específicas para métodos não supervisionados, como Silhouette Score, Calinski–Harabasz, Davies–Bouldin e ARI. Essa subseção enfatiza o dilema entre precisão e eficiência computacional, discutindo como esse trade-off se manifesta em arquiteturas de dois estágios, em detectores de um único estágio, como a família YOLO, e em redes otimizadas com mecanismos de atenção.

Por fim, a Seção 4.3 apresenta um panorama das soluções de hardware empregadas na indústria. São comparadas implementações baseadas em CPUs, GPUs embarcadas, microcontroladores e, principalmente, FPGAs, utilizados tanto para processamento de imagem clássico quanto para aceleradores dedicados a redes profundas.

### 4.1 Técnicas para lidar com escassez de dados

Um desafio na aplicação industrial de técnicas de visão computacional na inspeção de PCBs é a escassez de dados rotulados, pois os defeitos reais tendem a ocorrer com baixa frequência. Isso limita o treinamento de modelos supervisionados tradicionais e tem motivado a adoção de abordagens que reduzam a dependência de grandes conjuntos anotados. Nesse contexto, métodos baseados em Few-Shot e em Transfer Learning têm se destacado [9].

Few-Shot Learning busca generalizar a partir de um número reduzido de exemplos, e modelos como o *Feature Pyramid Fusion Module*, que combinam aprimoramento de características e fusão multiescala, demonstram capacidade de detectar defeitos com alta precisão utilizando apenas 10 a 30 amostras, superando técnicas que normalmente requerem milhares de imagens [9, 18]. Paralelamente, estratégias baseadas em Transfer Learning e na geração de dados sintéticos permitem adaptar modelos previamente treinados em grandes bases de dados a tarefas específicas de inspeção de PCBs, reduzindo o esforço de anotação manual [11, 24].

Quando há disponibilidade de grandes volumes de dados não rotulados, abordagens semi-supervisionadas e auto-supervisionadas tornam-se alternativas eficazes. Métodos semi-supervisionados, como o PCB\_SS, empregam técnicas como FixMatch para produzir pseudo-rótulos consistentes e reforçar a estabilidade das predições, aumentando a robustez do modelo [20]. Em contrapartida, frameworks auto-supervisionados integram mecanismos de reconstrução, como Autoencoders, e algoritmos de detecção de outliers, como Isolation Forests, possibilitando a identificação de anomalias sem supervisão explícita e fornecendo interpretabilidade ao processo decisório [10, 21].

Para cenários em que defeitos desconhecidos precisam ser identificados, técnicas de detecção de anomalias não supervisionadas têm ganhado relevância. Abordagens baseadas em redes adversariais generativas (RAGs), como o método Tinynomaly, aprendem exclusivamente a distribuição de imagens normais e detectam anomalias a partir da incapacidade do gerador de reconstruir padrões defeituosos. Esse tipo de solução elimina a necessidade de amostras de defeito durante o treinamento, ampliando a aplicabilidade industrial em ambientes em que a ocorrência de falhas é extremamente rara [19, 21].

### 4.2 Métricas de desempenho e requisitos

A validação de modelos de visão computacional para a inspeção de PCBs em ambientes industriais exige uma análise entre a precisão de detecção e a latência de processamento. Esses dois fatores, embora igualmente essenciais, frequentemente entram em conflito, uma vez que o aumento da acurácia costuma implicar um custo computacional maior. Assim, o sucesso de um sistema de Inspeção Óptica Automática (IOA) não depende apenas da capacidade de identificar defeitos, mas também da viabilidade de operar em sincronia com a velocidade das linhas de produção, preservando o ritmo industrial sem comprometer a confiabilidade [6, 23].

No que diz respeito à precisão, a literatura adota amplamente a métrica de Precisão Média (mAP), calculada como a média da Average Precision (AP) de todas as classes de defeitos. O desempenho é diretamente influenciado pelo limiar de Intersection over Union (IoU) utilizado para definir a correspondência entre uma predição e a região real do defeito. As variantes mAP@0.5 e mAP@0.5:0.95 tornaram-se padrões de avaliação, sendo esta última mais rigorosa por considerar múltiplos limiares. Arquiteturas modernas, especialmente aquelas baseadas em mecanismos de atenção e fusão multiescala, como LW-YOLO e versões otimizadas do YOLOv7, já atingem índices superiores a 97–99% em mAP@0.5, demonstrando avanços significativos na detecção de microdefeitos e consolidando o estado da arte [1, 2, 10, 13].

A análise de desempenho em tempo real é determinante para o uso prático desses modelos. A taxa de quadros por segundo (QPS) e a latência de inferência determinam se o sistema consegue acompanhar o fluxo contínuo de produção. Detectores de dois estágios, como o Faster R-CNN, ainda que altamente precisos, apresentam limitações devido à sua arquitetura sequencial, resultando em velocidades inferiores a 20 QPS em hardware convencional, o que restringe sua aplicação em linhas de alta cadência. Em contraste, detectores de estágio único, representados pela família YOLO, foram projetados para maximizar a velocidade. Com otimizações recentes, como a adoção de backbones FasterNet e o uso de convoluções parciais (PConv), esses modelos alcançaram mais de 83 QPS, mantendo níveis competitivos de mAP e sugerindo que a redução de redundâncias computacionais pode ser um fator determinante para que o sistema atenda aos requisitos de latência em tempo real. O desempenho também é fortemente influenciado pela plataforma de hardware, e implementações em FPGAs modernas, como dispositivos Zynq UltraScale+ equipados com unidades de processamento dedicadas (DPUs), têm demonstrado capacidade de manter precisão elevada (97,45% mAP) com latência determinística e estável, uma vantagem significativa em relação à variabilidade temporal típica de CPUs e GPUs [1, 8, 23].

Com o avanço de métodos não supervisionados e auto-supervisionados, tornaram-se necessárias novas métricas para avaliar a eficiência desses modelos em distinguir placas normais de amostras defeituosas, sem o uso de rótulos explícitos. Para a validação interna, utilizam-se indicadores de qualidade de clusterização, como o Coeficiente de Silhouette, o Índice Calinski-Harabasz e o Índice Davies-Bouldin, que avaliam a compactação e separabilidade dos agrupamentos formados no espaço latente. Já a validação externa compara os agrupamentos obtidos com informações reais, utilizando métricas como o Adjusted Rand Index (ARI), permitindo verificar se o modelo realmente captura a estrutura subjacente dos dados e identifica anomalias de forma confiável. Essas abordagens ampliam o escopo da inspeção automatizada ao oferecer alternativas robustas mesmo em cenários com dados escassos ou rótulos limitados [20, 21].

### 4.3 Implementações em hardware

A literatura sobre inspeção automática de PCBs apresenta uma grande diversidade de técnicas de implementação em hardware, desde microcontroladores de baixo custo até FPGAs dedicados à aceleração de redes profundas. Em soluções mais simples, a detecção é realizada em um computador convencional ou em um microcontrolador, com apenas uma câmera e o processamento de imagens na CPU. Santos et al. (2022) desenvolvem um sistema de inspeção com câmera e transportador, no qual as imagens da placa são processadas em um laptop usando YOLO e bibliotecas OpenCV/TensorFlow, alcançando acurácia superior a 95% para trilhas rompidas e furos não perfurados, porém com dependência de um computador geral e sem otimização de hardware [25]. Sun et al. (2025) otimizam a inferência do YOLOv4 com a ferramenta OpenVINO em uma plataforma embarcada Intel, explorando quantização e aceleração em CPU e GPU para aumentar a taxa de quadros, mantendo o mAP próximo de 99% para defeitos de soldagem [26].

Arquiteturas baseadas em FPGA podem ser classificadas em dois grupos. No primeiro, o FPGA é usado para implementar apenas o processamento de imagem clássico. Navaneeth e Kannan propõem um sistema de inspeção de PCBs nus em que a imagem da placa em teste é convertida em formato binário e comparada, pixel a pixel, com uma imagem de referência ideal, totalmente em hardware [22]. O método localiza defeitos funcionais (curtos, aberturas, trilhas faltantes) em uma grade de 1024×1024 pixels, atingindo operação em tempo real e baixo custo, mas exige uma imagem de referência precisa e é sensível a desalinhamentos e variações de iluminação.

No segundo grupo, o FPGA atua como acelerador dedicado para redes profundas de detecção de objetos. Pan et al. (2024) propõem o YOLOx-Plus, que incorpora atenção SimAM, FPN+PAN e a função de perda SIoU, seguido de quantização fixa e do projeto de um acelerador em uma placa PYNQ-Z2 [23]. Os autores relatam acurácia média de 93,2% no conjunto de defeitos, redução de 64% no tamanho do modelo e aumento de 68,1% na velocidade de detecção, alcançando 72,6 QPS após a quantização e implementação no FPGA. Já Abraham et al. (2025) mostram um fluxo alternativo no qual uma rede leve é implantada sobre um DPU (Deep Learning Processing Unit) em FPGA, atingindo uma vazão de 2,44 trilhões de operações por segundo (TOPS), latência de aproximadamente 97,6 ms e acurácia de 0,789, com baixo consumo de energia [27]. Essa abordagem explora um IP de acelerador genérico (DPU), simplificando o desenvolvimento, mas nem sempre atinge a mesma eficiência tarefa-específica dos aceleradores desenhados sob medida para um único modelo.

Do ponto de vista de custo e desempenho, esses trabalhos fornecem referências importantes para este estudo comparativo. Implementações puramente em CPU ou em microcontrolador apresentam baixo custo de entrada e desenvolvimento fácil, porém consomem mais tempo de processamento e dependem de hardware geral. FPGAs com processamento de imagem clássico oferecem baixa latência e consumo moderado, mas são limitados a defeitos que possam ser descritos por operações de comparação com referência. Já FPGAs com DPU ou aceleradores específicos de YOLO representam um ponto intermediário: exigem maior esforço de projeto e ferramentas de síntese de alto nível; em compensação, permitem explorar paralelismo massivo, quantização agressiva e arquiteturas otimizadas, o que se reflete em QPS mais altos e em uma melhor relação desempenho/consumo. Nesta seção, os resultados de hardware do presente trabalho serão analisados à luz desses exemplos, destacando o papel de um DPU em FPGA na execução de modelos de detecção de defeitos e comparando-os à solução baseada em YOLOx-Plus.

## 5 Discussão

A partir da análise apresentada nas seções anteriores, torna-se evidente que as abordagens de detecção de defeitos em PCBs evoluíram significativamente tanto do ponto de vista algorítmico quanto em termos de plataformas de execução. Para sintetizar essa evolução e permitir uma comparação estruturada entre métodos, estratégias de tratamento da escassez de dados e arquiteturas de hardware, foram elaboradas três tabelas comparativas, derivadas diretamente dos artigos revisados ao longo desta pesquisa.

**Tabela 1: Comparação entre métodos de detecção de defeitos em PCBs.**

<b>Categoria</b>	<b>Exemplos</b>	<b>Métricas Reportadas</b>	<b>Vantagens</b>	<b>Limitações</b>
Algoritmos clássicos	Subtração, XOR, Morfologia	Raro reporte de mAP/QPS	Baixa complexidade; implementação direta em hardware; tempo real	Sensível à iluminação e ao desalinhamento; incapaz de detectar microdefeitos
Detectores Two-Stage	Faster R-CNN, ResNet50-FPN	mAP > 95%	Maior precisão; robustez a microdefeitos	Latência elevada (<20 QPS); alto custo computacional
Detectores One-Stage	YOLOv7, YOLOv12, SSD	mAP 97–99% @0.5	Equilíbrio precisão/velocidade; adequado à inspeção em linha	Perda de desempenho em tiny defects sem atenção multiescala
Atenção / Multiescala	LW-YOLO, CBAM, GAM	mAP até 99%	Melhora microdetecção; mais robusto a variações industriais	Exige otimizações específicas para hardware de borda
Transformers	LPViT	SOTA em DeepPCB	Captura dependências globais; robusto a variações de contexto	Alto custo computacional; dificuldade de implantação embarcada

Essas tabelas desempenham papel central na discussão: (i) A Tabela 1 compara diferentes métodos de detecção, desde algoritmos clássicos até arquiteturas modernas baseadas em atenção e em Transformers; (ii) a Tabela 2 organiza as principais abordagens para mitigar a escassez de dados rotulados, um desafio recorrente na indústria; e (iii) a Tabela 3 sintetiza os trade-offs entre plataformas de hardware, destacando o papel emergente de FPGAs como solução energeticamente eficiente.

### 5.1 Comparação entre métodos de detecção

A Tabela 1 evidencia uma clara evolução no desempenho e na robustez dos métodos empregados na detecção de defeitos. Os detectores Two-Stage, como Faster R-CNN com backbones ResNet50-FPN, ainda estabelecem a maior precisão absoluta, registrando mAP superior a 95%. No entanto, sua latência típica inferior a 20 QPS inviabiliza a aplicação direta em linhas SMT de alta velocidade.

Em contraste, arquiteturas de um estágio, especialmente variantes otimizadas do YOLO, atingem níveis de acurácia comparáveis (97–99%) enquanto mantêm taxas de processamento superiores a 80 QPS, demonstrando que a remoção de etapas redundantes e o uso de backbones eficientes são fundamentais para conciliar precisão e vazão industrial.

Outro avanço observado na literatura é a adoção de mecanismos de atenção e de fusão multiescala, que mitigam as perdas históricas na detecção de microdefeitos, um dos maiores desafios da inspeção moderna. Modelos como LW-YOLO, CDI-YOLO e variantes com módulos de atenção de bloco convolucional demonstram que a atenção espacial e de canal reforça a discriminação de padrões de baixíssima granularidade, o que era uma limitação conhecida das CNNs tradicionais. Por fim, embora arquiteturas baseadas em Transformers, como o LPViT, ofereçam robustez diante de variações de contexto, o custo computacional ainda impede sua adoção prática em sistemas embarcados de inspeção.

**Tabela 2: Comparação de estratégias para lidar com a escassez de dados na inspeção de PCBs.**

<b>Abordagem</b>	<b>Exemplos</b>	<b>Métricas Reportadas</b>	<b>Vantagens</b>	<b>Limitações</b>
Few-Shot Learning	FPFM	Alta precisão com 10–30 amostras	Reduz esforço de anotação; adequado para defeitos raros	Sensível à variação da distribuição de classes
Transfer Learning	YOLO pré-treinado	mAP elevado com poucos ajustes	Rápida adaptação a novos domínios	Pode herdar vieses do dataset base
Semi-Supervisionado	PCB_SS, FixMatch	ARI, Silhouette, aumento de consistência	Usa grandes volumes de dados não rotulados; estabiliza treinamento	Dependência de pseudo-rótulos
Anomalia / Não-Supervisionado	Tinyomaly (GAN)	Métricas de reconstrução e erro	Detecta defeitos inéditos; elimina rótulos	Variável em ambientes ruidosos ou pouco estruturados

Tabela 3: Comparação entre plataformas de hardware para inspeção de PCBs.

Plataforma	Exemplos	Métricas	Vantagens	Limitações
CPU / GPU Convencionais	Laptop+YOLO, OpenVINO, TensorRT	mAP 95–99%; QPS dependente do hardware	Fácil prototipagem; ampla compatibilidade	Latência não determinística; jitter; maior consumo
FPGA (Processamento Clássico)	Comparação binária em Verilog	Operação em tempo real (ex.: 1024×1024)	Baixíssima latência; consumo reduzido	Restrito a defeitos geométricos; depende de referência
FPGA (Aceleradores DL / DPU)	YOLOx-Plus em FPGA; DPU UltraScale+	72.6 QPS; ~93.2% mAP; até 2.44 TOPS	Paralelismo massivo; latência determinística; eficiente energeticamente	Exige síntese, quantização e tuning especializado

## 5.2 Estratégias para lidar com escassez de dados

A escassez de dados rotulados permanece um obstáculo recorrente, especialmente na inspeção industrial, onde os defeitos reais são raros. Como sintetizado na Tabela 2, métodos Few-Shot, como o FPFM, alcançam alta precisão mesmo com apenas 10 a 30 amostras por classe, embora sejam sensíveis à distribuição dos dados. Estratégias semi-supervisionadas, como PCB\_SS, demonstram maior estabilidade ao explorar grandes volumes de imagens não rotuladas, enquanto métodos totalmente não supervisionados baseados em GANs, como o Tinyomally, eliminam completamente a necessidade de rótulos ao aprender apenas a distribuição de amostras normais.

Esses resultados são particularmente relevantes para ambientes em que a ocorrência de falhas é estatisticamente rara e de difícil rotulamento. No entanto, dependem de pipelines de validação mais complexos, que incluem métricas como ARI, Silhouette e erro de reconstrução, o que sugere que tais abordagens devem ser combinadas com sistemas de inspeção tradicionais para garantir a confiabilidade operacional.

## 5.3 Comparação entre plataformas de hardware

A Tabela 3 evidencia que a escolha da plataforma de execução é tão determinante quanto o modelo adotado. CPUs e GPUs convencionais continuam sendo soluções adequadas para desenvolvimento e prototipagem, mas apresentam latência não determinística e maior consumo de energia, características incompatíveis com inspeção contínua em sistemas industriais.

FPGAs para processamento clássico representam uma alternativa extremamente eficiente em termos de latência, mas são limitados a operações geométricas e dependem de imagens de referência precisamente alinhadas. Já os FPGAs dotados de aceleradores especializados em aprendizado profundo, como as DPUs UltraScale+, emergem como a solução mais equilibrada: combinam paralelismo massivo, latência determinística e eficiência energética, alcançando mais de 70 QPS mesmo após quantização.

Esse conjunto de características sugere que FPGAs dotados de aceleradores especializados apresentam um equilíbrio técnico favorável entre paralelismo, latência e eficiência energética. Em cenários industriais de alta cadência, nos quais a previsibilidade

temporal é um requisito crítico, esses dispositivos mostram-se como uma alternativa robusta para processamento em borda.

## 5.4 Limitações da revisão

Algumas limitações metodológicas devem ser consideradas na interpretação dos resultados apresentados. Primeiramente, a seleção bibliográfica concentrou-se em bases de dados científicas reconhecidas, que, embora assegurem qualidade editorial e revisão por pares, podem introduzir viés ao excluir estudos relevantes publicados em repositórios alternativos ou em literatura não indexada. Outra limitação refere-se ao recorte temporal adotado, concentrado em publicações entre 2011 e 2025, que privilegia a análise dos avanços recentes associados às técnicas baseadas em aprendizado profundo, mas pode resultar na exclusão de contribuições anteriores relevantes relacionadas a métodos clássicos de inspeção.

Adicionalmente, os trabalhos analisados apresentam elevada heterogeneidade experimental, incluindo diferenças nos conjuntos de dados utilizados, nas métricas de avaliação e nas plataformas de hardware empregadas. Essa variabilidade limita comparações quantitativas diretas entre abordagens, o que faz com que as análises apresentadas tenham caráter predominantemente qualitativo e comparativo. Além disso, observa-se que grande parte dos estudos analisados foi conduzida em ambientes experimentais controlados, condição que pode não representar integralmente os cenários operacionais de linhas industriais reais, especialmente no que diz respeito a variações de iluminação e presença de ruído no processo de aquisição de imagens.

## 6 Conclusão

Este trabalho teve como objetivo analisar a evolução das técnicas de inspeção automatizada de placas de circuito impresso, investigando a transição dos métodos clássicos de processamento de imagens para abordagens baseadas em aprendizado profundo, bem como o papel das plataformas de computação embarcada na viabilização de aplicações industriais em tempo real. A revisão buscou integrar perspectivas algorítmicas e arquiteturais, relacionando o desempenho de detecção, a eficiência computacional e os requisitos operacionais de ambientes produtivos.

A síntese das arquiteturas revisadas, apresentada na Tabela 1, indica que os detectores de estágio único, especialmente aqueles equipados com módulos de atenção e de fusão multiescala, têm demonstrado um equilíbrio competitivo entre acurácia e desempenho operacional, tornando-os candidatos viáveis para inspeção em linhas de produção de alta cadência. A capacidade de capturar microdefeitos, antes um desafio recorrente na inspeção óptica, foi ampliada significativamente por mecanismos como o CBAM, o GAM e variantes otimizadas do YOLO. Embora métodos baseados em Transformers apresentem resultados promissores, seu elevado custo computacional ainda restringe seu uso direto em ambientes embarcados.

Complementarmente, os métodos destinados a mitigar a escassez de dados rotulados, sintetizados na Tabela 2, demonstram que estratégias de Few-Shot Learning, semi-supervisionadas e baseadas em anomalia ampliam consideravelmente o alcance da inspeção automática. Esses paradigmas tornam o processo mais viável em cenários industriais reais, em que os defeitos são raros, caros de rotular e distribuídos de forma desequilibrada. Assim, tais métodos emergem como pilares essenciais para a integração futura de sistemas inteligentes em linhas SMT.

Do ponto de vista científico e tecnológico, os resultados indicam que o desempenho de sistemas de inspeção não depende exclusivamente da arquitetura de rede empregada, mas também do co-projeto entre o algoritmo e o hardware de execução. Embora CPUs e GPUs sejam adequadas para desenvolvimento e prototipagem, FPGAs com aceleradores dedicados demonstram maior aptidão para atender aos requisitos de latência determinística, eficiência energética e operação contínua necessários em produção real. Esse resultado converge diretamente com a discussão apresentada na Introdução sobre a importância crescente do processamento em borda em sistemas de inspeção modernos.

Em síntese, os achados desta revisão apontam para uma convergência entre algoritmos cada vez mais compactos e eficientes, métodos avançados de aprendizado com poucos dados e plataformas de hardware reconfiguráveis. Essa combinação delinea o caminho para a próxima geração de sistemas de inspeção automática, capazes de operar em velocidades industriais, com elevada robustez, interpretabilidade e capacidade de adaptação a condições variáveis de manufatura.

## Agradecimentos

Este trabalho foi financiado, em parte, pela Fundação de Amparo à Pesquisa e Inovação do Estado de Santa Catarina – FAPESC, Chamada 51/2024 (Contrato 2024TR001897), pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq (Processo 136279/2025-2), e pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – CAPES/PROSUC.

## Referências

- [1] Boyuan Chen and Zichen Dang. Fast pcb defect detection method based on fasternet backbone network and cbam attention mechanism integrated with feature fusion module in improved yolov7. *Ieee Access*, 11:95092–95103, 2023.
- [2] Zhi Chen and Bingxiang Liu. A high-accuracy pcb defect detection algorithm based on improved yolov12. *Symmetry*, 17(7):978, 2025.
- [3] ALLPCB. The hidden costs of PCB defects: Mastering quality control in mass production. <https://www.allpcb.com/blog/pcb-manufacturing/the-hidden-costs-of-pcb-defects-mastering-quality-control-in-mass-production.html>, 2025. Acesso em: 11 dez. 2025.
- [4] SS Zakaria, Amiza Amir, Naimah Yaakob, and S Nazemi. Automated detection of printed circuit boards (pcb) defects by using machine learning in electronic manufacturing: Current approaches. In *Iop conference series: Materials science and engineering*, volume 767, page 012064. IOP Publishing, 2020.
- [5] Ajay Pal Singh Chauhan and Sharat Chandra Bhardwaj. Detection of bare pcb defects by image subtraction method using machine vision. In *Proceedings of the world congress on engineering*, volume 2, pages 6–8, 2011.
- [6] Joon-Hyung Park, Yeong-Seok Kim, Hwi Seo, and Yeong-Jun Cho. Analysis of training deep learning models for pcb defect detection. *Sensors*, 23(5):2766, 2023.
- [7] Ka Nam Canaan Law, Mingshuo Yu, Lianglei Zhang, Yiyi Zhang, Peng Xu, Jerry Gao, and Jun Liu. Enhancing printed circuit board defect detection through ensemble learning. In *2024 IEEE 1st International Workshop on Future Intelligent Technologies for Young Researchers (FITR)*, pages 37–44. IEEE, 2024.
- [8] Bing Hu and Jianhui Wang. Detection of pcb surface defects with improved faster-rcnn and feature pyramid network. *Ieee Access*, 8:108335–108345, 2020.
- [9] Gayathri Lakshmi, V Udaya Sankar, and Y Siva Sankar. A survey of pcb defect detection algorithms. *Journal of Electronic Testing*, 39(5):541–554, 2023.
- [10] Xing Chen, Yonglei Wu, Xingyou He, and Wuyi Ming. A comprehensive review of deep learning-based pcb defect detection. *Ieee Access*, 11:139017–139038, 2023.
- [11] Wanli Huang and Peng Wei. A pcb dataset for defects detection and classification. *arXiv preprint arXiv:1901.08204*, 2019.
- [12] S Ozturk and Bayram Akdemir. Detection of pcb soldering defects using template based image processing method. *International Journal of Intelligent Systems and Applications in Engineering*, 4(5):269–273, 2017.
- [13] Zhaohui Yuan, Xiangyang Tang, Hao Ning, and Zhengzhe Yang. Lw-yolo: Lightweight deep learning model for fast and precise defect detection in printed circuit boards. *Symmetry*, 16(4):418, 2024.
- [14] Gaoshang Xiao, Shuling Hou, and Huiying Zhou. Pcb defect detection algorithm based on cdi-yolo. *Scientific Reports*, 14(1):7351, 2024.
- [15] Runwei Ding, Linhui Dai, Guangpeng Li, and Hong Liu. Tdd-net: A tiny defect detection network for printed circuit boards. *CAAI Transactions on Intelligence Technology*, 4(2):110–116, 2019.
- [16] Yu Ting Li and Jiun In Guo. A vgg-16 based faster rcnn model for pcb error inspection in industrial aoi applications. In *2018 IEEE international conference on consumer electronics-Taiwan (ICCE-TW)*, pages 1–2. IEEE, 2018.
- [17] Kang An and Yanping Zhang. Lpvit: A transformer-based model for pcb image classification and defect detection. *Ieee Access*, 10:42542–42553, 2022.
- [18] Haodong Wang, Jun Xie, Xinying Xu, and Zihao Zheng. Few-shot pcb surface defect detection based on feature enhancement and multi-scale fusion. *Ieee Access*, 10:129911–129924, 2022.
- [19] Wei Shi, Linlin Zhang, Yihui Li, and Hong Liu. Adversarial semi-supervised learning method for printed circuit board unknown defect detection. *The Journal of Engineering*, 2020(13):505–510, 2020.
- [20] Thi Tram Anh Pham, Do Kieu Trang Thoi, Hyohoon Choi, and Suhyun Park. Defect detection in printed circuit boards using semi-supervised learning. *Sensors*, 23(6):3246, 2023.
- [21] Md Rakibul Islam, Shahina Begum, and Mobyen Uddin Ahmed. Interpretable self-supervised learning for fault identification in printed circuit board assembly testing. *Applied Sciences*, 15(18):10080, 2025.
- [22] Kenguva Navaneeth and Rajesh Kannan. Fpga-based defect detection in pcbs using image processing. In *2025 8th International Conference on Circuit, Power & Computing Technologies (ICCPCT)*, pages 853–858. IEEE, 2025.
- [23] Yajing Pan, Lei Zhang, and Yujie Zhang. Rapid detection of pcb defects based on yolox-plus and fpga. *Ieee Access*, 12:61343–61358, 2024.
- [24] Can Zhang, Wei Shi, Xiaofei Li, Haijian Zhang, and Hong Liu. Improved bare pcb defect detection approach based on deep feature learning. *The Journal of Engineering*, 2018(16):1415–1420, 2018.
- [25] Agus Dwi Santoso, Ferry Budi Cahyono, Brendi Prahasta, Imam Sutrisno, and Agus Khumaidi. Development of pcb defect detection system using image processing with yolo cnn method. *International Journal of Artificial Intelligence Research*, 6(1):343, 2022.
- [26] Mau-Luen Tham, Boon Yoong Chong, Yu Han Tan, Yen Khai Wong, Swee Ling Chean, and Wei Kun Tan. Optimizing deep learning inference to detect pcb soldering defects. In *2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (ICAET)*, pages 1–5. IEEE, 2022.
- [27] Neethan Elizabeth Abraham, Joseph Biju, Nandu Krishna, and Navneeth Sunil Varrma. Circuit scan: Pcb error detection using microcontroller. In *2025 International Conference on Advances in Modern Age Technologies for Health and Engineering Science (AMATHE)*, pages 1–5. IEEE, 2025.