

UNIVERSIDADE FEDERAL DE SANTA MARIA
CAMPUS DE CACHOEIRA DO SUL
CURSO DE ENGENHARIA ELÉTRICA

Luan Silveira

**AVALIAÇÃO DE REDES NEURAIS APLICADAS NA CLASSIFICAÇÃO
E IDENTIFICAÇÃO DE FALHAS DE PRODUTOS EM LINHA
PRODUTIVA**

Cachoeira do Sul, RS
2023

Luan Silveira

**AVALIAÇÃO DE REDES NEURAIS APLICADAS NA CLASSIFICAÇÃO E IDENTIFICAÇÃO DE
FALHAS DE PRODUTOS EM LINHA PRODUTIVA**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Elétrica da Universidade Federal de Santa Maria (UFSM) – Campus Cachoeira do Sul, como requisito parcial para a obtenção do título de **Bacharel em Engenharia Elétrica**.

Orientador: Dr. Paulo César Vargas Luz

Cachoeira do Sul, RS
2023

Luan Silveira

**AVALIAÇÃO DE REDES NEURAIS APLICADAS NA CLASSIFICAÇÃO E IDENTIFICAÇÃO DE
FALHAS DE PRODUTOS EM LINHA PRODUTIVA**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Elétrica da Universidade Federal de Santa Maria (UFSM) – Campus Cachoeira do Sul, como requisito parcial para a obtenção do título de **Bacharel em Engenharia Elétrica**.

Aprovado em 10 de julho de 2023:

Paulo César Vargas Luz, Dr. (UFSM)

Laura Lisiane Callai dos Santos, Dra. (UFSM)

Nelson Knak Neto, Dr. (UFSM)

NUP: 23081.090090/2023-60

Prioridade: Normal

Homologação de ata de defesa de TCC e estágio de graduação

125.322 - Bancas examinadoras de TCC: indicação e atuação

COMPONENTE

Ordem	Descrição	Nome do arquivo
1	Ata de defesa de trabalho de conclusão de curso (TCC) (125.322)	Página de assinaturas TCC LUAN.pdf

Assinaturas

21/07/2023 08:58:29

PAULO CÉSAR VARGAS LUZ (PROFESSOR DO MAGISTÉRIO SUPERIOR (Ativo))
31.06.00.00.0.0 - COORDENAÇÃO ACADÊMICA - UFSM-CS - C_ACA_UFSM/CS

21/07/2023 09:25:13

LAURA LISIANE CALLAI DOS SANTOS (PROFESSOR DO MAGISTÉRIO SUPERIOR (Ativo))
31.06.00.00.0.0 - COORDENAÇÃO ACADÊMICA - UFSM-CS - C_ACA_UFSM/CS

21/07/2023 09:36:18

NELSON KNAK NETO (PROFESSOR DO MAGISTÉRIO SUPERIOR (Ativo))
31.06.00.00.0.0 - COORDENAÇÃO ACADÊMICA - UFSM-CS - C_ACA_UFSM/CS

Código Verificador: 2999676

Código CRC: 6bd6eac4

Consulte em: <https://portal.ufsm.br/documentos/publico/autenticacao/assinaturas.html>



RESUMO

AVALIAÇÃO DE REDES NEURAIS APLICADAS NA CLASSIFICAÇÃO E IDENTIFICAÇÃO DE FALHAS DE PRODUTOS EM LINHA PRODUTIVA

AUTOR: Luan Silveira
ORIENTADOR: Paulo César Vargas Luz

A Inteligência Artificial (IA) é uma área emergente que tem aplicações em diversos setores, como a engenharia, automação industrial, saúde, agricultura, entre outros. Apenas nos últimos seis anos, o número de publicações relacionadas a IA no arXiv cresceu mais de seis vezes, de 5.478 em 2015 para 34.736 em 2020. A engenharia elétrica também se beneficia muito da IA, com aplicações como previsão de demanda de eletricidade, controle de processos, seleção de produtos danificados em processo produtivo, entre outros. Esta área é extremamente complexa, pois abrange uma grande variedade de técnicas para solucionar problemas distintos, onde cada método pode ter uma infinidade de variações. Dentre as diversas ramificações da IA, o Deep Learning destaca-se devido à sua capacidade de resolver problemas complexos com maior precisão e rapidez. Neste trabalho, é proposto um sistema de reconhecimento de imagem utilizando redes neurais para identificação automatizada de objetos danificados em processos industriais além da classificação automatizada desses objetos por tipo. Foram realizados treinamentos de redes neurais com o propósito de classificar figuras geométricas, empregando variações paramétricas e estruturais, assim como a utilização de conjuntos de dados diversos e diferentes estratégias de pré-processamento. Os resultados obtidos evidenciam a eficácia das redes neurais na tarefa de classificação de imagens, com uma acurácia superior àquela estimada para o desempenho humano. O melhor modelo obtido nesse estudo foi comparado a trabalhos similares, revelando-se semelhante ou superior em determinados casos.

Palavras-chave: Inteligência Artificial. Redes Neurais. Automação Industrial. Engenharia Elétrica.

ABSTRACT

CLASSIFICATION AND IDENTIFICATION OF FAILURES IN OBJECTS OF AN INDUSTRIAL PRODUCTION LINE THROUGH NEURAL NETWORKS

AUTHOR: Luan Silveira

ADVISOR: Paulo César Vargas Luz

Artificial Intelligence (AI) is an emerging area that has applications in several sectors, such as engineering, industrial automation, health, agriculture, among others. In the last six years alone, the number of AI-related publications on arXiv has grown more than sixfold, from 5,478 in 2015 to 34,736 in 2020. Electrical engineering also benefits greatly from AI, with applications such as electricity demand forecasting, control of processes, selection of damaged products in the production process, among others. This area is extremely complex, as it covers a wide variety of techniques to solve different problems, where each method can have an infinity of variations. Among the various branches of AI, Deep Learning stands out due to its ability to solve complex problems with greater precision and speed. In this work, an image recognition system using neural networks for automated identification of damaged objects in industrial processes is proposed, in addition to the automated classification of these objects by type. Neural networks were trained with the purpose of classifying geometric figures, using parametric and structural variations, as well as the use of different data sets and different pre-processing strategies. The results obtained show the effectiveness of neural networks in the image classification task, with an accuracy higher than that estimated for human performance. The best model obtained in this study was compared to similar works, proving to be similar or superior in certain cases.

Keywords: Artificial Intelligence. Neural Networks. Industrial Automation. Electrical Engineering.

SUMÁRIO

1	Introdução	5
1.1	Objetivo geral.....	5
1.2	Objetivos específicos.....	5
1.3	Organização do documento	6
2	Fundamentação Teórica	7
2.1	Inteligência artificial	7
2.2	Deep Learning.....	7
2.2.1	Introdução às redes neurais	7
2.2.2	Tipos de redes neurais	11
2.2.3	Funções de ativação.....	15
2.2.4	Funções de custo	26
2.2.5	Otimizadores	28
2.3	Trabalhos relacionados	33
2.4	Considerações finais do capítulo.....	38
3	Metodologia	40
3.1	Considerações finais do capítulo.....	43
4	Resultados.....	44
4.1	Criação do dataset de figuras adequadas	44
4.2	Pré-processamento	47
4.3	Modelos de Inteligência Artificial.....	47
4.4	Resultados.....	49
4.5	Novo pré-processamento	52
4.6	Novos resultados	56
4.7	Danificação de figuras	61
4.8	Danificação automática de figuras	64
4.9	Sistema de classificação na esteira transportadora	67
5	Conclusão	71
5.1	Trabalhos futuros.....	71
6	Referências.....	73
7	Apêndices	79
7.1	União entre danificação automática e manual	79

1 INTRODUÇÃO

A inteligência artificial (IA) é a capacidade das máquinas de raciocinar e aprender, sendo capazes de realizar tarefas destinadas a humanos, muitas vezes com mais eficiência e velocidade (MODI, 2021). O uso de inteligência artificial em programas de computador já existe há anos, mas atualmente está sendo aplicado a muitos outros produtos e serviços. Por exemplo, câmeras podem determinar quais objetos estão presentes em uma imagem, robôs podem se passar por humanos na conversação, carros podem se tornar autônomos, etc. Especialistas preveem muitos outros usos inovadores para a inteligência artificial no futuro, incluindo redes elétricas inteligentes (ABHISHEK, 2022). Além disso, a inteligência artificial é o princípio de funcionamento da indústria 5.0, que adiciona inteligência à indústria 4.0, cujos pilares são automação e eficiência. Com a indústria 5.0, não somente os processos industriais são automatizados, mas também o processo de tomada de decisão, pois a inteligência artificial torna as máquinas capazes de trabalhar em conjunto com humanos (KLEININGS, 2022).

A inteligência artificial não tem como objetivo a substituição do trabalho humano, pois funciona de forma diferente do cérebro, mas as máquinas têm uma vantagem significativa: sua capacidade de analisar grandes quantidades de dados de forma extremamente rápida. Logo, a IA permite que os humanos se concentrem em tarefas mais críticas e menos repetitivas, aumentando a produtividade (ABHISHEK, 2022).

Assim como em outras indústrias, a inteligência artificial e o aprendizado de máquina estão mudando a engenharia, permitindo que os engenheiros façam coisas que antes não seriam possíveis (MARR, 2020), como detecção de perdas não técnicas, disagregação do consumo de energia, previsão de geração de energia renovável, gestão de recursos de energia distribuídos, atendimento ao consumidor, recarga inteligente de veículos elétricos, segurança cibernética no setor de energia, manutenção preditiva e previsão de falhas, etc (GONÇALVES, 2020).

Devido à eficiência, velocidade e vasta variedade de aplicações tanto em áreas diversificadas quanto na engenharia, a inteligência artificial demonstra-se extremamente promissora. Por isso, este trabalho propõe um estudo sobre IA contemplando suas ramificações e métodos, buscando entender principalmente o funcionamento do Deep Learning para possibilitar a criação de redes neurais capazes de resolver problemas de engenharia.

1.1 OBJETIVO GERAL

Desenvolver uma rede neural capaz de classificar e identificar falhas em objetos de uma linha produtiva industrial, bem como avaliar o desempenho desta rede perante variações paramétricas/estruturais.

1.2 OBJETIVOS ESPECÍFICOS

Para alcançar o objetivo principal deste trabalho é necessário a realização dos seguintes objetivos específicos:

- Revisar a bibliografia acerca da IA, Machine Learning e Deep Learning;
- Criar um banco de imagens para treinamento da rede neural;
- Rotular as imagens para criar um dataset de treinamento;
- Implementar a rede neural para classificar e identificar falhas em objetos;
- Avaliar o desempenho da rede neural para diferentes parâmetros e configurações;
- Buscar a otimização do algoritmo;
- Implantar o sistema de classificação à uma esteira transportadora automatizada.

1.3 ORGANIZAÇÃO DO DOCUMENTO

Este documento está estruturado em quatro capítulos principais a partir deste ponto. O primeiro capítulo (capítulo 2) consiste em uma revisão técnica, onde são explanadas as tecnologias existentes no campo da Inteligência Artificial, abrangendo os diferentes métodos, princípios de funcionamento e suas variações. O segundo capítulo (capítulo 2.3) aborda os trabalhos relacionados, fornecendo um contexto geral e destacando a relevância do tema deste trabalho, por meio da apresentação de aplicações de Inteligência Artificial para solucionar problemas diversos de engenharia e automação. O terceiro capítulo (capítulo 3), tem como objetivo descrever detalhadamente a metodologia adotada neste trabalho, onde serão apresentados os procedimentos, técnicas e abordagens utilizados na realização da pesquisa, visando alcançar os objetivos propostos. O quarto capítulo (capítulo 4), tem como propósito apresentar os resultados obtidos a partir da aplicação da metodologia descrita no capítulo anterior. Serão detalhados os principais achados e análises decorrentes desses resultados. Além disso, será realizado um estudo desses resultados e um comparativo com trabalhos similares.

2 FUNDAMENTAÇÃO TEÓRICA

A Inteligência Artificial é um termo amplo que pode definir a realização de qualquer atividade realizada por um computador que normalmente requer a inteligência humana. Técnicas de Inteligência Artificial vêm sendo amplamente desenvolvidas e aplicadas em diversas áreas resolvendo diferentes problemas devido ao seu grande potencial e capacidade de auto aprendizagem. Desta forma, este capítulo visa esclarecer os principais pontos referentes a Inteligência Artificial, destacando, métodos, funcionamento, aplicações e ramificações, como sendo a Inteligência Artificial propriamente dita, o Machine Learning e o Deep Learning (ramo que estuda as redes neurais artificiais), com maior ênfase em Deep Learning.

2.1 INTELIGÊNCIA ARTIFICIAL

A inteligência artificial é a capacidade de um computador realizar tarefas que normalmente requerem inteligência humana, sendo o Machine Learning a principal forma de alcançar esse resultado, em que o computador é capaz de aprender a realizar uma determinada tarefa a partir de dados e/ou instruções sem ser explicitamente programado. Porém, existe um subconjunto do Machine Learning chamado de Deep Learning, em que são utilizadas redes neurais artificiais (algoritmos com estrutura lógica que se baseiam no funcionamento do cérebro, sendo capazes de reconhecer padrões automaticamente). O Deep Learning é uma subárea do Machine Learning, assim como o Machine Learning é uma subárea da Inteligência artificial.

A principal diferença prática entre o Machine Learning e Deep Learning é que em geral o Machine Learning precisa de intervenção humana manual para a seleção de recursos a serem processados (entregar ao algoritmo os dados que possam ser explorados em busca de padrões), enquanto o Deep Learning é intuitivo, sendo capaz de realizar essa tarefa automaticamente. Devido às diferenças, identifica-se que o Machine Learning pode obter melhor acurácia para um conjunto de dados pequenos, porém o Deep Learning obtém vantagem para um grande conjunto de dados, pois além da intervenção humana se tornar inviável, as redes neurais (campo de estudo do Deep Learning) serão mais eficientes que a intervenção humana (LECON, 2022).

O Deep Learning é um dos temas mais notáveis da atualidade devido aos resultados inigualáveis em aplicações como visão computacional, reconhecimento de fala e compreensão de linguagem natural (LECON, 2022).

2.2 DEEP LEARNING

Visto os principais conceitos sobre Inteligência Artificial e Machine Learning, como os principais tipos, algoritmos mais populares e aplicações, as redes neurais (campo de estudo do Deep Learning) serão abordadas com mais profundidade nesta seção, contemplando seu funcionamento, diferentes tipos de redes neurais, funções de ativação, funções de custo e otimizadores. O Deep Learning foi escolhido para ser abordado com mais detalhes devido às suas propriedades auto aprendizado e excelente performance para temas modernos como visão computacional, reconhecimento de fala e compreensão de linguagem natural.

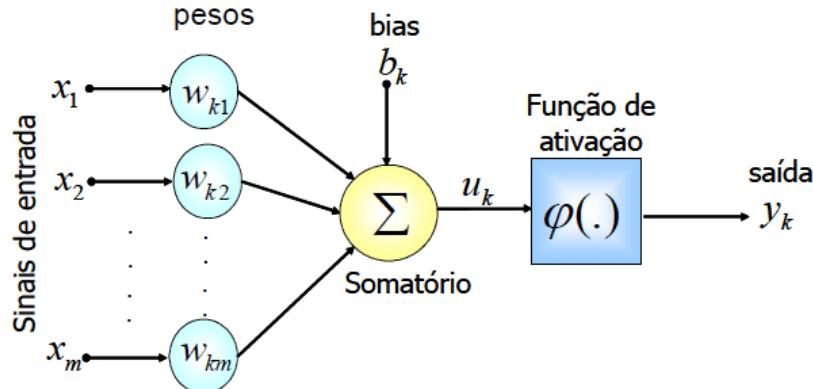
2.2.1 Introdução às redes neurais

Inicialmente, será introduzido o conceito de redes neurais, contemplando desde o modelo de um neurônio artificial, rede Perceptron, função de custo, método da descida do gradiente e backpropagation.

2.2.1.1 Modelo de um neurônio artificial

Uma rede neural artificial é formada por um grande número de neurônios para funcionar corretamente, mas para compreender o funcionamento de uma rede neural, deve-se definir o modelo de um único neurônio artificial. Esse modelo é apresentado na Figura 2.1.

Figura 2.1 - Modelo de um neurônio artificial



Fonte: (BRAGA e SILVA, 2011).

Os passos para a obtenção da saída de um neurônio artificial são:

1. O modelo recebe um número m de entradas x_1, x_2, \dots, x_m ;
2. Cada uma dessas entradas é multiplicada por um peso: $x_1 w_1, x_2 w_2, \dots, x_m w_m$;
3. Somam-se as entradas multiplicadas pelos seus respectivos pesos: $\sum_{n=1}^m w_n x_n$;
4. Adiciona-se o bias: $b + \sum_{n=1}^m w_n x_n$;
5. O resultado passa por uma função de ativação: $\varphi(b + \sum_{n=1}^m w_n x_n)$.

Seguidos os passos, a equação de saída de um neurônio artificial é descrita pela Equação (2.1).

$$y = \varphi \left(b + \sum_{n=1}^m w_n x_n \right) \quad (2.1)$$

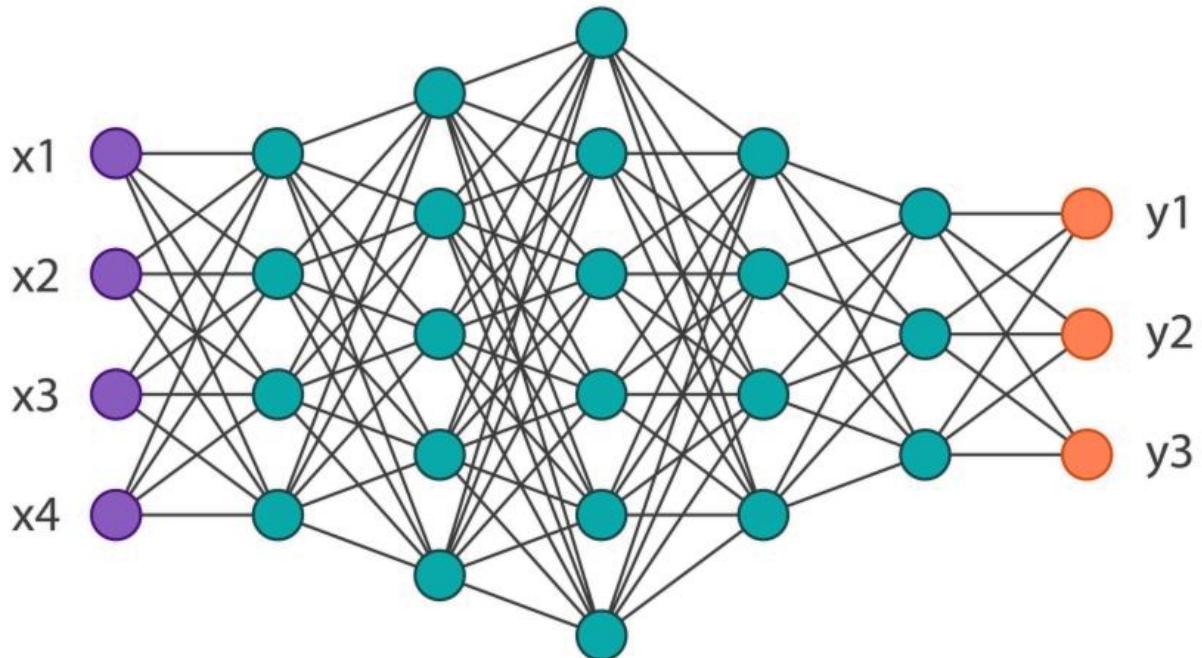
Visto o modelo de um único neurônio artificial, o conceito de rede neural composta pela associação de diversos neurônios é descrito a seguir.

2.2.1.2 Modelo de neurônio equivalente à porta lógica NAND

Um único neurônio isolado não possui aplicabilidade significativa, por isso, são utilizadas redes de neurônios, em que a saída de um neurônio de uma camada da rede neural serve como entrada para outro neurônio da camada seguinte e assim por diante.

A Figura 2.2 apresenta uma rede perceptron de 5 camadas escondidas, 4 entradas (x_1, x_2, x_3 e x_4) e 3 neurônios de saída (y_1, y_2 e y_3).

Figura 2.2 - Rede perceptron



Fonte: Adaptado de (HENRIQUE, 2018).

A rede mais simples é chamada de rede perceptron, na qual todos os neurônios de uma camada estão ligados a todos os outros neurônios da camada anterior e da camada seguinte (VEEN, 2016).

Uma rede perceptron é capaz de representar qualquer função lógica. Para provar isso, basta determinar o modelo de um neurônio que represente a porta lógica NAND, pois se trata de uma lógica universal e com a utilização de múltiplas portas NAND pode-se representar qualquer outra porta lógica.

Para simplificar, considerando um neurônio de duas entradas, deve-se determinar os pesos de cada uma dessas entradas, o bias e a função de ativação:

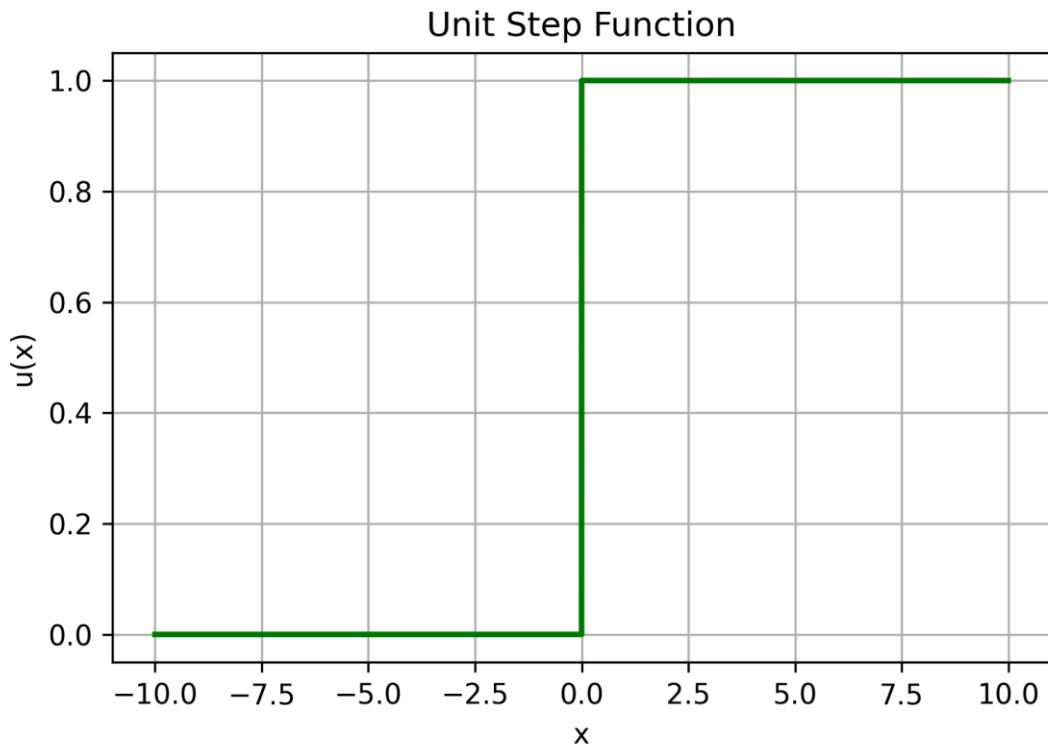
1. Definição dos valores dos pesos: $w_1 = -2$, $w_2 = -2$;
2. Definição do bias: $b = 1$;
3. Escolha da função de ativação: $\varphi(x) = u(x)$.

A função de ativação escolhida foi a função degrau unitário, que fará com que a saída do neurônio seja 0 caso sua entrada seja negativa e 1 caso a entrada seja positiva. Essa função é representada na Figura 2.3.

Onde:

- x : variável de entrada;
- $u(x)$: saída da função degrau.

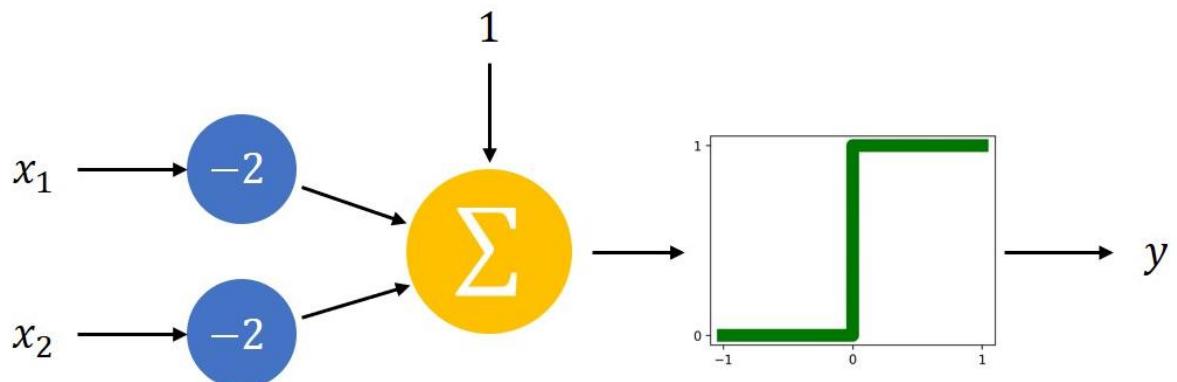
Figura 2.3 – Função degrau unitário



Fonte: Autor.

O modelo do neurônio que representa a porta lógica NAND é apresentado através da Figura 2.4.

Figura 2.4 - Modelo do neurônio da porta lógica NAND



Fonte: Autor.

Para testar se o modelo funciona corretamente, a tabela verdade apresentada na Figura 2.5 contém os possíveis valores de entrada e sua respectiva saída e a Equação (2.2) descreve a saída do neurônio em função da entrada.

Figura 2.5 – Tabela verdade da porta NAND

x_1	x_2	$b + \sum_{n=1}^m x_n w_n$	y
0	0	$3 + 0(-2) + 0(-2) = 3$	$u(3) = 1$
0	1	$3 + 0(-2) + 1(-2) = 1$	$u(1) = 1$
1	0	$3 + 1(-2) + 0(-2) = 1$	$u(1) = 1$
1	1	$3 + 1(-2) + 1(-2) = -1$	$u(-1) = 0$

Fonte: Autor.

$$y = u(3 - 2x_1 - 2x_2) \quad (2.2)$$

Pode-se perceber que foi possível criar uma porta lógica NAND através do modelo de um neurônio, o que significa que através de múltiplos neurônios conectados como uma rede perceptron, pode-se representar qualquer expressão lógica.

2.2.2 Tipos de redes neurais

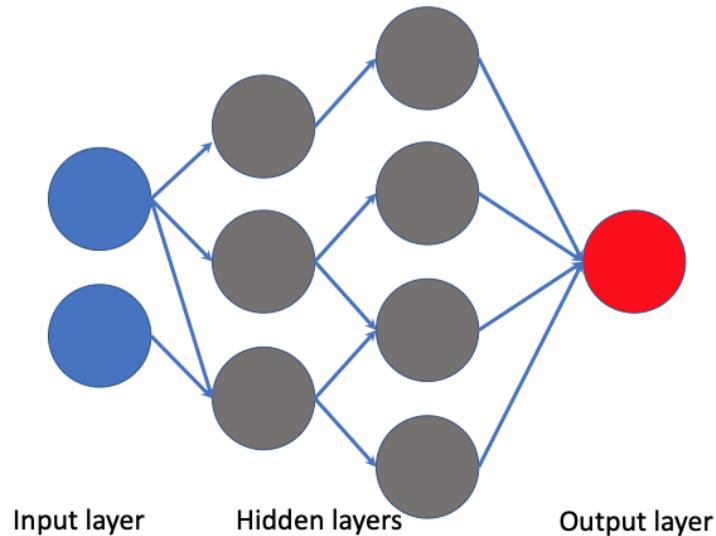
Existem diversos tipos de redes neurais que se distinguem tanto em termos de seus princípios de funcionamento quanto em suas aplicações práticas específicas. A seguir, serão discutidos alguns desses tipos de redes neurais de maneira individualizada, a fim de fornecer uma compreensão mais aprofundada sobre seu funcionamento e suas aplicações.

2.2.2.1 Perceptron (P), Feed Forward Network (FFN)

FFNs são o tipo mais básico de rede neural, em que a informação flui linearmente até a saída e cada neurônio realiza uma operação matemática linear do tipo $wx + b$, sendo x o valor de entrada, w o peso e b o bias do neurônio. O resultado passa por uma função de ativação e em seguida é enviado para a próxima camada.

As redes neurais do tipo FFN (exemplo mostrado através da Figura 2.6) possuem conexões em apenas um único sentido, geralmente limitadas a 5 camadas.

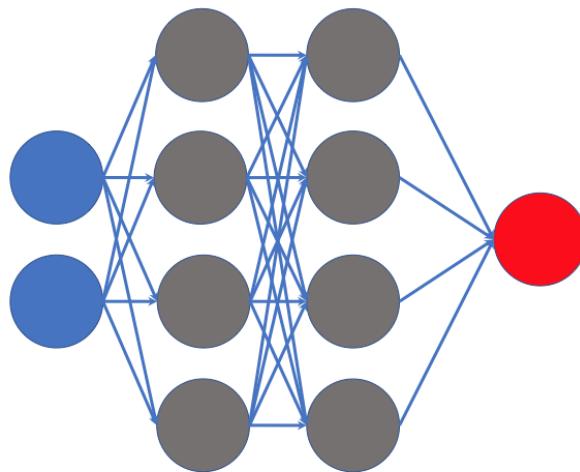
Figura 2.6 - FFN



Fonte: (ALEX, 2020).

As redes do tipo P, Figura 2.7, são um caso especial de uma rede FFN, em que todos os neurônios de uma camada são conectados com todos neurônios da camada seguinte.

Figura 2.7 – P



Fonte: (ALEX, 2020).

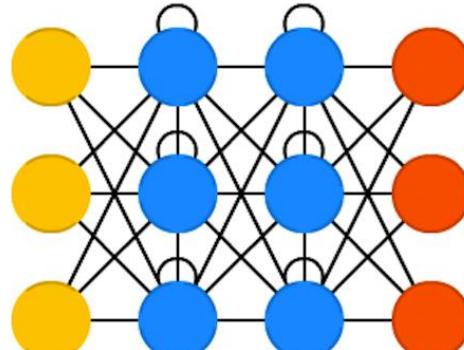
As FFNs são utilizadas para problemas em que os dados de entrada têm impacto atemporal nos dados de saída, em que a saída não depende do estado anterior da rede neural. Um exemplo é usar informações de um exame de sangue para determinar a presença de uma doença.

2.2.2.2 Recurrent Neural Network (RNN)

Além dos dados da camada anterior, os neurônios escondidos de uma RNN também recebem a sua própria saída anterior como um valor de entrada criando uma dependência temporal entre os dados de entrada. Por isso, são aplicadas em problemas com característica temporal como previsão do tempo dado o histórico climático passado.

A representação da RNN é apresentada através da Figura 2.8.

Figura 2.8 – RNN

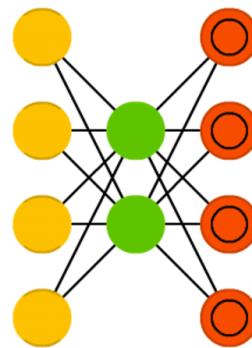


Fonte: (VEEN, 2016).

2.2.2.3 Auto-encoder (AE)

As redes do tipo AE (Figura 2.9) possuem a mesma arquitetura de uma FFN, porém possuem menos neurônios na camada escondida em relação às camadas de entrada e saída, fazendo com que os dados sejam primeiramente compactados e depois descompactados.

Figura 2.9 - AE

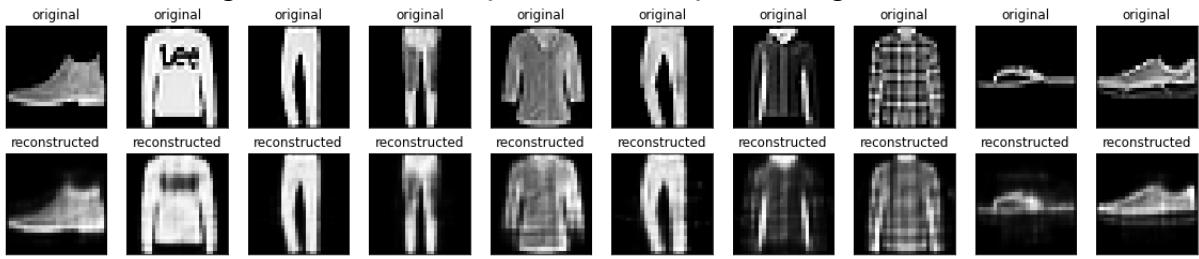


Fonte: (VEEN, 2016).

Os AEs são sempre simétricos (formato de ampulheta), pois os dados de entrada são iguais aos dados de saída e o objetivo é conseguir representar a mesma informação de entrada com um menor número de dados. Um exemplo é comprimir uma imagem para uma resolução menor sem perder muita informação.

A Figura 2.10 apresenta a codificação e decodificação de imagens realizada por uma rede neural AE, em que a as imagens de entrada e saída têm 28x28 pixels de resolução.

Figura 2.10 – Codificação e decodificação de imagens com AE

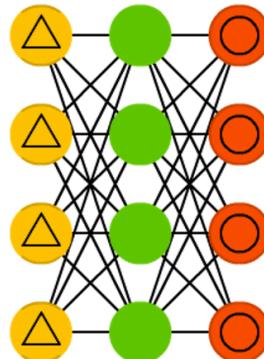


Fonte: (Introdução aos codificadores automáticos, 2022).

2.2.2.4 Denoising auto-encoder (DAE)

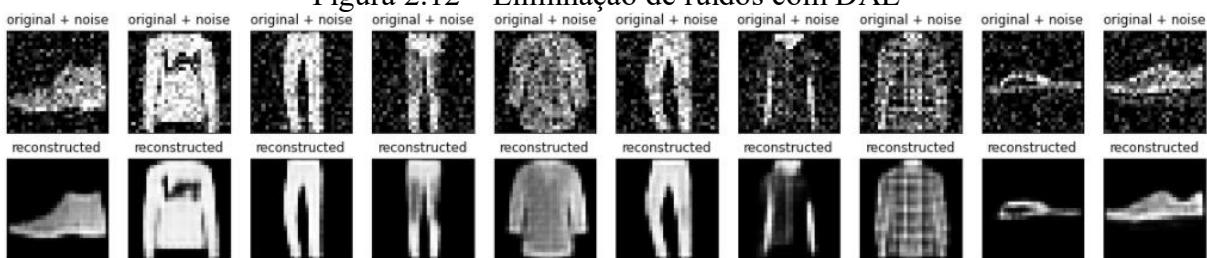
Redes do tipo DAE, mostrada na Figura 2.11, são AEs, porém ao invés dos dados de entrada serem iguais aos de saída, é adicionado um ruído aos dados de entrada, fazendo com que a DAE seja treinada para eliminar ruídos conforme ilustra a Figura 2.12.

Figura 2.11 – DAE



Fonte: (VEEN, 2016).

Figura 2.12 – Eliminação de ruídos com DAE



Fonte: (Introdução aos codificadores automáticos, 2022).

Uma aplicação é a remasterização de imagens (aumentar a resolução de imagens).

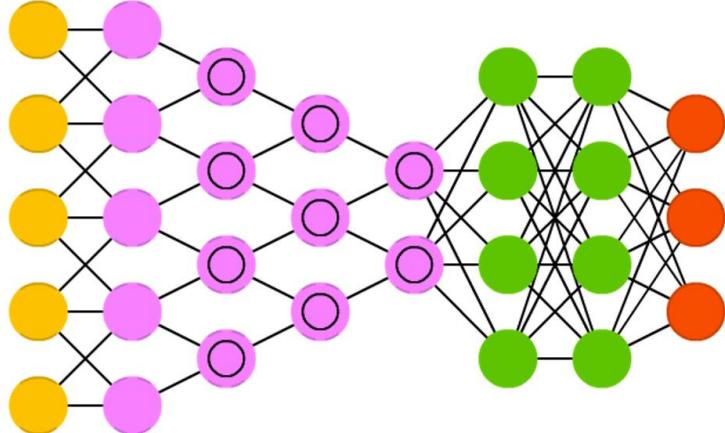
2.2.2.5 Convolutional neural network (CNN) ou Deep convolutional network (DCN)

As redes neurais estudadas até então, não consideram uma relação de vizinhança entre os dados de entrada. Por exemplo, não faria a menor diferença se antes do treinamento a posição de todos os dados de entrada fossem embaralhadas da mesma forma. Porém, essa relação de vizinhança pode ser muito importante para alguns casos específicos como no reconhecimento de imagens, reconhecimento de voz, análise grafista do mercado financeiro, etc. No

reconhecimento de imagens, por exemplo, grande parte da informação está contida na relação de vizinhança dos pixels como o contraste e a textura.

Uma CNN, cuja estrutura está representada através da Figura 2.13, percebe uma imagem como uma caixa retangular cuja largura e altura são medidas pelo número de pixels da imagem e a profundidade é representada por cada uma das três camadas de cores, vermelho, verde e azul referidas como canais.

Figura 2.13 – CNN, DCN



Fonte: (VEEN, 2016).

Ao longo das camadas de uma rede CNN, as dimensões da imagem se alteram, pois a medida em que a altura e largura da imagem diminuem, o número de canais aumenta, reduzindo o volume de dados. Esse processo é chamado de pooling, que faz um resumo dos dados através do descarte das saídas menos significativas, mantendo somente às de maior valor.

O processo de convolução é realizado arrastando uma janela (kernel) de dimensão menor pela imagem original, sendo essa janela uma rede FFN.

Por exemplo, se uma imagem 5x5 pixels passar pelo processo da convolução e supondo uma janela de 3x3 com passo 1 (stride), primeiramente os 3x3 pixels do canto superior esquerdo da imagem original passarão por uma FFN. Em seguida, essa janela é arrastada 1 pixel (tamanho do passo) para a direita e o processo se repete até ao final da imagem.

2.2.3 Funções de ativação

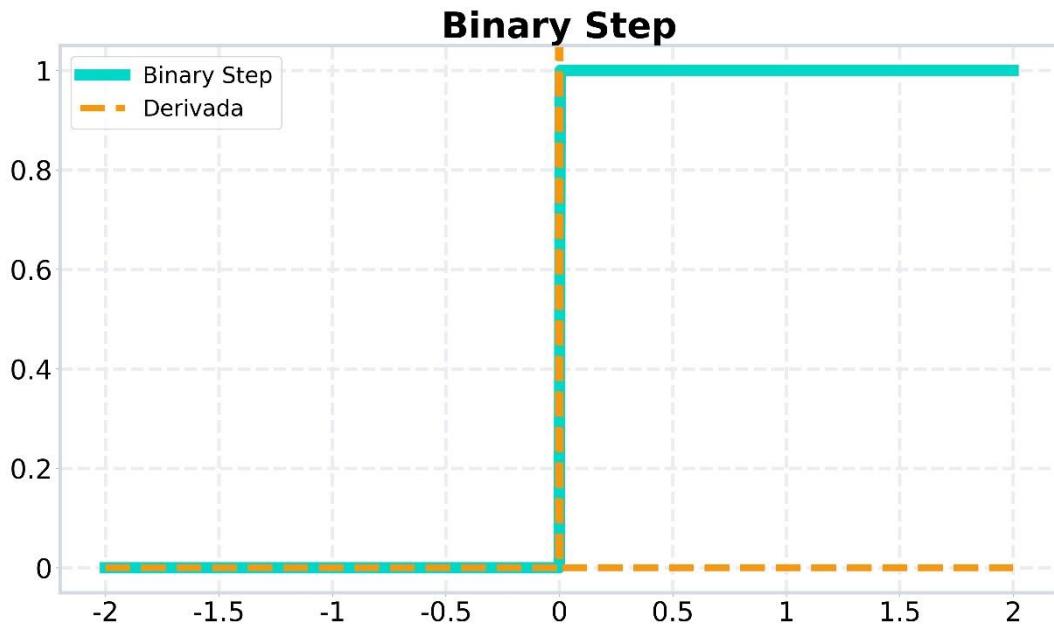
Outro conceito importante a ser definido sobre redes neurais são as funções de ativação. Seu objetivo se dá pela linearidade das redes neurais, que são compostas por neurônios que multiplicam seus valores de entrada por pesos, que em seguida são somados com o bias. Essas operações são lineares, logo, se o resultado for transferido para o próximo neurônio diretamente, a rede neural seria capaz de representar apenas relações lineares, não importando a quantidade de camadas, pois a composição de duas ou mais funções lineares resulta em uma função linear. Embora a rede neural com apenas funções lineares seja mais simples, não seria possível resolver problemas mais complexos e o modelo seria apenas um modelo de regressão linear. Por esse motivo, o objetivo de uma função de ativação é adicionar a “não linearidade” à rede neural.

Existem várias funções de ativação com aplicações diversas e uma única rede neural pode ter diferentes funções de ativação em cada camada. Algumas usualmente aplicadas nas camadas escondidas (camadas do meio da rede neural) e outras nas camadas de saída (última camada da rede neural). As principais são apresentadas a seguir, onde x representa a variável de entrada de cada função de ativação e $f(x)$ representa a saída.

2.2.3.1 Binary Step

A Figura 2.14 contém a função Binary Step expressa em forma de gráfico.

Figura 2.14 – Binary Step



Fonte: Autor.

A função degrau binária é capaz de decidir se um neurônio deve ser ativado ou desativado.

A representação matemática da função é apresentada através da Equação (2.3).

$$f(x) = \begin{cases} 0 & \text{se } x < 0 \\ 1 & \text{se } x \geq 0 \end{cases} \quad (2.3)$$

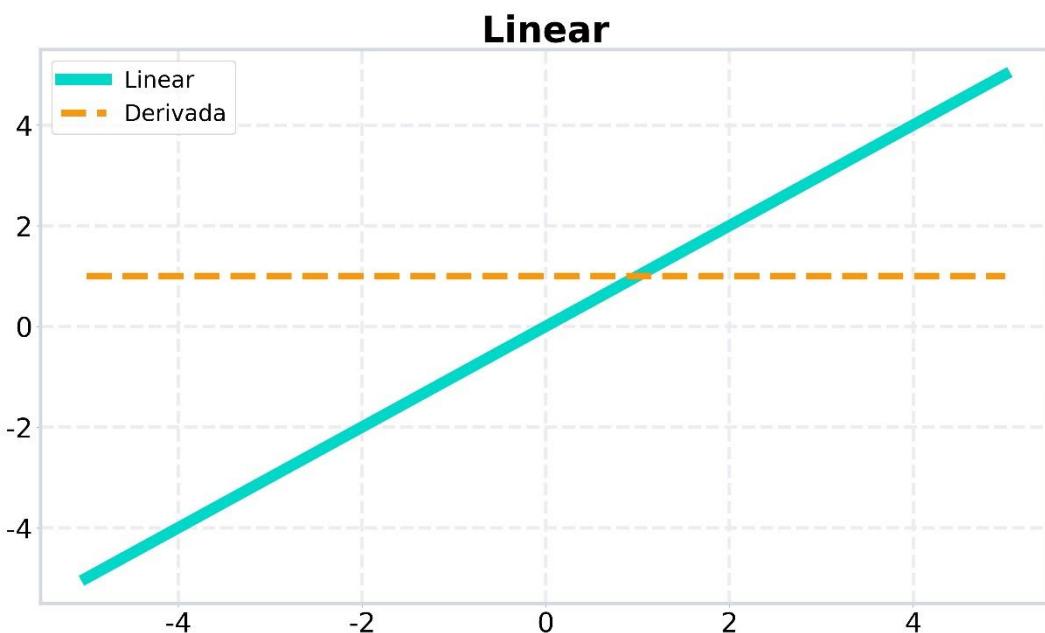
Limitações:

- Não pode ser usada para problemas de classificação de várias classes;
- O gradiente da função degrau é zero, dificultando o backpropagation.

2.2.3.2 Linear

O gráfico da função Linear está presente na Figura 2.15.

Figura 2.15 – Linear



Fonte: Autor.

A função de ativação linear também é conhecida como “sem ativação” ou “função de identidade”, pois é o equivalente a transferir o valor de um neurônio adiante sem passar por uma função de ativação.

A representação matemática da função é apresentada através da Equação (2.4).

$$f(x) = x \quad (2.4)$$

Limitações:

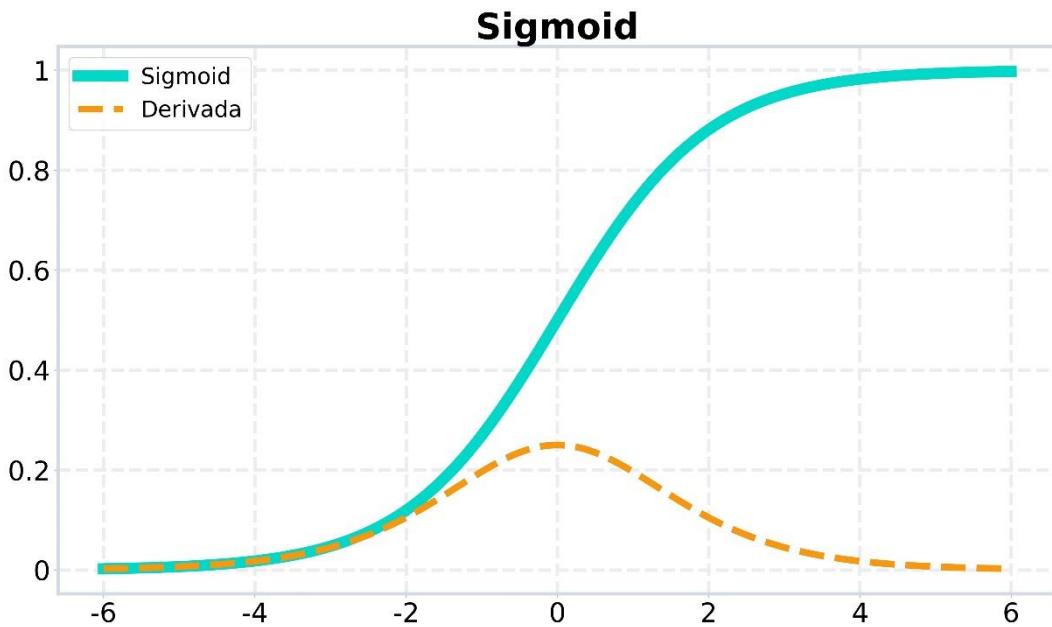
- O backpropagation não tende a convergir para produzir um erro estável próximo de zero, pois a derivada da função é constante e não se relaciona com a entrada;
- A saída sempre será uma função linear da primeira camada, ou seja, independentemente do tamanho da rede, se essa função for utilizada em todas as camadas a rede neural funcionará como se tivesse apenas uma única camada.

Devido às suas limitações, essa função não deve ser utilizada nas camadas escondidas, mas é a escolha ideal para a camada de saída em problemas de regressão, pois relaciona a entrada e saída de forma proporcional além de conter todo o conjunto de números reais tanto no domínio quanto na imagem.

2.2.3.3 Sigmoid

A Figura 2.16 expressa a função Sigmoid em um gráfico.

Figura 2.16 – Sigmoid



Fonte: Autor.

Essa função transforma qualquer entrada em valores entre 0 e 1, sendo que quanto menor entrada, mais próxima a saída estará de 0 e quanto maior a entrada, mais próxima a saída estará de 1. Além disso, a função sigmoid pode representar uma probabilidade como saída devido ao seu intervalo entre 0 e 1. Ela também é diferenciável e fornece um gradiente suave, evitando saltos nos valores de saída.

A representação matemática da função é apresentada através da Equação (2.5).

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.5)$$

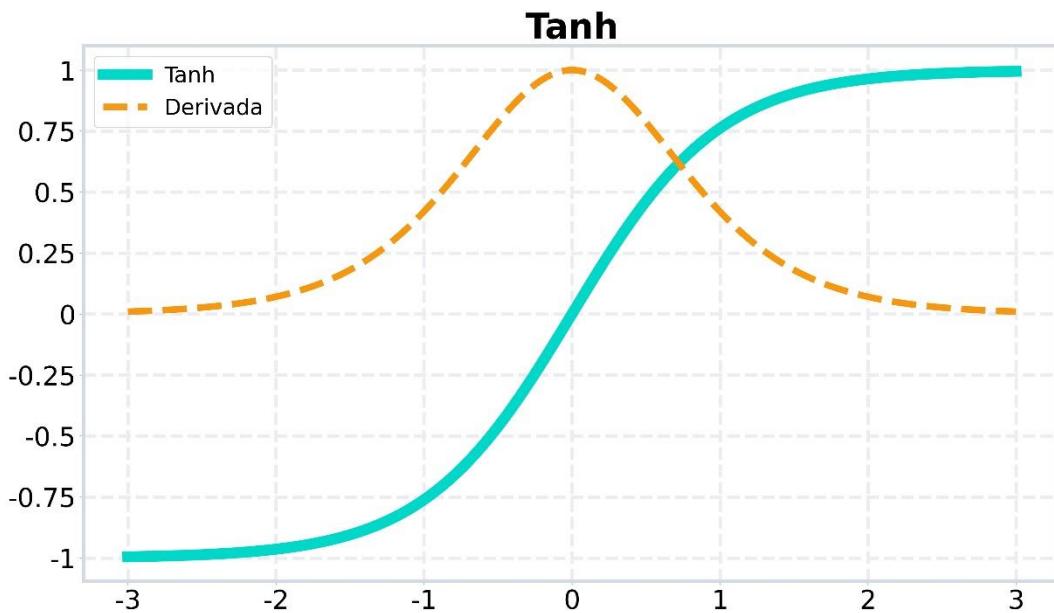
Limitações:

- A função possui gradientes muito pequenos para valores maiores do que 3 ou menores do que -3, devido à sua derivada, logo, o treinamento (baseado no gradiente) surtirá um efeito muito pequeno fora desse intervalo de valores, prejudicando o aprendizado da rede neural.

2.2.3.4 Tanh

A Figura 2.17 contém um gráfico da função Tanh.

Figura 2.17 – Tanh



Fonte: Autor.

A função tanh é semelhante à sigmoid, porém possui uma faixa de valores de -1 a 1 ao invés de 0 a 1.

A representação matemática da função é apresentada através da Equação (2.6).

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.6)$$

Vantagens:

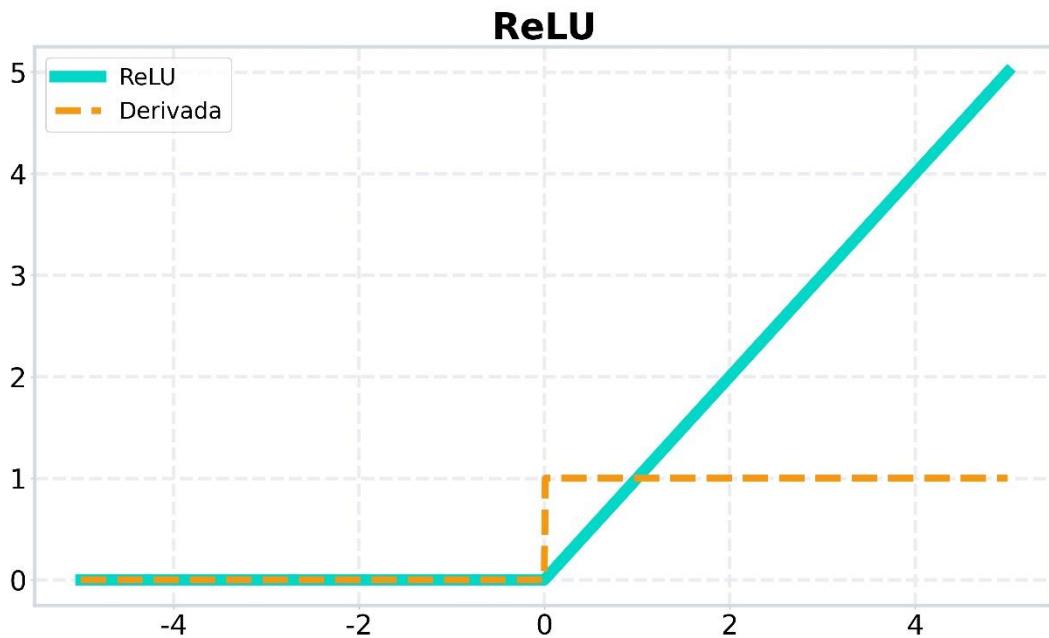
- A saída é centrada em 0, permitindo o mapeamento dos valores de saída como negativos, neutros ou positivos;
- Apresenta um bom funcionamento em camadas ocultas, pois a faixa de valores entre -1 e 1 juntamente com a média 0 facilitam o aprendizado da camada seguinte.

Apesar da função tanh apresentar o problema do gradiente de fuga (explicado na seção 2.2.5) como a função sigmoid, o centro em 0 permite que os gradientes não se restrinjam a se mover em uma única direção, portanto, a não linearidade tanh é preferível à não linearidade sigmoid.

2.2.3.5 ReLU

A ilustração apresentada na Figura 2.18 é um gráfico da função ReLU.

Figura 2.18 – ReLU



Fonte: Autor.

A função ReLU (Rectified Linear Unit) aparenta ser linear, porém ela possui uma derivada, permitindo o backpropagation, porém essa função só deve ser utilizada em camadas ocultas.

A representação matemática da função é apresentada através da Equação (2.7).

$$f(x) = \max(0, x) \quad (2.7)$$

Vantagens:

- É muito eficiente computacionalmente comparada às funções sigmoid e tanh;
- Acelera a convergência da descida do gradiente para o mínimo global da função de custo devido à sua propriedade linear e não saturante (CECCON, 2020).

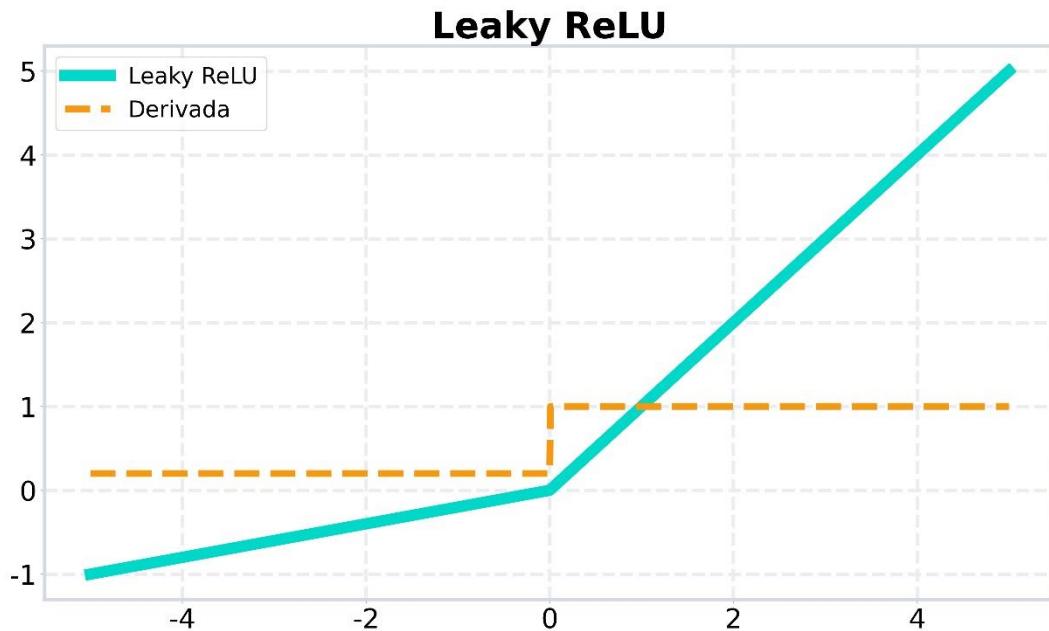
Limitações:

- Durante o backpropagation, ocorre um problema chamado Dying ReLU, em que os pesos e bias de alguns neurônios não são atualizados, podendo criar neurônios mortos que nunca são ativados, pois o lado negativo da derivada da função é 0;
- Valores de entrada negativos tornam-se 0 imediatamente, perdendo parte da informação.

2.2.3.6 Leaky ReLU

O gráfico da função Leaky ReLU é apresentado na Figura 2.19.

Figura 2.19 – Leaky ReLU



Fonte: Autor.

A função Leaky ReLU é uma versão melhorada da função ReLU para resolver o problema Dying ReLU, pois não possui uma derivada nula para valores de entrada menores do que 0.

A representação matemática da função é apresentada através da Equação (2.8).

$$f(x) = \max(0.1x, x) \quad (2.8)$$

Vantagens:

- Possibilita o backpropagation para valores negativos;
- É muito eficiente computacionalmente comparada às funções sigmoid e tanh;
- Acelera a convergência da descida do gradiente para o mínimo global da função de custo devido à sua propriedade linear e não saturante.

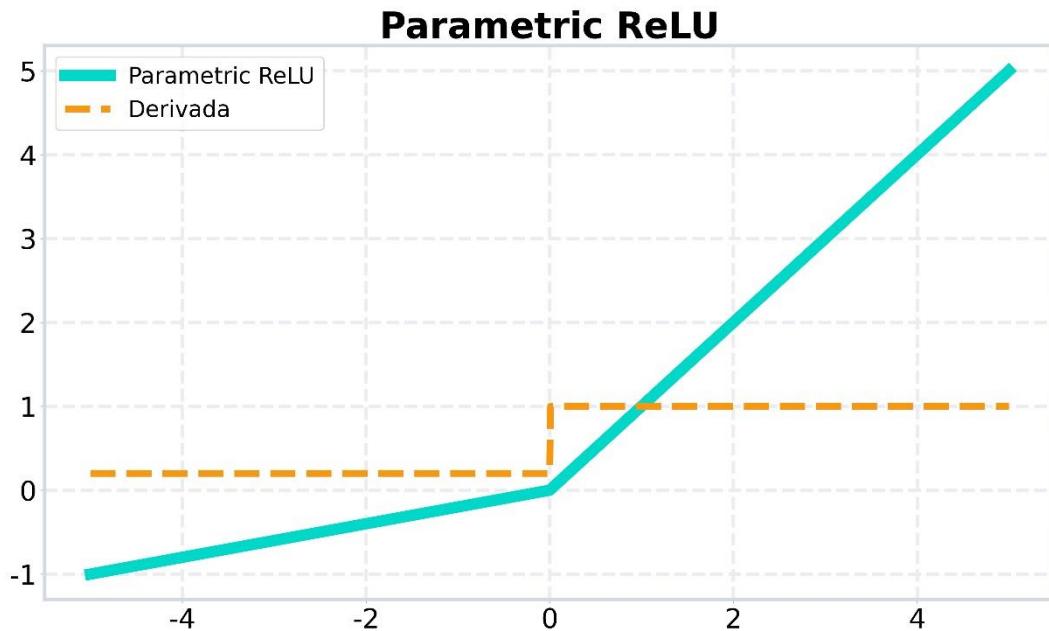
Limitações:

- Podem haver inconsistências na previsão para valores de entrada negativos;
- O gradiente para valores negativo é pequeno, o que torna o aprendizado demorado.

2.2.3.7 Parametric ReLU

A Figura 2.20 fornece uma visualização da função Parametric ReLU.

Figura 2.20 – Parametric ReLU



Fonte: Autor.

A função ReLU paramétrica é uma variação da função ReLU com o objetivo de corrigir o problema do gradiente nulo para valores negativos. Essa função pode ser usada quando a Leaky ReLU ainda apresenta neurônios mortos, que faz com que informações relevantes não sejam passadas para as próximas camadas.

A representação matemática da função é apresentada através da Equação (2.9).

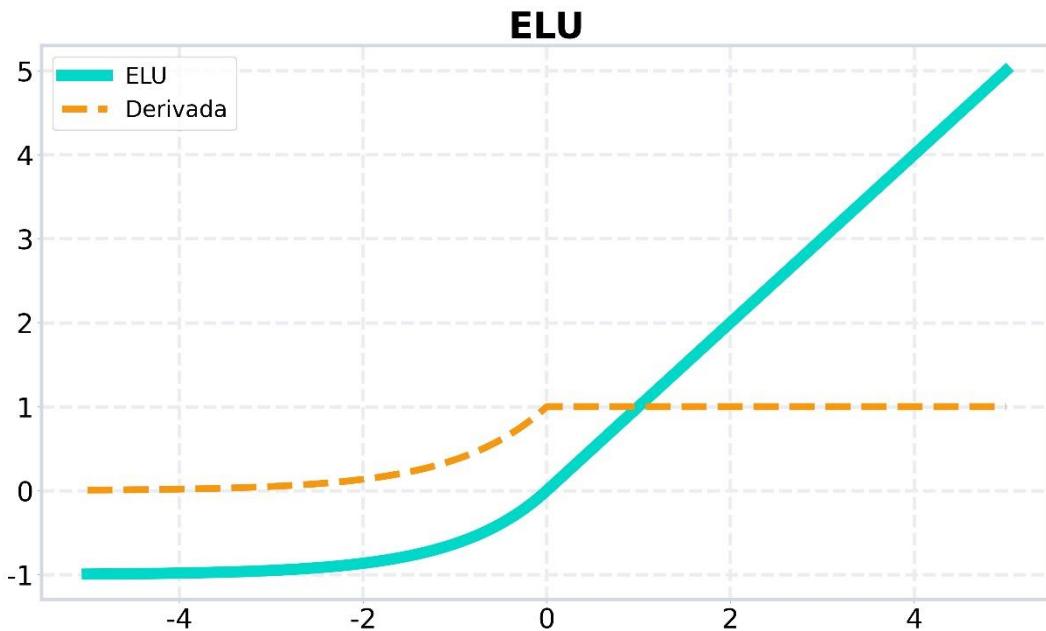
$$f(x) = \max(ax, x) \quad (2.9)$$

Em que “a” é a inclinação da reta para valores negativos.

2.2.3.8 Exponential Linear Units (ELU)

Uma representação gráfica da função Exponential Linear Units é mostrada na Figura 2.21.

Figura 2.21 – Exponential Linear Units



Fonte: Autor.

A função ELU é uma variação da função ReLU que modifica a inclinação para valores de entrada negativos, que utiliza uma curva logarítmica para definir valores de entrada menores do que 0.

A representação matemática da função é apresentada através da Equação (2.10).

$$f(x) = \begin{cases} x & \text{se } x \geq 0 \\ \alpha(e^x - 1) & \text{se } x < 0 \end{cases} \quad (2.10)$$

Vantagens:

- Apresenta uma curva suave até o limite inferior $-\alpha$. Isso pode ser considerado uma vantagem, pois uma mudança brusca dificulta a calibragem. Por exemplo: se uma torneira (cuja vazão é nula para um ângulo de abertura de 0° e máxima para um ângulo de abertura de 360°) apresentar um comportamento suave, a regulagem da vazão é fácil de ser executada, porém seria extremamente difícil realizar a regulagem caso houvesse uma mudança brusca da vazão do ângulo 180° para o ângulo 181° ;
- Evita o problema de neurônios mortos através da curva logarítmica para valores de entrada negativos.

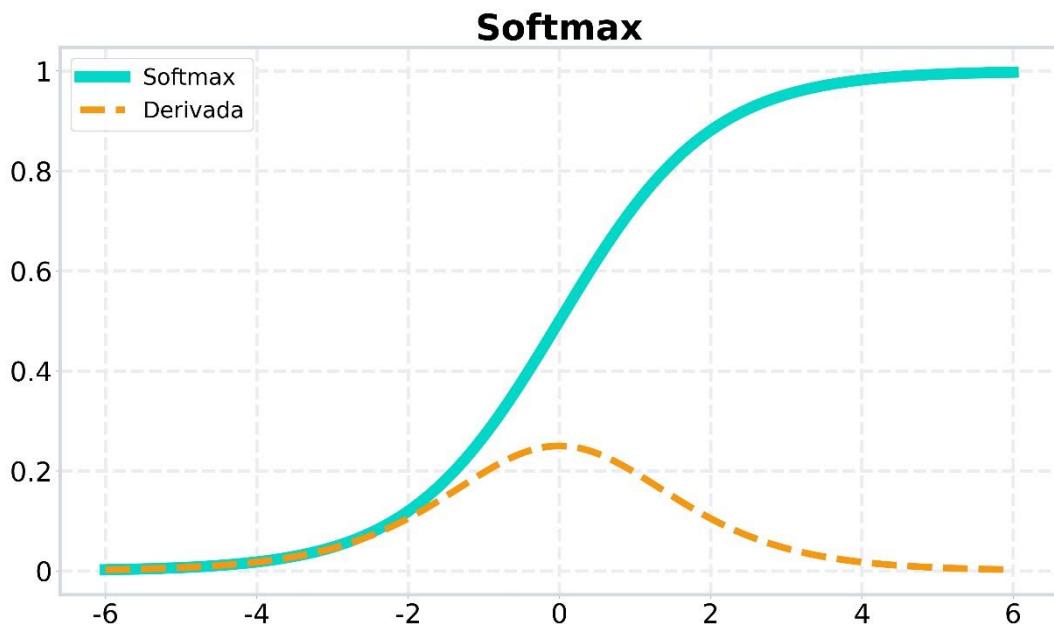
Limitações:

- Maior custo computacional devido a operação exponencial;
- O valor de α é fixo e não passa por um aprendizado;
- Pode apresentar o problema da explosão do gradiente (explicado na seção 2.2.5).

2.2.3.9 Softmax

A Figura 2.22 representa a função Softmax graficamente.

Figura 2.22 – Softmax



Fonte: Autor.

A função softmax é uma extensão da função sigmoid. Enquanto a sigmoid funciona somente para classificações binárias (dois valores de saída), a função softmax não sofre essa limitação e pode ser utilizada em problemas com múltiplas classes.

A função acentua o maior valor e transforma as saídas da rede neural na probabilidade de que o resultado final seja cada uma das classes.

Geralmente, a softmax é utilizada na última camada da rede neural para problemas de classificação multiclasse (problemas de classificação com mais de duas possibilidades de classificação, por exemplo determinar se a foto de um animal é um gato, um cachorro ou nenhum dos dois).

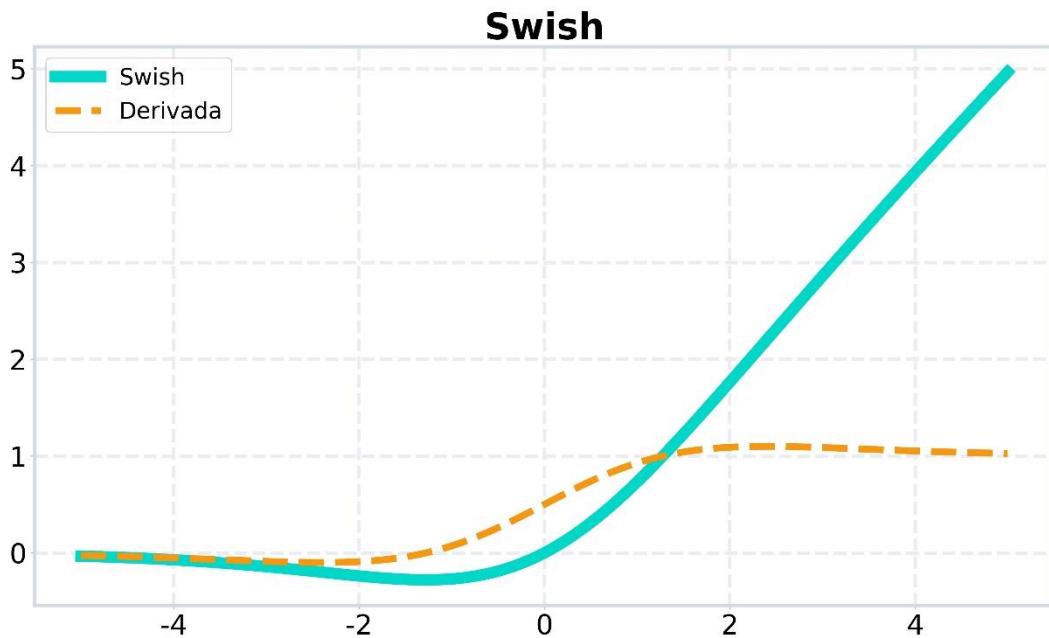
A representação matemática da função é apresentada através da Equação (2.11).

$$f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \quad (2.11)$$

2.2.3.10 Swish

A Figura 2.23 é uma forma gráfica da função Swish.

Figura 2.23 – Swish



Fonte: Autor.

A função Swish corresponde ou supera a função ReLU em redes neurais em domínios complexos como classificação de imagens, tradução automática, etc.

A representação matemática da função é apresentada através da Equação (2.12).

$$f(x) = \frac{x}{1 + e^{-x}} \quad (2.12)$$

Vantagens:

- Apresenta suavidade;
- Considera valores negativos;
- Não é monótona, melhorando a expressão dos dados de entrada e o aprendizado.

Limitações:

- A função Swish só deve ser utilizada em redes neurais com profundidade superior a 40 camadas.

A função de ativação deve ser escolhida com base no problema a ser resolvido (tipo de variável prevista). Como regra geral, pode-se começar utilizando a função de ativação ReLU (devido ao seu desempenho) e testar outras funções caso a ReLU não forneça resultados ideais.

Regras de escolha da função de ativação de camadas de saída com base no problema a ser resolvido:

- Regressão: Linear;
- Classificação binária: Sigmoid;
- Classificação multiclasse: Softmax;
- Classificação multirótulo: Sigmoid.

Regras de escolha da função de ativação de camadas ocultas com base no problema a ser resolvido:

- Rede neural convolucional: ReLU;
- Rede neural recorrente: Tanh/Sigmoid.

2.2.4 Funções de custo

As funções de custo (também chamadas de erro ou funções de perda) são um dos aspectos mais importantes das redes neurais, pois são responsáveis por ajustar o modelo aos dados de treinamento fornecidos juntamente com as funções de otimização. Elas medem a acurácia do modelo e retornam o quanto distantes estão os resultados de saída comparados aos resultados esperados, logo, o objetivo é minimizar a função de custo, pois ao se aproximar do custo zero, a rede neural se aproxima da precisão máxima.

Existem dois tipos principais de função de custo no aprendizado supervisionado relacionados aos dois principais tipos de redes neurais: funções de regressão e classificação. As funções de custo de regressão são usadas em modelos regressão, em que a rede neural prevê um valor numérico como saída ao invés de rótulos. Já as funções de custo de classificação, são usadas em modelos de classificação, em que a rede neural prevê a probabilidade da entrada pertencer a cada uma das categorias, permitindo que seja selecionada a categoria de maior probabilidade.

A seguir serão apresentadas as funções de custo mais populares bem como suas aplicações.

2.2.4.1 Erro Médio Quadrático (MSE)

Uma das funções de custo mais populares, o MSE encontra a média das diferenças quadradas entre as saídas previstas e as saídas esperadas.

A representação matemática da função é apresentada através da Equação (2.13).

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.13)$$

Legenda das equações:

- y_i : saída desejada;
- \hat{y}_i : saída da rede neural;
- n : número de saídas da rede neural.

Essa função eleva a diferença entre as saídas previstas e esperadas ao quadrado, penalizando valores com erro elevado não importando se está acima ou abaixo do valor esperado. Sendo o MSE uma função convexa, o mínimo global é claramente definido, permitindo o uso facilitado da otimização da descida do gradiente para definir pesos e bias.

Em alguns casos, o quadrado das diferenças pode ser uma desvantagem, devido a sua sensibilidade a valores discrepantes, fazendo com que o erro aumente significativamente.

2.2.4.2 Erro Médio Absoluto (MAE)

A função de custo MAE calcula a média das diferenças absolutas entre o as saídas previstas e as saídas esperadas.

A representação matemática da função é apresentada através da Equação (2.14).

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.14)$$

Essa função é uma alternativa à MSE em alguns casos, pois não eleva a diferença entre as previsões e saídas desejadas ao quadrado, fazendo com que a função não seja tão sensível para valores discrepantes. Essa característica torna o MAE mais adequado para casos em que os dados de treinamento possuem vários valores atípicos.

A função MSE possui a desvantagem de que à medida que a distância média se aproxima de 0, a otimização da descida do gradiente não funcionará, pois a derivada da função em 0 é indefinida.

2.2.4.3 Huber

A função de custo Huber foi desenvolvida com o propósito de unir as vantagens das funções MSE e MAE além de evitar suas desvantagens.

A representação matemática da função é apresentada através da Equação (2.15).

$$\text{Huber Loss} = \begin{cases} \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 & \text{se } |y_i - \hat{y}_i| \leq \delta \\ \frac{1}{n} \sum_{i=1}^n \delta \left(|y_i - \hat{y}_i| - \frac{1}{2} \delta \right) & \text{se } |y_i - \hat{y}_i| > \delta \end{cases} \quad (2.15)$$

Huber aplica a função de custo MAE para erros grandes, evitando a sensibilidade para valores discrepantes e aplica a função de custo MSE para valores pequenos (menores que um limite δ), evitando a derivada indefinida para valores próximos de zero.

2.2.4.4 Entropia Cruzada Binária/Perda de Log

A Entropia Cruzada Binária é utilizada em modelos de classificação binária, em que o modelo classifica uma entrada em uma de duas categorias predefinidas.

A representação matemática da função é apresentada através da Equação (2.16).

$$CELoss = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (2.16)$$

2.2.4.5 Entropia Cruzada Categórica

A Entropia Cruzada Categórica é utilizada em modelos de classificação com mais de duas categorias predefinidas. A Entropia Cruzada Binária é um caso especial da entropia Cruzada Categórica, onde $n = 2$, sendo n o número de categorias.

A representação matemática da função é apresentada através da Equação (2.17).

$$CELoss = -\frac{1}{n} \sum_{j=1}^n y_{ij} \log(\hat{y}_{ij}) \quad (2.17)$$

2.2.5 Otimizadores

Os otimizadores são algoritmos que permitem ajustar os parâmetros de um modelo de aprendizado de máquina para minimizar a função de custo. Existem diversos tipos de otimizadores, que funcionam de maneiras específicas e se aplicam a diferentes problemas. Alguns dos principais tipos de otimizadores são o Gradient Descent, Stochastic Gradient Descent, Adagrad, Adadelta, RMSprop e Adam levine.

2.2.5.1 Gradient Descent

A descida do gradiente é um dos otimizadores mais antigos e comuns usados em redes neurais. Ele tenta minimizar o custo atualizando os parâmetros do modelo para que se tornem mais adequados e correspondentes ao mínimo global (valor mínimo da função em todo o espaço de parâmetros).

O gradiente de uma função é apresentado através da Equação (2.18).

$$\nabla L(w_1, w_2, \dots, w_n) = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_2} \\ \vdots \\ \frac{\partial L}{\partial w_n} \end{bmatrix} \quad (2.18)$$

O método da descida do gradiente consiste em atualizar uma variável (no caso de redes neurais os pesos e bias) na direção em que a função decresce mais rapidamente.

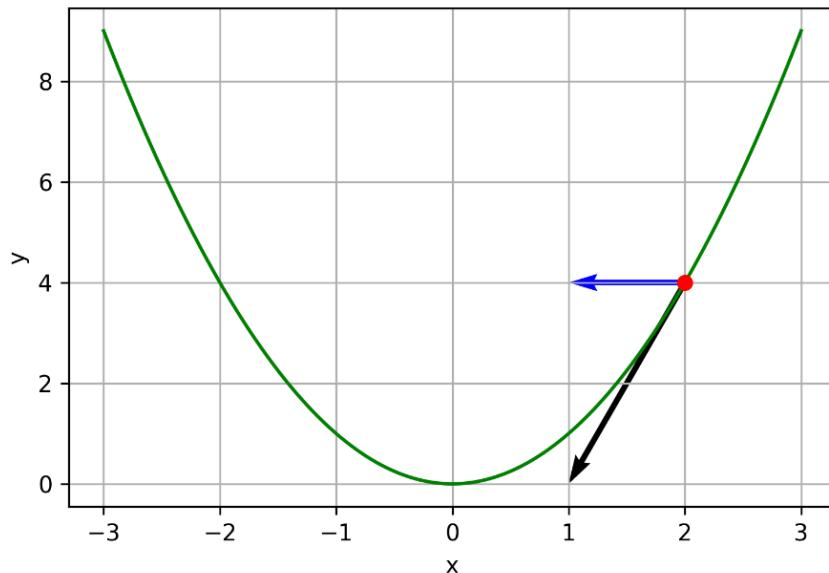
Se o gradiente de uma função aponta para a direção de maior acréscimo dessa função, então a direção de maior decréscimo será a direção contrária, ou seja, o gradiente com sinal negativo conforme a Equação (2.19).

$$w_{t+1} = w_t - \eta \nabla L(w_t) \quad (2.19)$$

Em que η é um número muito pequeno, pois a direção de maior decréscimo de uma função pode variar, logo, se η for um número muito grande, a descida do gradiente não convergirá para o ponto de mínimo da função.

A Figura 2.24 apresenta a representação de uma função de duas variáveis. A seta azul representa a direção na qual o gradiente negativo da função puxa o ponto vermelho e a seta preta a direção que o ponto percorrerá ao ser puxado pelo vetor em azul, pois o ponto pode se mover somente pela função representada pela cor verde.

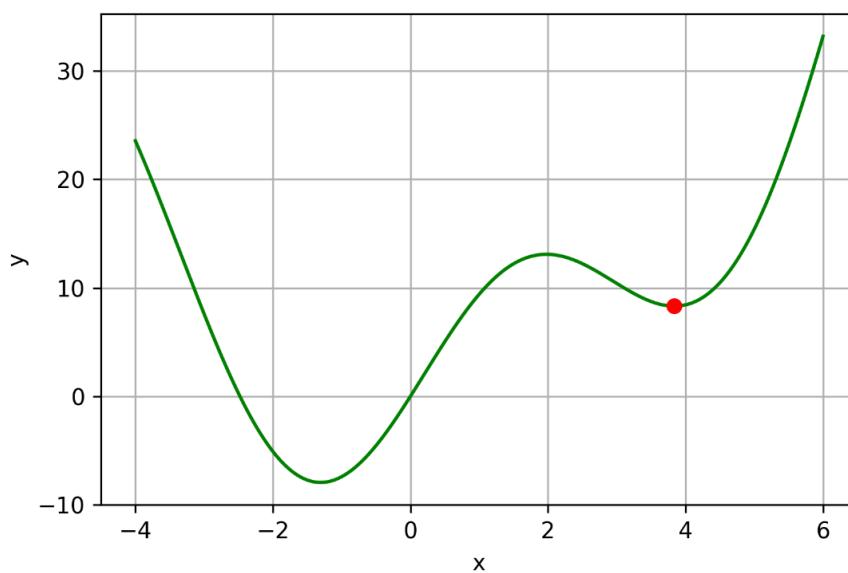
Figura 2.24 – Descida do gradiente



Fonte: Autor.

Um dos desafios da descida do gradiente é que para funções não convexas, a convergência pode ocorrer para um mínimo local ao invés de mínimo global, ou seja, o modelo pode convergir para o valor mínimo da função em um determinado ponto do espaço de parâmetros, podendo ser um ponto de sela, onde a descida do gradiente fica presa como mostrado na Figura 2.25, na qual o ponto vermelho (representando um peso ou um bias) não pode convergir para o mínimo global da função de custo, pois não existe descida em nenhuma das direções.

Figura 2.25 – Mínimo local



Fonte: Autor.

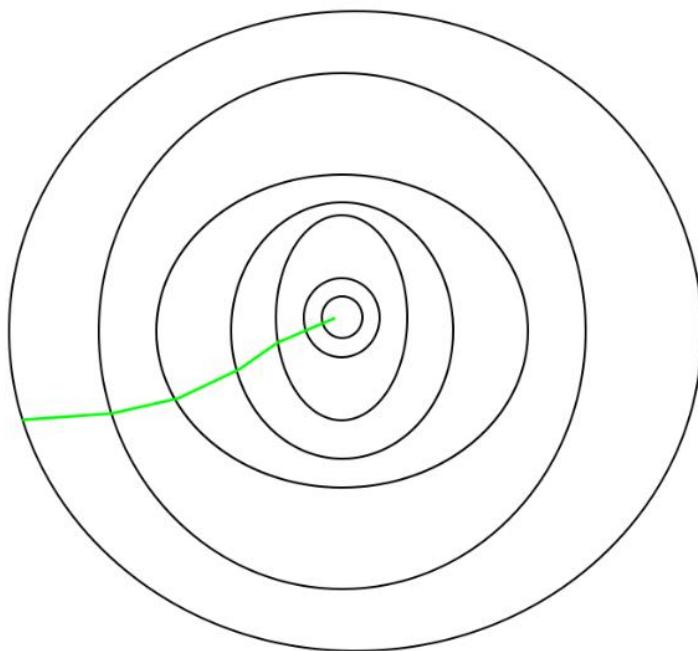
Para certos casos, problemas como gradiente de fuga (problema que ocorre quando o gradiente fica muito pequeno, fazendo com que as atualizações realizadas pelo backpropagation sejam insignificantes) ou explosão do gradiente (problema que ocorre quando o gradiente fica muito grande, fazendo com que o algoritmo seja instável e pare de funcionar) também podem ocorrer devido à inicialização incorreta dos parâmetros.

Uma das principais vantagens da descida do gradiente é que ela pode ser usada para otimizar qualquer função, desde que seja possível calcular o gradiente da função. No entanto, uma das principais desvantagens da descida do gradiente é que ela pode ser lenta para convergir em funções de custo complexas.

2.2.5.2 SGD

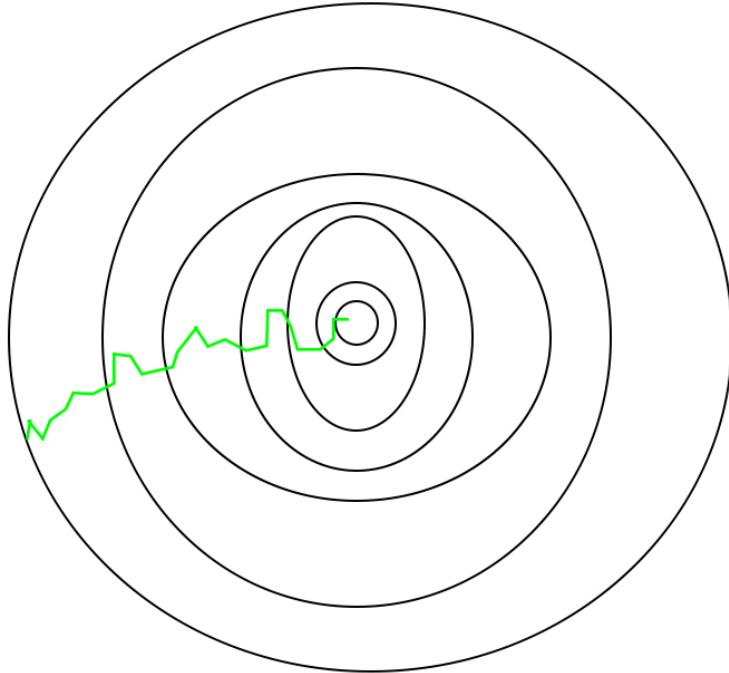
O otimizador Stochastic Gradient Descent (SGD) é uma variação do método da descida do gradiente que busca alcançar maior eficiência computacional, pois diferentemente da descida do gradiente, o SGD não utiliza todo o conjunto de dados para cada iteração no treinamento da rede neural. Ao invés disso, o SGD seleciona aleatoriamente um lote (amostra dos dados de entrada) para realizar a propagação para frente seguida do backpropagation. Consequentemente, devido à utilização de uma amostra ao invés de todo o conjunto de dados, o SGD percorre um caminho ruidoso que se aproxima do caminho mais curto em direção à minimização da função de custo como mostrado na Figura 2.27 (a convergência da descida do gradiente é mostrada na Figura 2.26), mas não exatamente o caminho mais curto, pois para encontrar a direção em que a função de custo decresce mais rapidamente é necessário considerar todo o conjunto de dados.

Figura 2.26 - Caminho do Gradient Descent



Fonte: (LIMA, 2022).

Figura 2.27 - Caminho do SGD



Fonte: (LIMA, 2022).

Apesar de necessitar de um número de iterações maior do que o Gradient Descent, o SGD ainda exige menos custo computacional. Portanto, na maioria dos casos, o SGD é preferível se comparado ao Gradient Descent para otimizar um algoritmo de aprendizado.

2.2.5.3 SGD with momentum

O algoritmo SGD with momentum (SGD com impulso) é uma variação do SGD que sempre funciona melhor, pois no SGD não é possível aumentar a taxa de aprendizado η devido à alta oscilação (caminho ruidoso), fazendo com que a convergência seja lenta. Por outro lado, o SGD with momentum utiliza médias exponencialmente ponderadas para calcular o gradiente conforme a Equação (2.20).

$$w_t = w_{t-1} - \eta V_t \quad (2.20)$$

Em que V_t é calculado através da Equação (2.21).

$$V_t = \beta V_{t-1} + (1 - \beta) \frac{\partial L}{\partial w_{t-1}} \quad (2.21)$$

O SGD with momentum adiciona uma dependência do seu gradiente anterior, fazendo com que a oscilação diminua e a convergência do SGD para a minimização do custo seja mais rápida.

2.2.5.4 Adagrad

O Adagrad (gradiente adaptativo) é um algoritmo de otimização que utiliza o conceito da descida do gradiente, porém com uma taxa de aprendizado η diferente para cada parâmetro baseado na interação. O motivo para usar taxas de aprendizado diferentes é que os parâmetros de recursos esparsos precisam de uma taxa de aprendizado maior, pois eles ocorrem com menos frequência.

Essa taxa de aprendizado é calculada através da Equação (2.22).

$$\eta_t = \frac{\eta}{\sqrt{\alpha_t + \epsilon}} \quad (2.22)$$

Em que ϵ é um número pequeno que serve para que não ocorra uma divisão por 0 caso o α_t seja nulo. Já o α_t é calculado através da Equação (2.23).

$$\alpha_t = \sum_{i=1}^t \left(\frac{\partial L}{\partial w_i} \right)^2 \quad (2.23)$$

Pode-se perceber que quanto maior o número de atualizações de um parâmetro, haverá mais elementos para serem somados, tornando o α_t maior e consequentemente diminuindo a taxa de aprendizado. O contrário também acontece, pois recursos esparsos terão uma taxa de aprendizagem maior.

O backpropagation ocorre de forma muito semelhante ao SGD, porém é utilizada a taxa de aprendizado personalizada para cada um dos pesos e bias da rede neural conforme a Equação (2.24).

$$w_t = w_{t-1} - \eta_t \nabla L \quad (2.24)$$

O Adagrad é um otimizador que utiliza uma taxa de aprendizado diferente para cada parâmetro, o que torna o aprendizado mais eficiente para aplicações com dados esparsos. No entanto, uma grande desvantagem é que a taxa de aprendizado é reduzida ao longo das iterações, o que pode fazer com que o modelo pare de aprender.

2.2.5.5 Adadelta

O algoritmo de otimização Adadelta é uma variação do Adagrad que tem como objetivo corrigir a redução da taxa de aprendizado infinitesimalmente que ocorre no Adagrad. Essa correção é feita através de uma mudança no cálculo do α , que é substituído por médias exponencialmente ponderadas dos gradientes quadrados (S_t), apresentadas através da Equação (2.25).

$$S_t = \gamma S_{t-1} + (1 - \gamma) \left(\frac{\partial L}{\partial w_t} \right)^2 \quad (2.25)$$

Em que γ é um valor tipicamente de 0,9 ou 0,95 que é capaz de restringir a taxa de aprendizado das iterações, evitando que o S_t se torne um valor muito grande e consequentemente evitando que η_t se torne um valor muito pequeno, corrigindo o problema do gradiente de fuga do Adagrad.

A taxa de aprendizado é calculada de forma semelhante ao Adagrad, como apresentado na Equação (2.26).

$$\eta_t = \frac{\eta}{\sqrt{S_t + \epsilon}} \quad (2.26)$$

2.2.5.6 Adam

O otimizador Adam é uma mesclagem entre o SGD with momentum e o Adadelta. Enquanto o SGD with momentum utiliza médias ponderadas exponenciais para calcular o gradiente anterior e o Adadelta utiliza o mesmo conceito para calcular uma taxa de aprendizado variável, o Adam utiliza ambos conceitos, que são apresentados na Equação (2.27) de atualização dos parâmetros.

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{S_t - \epsilon}} V_t \quad (2.27)$$

Em que S_t é descrito pela Equação (2.28).

$$S_t = \gamma S_{t-1} + (1 - \gamma) \left(\frac{\partial L}{\partial w_t} \right)^2 \quad (2.28)$$

Em que V_t é descrito pela Equação (2.29).

$$V_t = \beta V_{t-1} + (1 - \beta) \frac{\partial L}{\partial w_{t-1}} \quad (2.29)$$

O Adam está entre os otimizadores mais utilizados por sua simplicidade de implementação, eficiência computacional, pouca necessidade de memória e adequação para problemas com gradientes ruidosos/esparsos. Além disso, os hiperparâmetros (parâmetros que podem ser ajustados para melhorar o desempenho do algoritmo, como taxa de aprendizagem, tamanho do lote e número de iterações) têm uma interpretação intuitiva e normalmente requerem um pequeno ajuste.

2.3 TRABALHOS RELACIONADOS

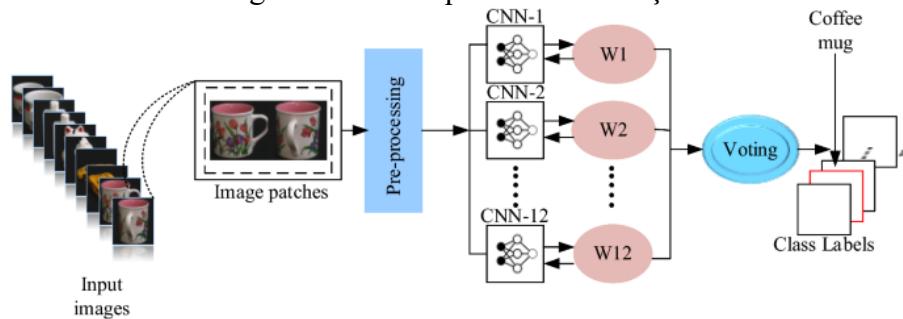
Nos últimos anos foram realizadas diversas pesquisas sobre aplicação de técnicas de inteligência artificial na área da engenharia, como diagnóstico de falhas de transformadores (HAN, 2022), diagnóstico de geradores (ILIC, 2022), previsão de consumo de energia elétrica (BANIK, 2021), detecção automática de COVID-19 usando imagens de raios-X (RESEARCHERS, 2022), dentre outras (WANG, 2019; SU, 2019; MUKHERJEE, 2022, YU, 2021; HUANG, 2020; ZHANG, 2022; PU, 2021; LU, 2022). Essas técnicas foram capazes de automatizar processos que anteriormente necessitavam do trabalho humano, proporcionando a redução de custos e em muitos casos o aumento da precisão na realização do processo.

Muitas pesquisas também foram realizadas na área da automação industrial, como detecção de falhas (WANG, 2022; FERNANDES, 2020; YU, 2020; WANG 2018), em que foram obtidos diagnósticos mais assertivos em relação a falhas em produtos genéricos, reconhecimento de objetos (BALAMURUGAN, 2022), controle de qualidade e segurança

(MEDEIROS, 2007; WANG 2018), dentre outras (KHODAYAR, 2017; MHLANGA, 2020; BÉCUE, 2021; SOOD, 2022).

O trabalho de Balamurugan (2022) aborda uma solução para a identificação de objetos de visão múltipla com o uso de uma Rede Neural Convolucional. O modelo consiste em três etapas, pré-processamento, pré-treinamento das CNNs e mecanismos de votação conforme mostrado na Figura 2.28.

Figura 2.28 – Arquitetura da solução



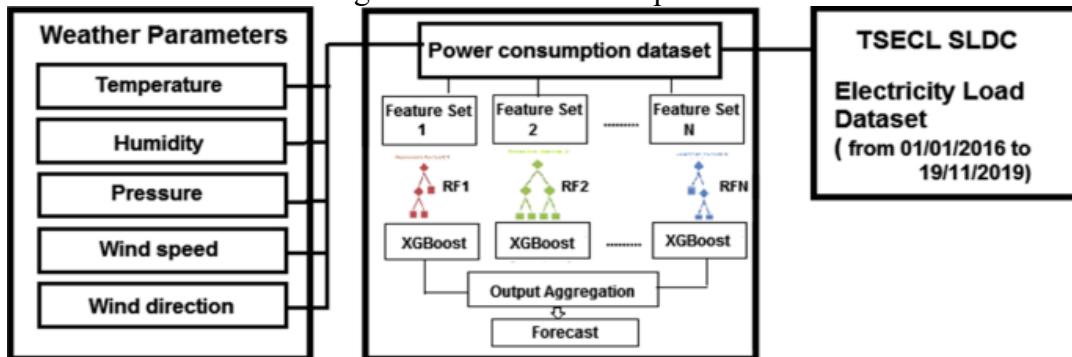
Fonte: (BALAMURUGAN, 2022).

O pré-processamento é realizado para eliminar qualquer ruído indesejado. Os modelos CNN pré-treinados são usados como extratores de características e classificam as imagens em suas categorias correspondentes. No trabalho, nove CNNs pré-treinadas são usadas, como Alex Net, VGGNet, GoogLeNet, Inceptionv3, SqueezeNet, ResNet v2, Xception, MobileNetV2 e ShuffleNet. Por fim, a classe de saída é escolhida com base no sistema de votação. O desempenho foi verificado com duas situações, imagens sem rotação e com rotação. A precisão geral foi de 99% e 93% para reconhecimento sem rotação e com rotação, respectivamente. Portanto, o sistema demonstra a eficácia do reconhecimento de objetos multivisão, o que pode ser usado para aplicações de automação industrial.

Outro trabalho é o realizado por Banik (2021), que propõe um modelo para prever o consumo de eletricidade em Agartala na Índia, alcançando previsões precisas para as próximas 24 horas, bem uma estimativa de 1 semana a 1 mês. Ele analisou características específicas da cidade para identificar variáveis que pudessem influenciar diretamente o padrão de consumo de energia. Demonstrou-se ainda o aumento da precisão da previsão através da combinação de algoritmos de aprendizado de máquina, incluindo Random Forest e XGBoost. O conjunto RF e XGBoost obteve um aumento de desempenho de 15 a 29%.

O modelo de previsão utilizado é apresentado através da Figura 2.29.

Figura 2.29 – Modelo de previsão

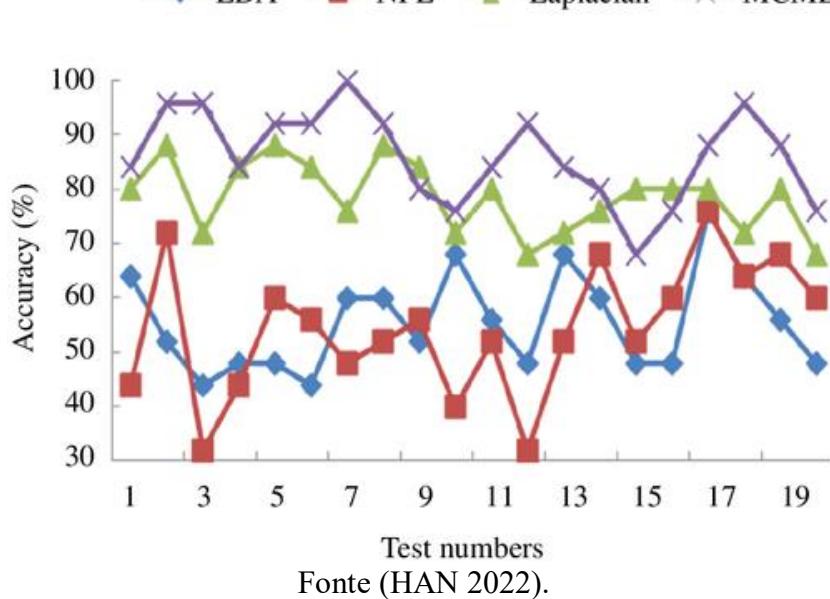


Fonte: (BANIK 2021).

Han (2022), propõe um novo método para diagnosticar falhas no transformador baseado em Maximally Collapsing Metric Learning (MCML) e Kernel Extreme Learning Machine

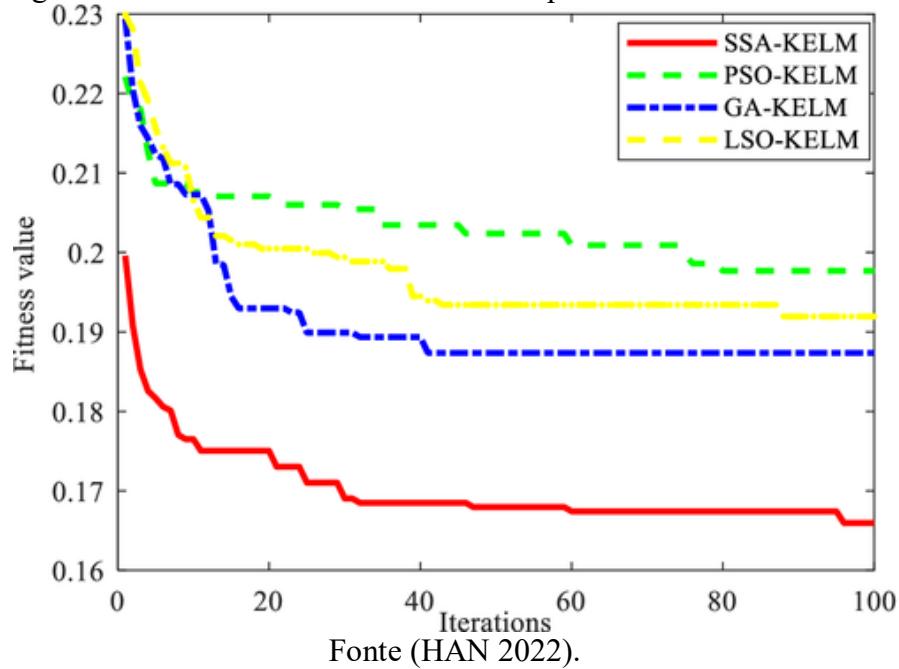
(KELM). Primeiro, uma nova combinação de características de análise de gás dissolvido (DGA), que pode refletir as informações de falha do transformador, é usada para formar o espaço de características de entrada. Em seguida, o MCML é usado para reduzir a dimensão do espaço de características para extrair um conjunto de características DGA ótimas. Por fim, o Salp Swarm Algorithm (SSA) é utilizado para otimizar os parâmetros no KELM para estabelecer um modelo SSA-KELM, que é usado para diagnosticar e identificar falhas no transformador. Os resultados mostram que a eficácia da extração de características do MCML é superior à da análise discriminante linear, neighborhood preserving embedding e Laplacian eigenmaps. Além disso, a precisão de diagnóstico do SSA-KELM estimada a partir do conjunto de testes foi de 86,2%, resultado melhor do que KELM, particle swarm optimization-KELM, genetic algorithm-KELM e loin swarm optimization-KELM. Os indicadores utilizados para essa conclusão foram a precisão média do diagnóstico de falhas das amostras de teste (Figura 2.30), os resultados da redução de dimensionalidade dos quatro métodos e a aptidão média dos testes de 20 vezes usando diferentes algoritmos no processo de otimização para KELM (Figura 2.31). Além disso, a precisão média dos testes de 20 vezes foi usada para comparar o SSA-KELM com os outros sete métodos (Figura 2.32).

Figura 2.30 – Resultados de diagnóstico de falhas de testes de 20 vezes



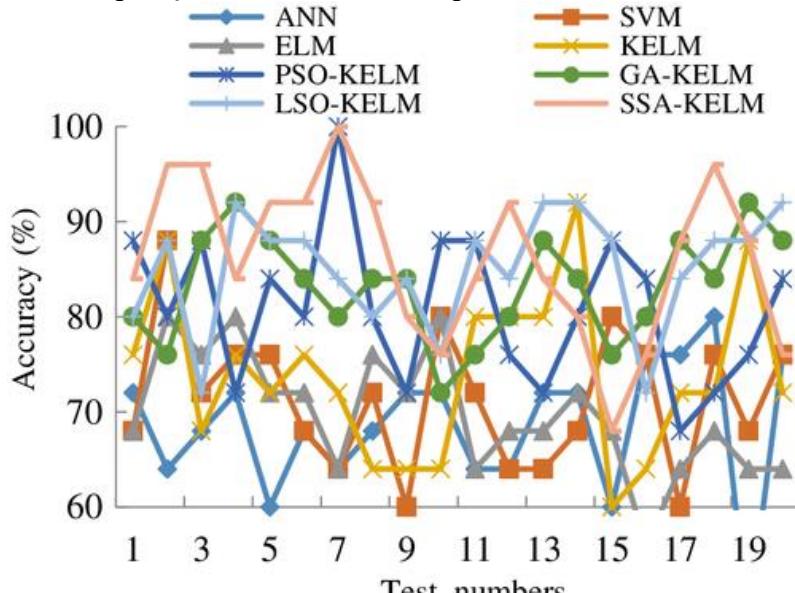
Fonte (HAN 2022).

Figura 2.31 – Curvas médias de valor de aptidão do testes de 20 vezes



Fonte (HAN 2022).

Figura 2.32 – Comparação dos resultados da precisão média dos testes de 20 vezes



Fonte (HAN 2022).

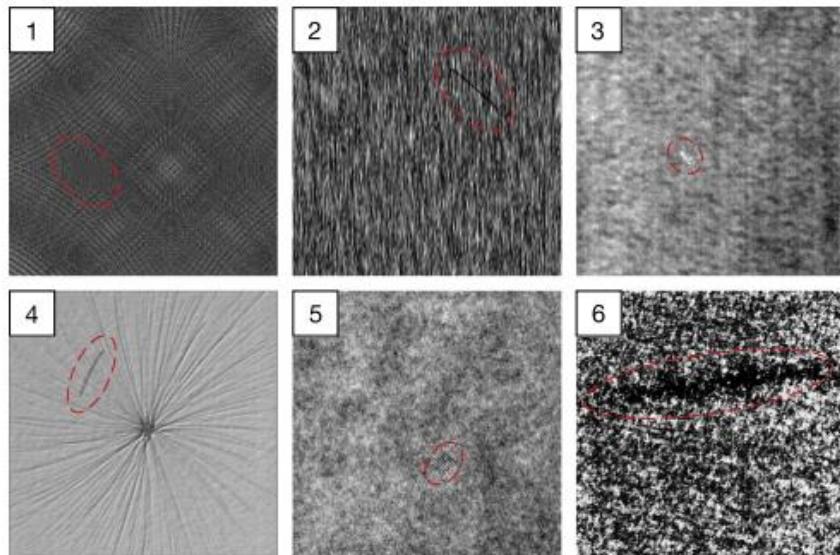
O trabalho de Ilic (2022), envolve a criação de um sistema de inteligência artificial (IA) para classificar geradores elétricos e determinar qualquer serviço necessário. A lógica usada na tomada de decisões será implementada em um sistema especialista fuzzy verificado em 82 geradores de usinas elétricas na Sérvia. A ferramenta criada objetiva contornar a deficiência ou a ausência de critérios nos padrões internacionais. Análises baseadas em relatórios e padrões vigentes são usadas para fundamentar as regras do sistema fuzzy.

Em Khodayar (2017), foi proposta uma arquitetura de Deep Neural Network (DNN) para prever velocidades do vento de curto e ultracurto prazo. Foram utilizados Auto-Encoders (SAE e SDAE) para extrair recursos não supervisionados dos dados de vento e uma camada de regressão supervisionada para prever velocidades. Para melhorar a precisão, Rough Neural Networks foram incorporadas, criando extensões novas de SAE e SDAE que sejam resistentes

às incertezas do vento. Os resultados experimentais mostraram que esses modelos têm melhor desempenho do que os modelos anteriores, com um RMSE e erro absoluto médio reduzidos.

Wang (2018) tem por objetivo criar um modelo de deep learning para a detecção de falhas em produtos genéricos utilizando o dataset DAGM, que consiste em 6 classes de imagens de 512 x 512 pixels, cada uma com 1.000 imagens sem defeitos e 150 com defeitos. A Figura 2.33 apresenta exemplos de imagens que podem ser encontradas no dataset DAGM.

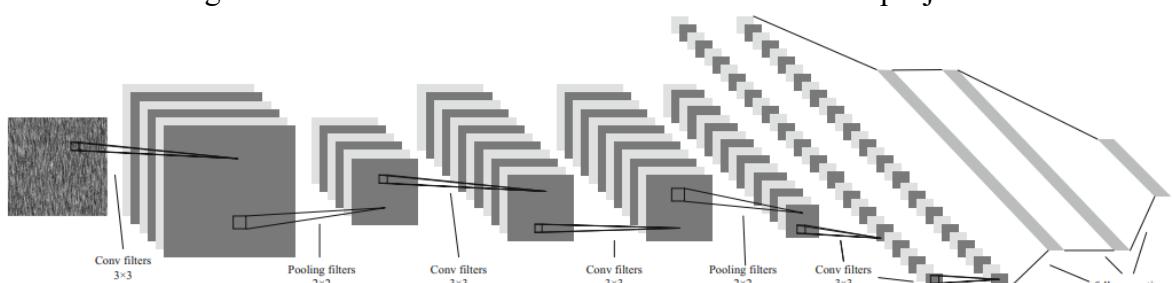
Figura 2.33 – Exemplo de imagens do dataset DAGM



Fonte: (WANG 2018).

O modelo de redes neurais convolucionais projetado (Figura 2.34) é capaz de extrair automaticamente recursos potentes sem muita informação prévia sobre as imagens, além de ser resistente a interferências. O modelo de detecção de defeitos DAGM foi treinado em uma GPU NVIDIA GTX1080 8GB por cerca de 8 horas e foi capaz de detectar 27 imagens por segundo, alcançando uma precisão geral de detecção de 99,8%.

Figura 2.34 – Modelo de redes neurais convolucionais projetado



Fonte: (WANG 2018).

Yu (2020) defende a implementação de uma Rede Neural Convolucional Amplia (BCNN) para identificar falhas em processos industriais. A BCNN pode retirar traços de tendência de defeito e estrutura não-regular a partir de amostras seguidas. Além disso, sua capacidade de aprendizado aditivo possibilita a modernização do modelo para incluir novos exemplos e categorias de falhas. O método proposto foi testado em um processo simulado e em uma aplicação industrial real, oferecendo resultados satisfatórios na detecção das características de processos defeituosos e na atualização do modelo de diagnóstico para acolher novas amostras de falhas.

O objetivo de Huang (2020) é examinar os benefícios da utilização da inteligência artificial no campo da automação elétrica, partindo da inteligência artificial e depois avaliando a aplicação estratégica da tecnologia com base em exemplos de engenharia. Esta investigação tem como objetivo reforçar o controle de automação elétrica, o qual está diretamente ligado à produção social e à qualidade de vida das pessoas.

Fernandes (2020) defende o uso de redes neurais convolucionais como um modelo gerador para sistemas estocásticos, encontrados com frequência na indústria de automação e modelagem, tais como detecção de falhas, visão artificial e análise de dados de sensores. Isso possibilita que pesquisadores de diferentes áreas como medicina, robótica e engenharia de controle desenvolvam uma ferramenta geral para criação de dados e simulações artificiais, dispensando a necessidade de identificar ou supor uma estrutura de sistema particular ou calcular seus parâmetros. Esta abordagem foi aplicada como um modelo gerador para dados obtidos a partir de uma vasta gama de processos deterministas e estocásticos relevantes tecnologicamente, desde osciladores amortecidos até modelos heteroscedásticos condicionais autorregressivos e difusão com salto. Além disso, uma estimativa e previsão não paramétrica foi realizada para o conjunto de dados de série temporal "Fisher River", resultando nos menores erros de previsão quando comparado a um modelo ARIMA padrão. Esta abordagem pode ter potenciais usos como uma alternativa a ferramentas de simulação, como Gibbs sampling e métodos baseados em Monte Carlo, na melhoria da compreensão de redes geradoras adversárias (GANs) e na geração de dados para treinamento de algoritmos de Aprendizado por Reforço.

Wang (2019) propõe um procedimento para prever o desempenho de uma bomba de calor R134a com EVI (Injeção de Vapor Economizado). Os modelos usam SVR (Regressão de Vetor de Suporte) como estimador de base e Adaboost.R2 como método de ensemble para prever a capacidade de aquecimento e COP (coeficiente de desempenho) da bomba de calor. Para a entrada do modelo, são formados diferentes grupos de traços guiados pelo princípio de funcionamento do sistema de bomba de calor e pela análise de correlação. Os parâmetros dos modelos são ajustados para melhorar o desempenho de previsão. A simulação é comparada com os resultados experimentais, e os erros relativos para a previsão de capacidade de aquecimento e COP estão dentro de 8,5%. Além disso, a influência da pressão de injeção no sistema de bomba de calor EVI é discutida e simulada usando o modelo estabelecido. A pressão de injeção ótima do sistema de bomba de calor pode ser obtida do modelo em diferentes condições de operação.

2.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO

A Inteligência Artificial é um tema extremamente amplo e abrangente, com várias ramificações que possibilitam a aplicação desta tecnologia nos mais diversos setores. Diferentes métodos e abordagens podem ser empregadas para o desenvolvimento de soluções inteligentes. A Inteligência Artificial é, portanto, uma ferramenta extremamente poderosa e versátil que continua a evoluir e a oferecer soluções cada vez mais avançadas para os problemas do mundo moderno. Ainda fica evidente a complexidade de definir os parâmetros de uma rede neural, desde sua estrutura, funções de ativação, funções erro e outros. Desta forma, essas definições devem ser feitas muito em função da aplicação e resultados esperados.

Neste capítulo, foi apresentado um amplo panorama das diversas abordagens de Inteligência Artificial. Foi mostrado que os métodos de IA são muito diversos, compartilhando os mesmos princípios de funcionamento descritos na fundamentação teórica e que todos eles têm potencial para serem aplicados a muitos problemas e domínios diferentes, oferecendo soluções eficientes. Embora a Inteligência Artificial ainda tenha muito a oferecer, é importante lembrar que o progresso da área depende do desenvolvimento de modelos mais robustos, e por isso é importante que os pesquisadores continuem a estudar novos métodos e a desenvolver

novas abordagens para a resolução de problemas. Ao fazer isso, o potencial da Inteligência Artificial será maximizado para resolver problemas em diversas áreas.

Tendo em vista a potencialidade da Inteligência Artificial e a vasta variedade de configurações e métodos, o capítulo 3 deste trabalho apresenta uma proposta de utilização prática desta tecnologia, mais especificamente do Deep Learning, na área da Automação Industrial.

3 METODOLOGIA

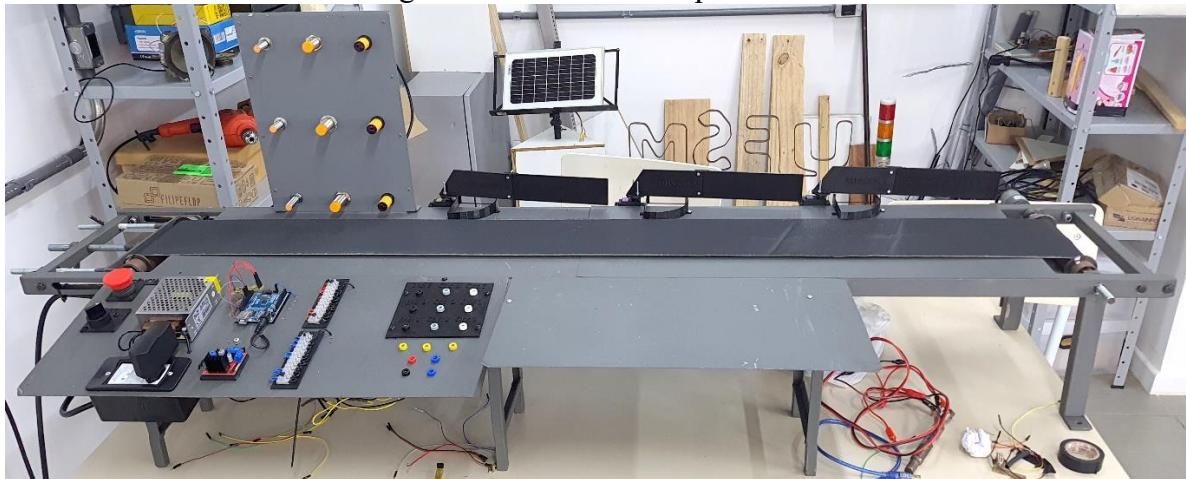
A proposta do presente trabalho consiste em um grande problema para escala de processos industriais: a identificação de produtos danificados. Esta identificação automatizada de objetos danificados elimina a incerteza/subjetividade da avaliação realizada por um colaborador, trazendo assim mais eficiência e precisão ao processo produtivo. Além disso, a automação também proporcionará a redução do tempo gasto para classificar os produtos.

Este trabalho tem como objetivo desenvolver uma metodologia capaz de classificar e identificar falhas em objetos de uma linha produtiva industrial, bem como avaliar o desempenho desta rede perante variações paramétricas/estruturais. Para alcançar o objetivo deste trabalho, será adotada a metodologia a seguir.

É proposta uma rede neural aplicada em um estudo de caso em um sistema de automação constituído de uma esteira transportadora de objetos e, a partir de imagens capturadas por uma câmera, identificar os objetos danificados e classificar o seu tipo. O algoritmo de reconhecimento de imagem construído com redes neurais artificiais é capaz de identificar padrões de defeitos e descartar objetos danificados. Com isso, pode-se melhorar a qualidade do produto final e aumentar a eficiência de produção.

A esteira transportadora utilizada no sistema apresentada através da Figura 3.1 foi desenvolvida por Dellazari (2022) em seu Trabalho de Conclusão de Curso. Já para cumprir a função de câmera utiliza-se um smartphone, evitando custos de desenvolvimento adicionais.

Figura 3.1 – Esteira transportadora



Fonte: Autor.

Para a implementação do sistema, necessita-se da criação de um banco de imagens para treinamento da rede neural, a rotulação das imagens para criar um dataset de treinamento, a implementação da rede neural para classificar objetos danificados e por tipo, a avaliação do desempenho da rede neural para diferentes parâmetros e configurações e a busca pela otimização do algoritmo.

A metodologia de ação para alcançar os objetivos propostos é apresentada através do fluxograma da Figura 3.2.

Figura 3.2 – Fluxograma da metodologia



Fonte: Autor.

Todas as etapas apresentadas pelo fluxograma da Figura 3.2 são consideradas fundamentais para que os objetivos deste trabalho sejam alcançados.

Primeiramente, foi construído um banco de imagens equilibrado, contendo aproximadamente a mesma quantidade de imagens de objetos defeituosos e apropriados. Caso contrário, se a quantidade de dados de objetos apropriados for muito superior à quantidade de objetos danificados, a rede neural pode não aprender adequadamente e responder que todos os objetos são apropriados. Os objetos utilizados para a realização dessa etapa são formas geométricas como quadrados, triângulos e círculos manualmente confeccionadas com Etileno Acetato de Vinila, sendo algumas dessas formas geométricas propositalmente danificadas. A altura desses objetos é mínima, correspondendo à espessura dos próprios materiais utilizados. A quantidade total de objetos é de 330 unidades, na qual 300 unidades são divididas em quantidades iguais para objetos danificados, ou seja, 100 unidades para cada forma geométrica

com defeito e 30 unidades totais de objetos apropriados, sendo 10 cópias para cada forma geométrica. A quantidade de cópias de objetos apropriados é menor, pois como não possuem defeitos, serão extremamente semelhantes. As imagens são registradas em ambientes diferentes para garantir que a rede neural não seja influenciada pela iluminação na tomada de decisão. O ângulo da câmera também é levemente variado juntamente a posição dos objetos na esteira.

O segundo passo está diretamente relacionado com o primeiro, pois consiste em rotular as imagens criadas anteriormente, adicionando ao dataset a saída esperada da rede neural para cada dado de entrada (imagens de objetos).

A Figura 3.3 apresenta um exemplo de diferenciação de objetos apropriados e objetos inapropriados (danificados). Esse mesmo critério foi utilizado para criar o banco de imagens rotuladas. Já o tipo dos objetos são as categorias de formas geométricas: quadrado, triângulo e círculo.

Figura 3.3 – Exemplo de diferenciação de objetos apropriados e inapropriados



Fonte: Autor.

Após a criação do dataset, se inicia o processo de criação da rede neural que deve ser capaz de classificar objetos apropriados e inapropriados além de identificar as formas geométricas. Nesta etapa, são conduzidos experimentos para testar diferentes configurações de redes neurais, incluindo perceptrons e redes convolucionais, com variações na quantidade de parâmetros. Esses experimentos visam analisar o desempenho e a eficácia das diferentes arquiteturas de rede neural utilizadas, considerando suas respectivas configurações e variações. As redes neurais terão 6 saídas correspondentes a diferentes classes de objetos, indicando o tipo do objeto e sua condição de defeito. As redes neurais serão criadas através da linguagem de programação Python associada à biblioteca de código aberto TensorFlow, desenvolvida pelo Google para facilitar a criação de IAs especialmente redes neurais.

Além disso, são avaliados os indicadores de qualidade provenientes da matriz de confusão, como acurácia, precisão, sensibilidade e F1-score. Uma matriz de confusão é uma tabela que descreve o desempenho de um modelo de classificação em um problema de aprendizado supervisionado, permitindo avaliar sua precisão ao prever as classes reais. Ela é construída com base nas previsões do modelo em relação aos rótulos verdadeiros dos dados de teste, organizada em uma tabela em que as linhas representam as classes reais e as colunas representam as classes previstas. Essa matriz fornece uma visão geral do desempenho do modelo, ajudando a identificar o número de previsões corretas (verdadeiros positivos e verdadeiros negativos) e incorretas (falsos positivos e falsos negativos) para cada classe, permitindo a análise do seu desempenho em diferentes categorias.

Por fim, o sistema de reconhecimento de imagem é implantado na esteira transportadora de objetos criada por Dellazari (2022), na qual um ESP32 controlará a angulação de servo motores acoplados a atuadores de forma que consiga empurrar os objetos separando-os por tipo e descartando os defeituosos. Um servidor web desenvolvido se encarregará de realizar a

categorização através redes neurais e informar o ESP32 via internet do resultado para que tenha a informação necessária para configurar o ângulo dos servo motores.

3.1 CONSIDERAÇÕES FINAIS DO CAPÍTULO

No presente capítulo, foi apresentada a metodologia para o desenvolvimento de um sistema de automação para identificação de tipos e defeitos de objetos. A metodologia proposta envolve a criação de um banco de imagens balanceadas e rotuladas para treinar a rede neural, a implementação de uma rede neural para classificar os objetos por tipo e identificar os danificados e a busca pela otimização do algoritmo. O sistema é implantado em uma esteira transportadora para que os objetos sejam entregues aos pontos de coleta. O ESP32 controla a angulação dos servo motores a fim de empurrar os objetos para separá-los conforme a sua categoria e descartar os defeituosos. Além disso, um servidor web se responsabiliza por realizar a categorização e informar o ESP32 do resultado.

4 RESULTADOS

Este capítulo aborda a implementação da metodologia em várias etapas distintas. Inicia-se com a criação do dataset de figuras adequadas, envolvendo a coleta, seleção e organização das imagens relevantes para o estudo (5.1). Em seguida, é detalhado o processo de pré-processamento das imagens (5.2). Diferentes modelos de redes neurais são explorados na seção 5.3. Os resultados obtidos por meio da aplicação desses modelos são analisados na seção 5.4, utilizando métricas de avaliação e destacando observações relevantes. Posteriormente, é apresentado um novo pré-processamento (5.5), seguido pelos resultados obtidos após sua aplicação (5.6). Na seção 5.7, aborda-se a danificação manual de figuras, enquanto na seção 5.8, além da introdução de danificação automática, são definidos critérios de avaliação de desempenho para os modelos. Por fim, descreve-se o desenvolvimento de um sistema de classificação na esteira transportadora, utilizando o melhor modelo treinado para classificar as figuras em tempo real (5.9).

Adicionalmente, foram conduzidos experimentos com conjuntos de dados de dimensões e complexidades superiores. No entanto, considerando que essas experiências estão além do escopo abordado neste contexto específico, optou-se por apresentar os resultados e as análises correspondentes no Apêndice, mais precisamente no capítulo 8.1.

Os códigos utilizados para o desenvolvimento deste trabalho estão disponíveis no repositório GitHub localizado no seguinte endereço: <https://github.com/luanws/TCC>.

4.1 CRIAÇÃO DO DATASET DE FIGURAS ADEQUADAS

Inicialmente, foram confeccionadas 125 figuras de cada tipo, ou seja, 125 quadrados, 125 círculos e 125 triângulos, totalizando 375 figuras criadas a partir de recortes de Etileno Acetato de Vinila. Todas essas figuras são inicialmente consideradas adequadas, pois as figuras danificadas serão posteriormente adaptadas a partir das já existentes após a criação do dataset de figuras adequadas. Essas figuras foram utilizadas para a criação de um conjunto de dados com a finalidade de realizar o treinamento e o teste de acurácia de diferentes modelos de Deep Learning projetados para reconhecimento de imagens.

Os materiais utilizados para a captura das imagens das figuras geométricas foram uma esteira transportadora, um smartphone e um suporte para manter o dispositivo móvel em uma posição fixa durante a captura das imagens. As imagens foram obtidas a partir da captura de fotografias sequenciais dos objetos que passavam pela esteira, conforme mostrado na Figura 4.1.

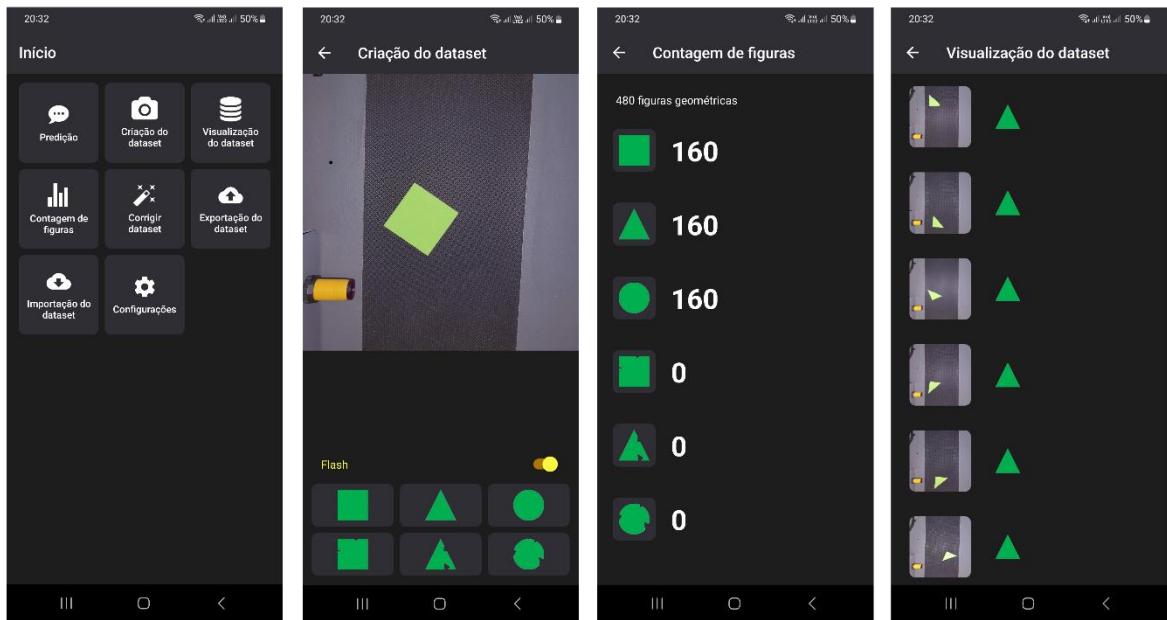
Figura 4.1 – Montagem do setup



Fonte: Autor.

Para auxiliar na criação do dataset, foi desenvolvido um aplicativo para fotografar as figuras e rotular as imagens simultaneamente, conforme mostrado na Figura 4.2. O aplicativo tem a função de auxiliar na captura, rotulação, visualização e contagem das imagens.

Figura 4.2 – Telas do aplicativo auxiliar



Fonte: Autor.

O processo de criação do dataset foi conduzido com o objetivo de maximizar a qualidade e consistência dos dados obtidos, garantindo que as imagens fossem capturadas com variações nas posições das figuras, nas posições da câmera e na iluminação para que os modelos de IA fossem capazes de identificar as figuras evitando o overfitting.

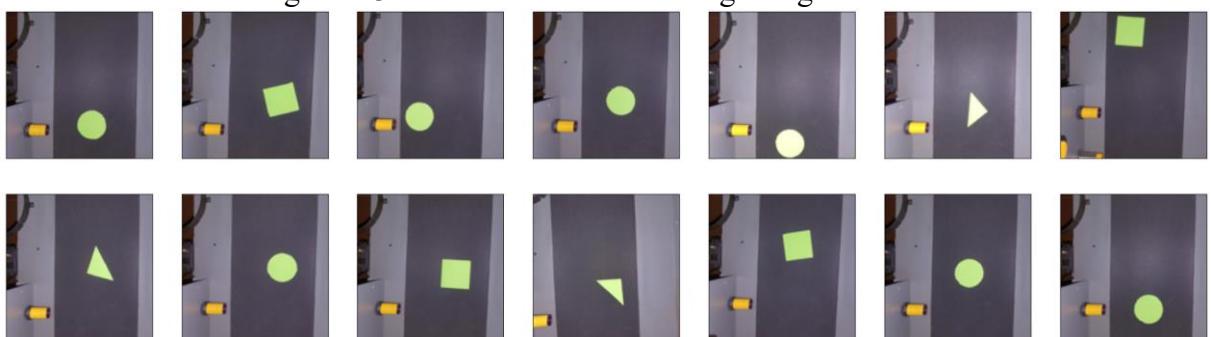
Overfitting é um termo usado em inteligência artificial e estatística para descrever quando um modelo se ajusta muito bem ao conjunto de dados anteriormente observado, mas se mostra ineficaz para prever novos resultados. Ou seja, o modelo não pode ser utilizado para nenhuma outra base de dados, pois apenas decorou as respostas dos dados de treino.

Ao todo, foram criadas 3 versões de datasets nomeados com sua data de criação no padrão ISO 8601:

- 2023-03-22: 300 fotos com boa iluminação artificial;
- 2023-04-02: reaproveitamento das fotos anteriores com acréscimo de 148 fotos com boa iluminação natural tiradas em diferentes horários do dia, totalizando 448 fotos;
- 2023-04-03: reaproveitamento das fotos anteriores com acréscimo de 212 fotos com iluminação ruim, totalizando 660 fotos.

A Figura 4.3, apresenta uma amostra das fotos tiradas das figuras geométricas.

Figura 4.3 – Amostra de fotos das figuras geométricas



Fonte: Autor.

Para efetuar o treinamento da rede neural, essas imagens devem ser representadas computacionalmente na forma numérica. Essa representação é feita na forma matricial e segue o formato $512 \times 512 \times 3$, em que as duas primeiras dimensões correspondem a largura e a altura da imagem e a terceira dimensão corresponde aos canais de cor.

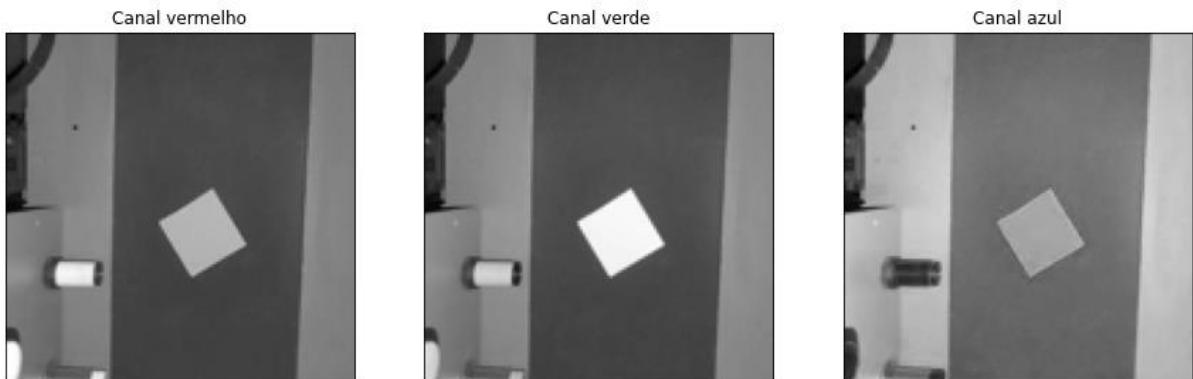
4.2 PRÉ-PROCESSAMENTO

O pré-processamento é uma etapa importante do treinamento de redes neurais, pois através dele pode-se preparar os dados de entrada removendo informações desnecessárias das imagens. Para este trabalho, uma simplificação interessante é remover os canais de cores menos significativos, uma vez que a cor das figuras se destaca em relação ao cenário.

Uma imagem é composta por três canais de cor, sendo eles o canal vermelho, verde e azul. Todas as outras cores de uma imagem são formadas através da combinação dessas três cores em diferentes intensidades.

A Figura 4.4 apresenta os diferentes canais de cor de uma mesma imagem, com o objetivo de identificar o canal em que a figura mais se destaca em relação ao cenário.

Figura 4.4 – Figuras geométricas separadas por canais de cor



Fonte: Autor.

Percebe-se através da Figura 4.4 que o canal verde é o que apresenta maior contraste entre a figura geométrica e o cenário. Por esse motivo, todos os dados de treino consideram somente o canal verde. Dessa forma, foi possível remover 66,66% da informação, mantendo somente o canal de cor mais relevante e reduzindo a dimensionalidade dos dados de entrada de $512 \times 512 \times 3$ para $512 \times 512 \times 1$, em que as duas primeiras dimensões representam a largura e a altura da imagem e a terceira dimensão representa os canais de cor.

4.3 MODELOS DE INTELIGÊNCIA ARTIFICIAL

Tendo um dataset pré-processado e pronto para o uso, foram definidos 4 tipos de modelos de redes neurais, sendo duas redes do tipo Perceptron e duas CNNs. As CNNs foram escolhidas devido ao seu grande potencial para reconhecimento de imagens e capacidades de encontrar padrões na relação de vizinhança entre os pixels de uma imagem. Já as redes do tipo Perceptron foram escolhidas devido à sua característica generalista, servindo também como um comparador para avaliar o desempenho das CNNs.

A Tabela 4.1, Tabela 4.2, Tabela 4.3 e Tabela 4.4 representam as camadas dos modelos de redes neurais criados com a seguinte legenda:

- InputLayer: camada de entrada;

- Flatten: camada que converte os dados multidimensionais em uma única dimensão (estica os dados);
- Dense: camada de neurônios;
- Conv2D: camada convolucional de duas dimensões;
- MaxPooling2D: camada de pooling de duas dimensões.

Tabela 4.1 – P1

Camada	Formato da saída	Ativação
InputLayer	512 x 512 x 1	-
Flatten	262144	-
Dense	128	relu
Dense	64	relu
Dense	32	relu
Dense	3	softmax

Fonte: Autor

Tabela 4.2 – P2

Camada	Formato da saída	Ativação
InputLayer	512 x 512 x 1	-
Flatten	262144	-
Dense	64	relu
Dense	32	relu
Dense	3	softmax

Fonte: Autor

Tabela 4.3 – CNN1

Camada	Formato da saída	Ativação
InputLayer	512 x 512 x 1	-
Conv2D	510 x 510 x 32	relu
MaxPooling2D	127 x 127 x 32	-
Conv2D	125 x 125 x 32	relu
MaxPooling2D	31 x 31 x 32	-
Flatten	30752	-
Dense	8	relu
Dense	3	softmax

Fonte: Autor

Tabela 4.4 – CNN2

Camada	Formato da saída	Ativação
InputLayer	512 x 512 x 1	-
Conv2D	510 x 510 x 32	relu
MaxPooling2D	127 x 127 x 32	-
Conv2D	125 x 125 x 64	relu
MaxPooling2D	31 x 31 x 64	-
Flatten	61504	-
Dense	8	relu
Dense	3	softmax

Fonte: Autor

Todos os modelos foram treinados utilizando a função de erro Entropia Cruzada Categórica Esparsa, projetada para problemas de classificação e o otimizador Adam, um algoritmo de otimização popular usado em modelos de aprendizado de máquina eficiente para a maioria dos problemas.

As funções de ativação das camadas escondidas são todas ReLU, pois no contexto de reconhecimento de imagens, não há razão para escolher outro tipo de função de ativação. Já as imagens são representadas por matrizes com valores entre 0 e 1, ou seja, somente valores positivos, a função ReLU se torna ideal para esse caso, pois as demais funções para camadas escondidas consideram valores negativos, mas são iguais ou extremamente semelhantes para valores positivos. Logo, variar a função de ativação tornaria o processo de treinamento mais lento e não surtiria efeito positivo na acurácia do modelo.

A função de ativação escolhida para a camada de saída foi a softmax, pois essa função pode ser aplicada para problemas de classificação com mais de duas classes.

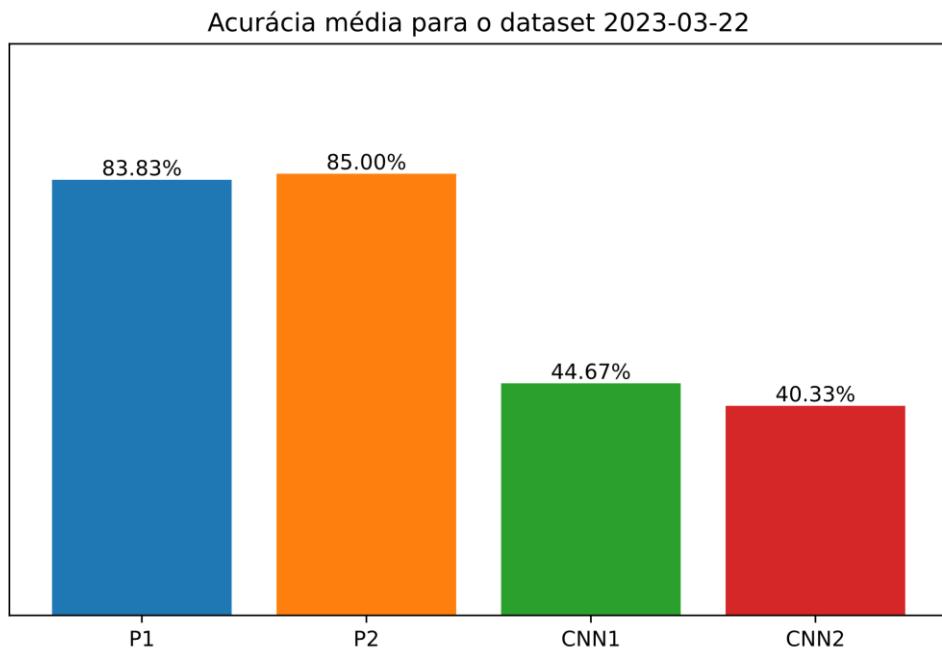
4.4 RESULTADOS

Devido as propriedades de treinamento de redes neurais, os pesos são iniciados de forma aleatória. Os dados de treino e teste também são sorteados. Essa característica de aleatoriedade faz com que a acurácia varie conforme o número de treinamentos da rede neural. Por esse motivo, foram realizadas 10 repetições de treinamento para cada modelo para que fossem identificados os valores médios e máximos dessas acuráncias, possibilitando uma comparação mais precisa dos modelos.

Os testes foram realizados separadamente para cada versão do dataset, ou seja, cada gráfico comparativo contém as acuráncias de cada modelo e as diferentes versões do dataset são apresentadas em gráficos diferentes.

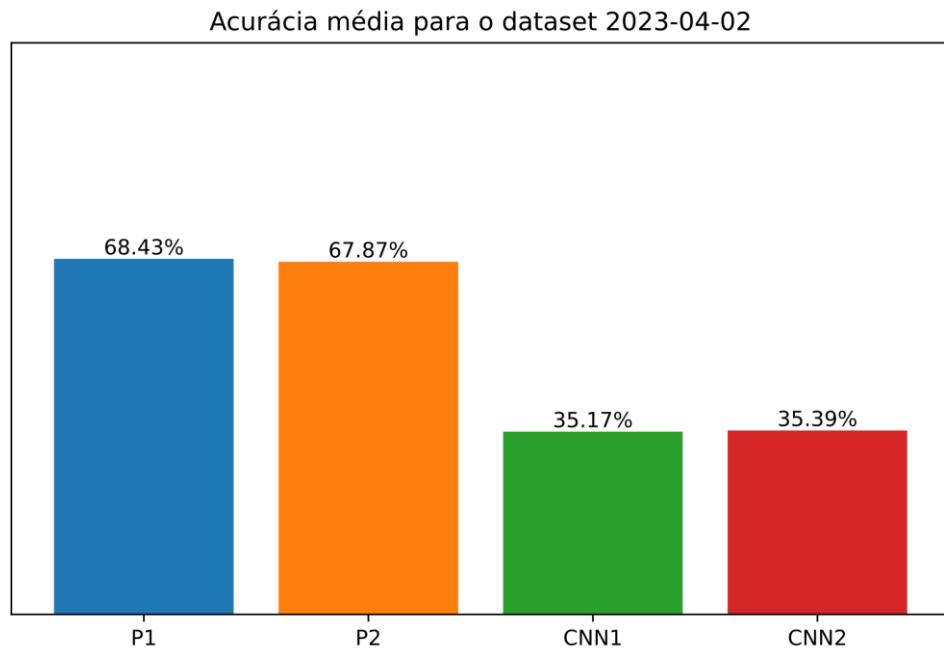
A Figura 4.5, Figura 4.6 e Figura 4.7 apresentam as acuráncias médias para cada versão do dataset.

Figura 4.5 – Acurácia média para o dataset 2023-03-22



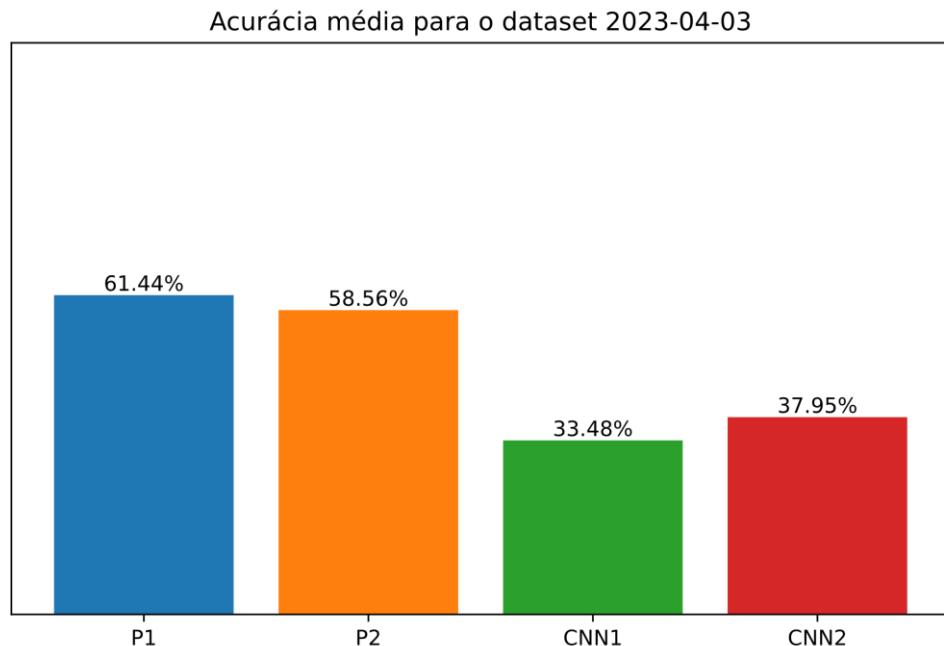
Fonte: Autor.

Figura 4.6 – Acurácia média para o dataset 2023-04-02



Fonte: Autor.

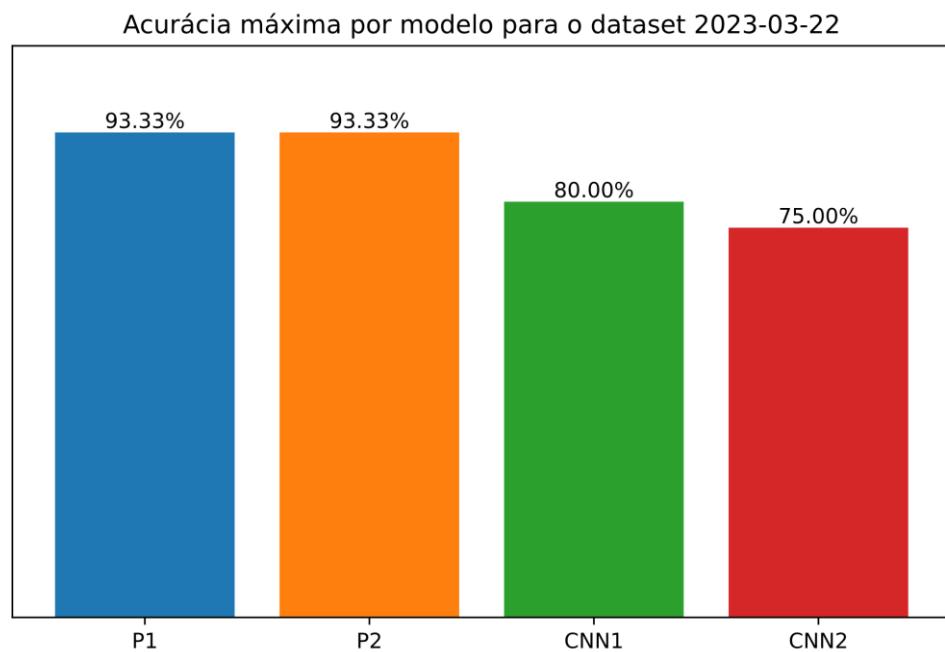
Figura 4.7 – Acurácia média para o dataset 2023-04-03



Fonte: Autor.

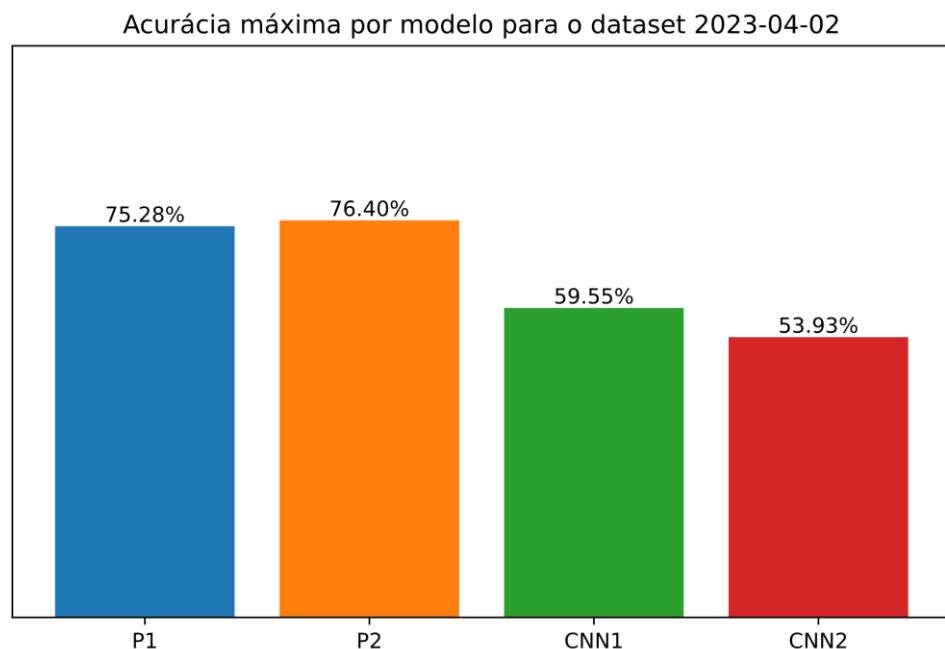
As acuráncias máximas por modelo e versão do dataset são mostradas através da Figura 4.8, Figura 4.9 e Figura 4.10.

Figura 4.8 – Acurácia máxima por modelo para o dataset 2023-03-22



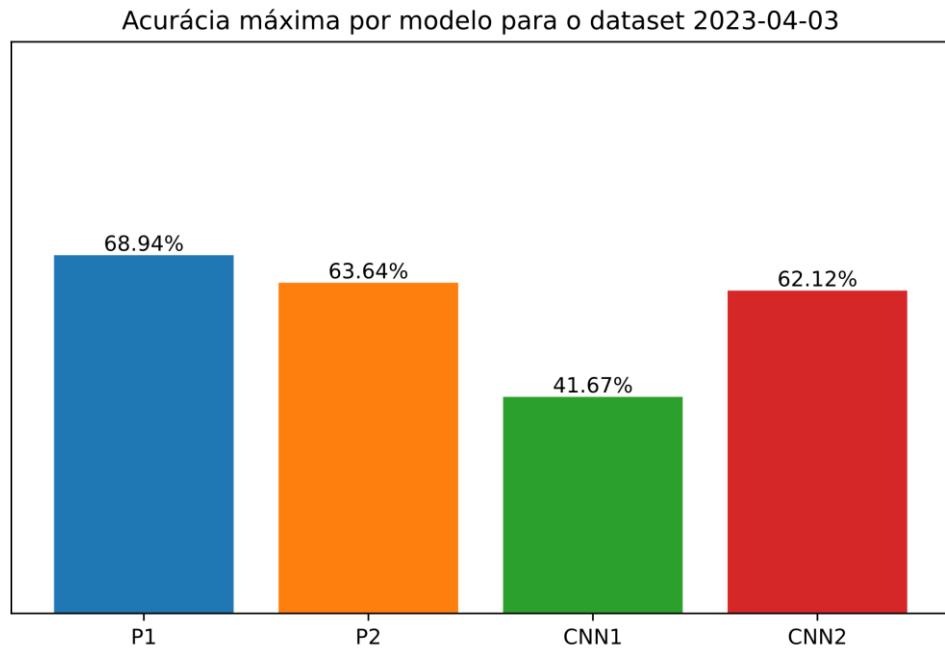
Fonte: Autor.

Figura 4.9 – Acurácia máxima por modelo para o dataset 2023-04-02



Fonte: Autor.

Figura 4.10 – Acurácia máxima por modelo para o dataset 2023-04-03



Fonte: Autor.

Percebe-se através dos gráficos mostrados que utilizando imagens com boa iluminação artificial e imagens bastante semelhantes (2023-03-22), foram obtidas acurácias satisfatórias, porém ao adicionar imagens com iluminação natural (2023-04-02), mas com variações de iluminação, as acurácias tanto médias quanto máximas foram consideravelmente reduzidas. Já ao adicionar figuras com iluminação ruim (2023-04-03), a acurácia se torna ainda menor, sendo inviável para o objetivo deste trabalho.

4.5 NOVO PRÉ-PROCESSAMENTO

Devido à baixa eficiência dos modelos para imagens com baixa iluminação, o pré-processamento foi aprimorado para remover ainda mais informações desnecessárias das imagens. Esse novo pré-processamento consiste nas etapas apresentadas através do fluxograma na Figura 4.11.

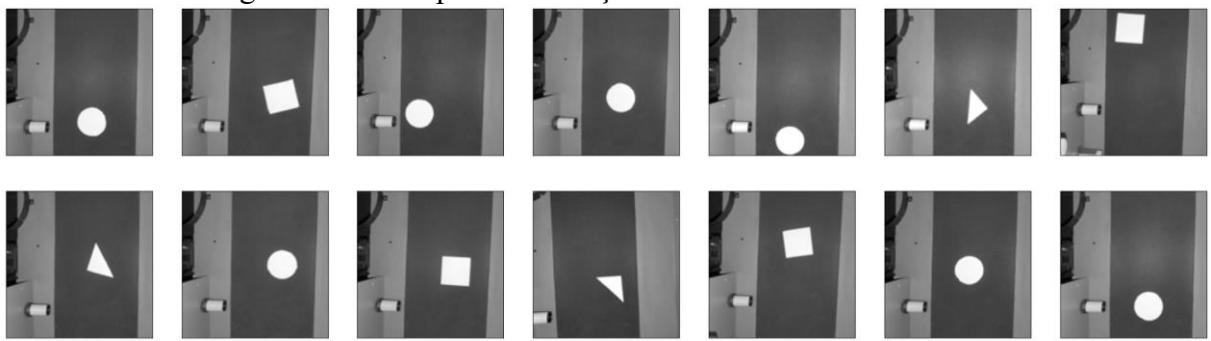
Figura 4.11 – Fluxograma do novo pré-processamento



Fonte: Autor.

A primeira etapa foi mantida conforme o pré-processamento anterior. Removendo os canais vermelho e azul para que a imagem seja representada apenas pelo canal de cor verde. A Figura 4.12 mostra como as mesmas figuras apresentadas na Figura 4.3 se parecem após a primeira etapa do novo pré-processamento.

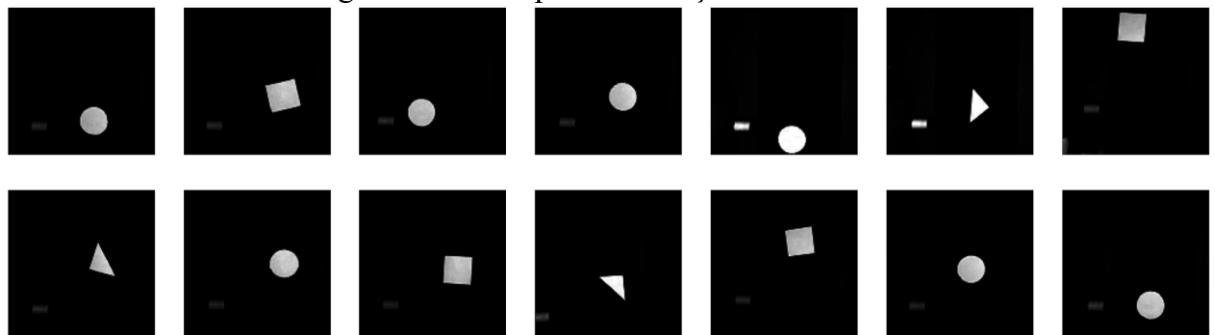
Figura 4.12 – Etapa 1 – Remoção dos canais vermelho e azul



Fonte: Autor.

A Figura 4.13, mostra as figuras após passar pela etapa de elevação de contraste, a qual foi realizada elevando cada valor da matriz que representa a imagem na potência 10. Em seguida, esses valores foram normalizados para que o pixel mais escuro se tornasse 0 e o pixel mais claro se tornasse 1. Os demais pixels foram ajustados proporcionalmente de forma linear.

Figura 4.13 – Etapa 2 – Elevação do contraste



Fonte: Autor.

Em seguida, todos os valores da matriz da imagem foram passados como parâmetro de uma função degrau deslocada, conforme mostrado na Figura 4.14.

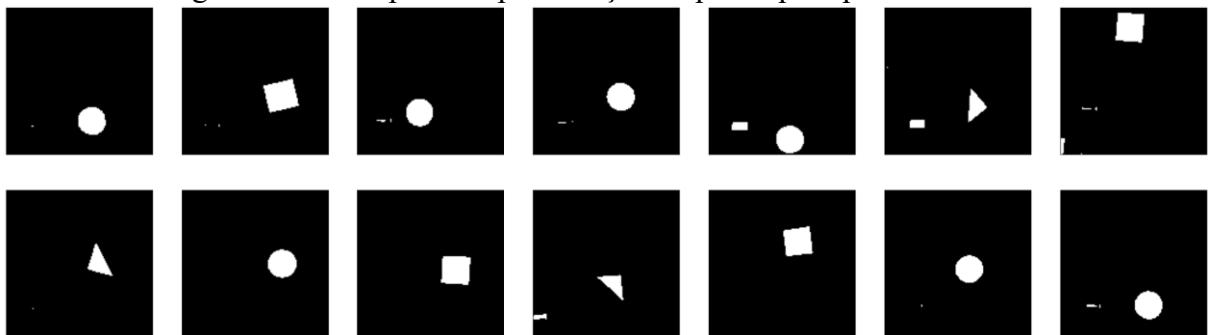
Figura 4.14 – Função degrau deslocada



Fonte: Autor.

Através da função degrau, foi possível transformar todos os pixels da imagem em preto ou branco, ou seja, tornaram-se 0 se anteriormente fossem menores do que 0,15 e 1 se anteriormente fossem maiores ou iguais a 0,15. O resultado é apresentado através da Figura 4.15.

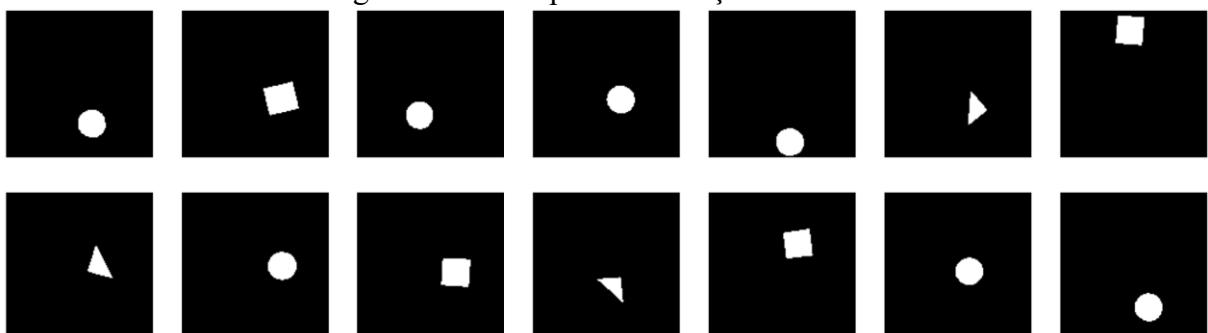
Figura 4.15 – Etapa 3 – Aproximação de pixels para preto ou branco



Fonte: Autor.

Após a aproximação dos pixels para preto ou branco, ainda restaram ruídos na imagem. Uma vez que os grupos de pixels brancos correspondentes às figuras geométricas são sempre maiores do que os grupos de ruídos, como etapa final do pré-processamento, foram identificados e removidos todos os pequenos grupos de pixels brancos. O resultado é apresentado através da Figura 4.16.

Figura 4.16 – Etapa 4 – Remoção de ruídos



Fonte: Autor.

O novo pré-processamento adotado não só simplifica as imagens das figuras, mas também possibilita uma independência entre a figura em si e o plano de fundo. Isso ocorre devido ao fato de que, independentemente da iluminação (exceto em situações extremas ou fundo com cores verdes), a imagem resultante após o pré-processamento será a mesma. Essa abordagem garante uma uniformidade nas características das figuras, permitindo que o modelo se concentre exclusivamente nos padrões de defeitos presentes nas imagens, sem ser influenciado pelas variações de iluminação ou pelo contexto do plano de fundo. Essa independência contribui para uma melhor generalização e robustez do modelo em relação à detecção e classificação dos defeitos.

Além dessas etapas de pré-processamento descritas, também foram aplicadas variações às imagens originais para ampliar o dataset de treinamento. Essas variações feitas de forma virtual são aplicadas aleatoriamente a cada imagem do conjunto de dados para criar novas imagens que podem ajudar a melhorar o desempenho do modelo e evitar o overfitting.

As variações aplicadas às imagens originais são as seguintes:

- Rotação: a imagem é rotacionada em um ângulo aleatório entre 0 e 360 graus;
- Deslocamento horizontal: a imagem é deslocada horizontalmente em uma quantidade aleatória que pode variar até 10% do tamanho da imagem;
- Deslocamento vertical: a imagem é deslocada verticalmente em uma quantidade aleatória que pode variar até 10% do tamanho da imagem;
- Flip horizontal: a imagem é espelhada horizontalmente com 50% de probabilidade;

- Flip vertical: a imagem é espelhada verticalmente com 50% de probabilidade.

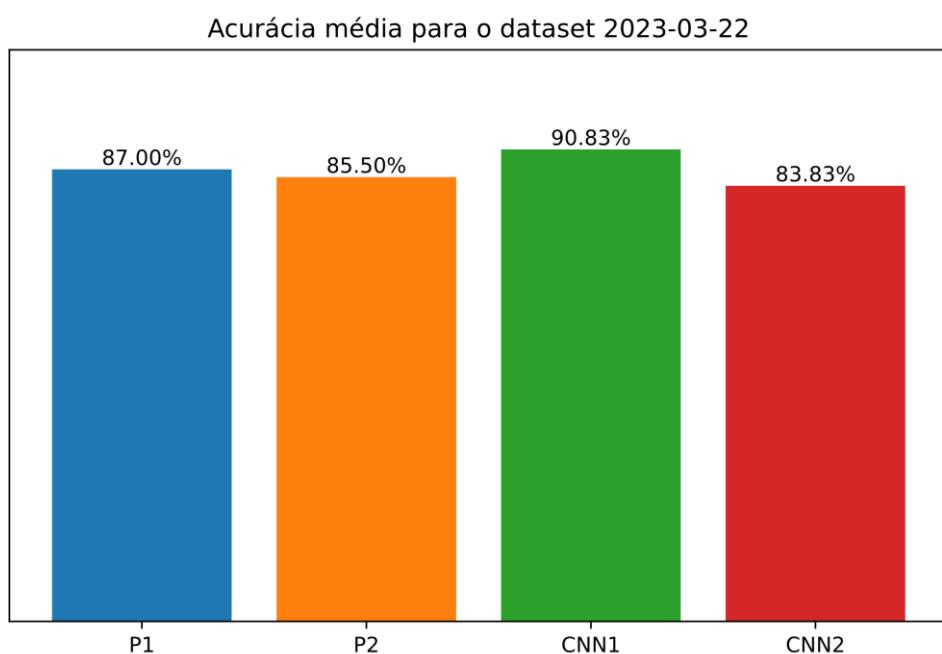
Essas variações permitem que o modelo aprenda a reconhecer as mesmas características em diferentes posições e orientações, o que pode levar a um desempenho geral melhor em dados de teste não vistos.

4.6 NOVOS RESULTADOS

Após a criação do novo pré-processamento, foram realizados os mesmos testes anteriores, porém, dessa vez foi possível reduzir ainda mais a resolução da imagem sem redução significativa da acurácia. Por esse motivo, foram utilizadas imagens no formato 128 x 128 x 1 ao invés de 512 x 512 x 1, portanto, os modelos utilizados serão os mesmos, apenas adaptando a camada de entrada para corresponder à nova resolução das imagens.

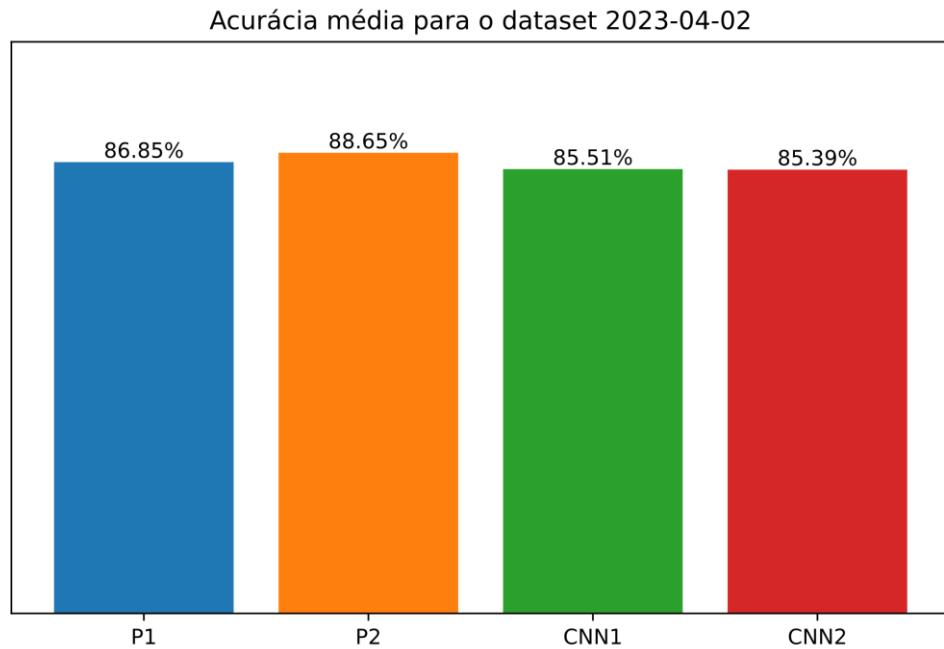
A Figura 4.17, Figura 4.18 e Figura 4.19 apresentam as acurárias médias para cada versão do dataset.

Figura 4.17 – Acurácia média para o dataset 2023-03-22 com o novo pré-processamento



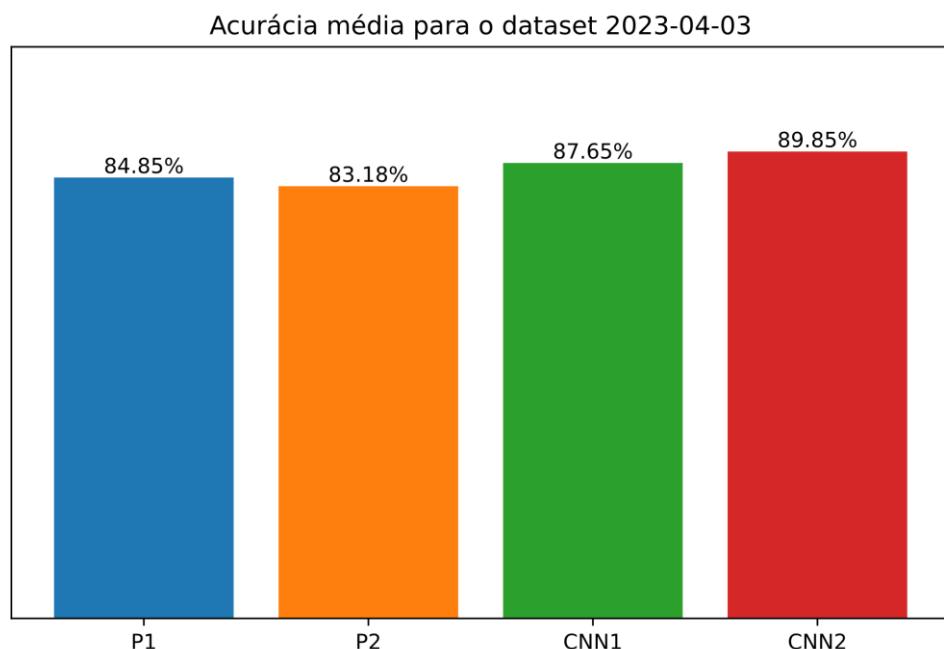
Fonte: Autor.

Figura 4.18 – Acurácia média para o dataset 2023-04-02 com o novo pré-processamento



Fonte: Autor.

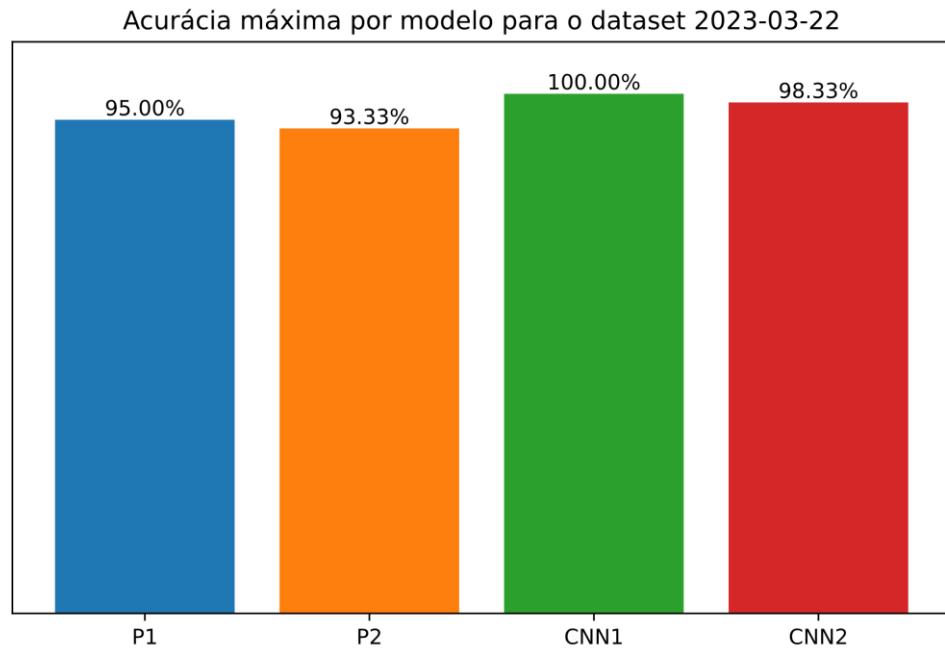
Figura 4.19 – Acurácia média para o dataset 2023-04-03 com o novo pré-processamento



Fonte: Autor.

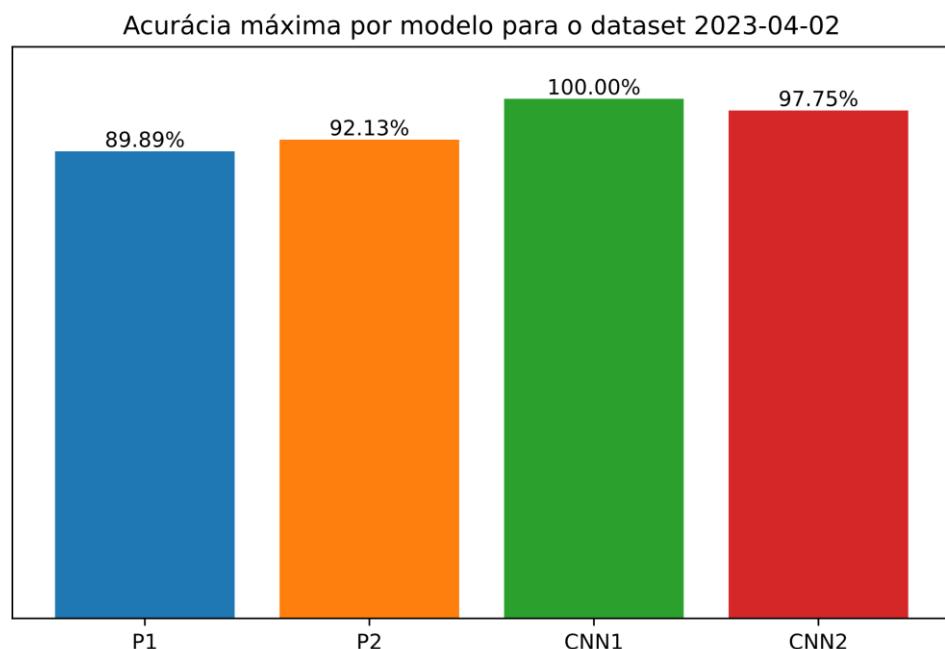
As acuráncias máximas por modelo e versão do dataset são mostradas através da Figura 4.20, Figura 4.21 e Figura 4.22.

Figura 4.20 – Acurácia máxima para o dataset 2023-03-22 com o novo pré-processamento



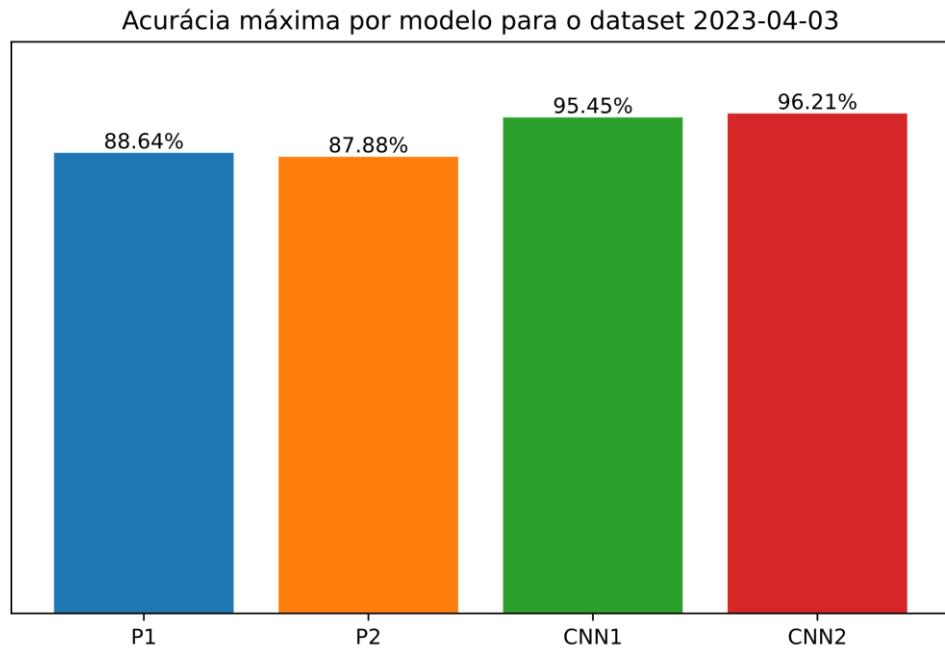
Fonte: Autor.

Figura 4.21 – Acurácia máxima para o dataset 2023-04-02 com o novo pré-processamento



Fonte: Autor.

Figura 4.22 – Acurácia máxima para o dataset 2023-04-03 com o novo pré-processamento



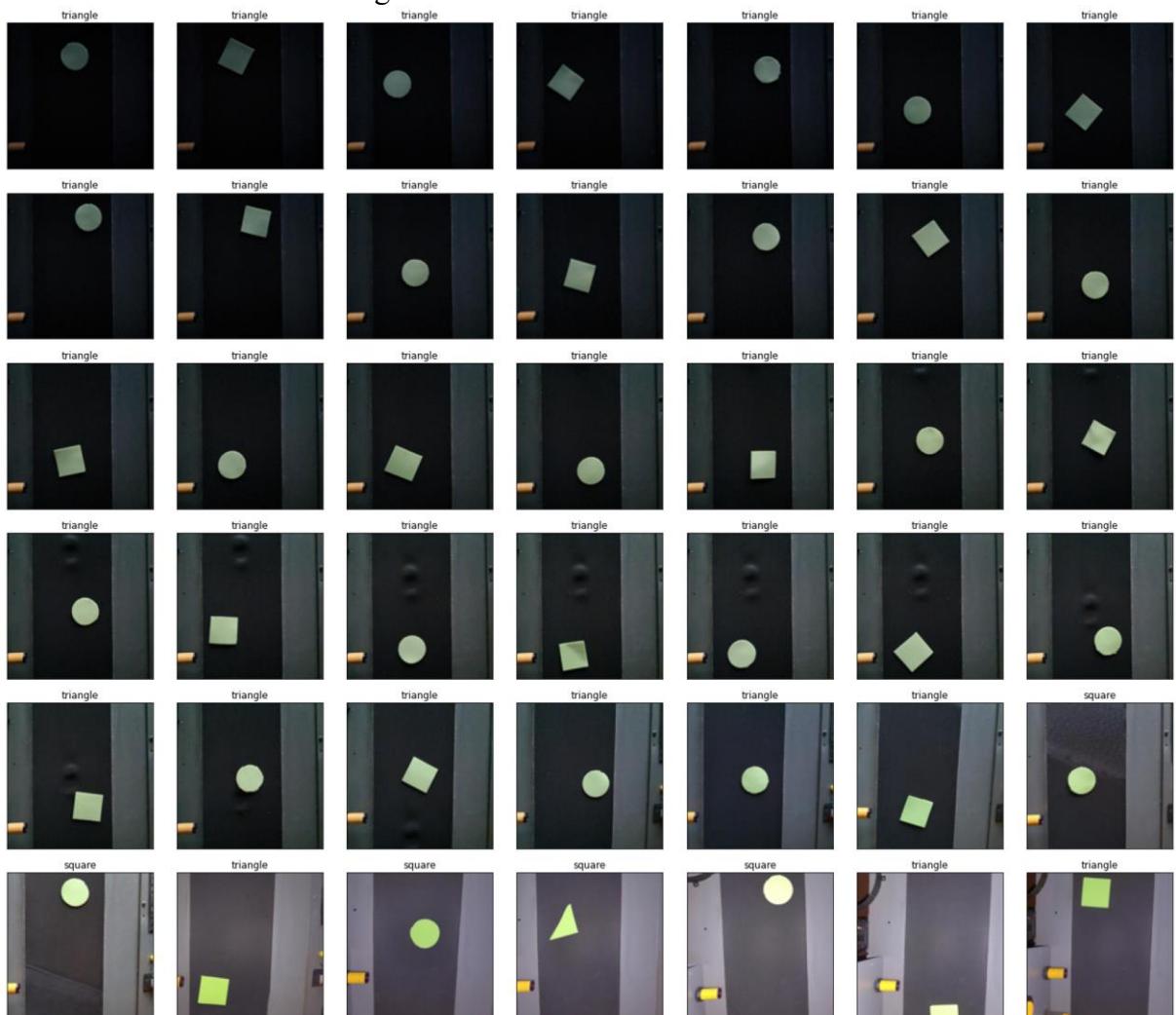
Fonte: Autor.

Percebe-se um aumento significativo da acurácia em relação aos resultados anteriores. Também não há queda de precisão da versão 2023-03-22 para a versão 2023-04-02 como ocorreu anteriormente, pois independentemente do tipo de iluminação, as figuras geométricas terão o mesmo formato após o novo pré-processamento. Na verdade, houve um aumento da acurácia, pois a variedade do dataset foi ampliada.

Apesar dos modelos Perceptron e CNNs alcançarem uma acurácia média semelhante, as CNNs demonstraram um maior potencial, chegando a 100% de acurácia máxima em alguns casos, provando a eficácia de CNNs para reconhecimento de imagens e corroborando com trabalhos semelhantes.

As acuráncias dos modelos para o dataset na versão 2023-04-03 foram consideravelmente inferiores comparado com as outras versões. A Figura 4.23 mostra quais foram os erros de previsão cometidos pelo melhor modelo treinado (modelo com maior acurácia).

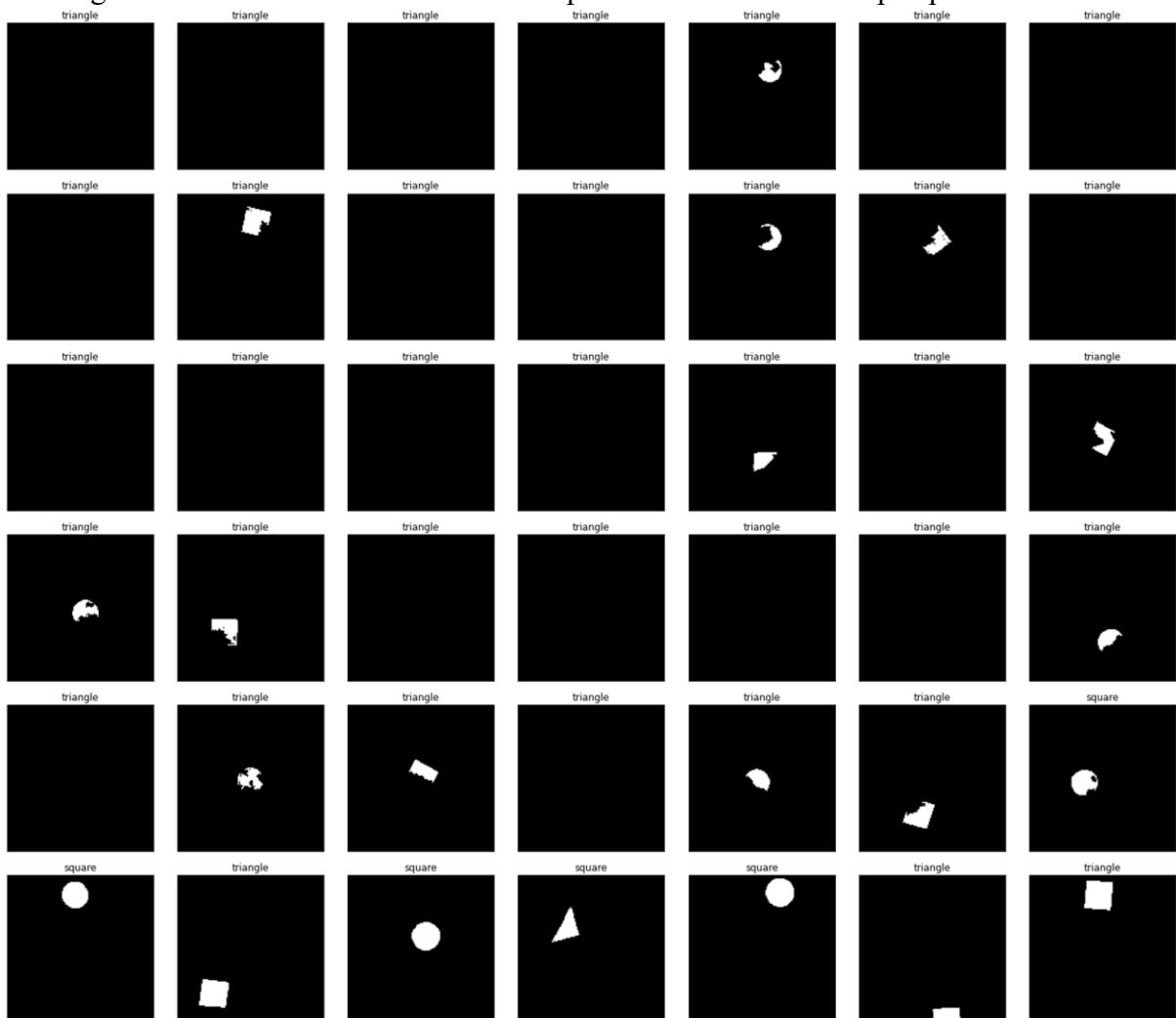
Figura 4.23 – Erros do melhor modelo



Fonte: Autor.

Percebe-se que a maioria dos erros de previsão cometidos pelo melhor modelo são das imagens com baixa iluminação. A Figura 4.24 apresenta essas mesmas figuras, porém na sua forma após o novo pré-processamento.

Figura 4.24 – Erros do melhor modelo apresentados com o novo pré-processamento



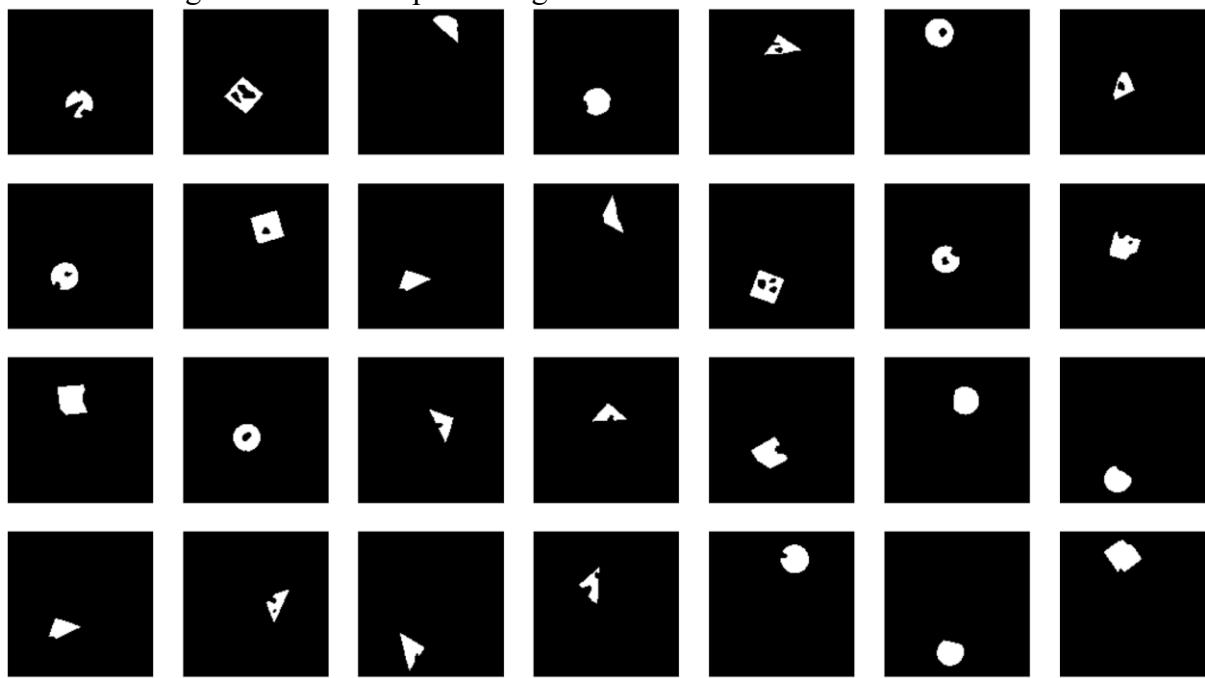
Fonte: Autor

Através da Figura 4.24, pode-se perceber que a causa dos erros de previsão do modelo foi a baixa iluminação, cujo novo pré-processamento não foi capaz de simplificar. Por essa razão, as imagens com baixa iluminação do dataset 2023-04-03 foram descartadas.

4.7 DANIFICAÇÃO DE FIGURAS

Após a criação, treinamento e teste do dataset de figuras adequadas com resultados positivos, foi criado um novo dataset intitulado 2023-04-23 com a adição de 171 imagens em relação ao dataset anterior, sendo 57 quadrados danificados, 57 círculos danificados e 57 triângulos danificados. Algumas dessas imagens são apresentadas através da Figura 4.25 já com o pré-processamento aplicado.

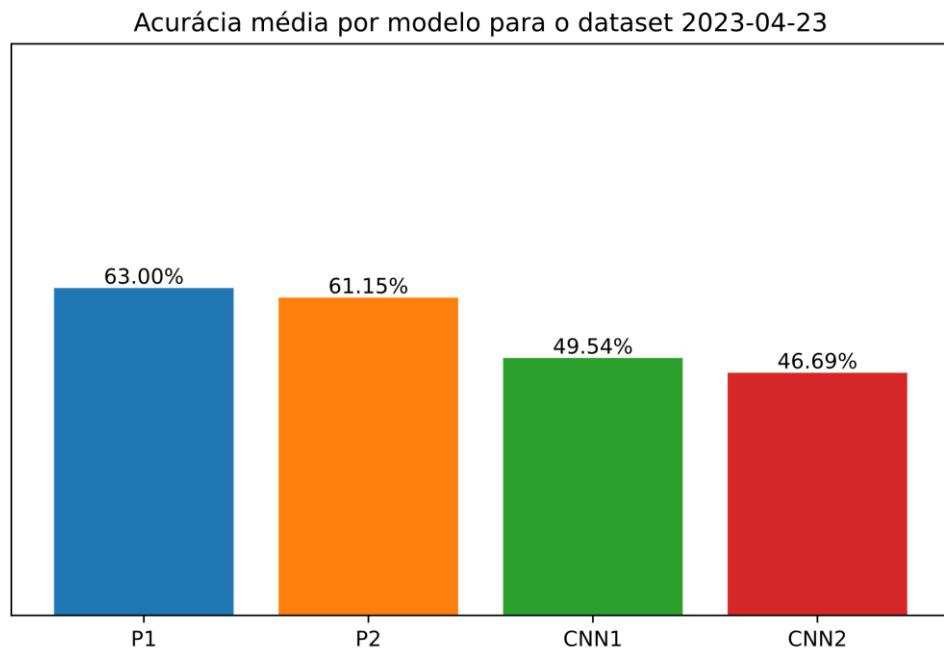
Figura 4.25 – Exemplos de figuras danificadas do dataset 2023-04-23



Fonte: Autor.

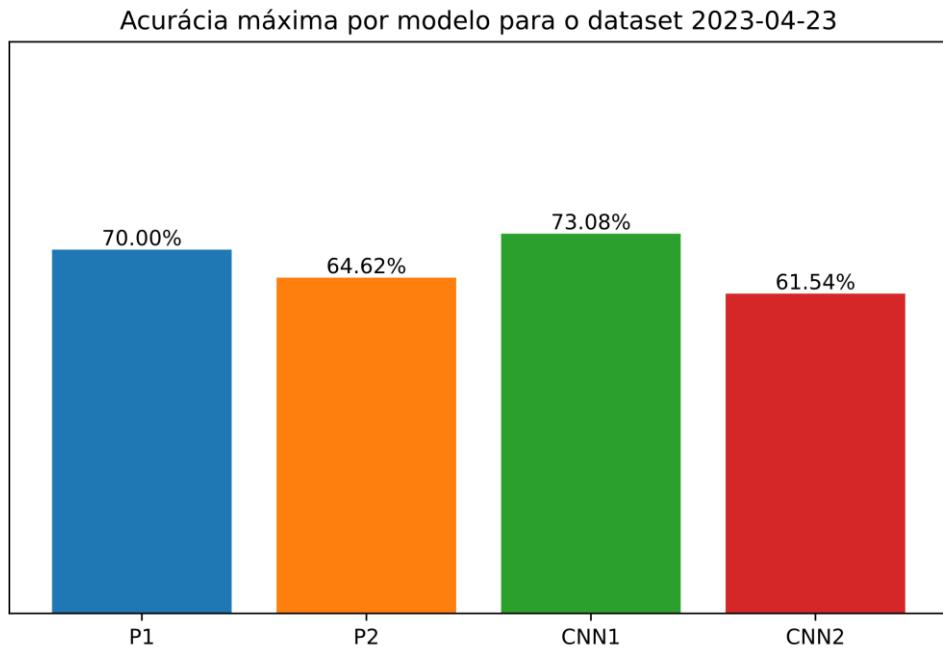
Conforme o procedimento realizado nos datasets anteriores, o dataset 2023-04-23 foi submetido a 10 testes por modelo, sendo esses modelos os mesmos anteriores, porém com 6 saídas ao invés de 3. Os resultados de acurácia média e máxima são apresentados através da Figura 4.26 e Figura 4.27 respectivamente.

Figura 4.26 – Acurácia média por modelo para o dataset 2023-04-23



Fonte: Autor.

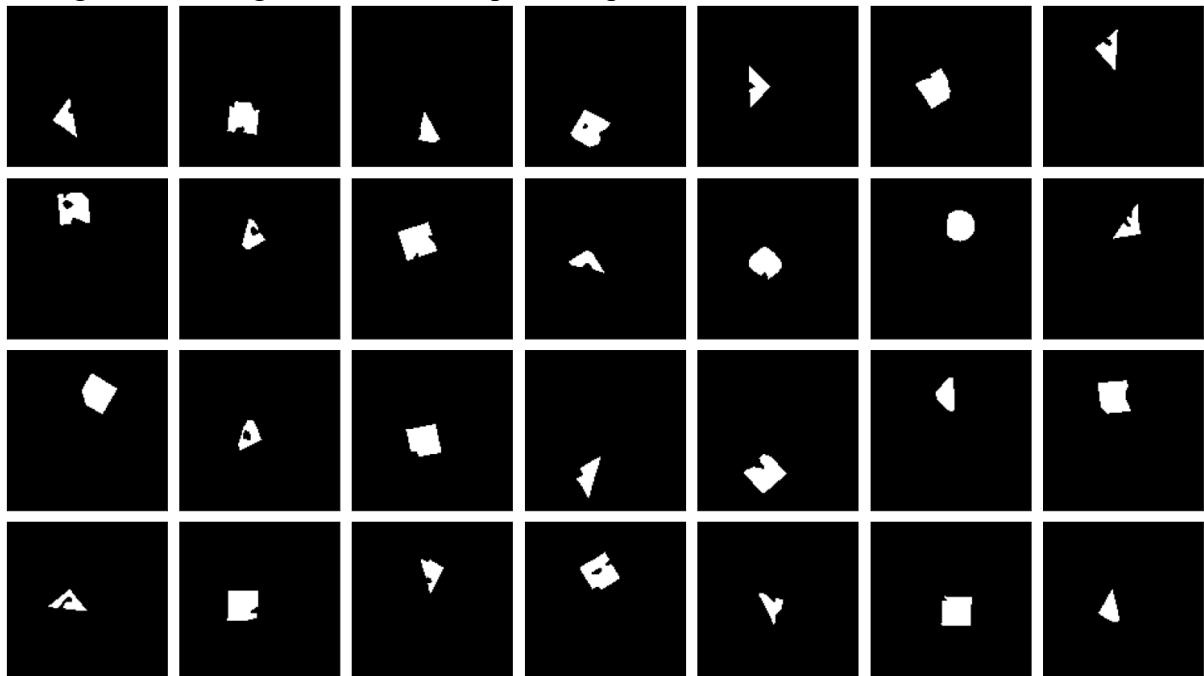
Figura 4.27 – Acurácia máxima por modelo para o dataset 2023-04-23



Fonte: Autor.

Percebe-se através dos gráficos que o resultado não foi satisfatório. Por esse motivo, foi realizada uma inspeção do melhor modelo cuja acurácia foi de 73,08% para identificar os erros cometidos. Algumas das imagens interpretadas incorretamente pelo modelo são apresentadas através da Figura 4.28.

Figura 4.28 – Figuras com erro de previsão pelo melhor modelo do dataset 2023-04-23



Fonte: Autor.

Percebe-se que os erros cometidos pelo modelo são majoritariamente de figuras danificadas, o que é plausível devido a menor quantidade de dados referentes a essa categoria de figuras.

4.8 DANIFICAÇÃO AUTOMÁTICA DE FIGURAS

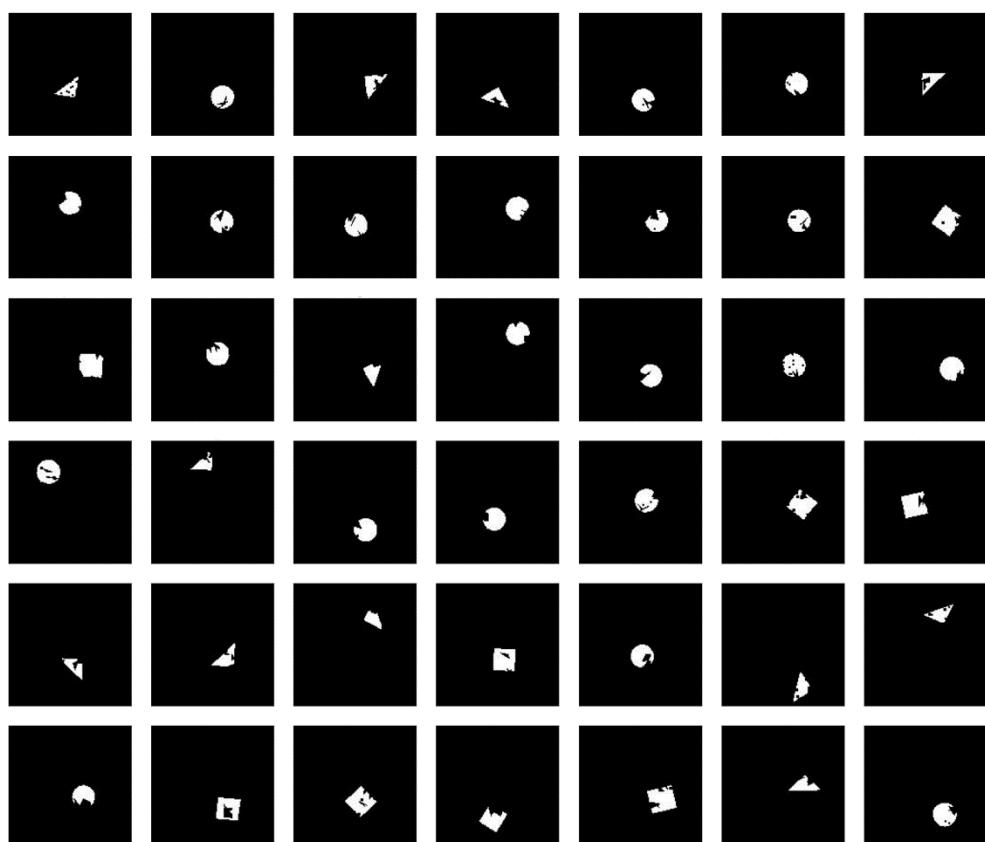
Devido ao resultado insatisfatório das acuráncias dos modelos aplicados ao dataset 2023-04-23, foi percebida a necessidade de ampliar grandemente a base de dados para que fossem alcançadas acuráncias maiores, porém, apenas danificar as peças de forma a ter a mesma quantidade de figuras adequadas e danificadas poderia não ser o suficiente, além de demandar muito tempo para fazê-lo. Por esse motivo, foi criado um algoritmo para reutilizar as imagens de figuras adequadas para danificá-las digitalmente de forma automática, seguindo padrões para se assemelhar o máximo possível com figuras danificadas manualmente.

Os defeitos automatizados foram aplicados através das seguintes etapas:

- Criação de polígono de 3 a 10 arestas posicionado de forma aleatória sobre a imagem original;
- Continuar criando polígonos até que a imagem tenha menos de 99,8% de similaridade com a imagem original (valor escolhido para se assemelhar com defeitos manuais);
- Aplicação de ruído com aglomerados de 2 a 4 pixels sobre a imagem original com 50% de probabilidade;
- Caso a figura tenha desaparecido completamente devido aos filtros, os polígonos são removidos, aplicando apenas o ruído.

O resultado da aplicação automática de defeitos é apresentado através da Figura 4.29.

Figura 4.29 – Figuras com defeitos automáticos

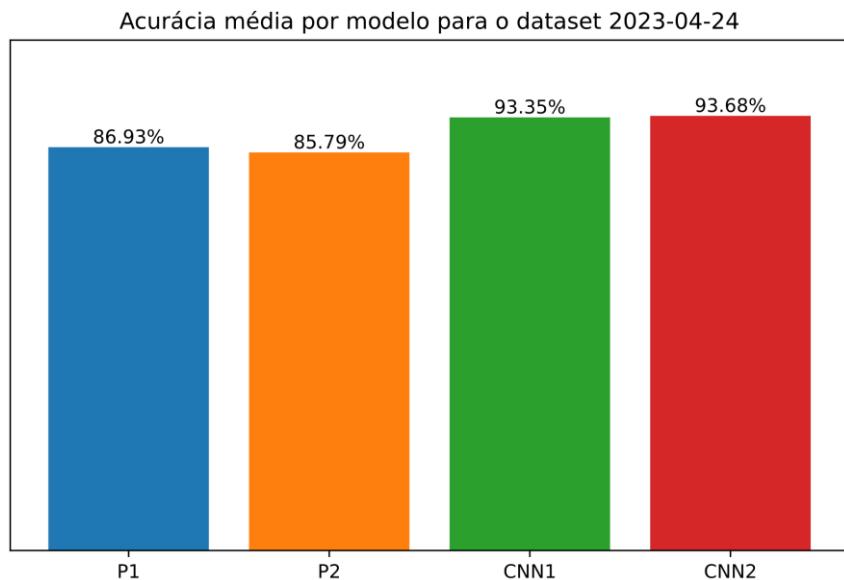


Fonte: Autor.

Para acompanhar o crescente número de figuras danificadas, o número de figuras adequadas também foi ampliado com as mesmas técnicas de variações utilizadas no pré-processamento, resultando em um novo dataset intitulado 2023-04-24 com 9405 figuras.

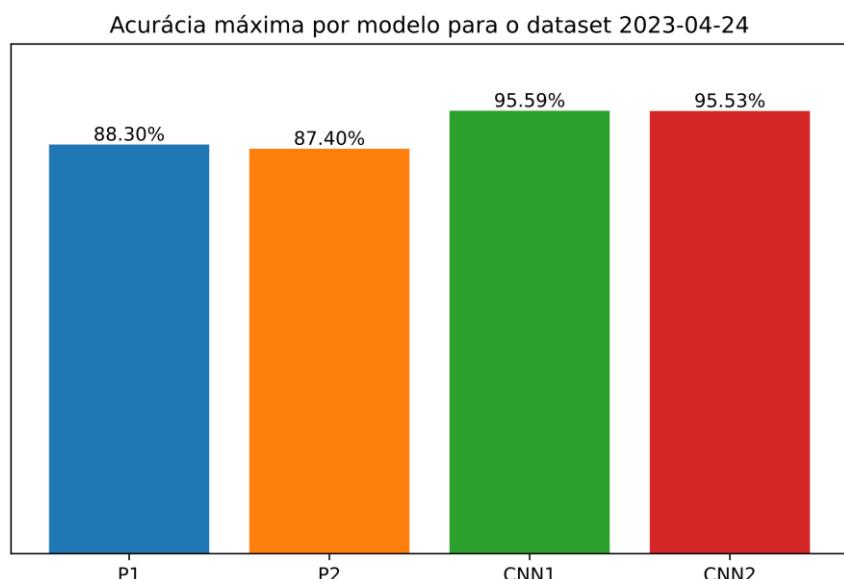
Realizando o treinamento e teste dos mesmos modelos para o novo dataset 2023-04-24, foram obtidos os valores de acurácia média, mostrados através da Figura 4.30 e acurácia máxima, mostrados através da Figura 4.31.

Figura 4.30 – Acurácia média por modelo para o dataset 2023-04-24



Fonte: Autor.

Figura 4.31 – Acurácia máxima por modelo para o dataset 2023-04-24

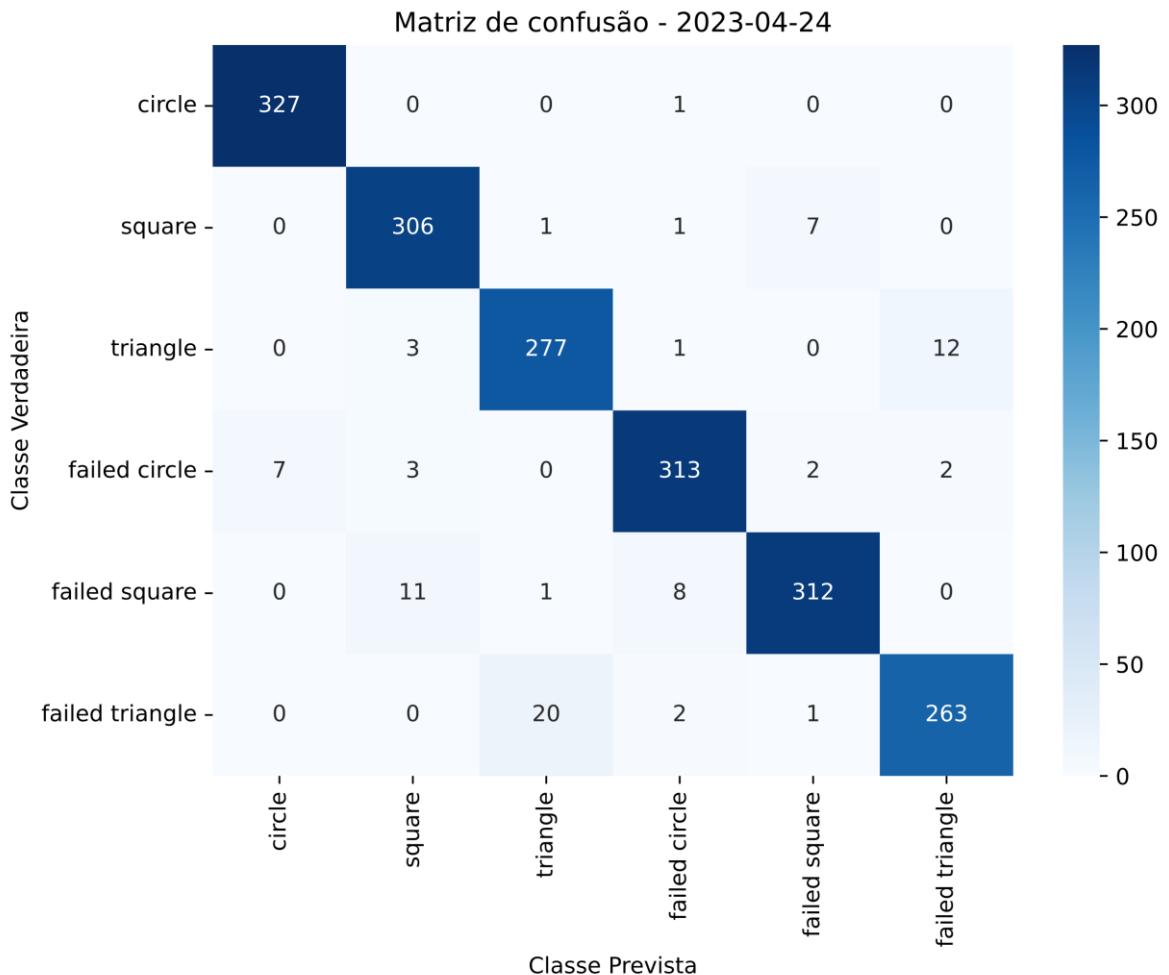


Fonte: Autor.

Graças a ampliação automática do dataset, foram obtidas acuráncias satisfatórias, pois o grande número de imagens diferentes impede o overfitting, tornando o modelo mais capaz de prever figuras desconhecidas (diferentes das utilizadas durante o treinamento).

Para o melhor modelo cuja acurácia foi de 95,59%, foi criada a matriz de confusão mostrada na Figura 4.32.

Figura 4.32 – Matriz de confusão para o dataset 2023-04-24



Fonte: Autor.

A partir da matriz de confusão apresentada na Figura 4.32, foram calculados os indicadores:

- Acurácia: 95,59%;
- Precisão: 95,60%;
- Sensibilidade: 95,59%;
- F1-score: 95,58%.

Os indicadores de desempenho do modelo apresentam valores próximos entre si, o que denota consistência e equilíbrio na tarefa de classificação. A acurácia, com um percentual de 95,59%, revela que a maioria das predições realizadas pelo modelo é correta em relação ao total de predições. A precisão, com um valor de 95,60%, indica que quando o modelo realiza uma predição positiva, há uma alta probabilidade de estar correta. A sensibilidade, também conhecida como recall, obtém um resultado de 95,59%, o que demonstra a capacidade do modelo de identificar corretamente a maioria dos casos positivos. O F1-score, uma medida que

combina precisão e sensibilidade em um único valor, alcança 95,58%, representando um bom equilíbrio entre essas duas métricas.

Com base na matriz de confusão mostrada na Figura 4.32 observa-se que o modelo apresentou uma tendência a cometer erros, principalmente em relação à detecção de defeitos em figuras triangulares. Essa análise destacada na matriz de confusão pode direcionar esforços para aprimorar o desempenho do modelo em relação a essa classe específica, visando reduzir os falsos positivos e falsos negativos associados a triângulos.

Um aspecto adicional relevante para avaliar o desempenho de um modelo é a curva de aprendizagem (Ozkahraman e Livatyalı, 2022). Ela é empregada para analisar o progresso de um modelo durante o processo de treinamento. Por exemplo, é possível traçar o custo do treinamento ao longo do tempo para observar como o modelo está aprendendo. Entretanto, vale ressaltar que, para o escopo deste trabalho, não será necessário realizar uma avaliação específica da curva de aprendizagem, visto que a simplicidade do conjunto de dados utilizado não demanda um treinamento significativamente demorado.

Para validar a qualidade da acurácia de 95,59% obtida pelo modelo, foi realizada uma comparação com a acurácia humana na classificação de imagens. Essa análise permite avaliar a eficiência do modelo em desempenhar o trabalho de classificação, que anteriormente era realizado por operadores humanos. Ao comparar a acurácia do melhor modelo, que foi de 95,59%, com o erro humano na classificação de imagens, que foi de 5,1% de acordo com Russakovsky et al. (2014), é possível estimar a acurácia humana em 94,9%. Portanto, com base nessa comparação, pode-se inferir que o modelo possui uma capacidade de classificação de figuras com uma precisão bastante semelhante à dos seres humanos.

Outra forma de validar a acurácia do modelo é compará-la com estudos e trabalhos semelhantes. Por exemplo, foi relatada a criação de um sistema de segurança baseado em câmera 2D para células de trabalho robóticas sem cerca, no qual foi alcançada uma acurácia de 95% (Ozkahraman and Livatyalı, 2022). Além disso, em pesquisas relacionadas à classificação de doenças, detecção de objetos e segmentação em radiografias de tórax, modelos de Deep Learning alcançaram uma acurácia de 94,6% (Santomartino et al., 2023). Outro exemplo inclui a classificação de objetos abandonados, removidos e movidos em vídeos de vigilância, com uma acurácia de 90% (Fitzsimons and Dawson-Howe, 2016). Portanto, ao comparar a acurácia do modelo em questão com esses trabalhos, é possível constatar que o desempenho alcançado é consistente com os resultados obtidos em aplicações similares. Essas referências fornecem uma validação adicional para a acurácia do modelo, demonstrando que ele está em níveis competitivos em relação a outros sistemas e modelos desenvolvidos para tarefas semelhantes.

4.9 SISTEMA DE CLASSIFICAÇÃO NA ESTEIRA TRANSPORTADORA

Sendo definida a rede neural mais apropriada para a seleção e classificação dos objetos, o modelo foi aplicado em uma bancada didática que simula um processo industrial de seleção de itens. A bancada é controlada por um microcontrolador ESP32, um Arduino e um smartphone que utiliza a câmera para coleta das imagens dos objetos através de um aplicativo mobile representado na Figura 4.2. Foi desenvolvido um sistema com o propósito de estabelecer a comunicação entre o aplicativo mobile e os servos motores que operam na esteira transportadora. O principal objetivo dessa implementação é viabilizar a capacidade de classificação na esteira transportadora. Nesse contexto, uma nova tela foi adicionada ao aplicativo, especificamente para o modo "automação", conforme demonstrado na Figura 4.33.

Figura 4.33 – Modo automação do aplicativo mobile

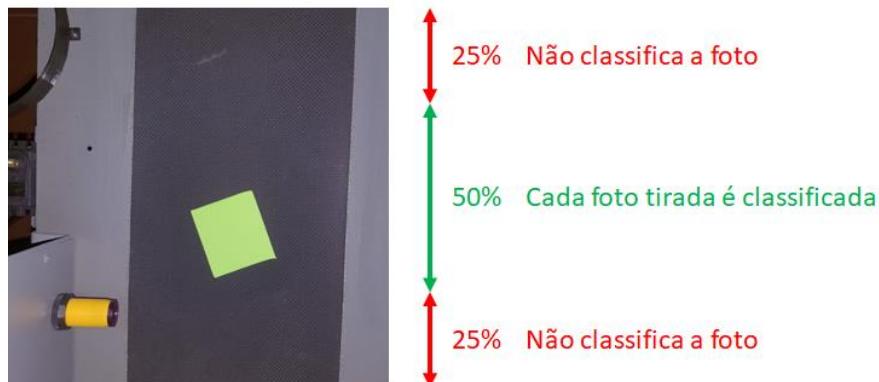


Fonte: Autor.

Ao acionar o botão localizado na parte inferior da tela, é iniciada uma sequência de captura de fotos de forma automática, com um intervalo de 3 segundos entre cada foto. Cada imagem capturada é enviada para o servidor que abriga a rede neural correspondente. Posteriormente, o servidor processará a imagem e fornecerá uma resposta que contém informações sobre a distância da figura em relação às bordas da imagem, expressa em porcentagem. Além disso, a resposta também conterá a classificação das figuras e indicará se uma figura está presente na foto ou não.

No caso de uma fotografia apresentar uma figura geométrica contida dentro da margem de 25% das bordas inferior e superior da imagem, conforme ilustrado na Figura 4.34, sua classificação será registrada e adicionada a uma lista de previsões. Esse processo é repetido continuamente até que a figura deixe a margem de enquadramento estabelecida. Quando isso ocorre, é efetuada uma contagem das previsões realizadas, sendo que a classificação mais frequente é considerada a classificação final para a fotografia em questão, como mostrado na Figura 4.35.

Figura 4.34 – Margem de enquadramento



Fonte: Autor.

Figura 4.35 – Contagem de predições



Fonte: Autor.

Após a classificação final da figura, o resultado é convertido em uma tabela que representa o status da esteira, conforme exemplificado na Figura 4.36. Essa tabela tem como objetivo mapear a posição de cada servo motor com base no tipo de figura identificado, permitindo assim a separação adequada das figuras com base em seu tipo e possíveis defeitos.

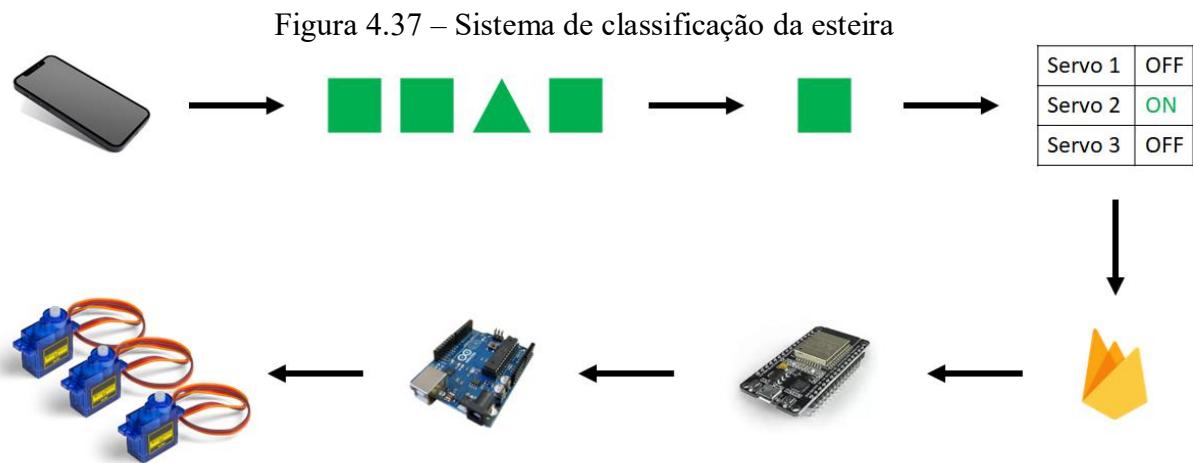
Figura 4.36 – Mapeamento do status da esteira

<table border="1"> <tbody> <tr><td>Servo 1</td><td>ON</td></tr> <tr><td>Servo 2</td><td>OFF</td></tr> <tr><td>Servo 3</td><td>OFF</td></tr> </tbody> </table>	Servo 1	ON	Servo 2	OFF	Servo 3	OFF	<table border="1"> <tbody> <tr><td>Servo 1</td><td>OFF</td></tr> <tr><td>Servo 2</td><td>OFF</td></tr> <tr><td>Servo 3</td><td>OFF</td></tr> </tbody> </table>	Servo 1	OFF	Servo 2	OFF	Servo 3	OFF
Servo 1	ON												
Servo 2	OFF												
Servo 3	OFF												
Servo 1	OFF												
Servo 2	OFF												
Servo 3	OFF												
<table border="1"> <tbody> <tr><td>Servo 1</td><td>OFF</td></tr> <tr><td>Servo 2</td><td>ON</td></tr> <tr><td>Servo 3</td><td>OFF</td></tr> </tbody> </table>	Servo 1	OFF	Servo 2	ON	Servo 3	OFF	<table border="1"> <tbody> <tr><td>Servo 1</td><td>OFF</td></tr> <tr><td>Servo 2</td><td>OFF</td></tr> <tr><td>Servo 3</td><td>OFF</td></tr> </tbody> </table>	Servo 1	OFF	Servo 2	OFF	Servo 3	OFF
Servo 1	OFF												
Servo 2	ON												
Servo 3	OFF												
Servo 1	OFF												
Servo 2	OFF												
Servo 3	OFF												
<table border="1"> <tbody> <tr><td>Servo 1</td><td>OFF</td></tr> <tr><td>Servo 2</td><td>OFF</td></tr> <tr><td>Servo 3</td><td>ON</td></tr> </tbody> </table>	Servo 1	OFF	Servo 2	OFF	Servo 3	ON	<table border="1"> <tbody> <tr><td>Servo 1</td><td>OFF</td></tr> <tr><td>Servo 2</td><td>OFF</td></tr> <tr><td>Servo 3</td><td>OFF</td></tr> </tbody> </table>	Servo 1	OFF	Servo 2	OFF	Servo 3	OFF
Servo 1	OFF												
Servo 2	OFF												
Servo 3	ON												
Servo 1	OFF												
Servo 2	OFF												
Servo 3	OFF												

Fonte: Autor.

O aplicativo móvel, tendo conhecimento da tabela de status da esteira, envia essas informações para o banco de dados em tempo real Firebase Realtime Database. Com isso, um dispositivo ESP32 é capaz de acessar essas informações continuamente, verificando o estado atualizado da esteira no banco de dados e transmitindo essas informações para um Arduino. O

Arduino, por sua vez, é responsável por controlar o ângulo de cada servo motor e executar a separação adequada das figuras geométricas, de acordo com as informações recebidas. A Figura 4.37 ilustra a sequência de etapas para o funcionamento desse sistema.



Fonte: Autor.

O sistema descrito foi implementado na prática e o resultado pode ser conferido através do vídeo disponível no seguinte endereço: <https://youtu.be/fukCSafNtTA>. O vídeo demonstra o funcionamento completo do sistema de classificação de formas geométricas, no qual os atuadores são movimentados de acordo com o tipo de figura identificada pela rede neural.

5 CONCLUSÃO

Os métodos de Inteligência Artificial são ferramentas poderosas para a automação de processos, análise de dados e tomada de decisão. Estas tecnologias estão em constante evolução e têm a capacidade de revolucionar diversas áreas, tornando os processos mais eficientes e rápidos. O Deep Learning se destaca devido à sua capacidade de resolver problemas complexos com mais precisão e rapidez do que outras tecnologias de Inteligência Artificial. Esta tecnologia usa redes neurais para autoaprendizagem, semelhante ao funcionamento do cérebro, e pode aplicar o que aprendeu para identificar padrões em grandes conjuntos de dados. Além disso, o Deep Learning é a ferramenta mais adequada para reconhecimento de imagens. No entanto, o uso destas tecnologias depende amplamente da disponibilidade e qualidade dos dados, bem como da compreensão da natureza dos problemas a serem resolvidos. Também é importante considerar que a Inteligência Artificial pode ser utilizada de forma segura e eficaz para aumentar a produtividade e melhorar a qualidade de vida de todos.

O sistema de reconhecimento de imagem proposto neste trabalho pode ser uma grande contribuição para a indústria de processos, pois foi capaz de categorizar os objetos e identificar os defeitos. A solução proposta para a identificação automatizada de objetos danificados em processos industriais é eficaz e promissora. A utilização de redes neurais artificiais para reconhecimento de padrões de tipo e defeitos demonstrou uma acurácia (95,59%) aproximada àquela alcançada pelo ser humano (94,9%) e também comparável a outros estudos similares.

Foi observado que a eficácia de uma rede neural é amplamente dependente da qualidade do conjunto de dados utilizado, bem como da forma como os dados de entrada são apresentados ao algoritmo de inteligência artificial. Constatou-se que a realização de um pré-processamento mais elaborado e a criação de um maior volume de dados tiveram um impacto significativo, superando a influência da complexidade e do número de parâmetros do modelo. Após a aplicação do pré-processamento, foi identificado um aumento de aproximadamente 50% no valor da acurácia média para as redes convolucionais utilizadas. A geração automática de figuras com defeitos exerceu um papel crucial no desempenho do modelo, uma vez que possibilitou a ampliação dos dados disponíveis para treinamento em qualquer quantidade desejada.

É importante ressaltar que o processo de desenvolvimento e aprimoramento do sistema de reconhecimento de imagem é contínuo e dependente dos resultados desejados e do cenário específico de análise. O tipo de imagem a ser classificada, bem como a qualidade do conjunto de dados disponível, exercem uma influência significativa nos resultados obtidos. O pré-processamento dos dados desempenha um papel crucial para garantir a qualidade e a relevância das informações fornecidas ao modelo de inteligência artificial. A compreensão de que sempre há espaço para melhorias e otimizações é fundamental, tanto na coleta e seleção dos dados quanto na aplicação de técnicas de pré-processamento. Além disso, é recomendado que a rede neural seja atualizada constantemente, incorporando novas imagens coletadas ao longo do tempo. Dessa forma, é possível manter o sistema atualizado e garantir sua eficácia contínua, permitindo a detecção precisa de objetos danificados em processos industriais.

5.1 TRABALHOS FUTUROS

Para trabalhos futuros, sugere-se a implementação de redes neurais treinadas nas condições de dataset mencionadas na Seção 7.1, que envolve a combinação de dados de figuras danificadas criadas tanto manualmente quanto automaticamente. No entanto, recomenda-se realizar esses treinamentos utilizando recursos computacionais de alto nível, visando obter resultados ainda mais aprimorados. Essa abordagem permitiria explorar o potencial da união

dessas duas fontes de dados, aproveitando os benefícios de cada uma delas e melhorando o desempenho geral do modelo de reconhecimento de defeitos em processos industriais.

6 REFERÊNCIAS

- ABHISHEK, K. Introduction to artificial intelligence. **Red gate**, 2022. Disponível em: <[ALEX. Feedforward Neural Networks and Multilayer Perceptrons. **boostedml**, 2020. Disponível em: <<https://boostedml.com/2020/04/feedforward-neural-networks-and-multilayer-perceptrons.html>>. Acesso em: 14 nov. 2022.

AS 10 Principais Arquiteturas de Redes Neurais. **Deep Learning Book**, 2022. Disponível em: <<https://www.deeplearningbook.com.br/as-10-principais-arquiteturas-de-redes-neurais/>>. Acesso em: 14 nov. 2022.

BAHETI, P. Activation Functions in Neural Networks \[12 Types & Use Cases\]. **v7labs**, 2022. Disponível em: <<https://www.v7labs.com/blog/neural-networks-activation-functions>>. Acesso em: 14 nov. 2022.

BERT, M. et al. Introducing TensorFlow Decision Forests. **TensorFlow**, 2021. Disponível em: <<https://blog.tensorflow.org/2021/05/introducing-tensorflow-decision-forests.html>>. Acesso em: 14 nov. 2022.

BRAGA, P.; SILVA, J. Aplicação de Redes Neurais Artificiais em Conjunto com o Método Vetorial da Propagação de Feixes na Análise de um Acoplador Direcional Baseado em Fibra Ótica. **Revista brasileira de computação aplicada**, 2011. 16.

CECCON, D. Funções de ativação: definição, características, e quando usar cada uma. **iaexpert**, 2020. Disponível em: <<https://iaexpert.academy/2020/05/25/funcoes-de-ativacao-definicao-caracteristicas-e-quando-usar-cada-uma/>>. Acesso em: 10 nov. 2022.

CECCON, D. Os tipos de redes neurais. **iaexpert**, 2020. Disponível em: <<https://iaexpert.academy/2020/06/08/os-tipos-de-redes-neurais/>>. Acesso em: 10 nov. 2022.

CIPULLO, G. INDÚSTRIA 5.0: CONHEÇA O NOVO CONCEITO DE PRODUÇÃO. **korp**, 2021. Disponível em: <<https://www.korp.com.br/industria-5-0-conheca-o-novo-conceito-de-producao/>>. Acesso em: 27 nov. 2022.

CLAUDIO, V. Machine Learning na saúde: conheça algumas aplicações. **Portal Telemedicina**, 19 dez. 2021. Disponível em: <<https://portaltelemedicina.com.br/blog/machine-learning-na-saude-conheca-algunas-aplicacoes>>. Acesso em: 10 nov. 2022.

DAS, D. A. Optimizers in Deep Learning. **medium**, 2020. Disponível em: <<https://medium.com/analytics-vidhya/this-blog-post-aims-at-explaining-the-behavior-of-different-algorithms-for-optimizing-gradient-46159a97a8c1>>. Acesso em: 14 nov. 2022.

GEEKSFORGEEKS. Decision Tree. **GeeksforGeeks**, 2022. Disponível em: <<https://www.geeksforgeeks.org/decision-tree/>>. Acesso em: 14 nov. 2022.](https://www.red-gate.com/simple-talk/development/data-science-development/introduction-to-artificial-intelligence/#:~:text=Artificial%20intelligence%20(AI)%20is%20the,many%20other%20products%20and%20services.>)

GONÇALVES, F. 8 Aplicações de Inteligência Artificial no Setor de Energia. **venturus**, 2020. Disponível em: <<https://www.venturus.org.br/8-aplicacoes-de-inteligencia-artificial-no-setor-de-energia/>>. Acesso em: 27 nov. 2022.

GUSMÃO, A. 5 princípios de aprendizagem por reforço explicados pelo especialista em IA, Hadelin de Ponteves. **Tempo de inovação**, 2020. Disponível em: <<https://tempodeinovacao.com.br/5-principios-de-aprendizagem-por-reforco-explicados-pelo-especialista-em-ia-hadelin-de-ponteves/>>. Acesso em: 15 nov. 2022.

HENRIQUE, J. Uma breve introdução a Redes neurais e "Deep Learning". **medium**, 2018. Disponível em: <https://medium.com/@joohenrique_7804/uma-breve-introdu%C3%A7%C3%A3o-a-redes-neurais-e-deep-learning-243bf283f731>. Acesso em: 14 nov. 2022.

HINTON, G. E. Deep belief networks. **scholarpedia**, 2009. Disponível em: <http://www.scholarpedia.org/article/Deep_belief_networks>. Acesso em: 14 nov. 2022.

ILUMEO. Um tour pelos 10 principais algoritmos de Machine Learning. **Ilumeo**, 2020. Disponível em: <<https://ilumeo.com.br/todos-posts/2020/06/22/um-tour-pelos-10-principais-algoritmos-de-machine-learning>>. Acesso em: 10 nov. 2022.

INTRODUÇÃO aos codificadores automáticos. **TensorFlow**, 2022. Disponível em: <<https://www.tensorflow.org/tutorials/generative/autoencoder>>. Acesso em: 14 nov. 2022.

JACOB, T. Vanishing Gradient Problem, Explained. **kdnuggets**, 2022. Disponível em: <<https://www.kdnuggets.com/2022/02/vanishing-gradient-problem.html>>. Acesso em: 14 nov. 2022.

KLEININGS, H. Industry 5.0: Leadership in the Era of AI. **levity**, 2022. Disponível em: <<https://levity.ai/blog/leadership-ai#:~:text=Simply%20put%2C%20Industry%205.0%20adds,fully%20replace%20the%20human%20workforce.>>. Acesso em: 27 nov. 2022.

LECON. Quais as Diferenças entre Machine Learning e Deep Learning? **Lecon**, 2022. Disponível em: <<https://www.lecom.com.br/blog/machine-learning-e-deep-learning/>>. Acesso em: 10 nov. 2022.

LENINE, F. Como selecionar atributos para resolver a maldição da dimensionalidade. **Medium**, 2017. Disponível em: <<https://medium.com/@fabiolenine/como-selecionar-atributos-para-resolver-a-maldi%C3%A7%C3%A3o-da-dimensionalidade-5c810bc8449f#:~:text=A%20maldi%C3%A7%C3%A3o%20da%20dimensionalidade%20diz,atributos%2C%20mesmo%20que%20sejam%20%C3%BAteis.>>. Acesso em: 18 nov. 2022.

LESSA, B. Conheça os métodos de machine learning mais populares. **sga**, 2017. Disponível em: <<https://sga.com.br/conheca-os-metodos-de-machine-learning-mais-populares/>>. Acesso em: 10 nov. 2022.

LIMA, A. ML | DESCIDA GRADIENTE ESTOCÁSTICO (SGD). **Acervo Lima**, 2022. Disponível em: <<https://acervolima.com/ml-descida-gradiente-estocastico-sgd/>>. Acesso em: 14 nov. 2022.

MARR, B. How Is Artificial Intelligence And Machine Learning Used In Engineering? **forbes**, 2020. Disponível em: <<https://www.forbes.com/sites/bernardmarr/2020/02/07/how-is-artificial-intelligence-and-machine-learning-used-in-engineering/?sh=493f4c9e4a85>>. Acesso em: 27 nov. 2022.

MAYANGLAMBAM, G. Deep Learning Optimizers. **towardsdatascience**, 2020. Disponível em: <<https://towardsdatascience.com/deep-learning-optimizers-436171c9e23f>>. Acesso em: 14 nov. 2022.

MODI, A. Uma rápida introdução à Inteligência Artificial (IA) e seus componentes. **cynoteck**, 2021. Disponível em: <<https://cynoteck.com/pt/blog-post/what-is-artificial-intelligence/>>. Acesso em: 27 nov. 2022.

OMNIA, R. Neurônio. **mundoeducacao**, 2022. Disponível em: <<https://mundoeducacao.uol.com.br/biologia/neuronios.htm>>. Acesso em: 14 nov. 2022.

SACRAMENTO, G. ÁRVORE DE DECISÃO: ENTENDA ESSE ALGORITMO DE MACHINE LEARNING. **somostera**, 2021. Disponível em: <<https://blog.somostera.com/data-science/arvores-de-decisao>>. Acesso em: 14 nov. 2022.

SAHA, S. A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way. **towardsdatascience**, 2018. Disponível em: <<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>>. Acesso em: 14 nov. 2022.

SARKAR, P. What is LDA: Linear Discriminant Analysis for Machine Learning. **knowledgehut**, 2022. Disponível em: <<https://www.knowledgehut.com/blog/data-science/linear-discriminant-analysis-for-machine-learning>>. Acesso em: 17 nov. 2022.

TANGRUAMSUB, S. Long Short-Term Memory (LSTM). **medium**, 2017. Disponível em: <<https://medium.com/@sinart.t/long-short-term-memory-lstm-e6cb23b494c6>>. Acesso em: 14 nov. 2022.

TAVASOLI, S. Top 10 Machine Learning Algorithms For Beginners: Supervised, Unsupervised Learning and More. **simplilearn**, 2022. Disponível em: <<https://www.simplilearn.com/10-algorithms-machine-learning-engineers-need-to-know-article>>. Acesso em: 14 nov. 2022.

TECH, D. Como funciona o algoritmo Árvore de Decisão. **Didática Tech**, 2019. Disponível em: <<https://didatica.tech/como-funciona-o-algoritmo-arvore-de-decisao>>. Acesso em: 14 nov. 2022.

VASQUES, L. H. Aplicação de técnicas de Machine Learning a diagnósticos na Saúde. **Profissão Biotec**, 2020. Disponível em: <<https://profissaobiotec.com.br/aplicacao-de-tecnicas-de-machine-learning-a-diagnosticos-na-saude>>. Acesso em: 14 nov. 2022.

VAZ, A. L. Gradientes Descendentes na prática — melhor jeito de entender. **medium**, 2020. Disponível em: <<https://medium.com/data-hackers/gradientes-descendentes-na-pr%C3%A1tica-melhor-jeito-de-entender-740ef4ff6c43>>. Acesso em: 14 nov. 2022.

VEEN, F. V. The Neural Network Zoo. **The Asimov Institute**, 2016. Disponível em: <<https://www.asimovinstitute.org/neural-network-zoo/>>. Acesso em: 10 nov. 2022.

WOLFEWICZ, A. Deep Learning vs. Machine Learning – What's The Difference? **Levity**, 2022. Disponivel em: <<https://levity.ai/blog/difference-machine-learning-deep-learning>>. Acesso em: 10 nov. 2022.

YATHISH, V. Loss Functions and Their Use In Neural Networks. **towardsdatascience**, 2022. Disponivel em: <<https://towardsdatascience.com/loss-functions-and-their-use-in-neural-networks-a470e703f1e9>>. Acesso em: 14 nov. 2022.

Han, Xiaohui, Shifeng Ma, Zhewen Shi, Guoqing An, Zhenbin Du, and Chunlin Zhao. "Transformer Fault Diagnosis Technology Based on Maximally Collapsing Metric Learning and Parameter Optimization Kernel Extreme Learning Machine." *IEEJ Transactions on Electrical and Electronic Engineering* 17.5 (2022): 665-73. Web.

"Researchers from Department of Electrical Engineering Discuss Findings in Artificial Intelligence (DCNN-FuzzyWOA: Artificial Intelligence Solution for Automatic Detection of COVID-19 Using X-Ray Images)." (2022): 654. Web.

Banik, Rita, Priyanath Das, Srimanta Ray, and Ankur Biswas. "Prediction of Electrical Energy Consumption Based on Machine Learning Technique." *Electrical Engineering* 103.2 (2021): 909-20. Web.

Meng, Bin, Xinyang Ji, and Gaoshan Hu. "Deduction of Artificial Intelligence Calculation Theory in Electrical Engineering Automation." *Journal of Physics. Conference Series* 1952.3 (2021): 32085. Web.

Ilic, Denis, Mileta Zarkovic, and Zlatan Stojkovic. "Artificial Intelligence System for Stator Condition Diagnostic." *Electrical Engineering* 104.3 (2022): 1503-513. Web.

Lu, Huimin. "Introduction to the Special Section on Artificial Intelligence and Robotics (VSI-air3)." *Computers & Electrical Engineering* 97 (2022): 107586. Web.

Pu, Hai. "Application Electrical Engineering Training and Intelligent Technology of Electrical and Electronic Technology under Artificial Intelligence Technology." *E3S Web of Conferences* 253 (2021): 1070. Web.

"New Research on Artificial Intelligence from Department of Electrical-Electronics Engineering Summarized (COVID-19 Epidemic and Opening of the Schools: Artificial Intelligence-Based Long-Term Adaptive Policy Making to Control the Pandemic ...)." (2021): 574. Web.

Zhang, Xiaotian, Yihua Hu, Jiamei Deng, Hui Xu, and Huiqing Wen. "Feature Engineering and Artificial Intelligence-Supported Approaches Used for Electric Powertrain Fault Diagnosis: A Review." *IEEE Access* 10 (2022): 29069-9088. Web.

Huang, ZhiYing. "Application Research of Electrical Engineering Automation Based on Artificial Intelligence." *Journal of Physics. Conference Series* 1533.3 (2020): 32021. Web.

Yu, Songkun. "Application of Artificial Intelligence in Physical Education." *International Journal of Electrical Engineering & Education* (2021): 2072092199660. Web.

Mukherjee, Debottam, Samrat Chakraborty, and Sandip Ghosh. "Power System State Forecasting Using Machine Learning Techniques." *Electrical Engineering* 104.1 (2022): 283-305. Web.

Su, Haiping, Cheng Lian, Jichuan Liu, and Honglai Liu. "Machine Learning Models for Solvent Effects on Electric Double Layer Capacitance." *Chemical Engineering Science* 202 (2019): 186-93. Web.

Wang, Yufeng, Wanyong Li, Ziqi Zhang, Junye Shi, and Jiangping Chen. "Performance Evaluation and Prediction for Electric Vehicle Heat Pump Using Machine Learning Method." *Applied Thermal Engineering* 159 (2019): 113901. Web.

Wang, Ke, Zicong Chen, Mingjia Zhu, Siu-Ming Yiu, Chien-Ming Chen, Mohammad Mehedi Hassan, Stefano Izzo, and Giancarlo Fortino. "Statistics-Physics-based Interpretation of the Classification Reliability of Convolutional Neural Networks in Industrial Automation Domain." *IEEE Transactions on Industrial Informatics* (2022): 1-8. Web.

Fernandes, Fernando, Rodrigo De Losso Da Silveira Bueno, Pedro Delano Cavalcanti, and Alemayehu Solomon Admasu. "Generating Stochastic Processes Through Convolutional Neural Networks." *Journal of Control, Automation & Electrical Systems* 31.2 (2020): 294-303. Web.

Yu, Wanke, and Chunhui Zhao. "Broad Convolutional Neural Network Based Industrial Process Fault Diagnosis With Incremental Learning Capability." *IEEE Transactions on Industrial Electronics* (1982) 67.6 (2020): 5081-091. Web.

Medeiros, J.P.S., A.C. Da Cunha, A.M. Brito, and P.S.M. Pires. "Automating Security Tests for Industrial Automation Devices Using Neural Networks." *2007 IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007)* (2007): 772-75. Web.

Wang, Tian, Yang Chen, Meina Qiao, and Hichem Snoussi. "A Fast and Robust Convolutional Neural Network-based Defect Detection Model in Product Quality Control." *International Journal of Advanced Manufacturing Technology* 94.9-12 (2018): 3465-471. Web.

Khodayar, Mahdi, Okyay Kaynak, and Mohammad E Khodayar. "Rough Deep Neural Architecture for Short-Term Wind Speed Forecasting." *IEEE Transactions on Industrial Informatics* 13.6 (2017): 2770-779. Web.

Balamurugan, D., S. S. Aravindh, P. Chandra Shaker Reddy, Ajay Rupani, and A. Manikandan. "Multiview Objects Recognition Using Deep Learning-Based Wrap-CNN with Voting Scheme." *Neural Processing Letters* 54.3 (2022): 1495-521. Web.

Mhlanga, David. "Industry 4.0 in Finance: The Impact of Artificial Intelligence (AI) on Digital Financial Inclusion." *International Journal of Financial Studies* 8.3 (2020): 45. Web.

Bécue, Adrien, Isabel Praça, and João Gama. "Artificial Intelligence, Cyber-threats and Industry 4.0: Challenges and Opportunities." *The Artificial Intelligence Review* 54.5 (2021): 3849-886. Web.

Sood, Sandeep Kumar, Keshav Singh Rawat, and Dheeraj Kumar. "A Visual Review of Artificial Intelligence and Industry 4.0 in Healthcare." *Computers & Electrical Engineering* 101 (2022): 107948. Web.

DELLAZARI, Gabriel Senger. Bancada didática de automação industrial: esteira seletora de itens. 2022. Trabalho de conclusão de curso (Bacharelado em Engenharia Elétrica) - Universidade Federal de Santa Maria, [S. l.], 2022.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. arXiv:1409.0575, 2014

Ozkahraman, Merdan, and Haydar Livatyalı. "Artificial Intelligence in Foreign Object Classification in Fenceless Robotic Work Cells Using 2-D Safety Cameras." Tehnički Vjesnik 29.5 (2022): 1491-498. Web.

Santomartino, Samantha M, Nima Hafezi-Nejad, Vishwa S Parekh, and Paul H Yi. "Performance and Usability of Code-Free Deep Learning for Chest Radiograph Classification, Object Detection, and Segmentation." Radiology. Artificial Intelligence 5.2 (2023): E220062. Web.

Mahesh R N, Uma, and Anith Nelleri. "Multi-Class Classification and Multi-Output Regression of Three-Dimensional Objects Using Artificial Intelligence Applied to Digital Holographic Information." Sensors (Basel, Switzerland) 23.3 (2023): 1095. Web.

Fitzsimons, Jack, and Kenneth Dawson-Howe. "Abandoned, Removed and Moved Object Classification." International Journal of Pattern Recognition and Artificial Intelligence 30.1 (2016): 1655002. Web.

Amelio, Alessia, Gianluca Bonifazi, Enrico Corradini, Simone Di Saverio, Michele Marchetti, Domenico Ursino, and Luca Virgili. "Defining a Deep Neural Network Ensemble for Identifying Fabric Colors." Applied Soft Computing 130 (2022): 109687. Web.

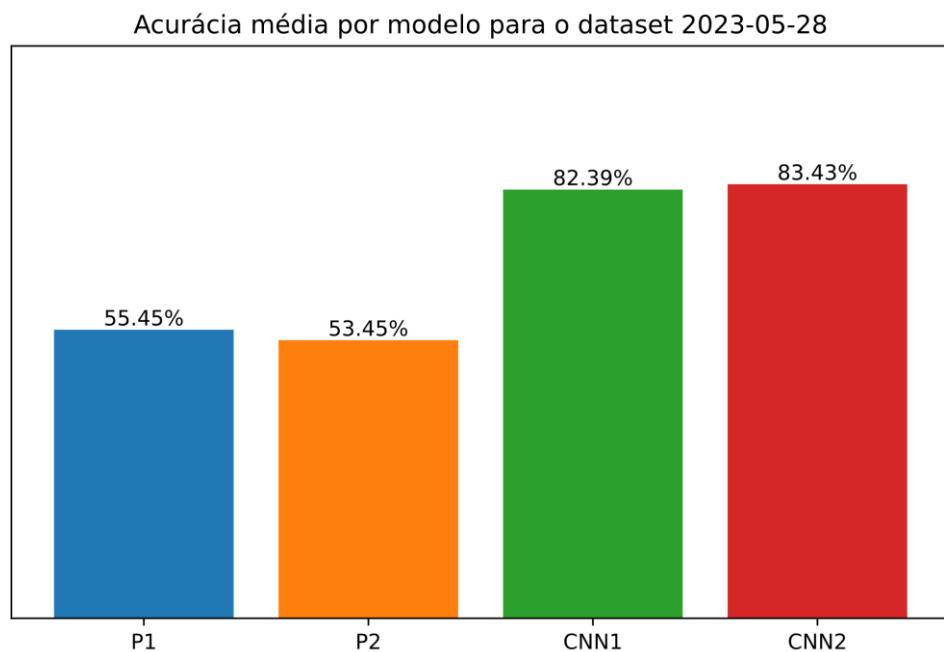
7 APÊNDICES

7.1 UNIÃO ENTRE DANIFICAÇÃO AUTOMÁTICA E MANUAL

Na tentativa de ampliar ainda mais a acurácia e tornar o modelo mais capaz de identificar defeitos reais, foi criado um novo dataset denominado 2023-05-28. Esse dataset contém as mesmas propriedades do dataset 2023-04-24, porém com a substituição de 171 figuras danificadas digitalmente por figuras danificadas manualmente, criando uma variedade maior de dados.

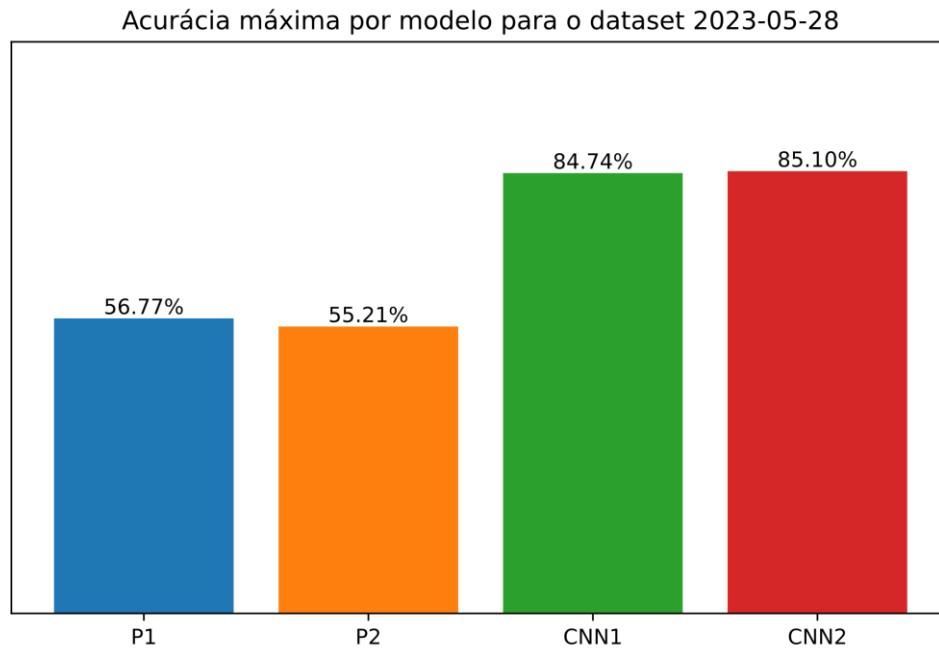
As acuráncias médias e máximas são mostradas através da Figura 7.1 e Figura 7.2, respectivamente.

Figura 7.1 – Acurácia média por modelo para o dataset 2023-05-28



Fonte: Autor.

Figura 7.2 – Acurácia máxima por modelo para o dataset 2023-05-28



Fonte: Autor.

Através da análise da Figura 7.1 e da Figura 7.2, é possível observar uma redução substancial na acurácia em comparação com o conjunto de dados 2023-04-24. Essa diminuição da acurácia fornece evidências que corroboram a existência de diferenças entre os padrões de defeitos gerados de forma digital e aqueles criados manualmente.

Com o objetivo de aprimorar a acurácia, um novo conjunto de dados, denominado 2023-05-29, foi criado. Esse conjunto herda as características do dataset 2023-05-28, porém, com uma quantidade dez vezes maior de dados, totalizando 96000 figuras. Além disso, foram adicionados novos tipos de figuras com defeitos, que envolvem a aplicação de defeitos digitais em figuras previamente danificadas manualmente. Essa abordagem resulta em uma combinação de defeitos digitais e manuais, visando diversificar os padrões de defeitos presentes no conjunto de dados.

Devido à significativa quantidade de dados presentes no novo conjunto de dados, foi necessário reduzir a resolução das figuras de 128x128 para 64x64 a fim de permitir o treinamento dentro das limitações dos recursos computacionais disponíveis. Além disso, em contraste com os testes anteriores que envolviam 10 treinamentos por modelo, apenas um único treinamento por modelo foi realizado, devido ao alto custo computacional associado ao processo de treinamento. Consequentemente, não foram calculadas médias e valores máximos de acurácia, mas apenas um único valor de acurácia para cada modelo. Apesar dessa abordagem, a disponibilidade de uma grande quantidade de dados contribui para uma maior estabilidade das métricas de avaliação entre os treinamentos, reforçando a confiabilidade da acurácia obtida em um único treinamento.

Os valores de acurácia para cada modelo são apresentados através da Tabela 7.1.

Tabela 7.1 – Acurárias do dataset 2023-05-29

P1	76,06%
P2	74,39%

CNN1	89,8%
CNN2	91,58%

Fonte: Autor.

De acordo com os dados apresentados na Tabela 7.1, foi obtida uma acurácia máxima de 91,58%. Observa-se que o melhor modelo alcançou um desempenho próximo ao do melhor modelo do conjunto de dados 2023-04-24, com uma diferença de acurácia de 4,01%. Para recuperar essa diferença e obter uma acurácia ainda maior, algumas medidas poderiam ser consideradas, como a geração de um conjunto de dados com mais de 96000 figuras, restaurar a resolução das imagens para 128x128 e a criação de uma nova arquitetura de rede neural convolucional com mais parâmetros. No entanto, devido às limitações de recursos computacionais, tais medidas não puderam ser implementadas.