

INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA DE MINAS
GERAIS - *CAMPUS* BETIM
BACHARELADO EM ENGENHARIA DE MECÂNICA

Luis Felipe Barbosa Ferreira

Aplicação de rede neural convolucional (CNN's) para verificação de uso de EPI.

Betim

2025

LUIS FELIPE BARBOSA FERREIRA

Aplicação de rede neural convolucional (*CNN's*) para verificação de uso de EPI.

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

Orientador: Prof. Me. Virgil Del Duca Almeida

Betim

2025

FICHA CATALOGRÁFICA

F383a Ferreira, Luis Felipe Barbosa

Aplicação de rede neural convolucional (CNN's) para verificação de uso de EPI. / Luis Felipe Barbosa Ferreira. – 2025

59 f.: il.

Trabalho de conclusão de curso (Bacharelado em Engenharia Mecânica) - Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais, Campus Betim, 2025.

Orientação: Prof. Me. Virgil Del Duca Almeida

1. EPI. 2. Redes neurais (Computação). 3. Visão computacional. 4. Engenharia Mecânica. I. Ferreira, Luis Felipe Barbosa. II. Título.

CDU: 004.8:351



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS
Campus Betim
Diretoria de Ensino
Docentes Automação Industrial e Tecnologia da Informação
Rua Itamarati - CEP 32677-564 - Betim - MG
3135976360 - www.ifmg.edu.br

ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO

Aos 21 dias do mês de fevereiro do ano de 2025, às dezessete horas, nas dependências do IFMG – *Campus* Betim, reuniu-se a banca examinadora presidida por mim, Virgil Del Duca Almeida e demais membros, Ricardo de Lima Silva e Felipe Augusto Rocha da Silva. Nesta ocasião o discente **Luis Felipe Barbosa Ferreira** do curso de Bacharelado em Engenharia Mecânica, com registro acadêmico de número 0034035 do IFMG – *Campus* Betim, defendeu seu Trabalho de Conclusão de Curso (TCC) intitulado “Aplicação de rede neural convolucional (CNN’s) para verificação de uso de EPI” e foi **APROVADO**, com 95 (noventa e cinco) pontos.

Este resultado reflete o cumprimento parcial dos critérios de avaliação estabelecidos pelo curso e reconhece os esforços e a dedicação do discente e seu orientador no desenvolvimento do seu TCC. O lançamento da nota e o conseqüente encerramento do respectivo processo está condicionado ao cumprimento dos procedimentos pós-defesa conforme previstos nos regulamentos vigentes. Tais procedimentos pós-defesa devem ser finalizados dentro do prazo limite de 20 dias, a contar da data desta ata. O descumprimento destes procedimentos até a data estipulada implicará em atribuição de nota 0 (zero) e conseqüente reprovação.

A sessão foi encerrada às dezoito horas e vinte minutos. Para constar, eu, Virgil Del Duca Almeida, redigi a presente ata que após lida publicamente, foi aprovada e assinada pelo discente e membros da banca examinadora.

Betim, 25 de fevereiro de 2025.



Documento assinado eletronicamente por **Virgil Del Duca Almeida, Professor**, em 25/02/2025, às 16:22, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Ricardo de Lima Silva, Professor**, em 25/02/2025, às 19:19, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Felipe Augusto Rocha da Silva, Usuário Externo**, em 07/03/2025, às 16:54, conforme Decreto nº 10.543, de 13 de novembro de 2020.



A autenticidade do documento pode ser conferida no site <https://sei.ifmg.edu.br/consultadocs> informando o código verificador **2214608** e o código CRC **28A01FEB**.

23792.000304/2025-60

2214608v1

DEDICATÓRIA

Gostaria de dedicar esse trabalho primeiramente a minha família. A minha mãe Agueda que esteve ao meu lado e me deu suporte em todos os momentos que precisei ao longo dessa caminhada e de toda a minha vida, me ensinando a ter coragem e enfrentar meus medos. Ao meu pai Leonardo, que trabalhou dia e noite, se dedicava em locais muitas vezes difíceis e perigoso e que se desgastava para que eu tivesse tudo que fosse preciso para a realização do meu sonho.

A minha noiva Jéssica, que desde início esteve ao meu lado e me incentivou a continuar nos momentos difíceis, que me escutava e me apoiava quanto as minhas decisões. A minha irmã Isabela, que muitas das vezes me fazia companhia nos momentos que precisei e escolheu o mesmo caminho que o meu, a engenharia.

AGRADECIMENTOS

Gostaria de agradecer a todos do IFMG Campus Betim, que me deram estrutura e me acolheram ao longo desses anos para que fosse possível realizar meu sonho. Ao meu Orientador Virgil principalmente, que foi paciente e entendeu todo o momento que eu passei durante o desenvolvimento desse trabalho e sempre me auxiliou quando precisei.

Gostaria de agradecer também a todos os meus amigos que conviveram comigo e dividiram essa jornada comigo, muito pela colaboração e por todos os momentos alegres que tivemos juntos. Á todos os professores que lecionaram aulas para mim no Campus Betim que colaboraram para a construção do meu conhecimento.

“A imaginação é mais importante que o conhecimento.”

Albert Einstein

RESUMO

Este trabalho apresenta uma proposta de utilização das Redes Neurais Convolucionais (CNNs) na verificação do uso de Equipamentos de Proteção Individual (EPIs) em ambientes de trabalho. O objetivo principal é criar um sistema de monitoramento visual que consiga verificar o uso do EPI assim sendo uma ferramenta de auxilia o controle, podendo contribuir para a redução de acidentes de trabalho. Para o desenvolvimento deste trabalho foi verificado o uso de capacete e óculos de proteção, utilizando a arquitetura de rede LeNet-5 que alcançou resultado de 96,5% de acurácia na validação e 89% nos testes dos modelos através de variações de hiper parâmetros de treinamento como *Epoch*, *Batch Size*, *Input Shape* e pela quantidade de ciclos de treinamento.

Palavras-chave: Redes Neurais Convolucionais, Visão Computacional, Segurança no Trabalho, EPI, Aprendizado de Máquina.

ABSTRACT

This work presents a proposal for the use of Convolutional Neural Networks (CNNs) to classify the use of Personal Protective Equipment (PPE) in work environments. The main objective is to create a visual monitoring system that can classify the use of PPE, thus being a control tool, contributing to the reduction of work accidents. For the development of this work, the use of a helmet and protective glasses was verified, using the LeNet-5 network architecture, which achieved a result of 96.5% accuracy in validation and 89% in model tests through variations in training hyperparameters such as Epoch, Batch Size, Input Shape and the number of training cycles.

Keywords: Convolutional Neural Networks, Computer Vision, Occupational Safety, PPE, Machine Learning.

LISTA DE ILUSTRAÇÕES

Figura 1 – Comparativo de Custo de Segurança.....	16
Figura 2 – Números de Acidentes no setor de Mineração.....	17
Figura 3 – Pirâmide de Heirich.....	20
Figura 4 – Abordagem de Machine Learning.....	23
Figura 5 – Funcionamento de Redes Neurais Feed-Forward.....	25
Figura 6 – Funcionamento de CNN para classificação de imagens.....	27
Figura 7 – Operação de Convolução.....	27
Figura 8 – Funções de Ativação.....	29
Figura 9 – Classes com EPI.....	31
Figura 10 – Classes Sem EPI.....	31
Figura 11 – Exemplo de aplicação do Data Augmentation.....	34
Figura 12 – Método aplicado ao Data Augmentation.....	34
Figura 13 – Imagem com Input Shape diferentes.....	37
Figura 14 – Resultados da Validação das Arquiteturas de Redes.....	39
Figura 15 – Plote predições e imagens LeNet.....	40
Figura 16 – Plote predições e imagens AlexNet.....	40
Figura 17 – Resultados da Teste das Arquiteturas de Redes.....	41
Figura 18 – Média de Validação e teste variando Epochs.....	42
Figura 19 – Média de Validação e teste variando Epochs.....	42
Figura 20 – Média de Validação e teste variando os Ciclos.....	43
Figura 21 – Média de Validação e teste variando Batch Size.....	44
Figura 22 – Média de Validação e teste variando Input Shape.....	46
Figura 23 – Tempo de Treinamento variando Input Shape.....	46

LISTA DE TABELAS

Tabela 1 – Strings para pesquisa de imagens.....	32
Tabela 2 – Hiper parâmetros utilizados para comparar Arquiteturas.....	38
Tabela 3 – Hiper parâmetros utilizados para determinar os Epochs.....	41
Tabela 4 – Hiper parâmetros utilizados para determinar os Batch Size...	44
Tabela 5 – Testes realizados com LeNet.....	47
Tabela 6 – Diferença entre Validação e Teste.....	48

LISTA DE EQUAÇÕES

(1) $f(x) = \max(0, x)$ – Função ReLU.

(2) $f(x) = \frac{1}{1+e^{-x}}$ – Função Sigmoid.

LISTA DE ABREVIATURAS E SIGLAS

EPI – Equipamento de Proteção Individual

NRs – Normas Regulamentadoras

CLT – Consolidação de Leis de Trabalho

CNN – *Convolutional Neural Networks*

API - *Application Programming Interface*

GPU – *Graphics Processing Unit*

CPU – *Central Processing Unit*

TPU - *Tensor Processing Unit*

ABNT - Associação Brasileira de Normas Técnicas

IFMG - Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais

NBR - Normas Técnicas Brasileiras

TCC - Trabalho de Conclusão de Curso

SUMÁRIO

1	INTRODUÇÃO.....	16
1.1	Justificativa.....	19
1.2	Objetivos.....	20
1.3	Organização do Trabalho.....	21
2	FUNDAMENTAÇÃO TEÓRICA.....	22
2.1	Inteligência Artificial e Visão Computacional.....	22
2.2	Recursos.....	29
3	METODOLOGIA.....	30
3.1	Método de Aquisição de Imagens.....	30
3.2	Desenvolvimento do Modelo.....	33
4	RESULTADOS E DISCUSSÃO.....	38
5	CONCLUSÃO.....	49
	REFERÊNCIAS.....	51
	ANEXO A.....	55
	ANEXO B.....	59

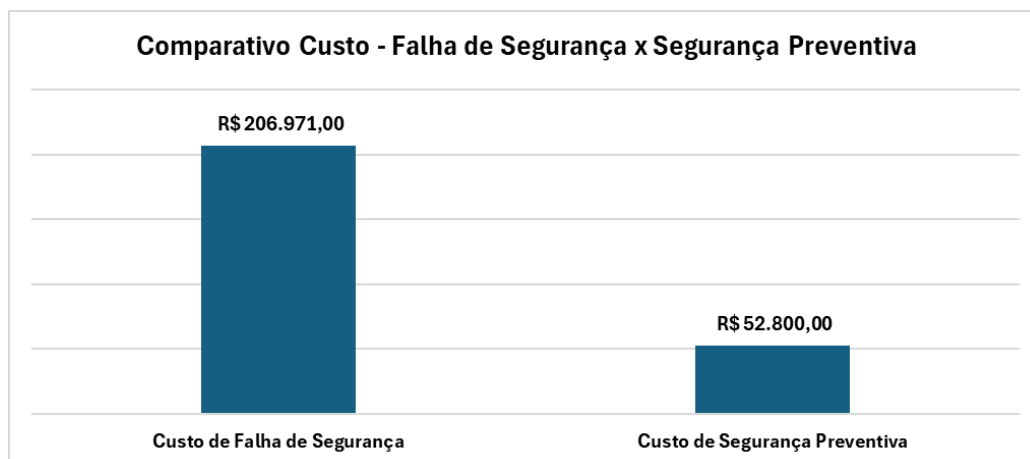
1 INTRODUÇÃO

A segurança é um elemento essencial dentro de ambiente industrial. Além de fatores produtivos, financeiros, qualitativos, ela se faz necessária para a sustentabilidade de uma indústria. Segundo Sousa e Rodolpho (2020), a segurança e saúde do trabalho garantem um ambiente organizacional mais estruturado, promovendo o aumento da produtividade tornando o local de trabalho mais seguro e agradável, melhorando a qualidade de vida do trabalhador, além reduzir transtornos administrativos e penalidades impostas pelos órgãos fiscalizadores sobre a empresa em caso de acidentes de trabalho.

O ramo industrial apresenta variados tipos de riscos, cada um com diferentes exposições para o trabalhador, desde contato com produtos químicos até movimentação de grandes cargas, e negligenciar tais riscos pode trazer consequências, acidentes, doenças ocupacionais e, em alguns casos, fatalidades. Quando ocorre um acidente com um colaborador, se perde uma mão de obra, assim, consequentemente, se perde produtividade e reputação da empresa, o que afeta a relação entre fornecedor e cliente, além dos custos com o tratamento e medidas médicas.

O custo com implantação de ferramentas e políticas voltadas para segurança é consideravelmente menor que quando ocorre uma falha na segurança do trabalho, conforme Figura 1.

Figura 1 – Comparativo de Custo de Segurança.

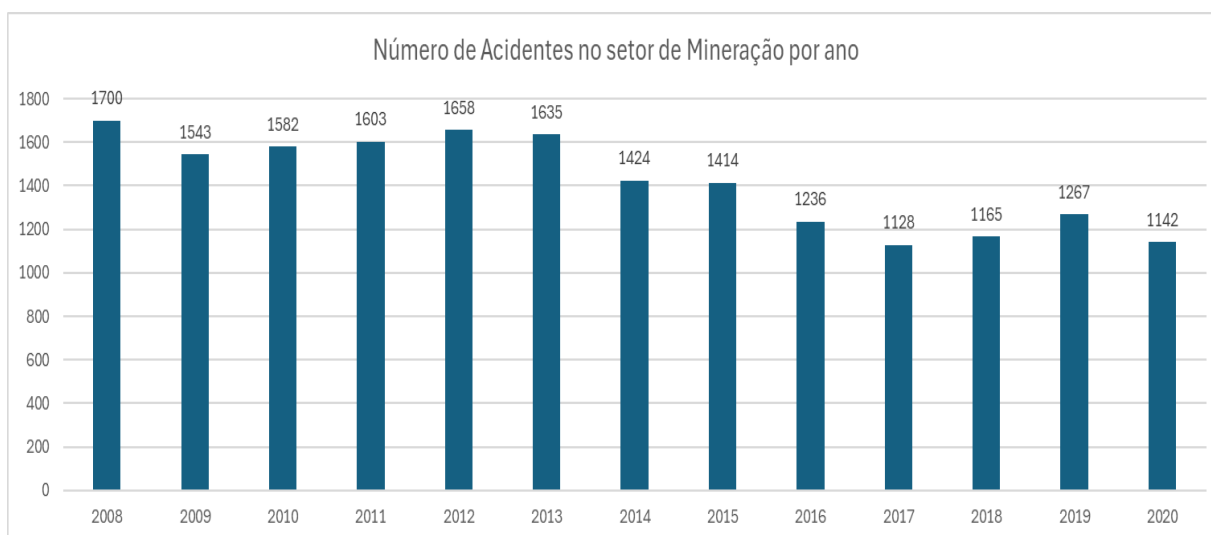


Fonte: Adaptado de: Gonçalves, Santos; 2023.

Segundo Sousa e Rodolpho (2020), é fundamental discutir a importância da segurança do trabalho nas empresas, pois ela desempenha um papel essencial na proteção dos trabalhadores e na prevenção de acidentes. No dia a dia, muitos profissionais não têm consciência ou desconhecem os riscos a que estão expostos ao exercerem suas atividades.

Então, promover um ambiente seguro, se passa pela construção de uma cultura focada em segurança, onde avaliar potenciais riscos e mitigá-los para promover um ambiente de trabalho saudável, esta ação se inicia desde o treinamento, conscientização até implementação de ferramentas com intuito de evitar acidentes.

Figura 2 – Números de Acidentes no setor de Mineração.



Fonte: Adaptado de: Vaz et al. (2023).

Segundo Vaz et al. (2023), entre 2012 e 2020 houve uma redução no quadro de acidentes no ramo da mineração, conforme demonstrado na Figura 2, isso ocorre devido ao acirramento de leis relacionadas à segurança do trabalho, implementação de novas tecnologias para controle do ambiente, novos procedimentos de segurança e condicionamentos dos ambientes de mineração, assim como uso de equipamentos automatizados, dispositivos de controle e monitoramentos. Tais dados demonstram que implementações de medidas que favorecem a promoção de segurança em um ambiente de trabalho trazem resultados satisfatórios.

No Brasil, a segurança do trabalho é orientada pelas Normas Regulamentadoras, também conhecidas como NRs, que definem os requisitos de segurança mínimos para cada tipo de atividade do mercado de trabalho. Segundo NR

1, as Normas Regulamentadoras são de execução obrigatória pelas empresas privadas, públicas e qualquer órgão público que possuam empregados contratados e regidos pela CLT (Consolidação de Leis de Trabalho), o que torna o cumprimento das NRs obrigatórias (BRASIL, 2024).

Dentro das normas de segurança, uma delas está explicitamente ligada à proteção individual: a NR 6, que trata do uso de Equipamentos de Proteção Individual (EPIs). Essa norma estabelece quando e sob quais condições esses equipamentos devem ser utilizados pelos trabalhadores (BRASIL, 2025). Segundo Camisassa (2015), EPI é todo produto ou dispositivo destinado a proteger o trabalhador individualmente, com o objetivo de reduzir riscos ou amenizar possíveis acidentes.

O EPI definitivamente não evita acidentes, mas oferece maior proteção contra riscos oriundos das atividades realizados pelos trabalhadores, como por exemplo, capacete para proteger a cabeça de lesões ou óculos de proteção para evitar projeção de partículas nos olhos. Administrar o uso do EPI, muitas vezes depende mais da conscientização do trabalhador do que da empresa, por mais que existam políticas de educação voltadas para segurança, ainda sim há a possibilidade do erro humano através de um ato ou condição insegura adotada.

De acordo com Silva, Figueiredo e Dutra (2023), para melhorar a prevenção nos ambientes de trabalho, tem-se adotado cada vez mais ferramentas tecnológicas, especialmente com o avanço da tecnologia. Esse movimento tem permitido a automação de processos industriais, incorporando tecnologias como inteligência artificial e robótica, com o objetivo de automatizar tarefas e tornar os processos mais eficientes. Diante dessa situação, a utilização de recursos tecnológicos pode ser uma alternativa para contornar as dificuldades existentes, e o uso de visão computacional para validar o uso do EPI surge como uma solução promissora para garantir a segurança no ambiente de trabalho.

A visão computacional, segundo Khan, Laghari e Awan (2021), pode desempenhar uma função com grandes capacidades na extração de informações de imagens, contribuindo para vários âmbitos como, sistemas de vigilância, reconhecimento óptico de caracteres, robótica, detecção anomalias e entre outras diversas aplicações. A tecnologia permite desde processo de monitoramento de circunstâncias até classificação de imagens e vídeos, o que lhe permite otimizar e prever condições e economizar tempo na correção das não conformidades.

Considerando a possibilidade de um erro humano ocasionar um ato inseguro e a viabilidade da utilização de novas tecnologias, a visão computacional se torna uma ótima alternativa, pois um sistema equipado com câmeras e algoritmos de reconhecimento de imagens que podem monitorar, detectar e classificar se os trabalhadores estão utilizando ou não os EPI, como por exemplo uso de capacete e óculos de proteção. A identificação automática com uso de visão computacional pode trazer diversas oportunidades desde mitigar o risco do processo ao verificar a ausência o EPI, até geração de dados que podem ser usados para criar de relatórios que podem servir para idealização e implementação de medidas corretivas e treinamento específicos contribuindo com construção de uma cultura focada em segurança no trabalho.

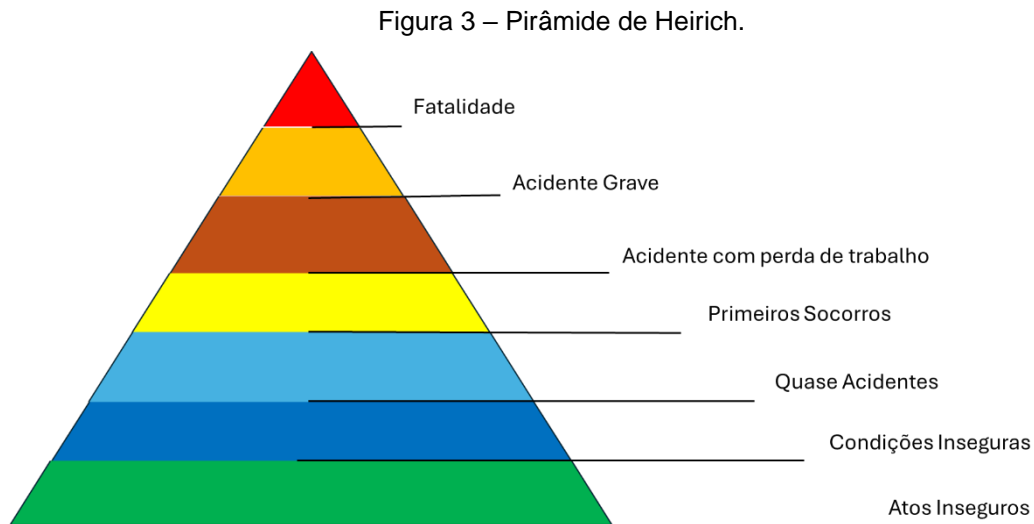
1.1 Justificativa

A dificuldade no controle do uso do EPI tem origem em um erro humano, seja tanto do responsável pelo uso do equipamento quanto pelo responsável pelo treinamento dos colaboradores, também conhecido por atos ou condições inseguras. Segundo Silva e Gonçalves (2022), atos ou condições inseguras ocorrem quando um trabalhador, mesmo com treinamento e com capacidade técnica referente aos processos de segurança, opta pelo não cumprimento de procedimentos e regras, realizando a atividade ciente do risco referente ao trabalho. O ato inseguro pode ser causado por diversos fatores como falta de formação adequada, manutenção não executada corretamente, uso incorreto do EPI, mal-estar e desatenção.

Atos ou condições inseguras são causa raiz para que acidentes de trabalho ocorram, pois aumentam o risco. Um conceito que corrobora com a linha de raciocínio é o da Pirâmide de Heinrich, estudo realizado pelo engenheiro de segurança Herbert William Heinrich na década de 30. Segundo Silva e Gonçalves (2022), Heinrich realizou um estudo sobre custos que estavam relacionados a acidentes, criando uma proporção direta entre número de acidentes, tipos e custo inerentes para o empregador, e estabeleceu parâmetros dos acidentes a alguns pontos, falha humana, perfil de acidentes, lesões e atos ou condições inseguras.

Na pirâmide de Heinrich, cada grau se relaciona a um tipo acidentes, ou quase acidente, de acordo com a consequência. A pirâmide começa com atos inseguros seguido por condição insegura, quase acidentes, primeiros socorros,

acidente com perda de tempo de trabalho, acidentes graves e no topo da pirâmide fatalidades. Se eliminarmos ou reduzirmos os atos e condições inseguras, consequentemente, seguindo a linha de Heinrich reduziremos o acidente e consequentemente os custos, conforme Figura 3.



Fonte: Adaptado de: Silva e Gonçalves; 2022.

1.2 Objetivos

O objetivo do projeto de pesquisa é adaptar o uso da visão computacional para classificar o uso ou não do EPI, especificamente o uso de óculos de proteção e capacete, com o fim de controle e evitar atos ou condições inseguras decorrentes da falta desses equipamentos. Os objetivos específicos do projeto são:

- Identificar a arquitetura de rede mais eficiente para obter a maior acurácia nos testes realizados;
- Determinar quais hiper parâmetros são mais adequados aos dados de treinamento;
- Encontrar a melhor acurácia possível dentro do contexto construído no trabalho.

1.3 Organização do Trabalho

A organização do trabalho está estruturada de forma a apresentar de maneira clara e lógica os processos e resultados da pesquisa. O primeiro capítulo, Fundamentação Teórica, aborda os conceitos essenciais relacionados à segurança do trabalho, visão computacional e redes neurais, com um enfoque especial na aplicação dessas tecnologias ao uso de Equipamentos de Proteção Individual (EPIs). Dentro deste capítulo, a seção 2 detalha os conceitos principais da inteligência artificial e da visão computacional, enquanto a subseção 2.1 expande o foco para as redes neurais e suas aplicações.

No capítulo seguinte, Seleção de Bibliotecas e Recursos, será discutida a escolha das ferramentas e bibliotecas necessárias para a implementação do modelo de visão computacional. Essa decisão é crucial para garantir a eficiência do processo e a adequação do modelo aos dados disponíveis. A seção 2.2 dedica-se a essa escolha, explorando as diferentes opções e suas aplicações.

Em seguida, o capítulo Metodologia inicia com a seção Método de Aquisição de Imagens, explicando como os dados de treinamento serão coletados, processados e preparados para alimentar o modelo. Já a seção 3.2, intitulada Desenvolvimento do Modelo, descreve a construção e o treinamento do modelo de *Machine Learning*, com foco no processo de implementação e nos desafios enfrentados.

O capítulo Resultados e Discussão é o próximo, com a seção 4 apresentando a análise inicial do desempenho da rede neural, onde se avalia sua acurácia em relação aos dados de validação e teste. A seguir, a realização de experimentos com variações nos hiper parâmetros, como *Epoch*, *Input Shape*, *Batch Size* e Ciclos, será discutida, buscando melhorar o desempenho do modelo. A seção também abordará a interpretação dos resultados, identificando os melhores cenários para o modelo.

Finalmente, o trabalho será encerrado com o capítulo Conclusão, onde serão sintetizadas as descobertas e feitas as considerações finais sobre a pesquisa, refletindo sobre seu impacto no contexto industrial.

2 FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica a seguir abordará os conceitos centrais de *Machine Learning* e suas subdivisões, com ênfase nas metodologias de aprendizado supervisionado, não supervisionado e por reforço. Além disso, será discutido o papel crucial do *Deep Learning*, particularmente as redes neurais convolucionais (CNNs), no processamento de imagens e em aplicações de visão computacional, apresentando os princípios de funcionamento dessas redes, suas camadas e as operações fundamentais como a convolução.

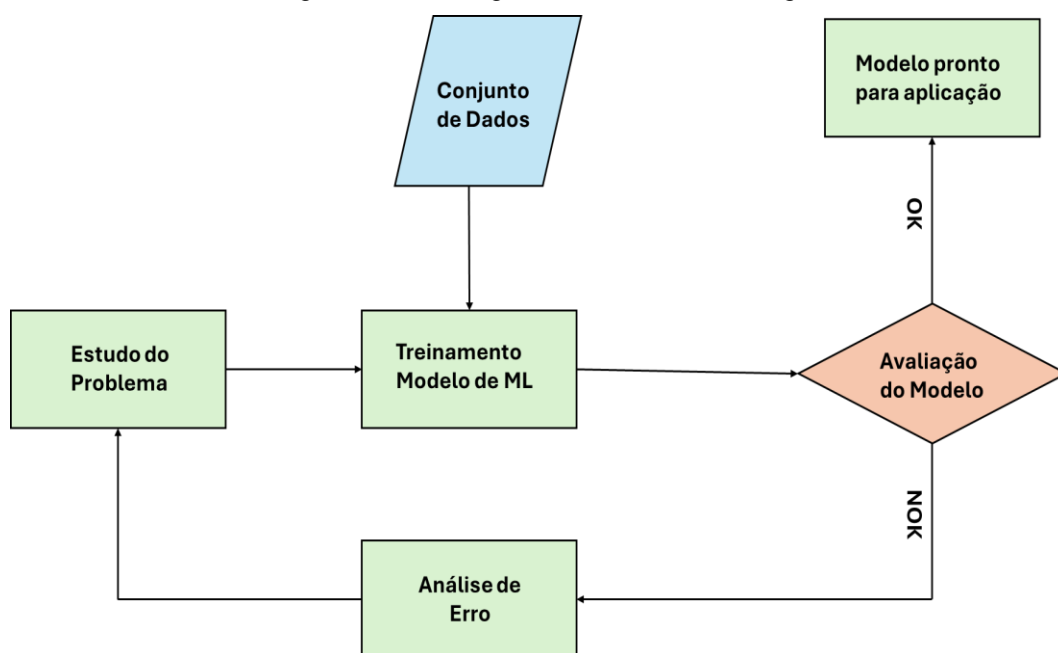
2.1 Inteligência Artificial e Visão Computacional

Dentro da área de inteligência artificial há diversos campos, mas um em específico tem destaque particular, o *Machine Learning* (ou Aprendizado de Máquina). Esta área se concentra no desenvolvimento de técnicas e algoritmos que permitem os computadores aprenderem determinada tarefa ou atividade através de dados (Grus, 2021). *Machine Learning* pode ser descrito de acordo com Tom Mitchell (1997):

Um programa de computador aprende com a experiência E em relação a alguma tarefa T e alguma medida de desempenho P, se seu desempenho em T, medido por P, melhora com a experiência E. Tom Mitchell (1997)

Segundo Géron (2021), o aprendizado de máquina pode ser útil para a resolução alguns de problemas que nós não conseguimos resolver de maneira tradicional ou são de grande complexidade, como identificar anomalias em conjunto de dados ou prever valor de uma variável ao longo do tempo. Essas aplicações vão desde a área de saúde (BEZERRA et al., 2024) até aplicações de gestão de produção industriais (HOMMA JUNIOR et al., 2021).

Para a executar um processo de *machine learning* é necessário realizar uma sequência de métodos para construir, avaliar e validar um modelo, fluxo pode ser visualizado na Figura 4.

Figura 4 – Abordagem de *Machine Learning*.

Fonte: Adaptado de: Géron, 2021.

O *Machine Learning* é subdividido em alguns métodos de aprendizagem, cada um de acordo com a condição da análise ou aspecto dos dados usados para o processo de treinamento da máquina, mas os que se destacam são:

- A aprendizagem por reforço é um sistema um pouco diferente dos dois outros pois podem assistir, selecionar e executar ações em um ambiente e obter *feedback* através de recompensas ou penalidades. O sistema deve aprender sozinho qual melhor estratégia para resolver, denominado de política, para obter a maior quantidade de recompensas, ela define qual a ação que o sistema deve escolher de acordo com a situação em que se encontra (Géron, 2021);
- Na aprendizagem supervisionada é fornecido nos dados de treinamentos algoritmos dos quais se encontram as soluções com características desejadas no processo de aprendizagem, também denominadas de *labels* (ou rótulos). Algumas tarefas comuns no desenvolvimento da aprendizagem supervisionada são a classificação e a regressão. A classificação consiste em atribuir dados a diferentes classes ou categorias com base em seus atributos e nos rótulos pré-estabelecidos no algoritmo.

Já a regressão é utilizada para prever valores numéricos contínuos a partir do conjunto de dados de entrada (Géron ,2021);

- Na aprendizagem não supervisionada, diferente da supervisionada, os dados utilizados para desenvolver o algoritmo de aprendizagem não são rotulados, assim o sistema tenta aprender sem orientação, tais características têm como objetivo entender os padrões presentes nos conjuntos de dados de entrada, segundo Géron (2021). Tarefas típicas da aprendizagem não supervisionada são:
 - Clusterização: Detecta e separa os dados em grupos seguindo suas características;
 - Detecção de Anomalias ou novidades: Identifica características incomuns no conjunto de dados;
 - Visualização e redução de dimensionalidades: Facilitam a visualização de dados complexos e detectam dados que apresentam correlações;
 - Aprendizagem de regras por associação: Investigam grandes quantidades de dados e identificam relações entre os atributos.

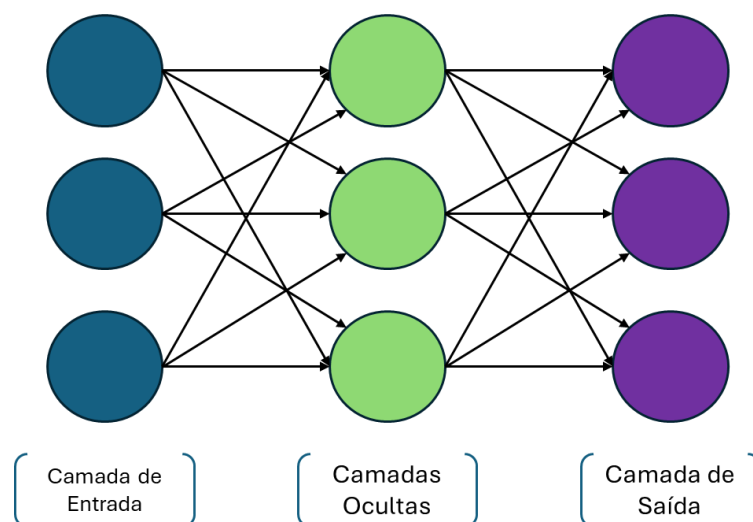
Dentro do campo de *Machine Learning*, existe um subcampo que tem potencial para resoluções de problemas de grande complexidade, o do *Deep Learning* ou aprendizagem profunda. A aprendizagem profunda está fortemente ligada ao uso de múltiplas camadas de neurônios artificiais, simulando o funcionamento dos neurônios do sistema nervoso humano (VAKALOPOULOU et al., 2023).

Segundo Grus (2021), as redes neurais são modelos preditivos que usam a dinâmica do cérebro e uma série de neurônios conectados, onde cada um realiza uma análise dos dados produzidos pelos outros neurônios e dispara uma nova saída de acordo com os critérios determinado. A topologia do cérebro humano é complexa, assim, simulação que mais se aproxima da nossa rede neural é a *feed-forward*. Nessa estrutura, as camadas de neurônios são conectadas de forma discreta, enviando os dados para a camada seguinte.

A camada de entrada recebe e transmite os dados sem modificações. As camadas ocultas recebem dados da camada anterior, processam as informações e as

enviam para a próxima camada. Por fim, a camada de saída gera as saídas do modelo, conforme ilustrado no esquema da Figura 5.

Figura 5 – Funcionamento de Redes Neurais *Feed-Forward*.



Fonte: Elaborado pelo Autor (2025).

As redes neurais são muito utilizadas para resolução de diversos tipos de problemas, como reconhecimento de caracteres, caligrafia e detecção de cenários. Tais aplicações as tornam muito úteis para o uso aliado a outro campo da inteligência artificial, denominado visão computacional, aliado a um tipo rede neural, as convolucionais.

Convolutional Neural Networks (Redes Neurais Convolucionais em português), ou CNN, são um tipo de redes neurais especiais do tipo *feed forward* muito utilizadas para processamento de imagens, exemplificado na Figura 5, principalmente em classificação de imagens (FAN, MA e ZHONG, 2021). Em vista do que foi mencionado por Krichen (2023) as CNNs têm obtido resultados no reconhecimento e processamento de imagens, com ótimos *benchmarks*, o seu sucesso se dá pela capacidade capturar características e padrões espaciais das imagens usando uma hierarquia específica de camadas que executam algumas operações de convolução e extraem informações em diferentes níveis de abstração.

As CNNs têm como estrutura de funcionamento alguns tipos de camadas convolucionais, são elas:

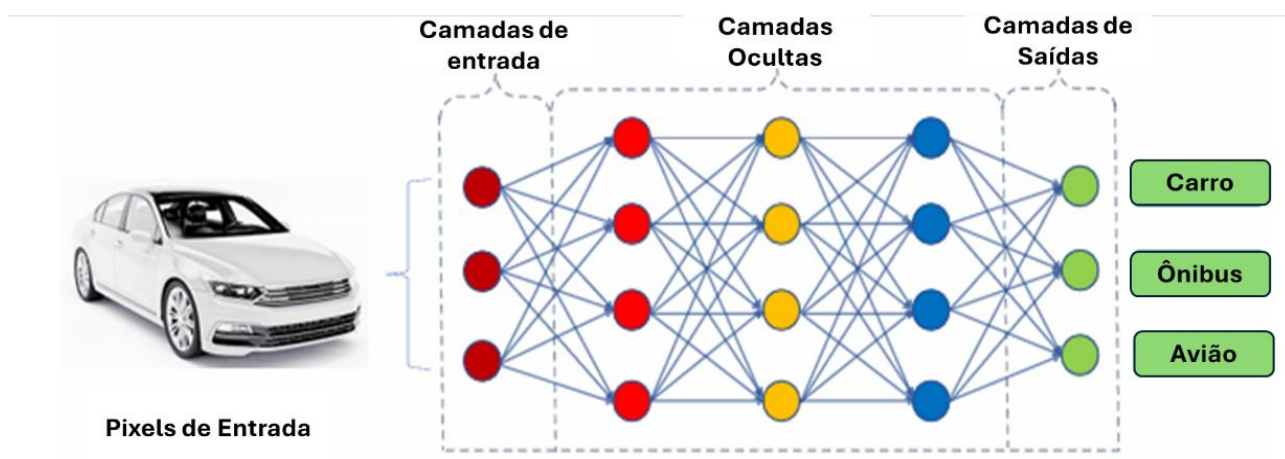
- Camadas de Convolucionais: Talvez o tipo mais importante de camada, ela executa uma operação de convolução na imagem de entrada com um conjunto de filtro para produzir um mapeamento de características (Grus, 2021);
- Camadas de *Pooling*: Este tipo de camada é responsável por trabalhar a redução da dimensão espacial dos mapas de características produzidos pela camada de convolução, pegando assim os valores máximos ou médios das regiões não sobrepostas (Grus, 2021);
- Camadas de Ativação: A ativação é basicamente uma ativação não linear através de funções específicas que ativam na camada anterior uma saída não linear, introduzindo a não linearidade na rede neural, permitindo a aprendizagem com recursos mais complexos (Grus, 2021);

Existem outros métodos amplamente empregados nas Redes Neurais Convolucionais (CNNs) que são essenciais para o processo de construção e aprimoramento da rede, como o *Batch Normalization*, o *Dropout* e *Fully Connected*. O *Batch Normalization* é uma técnica que visa otimizar a estabilidade e a eficiência do treinamento, normalizando a saída de camadas anteriores. Esse método ajusta os dados ao subtrair a média e dividir pelo desvio padrão, o que contribui para a redução da variação da covariância interna, promovendo uma convergência mais rápida e eficaz do modelo.

O *Dropout*, por sua vez, é uma técnica de regularização utilizada para mitigar o *overfitting* durante o treinamento. Essa abordagem elimina aleatoriamente uma porção dos neurônios da camada anterior, forçando a rede a aprender representações mais gerais e robustas, prevenindo que o modelo se ajuste excessivamente aos dados de treinamento.

Por fim, as *Fully Connected* desempenham um papel crucial ao conectar cada neurônio de uma camada à próxima, geralmente nas etapas finais da rede. Essas conexões são responsáveis por transformar as informações processadas ao longo das camadas anteriores, gerando a saída final do modelo, essencial para a realização das tarefas de classificação ou previsão.

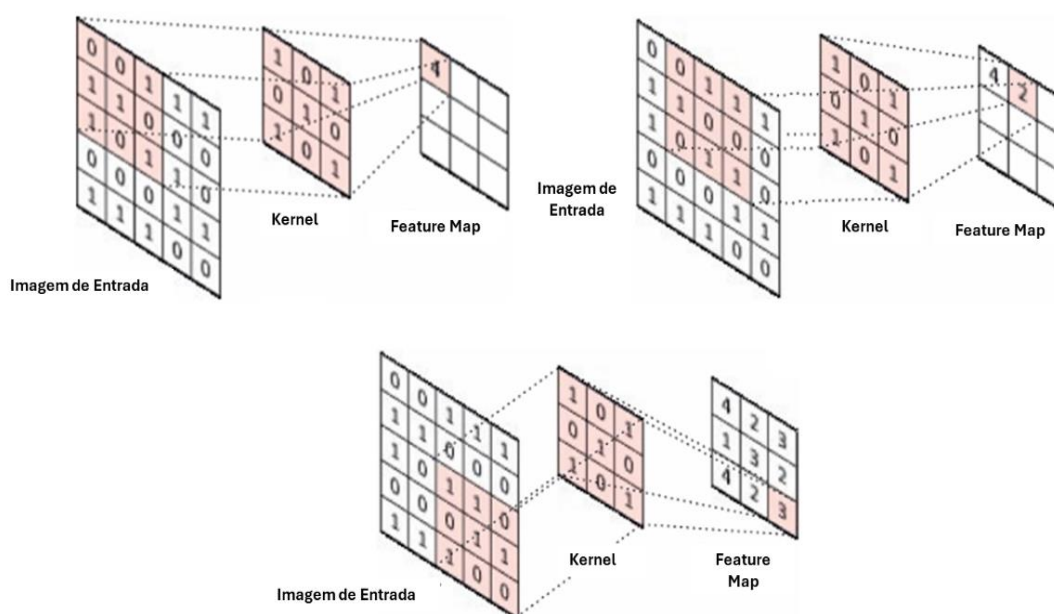
Figura 6 – Funcionamento de CNN para classificação de imagens.



Fonte: Adaptado de: Krichen; 2023.

A operação de Convolução, de acordo com Krichen (2023), é uma operação matemática que combina dois sinais ou funções com intuito de produzir um terceiro sinal que correlaciona a influência de um sinal sobre o outro. A operação de convolução é de extrema importância para as CNNs, pois é a partir dela é possível criar um mapeamento de característica da imagem, deslizando filtros ou *Kernels*, para calcular o produto escalar de cada posição da imagem de entrada, conforme Figura 7.

Figura 7 – Operação de Convolução.



Fonte: Adaptado de: Abirami et al. (2021).

Dois elementos fundamentais são o tamanho do filtro pois determina o tamanho do campo receptivo, basicamente a região de entrada afetando assim a posição de saída do filtro, e o passo que é a distância entre um filtro e os filtros adjacentes.

Como demonstrado por Taye (2023), os *Kernels* são um conjunto de valores discretos ou inteiros, e em cada número de *Kernels* há um peso como referência, que se modificam ao longo do processo de aprendizagem, iniciando seu valor de maneira aleatória, eles aprendem a extrair os recursos que são mais significativos de acordo com ajuste dos seus pesos. Os *Kernels* são capazes de executar operações em espaços de grandes dimensões sem a necessidade de calcular as coordenadas dos dados, pois calculam o produto interno da imagem dos pares de dados, sendo capaz de transformar um modelo linear em um não linear.

Existem mais dois tópicos importantes para entender o funcionamento de uma CNN, os *features maps* (ou mapeamento de atributos), e as funções de ativações. Conforme descrito no estudo de Krichen (2023), os *feature maps* são componentes importantes na saída de uma camada convolucional, simplificada é uma matriz bidimensional que traduz o grau de cada região da imagem de entrada correspondente aos filtros e kernels aplicados nela, onde cada elemento está relacionado a uma camada de neurônio e qual aspecto específico que ela deve capturar na imagem. Em geral, os *feature maps* são ferramentas poderosas para capturar as particularidades importantes das imagens, permitindo uma grande aplicação de tarefas de reconhecimento visual

Já as funções de ativação são importantes para o sucesso das CNNs, pois são responsáveis pela não-linearidade nos modelos, porque muito dos fenômenos que conhecemos exibem comportamentos complexos e não lineares. Assim se utilizar operações lineares não seria possível realizar diversas tarefas dadas a incapacidade de resoluções de problemas complexos. Conseqüentemente as funções de ativação são responsáveis por aumentar a expressão das CNNs quanto eficiência para solução de problemas.

As funções de ativação comumente usadas são a ReLU (unidade linear retificada), Sigmoid e a Tahn (Tangente hiperbólica). A ReLU é definida pela Equação (1):

$$f(x) = \max(0, x) \quad (1)$$

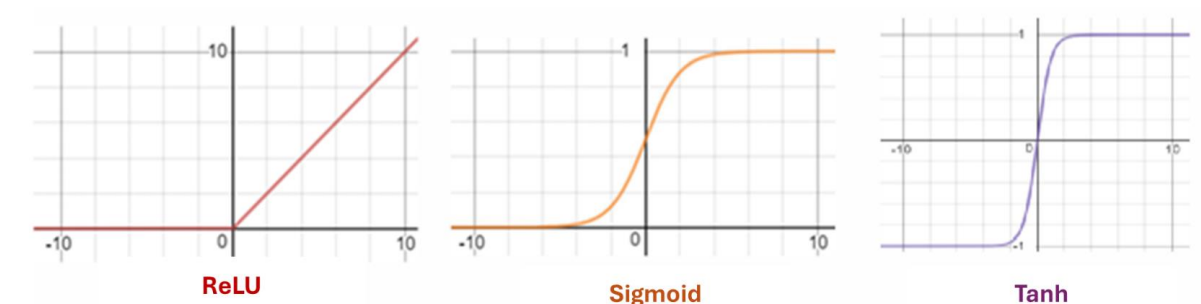
Onde x é definido pela entrada para o neurônio. Nesta função retorna sempre entre 0 e x , desconsiderando valores negativos e tornando os valores positivos inalteráveis. A ReLU ajuda a prevenir a saturação dos neurônios ao longo do treinamento do modelo.

A função Sigmoide, apresentada pela Equação (2), é definida por:

$$f(x) = \frac{1}{1+e^{-x}} \quad (2)$$

Onde x é definido pela entrada para o neurônio. A função Sigmoid tem um formato em S bem característico e mapeia todo número real, para valores entre 0 e 1, sendo uma boa alternativa em problemas de classificação de duas classes. A função Tanh é quase semelhante a Sigmoid, inclusive nas dificuldades de uso, diferenciando apenas no mapeamento dos números reais, enquanto a primeira está entre 0 e 1, a Tanh está entre -1 a 1, conforme Figura 8.

Figura 8 – Funções de Ativação.



Fonte: Adaptado de: Krichen; 2023.

2.2 Recursos

Para o desenvolvimento do projeto foi necessário o uso de uma linguagem de programação, mais especificamente *Python* em sua versão 3.10.12. As bibliotecas selecionadas para o desenvolvimento do projeto foram *Numpy* para cálculos e operações numéricas, *Pandas* para manipulação, análise estatística e limpeza de dados, *Matplotlib* para plote 2D de gráficos com resultados para análise, *TensorFlow* uma biblioteca específica para aprendizagem de máquina, e *Keras* uma API

(*Application Programming Interface*, ou Interface de Programação de Aplicação) de alto nível para redes neurais que roda sobre o *TensorFlow*, além das bibliotecas *OS* e *Time*.

O ambiente de desenvolvimento do projeto foi feito na plataforma do *Google Colaboratory*, ou *Colab*, da empresa Norte Americana *Google*, plataforma web para com *Python* como linguagem nativa e utilizando recursos em nuvem para rodar os algoritmos em sua versão gratuita, com especificações de *hardware* do Google Colaboratory com Intel(R) Xeon(R) CPU @ 2.20GHz e 13.61GB de Memória RAM. Outro elemento é o *Google Drive* para alocar as imagens para os dados de treinamento do modelo.

3 METODOLOGIA

A metodologia deste trabalho envolve a aquisição, processamento e treinamento de dados para desenvolver um modelo de *Machine Learning* capaz de identificar o uso de Equipamentos de Proteção Individual (EPIs). As imagens foram coletadas de fontes diversas, incluindo produção própria, pesquisa em bancos de dados como *Google* Imagens e *Kaggle*, e foram organizadas em duas classes: com e sem uso de EPI.

Para melhorar a variedade e robustez dos dados, foi aplicado o *Data Augmentation*, ajustando características como brilho e saturação das imagens para simular condições de iluminação industrial. O modelo foi desenvolvido com as arquiteturas LeNet-5 e AlexNet, sendo treinado com a definição dos hiper parâmetros *Batch Size*, *Input Shape* e *Epoch*, visando otimizar a performance e evitar problemas como *Overfitting*.

3.1 Método de Aquisição de Imagens

A etapa de aquisição das imagens é determinante para o desenvolvimento do projeto, pois é a partir do conjunto de imagens selecionadas para treinar o modelo que se pode obter um bom resultado. As imagens utilizadas para o processo de treinamento foram separadas em duas categorias para o desenvolvimento do

algoritmo de classificação, uma classe com uso de EPI e outra sem EPI, conforme Figura 9 e Figura 10, respectivamente. Para a classe de uso com EPI, foram selecionados dois equipamentos, os óculos de proteção e o capacete, porque ambos são itens possíveis de verificar visualmente a sua utilização.

Figura 9 – Classes com EPI.



Fonte: Elaborado pelo Autor (2025).

Figura 10 – Classes com EPI.



Fonte: Sbaghidi (2023).

A aquisição de imagens para a construção dos dados para treinar o modelo foi feita em três instâncias: imagens de produção própria, imagens de pesquisa através do *Google* Imagens com *strings* de pesquisas específicas, conforme Tabela 1, e um conjunto de dados da plataforma *Kaggle* para a classe com EPI, já para a classe sem EPI foi utilizado os Datasets de rostos humanos da plataforma *Kaggle* aliado com imagens de produção própria.

As imagens de produção própria foram realizadas com uso de um capacete, óculos de proteção em diferentes ângulos e de 8 diferentes rostos entre homens e mulheres (Conforme a Figura 9), enquanto as imagens de pesquisa seguem o critério de pesquisa de acordo com a Tabela 1, selecionando sempre as imagens que apresentassem indivíduos com os equipamentos definidos para o desenvolvimento do modelo.

Tabela 1 – *Strings* para pesquisa de imagens.

Idioma	Palavra
Português	EPI, Equipamento de Proteção Individual, Pessoa de EPI, Trabalhador de EPI, Uso de EPI, Aplicação de EPI, Capacete, óculos de proteção
Inglês	<i>PPE, Personal Protective Equipment, PPE Person, PPE Worker, PPE Use, PPE Application, Helmet, Safety Glasses</i>

Fonte: Elaborado pelo Autor (2025).

O objetivo de ter dois padrões para a classe com EPI, é dar maior variedade de características para que o modelo tenha maior capacidade de classificação. Para a classe sem EPI foi utilizado um critério para separação de imagens: variedade de rostos entre gênero, cor de pele, idade, qualidade e tamanho da imagem. Foram separadas as imagens em três conjuntos de dados, treinamento, teste e validação. Sendo 200 imagens para o treinamento e 40 para etapas de teste e validação, totalizando 240 imagens, divididos nas duas categorias, com e sem EPI.

Para o desenvolvimento de um modelo a etapa de validação serve para entender e ajustar as *features* e os hiper parâmetros durante o treinamento, enquanto o teste tem o intuito de promover a acurácia do modelo com amostras diferentes das que foram vistas durante o treinamento, como consequência aferindo a qualidade dele.

3.2 Desenvolvimento do Modelo

O modelo foi construído pensando em um contexto de aplicação industrial de forma que consiga identificar o uso ou não do EPI, categorizando o mesmo em duas classes. O objetivo é que o recurso desenvolvido identifique o ato inseguro com o não uso dos equipamentos de proteção, antes que seja possível iniciar ou retomar sua atividade.

Como exemplo de aplicação do modelo, ele pode ser utilizado como um dispositivo que esteja antes da entrada de uma determinada área em uma planta industrial, na qual o uso de EPI seja obrigatório para o exercício da atividade. Será avaliado com uma câmera se o trabalhador está com os equipamentos definidos corretamente, assim permitindo ou bloqueando a entrada no ambiente, como ferramenta de controle para evitar que o colaborador se coloque em uma condição insegura.

Para esse contexto de implementação, se entende que o ambiente industrial segue os padrões de iluminação determinados da NR17 (BRASIL,2022), avaliando as condições do ambiente de trabalho, onde se o tema da Ergonomia. Logo o mecanismo criado no exemplo com procedimento de verificação deve ser treinado com imagens com bom brilho e nitidez para garantir melhor assertividade do modelo e similaridade com o contexto apresentado. Para que a base de imagens utilizadas para o treinamento tenha tais características, será necessário utilizar uma técnica que aumenta a quantidade das imagens e executa um tratamento quanto as características das imagens, denominada *Data Augmentation*.

Segundo Maharana et al. (2022) o processo de *Data Augmentation* ou *Image Augmentation*, é um método de pré-processamento de imagens que tem o intuito de aumentar o volume de aprendizado durante o *Machine Learning*. Tal processo tem como objetivo garantir melhor performance do modelo. Esta técnica garante uma maior variedade de *features* para a base de dados utilizada para o treinamento, ampliando assim a capacidade de reconhecimento pois consegue replicar a mesma imagem com diferentes características como rotação, inversão, corte, redimensionar imagens além de características da imagem como saturação, brilho e contraste, como demonstrado na Figura 11.

Figura 11 – Exemplo de aplicação do *Data Augmentation*.



Fonte: Adaptado de: Alomar et al. (2023).

O processo de *Data Augmentation* realizado no desenvolvimento do modelo desse trabalho teve como objetivo aproximar ao ambiente industrial, local onde a iluminação e a nitidez do local de trabalho são prioridade conforme a NR17. Para que fosse possível melhorar a nitidez das imagens foi aumentado a saturação das imagens em 30%, incluído também ao base de treinamento uma característica de imagem que se aproximaria ao de um ambiente industrial, conforme a Figura 12.

Figura 12 – Método aplicado ao *Data Augmentation*.



Fonte: Elaborado pelo Autor (2025).

Após o *Data Augmentation* a base de treinamento foi de 200 para 400 imagens, pois cada imagem base teve uma imagem gerada equivalente, mas com aumento de saturação, aumentando assim a quantidade de todo o conjunto de treinamento.

Uma característica importante de um modelo de rede neural convolucional é a arquitetura de rede, onde ela organiza qual tipo e ordem de cada camada. De maneira geral as arquiteturas das CNN's são constituídas por uma camada de entrada, seguida por uma ou mais sequências que alternam camadas convolucionais e de *pooling*. Após essa sequência, encontram-se camadas densamente conectadas, responsáveis pela classificação ou regressão final (Souza,2013).

Dentro das diversas variedades de tipos de arquiteturas duas se destacam para aplicação nos processamentos de imagens a LeNet-5 e a AlexNet. A LeNet consiste em uma arquitetura desenvolvida por Yann LeCun em 1998, é uma das arquiteturas de redes neurais convolucionais (CNNs) mais conhecidas. Projetada inicialmente para o reconhecimento de dígitos manuscritos no conjunto de dados MNIST, a LeNet-5 apresenta uma estrutura composta por camadas convolucionais e de *pooling*, seguidas por camadas totalmente conectadas.

A arquitetura utiliza técnicas como *padding* para preservar informações nas bordas da imagem e *pooling* com coeficientes e vieses aprendíveis, além de conexões esparsas entre algumas camadas. Embora originalmente empregasse uma função de saída baseada em distância euclidiana, a LeNet-5 demonstrou a viabilidade das CNNs para tarefas de visão computacional, influenciando significativamente o desenvolvimento de modelos mais complexos subsequentes (Géron, 2021).

A AlexNet foi desenvolvida por Alex Krizhevsky, Ilya Sutskever e Geoffrey Hinton, alcançou um marco significativo ao vencer o desafio ImageNet ILSVRC 2012 com uma margem substancial. Essa arquitetura CNN se destaca por empregar múltiplas camadas convolucionais consecutivas, em contraste com a prática anterior de intercalar camadas convolucionais e de *pooling*. Além disso, AlexNet incorporou técnicas inovadoras como o uso da função de ativação ReLU, dropout e aumento de dados, contribuindo significativamente para o avanço do aprendizado profundo na área de visão computacional (Géron, 2021).

A LeNet possui uma estrutura sequencial composta por camadas convolucionais seguidas de *pooling* médio para extração de características, seguidas por camadas densas para classificação. A rede inclui duas camadas convolucionais (6 e 16 filtros), duas camadas de *pooling* médio, uma camada de achatamento e três camadas densas, finalizando com ativação sigmoide para classificação binária.

A AlexNet é uma rede mais profunda, utilizando múltiplas camadas convolucionais intercaladas com *pooling* máximo para captura de padrões mais

complexos. Sua estrutura inclui cinco camadas convolucionais (96, 256, 384, 384 e 256 filtros), três camadas de *pooling* máximo, uma camada de achatamento e três camadas densas, onde as duas primeiras usam ReLU e a última sigmoide para saída binária.

Para o desenvolvimento do modelo existem alguns hiper parâmetros que são importantes para o treinamento e definem o quão capaz o modelo será de realizar a atividade determinada e podem ser determinantes para a sua performance, levando em conta o tempo para treinamento e acurácia. Para o modelo em questão foram considerados 3 elementos relacionados a informações das imagens, Input Shape, e ao processo de treinamento, *Epoch* e *Batch Size*, e é realizado o treinamento do modelo em alguns ciclos e considerada a média dos resultados para entender como o modelo se comportará de maneira global e não local quanto aos dados de treinamento e os parâmetros utilizados.

O *Batch Size*, ou tamanho do lote, é um hiper parâmetro que corresponde ao número de amostras processadas rede neural ao longo do processo, sendo assim crucial. Lotes menores, tipicamente inferiores a 32, aceleram o treinamento, pois cada iteração envolve menos cálculos, entretanto pode ser menos precisa com lotes pequenos. Lotes maiores, por sua vez, oferecem uma assertividade mais robusta, mas podem tornar o treinamento mais lento.

A escolha do *Batch Size* deve equilibrar eficiência computacional e desempenho do modelo, muitas vezes ajustada com base na arquitetura da rede e na capacidade de hardware disponível, no entanto, para aproveitar as otimizações de hardware e software, especialmente para operações matriciais, é recomendado um tamanho de lote superior a 10 (Géron, 2021).

O *Input Shape*, ou formato de entrada, é as informações referente a entrada dos dados que a CNN recebe em sua primeira camada com formato que inclui altura e largura da imagem e canais da imagem, colorida ou preto e branco. Estrutura que corresponde a quantidade de pixels que uma imagem tem, conforme Figura 13.

Figura 13 – Imagem com *Input Shape* diferentes.



Fonte: Elaborado pelo Autor (2025).

Escolher um formato de entrada apropriado é crucial para o desempenho da rede, pois tamanhos muito pequenos podem levar à perda de detalhes importantes, mas menor tempo de processamento, enquanto tamanhos muito grandes se tem maior riqueza de detalhes, mas aumenta significativamente o custo computacional e maior tempo de treinamento. Além disso, o *Input Shape* deve ser compatível com a arquitetura da rede, garantindo que as operações de convolução e *pooling* funcionem corretamente ao longo das camadas.

O conceito de *Epoch*, ou épocas, é fundamental no treinamento de redes neurais, cada *Epoch* representa uma passagem completa do modelo por todo o conjunto de dados de treinamento. O número de épocas influencia diretamente o desempenho da rede. Segundo Géron (2021), número excessivo de épocas pode levar ao *overfitting*, se ajusta excessivamente ao conjunto de dados, enquanto um número insuficiente pode resultar em *underfitting*, quando o modelo não consegue captar as características dos dados dada a complexidade dele.

A escolha do número ideal de *Epochs* requer um equilíbrio cuidadoso e pode ser auxiliada por técnicas como validação cruzada e monitoramento de métricas durante o treinamento (Géron, 2021).

4 RESULTADOS E DISCUSSÃO

Para análise das métricas de sucesso dos modelos é importante avaliar dois pontos: o tempo de treinamento e a acurácia na validação e teste. O tempo de treinamento indica a capacidade de processamento de acordo com as características de cada arquitetura e valores de hiper parâmetros, que no contexto industrial é de grande importância já que quanto mais rápido o uso do EPI foi identificado, melhor para o contexto que se insere.

A acurácia é outro elemento a ser avaliado, pois ele dirá sobre a capacidade de classificar o uso ou não do equipamento de proteção, cumprindo o objetivo principal do trabalho. Para uma conclusão coerente, torna-se de suma importância avaliar ambos os elementos apresentados. Com este critério o modelo desenvolvido no presente trabalho, torna possível uma ferramenta de identificação de EPI. Tal modelo deve ser factível e não deve obstruir o processo industrial com a alta demanda de tempo, deve ser eficaz e atingir os resultados indicados.

O primeiro resultado que foi avaliado foi um comparativo entre os tipos de arquiteturas a LeNet e a AlexNet. Ambos possuem os mesmos hiper parâmetros e base de dados para treinamento, divergindo apenas as características das próprias arquiteturas de redes, conforme Tabela 2. Nos dois testes e validações foram realizados dez ciclos avaliando a média de acurácia e os valores encontrados ao longo do processo de treinamento.

Tabela 2 – Hiper parâmetros utilizados para comparar Arquiteturas.

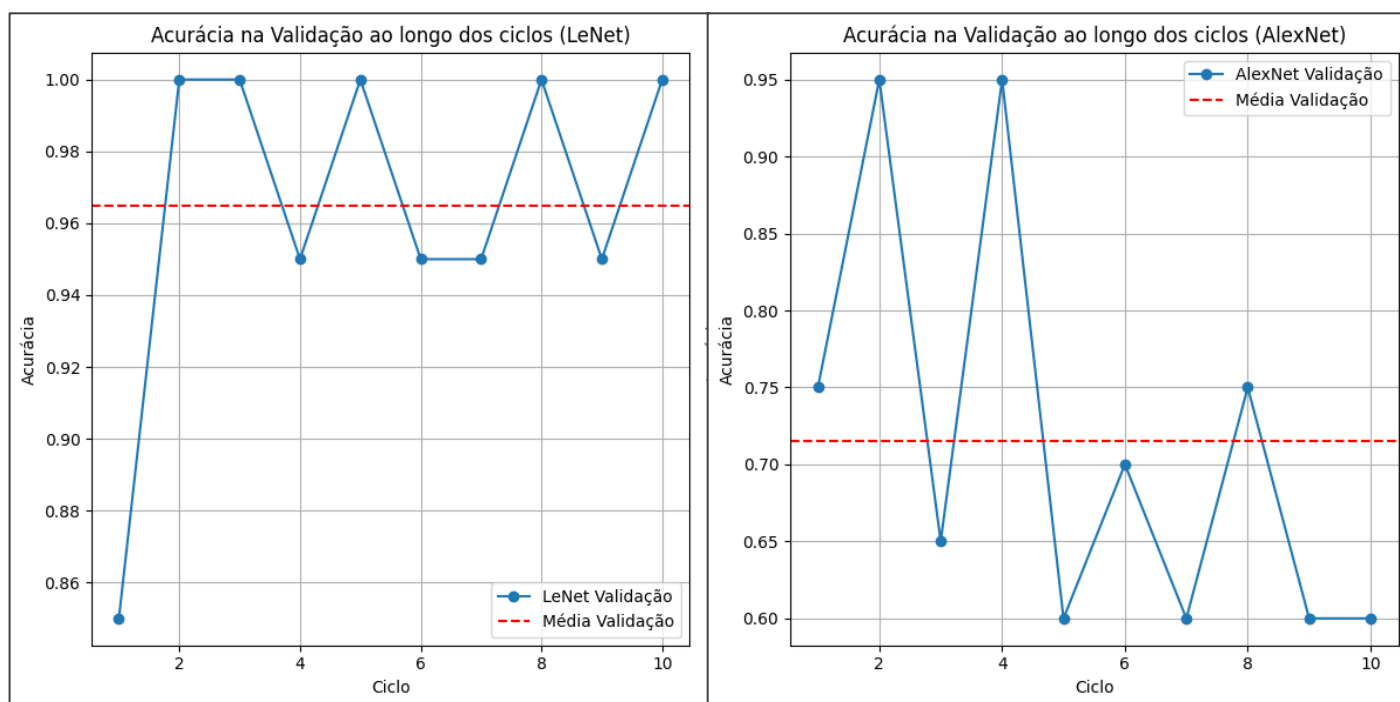
Hiper Parâmetros	Valores
<i>Input Shape</i>	150x150
<i>Batch Size</i>	32
<i>Epoch</i>	10

Fonte: Elaborado pelo Autor (2025).

Durante o primeiro teste realizado, a LeNet se comportou melhor em relação as características apresentadas em todos os aspectos em comparação a AlexNet, tanto na validação do modelo, quanto nos testes. Durante a validação o

comportamento da LeNet foi mais estável e apresentou média de 96,50% enquanto a AlexNet foi consideravelmente inferior, de 71,50%, de acordo com os gráficos da Figura 14.

Figura 14 – Resultados da Validação das Arquiteturas de Redes.



Fonte: Elaborado pelo Autor (2025).

Analisarmos o contexto das imagens e as predições, sugere qual o comportamento dos modelos em relação a base de dados. Como na metodologia de aquisição de imagens foi mesclada entre produção própria, das quais apresentam características com composição de fundo claro e EPI's semelhantes, e as demais apresentam uma variedade de equipamentos e fundo em vários cenários dando aos dados de treinamento muitas possibilidades para o desenvolvimento dos modelos.

Nos ciclos no qual a LeNet consegue classificar todos os exemplos corretamente percebe-se que ele se comporta bem em relação a diversos cenários com diferentes características, como visto na Figura 15 e quando analisamos a AlexNet percebemos maior dificuldade para lidar com cenários com características diversas, conforme Figura 16.

Figura 15 – Plote predições e imagens LeNet.



Fonte: Adaptado de: Trung Le, 2023.

Figura 16 – Plote predições e imagens AlexNet.

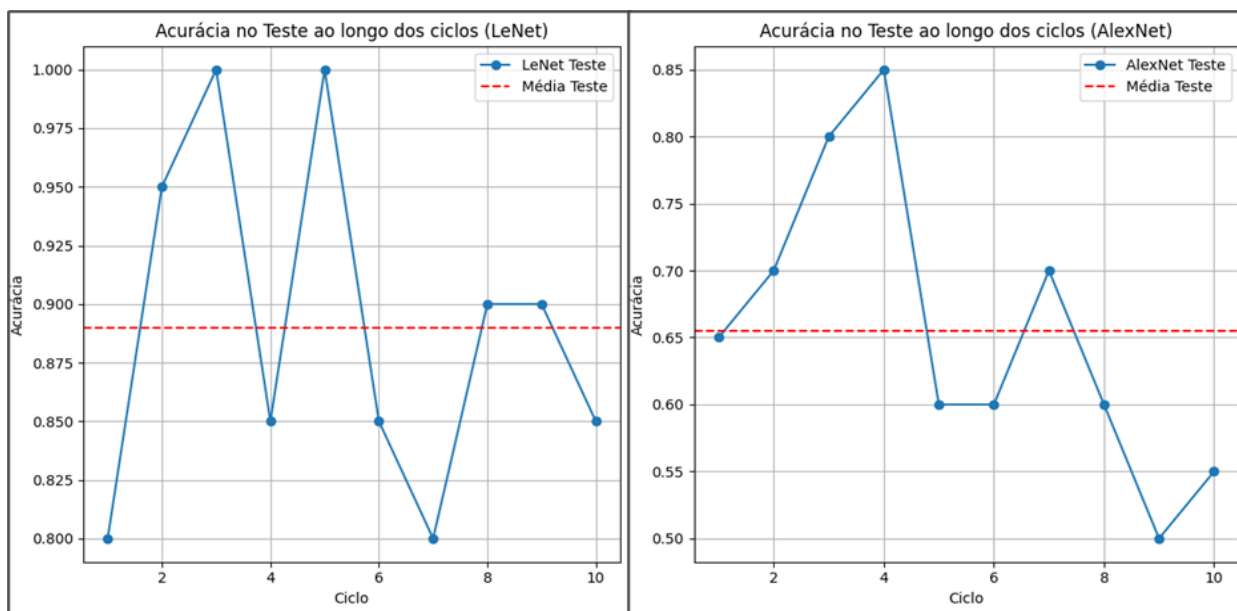


Fonte: Trung Le, 2023 (Adaptado).

Os resultados da etapa de testes corroboram ainda mais para concluirmos a melhor precisão da LeNet. Ao longo dos ciclos, percebe que a acurácia da LeNet tem seu pior resultado com o valor entre 80% e o valor de maior relevância nos testes da AlexNet é de 85%, sendo assim uma diferença pouco relevante entre o seu melhor

resultado e o pior de seu concorrente. A média para LeNet após o treinamento é de 89%, enquanto do seu concorrente é de 65,50%, conforme a Figura 17.

Figura 17 – Resultados da Teste das Arquiteturas de Redes.



Fonte: Elaborado pelo Autor (2025).

Após compreender que a LeNet é a arquitetura que se apresentou melhor quanto ao primeiro teste, neste momento do desenvolvimento do modelo foi feita a avaliação de qual quantidade de *Epochs* seria mais adequada para o atingir maior precisão. Para essa segunda etapa dos testes, foi mantido os valores dos outros hiper parâmetros e comparados com o resultado variando apenas os valores de *Epochs*, conforme Tabela 3.

Tabela 3 – Hiper parâmetros utilizados para determinar os *Epochs*.

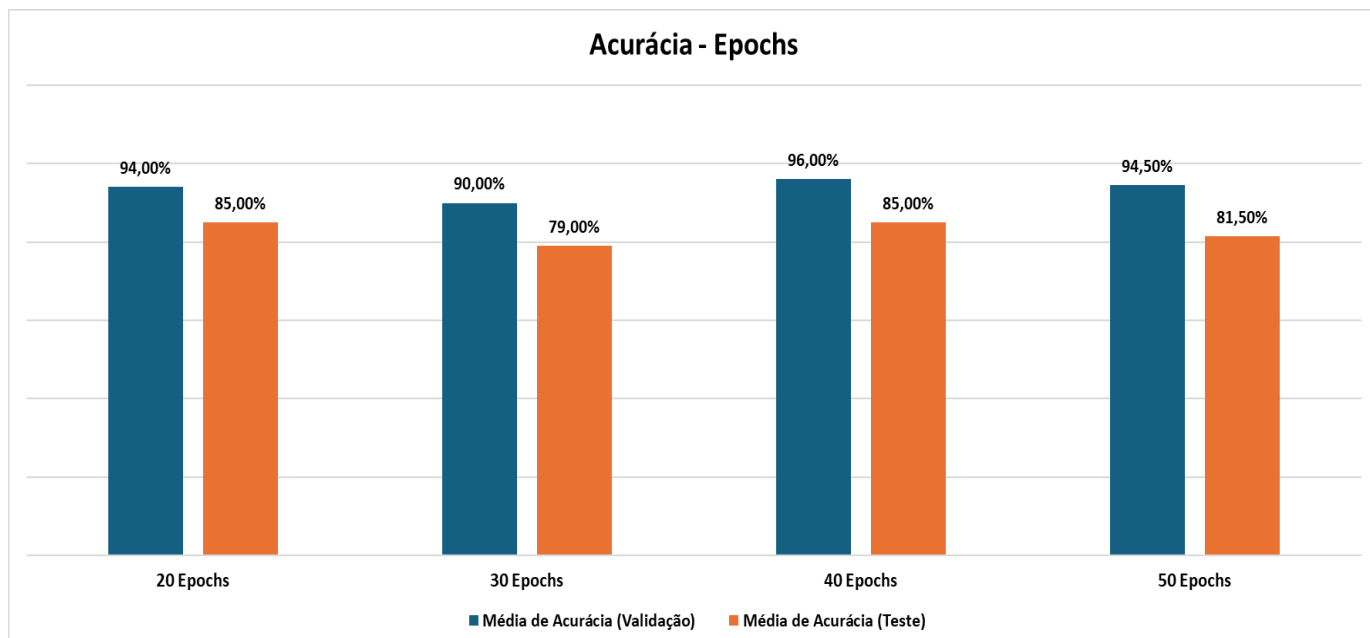
Hiper Parâmetros	Valores
<i>Input Shape</i>	150x150
<i>Batch Size</i>	32
<i>Epoch</i>	20,30,40 e 50

Fonte: Elaborado pelo Autor (2025).

Os resultados de acurácia para validação e teste encontrados foram próximos dos vistos no primeiro teste, entretanto não superaram o valor de 96,5% de

validação e 89% de teste, conforme Figura 18. O valor que mais se aproximou foi de 40 Epochs, de 96% na validação e 85% no teste.

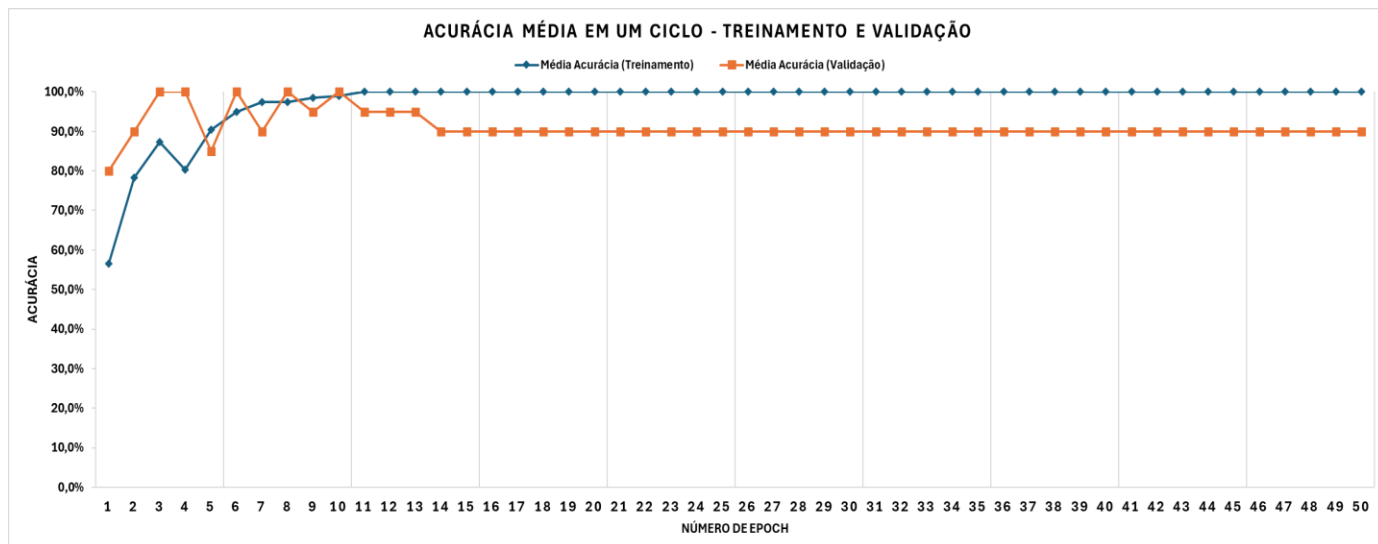
Figura 18 – Média de Validação e teste variando *Epochs*.



Fonte: Elaborado pelo Autor (2025).

No treinamento pôde ser percebido ao longo do ciclo que após 10 épocas, o modelo se ajusta os valores de treinamento e validação, o que indica que apresenta um bom desempenho, mas que a partir desse momento não apresenta grandes progressões, como pode ser visto na Figura 19, que a partir do *Epoch* 10 o valor de acurácia de treinamento atinge 100% e os valores de validação se mantém entre uma dentro 95% e 100%. Diante disso não se faz necessário aumentar os valores de *Epoch* em relação primeiro teste para o desenvolvimento do modelo, já que os valores de validação e teste não são maiores que os encontrados anteriormente e os valores durante o treinamento não progridem com seu aumento.

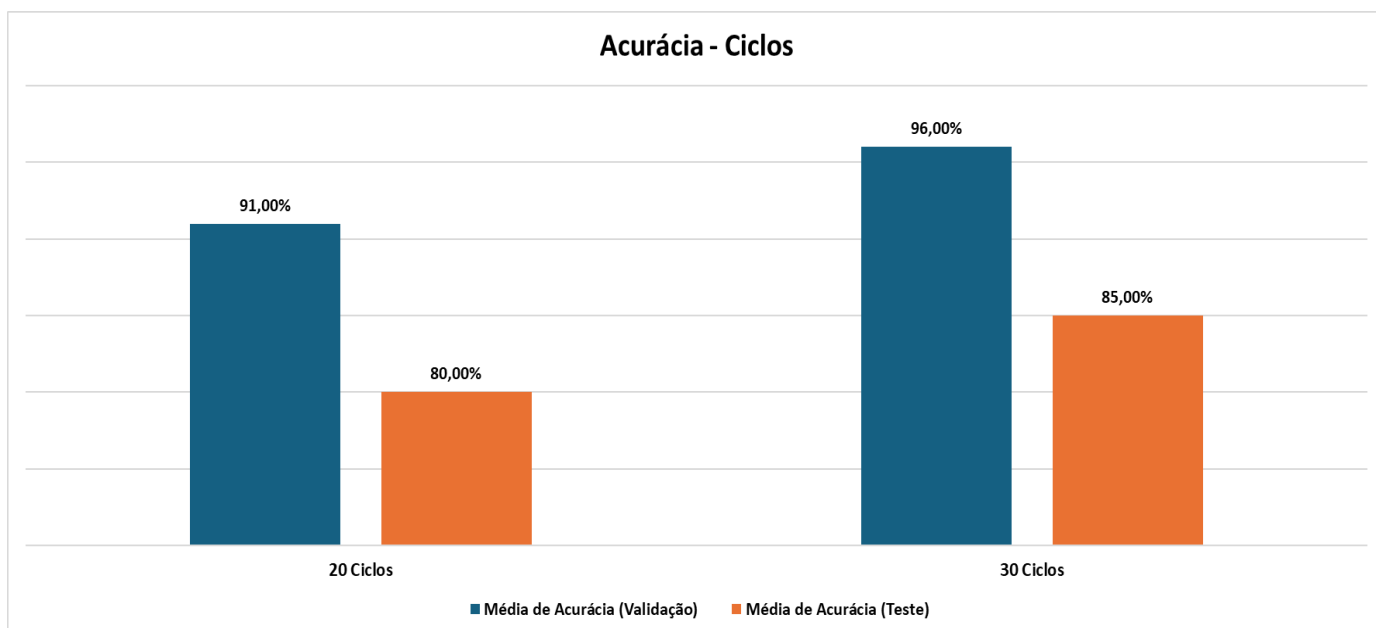
Figura 19 – Média de Validação e teste variando *Epochs*.



Fonte: Elaborado pelo Autor (2025).

Os ciclos são um conjunto de vários *Epochs* onde o valor encontrado em um ciclo é a média dos mesmos e variar tais valores pode demonstrar como seria a média de comportamento do modelo. Para este momento foi avaliado se dermos mais ciclos as acurácias irão melhorar, já que as métricas vêm a partir das médias no decorrer do treinamento. Para etapa foi mantido os parâmetros vistos na Tabela 3, mas diferente dos outros dois testes foram feitos especificamente 20 e 30 ciclos.

Figura 20 – Média de Validação e teste variando os Ciclos.



Fonte: Elaborado pelo Autor (2025).

Os valores encontrados para a acurácia de validação e teste não foram maiores do que os vistos no primeiro teste, que pode ser visto na Figura 20, então os ciclos não foi um fator determinante para a progressão do modelo.

O *Batch Size* foi outro hiper parâmetro que foi avaliado em mais 4 valores, além do já usado nos outros testes. O intuito é entender se aumentar ou diminuir a quantidade de amostras que o modelo irá processar influenciará na precisão do modelo. Assim como os testes anteriores os hiper parâmetros foram mantidos e foi variado, apenas os valores de *Batch Size*, conforme Tabela 4.

Tabela 4 – Hiper parâmetros utilizados para determinar os *Batch Size*.

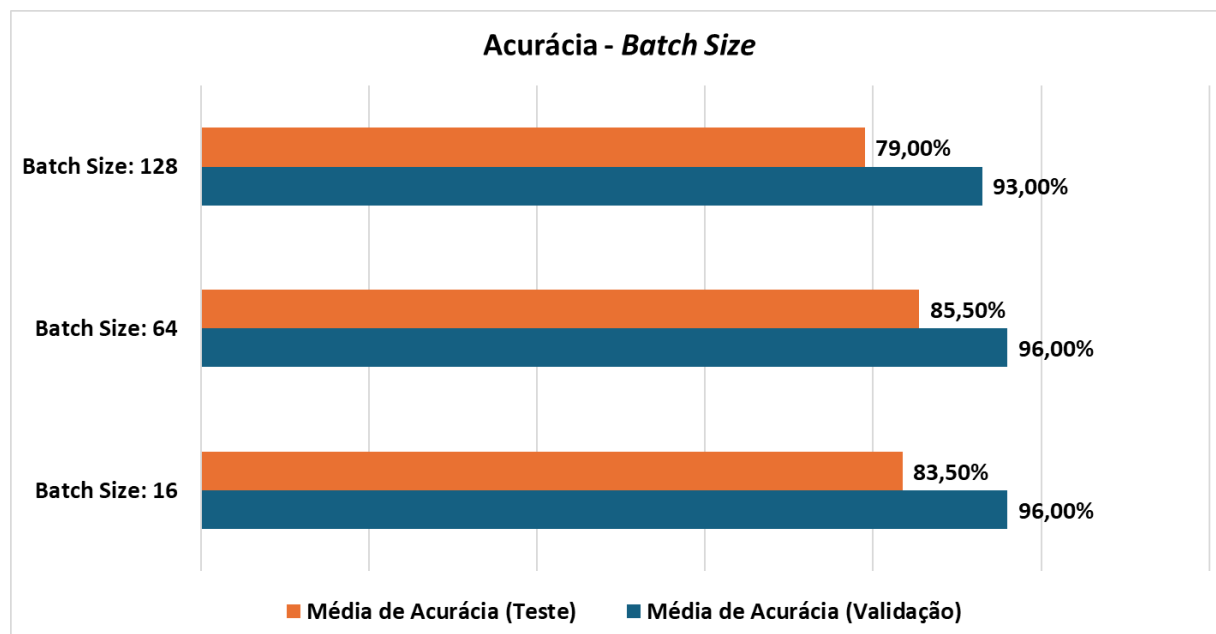
Hiper Parâmetros	Valores
<i>Input Shape</i>	150x150
<i>Batch Size</i>	16,64,128
<i>Epoch</i>	10

Fonte: Elaborado pelo Autor (2025).

Os resultados desse teste demonstram valores de acurácia média para validação muito próximas dos 96,50% encontrados no primeiro teste com valor de *Batch Size* de 32, entretanto o valor encontrado para as acurácias de teste não obteve

melhora e especificamente o de 128, que apresentou um resultado de 79%, conforme Figura 21.

Figura 21 – Média de Validação e teste variando *Batch Size*.



Fonte: Elaborado pelo Autor (2025).

Um fator determinante para a factibilidade do uso do modelo é a capacidade de ele realizar as classificações de maneira mais rápida e o tempo de classificação está ligada ao tempo de processamento. O tempo de processamento por sua vez, está relacionado com a quantidade de dados que a máquina processa, considerando que temos uma máquina com as mesmas especificações técnicas, é possível comparar o as acurácias e o tempo de processamento com diferentes *inputs*.

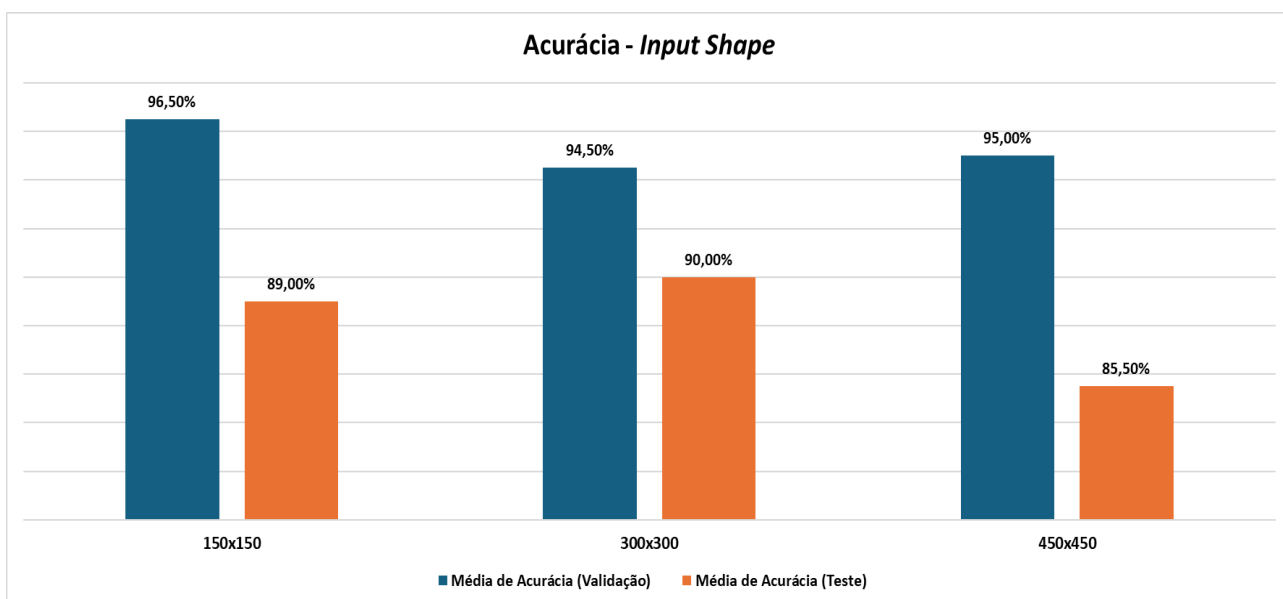
Como já citado anteriormente o *Input Shape*, é um hiper parâmetro de grande importância para o desenvolvimento e sucesso de um modelo, pois quanto maior o formato de entrada dos dados maior detalhe você terá e conseqüentemente maior tempo de processamento, considerando que em todos os testes foi utilizado o mesmo hardware. Nesse contexto é necessário avaliar o tempo e acurácia de maneira conjunta e entender se com o aumento de tempo para o processamento é conveniente para a progressão do modelo.

Como não há possibilidade de implementar o modelo de maneira prática, para que seja possível avaliar dentro deste contexto, há um indicador que pode ser usado, o tempo de treinamento. Durante essa etapa também há uma necessidade de processar todos os dados, logo também é possível identificar uma proporcionalidade

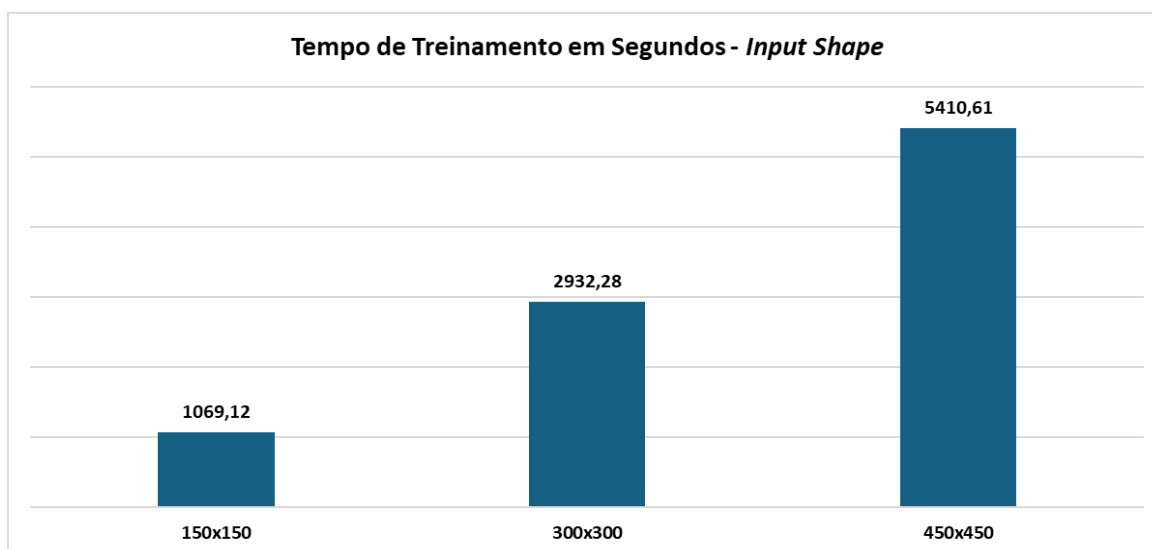
entre tempo e processamento de dados de acordo com o formato que inserimos no modelo.

Os resultados em relação as acurácias encontradas nesta etapa onde variamos o *Input Shape*, demonstraram uma variação irrelevante em relação a validação, pois os valores foram próximos dos encontrados, entretanto encontrou-se um valor de acurácia de teste de 90% quando o input foi de 300x300, de acordo com a Figura 22. Porém, como é possível notar na Figura 23 o tempo de treinamento em comparação a 150x150 é quase três vezes maior quando o *input* foi 300x300, fator a ser considerado já que ao aumentar o formato do dado você terá maior tempo de processamento o que pode influenciar na factibilidade, porque precisará de mais tempo para classificar dados, o que pode dificultar sua implementação em um ambiente industrial.

Figura 22 – Média de Validação e teste variando *Input Shape*.



Fonte: Elaborado pelo Autor (2025).

Figura 23 – Tempo de Treinamento variando *Input Shape*.

Fonte: Elaborado pelo Autor (2025).

Ao todo foram realizados 13 testes durante o desenvolvimento do modelo, 12 com a LeNet e 1 com a AlexNet. Analisando apenas a LeNet, Tabela 5, percebe que ao variar alguns hiper parâmetros, como *Epoch* e quantidade de ciclos, com valores maiores que 20 e 10 respectivamente, e *Batch Size*, há uma queda considerável no valor de acurácia dos testes, algo que pode estar relacionado com *Overfitting*. Quando o modelo apresenta uma boa acurácia de validação, entre 90% até 96%, mas os valores de acurácia de teste estão se distanciando, chegando a valores como 79%, o que significa que o modelo está se ajustando ao conjunto de treinamento, o que não é bom, por que não será útil em um contexto que seja ligeiramente diferente do treinamento.

Tabela 5 – Testes realizados com LeNet.

Teste	<i>Input Shape</i>	<i>Batch Size</i>	<i>Epoch</i>	Ciclo	Acurácia - Validação	Acurácia - Teste
Teste 1	150x150	32	10	10	96,50%	89,00%
Teste 2	150x150	32	20	10	94,00%	85,00%
Teste 3	150x150	32	30	10	90,00%	79,00%
Teste 4	150x150	32	40	10	96,00%	85,00%
Teste 5	150x150	32	50	10	94,50%	81,50%
Teste 6	150x150	32	10	20	91,00%	80,00%
Teste 7	150x150	32	10	30	96,00%	85,00%
Teste 8	300x300	32	10	10	94,50%	90,00%
Teste 9	450x450	32	10	10	95,00%	85,50%
Teste 10	150x150	16	10	10	96,00%	83,50%
Teste 11	150x150	64	10	10	96,00%	85,50%
Teste 12	150x150	128	10	10	93,00%	79,00%

Fonte: Elaborado pelo Autor (2025).

Então para que o modelo apresente bom desempenho, o valor da média de acurácia e de teste devem ser o mais próximo, métrica que pode ser obtida a partir da diferença entre as duas, logo, quanto menor a disparidade, melhor o desempenho do modelo. Considerando um valor de 10% como referência para separar em qual dos testes o modelo sofreu *Overfitting*, dos 12 testes realizados apenas 4 o modelo esteve com um bom desempenho, conforme Tabela 6.

Tabela 6 – Diferença entre Validação e Teste.

Teste	Diferença	<i>Overfitting</i>
Teste 1	7,5%	Não
Teste 2	9,0%	Não
Teste 3	11,0%	Sim
Teste 4	11,0%	Sim
Teste 5	13,0%	Sim
Teste 6	11,0%	Sim
Teste 7	11,0%	Sim
Teste 8	4,5%	Não
Teste 9	9,5%	Não
Teste 10	12,5%	Sim
Teste 11	10,5%	Sim
Teste 12	14,0%	Sim

Fonte: Elaborado pelo Autor (2025).

Dentre os 4, 2 se destacam por apresentar bons números, o Teste 1 e Teste 8, com 7,5% e 4,5% respectivamente. Ambos trazem bons resultados, mas o Teste 8 traz o contexto do tempo e da factibilidade visto que o tempo é um parâmetro importante para sua implantação em um contexto industrial, diante disso a escolha do Teste 1, com seus valores de hiper parâmetros para o desenvolvimento do modelo, demonstrou ser a melhor opção quanto ao contexto e objetivo do trabalho.

5 CONCLUSÃO

Este projeto de pesquisa teve como objetivo avaliar a eficácia das Redes Neurais Convolucionais (CNNs) para a identificação automática do uso de Equipamentos de Proteção Individual (EPIs) em ambientes industriais. O principal objetivo foi contribuir para o aprimoramento da segurança no trabalho. Para isso, foi desenvolvido e treinado um modelo de CNN utilizando um conjunto de imagens de indivíduos com e sem EPIs. O modelo apresentou resultados promissores, alcançando até 96,5% de acurácia, demonstrando sua capacidade de distinguir entre trabalhadores com e sem os equipamentos.

Durante o desenvolvimento do projeto, foram testadas diferentes arquiteturas de rede e ajustados hiper parâmetros para otimizar a acurácia e o tempo de processamento, visando a factibilidade do modelo. A arquitetura LeNet se destacou, apresentando um desempenho superior, com uma acurácia de 96,5%, superando a AlexNet (71,5%) em termos de validação. Ao longo de 12 testes, diversos hiper parâmetros foram avaliados, incluindo *Input Shape*, *Epochs*, *Batch Size* e número de ciclos.

Os testes de *Input Shape* revelaram que o formato 150x150 foi o mais eficiente, apresentando uma acurácia de 96,5% e um tempo de processamento de 1069,12 segundos, em comparação com o formato 450x450, que aumentou o tempo de processamento para 2932,28 segundos, com apenas uma diferença de 1% na acurácia. No caso dos *Epochs*, o número ideal foi 10, pois além de proporcionar boas acurácias de validação (entre 94% e 96,5%), evitou o *overfitting*, que foi observado nos testes a partir da décima época. O *Batch Size* de 32 amostras apresentou o melhor desempenho, com acurácia de 89% nos testes, enquanto os demais valores ficaram entre 79% e 85,5%. Quanto ao número de ciclos, o modelo apresentou resultados consistentes, sendo 10 ciclos a quantidade ideal, atingindo 96,5% de acurácia na validação e 89% nos testes.

A implementação de um sistema automatizado para monitorar o uso de EPIs pode trazer benefícios significativos para a segurança no ambiente industrial. Uma solução como essa permitiria monitorar continuamente os trabalhadores em áreas onde o uso de EPIs é obrigatório, alertando em tempo real sobre a falta do equipamento e gerando dados valiosos para a prevenção de acidentes. Integrar essa

tecnologia aos sistemas de gestão de segurança pode fortalecer a cultura de segurança dentro das organizações, tornando-a mais proativa e orientada por dados.

Embora os resultados deste estudo sejam promissores, o conjunto de dados utilizado foi relativamente pequeno, o que pode limitar a generalização do modelo para diferentes tipos de ambientes e EPIs. Além disso, o desempenho do modelo em condições reais, com iluminação variável e obstruções visuais, ainda precisa ser validado.

Futuras pesquisas podem focar na ampliação do conjunto de dados, incluindo maior diversidade de cenários e EPIs, e no desenvolvimento de técnicas para aumentar a robustez do modelo em ambientes dinâmicos. Também seria interessante explorar como integrar essa tecnologia com outras soluções de segurança, criando abordagens mais completas e eficientes para a proteção dos trabalhadores.

REFERÊNCIAS

ABIRAMI, R. Nandhini; VINCENT, P. M. Durai Raj; SRINIVASAN, Kathiravan; TARIQ, Usman; CHANG, Chuan-Yu. **Deep CNN and deep GAN in computational visual perception-driven image analysis**. In: *Hindawi*, v. 1, p.1–30. 2021. Disponível em: <<https://onlinelibrary.wiley.com/doi/full/10.1155/2021/5541134>>. Acesso em: 06 jun. 2024.

ALOMAR, Khaled; AYSEL, Halil Ibrahim; CA, Xiaohao. **Data augmentation in classification and segmentation: a survey and new strategies**. In: *Jornal Imaging*, v. 9, n. 46, p. 01–26, 2023. Disponível em: <<https://www.mdpi.com/2313-433X/9/2/46>>. Acesso em: 10 jun. 2024.

BEZERRA, Jackson Henrique da Silva; ALMEIDA, Fabrício Moraes de. **Desenvolvimento de modelos preditivos com machine learning: análise de dados para saúde de gestantes e puérperas**. *International Scientific Journal*, v. 19, art. 16, p. 304, jan./dez. 2024. DOI: <http://dx.doi.org/10.6020/1679-9844/v19a16>. Acesso em: 21 abr. 2024. Publicado em: 12 set. 2024. ISSN 1679-9844.

BRASIL. Ministério do Trabalho e Emprego. **Norma Regulamentadora nº 1 – Equipamentos de Proteção Individual (EPI)**. Brasília, DF: MTE, 2024. Disponível em:< <https://www.gov.br/trabalho-e-emprego/pt-br/aceso-a-informacao/participacao-social/conselhos-e-orgaos-colegiados/comissao-tripartite-partitaria-permanente/normas-regulamentadora/normas-regulamentadoras-vigentes/NR01atualizada2024II.pdf> >. Acesso em: 21 out. 2024.

BRASIL. Ministério do Trabalho e Emprego. **Norma Regulamentadora nº 6 – Equipamentos de Proteção Individual (EPI)**. Brasília, DF: MTE, 2025. Disponível em: <<https://www.gov.br/trabalho-e-emprego/pt-br/aceso-a-informacao/participacao-social/conselhos-e-orgaos-colegiados/comissao-tripartite-partitaria-permanente/normas-regulamentadora/normas-regulamentadoras-vigentes/nr-06-atualizada-2025.pdf> >. Acesso em: 15 jan. 2025

BRASIL. Ministério do Trabalho e Emprego. **Norma Regulamentadora nº 17 – Ergonomia**. Brasília, DF: MTE, 2022. Disponível em: < <https://www.gov.br/trabalho-e-emprego/pt-br/aceso-a-informacao/participacao-social/conselhos-e-orgaos-colegiados/comissao-tripartite-partitaria-permanente/normas-regulamentadora/normas-regulamentadoras-vigentes/nr-17-atualizada-2023.pdf>>. Acesso em: 15 jun. 2024.

CAMISASSA, Mara Q. **Segurança e Saúde no Trabalho: NRs 1 a36. Comentadas e Descomplicadas**. 1ª Ed. São Paulo: Editora Método, 2015.

FAN, Jianqing; MA, Cong; ZHONG, Yiqiao. **A selective overview of deep learning**. HHS Public Access, Author Manuscript. In: *Stat Sci.*, v. 1, p. 01–54, 2021. Disponível em: < <https://pmc.ncbi.nlm.nih.gov/articles/PMC8300482/> >. Acesso em: 12 jun. 2024.

GÉRON, Aurélien. **Hands-on Machine Learning With Scikit-Learn, Keras & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. 2ª ed. O' Reilly. United States of America, 2019.

GRUS, Joel. **Data Science do Zero: Primeiras Regras com o Python**. O' Reilly. Rio de Janeiro: Altas Book, 2016.

GONÇALVES, Jéssica Nathália da Silva; SANTOS, Priscila Ribeiro dos. **A importância de um sistema de segurança e saúde no trabalho para a produtividade nas empresas**. 2015. Trabalho de Conclusão de Curso (Graduação) – Universidade do Grande Rio Prof. José de Souza Herdy, Duque de Caxias, 2015.

HOMMA JUNIOR, Edmilson; BOMFIM, Marlon Alves; ALMEIDA FILHO, Ricardo Franco de; SILVA, Amauri Ornellas da; SANTOS, Bruno Samways dos. **Aplicações de técnicas de machine learning nas áreas da Engenharia de Produção: uma revisão sistemática em eventos nacionais**. *XL1 Encontro Nacional de Engenharia de Produção*, Foz do Iguaçu, 2021.

KHAN, Abdullah Ayub; LAGHARI, Asif Ali; AWAN, Shafique Ahmed. **Machine learning in computer vision: a review**. In: *EAI Endorsed Transaction on Scalable*

Informations Systems. 4ª Ed. v. 8 nº 32, p.1–11. 2021. Disponível em: <<https://publications.eai.eu/index.php/sis/article/view/2055>> Acesso em: 04 jun. 2024.

KRICHEN, Moez. **Convolutional neural networks: a survey**. In: *Computers*, v. 12, p. 01–41, 2023. Disponível em: <<https://www.mdpi.com/2073-431X/12/8/151>> Acesso em: 20 jun. 2024.

LE, Trung. **Worksite Safety Monitoring Dataset**. 2024. Disponível em: <https://www.kaggle.com/datasets/lbquctrung/worksite-safety-monitoring-dataset>.

Acesso em: 24 jun. 2024.

MAHARANA, Kiran; MONDAL, Surajit; NEMADE, Bhushankumar. **A review: Data pre-processing and data augmentation techniques**. In: *KeAi Chinese Roots Global Impact*, v. 3, p. 91–99, 2022. Disponível em:<<https://www.sciencedirect.com/science/article/pii/S2666285X22000565>>. Acesso em: 01 jul. 2024.

MITCHELL, Tom M. **Machine Learning**. New York: McGraw-Hill, 1997.

SBAGHBIDI, **Human Faces Object Detection**. 2023. Disponível em: <https://www.kaggle.com/datasets/sbaghbidi/human-faces-object-detection>. Acesso

em: 24 jun. 2024.

SILVA, Maria Adrielle da; GONÇALVES, Anderson Tiago Peixoto. **Contribuições da Implementação do Passo de Segurança do WCM: Um Estudo de Caso em uma Construtora de Habitação Popular**. In: *Revista Produção Online*. Florianópolis, SC, v. 22, n.1, p. 2400-2427, 2022. Disponível em: <<https://www.producaoonline.org.br/rpo/article/view/4585>> . Acesso: 06 jun. 2024.

SILVA, Alexandre Pinto da; FIGUEIREDO, Douglas Barbonaglia Sathler; DUTRA, Frederico Giffoni de Carvalho. **Aplicação da inteligência artificial na segurança do trabalho: uma revisão sistemática de literatura**. *Revista Científica Multidisciplinar Núcleo do Conhecimento*, v. 08, n. 06, p. 116-129, jun. 2023. ISSN 2448-0959.

SOUSA, Allany do Rosário Ferreira de; RODOLPHO, Daniela. **A importância da segurança do trabalho na produção industrial**. *Interface Tecnológica*, Taquaritinga, v. 17, n. 2, 2020.

SOUZA, Vinicius Samy Santana. **Introdução à interpretabilidade de redes neurais convolucionais**. 2023. Dissertação– Universidade Federal de Ouro Preto, Instituto de Ciências Exatas e Biológicas, Departamento de Computação. Orientador: Prof. Dr. Rafael Alves Bonfim de Queiroz. Coorientadora: Profa. Dra. Andrea Gomes Campos Bianchi.

TAYE, Mohammad M. **Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions**. In: *Computation*. Philadelphia University. v. 11 p.1–23. 2023. Disponível em: <<https://www.mdpi.com/2079-3197/11/3/52>>. Acesso: 08 jun. 2024.

VAKALOPOULOU, Maria; CHRISTODOULIDIS, Stergios; BURGOS, Ninon; COLLIOT, Olivier; LEPETIT, Vincent. **Deep learning: basics and convolutional neural networks (CNNs)**. In: COLLIOT, Olivier (ed.). *Machine learning for brain disorders*. Neuromethods, v. 197. Cham: Humana Press, 2023. p.101–139. https://doi.org/10.1007/978-1-0716-3195-9_3. Acesso em: 10 jun. 2024.

VAZ, Gabriella Lorrany Rodrigues; MENDES, Marcos Vinicius Agapito; SANTOS, Rita de Cássia Pedrosa; SILVA, Gabriel Gomes; COSTA, Leandro de Vilhena; PEREIRA, Paulo Elias Carneiro. **Análise dos acidentes de trabalho no setor de mineração brasileiro: analysis of workplace accidents in the Brazilian mining sector**. In: *Revista Cerus*, v. 15, n. 4, p. 226–237, 2023. Disponível em: <<https://www.ojs.unirg.edu.br/index.php/1/article/view/4434>>. Acesso em: 10 jun. 2024.

ANEXO A – ALGORITMOS DO PROJETO DE PESQUISA COMPLETO

```

import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras import layers, models
from tensorflow.keras.preprocessing.image import ImageDataGenerator

drive.mount('/content/drive')

train_dir = '/content/drive/MyDrive/TCC/Dataset/DIRECTORY_MAIN - TRAIN'
test_dir = '/content/drive/MyDrive/TCC/Dataset/DIRECTORY_MAIN - TEST'
validation_dir = '/content/drive/MyDrive/TCC/Dataset/DIRECTORY_MAIN - VALIDATION'

train_datagen = ImageDataGenerator(rescale=1./255)
validation_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    train_dir, target_size=(150, 150), batch_size=32,
    class_mode='binary'
)
validation_generator = validation_datagen.flow_from_directory(
    validation_dir, target_size=(150, 150), batch_size=32,
    class_mode='binary'
)
test_generator = test_datagen.flow_from_directory(
    test_dir, target_size=(150, 150), batch_size=32,
    class_mode='binary'
)

def build_lenet():
    model = models.Sequential([
        layers.Conv2D(6, (5, 5), activation='relu',
input_shape=(150, 150, 3)),
        layers.AveragePooling2D(pool_size=(2, 2)),
        layers.Conv2D(16, (5, 5), activation='relu'),
        layers.AveragePooling2D(pool_size=(2, 2)),
        layers.Flatten(),
        layers.Dense(120, activation='relu'),

```

```

        layers.Dense(84, activation='relu'),
        layers.Dense(1, activation='sigmoid')
    ])
    model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])
    return model

EPOCHS = 10
CYCLES = 30

models_to_train = [
    ('LeNet', 'LeNet')
]

average_accuracies_val = []
average_accuracies_test = []

def build_lenet():
    model = models.Sequential([
        layers.Conv2D(6, (5, 5), activation='relu',
input_shape=(150, 150, 3)),
        layers.AveragePooling2D(pool_size=(2, 2)),
        layers.Conv2D(16, (5, 5), activation='relu'),
        layers.AveragePooling2D(pool_size=(2, 2)),
        layers.Flatten(),
        layers.Dense(120, activation='relu'),
        layers.Dense(84, activation='relu'),
        layers.Dense(1, activation='sigmoid')
    ])
    model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])
    return model

start_time = time.time()

for cycle in range(1, CYCLES + 1):
    print(f"Iniciando ciclo {cycle} de {CYCLES} ciclos...")

    model = build_lenet()

    history = model.fit(train_generator, epochs=EPOCHS,
validation_data=validation_generator, verbose=1)

    _, val_accuracy = model.evaluate(validation_generator,
verbose=0)
    _, test_accuracy = model.evaluate(test_generator, verbose=0)

    plt.figure(figsize=(8, 5))
    plt.plot(history.history['accuracy'], label='Treinamento')

```

```

plt.plot(history.history['val_accuracy'],
label='Validação')
plt.title(f'Acurácia ao longo dos Epochs - LeNet (Ciclo
{cycle})')
plt.xlabel('Epochs')
plt.ylabel('Acurácia')
plt.xticks(range(EPOCHS))
plt.legend()
plt.grid(True)
plt.show()

results_df = pd.DataFrame({
    'Epoch': range(1, EPOCHS + 1),
    'Acurácia Treinamento': history.history['accuracy'],
    'Acurácia Validação': history.history['val_accuracy']
})

print(f"Tabela de resultados para LeNet (Ciclo {cycle}):")
print(results_df)

print(f"Acurácia na validação (Ciclo {cycle}):
{val_accuracy:.4f}")
print(f"Acurácia no teste (Ciclo {cycle}):
{test_accuracy:.4f}")

average_accuracies_val.append(val_accuracy)
average_accuracies_test.append(test_accuracy)

total_average_accuracy_val = np.mean(average_accuracies_val)
total_average_accuracy_test = np.mean(average_accuracies_test)

final_results_df = pd.DataFrame({
    'Ciclo': range(1, CYCLES + 1),
    'Acurácia Média Validação': average_accuracies_val,
    'Acurácia Média Teste': average_accuracies_test
})

print(f"Tabela de médias de validação e teste:")
print(final_results_df)

plt.figure(figsize=(8, 5))
plt.plot(range(1, CYCLES + 1), average_accuracies_val,
label='Acurácia Média Validação', marker='o', color='blue')

```

```

        plt.axhline(y=total_average_accuracy_val, color='r',
linestyle='--', label=f'Média total Validação:
{total_average_accuracy_val:.4f}')
        plt.title('Acurácia Média por Ciclo - Validação')
        plt.xlabel('Ciclo')
        plt.ylabel('Acurácia Média')
        plt.xticks(range(1, CYCLES + 1))

        plt.legend()
        plt.grid(True)
        plt.show()

        plt.figure(figsize=(8, 5))
        plt.plot(range(1, CYCLES + 1), average_accuracies_test,
label='Acurácia Média Teste', marker='x', color='green')
        plt.axhline(y=total_average_accuracy_test, color='g',
linestyle='--', label=f'Média total Teste:
{total_average_accuracy_test:.4f}')
        plt.title('Acurácia Média por Ciclo - Teste')
        plt.xlabel('Ciclo')
        plt.ylabel('Acurácia Média')
        plt.xticks(range(1, CYCLES + 1))
        plt.legend()
        plt.grid(True)
        plt.show()

        end_time = time.time()
        print(f"Tempo total de execução: {end_time - start_time:.2f}
segundos.")

```

ANEXO B – ARQUITETURA DE REDE ALEXNET

Algoritmo 1 – Estrutura AlexNet utilizado no Teste 1.

```
def build_lenet():
    model = models.Sequential()
    model.add(layers.Conv2D(6, (5, 5), activation='relu',
input_shape=(150, 150, 3)))
    model.add(layers.AveragePooling2D(pool_size=(2, 2)))
    model.add(layers.Conv2D(16, (5, 5), activation='relu'))
    model.add(layers.AveragePooling2D(pool_size=(2, 2)))
    model.add(layers.Flatten())
    model.add(layers.Dense(120, activation='relu'))
    model.add(layers.Dense(84, activation='relu'))
    model.add(layers.Dense(1, activation='sigmoid'))

    model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])
    return model
```

Fonte: Elaborado pelo Autor (2025).