

Utilizando Redes Neurais Convolucionais para Detecção de Defeitos Físicos em Módulos de Memória na Indústria 4.0

Prof. Dr. Fabio Santos da Silva

Departamento de Computação
Universidade do Estado do Amazonas (UEA) – Manaus – AM – Brasil
fssilva@uea.edu.br

Abstract. *This article proposes the use of Convolutional Neural Networks (CNNs) for the detection of physical defects in memory modules. Three canonical CNN architectures were evaluated: VGG19, MobileNet, and InceptionV3. Performance was measured using the accuracy metric and the defect detection model that showed the best performance, with an accuracy of approximately 98.12%, was the MobileNet model. This study provides a solid foundation for the development of automated defect detection systems for the information technology industry and contributes to the implementation of Industry 4.0.*

Resumo. *Este artigo propõe o uso de Redes Neurais Convolucionais (CNNs) para a detecção de defeitos físicos em módulos de memória. Foram avaliadas três arquiteturas de CNNs canônicas: VGG19, MobileNet e InceptionV3. O desempenho foi medido utilizando a métrica de acurácia e o modelo de detecção de defeitos que apresentou o melhor desempenho, com uma acurácia de aproximadamente 98,12%, foi o modelo da MobileNet. Este estudo fornece uma base sólida para o desenvolvimento de sistemas de detecção de defeitos automatizados para a indústria de tecnologia da informação e contribui para aplicar a indústria 4.0.*

1. Introdução

A automatização de processos de inspeção visual tem sido um tema de pesquisa crescente na indústria de tecnologia da informação, pois permite aumentar a eficiência e precisão, reduzir custos e melhorar a qualidade dos produtos. No caso específico da indústria de módulos de memória, a detecção de defeitos físicos é um processo crítico e atualmente é realizado de forma manual por um ser humano. Essa abordagem tem várias desvantagens, como a possibilidade de erros humanos, a fadiga dos operadores e a baixa velocidade de processamento.

Para superar esses desafios, tem-se recorrido ao uso de Inteligência Artificial (IA) para automatizar esses processos. Em particular, as Redes Neurais Convolucionais (CNNs) têm mostrado uma excelente capacidade para a detecção de defeitos em imagens. Alguns exemplos de trabalhos que já foram publicados são: "*Printed Circuit Board Defect Detection Using Deep Learning via A Skip-Connected Convolutional Autoencoder*" [Kim J. et al., 2021] e "*PCB Electronic Component Defect Detection Method based on Improved YOLOv4 Algorithm*" [Xin, H. et al., 2021].

Esse artigo apresenta um estudo que avaliou o desempenho de três arquiteturas canônicas de CNNs: VGG19, MobileNet e InceptionV3 para a detecção de defeitos físicos em imagens de módulos de memória. O objetivo é fornecer uma base sólida para

o desenvolvimento de sistemas de detecção de defeitos automatizados para a indústria de tecnologia da informação e contribuir para aplicar a indústria 4.0.

Esse artigo está estruturado da seguinte forma: a segunda seção, Revisão da literatura, revisa os estudos existentes relacionados ao tema, especialmente sobre o uso de CNNs e outras técnicas de IA para automatizar a detecção de defeitos em imagens. A terceira seção, Métodos, descreve detalhadamente os métodos utilizados no estudo, incluindo a descrição do conjunto de dados, método de detecção de defeitos proposto e as ferramentas utilizadas. A quarta seção, Resultados, apresenta e discute os resultados obtidos, incluindo a comparação do desempenho das diferentes arquiteturas canônicas de CNNs avaliadas. A quinta seção, Conclusão, sintetiza os principais resultados e suas implicações, além de sugestões para futuros estudos e possíveis aplicações na indústria. Por fim, a sexta seção, Referências, apresenta a lista de referências bibliográficas utilizadas no artigo.

2. Revisão da literatura

Essa seção versa sobre Inspeção Visual de Produtos Baseada em Inteligência Artificial, o uso de CNNs, a Abordagem de Aprendizagem de Máquina Supervisionada e as técnicas pré-processamento que foram empregadas nesse trabalho para melhorar o desempenho dos modelos como *Data Augmentation* e Dropout.

2.1. Redes Neurais Convolucionais (CNNs)

As Redes Neurais Convolucionais (CNNs) são uma classe de algoritmos de aprendizado de máquina que se mostraram muito eficazes para classificação de imagens e reconhecimento de padrões. Elas são baseadas na ideia de que as camadas de neurônios conectadas, chamadas de camadas de convolução, podem ser usadas para extrair características relevantes das imagens.

Uma das principais aplicações das CNNs é o reconhecimento de objetos em imagens, incluindo a detecção de defeitos em módulos de memória. Elas também são amplamente utilizadas em tarefas de processamento de linguagem natural, como tradução automática e geração de texto.

De acordo com o artigo "*Deep learning in medical image analysis*" de [Litjens, K. *et al.* 2017], as CNNs “funcionam ao aprender características hierárquicas das imagens, começando com características simples como bordas e formas, e progredindo para características mais complexas, como objetos inteiros. A cada camada, a CNN extrai características mais complexas e abstratas a partir das características extraídas nas camadas anteriores, permitindo que a CNN faça previsões precisas sobre as imagens, mesmo quando elas são apresentadas em diferentes orientações e escalas”.

Algumas das arquiteturas de CNNs mais populares incluem VGG19, MobileNet e InceptionV3, que são amplamente utilizadas em diversas tarefas de classificação de imagens, e são as arquiteturas utilizadas no estudo mencionado no artigo.

2.2. Inspeção Visual de Produtos Baseada em Inteligência Artificial

A inspeção visual de produtos é uma tarefa crítica na indústria de manufatura, pois garante a qualidade dos produtos e evita problemas de segurança e garantia. No entanto, a inspeção visual manual é uma tarefa tediosa e suscetível a erros humanos, o que pode resultar em altos custos e ineficiência.

A Inteligência Artificial (IA) tem se mostrado uma ferramenta eficaz para

automatizar a inspeção visual de produtos. De acordo com o artigo "*Artificial intelligence in advanced manufacturing: Current status and future outlook*" de [Arinez, J. F., et al. 2020], "a IA tem sido aplicada para melhorar a eficiência, precisão e confiabilidade da inspeção visual, bem como para reduzir os custos e aumentar a flexibilidade da inspeção".

Entre as técnicas de IA utilizadas na inspeção visual de produtos, as CNNs tem se destacado como uma abordagem eficaz para a detecção de defeitos. De acordo com o artigo "*Recent advances in surface defect inspection of industrial products using deep learning techniques*" de [Zheng, X. et al. 2021], "as CNNs têm mostrado desempenho superior para a detecção de defeitos superficiais em produtos industriais, devido à sua capacidade de aprender características relevantes das imagens dos produtos e lidar com grandes quantidades de dados".

Além disso, o artigo "*A review on industrial surface defect detection based on deep learning technology*" de [Qi, S. et al. 2020] descreve em detalhes as aplicações da Visão Computacional baseada em modelos de CNNs em uma variedade de cenários industriais para tarefas de detecção de defeitos de superfície, que abrangem principalmente a inspeção de defeitos de superfície de aço, inspeção de defeitos de superfície de telha magnética, inspeção de defeitos de superfície de trilhos, inspeção de detecção de superfície de tela, inspeção de defeitos de superfície de célula solar e alguns outros.

Em resumo, a inspeção visual de produtos baseada em Inteligência Artificial é uma técnica que tem o potencial de melhorar a eficiência, precisão e confiabilidade da inspeção visual de produtos, enquanto também reduz os custos e aumenta a flexibilidade da inspeção. Entre as técnicas de IA, as Redes Neurais Convolucionais têm se mostrado uma abordagem eficaz para a detecção de defeitos em imagens de produtos

2.3. Abordagem de Aprendizagem de Máquina Supervisionada

A abordagem de Aprendizagem de Máquina Supervisionada é uma técnica amplamente utilizada para resolver problemas de classificação e reconhecimento de padrões. Ela é baseada no princípio de fornecer ao algoritmo um conjunto de registros rotulados de entrada e saída, a fim de permitir que o algoritmo aprenda a fazer previsões sobre novos dados. Um registro pode ser, por exemplo, uma imagem.

No contexto da detecção de defeitos em imagens de módulos de memória, a abordagem de Aprendizagem de Máquina Supervisionada pode ser empregada para classificar as imagens em duas classes: boas ou ruins. Isso é alcançado por meio da construção de um conjunto de dados supervisionado, que inclui imagens de módulos de memória já rotuladas como boas ou ruins.

De acordo com o artigo "*A review on industrial surface defect detection based on deep learning technology*" de [Qi, S. et al. 2020], "a Aprendizagem de Máquina Supervisionada tem sido amplamente utilizada para detecção de defeitos em imagens de produção, incluindo a detecção de defeitos de superfície a partir de imagens, devido à sua capacidade de lidar com grandes quantidades de dados e aprender características relevantes das imagens".

Além disso, o artigo "*Deep learning in medical image analysis*" de [Litjens, K. et al. 2017] menciona que "as técnicas de Aprendizagem de Máquina Supervisionada, como as CNNs, têm mostrado desempenho superior para tarefas de classificação de imagens, incluindo a detecção de defeito.

2.4. Pré-processamento

O pré-processamento de imagens é uma etapa importante no treinamento de modelos de redes neurais convolucionais (CNNs). Ele inclui técnicas para ajustar o tamanho, cor, contraste e brilho das imagens, bem como para normalizar os dados. Além disso, técnicas como *data augmentation* podem ser usadas para aumentar o tamanho do conjunto de dados e evitar *overfitting*.

Algumas técnicas comuns de pré-processamento que podem melhorar a acurácia de um modelo de CNN para classificação de imagens incluem: redimensionamento e normalização de imagem; *data augmentation*, filtros para remoção de ruído, centralização e alinhamento, e transformações de espaço de cores são técnicas comuns de pré-processamento que podem melhorar a acurácia de um modelo de CNN para classificação de imagens. A seguir uma breve explanação das técnicas de pré-processamento que foram empregadas nesse trabalho.

2.4.1 Data Augmentation

Data Augmentation é uma técnica utilizada para ampliar o conjunto de dados de treinamento de uma rede neural. A fim de que as redes neurais possam aprender e generalizar bem é necessário um grande conjunto de dados. *Data Augmentation* é uma maneira de aumentar o número de exemplos de treinamento sem a necessidade de coletar mais dados. Ela é especialmente útil em problemas de visão computacional, onde existem variações naturais nos dados, como rotação, escala, iluminação e distorção.

A *Data Augmentation* é realizada aplicando transformações sintéticas nas imagens de treinamento, como rotação, escala, translação, reflexão e ruído. Essas transformações geram novos exemplos de treinamento que são similares aos dados originais, mas diferentes o suficiente para evitar o *overfitting*. Segundo o artigo "*Data augmentation on plant leaf disease image dataset using image manipulation and deep learning techniques*" de [Pandian, J. A., 2019], "*Data Augmentation* é uma técnica eficaz para aumentar a generalização de redes neurais, pois gera exemplos de treinamento que são similares aos dados originais, mas diferentes o suficiente para evitar o *overfitting*".

Além disso, o artigo "*Data Augmentation Techniques for Deep Learning-Based Medical Image Analyses*" de [Kim, M. and Bae, H. J., 2020] menciona que "a *Data Augmentation* tem se mostrado uma técnica eficaz para melhorar o desempenho de redes neurais em problemas de visão computacional, pois aumenta o número de exemplos de treinamento e ajuda a evitar o *overfitting*".

2.4.2 Dropout

Dropout é uma técnica de regularização utilizada para evitar o *overfitting* em redes neurais. A técnica consiste em "desligar" aleatoriamente alguns neurônios durante o treinamento, impedindo que eles sejam atualizados. Isso força a rede a aprender representações mais robustas, pois ela não pode contar com a ativação de todos os neurônios em todas as camadas.

A técnica foi introduzida pela primeira vez por [Hinton *et al.* 2012] e mostrou-se eficaz para melhorar o desempenho de redes neurais em diversas tarefas de aprendizado de máquina, como classificação de imagens e processamento de linguagem natural. Segundo o artigo "*Dropout: A Simple Way to Prevent Neural Networks from Overfitting*" de [Nitish Srivastava *et al.* 2014], "a técnica Dropout pode ser vista como

uma forma de ensinar a rede a ser robusta às variações dos dados de treinamento, pois força a rede a aprender representações mais genéricas que não dependem fortemente de alguns neurônios específicos”.

Além disso, o artigo "*Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*" de [Gal Y. et al. 2016] menciona que “a técnica Dropout pode ser interpretada como uma aproximação bayesiana para modelagem da incerteza em redes neurais, pois permite avaliar a robustez do modelo a diferentes subconjuntos de neurônios e camadas.”

Dessa forma, a técnica Dropout é uma forma eficaz de regularizar redes neurais, evitando o *overfitting* e aumentando a robustez do modelo. Sua utilização pode ser vista como uma forma de ensinar a rede a lidar com variações nos dados de treinamento e modelar a incerteza

2.4.3 Filtros para Melhorar as Imagens

Os filtros são, normalmente, empregados para ajudar a destacar as características importantes na imagem e a torná-las mais fáceis de serem detectadas pelo modelo [Reinhard et al. 2002].

Filtros são comumente usados para alcançar esse objetivo. Algumas dessas técnicas incluem: (i) Ajuste de brilho e contraste linear: ajusta o brilho e o contraste da imagem multiplicando cada pixel pela mesma constante e adicionando outra constante; (ii) Equalização de histograma: ajusta o contraste da imagem redistribuindo os valores de pixels para cobrir o intervalo total disponível; Filtro gamma: ajusta o brilho e o contraste da imagem aplicando uma transformação logarítmica ou potência em cada pixel. Nesse trabalho foi empregado o filtro para ajustar o contraste e o brilho das imagens.

3. Trabalhos Relacionados

A utilização de inteligência artificial (IA) para automatizar processos de inspeção tem sido um tema crescente de pesquisa na indústria. Em particular, as redes neurais convolucionais (CNNs) têm mostrado uma excelente capacidade para a detecção de defeitos em imagens. O uso de CNNs para a detecção de defeitos em imagens de módulos de memória é um tema de pesquisa relativamente novo, mas já há alguns estudos publicados que abordam essa questão.

Um exemplo é o trabalho de [Kim, J. et al. 2021], "*Printed Circuit Board Defect Detection Using Deep Learning via A Skip-Connected Convolutional Autoencoder*" no qual foi utilizado uma CNN do tipo Autoencoder para detectar defeitos na superfície de placas de circuito impresso, do inglês (*Printed Circuit Boards - PCBs*). Os autores mostraram que a CNN foi capaz de detectar com precisão os defeitos e superou as técnicas de inspeção tradicionais.

Outro exemplo é o trabalho de [Xin, H. et al. 2021] "*PCB Electronic Component Defect Detection Method based on Improved YOLOv4 Algorithm*" no qual foi utilizado o algoritmo YOLOv4 baseado em CNN também para detectar defeitos em placas de circuito impresso. Os autores mostraram que a YOLOv4 conseguiu detectar alguns tipos de defeitos como componente faltando ou invertido, mas que a solução proposta precisa passar por mais experimentos visando avaliar melhor a sua eficiência.

O trabalho "*Defect Detection and Classification of Electronic Circuit Boards Using Keypoint Extraction and CNN Features*" de [Usami, Y. T. T. S. H., et al. 2017]

propõe um método para detecção e classificação de defeitos em placas de circuito eletrônico, extraíndo pontos-chave sem imagens de referência. O objetivo final é distinguir um defeito problemático, como desconexão, de um não-defeito, como poeira no processo de fabricação, entre outros. Os pontos-chave são extraídos da imagem da placa de circuito eletrônico e, em seguida, uma imagem de *patch* é cortada usando as informações obtidas dos pontos-chave, como a posição. As imagens cortadas são usadas como entrada para a CNN AlexNet.

No trabalho de [Shiina T., *et al.* 2018] a apresentaram uma nova proposta no trabalho “*Defect Classification of Electronic Circuit Board Using Multi-Input Convolutional Neural Netrok*” de uma arquitetura de CNNs própria baseada em duas CNNs combinadas capaz de receber ao mesmo tempo duas imagens de entrada por meio de duas camadas diferentes de conexão.

Além disso, também já há algumas pesquisas que apontam a possibilidade de aplicar algoritmos de IA para inspeção de outros tipos de defeitos como, por exemplo, defeito de solda “*Automated Inspection of Welding Defects using Deep Learning*” [Wenhui Hou *et al.*, 2017] e defeitos em tecido, “*EfficientDet for fabric defect detection based on edge computing*” [Song, S. *et al.*, 2021].

Esses estudos mostram que o uso de CNNs para a detecção de defeitos em imagens de módulos de memória é uma abordagem promissora e que já existem alguns estudos que mostram resultados positivos. A pesquisa nessa área ainda é incipiente e há ainda muito espaço para o desenvolvimento de novos métodos e aplicações.

O estudo descrito neste artigo se diferencia dos trabalhos supracitados por avaliar o desempenho de três arquiteturas canônicas de CNNs: VGG19, MobileNet e InceptionV3. A maioria dos trabalhos anteriores não avaliou nenhuma das CNNs experimentadas nesse trabalho, empregaram apenas uma arquitetura de CNN para a detecção de defeitos em imagens de módulos de memória ou combinaram CNNs. Além disso, esse estudo se diferencia por aplicar filtro para ajustar o contraste e brilho das imagens usadas para o treinamento, avaliação e testes dos modelos de CNNs avaliados no trabalho. Isso proporciona uma base sólida para o desenvolvimento de sistemas de detecção de defeitos automatizados para a indústria de tecnologia da informação e contribui para aplicar a indústria 4.0.

4. Material e Métodos

A seção Métodos descreve a abordagem proposta que será utilizada para detecção de defeitos físicos em imagens de módulos de memória, as arquiteturas de CNNs que foram avaliadas, conjunto de dados empregado e o processo de treinamento dos modelos.

4.1. Abordagem de Inspeção de Módulos de Memória para Detecção de Defeitos

Um sistema de inspeção de defeitos físicos em módulos de memória utilizando CNNs Canônicas de classificação binária de imagens é uma abordagem eficaz para detectar e identificar defeitos físicos nos módulos de memória. Ele funciona da seguinte maneira:

Primeiro, as imagens dos módulos de memória são capturadas com uma câmera especializada e armazenadas em um banco de dados. Essas imagens são então pré-processadas para remover ruídos e melhorar a qualidade da imagem. Em seguida, as imagens pré-processadas são utilizadas como entrada para uma CNN canônica de classificação de imagens. Essa rede neural é treinada com uma grande quantidade de

imagens de módulos de memória, tanto com quanto sem defeitos físicos.

Durante o processo de treinamento, a CNN aprende a reconhecer padrões e características associadas a módulos de memória com defeitos físicos, ao mesmo tempo em que aprende a ignorar características irrelevantes. Quando uma nova imagem é inserida no sistema, a CNN é capaz de classificá-la como tendo ou não um defeito físico com uma elevada precisão. Se um defeito físico for detectado, o sistema pode acionar uma série de ações, como a rejeição do módulo de memória ou a marcação do módulo para inspeção adicional.

Além disso, o sistema pode ser configurado para aprender continuamente a partir de novos dados, permitindo que ele se adapte e melhore com o tempo. É importante ressaltar que as redes neurais convolucionais são muito eficazes para detectar padrões e características em imagens, o que faz delas uma excelente escolha para o sistema de inspeção de defeitos físicos em módulos de memória.

4.2. Arquiteturas de CNNs

Neste estudo, foram avaliadas quatro arquiteturas de CNNs populares para o sistema de detecção de defeitos em imagens de módulos de memória: VGG19, MobileNet, e InceptionV3. Cada uma dessas arquiteturas possui suas próprias vantagens e limitações, que serão discutidas a seguir.

A VGG19 é uma arquitetura de CNN desenvolvida por [Simonyan and Zisserman 2014] e foi utilizada para vencer a competição ImageNet 2014. Ela é conhecida por sua alta precisão, mas também é conhecida por ser uma arquitetura de alta complexidade computacional.

A MobileNet é uma arquitetura de CNN desenvolvida por [Howard *et al.*, 2017] com o objetivo de ser usada em dispositivos móveis com limitações de recursos. Ela é conhecida por sua eficiência computacional, mas também é conhecida por ter menor precisão em comparação com outras arquiteturas de CNN.

A InceptionV3 é uma arquitetura de CNN desenvolvida por [Szegedy *et al.*, 2015] e foi utilizada para vencer a competição ImageNet 2015. Ela é conhecida por sua alta precisão e eficiência computacional. A CNN criada especificamente para este estudo foi desenvolvida com base em uma combinação das arquiteturas VGG19 e InceptionV3, com o objetivo de obter uma boa precisão e eficiência computacional.

Vale ressaltar que a VGG19 é considerada uma das redes neurais mais profundas e complexas entre as três mencionadas. Ela tem 19 camadas de processamento, enquanto que MobileNet tem apenas 30 camadas e InceptionV3 tem apenas 159 camadas. Além disso, a VGG19 utiliza filtros de convolução de tamanhos 3x3 e 2x2, o que permite a extração de características mais complexas e detalhadas das imagens.

Também é importante mencionar que a InceptionV3 e MobileNet são redes neurais que foram projetadas para serem mais leves e eficientes em termos de recursos. MobileNet especificamente, foi projetado para funcionar bem em dispositivos móveis e dispositivos IoT. Este é um ponto forte para ele, pois ele tem uma boa precisão com menos recursos computacionais. A escolha dessas arquiteturas de CNN foi baseada em sua ampla utilização e desempenho comprovado em diversas tarefas de classificação de imagens. Além disso, estas arquiteturas de CNN possuem diferentes níveis de complexidade computacional, permitindo uma comparação entre diferentes abordagens e selecionar a melhor arquitetura para a tarefa de detecção de defeitos em imagens de módulos de memória.

4.3. Conjunto de Dados

Para o treinamento, avaliação e testes dos modelos de CNNs empregados para tarefa de detecção de defeitos em imagens de módulos de memória, foi utilizado um conjunto de dados supervisionado com 4804 imagens classificadas em duas classes: *Good* e *Bad*. O conjunto de dados foi construído por meio da captura de imagens dos módulos de memória. Para isso foi montado uma plataforma mostrado na Figura 1 para aquisição de imagens consistindo de uma fonte de luz baseada em anel de LED e suporte para câmera industrial de alta resolução de 16 Mega *pixels*. A fim de evitar distorção, uma lente industrial com zoom sem distorção também foi usada, a distância focal pode ser ajustada entre 6-12mm e a abertura máxima é f1.6.



Figura 1. Plataforma para captura de Imagens

A resolução da imagem obtida é de 5198 X 3462 *pixels*, a literatura sugere para o treinamento dos modelos, imagens com resoluções de 28x28 *pixels* até 1024x1024 *pixels*. Neste trabalho, conforme ilustrado na Figura 2, a imagem original do módulo de memória foi recortada para aquisição de imagens padronizadas na resolução de 456x193 *pixels*.

Os recortes destacados em vermelhos são os que apresentam algum tipo de defeito. Já os recortes destacados em verde são os que não apresentam defeitos. As razões para o emprego dessa abordagem são, especialmente, a redução do tempo e do custo computacional para o treinamento do modelo de aprendizagem de máquina profundo, além de viabilizar a aquisição de imagens com exemplos diversos de defeitos físicos.

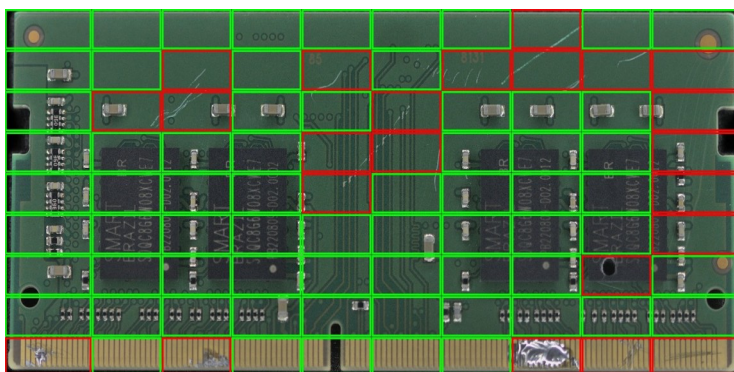


Figura 2. Imagem do Módulo de Memória

O conjunto de dados disponível foi dividido aleatoriamente seguindo o critério de validação *Hold-out*. O *Hold-out* é um método de validação cruzada usado para avaliar o desempenho de um modelo de aprendizado de máquina. Consiste em dividir os dados de treinamento em duas partes: um conjunto de treinamento e um conjunto de validação. O modelo é treinado usando o conjunto de treinamento e é avaliado usando o conjunto de validação. Isso permite avaliar o desempenho do modelo em dados que ele nunca viu antes, o que é importante para avaliar a capacidade de generalização do modelo. O *hold-out* é útil quando os dados são escassos [Maleki, F. 2020].

Nesse trabalho seguindo o *Hold-out*, 80% dos dados foram usados para a fase de treino, 10% para validação e, conseqüentemente, 10% para teste. Esse procedimento é comum para validar os modelos de aprendizado de máquina e garantir que os resultados sejam generalizáveis a novos conjuntos de dados. O objetivo do *Hold-out* é garantir que o modelo seja capaz de generalizar bem e fazer previsões precisas em novos dados. A Figura 3 apresenta alguns recortes da imagem original do módulo de memória que foram empregadas no procedimento de treinamento e validação dos modelos.

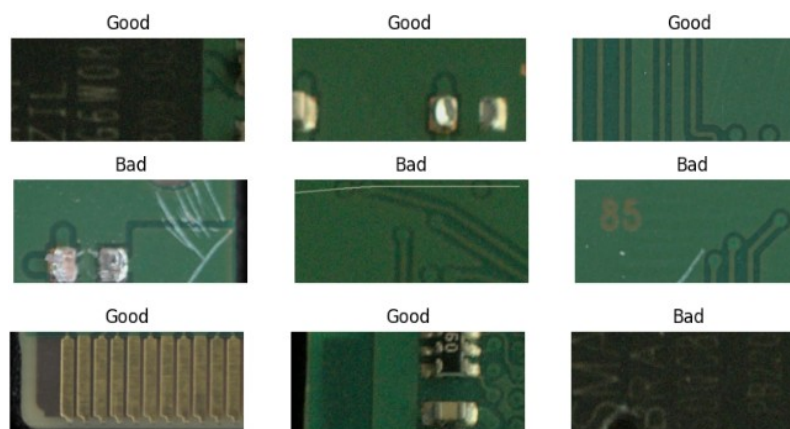


Figura 3. Imagens do Conjunto de Dados

A Figura 3 apresenta exemplos de recortes de imagens do módulo de memórias rotulados como “*Bad*” ou “*Good*”. Os recortes rotulados como “*Bad*” são exemplos de imagens que apresentam algum tipo de dano físico, como arranhões, circuito aberto, mordida de roedor, curto, entre outros. Já os recortes de imagens rotulados como “*Good*” são exemplos de imagens que não apresentam nenhum tipo de dano físico. Para o treinamento dos modelos, foram utilizadas 4804 imagens, sendo 2000 imagens rotuladas como “*Bad*” e 1836 como “*Good*”. Para a validação, foram utilizadas 768 imagens, divididas equitativamente entre as duas classes. Por fim, foram utilizadas 200 imagens para os testes, também divididas equitativamente entre as duas classes.

4.4. Treinamento de Modelos

Para treinar os modelos de CNNs, foi utilizado o software TensorFlow [Pang, B. *et al.*, 2020]. O treinamento dos modelos foi realizado em um servidor especialmente preparado para a tarefa de treinamento de modelos CNNs, disponível no Laboratório Ludus, o qual possui um processador Intel Core i7, 32 GB de RAM, SSD de 256 MB e placa gráfica com 11GB de GPU, o que viabilizou a aceleração em hardware do

treinamento.

O treinamento de modelos envolve ajustar os parâmetros e hiperparâmetros da sua configuração. Esse ajuste consiste em escolher o otimizador, adicionar ou retirar o *Patience*, *Data Augmentation*, Dropout e definir o número de épocas. Um otimizador é empregado para maximizar o desempenho de um modelo de Aprendizagem. As épocas são o número de vezes que o algoritmo de treinamento passa pelo conjunto de dados de treinamento. O uso de *Patience* permite que o treinamento pare quando a performance do modelo não melhora após um determinado número de épocas. Já o otimizador é utilizado para maximizar o desempenho de um modelo de Aprendizagem de Máquina ajustando iterativamente os seus pesos até que o erro seja reduzido.

Nesse trabalho foram utilizados os otimizadores RMSProp e o Adam. Ambos são no momento os dois dos otimizadores mais populares utilizados na otimização de redes neurais. São baseados no método de descida de gradiente estocástico, mas possuem algumas diferenças importantes. Adam, que significa "*Adaptive Moment Estimation*", foi proposto em 2014 por [Kingma e Ba, 2014]. Ele combina as ideias da descida de gradiente estocástica com o Momentum. Ele mantém duas médias móveis, uma para os gradientes e outra para os gradientes ao quadrado, e ajusta dinamicamente a taxa de aprendizado. Isso permite que o Adam se adapte automaticamente às diferentes escalas de parâmetros e aproveite melhor as informações dos gradientes anteriores.

Por outro lado, RMSProp foi proposto por [Tieleman and Hinton 2012]. Ele também mantém uma média móvel dos gradientes ao quadrado e ajusta dinamicamente a taxa de aprendizado. No entanto, ele utiliza uma constante de esquecimento para esquecer gradualmente os gradientes mais antigos e evitar problemas de acumulação de gradientes. Além disso, RMSProp divide a taxa de aprendizado pelo desvio padrão dos gradientes ao quadrado, o que permite uma adaptação mais precisa às diferentes escalas de parâmetros.

É importante mencionar que a acurácia de uma rede neural também pode ser aumentada através da otimização de hiperparâmetros e da realização de ajustes finos. Por isso, é recomendável experimentar com diferentes redes neurais e configurações para encontrar a melhor solução para o seu problema específico. Dessa forma, foram treinados um número total de 12 modelos com diferentes configurações de parâmetros e hiperparâmetros. A etapa de treinamento de todos os modelos demorou aproximadamente 20 horas. As configurações de treinamento de CNNs são apresentadas na tabela a seguir.

Tabela 1. Configurações para Treinamento de CNNs

| CNN | Configuração | Épocas | <i>Patience</i> | Otimizador | <i>Data Augmentation</i> | Dropout | Filtro |
|-------------|--------------|--------|-----------------|------------|--------------------------|---------|--------|
| VGG19 | 11 | 100 | Sim | Adam | Não | Não | Não |
| VGG19 | 12 | 100 | Sim | Adam | Sim | Sim | Sim |
| MobileNet | 11 | 100 | Sim | Adam | Não | Não | Não |
| MobileNet | 12 | 100 | Sim | Adam | Sim | Sim | Sim |
| InceptionV3 | 11 | 100 | Sim | Adam | Não | Não | Não |
| InceptionV3 | 12 | 100 | Sim | Adam | Sim | Sim | Sim |
| VGG19 | 21 | 100 | Sim | RMSProp | Não | Não | Não |
| VGG19 | 22 | 100 | Sim | RMSProp | Sim | Sim | Sim |
| MobileNet | 21 | 100 | Sim | RMSProp | Não | Não | Não |
| MobileNet | 22 | 100 | Sim | RMSProp | Sim | Sim | Sim |
| InceptionV3 | 21 | 100 | Sim | RMSProp | Não | Não | Não |
| InceptionV3 | 22 | 100 | Sim | RMSProp | Sim | Sim | Sim |

5. Resultados

Essa seção apresenta e discute dos resultados obtidos, incluindo a comparação do desempenho das diferentes arquiteturas de CNNs avaliadas. A fase de teste foi realizada com o conjunto de dados reservado para esse fim, e os resultados foram comparados com as previsões feitas pelos modelos treinados.

O desempenho do modelo foi avaliado com a métrica de acurácia, que é uma medida da proporção de previsões corretas feitas pelo modelo. A acurácia é uma das métricas mais comumente utilizadas para avaliar o desempenho de modelos de aprendizagem de máquina. Ela é definida como a proporção de previsões corretas feitas pelo modelo em relação ao total de previsões.

Em outras palavras, é a relação entre o número de previsões corretas e o número total de previsões. A acurácia é uma métrica simples e fácil de entender, mas pode ser enganosa em alguns casos, especialmente quando existe desequilíbrio de classe no conjunto de dados [He, *et al.* 2016]. Os resultados obtidos dos modelos treinados de CNNs são apresentados na tabela a seguir.

Tabela 2. Resultados de CNNs

| CNN | Configuração | Acurácia |
|-------------|--------------|----------|
| VGG19 | 11 | 88,53% |
| VGG19 | 12 | 90,15% |
| MobileNet | 11 | 88,56% |
| MobileNet | 12 | 90,84% |
| InceptionV3 | 11 | 87,43% |
| InceptionV3 | 12 | 90,93% |
| VGG19 | 21 | 90,04% |
| VGG19 | 22 | 91,78% |
| MobileNet | 21 | 95,71% |
| MobileNet | 22 | 98,12% |
| InceptionV3 | 21 | 90,02% |
| InceptionV3 | 22 | 91,40% |

Todas as CNNs que foram avaliadas conseguiram realizar o procedimento de treinamento sem atingir o valor estipulado na configuração de *patience*, o que indica um aprendizado contínuo. A melhor CNN que obteve o melhor valor de acurácia, foi a arquitetura MobileNet, de configuração 22 que atingiu a acurácia de 98,12%. Embora o VGG19 seja considerado um modelo de referência em classificação de imagens e tenha uma arquitetura mais profunda do que o MobileNet e o InceptionV3, isso não garante necessariamente que obterá uma acurácia melhor. Há várias razões pelas quais o MobileNet pode ter obtido uma acurácia melhor nesses resultados específicos.

Uma razão é que o MobileNet foi projetado especificamente para ser utilizado em dispositivos móveis, onde a eficiência computacional é crucial. Isso é alcançado através do uso de camadas de profundidade móvel e do uso de camadas de ponto de fusão, que reduzem o número de parâmetros e a complexidade computacional do modelo. Isso pode dar ao MobileNet uma vantagem sobre os outros modelos em termos de generalização, ou seja, a capacidade do modelo de fazer previsões precisas em novos conjuntos de dados.

Outra razão pode ser que o conjunto de dados usado para treinar os modelos foi mais adequado para o MobileNet. Ele pode ter sido projetado para lidar com imagens semelhantes às presentes no conjunto de dados de treinamento, o que levou a uma melhor performance. A acurácia de uma rede neural na tarefa de classificação binária de imagens varia de acordo com o conjunto de dados e as configurações utilizadas. É possível que as configurações usadas para treinar o MobileNet foram mais adequadas para esse conjunto de dados específico e, portanto, levaram a uma melhor performance. Ainda vale ressaltar que os modelos que apresentaram melhor acurácia foram aqueles treinados com imagens pré-processadas, o que comprova a eficiência das técnicas de suporte ao pré-processamento.

Em geral, redes neurais mais profundas e complexas, como a VGG19, tendem a ter melhores desempenhos em tarefas de classificação. No entanto, essas redes também tendem a ser mais pesadas e requerem mais recursos computacionais, o que exigirá um hardware com maior poder computacional para oferecer o suporte adequado a tarefa de inspeção de defeitos em módulos de memória. Os gráficos de desempenho do melhor modelo são apresentados nas Figura 4.



Figura 4. Gráficos de Desempenho do Melhor Modelo

Na Figura 4 é apresentado o gráfico da acurácia do modelo em relação ao número de épocas quando empregado os conjuntos de treinamento e validação. Também é mostrado o gráfico de perda do modelo em relação ao número de épocas quando utilizado os conjuntos de treinamento e validação. Por meio dos gráficos é possível constatar que a acurácia do treinamento e validação estão muito próximas, comportamento também observado no gráfico de perdas o que indicar que o modelo não apresenta *overfitting*. Isso leva a crer que o uso das técnicas de *Data Augmentation*, *Droupot* surtiram o efeito desejado de minimizar o problema do *overfitting*.

Também foi possível constatar por meio dos gráficos que durante as épocas 2 e 19 houve uma queda da acurácia e um aumento da perda, isso levar a crê que o conjunto

de dados usado para treinamento e validação deve ter algumas imagens que necessitam a correção do rótulo, apesar disso, a acurácia final do modelo de aproximadamente 98,12% indica que a utilização de CNNs para detecção de defeitos módulos de memória é promissor. A Figura 5 apresentam algumas imagens classificadas pelo modelo MobileNet. É possível notar que todos os recortes de imagens foram classificados corretamente, dentre eles, alguns defeitos detectados de um módulo de memória.

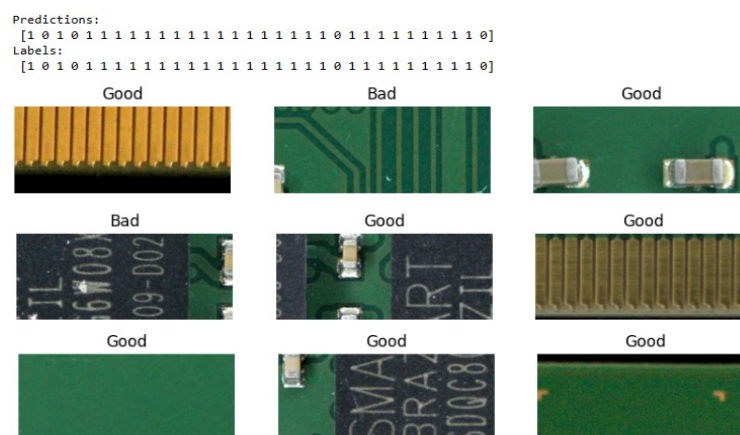


Figura 5. Imagens Classificadas

Conforme apresentado na seção 3. de Trabalhos Relacionados, [Usami, Y. T. T. S. H., *et al.* 2017] propuseram o emprego da CNN AlexNet para extração de características da imagem de um circuito eletrônico visando a detecção de defeitos por meio de um modelo baseado em *Support Vector Machine* (SVM) que foi treinado e obteve uma acurácia de 82,64%. Já [Kim, J., *et al.* 2021] propuseram o emprego de CNN Autoencoder para detecção de defeito em Placa de Circuito Impresso, onde o Autoencoder Convolutacional obteve uma acurácia de 95,08%. Considerando os resultados apresentados na Tabela 2, A MobileNet configuração 22 apresentou uma acurácia melhor do que as acurácias apresentadas nos trabalhos de [Usami, Y. T. T. S. H., *et al.* 2017] e [Kim, J., *et al.* 2021].

6. Conclusão

O objetivo principal deste trabalho consistiu em realizar um estudo visando avaliar o uso de CNNs na detecção de defeitos físicos em imagens de módulos de memória. Para isso, foi criada um conjunto de dados com 4804 imagens classificadas em duas classes: *Good* e *Bad*. A partir dessa etapa, foram treinadas arquiteturas canônicas de CNNs amplamente utilizadas na tarefa de Visão Computacional implementada por meio de classificação binária de imagens. Nesse estudo foram avaliadas as CNNs: VGG19, MobileNet e InceptionV3. No treinamento dessas CNNs foram usados diferentes configurações de parâmetros e hiperparâmetros com um total de 12 variações das redes.

A partir deste estudo, foi possível concluir que a CNN MobileNet com a configuração 22 foi a melhor arquitetura para a detecção de defeitos físicos em módulos de memória, apresentando uma acurácia de aproximadamente 98,12%. Estes resultados apresentados são significativos para o desenvolvimento de sistemas de detecção de defeitos automatizados e podem contribuir para aplicar a indústria 4.0. Sugerimos

futuros estudos para aprimorar a arquitetura da MobileNet, e aplicar outros filtros de imagem eliminar ruídos como o filtro Gaussiano e a binarização de imagem a fim de aumentar a acurácia do sistema de detecção de defeitos. Além disso, avaliar novas CNNs que estão sendo lançadas, como ResNetxt50 e EfficientNetB50 visando melhorar ainda mais o desempenho do sistema. Vale ressaltar que o problema detecção de defeitos físicos em imagens de módulos de memória é relevante para o cenário atual da indústria inteligente também conhecida como indústria 4.0.

Referências

- Arinez, J. F., Chang, Q., Gao, R. X., Xu, C., & Zhang, J. (2020). Artificial intelligence in advanced manufacturing: Current status and future outlook. *Journal of Manufacturing Science and Engineering*, 142(11).
- Gal, Y., & Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning* (pp. 1050-1059).
- He, H., and Garcia, E. A. (2016). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 28(6), 1421-1439.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
- Kim, J., Ko, J., Choi, H., & Kim, H. (2021). Printed circuit board defect detection using deep learning via a skip-connected convolutional autoencoder. *Sensors*, 21(15), 4968.
- Kim, M., Bae, H. J. (2020). Data Augmentation Techniques for Deep Learning-Based Medical Image Analyses. *Journal of the Korean Society of Radiology*, 81(6).
- Kingma, D. P.; Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
- Litjens, K., Kooi, T., Bejnordi, B. E., Setio, A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19, 221-248.
- Maleki, F., Muthukrishnan, N., Ovens, K., Reinhold, C., & Forghani, R. (2020). Machine learning algorithm validation: from essentials to advanced applications and implications for regulatory certification and deployment. *Neuroimaging Clinics*, 30(4), 433-445.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". *Journal of Machine Learning Research*, 2014.
- Pang, B., Nijkamp, E., & Wu, Y. N. (2020). Deep learning with tensorflow: A review. *Journal of Educational and Behavioral Statistics*, 45(2), 227-248.
- Pandian, J. A., Geetharamani, G., & Annette, B. (2019, December). Data augmentation on plant leaf disease image dataset using image manipulation and deep learning techniques. In *2019 IEEE 9th international conference on advanced computing (IACC)* (pp. 199-204). IEEE.
- Reinhard, E., Stark, M., Shirley, P., & Ferwerda, J. (2002, July). Photographic tone reproduction for digital images. In *Proceedings of the 29th annual conference on*

Computer graphics and interactive techniques (pp. 267-276).

- Qi, S., Yang, J., & Zhong, Z. (2020, September). A review on industrial surface defect detection based on deep learning technology. In *2020 the 3rd international conference on machine learning and machine intelligence* (pp. 24-30).
- 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(1), 1929-1958.
- Simonyan and K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
- Song, S., Jing, J., Huang, Y., & Shi, M. (2021). EfficientDet for fabric defect detection based on edge computing. *Journal of Engineered Fibers and Fabrics*, 16, 15589250211008346.
- Shiina, T., Iwahori, Y., & Kijisirikul, B. (2018). Defect classification of electronic circuit board using multi-input convolutional neural network. *Int. J. Comput. Softw. Eng.*, 3(2), 2456-4451.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2818-2826).
- Tieleman, T.; Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2), 26-31.
- Usami, Y. T. T. S. H., Iwahori, Y., & Bhuyan, M. K. Defect Detection and Classification of Electronic Circuit Boards Using Keypoint Extraction and CNN Features. In *PATTERNS 2017 : The Ninth International Conferences on Pervasive Patterns and Applications*
- Wenhui Hour et al. (2019), "Automatic Detection of Welding Defects using Deep Neural Network", *Journal of Physics: Conference Series*, Volume 933, *The 2nd Aceh International Symposium on Civil Engineering*.
- Xin, H., Chen, Z., & Wang, B. (2021, March). PCB electronic component defect detection method based on improved YOLOv4 algorithm. In *Journal of Physics: Conference Series* (Vol. 1827, No. 1, p. 012167). IOP Publishing.
- Zheng, X., Zheng, S., Kong, Y., & Chen, J. (2021). Recent advances in surface defect inspection of industrial products using deep learning techniques. *The International Journal of Advanced Manufacturing Technology*, 113, 35-58.