

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS
GERAIS - CAMPUS SABARÁ
BACHARELADO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

LUIZ HENRIQUE DINIZ DE OLIVEIRA

**ANÁLISE DE FALHAS EM SUPERFÍCIES: utilização de
sistemas de visão para identificação e classificação de
falhas em placas de circuito impresso**

Sabará
2024

Luiz Henrique Diniz De Oliveira

ANÁLISE DE FALHAS EM SUPERFÍCIES: utilização de sistemas de visão para identificação e classificação de falhas em placas de circuito impresso

Trabalho de Conclusão de Curso apresentado à banca examinadora do curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais Campus Sabará, como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Controle e Automação.

Orientador: Prof. Dr. Diego Miranda

Sabará
2024

O48a

Oliveira, Luiz Henrique Diniz de

Análise de falhas em superfícies [manuscrito]: utilização de sistemas de visão para identificação e classificação de falhas em placas de circuito impresso. / Luiz Henrique Diniz de Oliveira. - 2024.

49 f. : il.

Orientação: Prof. Dr. Diego Oliveira Miranda.

Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Controle e Automação) – Instituto Federal de Minas Gerais, *Campus* Sabará.

1. Localização de falhas (Engenharia). – Monografia. 2. Superfícies (Tecnologia). – Monografia. 3. Circuitos impressos. – Monografia. I. Miranda, Diego Oliveira. II. Instituto Federal de Minas Gerais, *Campus* Sabará. III. Bacharelado em Engenharia de Controle e Automação. IV. Título.

CDU 681.5.09

ATA DE DEFESA DO TCC

Ao dia 09 do mês de outubro do ano de 2024, às 17:05 horas, sob a presidência do Professor Diego Oliveira Miranda, o discente Luiz Henrique Diniz de Oliveira do Curso de Engenharia de Controle e Automação, R.A nº 0050043 do IFMG – Campus Sabará, defendeu o Trabalho de Conclusão de Curso (TCC) intitulado “**ANÁLISE DE FALHAS EM SUPERFÍCIES: utilização de sistemas de visão para identificação e classificação de falhas em placas de circuito impresso**” e foi avaliado com a nota final média de 60 pontos, que está condicionada ao cumprimento dos procedimentos pós-defesa do TCC.

Compuseram a Banca Examinadora:


Membro 1: Prof Diego Oliveira Miranda (Presidente),

Membro 2: Prof Luiz Guilherme Hilel Drumond Silveira,


Membro 3: Prof Daniel Neves Rocha.

O discente deverá apresentar o trabalho com as devidas modificações em formato pdf, até 22 /11/2024 à Coordenação de TCC e fazer o depósito no repositório institucional de TCC do IFMG. O não cumprimento dos procedimentos pós-defesa de TCC até a data estipulada implica no não cumprimento das horas referentes aos componentes curriculares de TCC I e II.


Assinaturas

Documento assinado digitalmente
 **DIEGO OLIVEIRA MIRANDA**
Data: 10/10/2024 22:18:05-0300
Verifique em <https://validar.iti.gov.br>

Membro 1: Prof Diego Oliveira Miranda (Presidente), _____

Documento assinado digitalmente
 **LUIZ GUILHERME HILEL DRUMOND SILVEIRA**
Data: 11/10/2024 11:28:06-0300
Verifique em <https://validar.iti.gov.br>

Membro 2: Prof Luiz Guilherme Hilel Drumond Silveira, _____

Documento assinado digitalmente
 **DANIEL NEVES ROCHA**
Data: 10/10/2024 19:39:54-0300
Verifique em <https://validar.iti.gov.br>

Membro 3: Prof Daniel Neves Rocha. _____

AGRADECIMENTOS

Em primeiro lugar gostaria de agradecer aos meus pais, que sempre me apoiaram e me deram suporte, seja qual fosse a situação. Me deram amor, acolhimento e se desdobraram para me fornecer uma boa educação, que no final das contas foi determinante para que eu conquistasse todo o resto.

À minha família e amigos que estiveram comigo em todos os momentos importantes. Me estimularam, apoiaram e me proporcionaram bons momentos de descanso, distração e alegria.

Gostaria de agradecer também aos meus amigos do IFMG os quais compartilhei tantos momentos, tenham sido eles de alegria, preocupação, de aprendizado ou os roles no meio da semana, que sem dúvida aliviaram de forma considerável a carga do dia a dia. Em especial a Vitoria, minha namorada, que esteve comigo nas madrugadas de estudo, nos trabalhos em grupo e em praticamente todo o resto.

Aos professores e funcionários do IFMG que nessa caminhada estiveram tão disponíveis, seja para ensinar, ajudar ou simplesmente conversar, deixo também o meu muito obrigado.

Por fim, gostaria de agradecer principalmente a Deus, que me guia a todos os momentos, me acalenta e me conforta. Aquele que se faz presente em todas as coisas e que quando mais ninguém pode ajudar é o verdadeiro provedor. A Fé sempre foi uma das coisas que me mantiveram de pé, alegre e durante a graduação não foi diferente.

RESUMO

Os sistemas eletrônicos modernos desempenham uma função muito importante na nossa sociedade, estando presentes em equipamentos das mais variadas vertentes e contribuindo ativamente com vários setores da economia, dentre eles o setor de comunicação, transporte, saúde e segurança. Parte fundamental desses sistemas, as chamadas PCI's (Placas de circuito impresso) tem a importante função de integrar todos os outros componentes, que são responsáveis pela execução dos comandos em si. No entanto, a crescente complexidade desses sistemas, muito devido à ideia de compactação dos equipamentos, traz consigo a possibilidade de falhas, que podem originar diversas consequências.

Na indústria, um dos problemas mais recorrentes encontrados em placas de circuito impresso, são as falhas envolvendo o processo de soldagem, o que pode impactar no funcionamento do equipamento, na redução de sua vida útil ou até mesmo na quebra total do equipamento, o que em atividades mais específicas poderia significar colocar em risco a vida das pessoas, seja um operador ou o consumidor final.

Logo, o objetivo do presente trabalho é desenvolver um sistema inteligente, orientado por uma RNA Classificatória (IA), que utiliza imagens de alta definição para executar varreduras nas PCI's com o intuito de identificar e classificar esses erros, minimizando ou até mesmo mitigando seus impactos, proporcionando maior segurança e confiabilidade aos produtos ou equipamentos aos quais as PCI's fazem parte.

Palavras-chave: Falhas; Placas de circuito Impresso (PCI); Inteligência Artificial; Solda.

RESUMEN

Los sistemas electrónicos modernos juegan un papel muy importante en nuestra sociedad, estando presentes en equipamientos de las más variadas vertientes y contribuyendo activamente a diversos sectores de la economía, incluyendo los sectores de comunicaciones, transporte, salud y seguridad. Parte fundamental de estos sistemas, las llamadas PCI's (Placas de Circuito Impreso) tienen la importante función de integrar todos los demás componentes, que son los encargados de ejecutar los comandos por sí mismos. Sin embargo, la creciente complejidad de estos sistemas, en gran parte por la idea de compactar los equipos, trae consigo la posibilidad de fallas, que pueden acarrear en demasiadas consecuencias.

En la industria, uno de los problemas más recurrentes que se encuentran en los circuitos impresos son las fallas en el proceso de soldadura, que pueden impactar en el funcionamiento del equipo, reducir su vida útil o incluso averiar por completo el equipo, lo que en actividades más específicas puede significar ponerse en riesgo la vida de las personas, sea un operador o el consumidor final.

Por lo tanto, el objetivo del presente trabajo es desarrollar un sistema inteligente, guiado por una RNA Clasificadora (IA), que utilice imágenes de alta definición para realizar escaneos sobre los PCBs con el fin de identificar y clasificar estos errores, minimizando o incluso mitigando sus impactos, proporcionando mayor seguridad y confiabilidad a los productos o equipos a los que forman parte los PCBs..

Palabras-llaves: Fracasos; placas de circuito impreso (PCI); Inteligencia artificial; Soldar.

LISTA DE ILUSTRAÇÕES

Figura 1 – Sistema de Visão.....	17
Figura 2 – Visão Computacional.....	18
Figura 3 – Estrutura da Rede Neural Simples	20
Figura 4 – Rede Neural Convulacional.....	21
Figura 5 – Placa de circuito impresso.....	24
Figura 6 – Variações dos pontos de solda.....	24
Figura 7 – Comparação da qualidade entre soldas boas e ruins.....	24
Figura 8 – Falha de solda em placa de circuito impresso real.....	24
Figura 9 – PCI's danificadas.....	25
Figura 10 – Defeitos de solda.....	27
Figura 11 – Camadas do modelo.....	30
Figura 12 – Divisão dos dados.....	32
Figura 13 – Simulações cenário 4T.....	37
Figura 14 – Simulação Tanh cenário 2T.....	38
Figura 15 – Simulação Sigmoid cenário 2T.....	38
Figura 16 – Simulação Tanh cenário 3T.....	39
Figura 17 – Simulação validação sub-ótima.....	40
Figura 18 – Relatório de classificação de imagens.....	42
Figura 19 – Predição por classes do Modelo.....	42
Figura 20 – Rotulagem individual Aptas.....	43
Figura 21 – Rotulagem individual N-Aptas.....	43

LISTA DE TABELAS

Tabela 1 – Machine learning e Deep Learning.....	19
Tabela 2 - Simulação Sigmoid.....	36
Tabela 3 - Simulação Tanh.....	36
Tabela 4 - Simulação Relu.....	37

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
IFMG	Instituto Federal de Minas Gerais
PCI	Placa de circuito impresso
IA	Inteligência artificial
SMT	Surface Hole Technology
SMD	Surface Mounting Devices
THT	Through Hole Technology
PTH	Pin Through Hole
CNN	Convolutional neural network

Sumário

1	INTRODUÇÃO	10
1.1	Objetivos	12
1.1.1	<i>Objetivo geral</i>	12
1.1.2	<i>Objetivos específicos</i>	12
1.2	Justificativa	13
1.3	Organização do Texto	14
2	REVISÃO BIBLIOGRÁFICA	15
3	FUNDAMENTAÇÃO TEÓRICA	17
3.1	Sistemas de visão	17
3.2	Visão computacional e Processamento de Imagens	18
3.3	Inteligência Artificial e o Aprendizado de Máquina	19
3.4	Redes Neurais (MLP) e Redes Neurais convolucionais (CNNs)	20
3.4.1	<i>Overfitting e underfitting</i>	21
3.5	Problemas de Solda	22
4	METODOLOGIA	26
4.1	Definição do Instrumento de aquisição de imagens	26
4.2	Delimitação dos tipos de falhas	27
4.3	Pré-processamento dos Dados	28
4.3.1	<i>Organização dos dados</i>	28
4.3.2	<i>Redimensionamento e Normalização</i>	28
4.3.3	<i>Aumentação de Dados</i>	29
4.4	Desenvolvimento do Modelo	29
4.4.1	<i>Definição dos Hiper parâmetros</i>	30
4.5	Treinamento do Modelo	32
4.5.1	<i>Testes de Desempenho</i>	33
4.6	Avaliação e Métricas de Desempenho	33
5	RESULTADOS	34
6	CONCLUSÃO	45
7	REFERÊNCIAS	47

1 INTRODUÇÃO

A eletrônica moderna, assim denominada, constitui a área de estudo dedicada às aplicações de dispositivos baseados na movimentação dos elétrons, seja em semicondutores, gases ou no vácuo. (LANGUAGES, 2013). Essa área de estudo vem se desenvolvendo de forma bastante acelerada desde meados dos anos 1940, embora tenha surgido bem antes disso, fomentada por grandes nomes como Thomas Edson e Nikola Tesla. Esse crescimento foi impulsionado pela invenção dos transistores, componentes responsáveis por ampliar ou interromper a passagem de corrente elétrica, invenção oriunda dos trabalhos publicados pelos estudiosos John Bardeen, Walter Brainttain e Willian Shockley (BRAGA, 2023).

Ao longo dos anos, os sistemas eletrônicos evoluíram significativamente, integrando dispositivos, componentes e circuitos de maneira cada vez mais sofisticada. Essa evolução impulsionou sua popularização nos principais nichos comerciais, tornando-os fundamentais para executar funções primárias desses dispositivos, como alimentação, parametrização, interpolação de dados e comunicação ativa. (MEELO; RIOS; GUTIERREZ, 2001). Além disso, a indústria moderna tem sido marcada pela redução física dos insumos que constituem os principais sistemas eletrônicos, tais como os diodos, resistores, capacitores e indutores, componentes esses que tiveram uma redução dimensional significativa, o que não atrapalhou em nada o avanço no desempenho destes sistemas, que inclusive estão cada vez mais robustos (MEELO; RIOS; GUTIERREZ, 2001).

Atualmente grande parte dos equipamentos que são comercializados no mundo tem em sua composição estruturas eletrônicas, responsáveis por controlar e coordenar o dispositivo na execução da atividade para a qual foi desenvolvido, Dentro desse contexto, as placas de circuito impresso (PCI's) ganharam grande relevância, consolidando-se como um dos principais componentes eletrônicos existentes. Sua função essencial é integrar os diversos componentes eletrônicos, permitindo que trabalhem de forma conjunta para executar uma determinada função (CROMATEK, 2023). É correto afirmar que o bom funcionamento das PCI's está totalmente atrelado à qualidade do conjunto funcional ao qual ela pertence (Blom, 2013).

Se tratando das placas de circuito impresso, existem dois métodos de montagem principais que são atualmente utilizados: a montagem SMT (do inglês, Surface Mount Technology), que emprega componentes do tipo SMD (do inglês, Surface Mount Device), e a montagem THT (do inglês, Through Hole Technology), que utiliza componentes do tipo PTH (do inglês, Pin Through Hole). (CROMATEK, 2023).

Os componentes SMD (do inglês, Surface Mount Device) geralmente são muito pequenos e de difícil manuseio o que faz com que as montagens do tipo SMT (do inglês, Surface Mount Technology) geralmente sejam executadas por meio de máquinas especializadas. Nesse processo de montagem os componentes são fixados de forma superficial na placa, sem a necessidade de furações, de modo que o posicionamento dos componentes é orientado por meio de sistemas automáticos de coordenadas. Já nas montagens do tipo THT (do inglês, Through Hole Technology), as placas possuem furações em pontos específicos e os componentes são inseridos de forma manual. Por este motivo, nas grandes empresas geralmente são exigidas verificações de rotina, em sua maioria por meio de análises visuais ou por meio dos testes de amostragem (CROMATEK, 2023).

Embora distintas fisicamente e separadas pela forma como são montadas, as PCI's em sua maioria têm os processos de fixação atrelados ao processo de soldagem, que consiste na união das ligas metálicas, presentes nos componentes, à superfície desejada, por meio da inserção de um "metal de adição", que passou por um processo de aquecimento intenso. O principal objetivo desse processo é proporcionar a junção permanente dos insumos e geralmente é realizado com a utilização de materiais de boa condutância, de forma a facilitar o fluxo de corrente e a boa comunicação entre os componentes eletrônicos utilizados na placa.

Entretanto, caso não sejam realizadas de forma adequada, as soldas se tornam um grande empecilho nos sistemas, sendo muitas vezes responsáveis pela execução incorreta da atividade e, em casos mais graves, a fonte de grandes custos, causando a obsolescência da placa e podendo até provocar a quebra total do equipamento (BRAGA, 2023).

Os erros mais comuns encontrados durante as análises de placas de circuito impresso geralmente estão relacionados a problemas no processo de soldagem, como soldas frias, soldas quebradiças, excessos de solda e as chamadas solder balls. Com base nessas falhas, frequentemente identificadas, diversas técnicas de análise começaram a ser exploradas para melhorar a qualidade das placas de circuito impresso (PCI's), entre elas o uso de visão computacional e das técnicas de processamento avançado de imagem. Essas abordagens são detalhadas nas obras de (JESUS; ORELLANA, 2021) e (CABELLO, 2013), respectivamente, que têm como objetivo principal a identificação e a otimização dos processos de soldagem em PCI's. A utilização dessas tecnologias não só aprimora a precisão na análise, mas também contribui para a automação e eficiência da produção. Este tema, que aborda a integração de técnicas avançadas de análise de imagem nos processos de soldagem, será o foco central do presente trabalho.

1.1 Objetivos

1.1.1 Objetivo geral

O objetivo deste trabalho é desenvolver um sistema de inteligência artificial, capaz de identificar e classificar as principais falhas oriundas do processo de soldagem em placas de circuito impresso (PCI's).

Sua implementação visa aumentar a eficiência no processo de inspeção das placas, resultando em uma análise automatizada de retorno rápido, que possa auxiliar na manutenção e na qualidade da produção, reduzindo o tempo e os custos associados à detecção das falhas.

1.1.2 Objetivos específicos

O objetivo é estruturar um sistema capaz de separar as placas analisadas em dois grupos: "**Aptas**" e "**Não Aptas**", se baseando nas técnicas de aprendizado de máquina, especificamente a técnica baseada em rede Neural Convolucional (CNN), que fará a análise das imagens identificando potenciais problemas construtivos, que podem vir a comprometer o seu funcionamento.

Para cumprir com este objetivo será necessário:

1. Um objeto de captura de imagens, podendo ser utilizada uma câmera de boa resolução ou um microscópio eletrônico industrial.
2. Separação das imagens em grupos específicos de falhas o que mais a frente irá contribuir com o treinamento da rede.
3. Estruturação de um código (preferencialmente em Python) que consiga ordenar as informações obtidas e processá-las, de forma a retornar se existe ou não uma falha.
4. Estruturação de uma interface que exiba ao usuário, de forma visual, o tipo de falha apresentado.

O objetivo é identificar como "**Não Apta**" todas as placas que não estejam em plena condição de uso.

1.2 Justificativa

Com o avanço contínuo da tecnologia, os sistemas eletrônicos têm se tornado cada vez mais robustos e confiáveis, porém, os desvios produtivos e falhas ainda são realidade em diversos processos de fabricação. Esses problemas podem comprometer a qualidade e a funcionalidade das placas de circuito impresso (PCI's), gerando perdas significativas além de todo o retrabalho. Diante desse cenário, torna-se imprescindível a implementação de métodos de verificação preditiva, que permitam identificar e corrigir as falhas de forma antecipada. Desse modo, investir em sistemas que envolvam visão computacional e o processamento avançado de imagem, se mostram muito eficientes para essa finalidade, sejam elas verificações amostrais ou peça a peça. Esses sistemas não só aumentam a precisão na detecção de falhas, como também oferecem a possibilidade de realizar análises em tempo real, reduzindo o risco de erros e garantindo a produção de produtos de alta qualidade. A adoção dessas tecnologias, portanto, não é apenas uma estratégia de mitigação de falhas, mas também uma forma de otimizar o processo produtivo, aumentando a confiabilidade e a competitividade no mercado.

Alguns dos problemas mais comuns que os métodos de verificação automatizada podem auxiliar na identificação incluem:

- Pontos em Curto-Circuito;
- Trilhas danificadas;
- Soldas frias
- Solder ball's
- Pontos em sobreaquecimento;

A relevância dessas verificações torna-se ainda mais evidente quando se considera que, na sociedade atual, a vida cotidiana é amplamente dependente de equipamentos eletrônicos, os quais podem ser diretamente impactados por essas falhas.

No contexto industrial, a prevenção de falhas de solda em placas de circuito impresso (PCI's) apresenta inúmeros benefícios, essenciais para otimizar os processos de produção e garantir a entrega de produtos de alta qualidade. Entre esses benefícios, destacam-se:

- **Identificação da causa raiz do problema:** permite corrigir falhas de forma definitiva, reduzindo a probabilidade de recorrência e aprimorando continuamente os processos.

- **Melhoria na qualidade dos produtos:** assegura que os itens finais atendam aos padrões exigidos pelo mercado e superem as expectativas dos consumidores.
- **Redução de quebras e paralisações inesperadas:** minimiza interrupções na produção, aumentando a produtividade e reduzindo impactos financeiros.
- **Diminuição dos custos com substituições ou retrabalhos:** reduz despesas operacionais, aumentando a eficiência econômica da linha de produção.
- **Maior segurança para os operadores e o cliente final:** mitiga riscos associados ao manuseio de produtos defeituosos, preservando a integridade tanto dos colaboradores quanto dos consumidores.
- **Aumento da confiabilidade do produto:** eleva a reputação da marca, promovendo maior fidelização e competitividade no mercado.

Com base nesses fatores, destaca-se a importância de adotar métodos avançados de verificação, como os baseados em visão computacional. Essas tecnologias não apenas ampliam a eficiência e precisão na identificação de falhas, mas também garantem produtos mais seguros e confiáveis. Além disso, a utilização de tais métodos reforça a capacidade das empresas de se adaptarem às exigências de um mercado cada vez mais competitivo, assegurando maior sustentabilidade e inovação nos processos produtivos.

1.3 Organização do Texto

- (a) Introdução - Capítulo que comporta uma breve explicação do tema e de suas aplicações.
- (b) Revisão Bibliográfica - Capítulo onde serão abordados trabalhos similares ou que forneçam um viés que contribua ativamente para o tema.
- (c) Fundamentação Teórica - Capítulo que tem como intuito fornecer as informações necessárias para um melhor entendimento do tema.
- (d) Metodologia – Parte do texto onde são apresentados os métodos e as técnicas que foram utilizadas para desenvolver o trabalho.
- (e) Conclusão – Neste capítulo serão apresentadas as conclusões acerca da metodologia desenvolvida e dos resultados obtidos do decorrer do trabalho.
- (f) Referências – Tópico onde estão contidos todos os artigos, livros e sites que serviram de base para execução do trabalho.

2 REVISÃO BIBLIOGRÁFICA

Com o avanço da tecnologia, tornou-se eminente a redução dimensional dos componentes eletrônicos, o que gerou a necessidade de refinar e otimizar as técnicas de inspeção existentes, isso devido à maior complexidade construtiva que as placas de circuito impresso tem apresentado. Essas inspeções têm o intuito de estabelecer parâmetros de qualidade e verificar de forma amostral o estado produtivo das mesmas (Ali e Rahman, 2015).

Ainda de acordo com (Ali e Rahman, 2015), a modernização das técnicas de verificação de qualidade tem como intuito atestar de forma mais assertiva, o correto funcionamento dos equipamentos e garantir que eles não apresentem falhas ou quebras, reduzindo então os prejuízos financeiros e a sua obsolescência prematura. Isso porque, para os autores, as falhas ocorridas durante o processo de soldagem são talvez o ponto mais crítico na montagem das placas de circuito impresso, tornando as técnicas de verificação tão importantes, assegurando maior confiabilidade e longevidade aos equipamentos.

Já segundo o trabalho de (Davis, 2005), focado na identificação das falhas e nos métodos para mitigá-las, o primeiro passo é estabelecer um levantamento abrangente sobre os tipos de falha e os mecanismos que podem contribuir para o surgimento delas, além de avaliar se as mesmas ocorrem vezes suficientes para configurar um padrão, ou se são apenas problemas isolados, originados por casualidades e erros pontuais.

Feito isso, torna-se possível empregar um método sistemático para análise dessas falhas, onde (D.L Olson, 2002), propôs os seguintes passos:

- Identificação da falha;
- Coleta das evidências;
- Análise aprofundada das evidências;
- Identificação da causa;
- Recomendações para evitar a repetição da falha;

De modo que, o último passo seria a correção da falha por si só.

Em suas obras (J.R. Davis, 2005) e (D.L Olson, 2002) trataram de forma bem minuciosa sobre os vários fatores que contribuem para o surgimento das falhas em placas de circuito impresso e propuseram métodos para minimizar o seu surgimento. Os dois autores destacam também as principais formas de

rastrear, identificar e categorizar as falhas envolvendo solda, dando grande destaque às análises microscópicas.

(D.L Olson, 2002), destacou em suas obras que embora existam variados métodos para identificação de falhas em placas de circuito impresso, a microscopia ainda fornece um resultado mais eficiente, com uma imagem mais limpa e detalhes mais ricos.

Isso se reafirma na publicação de (Ali e Rahman, 2016) onde muito se discute sobre a importância da microscopia, abordando importantes conceitos como o de SEM (microscopia eletrônica de varredura), utilizada para obter imagens de alta resolução que facilitam na identificação e classificação da falha, sendo um dos instrumentos de análise empregados dentro das principais plantas industriais.

Em relação às análises microscópicas modernas, obras como as de (Maas,2014) e (Jones, 2016) abordam de forma bem abrangente o que há de mais moderno, onde além de tratarem dos principais conceitos envolvendo a microscopia, abordam as principais técnicas de execução, as limitações dos métodos e a forma mais eficiente de utilizar a microscopia em análises de circuito impresso.

De acordo com (Zhang, Lu, & Fu, 2018), os métodos de obtenção das imagens são apenas o início do processo, em que as ferramentas de processamento e parametrização se fazem extremamente necessárias para eficiência das análises. Foi o que abordaram em sua publicação à revista multidisciplinar IEEE Transactions on Industrial Informatics, em um artigo exploratório sobre a utilização das redes neurais convolucionais (CNNs) na detecção de defeitos em placas de circuito impresso e trouxeram à tona temas relevantes sobre a implementação da visão computacional no cenário mundial.

Nesse cenário, trabalhos como o de (Chen, J., & Li, H, 2020), se fazem extremamente relevantes, pois compilam, de forma bastante coerente, várias abordagens sobre o uso da visão computacional nos sistemas de identificação de falhas, explorando de forma sistemática diferentes modelos e técnicas de pré-processamento de imagens.

3 FUNDAMENTAÇÃO TEÓRICA

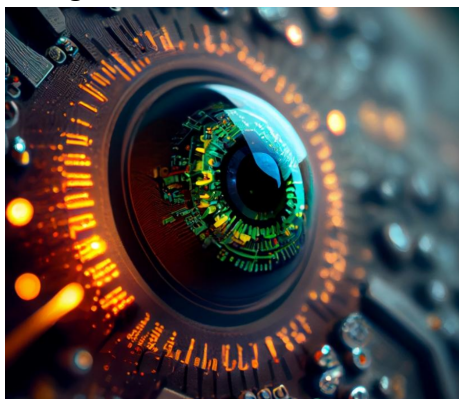
Neste tópico serão abordados os principais conceitos que se fazem necessários para um entendimento mais claro do tema, além de apresentar de forma mais abrangente, o modo como a inteligência artificial, aliada a um sistema de visão computacional robusto pode contribuir para identificação e classificação das falhas de soldas, presentes nos mais variados e complexos tipos de placas, contribuindo de forma fundamental para redução dos impactos provocados por elas.

3.1 Sistemas de visão

O sentido da visão é um dos instrumentos utilizados pelos seres humanos para capturar e interpretar as informações do meio em que vivem. É o sentido que lhes permite interagir ativamente e desfrutar do dinamismo do meio, além de perceber as cores e contribuir com cerca de 75% da nossa percepção das coisas (SIEGWART et al, 2004).

A imagem é formada na retina e é resultado das interações com a luz, captada no meio ambiente, o que se traduz em uma imagem projetada no cristalino, que na prática faz um trabalho semelhante às lentes das câmeras fotográficas, fazendo o ajuste do plano focal (VENTURA, 1995).

Figura 1 – Sistema de Visão.



Fonte: [Visão Computacional: o que é e como ela impacta o varejo físico? - Deconve](#)

Após a percepção da imagem, os nervos ópticos conduzem os impulsos elétricos ao cérebro, onde a informação é processada e transformada em uma imagem compreensível. Inspiradas na complexidade e eficiência desse processo, as câmeras fotográficas modernas replicam o dinamismo do olho humano para capturar imagens de forma semelhante. Essa inspiração biológica impulsionou o desenvolvimento da visão computacional, tornando-a uma área de estudo cada vez mais relevante e essencial em diversas aplicações tecnológicas.

3.2 Visão computacional e Processamento de Imagens

Embora a indústria moderna disponha de inúmeros recursos e sensores capazes de operar em escalas mínimas, os sistemas de imagem destacam-se como os mais confiáveis e com maior capacidade de medição (FU, 1987). Nesse contexto, a visão computacional desempenha um papel fundamental, ao capturar, processar e analisar dados visuais com o objetivo de identificar padrões, características ou objetos relevantes, oferecendo soluções precisas e eficientes para diversas aplicações industriais.

Figura 2 – Visão Computacional.



Fonte: [2Q== \(225x225\)](#)

O desenvolvimento dos algoritmos avançados e das técnicas de processamento de imagens permitiu a criação de ferramentas capazes de segmentar objetos, detectar bordas, remover ruídos e aprimorar a qualidade visual das imagens (BRADSKI, G, KAEHLER, 2008).

Um bom exemplo de ferramenta utilizada para aplicações envolvendo visão computacional é a OpenCV (do inglês Open Source Computer Vision Library). Essa biblioteca utiliza muito dos atributos que envolvem as CNNs (do inglês Convolutional neural network) permitindo que os modelos desenvolvidos “aprendam” a partir das representações abstratas das características das imagens. Essas redes são treinadas em conjuntos de dados volumosos e rotulados, permitindo que consigam a associar padrões visuais. (Franke, E., Zonta, T., & Hennrichs, J. C, 2022)

Combinadas com uma linguagem de programação simples e robusta (como por exemplo o Python), as CNNs podem ser implementadas e ajustadas com facilidade, tornando-se uma ferramenta essencial para o desenvolvimento de sistemas de reconhecimento de imagens altamente precisos (KRIZHEVSKY, 2012).

3.3 Inteligência Artificial e o Aprendizado de Máquina

A Inteligência artificial (IA) traduz toda a relação existente entre a capacidade das máquinas de simularem ou esboçarem a inteligência humana para realização de determinadas tarefas, que por sua vez demandam altos níveis de aprendizado e adaptação. Apesar disso, a IA ainda se mostra bastante limitada na forma como transmitir essas informações, não tendo alcançado seu potencial máximo (Russell & Norvig, 2016).

A Inteligência Artificial tem demonstrado um impacto muito relevante no contexto da indústria 4.0, especialmente na parte que diz respeito a automação dos processos industriais e às inspeções de qualidade nos ambientes mais agressivos. Nesse novo cenário, marcado pela constante necessidade de automatização dos processos industriais, outros importantes conceitos têm se destacado bastante, dentre eles o Aprendizado de Máquina (Machine Learning) e o Aprendizado profundo (Deep Learning), responsáveis por viabilizar o aprendizado autônomo e contínuo dos sistemas computacionais. Para isso, se utilizam de diversos dados históricos fornecidos pelo sistema ou processo, aumentando ainda mais sua confiabilidade e assertividade a cada nova interação, mesmo que não tenham sido programados especificamente para isso (Gonzalez & Woods, 2008; Szeliski, 2010). As principais diferenças entre as duas técnicas podem ser visualizadas na tabela 1.

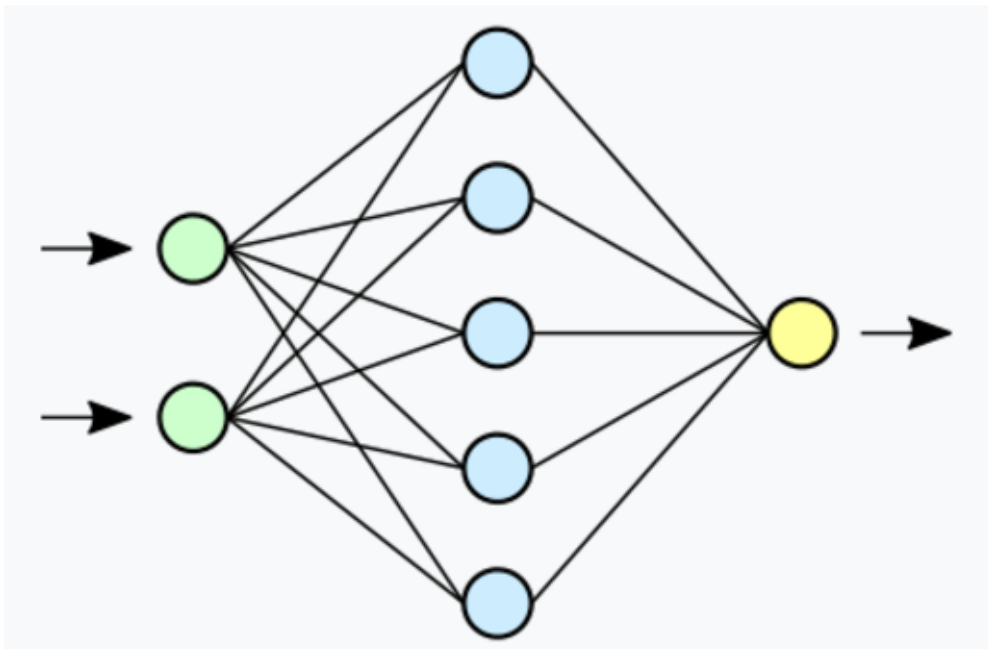
Tabela 1 – Machine learning e Deep Learning.

Aspecto	Machine Learning (ML)	Deep Learning (DL)
Entrada de dados	Requer que o especialista defina manualmente as características que serão analisadas nas imagens de PCs.	Extraí automaticamente as características mais relevantes das imagens, detectando padrões complexos.
Complexidade	Funciona bem para problemas de baixa complexidade com poucos dados.	Ideal para problemas complexos e grandes volumes de dados, como a análise detalhada de PCs.
Precisão	A precisão depende da qualidade das características definidas pelo especialista.	Altíssima precisão ao lidar com imagens detalhadas e de alta resolução.
Tempo de treinamento	Geralmente, mais rápido e com menor necessidade de recursos computacionais.	Pode demandar mais tempo e poder de processamento devido à profundidade das redes neurais.

3.4 Redes Neurais (MLP) e Redes Neurais convolucionais (CNNs)

Em um cenário onde muito se houve falar sobre Aprendizado de Máquina (Machine Learning) e sobre Inteligência Artificial (IA), emerge um importante objeto de estudo, as chamadas Redes Neurais, que são modelos computacionais inspirados nas sinapses do cérebro humano e que possuem a capacidade de aprender e executar os mais variados tipos de tarefas, das mais simples às mais complexas (LeCun, Bengio & Hinton, 2015). As Redes Neurais podem ser de diversos tipos, onde em geral a principal diferença está no tipo de arquitetura e na maneira como os dados são compilados.

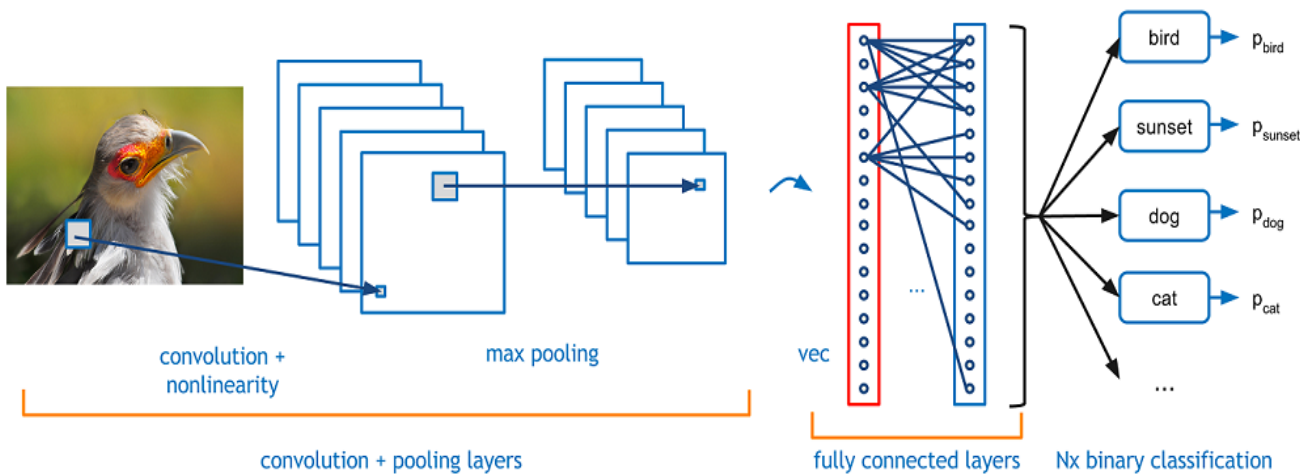
Figura 3 – Estrutura da Rede Neural Simples (MLP — Multi-Layer Perceptron)



Fonte: https://pt.wikipedia.org/wiki/Rede_neural_artificial

Dentre as principais Redes Neurais existentes, destacam-se as Redes Neurais Convolucionais (CNNs), utilizadas principalmente nas ocasiões em que se faz necessário o processamento de imagens, explorando a estrutura espacial dos dados (Rodrigues, 2019). As CNNs são comumente utilizadas para tarefas que envolvem visão computacional, como sistemas de classificação de imagens ou detecção de objetos, reconhecendo diversos padrões presentes nas imagens. Isso tudo devido à sua capacidade de detecção de características locais, como bordas, formas ou texturas (LeCun, Bengio & Hinton, 2015). Por conta disso essa técnica possui grande vantagem diante de redes neurais menos robustas.

Figura 4 – Estrutura da Rede Neural Convulucional.



Fonte: <http://endigital.orgfree.com/pci/ensolda.htm>

A Rede Neural Convulucional (CNN) é uma arquitetura de inteligência artificial composta por diversas camadas, sendo as camadas convolucionais as responsáveis por extrair as características essenciais das imagens, como bordas, texturas e formas. Já as camadas de pooling têm a função de reduzir a dimensionalidade das informações, simplificando os dados sem causar perda significativa de conteúdo relevante. Essa estrutura permite que a CNN seja altamente eficaz no processamento de imagens e na identificação de padrões complexos. Além disso, a aplicação de IA nos sistemas de CNN permite uma adaptação contínua, pois os modelos podem ser aprimorados de forma constante com a inserção de novos dados ou a utilização de técnicas de aumento de dados (Shorten & Khoshgoftaar, 2019).

3.4.1 Overfitting e underfitting

Overfitting e underfitting são dois problemas comuns no treinamento de modelos de machine learning, especialmente em redes neurais, onde ambos estão relacionados ao desempenho do modelo, aos dados de treinamento fornecido e nos dados não evidenciados no modelo, que são os dados de teste e de validação (Goodfellow, I., Bengio, Y., & Courville, A, 2016).

- **Overfitting (Superajuste):** Ocorre quando um modelo se ajusta excessivamente aos dados de treinamento, capturando não apenas os padrões principais, mas também o ruído presente nos dados. Esse fenômeno faz com que o modelo apresente um excelente desempenho nos dados de treinamento, mas perca sua capacidade de generalização ao ser avaliado em novos dados, como os conjuntos de validação ou teste.

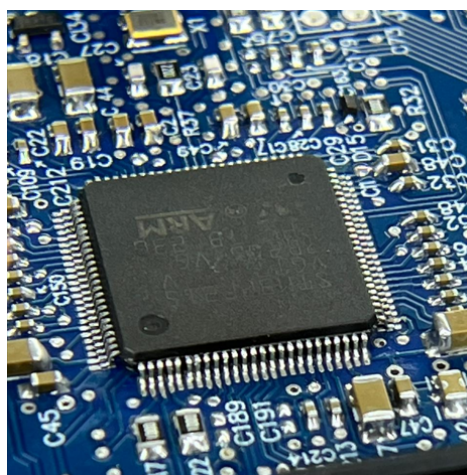
Na prática, o overfitting pode ser identificado por uma grande discrepância entre a acurácia de treinamento e a acurácia de validação ou teste. Isso ocorre porque o modelo está memorizando os dados de treinamento, em vez de aprender os padrões generalizáveis. Medidas para mitigar o overfitting incluem a utilização de mais dados de treinamento, a aplicação de regularização e a simplificação da arquitetura do modelo (Goodfellow, I., Bengio, Y., & Courville, A, 2016).

- **Underfitting (Subajuste):** Ocorre quando o modelo não é complexo o suficiente para capturar os padrões nos dados de treinamento, resultando em um desempenho baixo tanto no treinamento quanto nos dados de validação. Isso geralmente acontece quando o modelo é muito simples para o problema apresentado ou quando os dados não estão adequados ou processados de maneira correta (Goodfellow, I., Bengio, Y., & Courville, A, 2016).

3.5 Problemas de Solda

As Placas de circuito impresso são itens cruciais para a eletrônica moderna, servindo de base estrutural para conectar e suportar componentes eletrônicos em dispositivos de diferentes complexidades, de modo que sua correta inspeção é fundamental durante o processo de fabricação, sendo determinante para o funcionamento do equipamento como um todo (Blom, 2013).

Figura 5 – Placa de circuito impresso



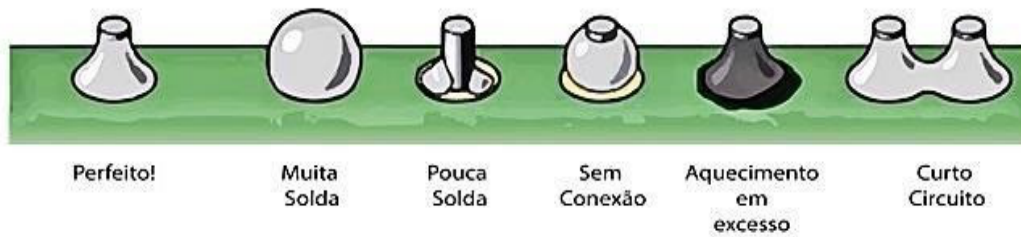
Fonte: Próprio autor

Entre os principais defeitos identificados em uma análise visual de PCIs, destacam-se problemas como:

- Soldas sem conexão: Nesses pontos, embora a solda consiga se fixar e manter-se firme na trilha, pode acontecer de não ser estabelecido um contato suficiente com o componente o que pode causar diversos problemas. Em casos em que a falha se apresenta de forma amena a mesma pode se assemelhar a um simples mal contato, já em casos mais severos a falha pode provocar a quebra total do equipamento.
- Pontos com excessos ou insuficiência de solda: Nesses pontos em especial, os riscos estão ligados à possibilidade de má fixação da solda, podendo provocar o seu deslocamento. Essa falha pode causar não só a interrupção do fluxo de corrente naquele local específico, mas originar um curto-circuito em algum outro ponto da placa.
- Pontos de curto: Os curtos-circuitos geralmente são facilmente identificados e geralmente se dão pelo contato inapropriado de um componente com outro, o que em geral pode causar vários problemas ao equipamento.
- Pontos de solda queimada: Nesses pontos, geralmente pelo aquecimento demasiado da solda ou do componente, pode haver perda das propriedades condutivas de ambos interrompendo assim o fluxo de corrente, o que pode significar mal funcionamento do equipamento.
- Soldas frias: São as junções inadequadas, oriundas da falta de calor adequado ou da insuficiência de metal de adição durante o processo de soldagem;
- Solder ball's: São os “restos” de solda que se desgrudam de um local específico da placa e circulam livremente pela mesma podendo originar um curto-circuito;

Na Figura 6, é possível observar claramente a diferença existente entre uma solda que está dentro dos parâmetros de qualidade esperados, em relação a outras soldas, que apresentam problemas diversos. O intuito é que a Inteligência Artificial desenvolvida consiga captar essas diferenças de forma precisa e automática.

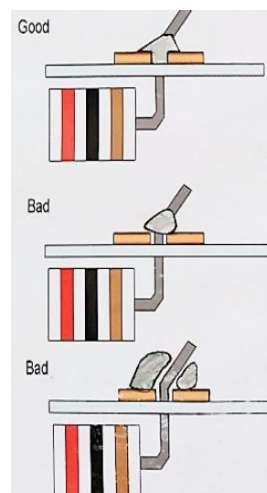
Figura 6 – Variações dos pontos de solda.



Fonte: <http://endigital.orgfree.com/pci/ensolda.htm>

Nas Figuras 7 e 8, é possível ter uma perspectiva ainda mais clara sobre como é a aparência de uma solda em boa condição se comparada a uma solda que apresenta falha.

Figura 7 – Comparação da qualidade entre soldas boas e ruins.



Fonte: <http://endigital.orgfree.com/pci/ensolda.htm>

Figura 8 – Falha de solda em placa de circuito impresso real.

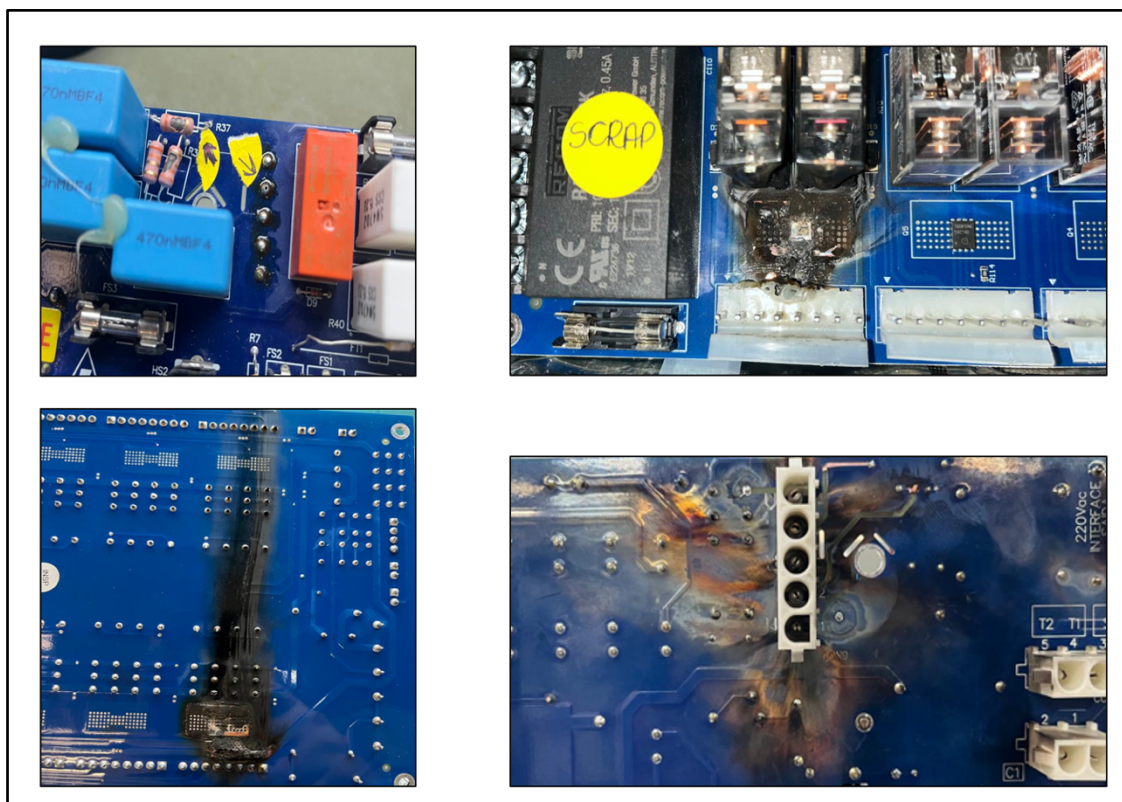


Fonte: <http://endigital.orgfree.com/pci/ensolda.htm>

Nos casos em que os defeitos de solda não são identificados de forma antecipada, o impacto final pode ser significativo, colocando em risco a segurança das pessoas que entram em contato direto com o equipamento. Falhas de soldagem podem resultar em curtos-circuitos, aquecimento excessivo ou até mesmo falhas críticas nos sistemas, comprometendo a integridade do dispositivo eletrônico e, em alguns casos, causando danos irreparáveis. Quando essas falhas não são detectadas durante a produção ou testes, a placa danificada frequentemente precisa ser substituída, o que implica em custos adicionais e possíveis atrasos na entrega do produto.

Para ilustrar melhor esses cenários, a Figura 9 apresenta exemplos visuais sobre o desfecho final de algumas dessas falhas, onde é possível identificar o derretimento de componentes ou a queima de uma parte considerável da placa.

Figura 9 – PCI's danificadas



Fonte: Próprio autor

4 METODOLOGIA

Após o correto entendimento acerca dos conceitos que envolvem as análises em placas de circuito impresso propriamente ditas, é necessário estabelecer a metodologia do trabalho e a forma como ele será desenvolvido. Todas essas informações estarão disponíveis nesse capítulo.

Para estruturar um sistema de classificação funcional, utilizando as técnicas de inteligência Artificial, existe um fluxo bem definido que pode ser seguido, composto por etapas pré-estabelecidas que vão desde o tratamento das imagens até a avaliação dos resultados obtido. Segue abaixo a metodologia utilizada para estruturação do trabalho.

4.1 Definição do Instrumento de aquisição de imagens

Nessa etapa, é necessário definir qual o instrumento a ser afixado na estrutura, ficando responsável por fazer a captação das imagens.

Os insumos levantados foram:

- **Uma câmera industrial de alta resolução.**
Prós: Poderia fornecer um detalhamento muito bom das placas e poderia ser facilmente configurada.
Limitação: A câmera precisaria estar associada a um sistema de fixação que permitisse garantir um posicionamento e uma iluminação uniforme para todas as imagens.
- **Microscópio Industrial USB**
Prós: Possui um bom custo-benefício e fornece imagens satisfatórias para o tipo de aplicação desejada.
Limitação: O Microscópio precisa estar associado a uma estação de trabalho que permitisse acesso ao software de captura, além da aquisição das imagens se dar de forma muito mais lenta.
- **Aparelho celular com resolução de imagem de 4K**
Prós: Possui imagens de qualidade satisfatória para o tipo de aplicação; não há necessidade de aquisição; pode ser implementado em sistemas de baixo custo.
Limitação: Menor robustez e maior dificuldade no enquadramento das imagens.

Embora forneça menor robustez comparado aos demais, o insumo utilizado para aquisição das imagens foi o aparelho celular. Ele possui uma câmera de boa qualidade e que em um primeiro momento seria mais do que suficiente para adquirir as imagens para o modelo.

4.2 Delimitação dos tipos de falhas

Nesse passo, foi estabelecido que as análises seriam feitas com base nos erros de soldas mais recorrentes durante a fabricação/montagem das placas de circuito impresso. São elas:

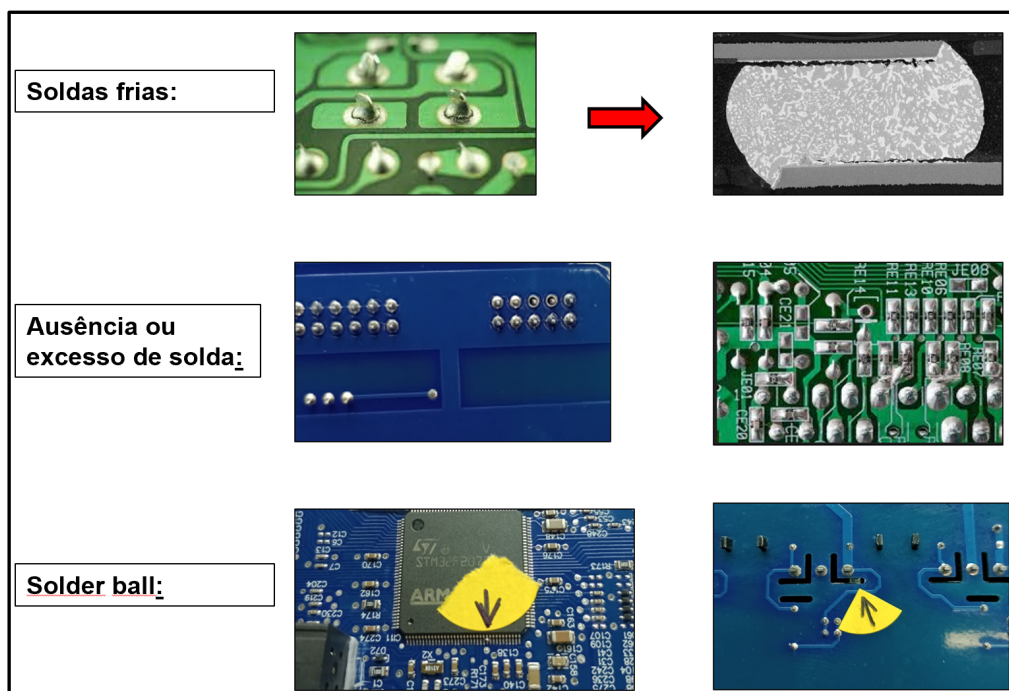
- Pontos com excessos de solda
- Pontos com insuficiência de solda
- Pontos de solda em curto
- Trilhas rompidas

Todos esses conceitos foram detalhados durante a fundamentação teórica e para delimitar quais seriam utilizados no estudo, foi realizada uma verificação visual em aproximadamente 120 placas in loco, de forma a verificar se com o método a ser aplicado, as verificações seriam eficientes.

Concluiu-se que inicialmente todas estavam aptas para passarem pela verificação.

Na figura x é possível identificar alguns dos problemas mais recorrentes encontrados.

Figura 10 – Defeitos de solda.



Fonte: Próprio autor

4.3 Pré-processamento dos Dados

O primeiro passo é carregar as imagens das placas, que foram anteriormente adicionadas a um banco de dados local (**item 4.2**) e se encontram classificadas entre aptas e não aptas.

4.3.1 Organização dos dados

As imagens foram organizadas em dois grupos, estruturados da seguinte forma:

- **Aptas** - Contém todas as imagens de placas boas para serem utilizadas, onde não foram encontrados nenhum desvio visualmente evidente.

- **Não Aptas** – Contém todas as imagens que apresentavam algum desvio construtivo e que não estavam aptas para uso, são eles:
 - Curto Circuito/
 - Solda Ruim/
 - Excesso Solda/
 - Trilhas Rompidas/

4.3.2 Redimensionamento e Normalização

Para garantir a uniformidade dos itens a serem analisados foi feito um ajuste das imagens, deixando todas com um formato fixo de 224x224 pixels.

```
"img_height = 224"  
"img_width = 224"
```

Esse ajuste pode ser alterado com base no objeto de estudo a ser analisado, onde para imagens de alta complexibilidade e padrões muito variados, definir a máscara para dimensões maiores pode ajudar a rede a convergir com mais assertividade. O intuito é que durante o treinamento da rede, sejam testados valores distintos de máscara, atestando que o valor de 224x224 é de fato o mais adequado.

Em seguida, será aplicada a normalização dos valores de pixel para o intervalo [0, 1], dividindo os valores por 255. O intuito dessa normalização é facilitar a convergência do modelo e tornar o processamento mais eficiente. A

normalização evita que valores altos ou discrepantes prejudiquem o cálculo do gradiente, aumentando a estabilidade do modelo.

```
“normalization_layer = layers.Rescaling(1./255)”
```

4.3.3 Aumentação de Dados

Para aumentar a robustez do modelo e evitar o overfitting, ou seja, a perda prematura de eficiência do sistema, foram testadas algumas métricas de aumento de dados que são responsáveis por gerar variações sutis nas imagens de forma a não alterar suas características fundamentais. Dentre as variações mais comumente empregadas é possível citar os movimentos de rotação, inversão ou simplesmente aplicação de mais ou menos Zoom na imagem.

Essa prática foi de suma importância para lidar com a quantidade limitada de imagens que estavam disponíveis para serem analisadas.

4.4 Desenvolvimento do Modelo

Para classificação das placas a escolha mais acertada foi a utilização da Rede Neural Convolutiva (CNN), que dentre todos os métodos disponíveis se mostrou mais eficaz para as tarefas que envolviam o reconhecimento de padrões visuais e a detecção de anomalias, que é o intuito fundamental deste sistema.

Dentre os recursos específicos desse tipo de rede, que foram utilizados se destacam:

- Extração automática de características
- Reconhecimento de padrões espaciais
- Redução de dimensão sem perda ativa das características
- Estabilidade e transferência ativa do conhecimento obtido a cada interação

A ideia principal é desenvolver um modelo onde possam ser conflitados parâmetros distintos, de modo a definir o mais adequado para a aplicação específica, uma vez que, se tratando de Inteligência Artificial e de Redes Neurais, até mesmo uma alta capacidade de desempenho pode repercutir em um sistema ineficiente. Aqueles casos em que ocorre Overfitting são exemplos claros disso. Outro ponto importante a se destacar é que a partir do redimensionamento de qualquer um dos Hiperparâmetros utilizados, a eficiência do modelo passa por alterações, de modo que para estabelecer o melhor resultado possível será necessário simular variâncias coordenadas, de modo a compreender melhor o funcionamento da IA.

4.4.1 Definição dos Hiper parâmetros

Para compilar o modelo de forma adequada é necessário definir algumas métricas importantes, como por exemplo, qual vai ser o otimizador utilizado, quais serão as funções de ativação, a quantidade de camadas de Pooling e a melhor métrica de avaliação, garantindo a maior eficiência possível. Para tal, serão utilizados como referência, modelos pré treinados, disponíveis na literatura e nos documentos oficiais das bibliotecas escolhidas (Tensor Flow e Keras).

- **Definição das camadas:** Inicialmente foram utilizadas 3 Camadas de Convolução com 32, 64, e 128 filtros respectivamente, isso ajudaria o modelo a extrair mais características complexas, principalmente se tratando de itens como placas de circuito impresso. Além disso foram incluídas 3 Camadas MaxPooling2D que são as camadas que sucedem as camadas de convolução e são responsáveis pela redução da dimensionalidade das características adquiridas, 1 Camada de Flattening que após a aquisição dos dados obtidos pelas camadas de convolução, suprime as informações dentro de um vetor e 2 Camadas Densas, onde a camada final consiste em uma camada densa com 128 neurônios e função de ativação Softmax, que será utilizada na camada de saída para converter logits em probabilidades, facilitando a classificação. Segue:

Figura 11 – Camadas do modelo

```
# Modelo ajustado
model = models.Sequential([
    layers.Conv2D(32, 3, padding='same', activation='relu', input_shape=(img_height, img_width, 3)),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(128, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dense(128, activation='relu'), # Camada densa com mais neurônios
    layers.Dense(len(class_names), activation='softmax') # Função softmax na saída
])
```

Fonte: Próprio autor

No decorrer dos treinamentos, serão utilizadas outras configurações de camadas com quantidades de neurônios diferentes em cada uma delas, de forma a garantir que o custo computacional é de fato a mais adequado ao problema apresentado. Além disso é esperado que o sistema se mostre o mais eficiente possível, tendo em vista que além da assertividade na identificação dos defeitos, reduzir o tempo de resposta é primordial para aplicações industriais.

- **Função de ativação:** Para esse modelo serão feitos testes utilizando as 3 funções de ativação mais indicadas para sistemas desse tipo. São elas:

- **ReLU:** Geralmente convergem mais rapidamente e apresentam melhor desempenho em tarefas de visão computacional, como reconhecimento de objetos, classificação de imagens, etc.

- **Sigmoide:** Boa para classificações binárias, mas sofre com a saturação em valores extremos (próximo de 0 e 1), resultando em gradientes muito pequenos.

- **Tanh:** Tem uma curva mais acentuada em relação à sigmoide, mas ainda sofre com o problema de desvanecimento do gradiente para valores extremos.

- Muito útil para entradas que têm valores negativos.

- **Otimizador:**

O modelo foi compilado utilizando o otimizador **Adam** que é um dos otimizadores mais usados no treinamento de redes neurais. Ele combina as vantagens de diferentes algoritmos para ajustar dinamicamente a taxa de aprendizado de cada parâmetro, tornando o processo de otimização mais eficiente. O otimizador Adam é geralmente a melhor escolha nos problemas de aprendizado profundo devido à sua capacidade de convergir rapidamente.

- **Função de perda:**

A função de perda que se mostrou mais eficiente para esse sistema foi a **SparseCategoricalCrossentropy**, que é perfeita para problemas que envolvem Multi-Classes, onde o modelo precisa aprender a associar cada entrada (imagem) a uma única classe, dentre várias disponíveis, funcionando muito bem com a função Softmax, no cálculo das discrepâncias geradas pelo modelo em relação à imagem real.

Outro ponto positivo é que diferentemente de outras funções de perda, ela utiliza Rotulos inteiros (Sparse) o que simplifica o processamento dos rótulos e economiza memória deixando a compilação mais limpa e ágil.

4.5 Treinamento do Modelo

Para o treinamento do modelo será realizada a divisão dos dados, onde 80% dos dados serão disponibilizados para treinamento enquanto os outros 20% ficarão reservados para a validação, garantindo que o modelo não tenha acesso total aos dados de teste durante o treinamento. Segue:

Figura 12 – Divisão dos dados

```
# Carregar e dividir os dados (80% treinamento e 20% validação)
train_ds = tf.keras.utils.image_dataset_from_directory(
    data_dir,
    validation_split=0.2,
    subset="training",
    seed=123,
    image_size=(img_height, img_width),
    batch_size=batch_size
)

val_ds = tf.keras.utils.image_dataset_from_directory(
    data_dir,
    validation_split=0.2,
    subset="validation",
    seed=123,
    image_size=(img_height, img_width),
    batch_size=batch_size
)
```

Fonte: Próprio autor

Será utilizado também o método de **lote** (batch size) com o intuito de definir o número de amostras, imagens, nesse caso, que serão processadas antes da atualização dos pesos.

O tamanho de batch inicial foi definido em 16 imagens e o treinamento será realizado em um **Range de 50 épocas**, de modo a identificar ao longo desse período, os momentos em que o aprendizado se deu de forma mais efetiva.

```
“epochs = range (0,51)”
```

4.5.1 Testes de Desempenho

Para o treinamento, a biblioteca escolhida foi Tensorflow.keras, que já faz de forma automática o monitoramento de algumas métricas de desempenho, como **acurácia** e **perda** tanto para o conjunto de treinamento quanto para o conjunto de validação. O progresso do modelo pode ser acompanhado a partir de gráficos simples, responsáveis por comparar os resultados ao longo das épocas.

Esses gráficos serão utilizados para elucidar o comportamento da rede e a forma como os dados se dispõem, o que será de grande valia para o refino do sistema.

4.6 Avaliação e Métricas de Desempenho

Após treinado, o modelo passará por uma etapa de validação final, onde serão utilizadas algumas métricas de avaliação de desempenho adicionais que vão ajudar a quantificar as informações, tendo em vista que a Acurácia em si não se refere a um valor percentual real e sim a uma média ponderada, calculada com base nos índices de aprendizado da rede.

- Precisão, Recall e F1-score

Essas métricas serão calculadas para cada uma das classes, fornecendo uma visão detalhada do desempenho do modelo em termos de:

- **Precisão:** Percentual de previsões corretas entre todas as previsões feitas para uma classe.
- **Recall:** Capacidade do modelo de identificar corretamente todas as instâncias reais de uma classe.

5 RESULTADOS

Nesse tópico, serão abordados os resultados obtidos a partir da compilação dos dados pelo modelo de Rede Neural Convolutiva (CNN) desenvolvido, que tinha como intuito identificar e classificar falhas construtivas em placas de circuito impresso.

O treinamento da Rede foi realizado utilizando um conjunto de imagens previamente adquiridas e rotuladas, conforme o tipo de defeito que apresentavam.

Foram utilizados 80% dos dados adquiridos para o treinamento da rede e separados 20% para o processo de validação.

A avaliação de desempenho e a definição do melhor cenário de trabalho foram definidos a partir de testes comparativos, feitos com base em vários **SETUPS de Hiper Parâmetros distintos**.

Os resultados obtidos ao longo dos testes, refletem uma análise cuidadosa sobre o impacto de diferentes configurações de hiperparâmetros no desempenho de uma rede neural aplicada à classificação de placas de circuito impresso.

Foram simulados 12 cenários distintos, variando a quantidade de camadas, o número de neurônios em cada camada e as funções de ativação (Sigmoid, Tanh e ReLU). Para garantir uma comparação justa entre os cenários, todas as simulações foram realizadas considerando 50 épocas de treinamento, o que permitiu identificar padrões consistentes no comportamento do modelo.

É importante salientar que utilizar um número largo de épocas ajuda a visualizar de forma mais concisa até qual época o modelo realmente apresenta ganhos significativos de desempenho e em que ponto o treinamento se torna redundante (indicando overfitting).

OBS: Essa abordagem oferece insights fundamentais para otimizar futuros experimentos, economizar recursos computacionais e principalmente reduzir o tempo de trabalho.

Em relação à quantidade de camadas e de neurônios, foram escolhidos 4 cenários distintos (1T, 2T, 3T e 4T), onde o intuito é avaliar se a quantidade de camadas escolhida corresponde ao nível de complexibilidade do sistema. Os arranjos das camadas testadas foram:

-----1T-----

```
# Modelo ajustado
model = models.Sequential([
    layers.Conv2D(8, 3, padding='same', activation='relu', input_shape=(img_height, img_width, 3)),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dense(16, activation='relu', kernel_regularizer=l2(0.01)), # Camada densa com mais neurônios
    layers.Dense(len(class_names), activation='softmax') # Função softmax na saída
])
```

-----2T-----

```
# Modelo ajustado
model = models.Sequential([
    layers.Conv2D(16, 3, padding='same', activation='relu', input_shape=(img_height, img_width, 3)),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dense(32, activation='relu', kernel_regularizer=l2(0.01)), # Camada densa com mais neurônios
    layers.Dense(len(class_names), activation='softmax') # Função softmax na saída
])
```

-----3T-----

```
# Modelo ajustado
model = models.Sequential([
    layers.Conv2D(16, 3, padding='same', activation='relu', input_shape=(img_height, img_width, 3)),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dense(64, activation='relu', kernel_regularizer=l2(0.01)), # Camada densa com mais neurônios
    layers.Dense(len(class_names), activation='softmax') # Função softmax na saída
])
```

-----4T-----

```
#model = models.Sequential([
    layers.Conv2D(32, 3, padding='same', activation='relu', input_shape=(img_height, img_width, 3)),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(128, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dense(128, activation='relu'), # Camada densa com mais neurônios
    layers.Dense(len(class_names), activation='softmax') # Função softmax na saída
])
```

Nas **Tabelas 2, 3 e 4** foram dispostos os resultados obtidos após simular os 12 cenários de configuração existentes, que perpassam pelas 3 funções de ativação e os 4 arranjos de camadas propostos anteriormente.

Estão sendo armazenados alguns dados importantes, como por exemplo os Valores de acurácia de treinamento e de validação, o número de épocas necessário para o máximo aprendizado, o tempo de execução e a quantidade de épocas necessárias para saturar o aprendizado.

Tabela 2 - Simulação Sigmoid

Cenário das camadas	Acurácia de treinamento	Acurácia MAX de Validação	Épocas para o MAX aprendizado (Treino)	Houve saturação do aprendizado?	Tempo de execução
1T	0.547	0.75	8	N	23.0s
2T	0.5431	0.72	11	N	53.5s
3T	0.544	0.74	7	N	1m 7.6s
4T	0.506	0.74	9	N	3m 47s

Fonte: Próprio autor

Tabela 3 - Simulação Tanh

Cenário das camadas	Acurácia de treinamento	Acurácia MAX de Validação	Épocas para o MAX aprendizado (Treino)	Houve saturação do aprendizado?	Tempo de execução
1T	1	0.9	6	S	19s
2T	1	0.95	17	S	40.4s
3T	0,98	0,89	36	N	57.7s
4T	1	0,95	14	S	3m 27s

Fonte: Próprio autor

Tabela 4 - Simulação Relu

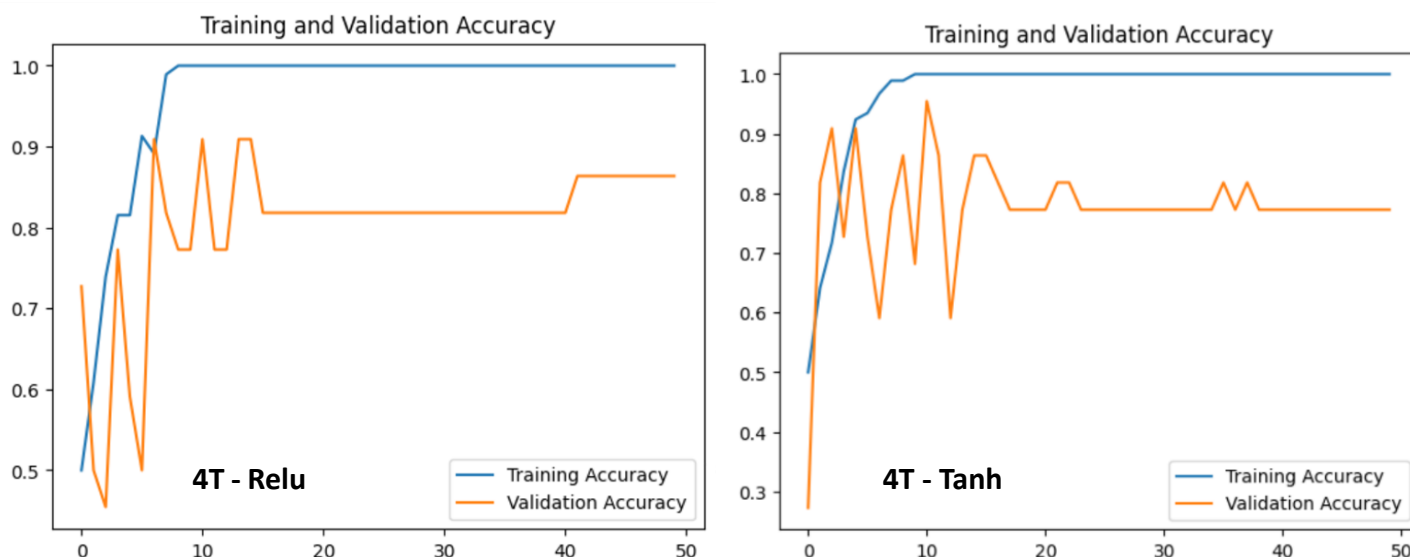
Cenário das camadas	Acurácia de treinamento	Acurácia MAX de Validação	Épocas para o MAX aprendizado (Treino)	Houve saturação do aprendizado?	Tempo de execução
1T	1	0.85	11	S	20.8s
2T	1	0.91	6	S	51.2s
3T	1	0.9	8	S	1m 40s
4T	1	0.91	9	S	3m 35s

Fonte: Próprio autor

Análise comparativa dos resultados

Os cenários analisados revelaram diferenças significativas nos valores de acurácia de treinamento e acurácia de validação. Redes com mais camadas e maior número de neurônios, como simulado nos cenários 4T (Funções Relu e Tanh), apresentaram uma elevada acurácia de treinamento, devido à sua maior capacidade de aprendizado. No entanto, essas configurações também demonstraram maior propensão ao **overfitting**, ou seja, o modelo aprendeu de forma tão detalhada e rápida os padrões do conjunto de treinamento que acabou perdendo a capacidade de generalizar para os dados de validação. Isso pode ser observado com mais clareza nos gráficos de acurácia abaixo.

Figura 13 – Simulações cenário 4T

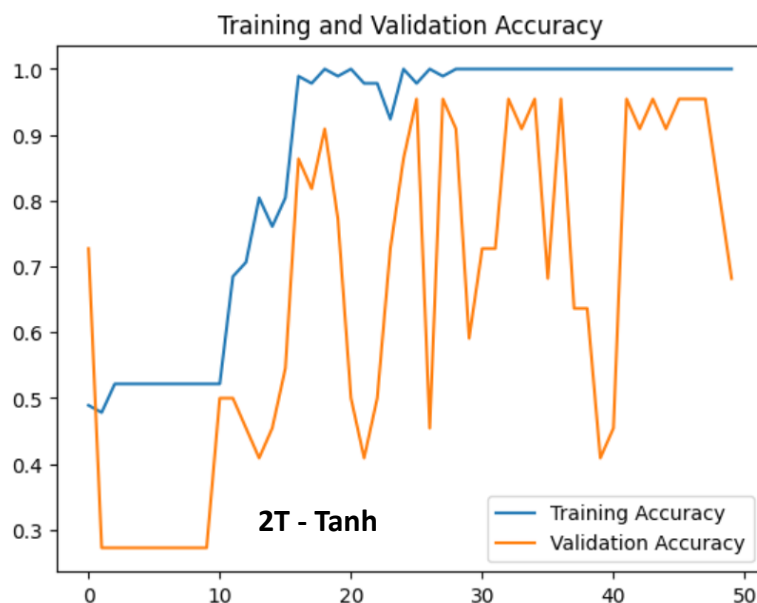


Fonte: Próprio autor

Uma boa prática para minimizar essa carência de generalização e fazer com que a rede aprenda de forma mais lenta é reduzir sua capacidade operacional, diminuindo a quantidade de camadas e seus respectivos neurônios, o que foi experimentado nas simulações onde foram utilizados os cenários 1T e 2T.

Na simulação onde a função de ativação utilizada foi a função Tanh, isso se deu de forma bem mais evidente, marcado pelo deslocamento da curva de treinamento (azul), que teve um crescimento mais equilibrado nas primeiras 20 épocas.

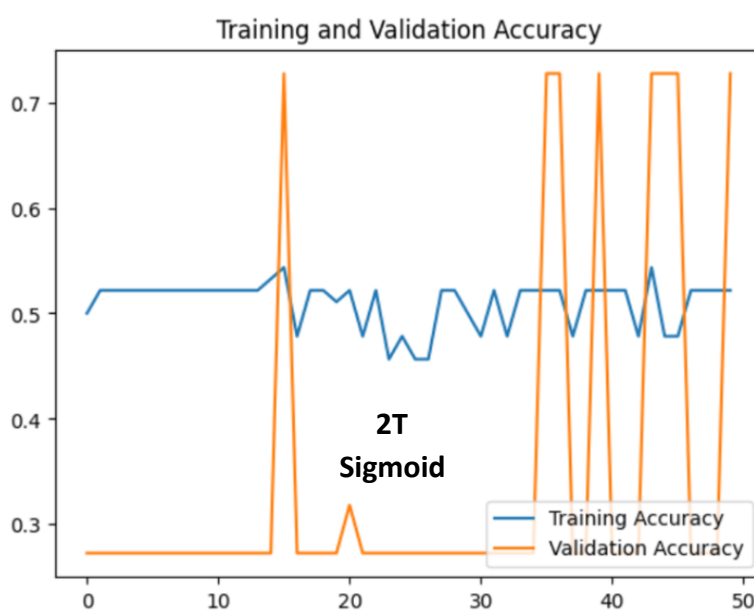
Figura 14 – Simulação Tanh cenário 2T



Fonte: Próprio autor

Quando essa estratégia é utilizada associada a uma função de ativação não tão adequada, como foi o caso da Função Sigmoid, em camadas intermediárias, o sistema acaba resultando em **underfitting**, evidenciado por acurácias mais baixas tanto no treinamento quanto na validação. Isso indica que a rede não conseguiu assimilar de forma eficiente, as características das placas e pode ser observado no gráfico abaixo.

Figura 15 – Simulação Sigmoid cenário 2T



Fonte: Próprio autor

Outro fator importante é que devido à complexidade das informações que podem ser identificadas nas placas de circuito impresso, pode necessário usar lotes maiores de amostras e aumentar ainda mais o número de épocas, para evitar um aprendizado insuficiente ou a evolução prematura da rede, como pôde ser observado anteriormente.

Épocas para o Máximo Aprendizado

Analisando as tabelas e os gráficos de desempenho, gerados para as 50 épocas, é possível definir quantas interações são necessárias para atingir o máximo aprendizado antes que o modelo sature.

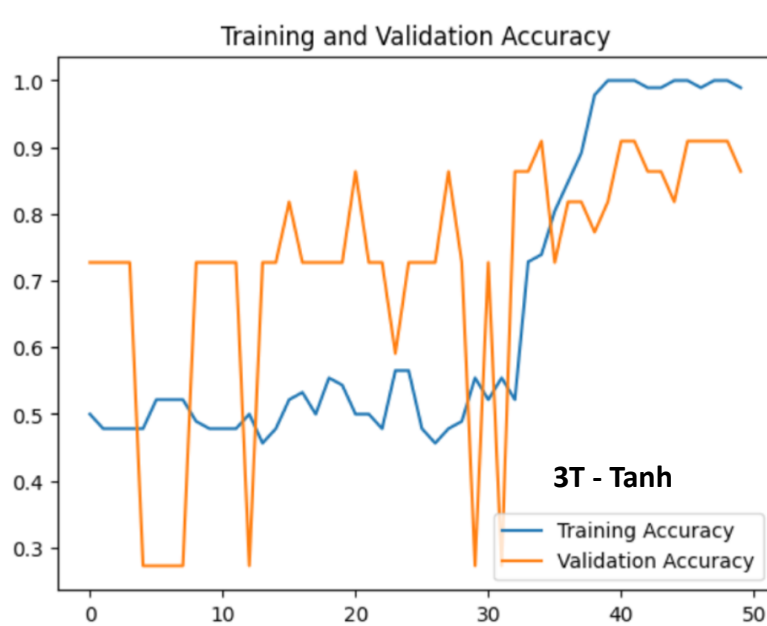
Cenários com a função de ativação ReLU e redes moderadamente profundas (aquelas com 3 ou 4 camadas) em geral, alcançaram o máximo aprendizado antes mesmo de 10 épocas.

Após esse período, observou-se uma estabilização da acurácia de treinamento e em alguns casos, um aumento das perdas no conjunto de validação, indicando overfitting.

Esse cenário ainda se mostrou melhor que os cenários obtidos para a função de ativação Sigmoide por exemplo, que além de aprender de forma rápida, teve uma curva de crescimento quase que irrelevante, indicando sua ineficiência.

O Modelo treinado com a função de ativação Tahn por outro lado, se mostrou bastante interessante, principalmente para o cenário 3T, onde obteve um crescimento constante ao longo das épocas e apesar de apresentar um pouco de ruído, demonstrou uma menor propensão ao overfitting, atendendo muito bem à demanda apresentada.

Figura 16 – Simulação Tanh cenário 3T



Fonte: Próprio autor

Tempo de Execução

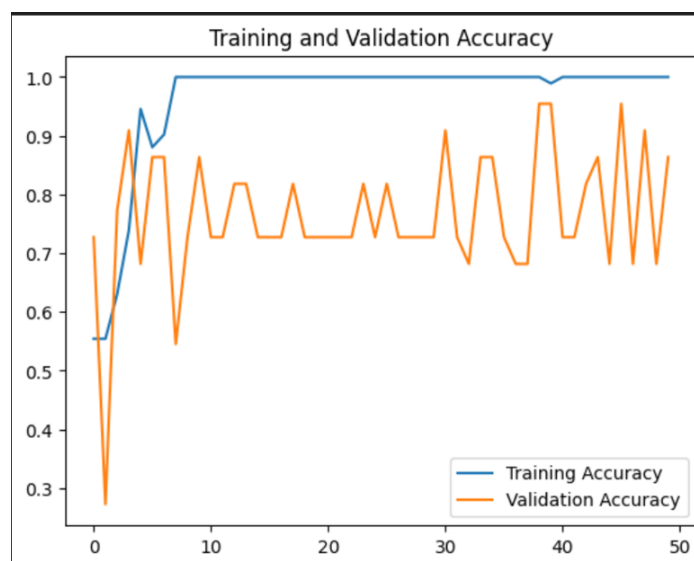
O tempo de execução foi diretamente proporcional à complexidade do modelo. Redes com maior número de camadas e neurônios consumiram mais tempo para serem treinadas, enquanto configurações mais simples foram significativamente mais rápidas. Essa análise foi essencial para justificar a escolha de configurações intermediárias, que oferecem um equilíbrio entre desempenho e eficiência computacional.

Quantidade de Épocas para Saturação

Um aspecto crucial monitorado durante o treinamento, foi o número de épocas necessárias para que o modelo atingisse a saturação no aprendizado. Em redes neurais bem configuradas, com arquitetura apropriada e funções de ativação adequadas, espera-se um aprendizado progressivo e consistente, com a diferença entre a acurácia de treinamento e a acurácia de validação permanecendo mínima. Esse equilíbrio indica que o modelo está generalizando bem, sem superajustar os dados de treinamento ou apresentar desempenho insuficiente nos dados de validação.

Alguns dos resultados obtidos sugerem uma condição sub-ótima, onde a saturação de treinamento também se dá de forma bem precoce, enquanto que os resultados de validação se mostram bem atribulados. Essas condições podem demonstrar limitações estruturais do modelo ou indicar que os dados utilizados para treinamento não são suficientes para um aprendizado completo, onde a probabilidade de aparecerem novas características ainda não assimiladas é bem grande e caso a rede não se adapte, pode tender ao overfitting. Isso pode ser observado de forma mais sutil no gráfico abaixo.

Figura 17 – Simulação validação sub-ótima



Fonte: Próprio autor

Entendimento final dos testes

Com base nesses resultados, é possível inferir que o melhor cenário para a situação proposta no texto é a utilização da função de ativação Tahn no cenário 3T. Essa configuração mostrou-se eficiente ao atingir acurácias satisfatórias de treinamento com o passar das épocas, reduzindo o risco de overfitting e uma acurácia de validação que acompanhou esse progresso. Além disso esse cenário manteve um tempo de execução aceitável e um equilíbrio excelente entre processamento e recurso disponibilizado.

Por outro lado, é possível inferir que para modelos mais complexos, a função Relu provavelmente seria a escolha certa, principalmente naqueles cenários onde é necessária uma boa distribuição das camadas intermediárias, com um número moderado de neurônios. Aliado a isso, destaca-se a sua capacidade de evitar problemas de gradiente, que acabam sendo muito comuns nas funções sigmoide e Tahn.

Validação prática do Sistema

Com o objetivo de avaliar o desempenho do sistema desenvolvido, foi estruturado um cenário de testes, responsável por averiguar se o modelo era capaz de classificar corretamente as placas de circuito impresso como aptas ou não aptas para uso. Essa etapa foi essencial para validar o funcionamento do sistema em situações que simulam condições reais de uso, indo além da análise baseada exclusivamente em métricas de treinamento e validação. A proposta visou testar o modelo com novos dados e comprovar sua capacidade de generalização. Os resultados obtidos foram adquiridos em dois formatos distintos, permitindo uma análise mais detalhada dos mesmos.

- **Relatório de classificação:** Nesse formato, após análise total das imagens fornecidas, o sistema retorna um relatório completo das imagens processadas. Esse relatório inclui informações como o nome do arquivo da imagem, o status (apta ou não apta) e a classificação atribuída pelo modelo. Essa abordagem permitiu documentar e organizar os resultados de maneira clara, facilitando a análise e a identificação de padrões nos acertos e erros do sistema. Segue:

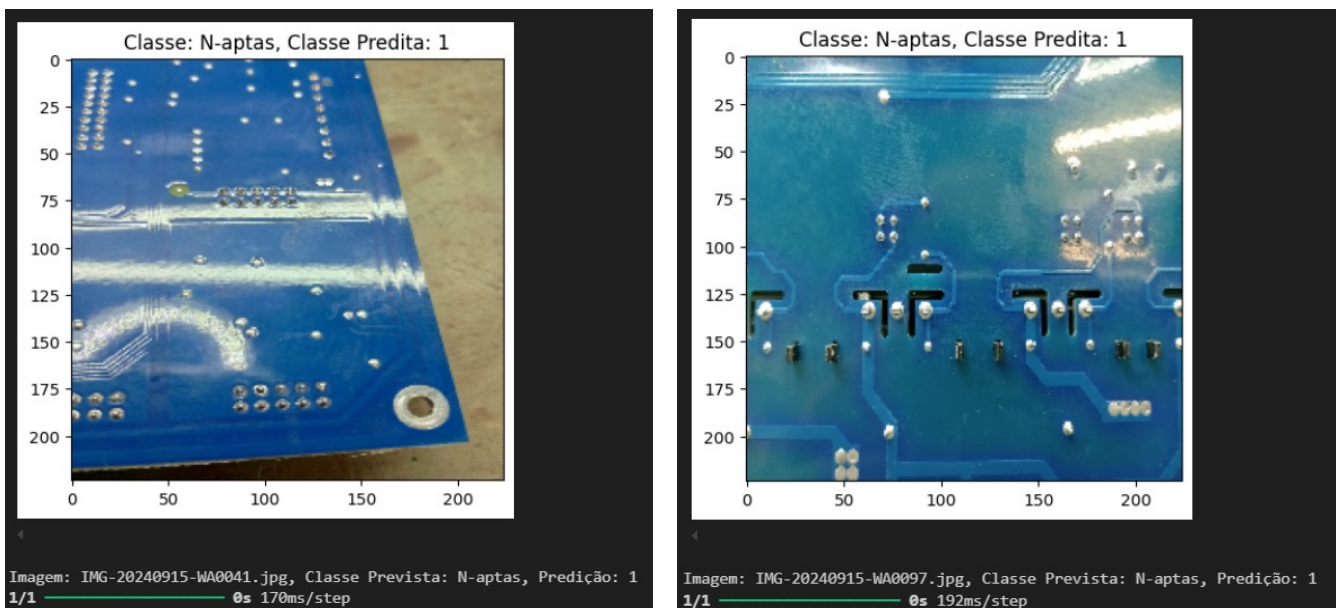
- **Rotulagem individual:** Além do relatório de classificação geral que foi utilizado, foi implementado uma função para exibir cada uma das imagens processadas juntamente a um rótulo único (apta ou não apta). Esse método foi particularmente útil para análises qualitativas, permitindo que os resultados fossem avaliados de forma mais intuitiva e direta, especialmente em relação à consistência visual da classificação realizada pelo sistema.

Figura 20 – Rotulagem individual Aptas



Fonte: Próprio autor

Figura 21 – Rotulagem Individual N-Aptas



Fonte: Próprio autor

Para executar essas validações, foi estruturado um novo código em Python/Jupyter, projetado exclusivamente para simular o sistema de predição a ser implementado na realidade. Esse código utilizou do modelo previamente treinado, declarando-o como uma espécie de biblioteca, juntamente às outras bibliotecas utilizadas.

Para os testes, foram adicionadas ao banco de dados existente, novas imagens, garantindo que o modelo fosse testado com dados desconhecidos, permitindo uma avaliação mais precisa de sua capacidade de generalização e robustez.

Os resultados obtidos foram promissores, superando as expectativas iniciais baseadas na acurácia alcançada durante o treinamento. O número de classificações corretas foi superior ao previsto, evidenciando a capacidade do modelo em identificar padrões relevantes nos dados e aplicá-los de forma eficaz em testes práticos. Contudo, algumas classificações equivocadas ainda foram observadas, o que é esperado em sistemas que demandam maior aprendizado e refinamento. Esses erros ressaltam a importância de continuar aprimorando o modelo, seja por meio do aumento e diversificação do conjunto de dados, do ajuste e otimização dos hiperparâmetros ou, ainda, pela exploração de arquiteturas mais avançadas e robustas.

Por fim, os resultados obtidos reforçam o potencial prático do sistema desenvolvido e sua viabilidade para aplicação em inspeções automatizadas de placas de circuito impresso. Apesar das limitações, o modelo demonstrou uma capacidade significativa de classificar corretamente as imagens, representando um avanço relevante e servindo como base para futuras melhorias e implementações.

6 CONCLUSÃO

O presente trabalho teve como objetivo principal desenvolver um sistema de classificação para placas de circuito impresso, que fosse capaz de identificar padrões e rotular as placas como aptas ou não aptas para uso, isso tudo a partir da estruturação de uma Rede neural do tipo convolucional e das técnicas de aprendizado profundo. Ao longo do estudo, foi possível explorar conceitos fundamentais de machine learning, como overfitting, underfitting, funções de ativação, otimização de hiperparâmetros e a estruturação das redes neurais convolucionais propriamente ditas. A análise crítica dos resultados obtidos e a implementação de ajustes na arquitetura e nos parâmetros do modelo proporcionaram insights valiosos sobre os desafios e as possibilidades dessa abordagem. A partir das simulações realizadas, ficou evidente que a escolha adequada dos hiperparâmetros desempenha um papel essencial no desempenho da rede.

A função de ativação ReLU destaca-se bastante por sua eficiência em melhorar o aprendizado das redes convolucionais, ainda mais aliado a arquiteturas intermediária, com 3 ou 4 camadas e um número de neurônios consistente, oferecendo um equilíbrio muito interessante entre a complexidade do modelo e eficiência computacional oferecida. Entretanto, o número limitado de amostras que foi fornecido e a complexibilidade mediana de interpretação exigidas para o sistema proposto, favoreceram muito a utilização da função de ativação Tahn que devido as suas características fundamentais, teve uma menor propensão ao overfitting que é extremamente prejudicial em sistemas classificatórios como o mesmo.

A análise detalhada dos gráficos de acurácia e perda permitiu identificar a quantidade ideal de épocas para treinamento, otimizando o uso de recursos e reduzindo o risco de overfitting. No entanto, o estudo também evidenciou limitações significativas, principalmente relacionadas à quantidade reduzida de dados de treinamento, que restringiu a capacidade do modelo de generalizar para novos dados. Esse fator contribuiu para o surgimento de overfitting em alguns cenários, demonstrando a importância de ampliar o conjunto de dados e adotar estratégias de aumento de dados, em estudos futuros. Além disso, a exploração de técnicas avançadas, como validação cruzada e regularização, poderiam contribuir muito para melhorar a robustez do modelo.

O trabalho mostrou-se bem-sucedido ao demonstrar de forma efetiva a aplicabilidade das redes neurais para atividades de inspeção automatizada como um todo, fornecendo uma base sólida para o desenvolvimento de soluções práticas na área, bem como foi demonstrado durante ao longo do texto. Os resultados alcançados, embora promissores, abrem caminho para novas pesquisas e diversas melhorias, como o teste de arquiteturas mais complexas, a implementação de novos métodos de otimização e a avaliação do desempenho em um cenário real de produção.

Por fim, o estudo reforça a relevância da visão computacional e do aprendizado profundo como ferramentas indispensáveis para a automação industrial e melhoria contínua dos processos industriais, destacando seu potencial em áreas que demandam precisão, eficiência e análise de grandes volumes de dados.

Recomendações para Trabalhos Futuros

- **Análise Detalhada das Características Aprendidas:** Investigar o que o modelo está realmente aprendendo, por meio de técnicas como visualização de camadas intermediárias ou mapas de ativação (Grad-CAM), ajudaria a entender se a rede foca nas características corretas das placas. Esse tipo de análise pode revelar possíveis melhorias para o ajuste dos filtros e camadas.
- **Uso de estratégias de otimização automática de hiperparâmetros,** como busca em grade ou algoritmos genéticos
- **Aplicação de técnicas avançadas para reduzir o overfitting,** como aumento de dados, regularização por dropout e validação cruzada.
- **Implementação de Técnicas de Aumento de Dados (Data Augmentation):** Para combater o overfitting, seria útil aplicar técnicas de data augmentation, como rotações, zoom, cortes, ou ajustes de brilho e contraste, que permitiriam ao modelo ver diferentes versões de cada imagem e aprender características mais robustas e generalizáveis.

REFERÊNCIAS

- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 10520**: Informação e documentação - apresentação de citações em documentos. Rio de Janeiro, 2002. 1-7 p. Citado na página 17.
- BRAGA, I. N. C. **A História do transistor**. 2023. Disponível em: <<https://www.newtoncbraga.com.br/index.php/artigos/67-historia-daeletronica/407ahistoria-do-transistor.html>>. Citado 2 vezes nas páginas 14 e 15.
- Blom, H. A. P. (2013). **Automated optical inspection (AOI) systems for printed circuit board assemblies (PCBA)**. In: Integrated Circuit and System Design (pp. 33-48). Springer.
- CABELLO, F. A. C. **Contribuições à detecção de falhas mecânicas sobre placas de circuito impresso usando processamento de imagens e o algoritmo SIFT**. Dissertação (Mestrado) — Unicamp, 2013. Citado na página 15.
- CROMATEK. **Montagem de PCB**. 2023. Disponível em: <<https://cromatek.com.br/montagem-de-pcb/>>. Citado 2 vezes nas páginas 14 e 15.
- ELFES, A. Using occupancy grids for mobile robot perception and navigation. **Computer**,
IEEE, v. 22, n. 6, p. 46–57, 1989. Disponível em:
<<https://doi.org/10.1109/2.30720>>. Citado na página 17.
- Gonzalez, R. C., & Woods, R. E.** (2008). *Digital Image Processing* (3rd ed.). Pearson Prentice Hall.
- JESUS, C. B.; ORELLANA, E. T. V. Detecção de falhas na montagem de pcbs: Utilizando técnicas de processamento paralelo para otimizar o desempenho das ferramentas de visão computacional em computadores de placa Única. **VI Seminário de Avaliação de Pesquisa Científica e Tecnológica SENAI CIMATEC - 2021**, Bahia, 2021. Citado na página 15.
- Kumar, S., & Jain, V. (2019). **Machine learning for automated optical inspection of printed circuit boards**: A review. *Journal of Manufacturing Processes*, 42, 134-144.

LANGUAGES, O. **Paperback Oxford English Dictionary**. USA: Oxford University Press, USA, 2013. ISBN 0199640947. Citado na página 14.

MEELO, P. R. d. S.; RIOS, E. C. S. D.; GUTIERREZ, R. M. V. Placas de circuito impresso, mercado atual e perspectivas. **Banco Nacional de Desenvolvimento Econômico e Social**, Rio de Janeiro, 2001. Citado na página 14.

OGATA, K. **Engenharia de Controle Moderno**. São Paulo: Person Prentice Hall, 2010. ISBN 8576058103. Citado 2 vezes nas páginas 17 e 19.

Russell, S., & Norvig, P. (2016). **Artificial Intelligence: A Modern Approach (3rd ed.)**. Prentice Hall.

Szeliski, R. (2010). **Computer Vision: Algorithms and Applications**. Springer.

VASCONCELLOS, A. B. d.; CAMOLESI, S. C.; SILVA, L. O. da; ANABUKI, E. T.; MALHEIRO, T. I. R. C. Análise da Migração de uma Consumidora para o Mercado Livre de Energia. **IV Simpósio Brasileiro de Sistemas Elétricos (SBSE)**, 2012. Citado na página 17.

BRADSKI; KAEHLER, **Learning OpenCV**. O'Reilly Media, 28 outubro 2008 ISBN-13: 978-1491937990

SZELISKI, **Computer Vision: Algorithms and Applications**. Springer, 19 outubro 2010 ISBN-13: 978-1848829343

Géron, A. (2019). **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow**. O'Reilly Media.
ISBN-13: 978-1492032649

LI, Z., XIE, Y., PU, S., WANG, J., GONG, C., ZHANG, Z. (2019). **HDFace: An RGB-D Based**

HighDefinition 3D Face Imaging System. *IEEE Transactions on Multimedia*, 21(5), 1154-1166.

ZHANG, X., LU, X., & FU, L. (2018). **Image-Based Defect Detection for Printed Circuit Boards Using Convolutional Neural Networks**. *IEEE Transactions on Industrial Informatics*, 14(9), 4099-4107.

CHEN, J., & LI, H. (2020). **Automated Visual Inspection of PCB Defects Using Deep Learning: A Survey**. *Journal of Manufacturing Systems*, citado nas páginas 47-64.

RODRIGUES (2019) **Conhecendo a visão do computador: Redes Neurais Convolucionais | by Vitor Rodrigues | Medium**

SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., SUTSKEVER, I., & SALAKHUTDINOV, R. (2014). ***Dropout: A Simple Way to Prevent Neural Networks from Overfitting***. *Journal of Machine Learning Research*, 15, 1929-1958.

GOODFELLOW, IAN; BENGIO, YOSHUA; COURVILLE, AARON. ***Deep learning***. Cambridge: MIT Press, 2016. Disponível em: <https://www.deeplearningbook.org>. Acesso em: 20 out. 2024.