

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

**Aplicação de Redes Neurais Convolucionais na
Classificação de Imagens de Raio-X do Tórax para
Auxílio no Diagnóstico da COVID-19**

Flávia Guimarães Gaia Paula

Monografia - MBA em Inteligência Artificial e Big Data

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Flávia Guimarães Gaia Paula

Aplicação de Redes Neurais Convolucionais na Classificação de Imagens de Raio-X do Tórax para Auxílio no Diagnóstico da COVID-19

Monografia apresentada ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientadora: Prof. Dr. Zhao Liang

Versão original

São Carlos

2023

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

G963a Guimarães Gaia Paula, Flávia
 Aplicação de Redes Neurais Convolucionais na
 Classificação de Imagens de Raio-X do Tórax para
 Auxílio no Diagnóstico da COVID-19 / Flávia Guimarães
 Gaia Paula; orientador Zhao Liang. -- São Carlos,
 2023.
 P.

 Trabalho de conclusão de curso (MBA em
 Inteligência Artificial e Big Data) -- Instituto de
 Ciências Matemáticas e de Computação, Universidade
 de São Paulo, 2023.

 1. Inteligência Artificial. 2. Redes Neurais. 3.
 COVID-19. I. Liang, Zhao, orient. II. Título.

Flávia Guimarães Gaia Paula

**Application of Convolutional Neural Networks in the
Classification of Chest X-ray Images for Assistance in the
Diagnosis of COVID-19**

Monograph presented to the Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, as part of the requirements for obtaining the title of Specialist in Artificial Intelligence and Big Data.

Concentration area: Artificial Intelligence

Advisor: Prof. Dr. Zhao Liang

Original version

São Carlos

2023

Este trabalho é dedicado aos alunos da USP, como uma contribuição das Bibliotecas do Campus USP de São Carlos para o desenvolvimento e disseminação da pesquisa científica da Universidade.

AGRADECIMENTOS

Sinto-me profundamente agradecida a Deus, cuja graça me proporcionou a resiliência e coragem para enfrentar e superar cada desafio e complexidade envolvidos na criação deste trabalho.

Aos meus amados pais, Roberto e Maria de Fátima, devo um tributo de gratidão eterna pelo seu amor inabalável e apoio constante, que têm sido a minha rocha durante minha jornada acadêmica. Obrigada por sempre terem fé em minhas capacidades.

Ao meu querido irmão Victor, expresso meu sincero agradecimento pelo seu contínuo incentivo e apoio moral, que foram elementos chave ao longo deste percurso.

Um muito obrigada ao meu orientador, Professor Dr. Zhao Liang, por suas insights valiosas, conhecimento e disponibilidade. Seu acompanhamento foi fundamental para o sucesso deste projeto.

Por último, mas não menos importante, agradeço de coração a todos que contribuíram, direta ou indiretamente, para esta fase crucial da minha vida acadêmica e profissional.

“Inteligência é a habilidade de se adaptar à mudança.”

Stephen Hawking

RESUMO

PAULA, F. **Aplicação de Redes Neurais Convolucionais na Classificação de Imagens de Raio-X do Tórax para Auxílio no Diagnóstico da COVID-19.** 2023. 77p. Monografia (MBA em Inteligência Artificial e Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2023.

O presente trabalho aborda o desenvolvimento e validação de um modelo de classificação de imagens de raio-x para detecção de casos de COVID-19. Em meio à crise global da pandemia de COVID-19, a detecção rápida e precisa de infecções tornou-se crucial. As técnicas tradicionais de diagnóstico, embora eficazes, podem ser lentas ou apresentar limitações. Assim, a Inteligência Artificial, em particular as Redes Neurais Convolucionais, surgiu como uma ferramenta promissora para auxiliar no diagnóstico. No contexto deste estudo, foi criada uma base de dados contendo 2.089 imagens, das quais 505 eram de pacientes diagnosticados com COVID-19 e 1.580 eram de pacientes considerados normais. O modelo proposto foi treinado utilizando-se de técnicas de aumento de dados, alcançando uma acurácia de 98.40% no conjunto de teste. Os resultados destacam o potencial das CNNs como uma ferramenta auxiliar no diagnóstico de COVID-19, promovendo diagnósticos mais rápidos e precisos, e consequentemente, uma resposta médica mais efetiva. Os códigos desenvolvidos durante essa pesquisa podem ser acessados pelo público no GitHub.

Palavras-chave: COVID-19. Diagnóstico. Redes Neurais Convolucionais. Raio-x. Inteligência Artificial.

¹ <https://github.com/flaviagaia/CNNCOVID>

ABSTRACT

PAULA, F. **Application of Convolutional Neural Networks in the Classification of Chest X-ray Images for Assistance in the Diagnosis of COVID-19.** 2023. 77p. Monograph (MBA in Artificial Intelligence and Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2023.

The present work addressed the development and validation of an x-ray image classification model for detecting COVID-19 cases. Amid the global COVID-19 pandemic crisis, rapid and accurate detection of infections has become crucial. Traditional diagnostic techniques, although specific, can be slow or have limitations. Thus, Artificial Intelligence, in particular Convolutional Neural Networks, has emerged as a promising tool to assist in diagnosis. In the context of this study, a database was created containing 2,089 images, of which 505 were from patients confirmed with COVID-19 and 1,580 were from patients considered normal. The proposed model was trained using data augmentation techniques, achieving an accuracy of 98.40% on the test set. The results highlight the potential of CNNs as an auxiliary tool in the diagnosis of COVID-19, promoting faster and more accurate diagnoses, and consequently, a more effective medical response. The codes developed during this research can be accessed by the public on GitHub.

Keywords: COVID-19. Diagnosis. Convolutional Neural Networks. X-ray. Artificial intelligence.

¹ <https://github.com/flaviagaia/CNNCOVID>

LISTA DE FIGURAS

Figura 1 – Visão de alto nível dos principais componentes do mundo da IA.	31
Figura 2 – Organização de uma RNA em camadas	34
Figura 3 – Estrutura simplificada de uma rede neural convolucional	35
Figura 4 – Exemplo de imagens de raio-x	37
Figura 5 – Efeito das técnicas do ImageDataGenerator em uma imagem de raio-x.	43
Figura 6 – Gráfico de calor mostrando a acurácia de validação para diferentes números de épocas e tamanhos de lote.	45
Figura 7 – Diagrama do Modelo.	46
Figura 8 – Acurácia do Modelo.	48
Figura 9 – Perda do Modelo.	48
Figura 10 – Acurácia por Época.	49
Figura 11 – Matriz de confusão.	52
Figura 12 – Curva ROC.	53
Figura 13 – Relatório de classificação do modelo.	53

LISTA DE TABELAS

Tabela 1 – Comparação dos Trabalhos Correlatos	39
Tabela 2 – Distribuição do número de imagens por classe em diferentes conjuntos de dados	42
Tabela 3 – Comparação de Resultados com Trabalhos Correlatos	54

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
Adam	Algoritmo de otimização adaptativa
AUC	Área Sob a Curva (Area Under the Curve)
AUC-ROC	Área Sob a Curva da Característica de Operação do Receptor (Area Under the Receiver Operating Characteristic curve)
Binary Cross-Entropy	Entropia cruzada binária, função de perda usada em problemas de classificação binária
CNNs	Redes Neurais Convolucionais
COVID-19	Doença causada pelo SARS-CoV-2
CT	Tomografia Computadorizada
DL	Deep Learning (Aprendizado Profundo)
GPU	Unidade de processamento gráfico
Grid Search	Técnica de otimização de hiperparâmetros que explora várias combinações predefinidas
IgG e IgM	Classes de anticorpos
ImageDataGenerator	Classe do TensorFlow para aumento de dados em imagens
IBGE	Instituto Brasileiro de Geografia e Estatística
IA	Inteligência Artificial
LaTeX	Lamport TeX
MaxPooling	Camada que realiza redução de dimensionalidade em uma CNN
OMS	Organização Mundial da Saúde
OPAS-W	Organização Pan-Americana da Saúde - Organização Mundial da Saúde
Raio-X	Radiografia do Tórax
ReLU	Rectified Linear Unit (Unidade Linear Retificada)
RNA	Redes Neurais Artificiais

ROC	Curva Característica de Operação do Receptor (Receiver Operating Characteristic curve)
SARS-CoV-2	Vírus responsável pela COVID-19
TPU	Unidade de processamento de tensor
USP	Universidade de São Paulo
USPSC	Campus USP de São Carlos
Google Colab	Plataforma de pesquisa colaborativa baseada em nuvem para desenvolvimento interativo
classification_report	Um relatório que fornece várias métricas de avaliação de modelos de classificação, como precisão, recall e f1-score
CT	Tomografia Computadorizada
RT-PCR	Reação em Cadeia da Polimerase com Transcrição Reversa (um tipo de teste molecular)

LISTA DE SÍMBOLOS

Γ	Letra grega Gama
Λ	Lambda
ζ	Letra grega minúscula zeta
\in	Pertence

SUMÁRIO

1	INTRODUÇÃO	29
1.1	Objetivos	30
1.2	Conteúdo e Organização	30
2	FUNDAMENTAÇÃO TEÓRICA	31
2.1	Inteligência Artificial	31
2.2	Machine Learning	32
2.3	Deep Learning	33
2.3.1	Redes Neurais Artificiais	34
2.3.2	Redes Neurais Convolucionais	35
2.4	COVID-19	36
2.5	Trabalhos Correlato	37
3	METODOLOGIA E DESENVOLVIMENTO	41
3.1	Coleta de Dados	41
3.2	Pré-processamento	42
3.3	Seleção de Hiperparâmetros	44
3.4	Arquitetura do Modelo	45
3.5	Treinamento e Teste	47
4	RESULTADOS	51
4.1	Métricas de Desempenho	51
4.2	Matriz de Confusão	51
4.3	Análise da Curva ROC	52
4.4	Relatório de Classificação	53
4.5	Avaliação Comparativa dos Resultados	54
5	CONCLUSÃO	57
	REFERÊNCIAS	59
	APÊNDICES	63
	APÊNDICE A – CÓDIGO DE EXTRAÇÃO DOS DADOS	65
	APÊNDICE B – CÓDIGO DE DIVISÃO DOS DADOS	69

APÊNDICE C – CÓDIGO DE AJUSTE DOS HIPERPARÂMETROS	71
APÊNDICE D – CÓDIGO DE TREINAMENTO E TESTE DO MO- DELO	75

1 INTRODUÇÃO

Desde o seu surgimento na China em dezembro de 2019, a COVID-19, uma doença infecciosa causada pelo SARS-CoV-2, desencadeou uma crise global sem precedentes, afetando profundamente a saúde e a economia das nações. Dada a velocidade alarmante com que o vírus se propagou, a OMS declarou a situação como uma Emergência de Saúde Pública de Interesse Internacional em 30 de janeiro de 2020, posteriormente reconhecendo-a como pandemia em 11 de março do mesmo ano (OMS, 2023). Os sintomas mais comuns associados a esta doença incluem desconforto respiratório, febre e tosse, com casos mais graves evoluindo para pneumonia (MAHASE, 2020).

Em resposta à disseminação massiva da doença, o uso de imagens médicas, particularmente da Tomografia Computadorizada (CT) e Radiografia do Tórax (Raio-X), emergiu como uma abordagem inicial de diagnóstico. Diferentemente dos testes moleculares, como o RT-PCR, e dos testes sorológicos, os exames por imagens CT e Raio-X apresentam diversas vantagens. Primeiramente, são exames precisos que fornecem detalhes claros das condições pulmonares do paciente. Além disso, essas modalidades de imagem permitem acompanhar a evolução da doença, oferecendo insights sobre a progressão ou regressão da infecção. Por fim, eles não só servem para teste, mas também desempenham um papel crucial no diagnóstico e tratamento da doença, guiando médicos em suas decisões clínicas. Isto é atribuído tanto ao alto custo de outras modalidades de diagnóstico quanto à prevalência crescente da doença globalmente. Atualmente, essas modalidades de imagem formam o protocolo padrão para a avaliação de casos suspeitos necessitando de um diagnóstico imediato (OLIVEIRA, 2020).

Neste contexto desafiador, evidencia-se a demanda por métodos de diagnóstico ágeis e precisos. Apesar da relevância das imagens de raio-x do tórax no diagnóstico da COVID-19, sua interpretação manual pode ser um processo complexo e demorado. Aqui, as Redes Neurais Convolucionais (CNNs) revelam um potencial significativo. Elas oferecem uma solução automatizada, permitindo diagnósticos rápidos e precisos, essenciais para um tratamento adequado e tempestivo (HOSNY MOHAMED M. DARWISH, 2021).

A eficácia das CNNs na classificação de imagens de raio-x do tórax sugere um horizonte promissor no diagnóstico da COVID-19. Tais redes são projetadas para discernir características vitais das imagens ao serem treinadas com vastos conjuntos de dados, facilitando a identificação de padrões que poderiam ser imperceptíveis para a análise humana (SARKI KHANDAKAR AHMED, 2022). Especificamente em relação à COVID-19, as CNNs têm a capacidade de detectar sinais característicos nas imagens, como opacidades em vidro fosco e outras anormalidades pulmonares típicas da doença (FLOREZ YESENIA CONCHA-RAMOS, 2022).

Considerando a magnitude dos impactos da pandemia e a necessidade urgente de soluções eficazes, este estudo propõe-se a explorar profundamente a aplicação das CNNs como ferramenta vital para o diagnóstico acelerado da COVID-19.

1.1 Objetivos

O estudo em questão tem como objetivo central o desenvolvimento de um modelo automatizado, baseado em técnicas de Inteligência Artificial, para auxiliar no diagnóstico da COVID-19. A eficácia das Redes Neurais Convolucionais na análise e classificação de imagens motiva os seguintes objetivos específicos:

1. Preparar e pré-processar conjuntos de dados de imagens para treinamento;
2. Aplicar e otimizar técnicas de CNNs para a classificação de imagens;
3. Desenvolver um protótipo de sistema computacional para diagnóstico da COVID-19.

Este trabalho traz inovações significativas na área de diagnóstico de COVID-19 através de imagens médicas. As principais inovações introduzidas neste TCC são:

- A implementação de técnicas avançadas de pré-processamento para melhorar a qualidade e relevância das imagens médicas, tornando o modelo mais eficiente e preciso.
- Um estudo aprofundado dos hiperparâmetros, explorando diferentes combinações para otimizar a performance da rede neural.
- Realização de experimentos extensivos, testando a eficácia do modelo em diversos cenários e configurações para garantir robustez e confiabilidade.

1.2 Conteúdo e Organização

O trabalho é estruturado em cinco capítulos. O Capítulo 2 é dedicado ao referencial teórico, enquanto o Capítulo 3 detalha o desenvolvimento do modelo, e o Capítulo 4 foca nos resultados. O Capítulo 5 conclui o estudo e aponta para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

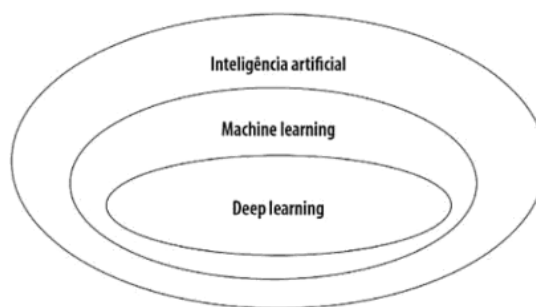
A evolução tecnológica permitiu a inserção de ferramentas robustas para o auxílio no diagnóstico de doenças, como a COVID-19. Esta seção delineia os princípios subjacentes a tais técnicas, centrando-se na aplicação da Inteligência Artificial, particularmente no uso de Redes Neurais Convolucionais.

2.1 Inteligência Artificial

A Inteligência Artificial (IA) emerge como um segmento intrincado da ciência da computação e engenharia de computação. O principal objetivo da IA é mimetizar habilidades cognitivas humanas, abrangendo raciocínio, planejamento, solução de problemas e outras funções cognitivas. (NORVIG; RUSSELL, 2013) ressaltam que a IA tem a intenção de produzir máquinas que adotem ações e fornecem respostas com base em sua programação e análise de dados, semelhantemente à cognição humana.

A vastidão da IA engloba diferentes teorias e tecnologias, e, no interior dessa diversidade, surgem duas categorias proeminentes: Machine Learning e Deep Learning, como é ilustrada na Figura 1.

Figura 1 – Visão de alto nível dos principais componentes do mundo da IA.



Fonte: Taulli (2020)

A IA na medicina é usada para analisar grandes quantidades de dados, identificar padrões e prever resultados (Medicine Cursos, 2023). Essa tecnologia pode ser aplicada em diversas áreas da medicina, incluindo diagnóstico, tratamento e pesquisa (Medicine Cursos, 2023).

As principais áreas em que a IA é aplicada na medicina incluem triagem em massa, diagnóstico por imagem, dados laboratoriais, eletrodiagnóstico, diagnóstico genético, dados clínicos, notas de operação, registros eletrônicos de saúde e registros de dispositivos vestíveis (eDialog, 2023).

A IA também tem sido usada para melhorar a precisão das avaliações médicas. As máquinas são ágeis e racionais, elevando a acurácia da hipótese diagnóstica (José Aldair Morsch , 2021). Elas também identificam sintomas relacionados a doenças e avisam sobre erros na dosagem de medicamentos (José Aldair Morsch , 2021).

Além disso, a IA tem sido usada para otimizar tarefas e melhorar a precisão das avaliações médicas. Isso é possível graças ao suporte de tecnologias como big data, machine learning e algoritmos sofisticados (José Aldair Morsch , 2021).

2.2 Machine Learning

Machine Learning é um subcampo da IA que se concentra no uso de dados e algoritmos para imitar a maneira como os humanos aprendem, melhorando gradualmente sua precisão (IBM, 2023b). É um método de análise de dados que automatiza a construção de modelos analíticos (RESPOSTAS, 2022). A ideia é que sistemas podem aprender com dados, identificar padrões e tomar decisões com o mínimo de intervenção humana (RESPOSTAS, 2022).

Os algoritmos de Machine Learning são treinados para fazer classificações ou previsões, revelando os principais insights em projetos de mineração de dados (IBM, 2023b). Esses insights subsequentemente conduzem a tomada de decisões em aplicativos e negócios, impactando de forma ideal as principais métricas de crescimento (IBM, 2023b).

Os métodos de Machine Learning são normalmente criados usando frameworks que aceleram o desenvolvimento da solução, como TensorFlow e PyTorch (IBM, 2023b). À medida que o big data continua a expandir e crescer, a demanda de mercado por cientistas de dados aumentará cada vez mais (IBM, 2023b).

Existem três tipos principais de Machine Learning: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço (RESPOSTAS, 2022).

- **Aprendizado Supervisionado:** Este é o tipo mais comum. Os algoritmos são treinados em um conjunto de dados “rotulados”, onde as respostas corretas são conhecidas. O algoritmo faz previsões com base nesses rótulos e ajusta suas previsões à medida que aprende com seus erros.
- **Aprendizado Não Supervisionado:** Neste tipo, os algoritmos são treinados em conjuntos de dados “não rotulados”. O algoritmo deve identificar padrões e relações nos dados por conta própria.
- **Aprendizado por Reforço:** Este é um tipo especializado de Machine Learning onde um agente aprende a tomar decisões ao interagir com seu ambiente. O agente recebe recompensas ou punições com base nas decisões que toma, e seu objetivo é maximizar a recompensa total.

Machine Learning tem uma ampla gama de aplicações em muitos campos diferentes, como na saúde pode ser usado para prever doenças, ajudar no diagnóstico e personalizar o tratamento, nas finanças os algoritmos podem ser usados para detectar fraudes, otimizar carteiras de investimentos e fazer previsões econômicas. Também pode ser aplicado no varejo, as empresas usam Machine Learning para personalizar as recomendações do cliente, otimizar os preços e gerenciar o inventário. No transporte é fundamental para o desenvolvimento de veículos autônomos. Também pode ser usado para otimizar rotas e prever a demanda por transporte (Quigo, 2022).

Essa seção fornece uma introdução abrangente ao Machine Learning, destacando a importância de diferentes abordagens de aprendizado em problemas de classificação, regressão, aprendizado supervisionado, não supervisionado e por reforço. Esses conceitos serão fundamentais para a compreensão da aplicação de Redes Neurais Convolucionais no diagnóstico de COVID-19 por imagens de raio-X.

2.3 Deep Learning

Classificado como um subdomínio de Machine Learning, o Deep Learning visa uma análise profunda e acurada de grandes volumes de dados, ultrapassando a capacidade das técnicas de aprendizado de máquina convencionais (Kellison Ferreira, 2019). As bases do Deep Learning assentam-se em redes neurais multicamadas, que imitam a complexa maneira pela qual o cérebro humano processa informações (IBM, 2023).

A potência do Deep Learning reside em sua capacidade de operar com dados não estruturados, como imagens e textos, eliminando etapas laboriosas de pré-processamento que demandam expertise manual. Esta tecnologia sustenta aplicações inovadoras que potencializam a automação em domínios variados de Inteligência Artificial, variando desde assistentes virtuais a sistemas avançados de detecção de fraude (IBM, 2023).

As aplicações do Deep Learning são vastas e variadas. Por exemplo, ele é usado em carros autônomos, onde milhões de conjuntos de dados são alimentados em um sistema para construir um modelo, treinar as máquinas para aprender e, em seguida, testar os resultados em um ambiente seguro (Marina Chatterjee, 2022). Outra aplicação popular do Deep Learning são os assistentes virtuais, como Alexa, Siri e Google Assistant. Cada interação com esses assistentes fornece a eles uma oportunidade de aprender mais sobre sua voz e sotaque, proporcionando uma experiência de interação humana secundária (Marina Chatterjee, 2022).

No setor de saúde, o Deep Learning encontrou sua aplicação na detecção de doenças auxiliada por computador e no diagnóstico auxiliado por computador. É amplamente utilizado para pesquisa médica, descoberta de medicamentos e diagnóstico de doenças potencialmente fatais, como câncer e retinopatia diabética, por meio do processo de imagem

médica (Avijeet Biswal, 2023).

No entretenimento, empresas como Netflix, Amazon, YouTube e Spotify fornecem recomendações relevantes de filmes, músicas e vídeos para melhorar a experiência do cliente. Isso é possível graças ao Deep Learning. Com base no histórico de navegação, interesse e comportamento de uma pessoa, as empresas de streaming online fornecem sugestões para ajudá-los a fazer escolhas de produtos e serviços (Avijeet Biswal, 2023).

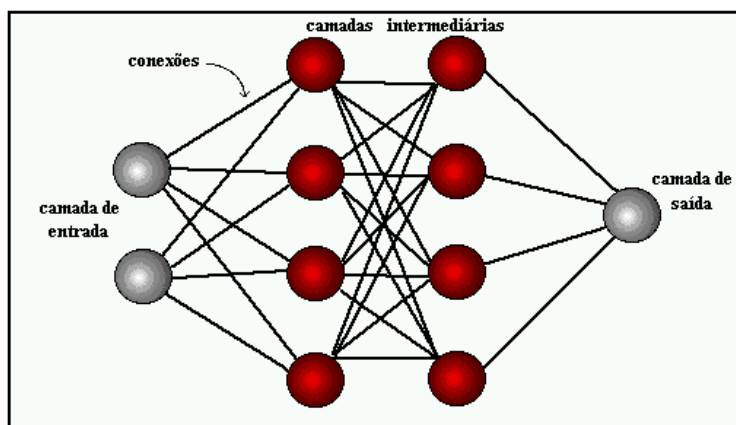
2.3.1 Redes Neurais Artificiais

A pesquisa em Redes Neurais Artificiais (RNAs) representa uma abordagem poderosa na área de Aprendizado de Máquina, inspirada na estrutura e funcionalidade do sistema nervoso humano. A origem das RNAs remonta a 1943, quando o artigo de McCulloch e Pitts, "A Logical Calculus of the Ideas Immanent in Nervous Activity", estabeleceu os princípios iniciais inspirados no funcionamento cerebral (MCCULLOCH; PITTS, 1943).

Uma característica fundamental das RNAs é a capacidade de aprender por meio de exemplos e, igualmente importante, de generalizar esse conhecimento adquirido. Essa habilidade é análoga à forma como os seres humanos aprendem com base em experiências anteriores (Fleck, Leandro *et al.*, 2016).

As RNAs consistem em neurônios artificiais que executam cálculos de funções matemáticas, muitas vezes não-lineares. Esses neurônios são organizados em camadas interconectadas, as quais refletem a organização do cérebro humano. A arquitetura típica de uma RNA inclui a Camada de Entrada, Camadas Intermediárias ou Escondidas, e a Camada de Saída, conforme ilustrado na Figura 2.

Figura 2 – Organização de uma RNA em camadas



Fonte: ICMC (2023)

A estrutura dos neurônios artificiais é essencial para o processamento de informações em uma RNA. Cada neurônio recebe entradas ponderadas por conexões que armazenam o conhecimento adquirido. A função de ativação dos neurônios determina como as entradas

são processadas e geram uma saída correspondente (BRAGA; LUDERMIR; CARVALHO, 2007).

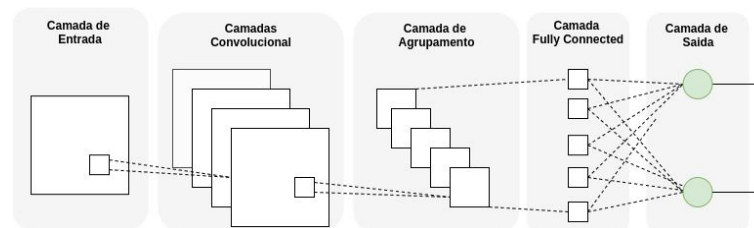
As RNAs podem adotar diversas arquiteturas, desde redes feedforward com uma única camada até redes recorrentes com realimentação. Essa diversidade de arquiteturas permite que essas redes abordem uma ampla gama de problemas, desde tarefas simples até problemas complexos de processamento de informações (HAYKIN, 2001).

Em resumo, as RNAs são uma ferramenta poderosa na área de Aprendizado de Máquina que se baseia na emulação da funcionalidade cerebral. Sua capacidade de aprender por meio de exemplos, generalizar informações e adaptar-se a várias arquiteturas as torna valiosas em uma variedade de aplicações de IA e aprendizado de máquina.

2.3.2 Redes Neurais Convolucionais

As Redes Neurais Convolucionais (CNNs) constituem-se como uma categoria distinta dentro do vasto domínio do aprendizado de máquina, sendo cruciais para o progresso do aprendizado profundo (IBM, 2023a). Caracterizam-se pela sua estrutura em camadas, abrangendo uma camada de entrada, várias camadas ocultas e uma camada de saída. O que as diferencia de outras redes neurais é sua notável eficácia no processamento de dados de imagem, áudio e fala. Sua composição engloba principalmente três tipos de camadas: Camada Convolucional, Camada de Agrupamento (Pooling) e Camada Totalmente Conectada (Fully Connected) (IBM, 2023a), como ilustra a Figura 3.

Figura 3 – Estrutura simplificada de uma rede neural convolucional



Fonte: (BARBOSA GOVINDA MOHINI G. BEZERRA, 2021)

Tal arquitetura é inspirada no sistema visual humano. Analogamente a como certos neurônios em nosso cérebro identificam características em imagens, as CNNs são treinadas para reconhecer tais características de maneira autônoma (AWARI, 2023). Segundo (GOODFELLOW YOSHUA BENGIO, 2016), as CNNs, embora semelhantes às Redes Neurais Artificiais, contêm mais camadas e, portanto, mais operações. Em tal estrutura, cada camada tem o papel de discernir informações específicas dos dados de entrada. A arquitetura de uma rede neural convolucional é formada por uma sucessão de blocos de construção que extraem as características que discriminam a classe pertencente à imagem dos outros (Sarah Madeleine, 2021). Um bloco de construção consiste em

uma ou mais: camada convolucional que processa os dados de um campo receptor (Sarah Madeleine , 2021).

No contexto médico, CNNs destacam-se em diagnósticos como detecção de câncer e análise de exames (Fenestra , 2023), superando métodos tradicionais de classificação de imagens, como k-NN, SVM e Random Forest (Sarah Madeleine , 2021). Suas aplicações se estendem a outras áreas como aerofotogrametria e planejamento de tráfego urbano (AYRES, 2021).

O que amplifica o potencial das CNNs é seu desempenho elevado em tarefas específicas de aprendizado de máquina, sobretudo no processamento de imagens e áudio. Elas simplificam o processo de aprendizado ao extrair características de forma autônoma, dispensando métodos manuais anteriores (IBM, 2023a). Contudo, é vital reconhecer suas limitações: o treinamento intensivo, a complexidade crescente com mais camadas e a necessidade de vastos conjuntos de dados (IBM, 2023a).

2.4 COVID-19

A COVID-19 é uma doença respiratória que apresenta uma variedade de sintomas, que vão desde leves a graves. Estes incluem febre, tosse, fadiga, dor de cabeça, perda do olfato ou paladar, dor muscular ou de garganta, diarreia, entre outros. Em alguns casos, os pacientes podem apresentar sintomas mais graves, como dificuldade para respirar, insuficiência pulmonar, pneumonia e até mesmo a morte (CONTROL; PREVENTION, 2022).

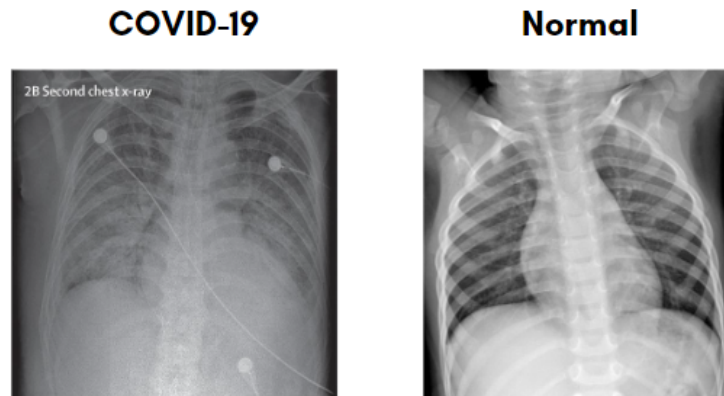
O diagnóstico da COVID-19 é realizado por meio de testes moleculares (RT-PCR) no início da doença ou testes sorológicos a partir da segunda semana. Neste período, a quantidade de vírus diminui e o indivíduo começa a produzir anticorpos contra o vírus, principalmente das classes IgG e IgM (ANDRADE, 2020).

Os exames de imagem do tórax são parte integrante da investigação diagnóstica de pacientes com suspeita ou probabilidade de COVID-19. Eles são especialmente úteis em locais onde a RT-PCR não está disponível ou quando os resultados são inicialmente negativos na presença de sintomas sugestivos de COVID-19 (OPAS-W, 2020).

Um raio-x normal do tórax geralmente apresenta a transparência pulmonar preservada (ICRX Solucion, 2023). Este exame é usado para observar aspectos como a estrutura do pulmão, as bordas cardíacas, os vasos sanguíneos, o mediastino e o arcabouço ósseo (ICRX Solucion, 2023). Na radiografia do tórax, os ossos (costelas, vértebras da coluna) aparecem em branco (indicando que os ossos são mais densos), enquanto os pulmões aparecem mais escuros por estarem preenchidos essencialmente por ar (Você pergunta , 2021), como mostra a Figura 4.

Atualmente, não existe um tratamento específico aprovado para o combate à

Figura 4 – Exemplo de imagens de raio-x



Fonte: Elaboração própria

COVID-19. No entanto, várias terapias estão sendo avaliadas em ensaios clínicos (souto, Xênia Macedo, 2020). A vacinação tem se mostrado uma ferramenta eficaz na luta contra a doença. As vacinas funcionam introduzindo uma forma inativada do coronavírus SARS-CoV-2 ou uma parte dele no sistema imunológico. Isso prepara o organismo para combater possíveis infecções futuras pelo vírus (Hewings-Martin, Yella, 2021).

Medidas de prevenção e controle de infecção são fundamentais para conter a pandemia. Estas incluem o uso de máscaras, higienização das mãos, distanciamento social e manter-se atualizado com as vacinas (Center for Disease Control and Prevention, 2023).

Nesse contexto, técnicas de diagnóstico de fácil acesso e rápidas podem ser ferramentas promissoras para auxiliar no diagnóstico da COVID-19. Um exemplo disso é a classificação de imagens de raio-x do tórax por meio de CNNs, que têm se mostrado eficazes na detecção e classificação da doença.

2.5 Trabalhos Correlato

A aplicação das CNNs na análise e classificação de imagens médicas tem atraído atenção nos últimos anos, especialmente na detecção da COVID-19. Esta seção aborda trabalhos notáveis no campo, destacando suas metodologias, impactos e limitações.

Um estudo relevante de (RESHI *et al.*, 2021) utiliza CNNs na classificação de imagens de raio-x do tórax relacionadas à COVID-19. O trabalho aborda fases como balanceamento de dados, avaliação de imagens por profissionais médicos e aumento de dados. A estratégia empregada, que utiliza uma CNN profunda, alcança uma precisão surpreendentemente próxima de 100%.

Em contrapartida, o estudo de (OHATA *et al.*, 2020) experimenta com várias arquiteturas de CNNs que foram inicialmente treinadas com o conjunto de dados ImageNet. Utilizando técnicas de Transfer Learning, o estudo avalia a eficácia dessas redes em

conjunto com algoritmos tradicionais, como k-Nearest Neighbor e SVM. As combinações mais eficientes alcançaram precisão e pontuação F1 superiores a 95%.

Com foco na eficiência computacional, (BARBOSA; COELHO; BAFFA, 2021) otimiza uma CNN para reduzir o custo de processamento sem sacrificar a acurácia na detecção da COVID-19. A rede, que é relativamente simplificada com apenas três camadas convolucionais, demonstrou uma notável precisão de 99,36%.

É importante ressaltar que a aplicação de CNNs para análise de imagens de raio-x não se limita à COVID-19. Como demonstrado por (CHRISTOFOLLETTI; ATTUX, 2022), as Dense Convolutional Networks foram utilizadas para identificar outras doenças com resultados encorajadores.

Além disso, com o objetivo de pré-diagnóstico da COVID-19 pela detecção de pneumonia, (SILVA *et al.*, 2020) adotou as arquiteturas InceptionResNetV2 e ResNetX50 para classificar imagens de raio-x. Dentre as arquiteturas testadas, a ResNetX50 mostrou-se superior conforme as métricas utilizadas.

Vários estudos exploraram metodologias distintas no uso de CNNs para classificação de imagens médicas. No trabalho de (RESHI *et al.*, 2021), por exemplo, destaca-se a abordagem de balanceamento de conjuntos de dados e aumento de dados, um aspecto crucial para evitar modelos tendenciosos.

Por outro lado, (OHATA *et al.*, 2020) focaram na utilização do Transfer Learning, um método que busca aproveitar conhecimento pré-adquirido de outras tarefas para melhorar a performance na tarefa alvo. A escolha dos algoritmos k-Nearest Neighbor e SVM nesse contexto sugere uma tentativa de combinar abordagens tradicionais do aprendizado de máquina com técnicas do deep learning.

A tabela a seguir apresenta uma comparação das abordagens e dos dados utilizados nos trabalhos correlatos com a abordagem proposta neste trabalho. Os resultados específicos de cada abordagem serão discutidos em detalhes no capítulo de Resultados.

Tabela 1 – Comparação dos Trabalhos Correlatos

Trabalho	Metodologia	Dados	Objetivo
(RESHI <i>et al.</i> , 2021)	CNN profunda	Imagens de raio-x do tórax	Detecção de COVID-19
(OHATA <i>et al.</i> , 2020)	Transfer Learning	Imagens de raio-x do tórax	Detecção de COVID-19
(BARBOSA; COELHO; BAFFA, 2021)	CNN	Imagens de raio-x do tórax	Detecção de COVID-19
(CHRISTOFOLLETTI; ATTUX, 2022)	CNN	Imagens de raio-x do tórax	Detecção de COVID-19 e Pneumonia viral
(SILVA <i>et al.</i> , 2020)	InceptionResNetV2 e ResNetX50	Imagens de raio-x do tórax	Detecção de COVID-19 e Pneumonia viral

3 METODOLOGIA E DESENVOLVIMENTO

Este capítulo detalha o processo de implementação do projeto. O projeto foi desenvolvido na linguagem de programação Python, conhecida por seu alto desempenho, produtividade e legibilidade, tornando-a ideal para trabalhar com problemas de análise de dados, visualização de dados e uso de técnicas de DL e RNAs. Várias bibliotecas do Python foram utilizadas, incluindo: NumPy, Pandas, Scikit-learn, Matplotlib, Seaborn, Keras e Tensorflow (Data Science Academy, 2023). Outra tecnologia empregada foi o Google Colaboratory, comumente referido como “Google Colab” ou simplesmente “Colab”, um projeto de pesquisa para prototipagem de modelos de aprendizado de máquina em opções de hardware poderosas, como GPUs e TPUs. Ele fornece um ambiente de notebook Jupyter sem servidor para desenvolvimento interativo (BISONG, 2019).

3.1 Coleta de Dados

Para este trabalho, foram selecionados três conjuntos de dados para conduzir os experimentos:

1. Chest X-Ray Images (Pneumonia), conjunto de dados de 3 classes, 'Normal', 'Pneumonia Bacteriana' e 'Pneumonia Viral', disponível no Kaggle (KERMANY *et al.*, 2018).
2. Conjuntos de dados de 4 classes, 'Normal', 'Covid', 'Pneumonia Bacteriana' e 'Pneumonia Viral', do CMSC-678-ML-Project disponível no GitHub (JAMDADÉ, 2020).
3. Conjunto de dados públicos abertos de radiografias de tórax e imagens de tomografia computadorizada de pacientes positivos ou suspeitos de COVID-19 ou outras pneumonias virais e bacterianas, disponível no GitHub (COHEN *et al.*, 2020)

Basicamente, todos os conjuntos de dados selecionados possuem ao menos uma das classes utilizadas nesse trabalho Covid ou Normal e contém imagens de raio-x do tórax.

Foi criada uma base de dados para este projeto onde os dados foram obtidos através da seleção de 2 classes de imagens de raio-x, sendo elas "Covid" e "Normal", oriundas dos conjuntos de dados citados acima para alimentar o modelo com o maior número de iamgens possíveis, como mostra a tabela a seguir:

A base de dados criada foi filtrada para não trazer imagens duplicadas, por isso foi extraída apenas 1 imagem da base (JAMDADÉ, 2020). Sem nenhum tratamento, a base é constituída de 2.089 imagens. Das 2.089 imagens, 505 possuem o diagnóstico

Tabela 2 – Distribuição do número de imagens por classe em diferentes conjuntos de dados

Dataset	Classe	Nº de imagens extraídas da classe	Total de imagens
(KERMANY <i>et al.</i> , 2018)	Covid	504	504
	Normal	00	
(JAMDADE, 2020)	Covid	01	80
	Normal	79	
(COHEN <i>et al.</i> , 2020)	Covid	00	1502
	Normal	1502	

positivo de COVID-19, e 1580 foram classificadas sobre o diagnóstico de normal. Portanto, aproximadamente 24.18% das imagens são da classe COVID-19 e 75.82% são da classe Normal.

Para o processo de extração, tratamento e organização destas imagens, foi utilizado um código, que está disponível no Apêndice A deste trabalho. Este código foi responsável por acessar os diferentes repositórios de dados, coletar as imagens de interesse conforme os critérios estabelecidos (como a classe de diagnóstico) e, em seguida, organizá-las na estrutura desejada, garantindo a exclusão de duplicatas. A estruturação cuidadosa dos dados é essencial para garantir a eficácia do modelo de aprendizado de máquina subsequente e evitar ruídos ou vieses indesejados.

Após a coleta e organização inicial das imagens, foi fundamental segmentar esse conjunto de dados em subconjuntos para treinamento e teste. Para essa divisão, foi utilizado um código, que pode ser encontrado no Apêndice B deste TCC. Uma função chamada `split_data` foi definida para dividir os dados. Esta função, basicamente, pega todos os arquivos da pasta de origem, embaralha-os aleatoriamente, e então separa-os de acordo com uma proporção especificada sendo 80% dos dados para treinamento do modelo e 20% usados para teste. Os arquivos são então copiados para suas respectivas pastas de treinamento e teste.

Aplicando essa função, as imagens foram divididas em 353 imagens de treinamento e 152 imagens de teste para COVID; e 1106 imagens de treinamento e 474 imagens de teste para imagens Normais. No total, o dataset segmentado é constituído por 2085 imagens. Este processo de divisão é essencial para evitar o sobreajuste durante o treinamento do modelo, e para garantir que o modelo seja avaliado em um conjunto de dados que nunca viu antes, proporcionando uma avaliação realista da sua capacidade de generalização.

3.2 Pré-processamento

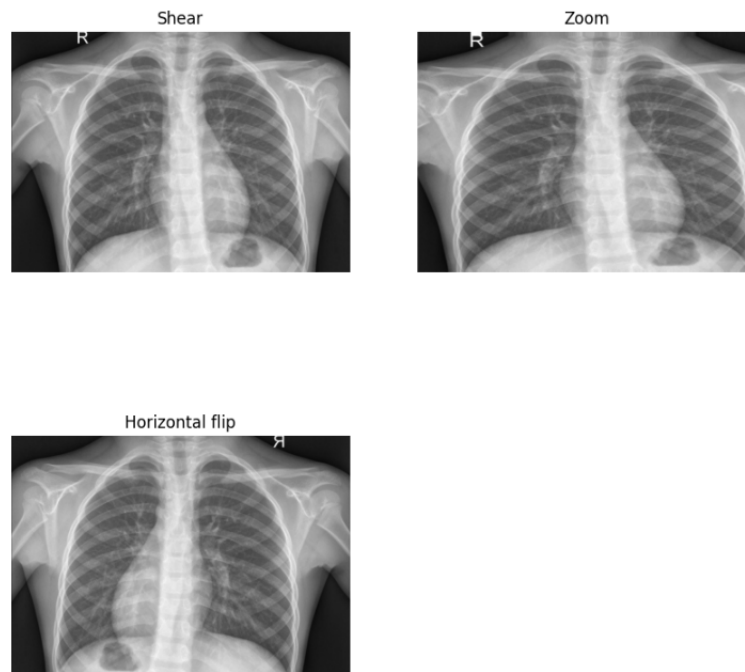
O pré-processamento de dados é uma etapa fundamental em qualquer projeto de aprendizado de máquina, especialmente em tarefas de visão computacional (WIKIPEDIA,

2022). Para este trabalho, várias etapas de pré-processamento foram realizadas para preparar as imagens para a etapa de modelagem.

Para evitar overfitting e melhorar a generalização do modelo, foi aplicado um aumento de dados no conjunto de treinamento. Utilizando a classe ImageDataGenerator do TensorFlow:

As imagens foram reescaladas para terem valores entre 0 e 1. Aplicou-se um range de cisalhamento (shear_range) de 0.2 para variar a forma da imagem, um zoom de até 20% (zoom_range), e também a virada das imagens horizontalmente (horizontal_flip) como ilustra a Figura 5. Para o conjunto de teste, apenas o reescalamento foi aplicado, pois, durante o teste, não queremos alterar ou aumentar os dados.

Figura 5 – Efeito das técnicas do ImageDataGenerator em uma imagem de raio-x.



Fonte: Elaboração própria.

Usando o ImageDataGenerator aprimorado, os conjuntos de treinamento e teste foram preparados. As imagens foram redimensionadas para 64x64 pixels e foram organizadas em lotes de 16 imagens para otimizar o processo de treinamento. A classificação binária foi escolhida para esta tarefa, pois existem apenas duas categorias: "Normal" e "Covid".

Após o uso do imagedatagenarator as imagem do datset tiveram um aumento considerável, o que influencia bastante na performance do modelo, por ter mais dados para treinamento.

Concluindo, as etapas de pré-processamento garantiram que as imagens estavam prontas e otimizadas para serem alimentadas na CNN para classificação.

3.3 Seleção de Hiperparâmetros

A otimização de hiperparâmetros desempenha um papel crucial no desenvolvimento de modelos de aprendizado profundo, uma vez que a escolha dos hiperparâmetros adequados pode ter um impacto significativo no desempenho do modelo. Essa etapa envolve a afinação de vários parâmetros para encontrar a combinação ideal que resulte no melhor desempenho do modelo.

Para conduzir a otimização de hiperparâmetros, foi empregado o método de pesquisa em grade (Grid Search), uma abordagem que explora várias combinações predefinidas de hiperparâmetros. O código completo utilizado para a sintonização dos hiperparâmetros está disponível no Apêndice C deste documento. O Apêndice C fornece um código detalhado, escrito em Python, que descreve a metodologia adotada para realizar o ajuste dos hiperparâmetros.

Para esta pesquisa, foram avaliadas diferentes combinações de três hiperparâmetros: número de épocas, tamanho do lote (batch size) e taxa de aprendizado. As combinações experimentadas foram:

- Número de épocas: 5, 10 e 15.
- Tamanho do lote: 16, 32 e 64.
- Taxa de aprendizado: 0.1, 0.01, 0.005, 0.001 e 0.0001.

Para cada combinação de hiperparâmetros, um modelo foi treinado utilizando o conjunto de treinamento e validado com o conjunto de teste. Durante esse processo, os resultados foram monitorados e registrados para análises posteriores.

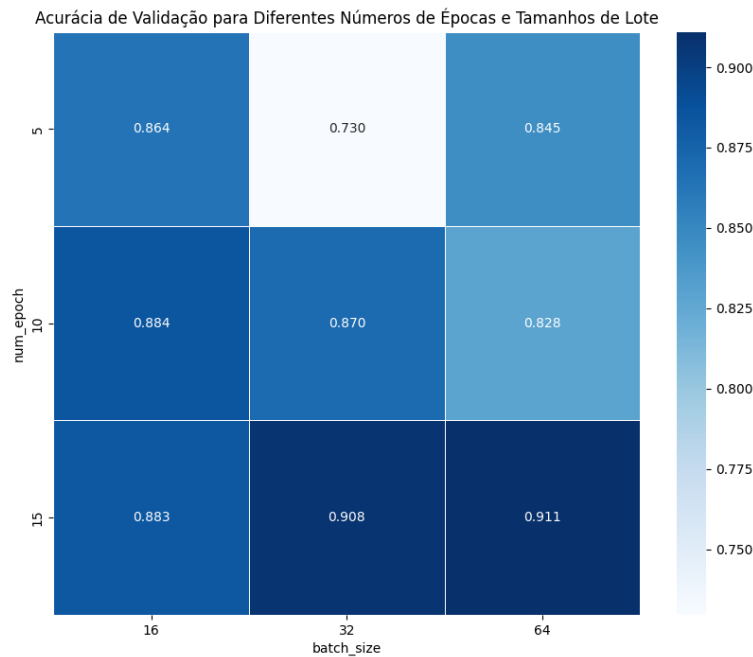
Ao final dos testes, a combinação que resultou na maior acurácia de validação foi identificada como a melhor configuração. Os hiperparâmetros otimizados foram os seguintes:

- Número de épocas: 15.
- Tamanho do lote: 16.
- Taxa de aprendizado: 0.001.

Esta combinação resultou em uma acurácia de treinamento de aproximadamente 92.19% e uma acurácia de validação de aproximadamente 96.88%, a combinação otimizada de hiperparâmetros foi, portanto, adotada na configuração final do modelo, garantindo que o modelo seja treinado de forma eficiente e generalize bem para dados não vistos.

Para ilustrar o desempenho das diferentes combinações de hiperparâmetros, um gráfico de calor foi gerado na Figura 6. Este gráfico mostra a acurácia de validação para cada combinação de número de épocas e tamanho do lote. As cores mais escuras indicam uma maior acurácia de validação.

Figura 6 – Gráfico de calor mostrando a acurácia de validação para diferentes números de épocas e tamanhos de lote.



Fonte: Elaboração própria.

Como pode ser visto na Figura 6, a combinação que resultou na maior acurácia de validação (representada pela cor mais escura) corresponde aos hiperparâmetros otimizados identificados.

3.4 Arquitetura do Modelo

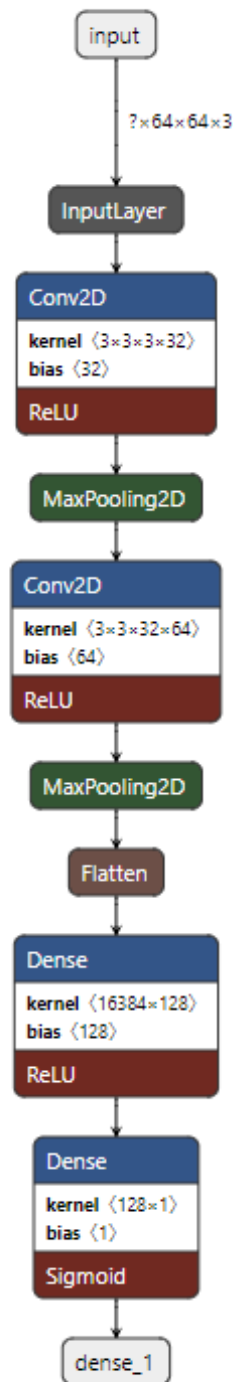
Nesta seção, descreverei em detalhes a arquitetura do modelo utilizado neste estudo, ilustrando-a com o diagrama gerado usando a biblioteca Netron do Python.

Antes de mergulharmos nos detalhes técnicos, é importante ter uma visão geral da arquitetura do modelo. O diagrama exposto na Figura 7, gerado com a biblioteca Netron, oferece uma representação visual da estrutura do modelo, suas camadas e conexões.

O modelo utilizado é uma CNN, uma arquitetura profunda amplamente empregada em tarefas de processamento de imagens. A arquitetura da CNN foi projetada especificamente para extrair características relevantes de imagens radiográficas do tórax.

O modelo inclui várias camadas convolucionais que aplicam filtros às imagens de entrada para detectar padrões e características específicas. As camadas convolucionais

Figura 7 – Diagrama do Modelo.



Fonte: Elaboração própria.

geralmente usam funções de ativação ReLU para introduzir não linearidade. Após cada camada convolucional, há camadas de pooling, geralmente MaxPooling, que reduzem a dimensionalidade da saída das camadas anteriores, preservando as características mais importantes.

A saída das camadas de pooling é achatada para criar um vetor unidimensional que servirá como entrada para as camadas densas. O modelo possui uma ou mais camadas densas, também conhecidas como totalmente conectadas, que realizam a classificação final com base nas características extraídas das camadas anteriores. A camada de saída utiliza uma função de ativação sigmoideal, adequada para problemas de classificação binária, e fornece a probabilidade de pertencimento a uma das classes ("Covid" ou "Normal").

3.5 Treinamento e Teste

Para a etapa de treinamento e teste do modelo de classificação de imagens de raio-x, foi utilizada uma Rede Neural Convolucional. Este modelo foi desenvolvido no ambiente do Google Colab e fez uso de diversas bibliotecas Python, tais como TensorFlow, NumPy e Matplotlib. O código referente a essa implementação pode ser encontrado no Apêndice D deste trabalho.

Após testes de hiperparâmetros, foi determinado que a melhor combinação é de 15 épocas, um tamanho de lote (batch size) de 16, e uma taxa de aprendizado de 0.001. Com essa configuração, foi alcançada uma acurácia de treinamento de 92.19% e uma acurácia de validação de 96.87%.

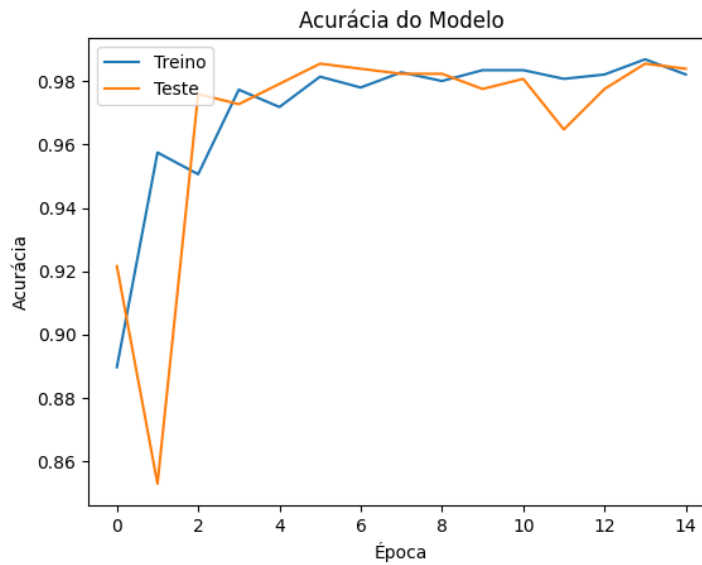
A arquitetura da CNN foi definida como:

- Uma camada convolucional com 32 filtros e função de ativação ReLU.
- Uma camada de pooling (MaxPooling).
- Uma camada convolucional com 64 filtros e função de ativação ReLU.
- Uma camada de pooling.
- Uma camada de achatamento (Flatten).
- Uma camada densa (Fully Connected) com 128 neurônios e função de ativação ReLU.
- Uma camada de saída com função de ativação sigmoideal, adequada para classificação binária.

A seguir, são exibidos gráficos que mostram o progresso do treinamento do modelo: A Figura 8 ilustra o gráfico da acurácia do modelo ao longo das épocas no conjunto de treinamento (linha azul) e no conjunto de teste (linha laranja). Observa-se que as duas curvas estão subindo e mantendo-se próximas uma da outra, o que indica que o modelo está aprendendo bem e generalizando para novos dados.

A Figura 9 contém o gráfico de perda que mostra como a perda do modelo se comportou durante o treinamento. É possível notar ambas as curvas de treinamento e teste estão

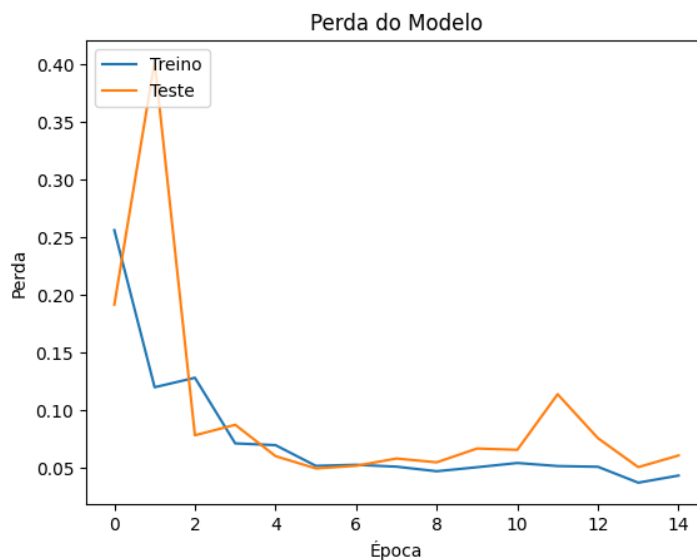
Figura 8 – Acurácia do Modelo.



Fonte: Elaboração própria.

diminuindo, o que significa que o modelo está ajustando seus pesos para minimizar a perda e, portanto, fazendo previsões mais precisas.

Figura 9 – Perda do Modelo.

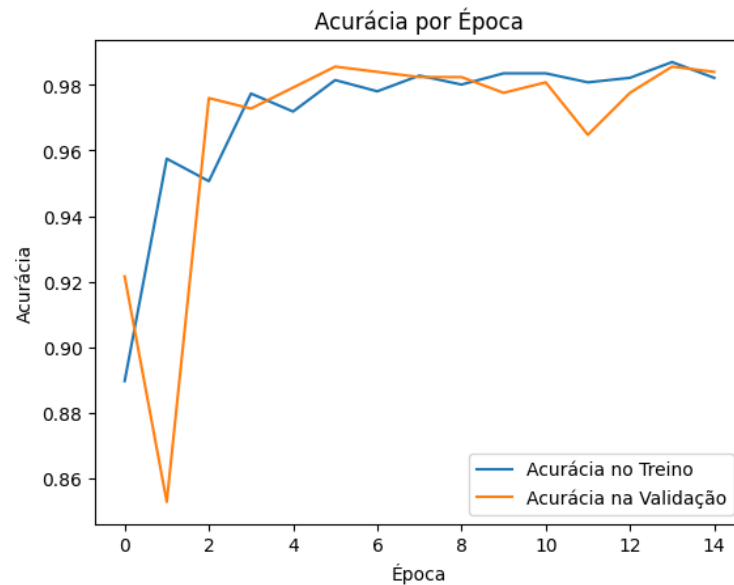


Fonte: Elaboração própria.

A Figura 10 exibe o gráfico que combina a acurácia de treinamento (linha azul) e validação (linha laranja) em um único gráfico, este gráfico é semelhante ao primeiro gráfico, mas mostra a acurácia em vez da perda. Isso nos proporciona que o modelo está aprendendo bem e generalizando para novos dados.

Esses gráficos são essenciais para visualizar o progresso do treinamento, a conver-

Figura 10 – Acurácia por Época.



Fonte: Elaboração própria.

gência do modelo e sua capacidade de generalização para o conjunto de teste. As legendas claras e interpretações relevantes destacam qualquer tendência notável ou comportamento do modelo durante o treinamento.

O modelo foi compilado utilizando o otimizador Adam com a taxa de aprendizado definida anteriormente. A função de perda utilizada foi a 'binary_crossentropy', adequada para problemas de classificação binária.

Os dados de treinamento e teste foram pré-processados utilizando o `ImageDataGenerator` do TensorFlow, que permite aumentar a quantidade de dados através de técnicas de data augmentation, como rotação, zoom e inversão horizontal. Além disso, as imagens foram redimensionadas para 64×64 pixels e normalizadas.

Com os dados preparados, o modelo foi treinado utilizando as 1459 imagens do conjunto de treinamento e validado com as 625 imagens do conjunto de teste. Ao término do treinamento, o modelo alcançou uma acurácia de 98.40% no conjunto de teste.

Após o treinamento, o modelo foi avaliado utilizando o conjunto de teste, onde foi possível observar uma perda de 0.0610 e uma acurácia de 98.40%. Adicionalmente, predições foram realizadas para cada imagem do conjunto de teste, possibilitando análises futuras, como a construção de uma matriz de confusão ou a análise da curva ROC.

Esse treinamento e teste fornecem uma base sólida para a utilização do modelo em cenários reais, dada a alta acurácia alcançada e a robustez do modelo em lidar com variações nas imagens.

Com o conjunto de dados preparado, esta etapa consiste na aplicação dos modelos,

para garantir que um modelo de aprendizado profundo seja eficaz em suas previsões em dados não vistos, é crucial passar por um processo rigoroso de treinamento e teste. No projeto em questão, ambas as etapas foram cuidadosamente configuradas e monitoradas.

Em resumo, o processo de treinamento e teste foi meticulosamente executado para garantir que o modelo não apenas se ajustasse bem aos dados de treinamento, mas também generalizasse bem para novos, dados não vistos. As várias métricas e visualizações usadas oferecem uma compreensão abrangente do desempenho do modelo.

Neste capítulo, mergulhou-se profundamente na metodologia utilizada na construção e treinamento do modelo de classificação de imagens para distinguir entre casos "Covid" e "Normal" em radiografias de tórax. Ao longo deste processo, várias etapas críticas foram abordadas com rigor e precisão.

Iniciou-se com a coleta e preparação dos dados, destacando a importância de um conjunto de dados de alta qualidade e o pré-processamento necessário para garantir que o modelo seja alimentado com informações relevantes e limpas. Isso incluiu a normalização das imagens, o tratamento de desequilíbrios de classe e a divisão adequada dos dados em conjuntos de treinamento e teste.

A seleção cuidadosa dos hiperparâmetros foi abordada na seção sobre a otimização de hiperparâmetros. Demonstrou-se como a escolha apropriada desses parâmetros pode ter um impacto significativo no desempenho do modelo. Através de experimentos exaustivos e a metodologia de pesquisa em grade, identificou-se a combinação ideal de hiperparâmetros que resultou em um modelo altamente eficiente.

Além disso, apresentou-se uma descrição detalhada da arquitetura da CNN, projetada para capturar efetivamente as características das imagens de raio-x. Foram destacadas a sequência de camadas, funções de ativação e outros componentes críticos que compõem o modelo.

Por fim, é importante notar que as escolhas metodológicas desempenharam um papel crucial nos resultados obtidos. A identificação da melhor combinação de hiperparâmetros e o design cuidadoso da arquitetura da CNN permitiram que o modelo alcançasse uma acurácia de validação de aproximadamente 96,88%. Esses resultados destacam a importância de seguir uma metodologia sólida e sistemática na construção de modelos de aprendizado profundo.

4 RESULTADOS

Este capítulo detalha o desempenho dos modelos de classificação utilizados nesta pesquisa, com base em diversas técnicas e métricas. As avaliações foram conduzidas com ferramentas do scikit-learn e da biblioteca TensorFlow.

4.1 Métricas de Desempenho

Foi utilizado o método *evaluate* da biblioteca TensorFlow para obter as métricas de perda e acurácia. Este método é comumente usado após o treinamento do modelo para avaliar o desempenho do modelo em um conjunto de dados de teste, o mesmo retorna a perda de teste e as métricas de teste calculadas pelo modelo nos dados de teste (TensorFlow newsletter, 2023).

No contexto deste trabalho, o modelo apresentou uma perda (loss) de 0.0610 no conjunto de teste. A perda é uma medida de quão bem o modelo conseguiu aprender os padrões nos dados de treinamento (TensorFlow newsletter, 2023). Um valor menor indica que o modelo fez previsões mais próximas dos valores reais (TensorFlow newsletter, 2023). Neste caso, o valor de 0.0610 significa que, em média, o modelo estava errado por esse valor ao realizar suas previsões.

Além da perda, a acurácia é outra métrica importante retornada pelo método *evaluate*, a acurácia é a proporção de previsões corretas feitas pelo modelo em relação ao total de previsões (TensorFlow newsletter, 2023). Neste trabalho, a acurácia obtida foi de 98.4%, indicando que o modelo acertou as previsões em 98.4% das ocasiões no conjunto de teste.

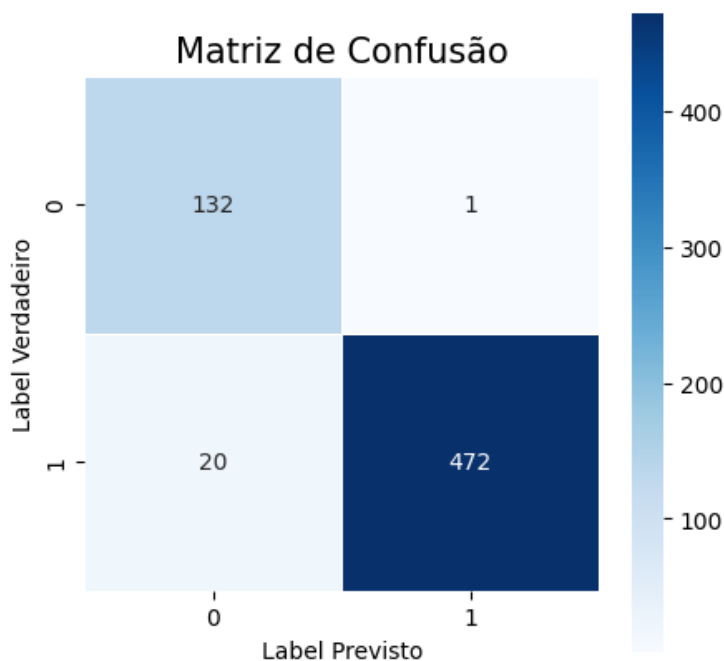
Esses resultados demonstram que o modelo foi eficaz em aprender os padrões dos dados de treinamento e generalizar para dados não vistos, também sugere que o modelo pode ser útil para fazer previsões precisas em novos dados no futuro (TensorFlow newsletter, 2023).

4.2 Matriz de Confusão

A matriz de confusão é uma ferramenta poderosa que permite a visualização do desempenho de um algoritmo de classificação (WIKIPEDIA, 2023). Cada linha da matriz representa instâncias de uma classe prevista, enquanto cada coluna representa instâncias da classe real (ou vice-versa) (WIKIPEDIA, 2023). É uma forma especial de tabela de contingência, com duas dimensões (“real” e “prevista”), e conjuntos de “classes” idênticos em ambas as dimensões (WIKIPEDIA, 2023).

No contexto deste trabalho, a matriz de confusão, representada na Figura 11, foi calculada e visualizada usando a biblioteca `sklearn.metrics` para a função `confusion_matrix` e revela que o modelo fez as seguintes previsões: acertou 132 diagnósticos de Covid e 472 de Normal; errou ao prever 20 casos de Covid e um de Normal. Isso sugere que o modelo foi eficaz em distinguir entre as duas classes.

Figura 11 – Matriz de confusão.



Fonte: Elaboração própria.

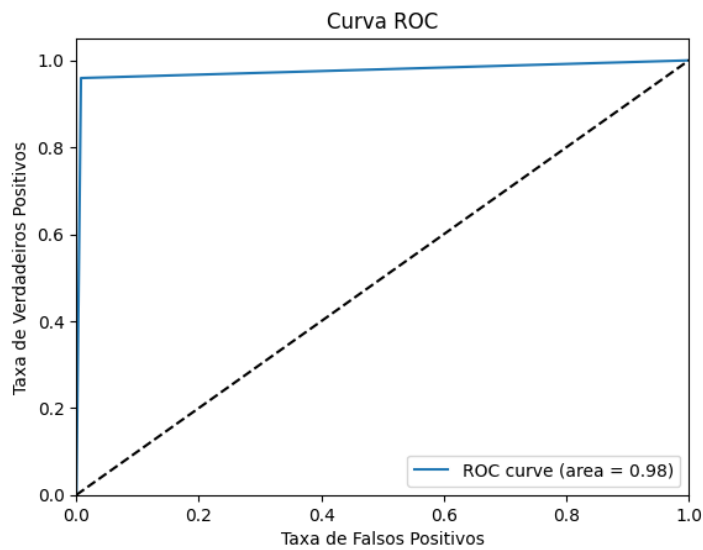
4.3 Análise da Curva ROC

A métrica AUC-ROC, que é uma forma de avaliar o desempenho de um modelo de classificação binária, foi aplicada para medir a capacidade do modelo em distinguir as classes. A AUC-ROC mostra a relação entre a taxa de verdadeiros positivos (TPR) e a taxa de falsos positivos (FPR) do modelo em diferentes limiares de probabilidade. Quanto maior a área sob a curva ROC, melhor o modelo (Vinicius Rodrigues, 2018). No contexto deste trabalho, a AUC-ROC alcançou o valor de 0.9759, aproximando-se bastante de 1. Isso evidencia a excelente capacidade discriminativa do modelo. Um valor de AUC-ROC igual a 1 indica que o modelo tem um poder discriminatório perfeito, ou seja, ele consegue separar as classes sem erros. Portanto, um valor de 0.9759 sugere que o modelo tem um alto poder discriminatório e é capaz de fazer uma distinção clara entre as classes.

A Curva Característica de Operação do Receptor (ROC) é uma ferramenta crucial para entender o desempenho de modelos de classificação. Essa curva representa a taxa de verdadeiros positivos versus a taxa de falsos positivos para o modelo. A Área Sob

a Curva (AUC) é uma métrica que resume a performance global de um classificador binário, variando de 0 a 1, onde 1 representa um classificador perfeito e 0.5 um classificador aleatório (IBM Documentation Help, 2023). Como mostra a Figura 12, a posição da curva em relação à diagonal indica que o modelo tem uma alta taxa de verdadeiros positivos enquanto mantém baixa a taxa de falsos positivos. Esse é um indicativo de um modelo de classificação de alta qualidade, especialmente quando consideramos a importância de minimizar falsos negativos em um contexto de diagnóstico médico.

Figura 12 – Curva ROC.



Fonte: Elaboração própria.

4.4 Relatório de Classificação

Outra ferramenta empregada foi o *classification_report*, uma função útil do pacote *sklearn.metrics* na biblioteca *scikit-learn* do Python, usada para construir um relatório de texto mostrando as principais métricas de classificação que avalia a qualidade das previsões do modelo (scikit-learn, 2023), conforme a Figura 13.

Figura 13 – Relatório de classificação do modelo.

	precision	recall	f1-score	support
0	0.99	0.87	0.93	152
1	0.96	1.00	0.98	473
accuracy			0.97	625
macro avg	0.98	0.93	0.95	625
weighted avg	0.97	0.97	0.97	625

Fonte: Elaboração própria.

O relatório apresenta métricas como precisão, recall, f1-score e support, detalhadas a seguir de acordo com (scikit-learn, 2023):

- **Precisão:** Refere-se à razão entre os verdadeiros positivos e a soma de verdadeiros e falsos positivos. Indica a habilidade do classificador em não rotular como positiva uma amostra negativa.
- **Recall:** Relação entre os verdadeiros positivos e a soma de verdadeiros positivos e falsos negativos. Representa a habilidade do modelo em identificar todas as amostras positivas.
- **F1-score:** Média harmônica ponderada entre precisão e recall. Seu melhor valor é 1 e o pior é 0.
- **Support:** Número de ocorrências de cada classe no conjunto real.

4.5 Avaliação Comparativa dos Resultados

A fim de avaliar o desempenho do nosso modelo em relação a trabalhos correlatos na detecção da COVID-19 em imagens de raio-x do tórax, realizamos uma comparação com os estudos mencionados na seção de Trabalhos Correlatos (seção 3.5). Os principais resultados obtidos em nosso estudo foram confrontados com as métricas alcançadas por esses trabalhos. A tabela a seguir apresenta essa comparação:

Tabela 3 – Comparação de Resultados com Trabalhos Correlatos

Trabalho	Acurácia (%)
Este Estudo	98.40
(RESHI <i>et al.</i> , 2021)	100.00
(OHATA <i>et al.</i> , 2020)	95.00
(BARBOSA; COELHO; BAFFA, 2021)	99.36
(CHRISTOFOLLETTI; ATTUX, 2022)	90.26
ResNetX50 (SILVA <i>et al.</i> , 2020)	99.33
InceptionResNetV2 (SILVA <i>et al.</i> , 2020)	98.00

Os resultados obtidos neste estudo demonstram uma alta acurácia de 98.4% na detecção de COVID-19 em imagens de raio-x do tórax. Ao compararmos esses resultados com os trabalhos correlatos, observamos que nosso modelo se aproxima do desempenho alcançado por (RESHI *et al.*, 2021) com 100.0% de acurácia e supera o estudo de (OHATA *et al.*, 2020) que obteve 95.0% de acurácia. Além disso, nossa abordagem se mostrou mais eficiente que a otimização computacional proposta por (BARBOSA; COELHO; BAFFA, 2021) com uma acurácia de 99.36%.

Além disso, quando comparamos nossos resultados com as arquiteturas ResNetX50 e InceptionResNetV2, conforme apresentado na tabela, observamos que nosso modelo

obteve uma acurácia de 98.4%, superando a InceptionResNetV2 com 98.00% de acurácia e se aproximando do desempenho da ResNetX50, que alcançou 99.33% de acurácia.

5 CONCLUSÃO

A detecção precisa e eficaz da Covid-19 por meio de imagens é de suma importância no combate à pandemia, dado que uma identificação rápida pode levar a intervenções médicas mais céleres. Este estudo empregou técnicas de aprendizado profundo, especificamente uma Rede Neural Convolucional, para classificar imagens em duas categorias: "Normal" e "Covid".

A implementação da CNN, conforme detalhado no código, revelou-se robusta. Ao otimizar hiperparâmetros, chegou-se a uma configuração com 15 épocas, um tamanho de lote de 16 e uma taxa de aprendizado de 0.001, resultando em uma acurácia de treinamento de 0.921875 e uma acurácia de validação de 0.96875.

A análise subsequente das métricas, incluindo a matriz de confusão, sensibilidade, especificidade e a AUC-ROC, forneceu insights adicionais sobre o desempenho do modelo. A AUC-ROC, por exemplo, é uma indicação clara da capacidade do modelo em diferenciar entre as categorias de imagem.

Este trabalho ressalta a viabilidade e eficácia de se usar CNNs na análise de imagens para detecção da Covid-19. Atingir acurácias tão altas em um cenário real tem implicações potenciais para aplicações clínicas, onde a detecção rápida e precisa pode fazer uma diferença significativa no tratamento e isolamento dos pacientes.

O desempenho do modelo em uma ampla gama de dados deve ser testado para garantir sua aplicabilidade. No entanto, um obstáculo significativo deste estudo é a quantidade limitada de dados. Para aumentar a precisão e a resiliência do modelo, a base de dados deverá ser ampliada. Independentemente do seu sucesso no presente conjunto de testes, o desempenho do modelo deve ser avaliado em dados variados e extras.

Para estudos futuros, sugere-se a exploração de arquiteturas de redes mais complexas, assim como a integração de técnicas adicionais de pré-processamento e aumento de dados. A investigação de abordagens de transferência de aprendizado, onde modelos pré-treinados em grandes datasets são finamente ajustados para tarefas específicas, também pode ser proveitosa.

A implementação bem-sucedida e avaliação da CNN para classificar imagens em Normal ou Covid sublinha o potencial das técnicas de aprendizado profundo na área da saúde. Espera-se que, à medida que a tecnologia avança e os conjuntos de dados crescem, modelos como o proposto aqui possam ser ainda mais refinados, contribuindo significativamente para o combate a doenças e pandemias.

REFERÊNCIAS

ANDRADE, G. Rt-pcr ou sorológico? entenda as diferenças entre os testes para a covid-19. 2020. Available at: <https://www.medicina.ufmg.br/rt-pcr-ou-sorologico-entenda-as-diferencas-entre-os-testes-para-a-covid-19/>.

Avijeet Biswal. Top 25 deep learning applications used across industries. 2023. Available at: <https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learning-applications>.

AWARI. Redes neurais convolucionais: Como elas funcionam. 2023. Available at: https://awari.com.br/redes-neurais-convolucionais/?utm_source=blog&utm_campaign=projeto+blog&utm_medium=Redes%20Neurais%20Convolucionais:%20Como%20elas%20Funcionam.

AYRES, F. Da análise de imagens ao reconhecimento de falas, as redes neurais têm vasta aplicação na vida moderna. 2021. Acessado em: 25 setembro 2023. Available at: <https://www.insper.edu.br/noticias/da-analise-de-imagens-ao-reconhecimento-de-falas-as-redes-neurais-tem-vasta-aplicacao-na-vida-moderna>.

BARBOSA, F. J.; COELHO, A. M.; BAFFA, M. F. O. Uma arquitetura de rede neural convolucional simplificada para reconhecimento da covid-19 em imagens de raios-x. 2021. Available at: <https://sol.sbc.org.br/index.php/ercas/article/view/17428/17264>.

BARBOSA GOVINDA MOHINI G. BEZERRA, D. S. V. d. M. M. A. L. D. M. F. M. G. N. N. Segurança em redes 5g: Oportunidades e desafios em detecção de anomalias e predição de tráfego baseadas em aprendizado de máquina. **XXI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais**, 2021.

BISONG, E. Google colab. *In*: _____. **Building Machine Learning and Deep Learning Models on Google Cloud Platform**. Apress, 2019. p. 59–64. Available at: https://doi.org/10.1007/978-1-4842-4470-8_7.

BRAGA, A. d. P.; LUDERMIR, T. B.; CARVALHO, A. C. P. d. L. F. **Redes Neurais Artificiais: Teoria e Aplicações**. 2. ed. [S.l.: s.n.]: LTC, 2007. 248 p. ISBN 9788521615644.

Center for Disease Control and Prevention. How to protect yourself & others. 2023. Available at: <https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/prevention.html>.

CHRISTOFOLLETTI, J. P.; ATTUX, R. R. de F. Redes neurais e detecção de covid-19 a partir de imagens de raios-x do tórax. 2022. Acessado em: 13 abril 2023. Available at: <https://www.prp.unicamp.br/inscricao-congresso/resumos/2022P19679A36214O568.pdf>.

COHEN, J. P. *et al.* Covid-19 image data collection: Prospective predictions are the future. **arXiv preprint arXiv:2006.11988**, 2020. Available at: <https://github.com/ieee8023/covid-chestxray-dataset>.

CONTROL, C. for D.; PREVENTION. Symptoms. 2022. Available at: <https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html>.

Data Science Academy. Top 25 bibliotecas python para data science. 2023. Available at: https://blog.dsacademy.com.br/top-25-bibliotecas-python-para-data_science/.

eDialog. Inteligência artificial na medicina: 8 aplicações e seus benefícios. 2023. Available at: <https://maislaudo.com.br/blog/inteligencia-artificial-na-medicina/>.

Fenestra . Redes neurais convolucionais: O que são e como funcionam? 2023. Available at: <https://fenestra.com.br/blog/2023/04/08/redes-neurais-convolucionais-o-que-sao-e-como-funcionam/>.

Fleck, Leandro *et al.* Redes neurais artificiais: principios basicos. **Revista Eletrônica Científica Inovação e Tecnologia**, 1, n. 13, p. 47–57, 2016. Available at: {<https://periodicos.utfpr.edu.br/recit/article/view/433>}.

FLOREZ YESENIA CONCHA-RAMOS, F. P.-Q. R. J. C.-C. R. Cnn for the detection of covid-19 from chest x-ray images. 2022. Available at: <https://ieeexplore.ieee.org/document/9934820>.

GOODFELLOW YOSHUA BENGIO, A. C. I. **Deep Learning**. MIT Press, 2016. Available at: <http://www.deeplearningbook.org>.

HAYKIN, S. **Redes Neurais: Principios e Pratica**. 2. ed. [*S.l.: s.n.*]: Bookman, 2001.

Hewings-Martin, Yella. How do covid-19 vaccines work? **Medical News Today**, 2021. Available at: https://www.medicalnewstoday.com/articles/how-do-covid-19-vaccines-work?utm_source=ReadNext.

HOSNY MOHAMED M. DARWISH, K. L. A. S. K. M. Covid-19 diagnosis from ct scans and chest x-ray images using low-cost raspberry pi. 2021. Available at: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0250688>.

IBM. O que são redes neurais convolucionais? 2023. Available at: <https://www.ibm.com/br-pt/topics/convolutional-neural-networks>.

IBM. O que é machine learning? 2023. Available at: <https://www.ibm.com/br-pt/topics/machine-learning>.

IBM. O que é o Deep Learning? 2023. Available at: {<https://www.ibm.com/br-pt/topics/deep-learnin>}.

IBM Documentation Help. Analise de caracteristica de operação do receptor. 2023. Available at: <https://www.ibm.com/docs/pt-br/spss-statistics/29.0.0?topic=features-roc-analysis>.

ICMC. Redes neurais artificiais. 2023. Available at: <https://sites.icmc.usp.br/andre/research/neural/>.

ICRX Solucion. Raio x do tÓrax: O que analisar e para que serve? 2023. Available at: <https://www.icrx.com.br/blog/torax>.

JAMDADÉ, V. Cmsc-678-ml-project. 2020. Available at: <https://github.com/vj2050/Transfer-Learning-COVID-19>.

José Aldair Morsch . Inteligência artificial na medicina: 7 aplicações e benefícios. 2021. Available at: <https://telemedicinamorsch.com.br/blog/inteligencia-artificial-na-medicina>.

Kellison Ferreira. Deep learning: a tendência de inteligência artificial que copia um cérebro humano. 2019. Available at: <https://rockcontent.com/br/blog/deep-learning/>.

KERMANY, D. S. *et al.* Identifying medical diagnoses and treatable diseases by image-based deep learning. **Cell**, v. 172, p. 889–1136, 2018. Available at: [https://www.cell.com/cell/issue?pii=S0092-8674\(17\)X0005-6](https://www.cell.com/cell/issue?pii=S0092-8674(17)X0005-6).

MAHASE, E. Coronavirus: Covid-19 has killed more people than sars and mers combined, despite lower case fatality rate. **BJM**, 2020. Available at: <https://doi.org/10.1136/bmj.m641>.

Marina Chatterjee. Top 20 applications of deep learning in 2022 across industries. 2022. Available at: <https://www.mygreatlearning.com/blog/deep-learning-applications/>.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the idea immanent in nervous activity. **Bulletin of Mathematical Biophysics**, v. 5, p. 115–133, 1943. Available at: <https://www.cse.chalmers.se/~coquand/AUTOMATA/mcp.pdf>.

Medicine Cursos. Inteligência artificial na medicina: entenda sua importância e como utilizar. 2023. Available at: <https://medicinecursos.com.br/blog/inteligencia-artificial-na-medicina/>.

NORVIG, P.; RUSSELL, S. **Inteligência Artificial**. 3. ed. [S.l.: s.n.]: GEN LTC, 2013.

OHATA, E. F. *et al.* Automatic detection of covid-19 infection using chest x-ray images through transfer learning. **IEEE/CAA JOURNAL OF AUTOMATICA SINICA**, v. 8, n. 1, p. 239–248, 2020. Available at: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9205687>.

OLIVEIRA, C. V. d. Coronavirus: uso de tomografia computadorizada na detecção. 2020. Available at: <https://brasilrad.com.br/artigos/coronavirus-uso-de-tomografia-computadorizada-na-deteccao/>.

OMS. Coronavirus disease (covid-19) pandemic. 2023. Available at: <https://www.who.int/europe/emergencies/situations/covid-19>.

OPAS-W. Uso de exames de imagem de tórax na covid-19. 2020. Available at: https://iris.paho.org/bitstream/handle/10665.2/52316/OPASWBRACOV1920079_por.pdf.

Quigo . Entenda o que É machine learning e as suas aplicações. 2022. Available at: <https://quigo.com.br/blog/entenda-o-que-e-machine-learning-e-as-suas-aplicacoes/>.

RESHI, A. A. *et al.* An efficient cnn model for covid-19 disease detection based on x-ray image classification. 2021. Available at: <https://www.hindawi.com/journals/complexity/2021/6621607/>.

RESPOSTAS, T. A. Quais os 3 tipos de machine learning? 2022. Available at: <https://todasasrespostas.pt/quais-os-3-tipos-de-machine-learning>.

Sarah Madeleine . Apresentação da rede neural convolucional aplicadas a imagens médicas. 2021. Available at: <https://www.imaio.com.br/recursos/blog/classificacao-de-imagens-medicas-entendendo-a-rede-neural-convolucional-cnn>.

SARKI KHANDAKAR AHMED, H. W. Y. Z. K. W. R. Automated detection of covid-19 through convolutional neural network using chest x-ray images. 2022. Available at: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0262052>.

scikit-learn. *sklearn.metrics.classification_report*. 2023. Available at : .

SILVA, L. *et al.* Aplicação de deep learning no pré-diagnóstico da covid-19 através de imagens de raio-x. 2020. Acessado em: 13 abril de 2023. Available at: <https://encurtador.com.br/DIJR9>.

souto, Xênia Macedo. Covid-19: Aspectos gerais e implicações globais. **RECITAL - Revista de Educação, Ciência e Tecnologia de Almenara**, v. 2, p. 2674–9270, 2020. Available at: <https://recital.almenara.ifnmg.edu.br/index.php/recital/article/view/90/37>.

TAULLI, T. **Introdução à Inteligência Artificial: uma Abordagem Não Técnica**. 1. ed. [S.l.: s.n.]: Novatec Editora, 2020.

TensorFlow newsletter. Training evaluation with the built-in methods. 2023. Available at: https://www.tensorflow.org/guide/keras/training_with_built_in_methods.

Vinicius Rodrigues . Entenda o que é auc e roc nos modelos de machine learning. 2018. Available at: [https://medium.com/bio-data-blog/entenda-o-que-%C3%A9-auc-e-roc-nos-modelos-de-machine-le](https://medium.com/bio-data-blog/entenda-o-que-%C3%A9-auc-e-roc-nos-modelos-de-machine-learning)

Você pergunta . Como descrever raio x de tórax normal? 2021. Available at: <https://vocepergunta.com/library/artigo/read/221822-como-descrever-raio-x-de-torax-normal>.

WIKIPEDIA. Pré-processamento de dados. 2022. Available at: https://pt.wikipedia.org/wiki/Pr%C3%A9-processamento_de_dados.

WIKIPEDIA. Matriz de confusão. 2023. Available at: https://pt.wikipedia.org/wiki/Matriz_de_confus%C3%A3o.

APÊNDICES

APÊNDICE A – CÓDIGO DE EXTRAÇÃO DOS DADOS

```
1 import os
2 import shutil
3 import hashlib
4 import pandas as pd
5 from google.colab import drive
6
7 drive.mount('/content/drive')
8
9 !git clone https://github.com/ieee8023/covid-chestxray-dataset
10
11 data = pd.read_csv('/content/covid-chestxray-dataset/metadata.csv',)
12
13 print("Columns:", data.columns)
14 print("Unique 'folder':", data['folder'].unique())
15 print("Unique 'finding':", data['finding'].unique())
16 print("Unique 'modality':", data['modality'].unique())
17
18 covid_images = data[data['finding'] == 'Pneumonia/Viral/COVID-19']
19 print("Número de imagens filtradas como COVID-19:", len(covid_images))
20
21 covid_xray_images = data[(data['finding'] == 'Pneumonia/Viral/COVID-19') & (data['modality'] == 'X-ray') & (data['folder'] == 'images')]
22 print("Número de imagens de raio-X filtradas como COVID-19:", len(covid_xray_images))
23
24 print("View counts:", data[data['finding'] == 'Pneumonia/Viral/COVID-19']['view'].value_counts())
25
26 def calculate_md5(file_path):
27     hasher = hashlib.md5()
28     with open(file_path, 'rb') as f:
29         buf = f.read()
30         hasher.update(buf)
31     return hasher.hexdigest()
```

```

32
33 def is_duplicate_image(source_image_path, destination_directory):
34     source_hash = calculate_md5(source_image_path)
35     for image_name in os.listdir(destination_directory):
36         dest_image_path = os.path.join(destination_directory,
37                                         image_name)
38         if source_hash == calculate_md5(dest_image_path):
39             return True
40     return False
41
42 def ensure_directory(directory_path):
43     if not os.path.exists(directory_path):
44         os.makedirs(directory_path)
45     else:
46         print(f"Directory {directory_path} exists!")
47
48 destination_directory = '/content/drive/MyDrive/CNN_COVID/
49 dados_fonte/covid_total/'
50 ensure_directory(destination_directory)
51
52 for image in covid_xray_images:
53     source_path = '/content/covid-chestxray-dataset/images/' +
54     image
55     dest_path = destination_directory + image
56     if not is_duplicate_image(source_path, destination_directory):
57         :
58         shutil.copy(source_path, dest_path)
59
60 print("Número de imagens no diretório:", len(os.listdir(
61     destination_directory)))
62
63 !git clone https://github.com/vj2050/Transfer-Learning-COVID-19.
64 git
65
66 paths = {
67     'covid': ['/content/Transfer-Learning-COVID-19/
68 dataset_3_classes/train/covid',
69             '/content/Transfer-Learning-COVID-19/
70 dataset_3_classes/test/covid'],
71     'normal': ['/content/Transfer-Learning-COVID-19/
72 dataset_3_classes/train/normal',
73              '/content/Transfer-Learning-COVID-19/

```

```
dataset_3_classes/test/normal']
65 }
66
67 destinations = {
68     'covid': '/content/drive/MyDrive/CNN_COVID/dados_fonte/
covid_total/',
69     'normal': '/content/drive/MyDrive/CNN_COVID/dados_fonte/
normal_total/'
70 }
71
72 for label, path_list in paths.items():
73     dest = destinations[label]
74     ensure_directory(dest)
75     for path in path_list:
76         for image in os.listdir(path):
77             source_path = os.path.join(path, image)
78             dest_path = os.path.join(dest, image)
79             if not is_duplicate_image(source_path, dest):
80                 shutil.copy(source_path, dest_path)
81         print(f"Total {label} images:", len(os.listdir(dest)))
82
83 for label, dest in destinations.items():
84     print(f"Total {label} images:", len(os.listdir(dest)))
```


APÊNDICE B – CÓDIGO DE DIVISÃO DOS DADOS

```
1 from google.colab import drive
2 import os
3 import shutil
4 import numpy as np
5
6 drive.mount('/content/drive')
7
8 covid_folder = '/content/drive/MyDrive/CNN_COVID/dados_fonte/
   covid_total/'
9 normal_folder = '/content/drive/MyDrive/CNN_COVID/dados_fonte/
   normal_total/'
10
11 train_covid_folder = '/content/drive/MyDrive/CNN_COVID/
   Dados_divididos/train/Covid'
12 test_covid_folder = '/content/drive/MyDrive/CNN_COVID/
   Dados_divididos/test/Covid'
13 train_normal_folder = '/content/drive/MyDrive/CNN_COVID/
   Dados_divididos/train/Normal'
14 test_normal_folder = '/content/drive/MyDrive/CNN_COVID/
   Dados_divididos/test/Normal'
15
16 os.makedirs(train_covid_folder, exist_ok=True)
17 os.makedirs(test_covid_folder, exist_ok=True)
18 os.makedirs(train_normal_folder, exist_ok=True)
19 os.makedirs(test_normal_folder, exist_ok=True)
20
21 def split_data(src_folder, train_folder, test_folder, train_ratio
   =0.7):
22     all_files = os.listdir(src_folder)
23     np.random.shuffle(all_files)
24     train_count = int(len(all_files) * train_ratio)
25     train_files = all_files[:train_count]
26     test_files = all_files[train_count:]
27     for file in train_files:
28         shutil.copy(os.path.join(src_folder, file), train_folder)
29     for file in test_files:
30         shutil.copy(os.path.join(src_folder, file), test_folder)
31
```

```
32 split_data(covid_folder, train_covid_folder, test_covid_folder)
33 split_data(normal_folder, train_normal_folder, test_normal_folder
   )
34
35 num_train_covid = len(os.listdir(train_covid_folder))
36 num_test_covid = len(os.listdir(test_covid_folder))
37 num_train_normal = len(os.listdir(train_normal_folder))
38 num_test_normal = len(os.listdir(test_normal_folder))
39
40 print(f"Número de imagens de treino com Covid: {num_train_covid}"
   )
41 print(f"Número de imagens de teste com Covid: {num_test_covid}")
42 print(f"Número de imagens de treino Normais: {num_train_normal}")
43 print(f"Número de imagens de teste Normais: {num_test_normal}")
44
45 total_images = num_train_covid + num_test_covid +
   num_train_normal + num_test_normal
46 print(f"Total de imagens: {total_images}")
```

APÊNDICE C – CÓDIGO DE AJUSTE DOS HIPERPARÂMETROS

```

1 from google.colab import drive
2 import tensorflow as tf
3 from tensorflow.keras.callbacks import TensorBoard
4 from tensorflow.keras.preprocessing.image import
    ImageDataGenerator
5 import time
6
7 drive.mount('/content/drive')
8
9 def create_model():
10     classifier = tf.keras.models.Sequential()
11     classifier.add(tf.keras.layers.Convolution2D(filters=32,
12         kernel_size=3, padding="same", input_shape=(64, 64, 3),
13         activation='relu'))
14     classifier.add(tf.keras.layers.MaxPooling2D(pool_size=2,
15         strides=2, padding='valid'))
16     classifier.add(tf.keras.layers.Convolution2D(filters=64,
17         kernel_size=3, padding="same", activation="relu"))
18     classifier.add(tf.keras.layers.MaxPooling2D(pool_size=2,
19         strides=2, padding='valid'))
20     classifier.add(tf.keras.layers.Flatten())
21     classifier.add(tf.keras.layers.Dense(units=128, activation='
22         relu'))
23     classifier.add(tf.keras.layers.Dense(units=1, activation='
24         sigmoid'))
25     return classifier
26
27 train_datagen = ImageDataGenerator(rescale=1. / 255, shear_range
28     =0.2, zoom_range=0.2, horizontal_flip=True)
29 test_datagen = ImageDataGenerator(rescale=1. / 255)
30
31 training_set = train_datagen.flow_from_directory('/content/drive/
32     MyDrive/CNN_COVID/Dados_divididos/train', target_size=(64, 64),
33     batch_size=32, class_mode='binary')
34 test_set = test_datagen.flow_from_directory('/content/drive/
35     MyDrive/CNN_COVID/Dados_divididos/test', target_size=(64, 64),
36     batch_size=32, class_mode='binary')

```

```

26 log = []
27 num_epochs = [5, 10, 15]
28 batch_sizes = [16, 32, 64]
29 learning_rates = [0.1, 0.01, 0.005, 0.001, 0.0001]
30
31 for num_epoch in num_epochs:
32     for batch_size in batch_sizes:
33         for learning_rate in learning_rates:
34             NAME = f"epochs-{num_epoch}-batch_size-{batch_size}-
learning_rate-{learning_rate}-{int(time.time())}"
35             tensorboard = TensorBoard(log_dir=f'logs/{NAME}')
36
37             classifier = create_model()
38             classifier.compile(optimizer=tf.keras.optimizers.Adam
(learning_rate=learning_rate), loss='binary_crossentropy',
metrics=['accuracy'])
39
40             history = classifier.fit(training_set,
steps_per_epoch=4, epochs=num_epoch, validation_data=test_set,
validation_steps=4, callbacks=[tensorboard])
41
42             log_entry = {'num_epoch': num_epoch, 'batch_size':
batch_size, 'learning_rate': learning_rate, '
final_train_accuracy': history.history['accuracy'][-1], '
final_val_accuracy': history.history['val_accuracy'][-1]}
43             log.append(log_entry)
44             print(f"Finished training with {log_entry}")
45
46 best_combination = max(log, key=lambda x: x['final_val_accuracy'
])
47 print(f"Melhor combinação:  pocas : {best_combination['num_epoch'
']}, Tamanho do Lote: {best_combination['batch_size']}, Taxa de
Aprendizado: {best_combination['learning_rate']}, Acurácia de
Treinamento: {best_combination['final_train_accuracy']}, Acurá
cia de Validação: {best_combination['final_val_accuracy']}")
48
49 sorted_log = sorted(log, key=lambda x: x['final_val_accuracy'],
reverse=True)
50 for entry in sorted_log:
51     print(f"  pocas : {entry['num_epoch']}, Tamanho do Lote: {
entry['batch_size']}, Taxa de Aprendizado: {entry['
learning_rate']}, Acurácia de Treinamento: {entry['

```

```
final_train_accuracy']:.2f}, Acurácia de Validação: {entry['  
final_val_accuracy']:.2f})
```


APÊNDICE D – CÓDIGO DE TREINAMENTO E TESTE DO MODELO

```

1 from google.colab import drive
2 import os
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import seaborn as sns
6 import tensorflow as tf
7 from sklearn.metrics import roc_auc_score, confusion_matrix,
   roc_curve, classification_report
8 from tensorflow.keras.preprocessing.image import
   ImageDataGenerator
9 from tensorflow.keras.preprocessing import image
10 import datetime
11 from PIL import Image
12
13 drive.mount('/content/drive')
14
15 def create_model():
16     classifier = tf.keras.Sequential()
17     classifier.add(tf.keras.layers.Conv2D(32, 3, padding="same",
18     input_shape=(64, 64, 3), activation='relu'))
19     classifier.add(tf.keras.layers.MaxPooling2D())
20     classifier.add(tf.keras.layers.Conv2D(64, 3, padding="same",
21     activation="relu"))
22     classifier.add(tf.keras.layers.MaxPooling2D())
23     classifier.add(tf.keras.layers.Flatten())
24     classifier.add(tf.keras.layers.Dense(128, activation='relu'))
25     classifier.add(tf.keras.layers.Dense(1, activation='sigmoid'))
26     return classifier
27
28 BASE_PATH = '/content/drive/MyDrive/CNN_COVID/Dados_divididos'
29 train_dir = os.path.join(BASE_PATH, 'train')
30 test_dir = os.path.join(BASE_PATH, 'test')
31
32 def count_images(directory):
33     return sum([len(files) for _, _, files in os.walk(directory)
34     ])

```

```
33 print(f"O diretório de treinamento contém {count_images(train_dir  
    )} imagens.")
34 print(f"O diretório de teste contém {count_images(test_dir)}  
    imagens.")
35
36 classifier = create_model()
37 optimizer = tf.keras.optimizers.Adam(learning_rate=0.001)
38 classifier.compile(optimizer=optimizer, loss='binary_crossentropy  
    ', metrics=['accuracy'])
39
40 train_datagen = ImageDataGenerator(rescale=1./255, shear_range  
    =0.2, zoom_range=0.2, horizontal_flip=True)
41 test_datagen = ImageDataGenerator(rescale=1./255)
42
43 training_set = train_datagen.flow_from_directory(train_dir,  
    target_size=(64, 64), batch_size=16, class_mode='binary')
44 test_set = test_datagen.flow_from_directory(test_dir, target_size  
    =(64, 64), batch_size=16, class_mode='binary')
45
46 print(training_set.class_indices)
47
48 history = classifier.fit(training_set, steps_per_epoch=len(  
    training_set), epochs=15, validation_data=test_set,  
    validation_steps=len(test_set))
49 total_images_processed = len(training_set) * 16
50
51 loss, accuracy = classifier.evaluate(test_set)
52 print("Perda:", loss)
53 print("Acurácia:", accuracy)
54
55 def is_image(file_path):
56     try:
57         Image.open(file_path)
58         return True
59     except IOError:
60         return False
61
62 y_pred = []
63 y_test = []
64
65 NORMAL_TEST_DIR = os.path.join(test_dir, "Normal")
66 COVID_TEST_DIR = os.path.join(test_dir, "Covid")
```

```
67
68 for path, label in [(NORMAL_TEST_DIR, 1), (COVID_TEST_DIR, 0)]:
69     for i in os.listdir(path):
70         img_path = os.path.join(path, i)
71         if is_image(img_path):
72             img = image.load_img(img_path, target_size=(64, 64))
73             img = image.img_to_array(img)
74             img = np.expand_dims(img, axis=0)
75             p = classifier.predict(img)
76             y_test.append(p[0, 0])
77             y_pred.append(label)
78
79 y_pred = np.array(y_pred)
80 y_test = np.array(y_test)
81
82 cm = confusion_matrix(y_test.round(), y_pred)
83 sns.heatmap(cm, annot=True, fmt=".0f", linewidths=.5, square=True
84             , cmap='Blues')
85 plt.show()
86
87 fpr, tpr, _ = roc_curve(y_test.round(), y_pred)
88 plt.plot(fpr, tpr, label=f'ROC curve (area = {roc_auc_score(
89     y_test.round(), y_pred):.2f})')
90 plt.show()
91
92 plt.hist(y_pred, bins=30, label='Previsões do Modelo')
93 plt.show()
94
95 plt.plot(history.history['accuracy'], label='Acurácia no Treino')
96 plt.plot(history.history['val_accuracy'], label='Acurácia na
97     Validação')
98 plt.show()
99
100 print(classification_report(y_pred, y_test.round()))
```