



**Modelo *Deep Learning* para Classificação de  
Imagens Tomográficas de Padrões de Pós-Covid**

**João Victor Rocha da Senhora Magalhães  
Cerqueira**

**Projeto Final de Graduação**

**Centro Técnico Científico – CTC  
Departamento de Informática  
Curso de Graduação em Engenharia da Computação**



**João Victor Rocha da Senhora Magalhães  
Cerqueira**

**Modelo *Deep Learning* para Classificação de  
Imagens Tomográficas de Padrões de Pós-Covid**

Relatório de Projeto Final, apresentado ao programa  
Engenharia de Computação da PUC-Rio como requisito parcial  
para a obtenção do título de Engenheiro de Computação.

Orientadora: Marley Maria Bernardes Rebuzzi Vellasco  
Coorientadora: Karla Tereza Figueiredo Leite

## **Agradecimentos**

Aos meus pais, sem os quais eu não teria atingido tantas conquistas.

À PUC–Rio, pela excelente formação acadêmica no decorrer desses anos.

À minha orientadora Marley Vellasco e coorientadora Karla Figueiredo pela oportunidade, apoio e motivação em um ano conturbado.

Aos médicos do HUPE pelo apoio com a rotulagem das imagens médicas utilizadas para treinar e testar o modelo.

À Isabela Amorim, minha namorada, pelo apoio e incentivo nos momentos mais difíceis durante toda a graduação.

## Resumo

Rocha da Senhora Magalhães Cerqueira, João Victor. Bernardes Rebuzzi Vellasco, Marley Maria. Figueiredo Leite, Karla Tereza. **Modelo Deep Learning para Classificação de Imagens Tomográficas de Padrões Pós-Covid.** Rio de Janeiro, 2022. 30p. Relatório de Projeto Final II — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Neste trabalho foi desenvolvido um modelo de classificação de imagens tomográficas ligadas às sequelas do pós-COVID ou COVID longa, que possibilita o auxílio ao diagnóstico de tais padrões aos médicos. O modelo foi implementado usando um algoritmo de Redes Neurais Convolucionais com *Transfer Learning* na linguagem Python, utilizando o *Google Colaboratory*, que é um serviço de nuvem gratuito hospedado pelo próprio Google e frameworks PyTorch. Os modelos desenvolvidos a partir de padrões pós-COVID, identificados em pacientes do HUPE, apresentaram resultados bastante promissores.

Palavras-chave

Redes Neurais Convolucionais, COVID, Tomografia Computadorizada, *Transfer Learning*

## Abstract

Rocha da Senhora Magalhães Cerqueira, João Victor. Bernardes Rebuzzi Vellasco, Marley Maria. Figueiredo Leite, Karla Tereza. **Deep Learning Model for Classification of Tomographic Images of Post-Covid Patterns.** Rio de Janeiro, 2022. 30p. Final Report for Final Project II — Department of Informatics. Pontifical Catholic University of Rio de Janeiro.

In this work, a classification model of tomographic images linked to post-COVID or long-term COVID sequelae was implemented, with the aim of helping doctors diagnose such patterns. The model was implemented using a Convolutional Neural Networks algorithm with *Transfer Learning* in Python language, using Google Colaboratory which is a free cloud service hosted by Google and PyTorch frameworks. The models developed from post-covid patterns, identified in HUPE patients, showed very promising results.

Keywords

Convolutional Neural Networks, COVID, Computed Tomography, Transfer Learning

# Sumário

|       |  |    |
|-------|--|----|
| 1     | Introdução   | 7  |
| 1.1   | Motivação  | 7  |
| 2     | Fundamentação Teórica                                | 10 |
| 2.1   | Deep Learning e COVID                                | 10 |
| 2.2   | Redes Neurais Convolucionais                         | 10 |
| 2.3   | Tratamento dos dados                                 | 12 |
| 3     | Objetivos  | 13 |
| 4     | Atividades Realizadas                                | 14 |
| 4.1   | Estudos Conceituais Preliminares                     | 14 |
| 4.2   | Estudo e Definição de Tecnologias a Serem Utilizadas | 14 |
| 4.3   | Metodologia e etapas do projeto                      | 15 |
| 4.4   | Cronograma   | 16 |
| 5     | Projeto e Especificação do Sistema                   | 17 |
| 5.1   | Metodologia proposta                                 | 17 |
| 5.2   | Arquiteturas de Rede                                 | 18 |
| 5.2.1 | ResNet-18  | 18 |
| 5.2.2 | SqueezeNet   | 19 |
| 5.2.3 | VGG-11   | 20 |
| 5.3   | Transfer Learning                                    | 21 |
| 5.4   | Parâmetros do modelo                                 | 22 |
| 5.4.1 | Otimizador AdamW                                     | 22 |
| 5.4.2 | Learning rate  | 23 |
| 5.4.3 | Weight decay   | 23 |
| 5.4.4 | Epsilon  | 23 |
| 5.4.5 | Criterion Cross Entropy                              | 23 |
| 5.4.6 | Decaimento da taxa de aprendizado StepLR             | 23 |
| 5.4.7 | Características adicionais                           | 24 |
| 6     | Implementação e Avaliação                            | 24 |
| 6.1   | Data Augmentation                                    | 24 |
| 6.2   | Arquitetura da rede                                  | 25 |
| 6.3   | Testes e avaliações                                  | 26 |
| 6.4   | Resultados do melhor modelo escolhido                | 30 |
| 7     | Considerações Finais                                 | 32 |
| 8     | Referências  | 33 |

## **Lista de figuras**

|  |    |
|--|----|
| Figura 1 – Arquitetura e funcionamento de uma CNN para classificação           | 12 |
| Figura 2 – Metodologia de um estágio para classificação de padrões pós-covid   | 17 |
| Figura 3 – Arquitetura da ResNet-18  | 19 |
| Figura 4 – Arquitetura da SqueezeNet   | 20 |
| Figura 5 – Arquitetura em camadas da VGG-11                                    | 21 |
| Figura 6 – Acurácia e perda nas etapas de treino e validação do Modelo 4       | 27 |
| Figura 7 – Matriz de confusão dos testes do Modelo 4                           | 28 |
| Figura 8 – Acurácia e perda nas etapas de treino e validação do Modelo 9       | 29 |
| Figura 9 – Matriz de confusão dos testes do Modelo 9                           | 29 |
| Figura 10 – Acurácia e perda nas etapas de treino e validação do melhor modelo | 31 |
| Figura 11 – Matriz de confusão do melhor modelo                                | 32 |

## **Lista de tabelas**

|  |    |
|--|----|
| Tabela 1 – Parâmetros utilizados para comparação das redes               | 26 |
| Tabela 2 – Configurações de hiper parâmetros avaliados durante o projeto | 26 |

# 1 Introdução

A doença conhecida como COVID-19, causada pelo vírus Sars-CoV-2, teve um impacto muito grande no mundo todo, sendo a responsável por centenas de milhares de mortes. A doença surgiu no final de 2019, e em fevereiro de 2020 a Organização Mundial da Saúde (OMS) informou que havia mais de 80 mil casos confirmados globalmente, mostrando quão rápida foi a propagação. Após isso, em 11 de março de 2020, a OMS declarou o novo coronavírus como uma pandemia mundial ([CUCINOTTA e VANELLI, 2020](#)).

Durante este período muito se noticiou a respeito das numerosas mortes e internações graves que tomaram as manchetes globais desde o início da pandemia. Uma estimativa da Organização Mundial da Saúde (OMS) ([ADAM, 2022](#)) é que o número de óbitos globais causados pelo vírus Sars-CoV-2 em 2020 e 2021 chegou a cerca de 15 milhões de pessoas, comprovando o impacto crítico causado pelo vírus.

No Brasil a situação foi bastante crítica devido ao grande número de óbitos e de pessoas contaminadas que precisaram de internação hospitalar como reflexo do cenário global. Atualmente houve uma melhora no cenário pandêmico, com o aumento da imunidade trazida pela vacinação, a qual levou a uma redução de 96,44% no número de mortes ([ARAÚJO e FERNANDES, 2022](#)). No entanto, apesar da retomada da rotina (tendo alguns lugares retirado as restrições de circulação ou mesmo de uso de máscaras), a doença ainda se faz presente e deve-se considerar a possibilidade de o vírus continuar em circulação no futuro.

## 1.1 Motivação

A COVID-19 é uma doença muito recente e novas variantes do vírus, apresentando mutações, têm aparecido com frequência, como é o caso da Ômicron que já consta com próprias subvariantes ([CORUM e ZIMMER, 2022](#)). Além disso, evidências de estudos de laboratório comprovam que algumas respostas imunes das vacinas atuais podem ser menos eficazes contra algumas dessas variantes ([BOLLINGER et al., 2022](#)). Portanto, ainda estão sendo feitos estudos no mundo todo para tentar mapear sinais e sintomas, buscando definir indicadores confiáveis que possam distinguir a doença com máxima antecedência e assim permitir iniciar o tratamento em tempo hábil para evitar o agravamento do quadro.

Uma abordagem que tem tido bastante destaque é a de tentar mensurar e acompanhar a COVID-19 através da análise de imagens médicas por meio de

algoritmos baseados em *Deep Learning*, como por exemplo os projetos “*Deep Learning based Diagnosis Recommendation for COVID-19 using Chest X-Rays Images*” (SAEEDAN et al., 2018) e “*Truncated inception net: COVID-19 outbreak screening using chest X-rays*” (DAS et al., 2020). Nestes trabalhos, algoritmos baseados em *Deep Learning* são utilizados para classificar COVID-19, tendo obtido resultados muito satisfatórios. Alguns deles até fazem uma comparação entre diferentes trabalhos desse tipo, destacando vantagens e desvantagens, além de suas respectivas acurárias, como é o caso do artigo “*A deep learning approach to detect Covid-19 coronavirus with X-Ray images*” (GOVARDHAN et al., 2020).

Além dessa abordagem, que busca alcançar a identificação de casos de COVID-19, seja para diagnóstico ou desfecho, há também projetos baseados em *Deep Learning* para detectar na imagem as anomalias compatíveis com os padrões registrados na literatura em imagens médicas, como é o caso do estudo “*COVID-19 pneumonia with hemoptysis: Acute segmental pulmonary emboli associated with novel coronavirus infection*” (KESSY et al., 2018). Nesse trabalho, o algoritmo busca identificar diferentes padrões no pulmão, destacando as áreas afetadas nas imagens de exames de TC (Tomografia Computadorizada) ou de Raio-X do pulmão de pacientes que estão com COVID-19. Com essas condições identificadas e destacadas, deseja-se associar as anomalias pulmonares ao COVID-19, e descobrir quais delas são decorrentes da doença e podem servir como caracterizadores da mesma.

Percebe-se que os estudos médicos realizados ainda têm se dedicado à compreensão da doença em si. Entretanto, hoje nos encontramos em um cenário social de convívio com a doença, em que muitas pessoas já contraíram a COVID-19, se recuperaram e, mesmo assim, permanecem com dores corporais, fadiga, perda de olfato e/ou paladar, perda de memória, problemas nos pulmões, dentre outras sequelas.

Um artigo publicado no jornal *The Lancet* intitulado “*1-year outcomes in hospital survivors with COVID-19: a longitudinal cohort study*” (HUANG et al., 2021) verificou as condições de 1227 pacientes do Hospital Jin Yin-tan, que fica em Wuhan, China, 6 meses e 1 ano após terem contraído a doença. Este estudo verificou que a proporção de pacientes com pelo menos uma sequela após 6 meses da doença foi de 68%, enquanto após 12 meses foi de 49%, sendo fadiga ou fraqueza muscular os sintomas mais comum entre eles.

Dentre as possíveis sequelas causadas pelo vírus Sars-CoV-2 e verificadas a longo prazo, algumas atingem o pulmão e são visíveis em imagens de

Tomografia Computadorizada. No artigo “*The Importance of Radiological Patterns and Small Airway Disease in Long-Term Follow-Up of Postacute COVID-19: A Preliminary Study*” ([MOGAMI et al., 2022](#)) são identificados alguns quadros como fibrose, reabsorção, vias aéreas e pines que causam sintomas como dificuldade de respirar, dores e obstrução no pulmão dos pacientes. O referido estudo ainda aponta que essas manifestações, pós-agudas de COVID-19, podem permanecer relevantes por muito tempo.

Dito isto, fica claro que existe uma quantidade expressiva de pessoas que se veem com algum sintoma duradouro após se recuperarem da doença. Entretanto, ainda são poucos os estudos que têm como enfoque a área do pós-COVID, a qual envolve entender, identificar, qualificar e medir as possíveis sequelas causadas pela doença após a fase aguda da COVID-19. Avanços nesse campo de pesquisa permitiriam um acompanhamento mais adequado das pessoas nessa situação.

Devido aos bem-sucedidos projetos de estudo da COVID-19 que fazem uso de *Deep Learning*, como os já expostos nesse trabalho, nada mais natural que buscar novamente nessa técnica a perspectiva de avançarmos no campo do pós-COVID. É neste contexto que reside a justificativa e a relevância deste trabalho, o qual tem como foco desenvolver um modelo de classificação das sequelas pulmonares identificadas no já mencionado estudo “*The Importance of Radiological Patterns and Small Airway Disease in Long-Term Follow-Up of Postacute COVID-19: A Preliminary Study*” ([MOGAMI et al., 2022](#)).

O projeto foi desenvolvido na linguagem Python, utilizando o Google Colaboratory, que permite a escrita de texto e código executável num mesmo lugar, facilitando a organização do trabalho e análise dos resultados. Essa ferramenta é um serviço de armazenamento em nuvem gratuito, hospedado e fornecido pelo próprio Google, o qual disponibiliza um alto poder computacional para ensino e pesquisa na área de aprendizado de máquina. Além de ser flexível, já possui o *PyTorch* ([PASZKE et al., 2019](#)) pré-instalado, que é uma biblioteca de código aberto em Python para *Deep Learning*, visando à investigação e à avaliação de modelos. Essa foi a biblioteca utilizada para o modelo implementado, tendo as ferramentas sido escolhidas após estudo e verificação do quanto são amplamente utilizadas para esse tipo de projeto ([MINAEE et al., 2021](#); [ORTIZ et al., 2022](#)).

## 2 Fundamentação Teórica

De forma geral, a classificação de imagens é um processo que se inicia com uma base de dados de qualidade e bem organizada de imagens anotadas por especialistas, passando por pré-processamento das imagens, e culminando na investigação de modelos capazes de cumprir a tarefa proposta pelo algoritmo. Portanto, se tornou necessário pesquisar sobre as tecnologias expoentes na área, sua fundamentação teórica e outros estudos existentes.

Considerando a computação para uso no campo da saúde, se destacam as Redes Neurais, modelo computacional com nós interconectados que foram inspirados nos neurônios do cérebro humano (HAYKIN, 2009). Os resultados alcançados por esse tipo de modelo em diferentes tarefas se mostraram excelentes, como por exemplo para classificação, segmentação e reconhecimento de imagens de outras doenças respiratórias. No caso da COVID-19, uma abordagem semelhante vem sendo utilizada com foco nos algoritmos de *Deep Learning* que serão apresentados a seguir.

### 2.1 Deep Learning e COVID

O fato da COVID-19 estar associada a doenças do trato respiratório, que têm como meio de diagnóstico exames que utilizam imagens, principalmente raio-X e tomografia computadorizada, faz com que seja possível o uso delas para desenvolver um modelo capaz de classificar os padrões presentes nas mesmas. Alguns estudos recentes se destacam na detecção da COVID-19 com essa estratégia, como por exemplo “*A Comprehensive Survey of COVID-19 Detection Using Medical Image*” (SHAH et al., 2020) e “*Deep Learning Approaches for Detecting COVID-19 From Chest X-Ray Images: A Survey*” (ALGHAMDI et al., 2021). Esses trabalhos explicam como o uso de Redes Neurais Convolucionais (do inglês, *Convolution Neural Network-CNN* (GOODFELLOW et al., 2016)) é particularmente adequado para a tarefa de classificação. Isso é feito por meio da extração das características desses dados de forma automatizada em um processo de aprendizado supervisionado, no qual o conjunto de imagens é previamente rotulado e fornecido ao modelo.

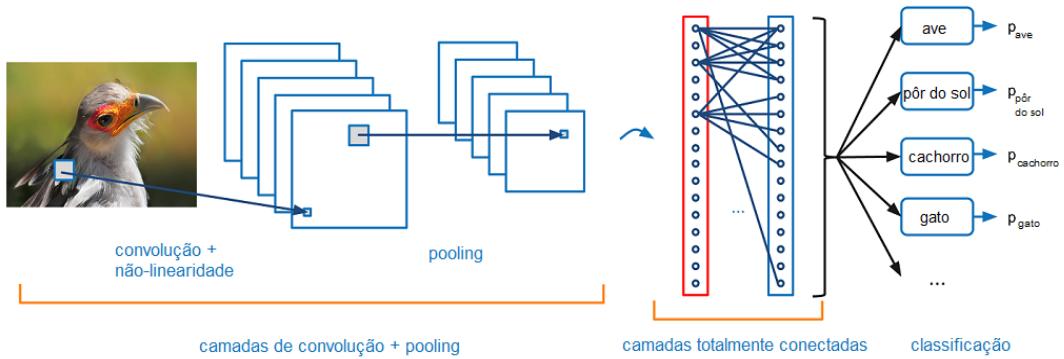
### 2.2 Redes Neurais Convolucionais

As Redes Neurais Convolucionais ou CNN (GOODFELLOW et al., 2016) funcionam da seguinte forma: cada imagem de um conjunto de treinamento passa por uma série de camadas convolutivas (JAVADIMOGHADDAM e

[GHOLAMALINEJAD, 2021](#)) dentro da rede, também chamadas de filtros, retornando uma saída, que pode ser composta por um conjunto de probabilidades. Dependendo da quantidade e organização dessas camadas é possível criar diferentes arquiteturas de rede, cada uma com suas vantagens e desvantagens ([KHAN et al., 2020](#)).

A arquitetura básica de uma CNN é composta de camadas alternadas de convolução e *pooling*, seguidas por uma ou mais camadas totalmente conectadas. Conforme explicado por Jason Brownlee ([BROWNLEE, 2019](#)) uma convolução se traduz na aplicação de um filtro na imagem resultando numa ativação. Ao se aplicar esse filtro repetidas vezes por toda a imagem, é gerado o chamado *feature map* que indica as regiões da imagem onde aquela *feature* (representada pelo filtro) é mais presente. As camadas de convolução aplicam diversos filtros em paralelo, servindo então para extrair a informação dos dados. Já as camadas de *pooling*, conforme descrito no Deep Learning Book ([DATA SCIENCE ACADEMY, 2022](#)), recebem o *feature map* da camada de convolução e condensam essa informação, reduzindo assim a dimensionalidade. No caso das CNN é aplicado um *Max Pooling* em cada camada de *pooling* que simplesmente calcula o maior valor de ativação da região filtrada, pois o importante para a rede é saber o quanto a *feature* está presente naquela região. Por fim, as camadas totalmente conectadas associam a saída das camadas iniciais aos rótulos (classes anotadas pelos especialistas), tendo o objetivo de aprenderem a predizer a classe que melhor descreve novas imagens.

O mecanismo que ocorre nas camadas de convolução e *pooling* conforme descrito acima é chamado de *Feature Extraction* ([VARSHNI et al., 2019](#)). É através desse processo que o modelo extrai as características que serão utilizadas pelo algoritmo de aprendizado para reconhecer os cenários que se deseja com o uso de *Feature Learning* ([SHORTEN, 2019](#)). A Figura 1 mostra um exemplo de uma arquitetura de CNN.



**Figura 1 – Arquitetura e funcionamento de uma CNN para classificação**

### 2.3 Tratamento dos dados

Uma etapa importante antes do treinamento de modelos de *Deep Learning* é o tratamento prévio dos dados. Algumas técnicas de tratamento utilizadas nesse projeto se mostraram benéficas para o aprendizado, principalmente a de *Data Augmentation* (SHORTEN e KHOSHGOFTAAR, 2019), na qual são incluídas imagens copiadas das imagens originais, porém com pequenas modificações inseridas nestas, aumentando a quantidade total de imagens que serão usadas para treinar o modelo. Essa técnica é essencial quando se tem uma pequena quantidade de dados de entrada, pois consegue gerar mais dados para o treinamento do modelo melhorando sua robustez e generalização.

Outra prática de pré-processamento que ajuda a melhorar o desempenho do algoritmo é a normalização:

“A normalização dimensiona as entradas de uma camada para um valor comum para cada lote durante o treinamento de redes neurais profundas. Isso estabiliza o processo de aprendizado e reduz significativamente o número de épocas necessárias para treinar redes profundas, permitindo que a rede treine mais rapidamente” (AMPADU, 2021) .

Ao incluir a normalização em conjunto com outros processos de regularização como o *dropout* e *early stopping* (FEDORENKO et al., 2018) é possível reduzir o chamado *overfitting*, que ocorre quando o modelo é ajustado demais ao conjunto de dados de treinamento. Nessa situação, apesar dos excelentes resultados apresentados durante o treinamento, o desempenho nos testes é prejudicado, pois o modelo não consegue responder adequadamente devido à sua incapacidade de generalizar. Portanto, a importância de tais práticas

é bem difundida na área, já que o objetivo final é obter modelos que possam manter alta acurácia com dados novos previamente desconhecidos. Isso está associado com os conceitos de viés e variância que devem ser equilibrados ao se construir um algoritmo supervisionado de aprendizado de máquina. Eles representam as capacidades do modelo de precisão e generalização, conforme explicado abaixo pelo guia do site *Master's in Data Science*<sup>1</sup>:

“Um modelo com alta variância pode representar o conjunto de dados com precisão, mas pode levar a um ajuste excessivo a dados de treinamento ruidosos ou não representativos. Em comparação, um modelo com alto viés pode se ajustar demais aos dados de treinamento devido a um modelo mais simples que ignora as regularidades nos dados.” (MASTER'S, 2021)

### 3 Objetivos

A partir da identificação de alguns padrões comuns de anomalias pulmonares em pessoas que tiveram COVID-19 e do bom desempenho dos modelos adotados para o diagnóstico da COVID-19, este trabalho investigou a utilização de modelos, baseados em *Deep Learning*, capazes de classificar padrões em exames TC (Tomografia Computadorizada) do pulmão de pacientes após estado agudo da doença.

Os dados utilizados neste trabalho são imagens de TC de pacientes que tiveram a doença, fornecidos pelo Hospital Pedro Ernesto (HUPE), em um projeto de parceria com a CAPES. As imagens foram classificadas por especialistas nos padrões Pines, Fibrose, Reabsorção e Vias Aéreas identificados no artigo “*The Importance of Radiological Patterns and Small Airway Disease in Long-Term Follow-Up of Postacute COVID-19: A Preliminary Study*” (MOGAMI et al., 2022) já citado. Essas imagens foram rotuladas por médicos do HUPE com os respectivos padrões presentes para então serem utilizadas nas etapas de treinamento, validação e teste dos modelos investigados. Estes contaram com diferentes arquiteturas de rede e configurações de parâmetros, onde o conhecimento adquirido previamente foi aplicado.

Assim, o objetivo do trabalho é identificar o melhor modelo entre os analisados e classificar as sequelas causadas pelo vírus em pacientes pós-covid, permitindo assim o acompanhamento dos impactos de médio-longo prazo da COVID-19. Como objetivos secundários pode-se mencionar que o presente

---

<sup>1</sup> Disponível em <<https://www.mastersindatascience.org/>>.

trabalho:

- ajudará na criação de uma ferramenta que visa automatizar a identificação de padrões pós-covid;
- trará maior celeridade ao diagnóstico de pacientes, importante para a escolha adequada da terapia médica.

## 4 Atividades Realizadas

### 4.1 Estudos Conceituais Preliminares

Primeiramente foi proposto um estudo teórico sobre Redes Neurais Convolucionais e sua aplicação nos estudos já mencionados, além de outros relevantes para o projeto. O aluno já contava com algum conhecimento sobre conceitos de *Deep Learning*, tais como retroalimentação de erros, camadas, neurônios, entre outros. Os conceitos das seguintes técnicas foram aprendidos e considerados para o projeto: *Autoencoders* (LIOU et al., 2014), *PCA whitening* (KESSY et al., 2018), *Pooling* (SAEEDAN et al., 2018), *Data Augmentation* (SHORTEN e KHOSHGOFTAAR, 2019), *Transfer Learning* (TORREY e SHAVLIK, 2009), entre outros. Após o estudo dessas técnicas foi possível passar para a avaliação e escolha das ferramentas que seriam utilizadas para implementar o algoritmo proposto.

### 4.2 Estudo e Definição de Tecnologias a Serem Utilizadas

Como mencionado, foram consideradas diversas ferramentas para o projeto, tais como o uso do Jupyter Notebook em conjunto com os frameworks *Pytorch* ou *Keras* e *TensorFlow*, muito utilizados nessa área de *Deep Learning*. Após avaliação e testes, foi decidido que seria utilizado o Google Colaboratory com a ferramenta *PyTorch*, já que facilitaria o compartilhamento do notebook e o uso de *Transfer Learning* de outros projetos usados como referência, além de contar com extenso suporte e documentação.

Um resumo desses pontos se encontra abaixo, detalhando um pouco mais as pesquisas e testes realizados inicialmente:

- Pesquisa:
  - Técnicas de Pré-Processamento de imagens;
  - Técnicas de *Data Augmentation* para aumentar a quantidade de imagens disponíveis;
  - Técnicas de *Transfer Learning* de outros projetos com *fine-tunning* das camadas finais;

- Arquiteturas populares para processamento de imagens médicas no formato DICOM.
- Testes:
  - o Uso da combinação de ferramentas *Google Colaboratory Notebook* e Pytorch;
  - Uso da combinação de ferramentas *Google Colaboratory Notebook* e TensorFlow;
  - Uso da combinação de ferramentas Keras e TensorFlow.

Tanto o Keras com Tensorflow como o Pytorch se mostraram ferramentas muito boas para a utilização no projeto.

### 4.3 Metodologia e etapas do projeto

A metodologia do projeto do sistema foi organizada nas seguintes fases:

- 1) A primeira parte das pendências foi aumentar a proficiência na ferramenta *PyTorch*, aprender como configurar os modelos, treiná-los e testá-los, além de gerar e analisar os gráficos e parâmetros de avaliação de resultados.
- 2) O desenvolvimento do projeto se iniciou com a definição dos tipos de *Data Augmentation* (Ng et al., 2013) dinâmicos a serem feitos, pois tínhamos um número limitado de imagens para fornecer ao modelo e precisávamos fazer uso dessa técnica para aumentar a quantidade de dados disponíveis. Além disso, também foi feito um estudo para aplicação de procedimentos para normalização e *early-stopping* (AMPADU, 2021) utilizados para aperfeiçoar o treinamento.
- 3) Em seguida, passou-se à etapa de realização de testes com algumas arquiteturas de rede que obtiveram bons resultados em projetos parecidos que tinham a tarefa de classificar imagens de Tomografia Computadorizada do pulmão para doenças como pneumonia e COVID-19, entre outras. Preliminarmente foi observado que redes menores pareciam mais adequadas para o caso, já que havia uma quantidade limitada de dados.
- 4) Após a escolha de uma arquitetura específica, passou-se a realizar testes modificando a configuração da rede escolhida utilizando diferentes *optimizadores* (VARSHNI et al., 2019), valores de *learning rate*, *momentum*, *epsilon* e *resize* de imagens buscando maximizar a acurácia de validação do modelo.

5) A etapa final se deu após a escolha do modelo mais eficiente e acurado para o conjunto de validação, com o conjunto de testes e análise dos resultados obtidos, assim como um estudo sobre a efetiva utilização do modelo como uma ferramenta de auxílio ao diagnóstico de novos pacientes.

#### **4.4 Cronograma**

##### **Cronograma Definido na Proposta**

*01/10/2021 - 30/10/2021: Estudo de CNN e técnicas de aprendizado não supervisionado de Deep Learning em imagens.*

*30/10/2021- 02/12/2021: Estudo e treinamento da biblioteca TensorFlow e da ferramenta Keras que serão utilizados para os modelos.*

*02/12/2021- 01/03/2022: Desenvolvimento e análise dos diferentes modelos propostos e avaliação de melhor arquitetura e técnicas adotadas para o modelo final.*

*01/03/2022- 14/04/2022: Ajustes finais do modelo selecionado e análise de performance. Estudo e Interpretação dos resultados e da classificação do modelo.*

*14/04/2022- 02/07/2022: Elaboração do relatório final e da apresentação final do Projeto Final II.*

##### **Cronograma Real de Atividades Realizadas**

*01/10/2021 - 30/10/2021: Estudo de CNN e técnicas de aprendizado não supervisionado de Deep Learning em imagens.*

*01/11/2021- 24/11/2021: Estudo sobre as opções de tecnologias disponíveis e escolha daquelas a serem utilizadas no projeto.*

*24/11/2021- 15/12/2021: Treinamento na utilização da biblioteca Pytorch para melhorar entendimento e técnicas de uso.*

*15/12/2021- 18/02/2022: Definição dos tipos de Data Augmentation a serem feitos, testes com diferentes arquiteturas de rede e escolha da arquitetura a ser utilizada pelo modelo.*

18/02/2021- 10/04/2022: Testes com diferentes parâmetros de rede visando melhorar a acurácia do modelo e encontrar os parâmetros que proporcionam o melhor desempenho e resultados.

10/04/2022- 24/04/2022: Ajustes finais do modelo selecionado e análise de performance. Estudo e Interpretação dos resultados e da classificação do modelo.

24/04/2022- 17/06/2022: Elaboração do relatório final que já tinha sido iniciado ao longo do semestre e elaboração da apresentação final do Projeto Final II.

## 5 Projeto e Especificação do Sistema

### 5.1 Metodologia proposta

Neste trabalho foi proposta uma metodologia para classificação das imagens pós-covid considerando um estágio classificatório, como pode ser visto na Figura 2 abaixo:



**Figura 2 - Metodologia de um estágio para classificação de padrões pós-covid**

Para o ajuste dos modelos foram usadas imagens tomográficas de 32 pacientes identificadas em padrões pós-covid (fibrose, vias aéreas, pines e reabsorção) e normal, separadas por radiologistas do HUPE (Hospital Universitário Pedro Ernesto – UERJ). A base de dados é constituída de cortes (*slices*) e foi dividida em 160 imagens de treinamento, 40 para validação e 44 para teste para cada classe de saída.

Devido à baixa quantidade de imagens para o treinamento do modelo foi necessário se recorrer a técnicas de *Data Augmentation* ([SHORTEST](#) e [KHOSHGOFTAAR, 2019](#)), onde determinadas transformações são realizadas nas imagens originais criando “novas” imagens do ponto de vista do modelo.

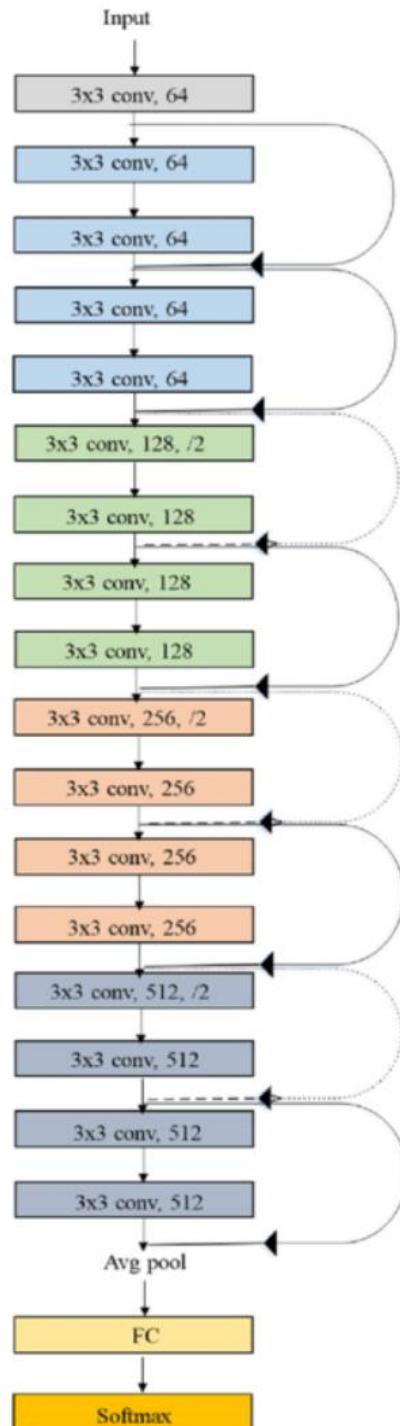
Considerando a natureza das imagens, foram utilizadas as seguintes técnicas, dentre as mais tradicionais de *data augmentation* citadas no artigo referenciado acima: *center\_crop*, *horizontal\_rotation* e *flip* que serão discutidas na seção de testes. Destaca-se que a base de teste não foi alterada com esta técnica, sendo o teste realizado com imagens originais.

## 5.2 Arquiteturas de Rede

Conforme já descrito na seção 2.2, considerando que cada arquitetura de rede possui quantidade e configuração únicas de camadas com respectivas vantagens e desvantagens, foram investigadas algumas arquiteturas de rede diferentes na busca daquela que melhor se adequaria ao presente trabalho. Devido à pequena quantidade de dados disponíveis, priorizou-se redes com tamanhos menores pois elas responderam mais adequadamente em comparação com as maiores. As redes investigadas foram:

### 5.2.1 ResNet-18

A ResNet é uma arquitetura que busca criar uma rede com grande profundidade de camadas convolucionais, mas permite conexões entre as camadas visando “pular” algumas camadas, criando conexões de atalhos na rede. Assim, apesar do grande número de camadas, o funcionamento é eficiente (HE et al., 2015), pois permite um grande número de camadas, reduzindo as perdas por *vanishing gradient* no processo de aprendizado e aumentando a velocidade de aprendizado. No caso da ResNet-18, ela possui 72 camadas sendo 18 camadas profundas. A arquitetura da ResNet-18 pode ser vista na Figura 4 abaixo.



**Figura 3 – Arquitetura da ResNet-18**

### 5.2.2 SqueezeNet

A SqueezeNet é uma rede que emprega estratégias de arquitetura para reduzir o número de parâmetros da rede, principalmente com o uso *fire modules* que "comprimem" os parâmetros usando convoluções 1x1 e os expandem posteriormente (LANDORA et al., 2017). A Figura 5 mostra a arquitetura da SqueezeNet, onde pode-se ver as várias camadas de contração e expansão

presentes na rede.

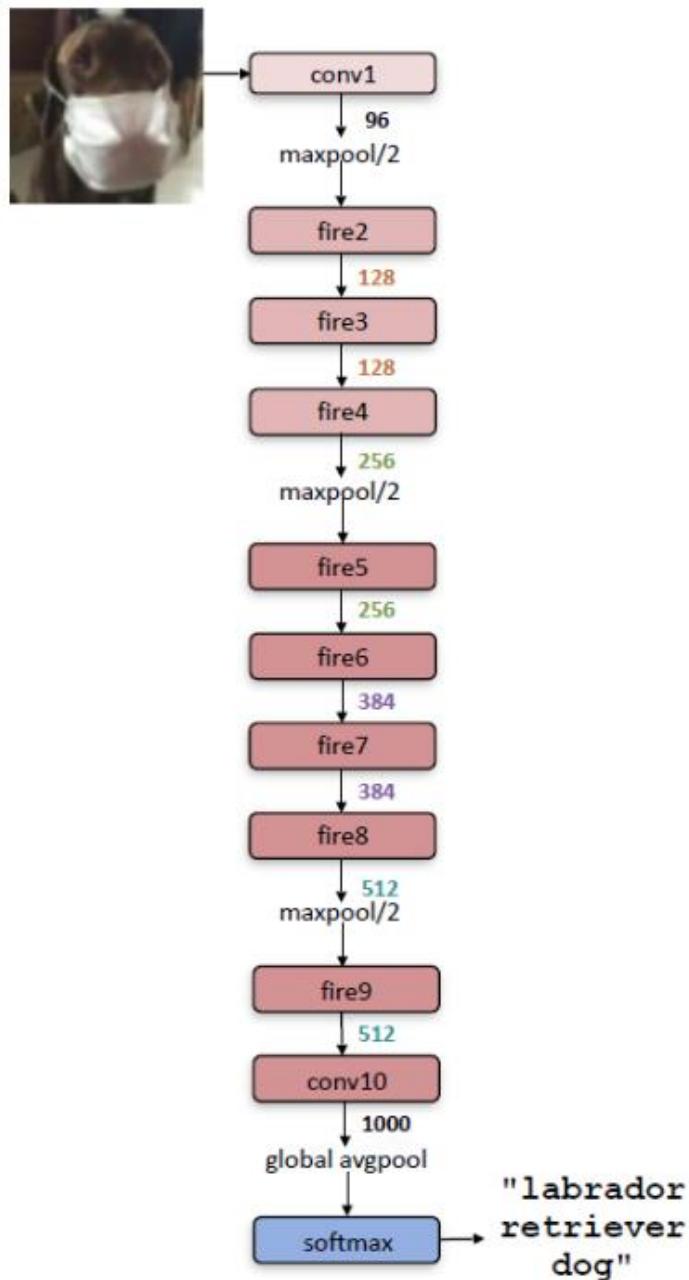
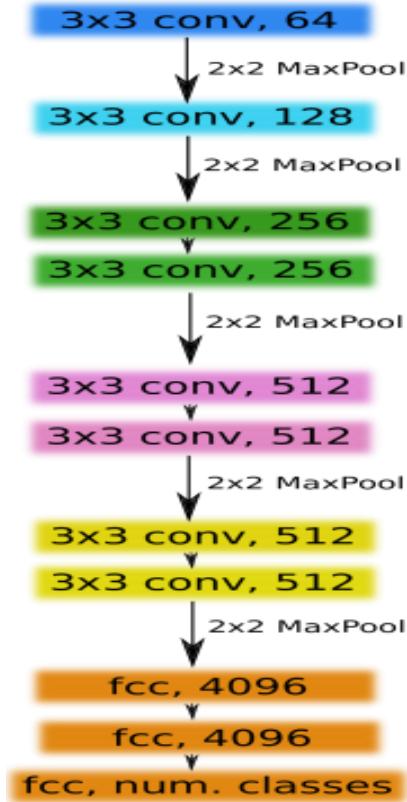


Figura 4 – Arquitetura da SqueezeNet

### 5.2.3 VGG-11

A VGG é uma arquitetura de CNN proposta por K. Simonyan e A. Zisserman (SIMONYAN e ZISSERMAN, 2014) que é composta por um modelo robusto devido à sua uniformidade e arquitetura. A VGG-11, como o nome indica, tem 11 camadas, 8 das quais são convolutivas e três estão totalmente conectadas ou densas como pode ser visto na Figura 3 abaixo:



**Figura 5 – Arquitetura em camadas da VGG-11**

Em geral, quanto maior o número de parâmetros (pesos sinápticos dos neurônios artificiais) maior é a quantidade de dados exigida para ajuste dos modelos, e quanto maior a rede convolucional, maior a quantidade de pesos ela tem. No caso da VGG-11 ela possui 133.000.000 parâmetros (pesos) a serem ajustados durante o treinamento.

### 5.3 Transfer Learning

Devido à baixa quantidade de imagens de padrões pós-covid disponíveis, optou-se pelo uso de *Transfer Learning* (TORREY e SHAVLIK, 2009), outra técnica bastante difundida que visa utilizar um modelo já treinado com a mesma arquitetura, mas a partir de uma base de dados de outro contexto (com grande quantidade de dados), para reaproveitar o conhecimento adquirido durante as camadas de convolução e *pooling*. Assim, com o uso de *Transfer Learning* foi possível reduzir o tempo necessário para o treinamento dos modelos e a quantidade de dados requisitada. Esse procedimento é muito conveniente quando não se dispõe dos recursos do contexto original (tamanho do dataset, poder computacional, tempo de prototipagem) e também é capaz de fornecer benefícios em problemas com maior quantidade de recursos. Esses benefícios são: melhorar

a capacidade de generalização das redes neurais, característica que se traduz na habilidade de responder corretamente a padrões desconhecidos pelo modelo, e a aprender mais rápido, já que permite que o novo aprendizado parta de um estado inicial mais favorável (SORIA OLIVAS e MARIN GUERRERO, 2010).

Com isso, foi feito o *Transfer Learning* a partir de um modelo pré-treinado (pesos e parâmetros) disponibilizado publicamente, nesse caso o da ImageNet (KRIZHEVSKY et al., 2012), cuja base de dados usada para gerar o modelo possui mais de 14 milhões de imagens e mais de mil classes. Isso permitiu que as características de imagens de TC fossem importadas para que em seguida os pesos das camadas densas da nossa arquitetura fossem ajustados com base nas imagens do nosso contexto de pós-covid, procedimento conhecido como *fine-tuning*. Nesse processo, as últimas camadas da rede pré-treinada são substituídas por camadas com pesos inicialmente aleatórios e pequenos que são ajustados dinamicamente durante o treinamento, enquanto os pesos importados da rede pré-treinada permanecem congelados. O algoritmo ajusta estes pesos considerando o erro ao final de uma iteração, que é calculado a partir da diferença entre o rótulo informado e o valor produzido pela rede para cada um dos dados, processo chamado de *backpropagation*. O processo de aprendizagem de uma rede neural realiza um processo iterativo de otimização de pesos objetivando minimizar o erro ao máximo possível (CHEN e JAIN, 1994) sendo então os pesos os parâmetros que armazenam o conhecimento adquirido nas etapas de aprendizagem.

## 5.4 Parâmetros do modelo

O modelo final desenvolvido foi treinado com uma rede VGG-11, onde os seguintes parâmetros foram testados em diferentes configurações:

### 5.4.1 Otimizador AdamW

Otimizadores são os algoritmos usados para alterar os atributos da rede neural durante o treinamento, como pesos dos neurônios e a taxa de aprendizado da rede. O Adam (KINGMA e BA, 2015) é um otimizador que é diferente do otimizador padrão SGD (RUDER, 2016) que também foi utilizado em alguns testes realizados. O Adam mantém uma taxa de aprendizado por parâmetro da rede e faz uso da média móvel das variâncias dos próprios gradientes controladas por parâmetros na atualização da rede, além da média móvel dos quadrados dos gradientes passados que é feita pelo SGD. Na prática, este algoritmo treina muito mais rápido que o SGD e atinge bons resultados. O AdamW é uma versão

modificada do Adam que aplica um *weight decay* (peso adicionado na atualização dos pesos da rede) após a modificação feita pelo cálculo dos gradientes. Isso faz com que o modelo atinja uma perda menor durante o treinamento e tenha uma melhor capacidade de generalização (LOSHCHILOV e HUTTER, 2019).

#### 5.4.2 Learning rate

A *learning rate* ou taxa de aprendizado controla a velocidade de aprendizado do modelo, ou seja, o quanto ele será atualizado a cada iteração do treinamento.

#### 5.4.3 Weight decay

O *weight decay* é um peso que serve para regularizar o aprendizado do modelo (diferente do que é aplicado no algoritmo AdamW). Ele é uma pequena penalidade que é adicionada à função de perda do modelo fazendo com que se reduza o *overfitting*, já que o modelo focará nos padrões que aparecerão mais frequentemente.

#### 5.4.4 Epsilon

O *epsilon* é um valor que é aplicado no denominador de uma das equações de erro, servindo para impedir que exista uma divisão por zero. O valor default dele é 1e-8, mas no nosso modelo utilizamos 1e-7 para verificar se haveria uma melhora. Apesar de não resultar em uma mudança expressiva, mantivemos o valor em alguns modelos.

#### 5.4.5 Criterion Cross Entropy

O *Cross Entropy Loss* (ELIJAH KOECH, 2020) foi a função de custo ou função de perda utilizada pelo modelo. Essa é a função responsável por calcular o erro a cada iteração, e cabe mencionar que o aprendizado do modelo é baseado na minimização desta função.

#### 5.4.6 Decaimento da taxa de aprendizado StepLR

A função StepLR utilizada é uma função que diminui a *learning rate* do modelo com o decorrer das iterações de treinamento. Esse procedimento de ajuste da *learning rate* é benéfico, pois queremos que ela mude conforme o treinamento progride para se adequar aos nossos objetivos ao longo do treinamento. No começo do processo de treinamento a *learning rate* é inicializada com um valor alto, pois se deseja ajustar os pesos com uma velocidade maior

para que ele consiga encontrar a mínima global da função de custo. Além disso, se inicializarmos uma *learning rate* baixa o modelo levará muito tempo para treinar. Com a evolução do treinamento, entretanto, o objetivo passa a ser realizar um ajuste fino nesses pesos, inicialmente definidos, de forma a aperfeiçoar o modelo, devendo-se reduzir a *learning rate*. Nos modelos investigados, utilizou-se um decaimento da taxa de aprendizado programado em que a cada 7 épocas (ou iterações) de treinamento a *learning rate* foi diminuída em 0.1.

#### 5.4.7 Características adicionais

Vale destacar que foi aplicado um redimensionamento das imagens de treinamento e teste em alguns dos modelos investigados. Isso agilizou o processo de treinamento sem que houvesse perda de informação. Os tamanhos utilizados nos testes foram o tamanho original das imagens, dois terços do tamanho original, um terço do tamanho original e um sexto do tamanho original. Além disso, foi aplicada uma normalização em todas as imagens para padronização dos dados, foi fixado o número de 30 épocas para o treinamento com possibilidade de parada antecipada e foi utilizado um batch size de 16 imagens em todos os treinamentos devido à limitação de memória do *Google Colaboratory*.

## 6 Implementação e Avaliação

### 6.1 Data Augmentation

A primeira tarefa realizada durante a implementação do código do trabalho foi a de selecionar e implementar os tipos de *data augmentation* a serem utilizados. Considerando a natureza das imagens de Tomografia Computadorizada, foram escolhidas as seguintes transformações visando variar as características, sem prejudicar o aprendizado do modelo:

- CenterCrop – Essa transformação funciona como um ‘zoom’ na imagem, seguido de um posterior redimensionamento para o tamanho original. Com isso pode-se gerar mais imagens levemente modificadas a partir de uma original, mas mantendo as mesmas características. Foram escolhidas 3 proporções de zoom a serem aplicadas aleatoriamente durante o treinamento: 70%, 80% e 90%.
- RandomHorizontalFlip – Essa transformação inverte a imagem no eixo horizontal e foi considerada como não prejudicial, enquanto a inversão no eixo vertical foi descartada. Isso foi definido pois apesar da inversão da posição do coração na imagem rotacionada horizontalmente, seria mantida a região do tórax e a orientação do pulmão nesse tipo de transformação, mantendo assim as

características que se desejava identificar nas imagens.

- RandomRotation – Essa transformação rotaciona a imagem aleatoriamente no sentido horário ou anti-horário, tendo sido imposto um limite de até 15 graus para cada um dos lados. Dessa forma, pode-se gerar imagens diferentes das originais, sem prejudicar o modelo com uma rotação que altere muito a orientação do pulmão no tórax.

Essas modificações foram aplicadas dinamicamente, ou seja, a cada *batch* de imagens de treinamento a ser utilizado na iteração corrente do algoritmo. Assim, as funções de *data augmentation* descritas acima foram aplicadas aleatoriamente nas imagens selecionadas para aquele batch com uma probabilidade de 80%, de forma a manter algumas imagens originais presentes na iteração. Considerando o grande número de iterações que ocorrem durante o treinamento, a quantidade de ‘novas’ imagens que são usadas pelo modelo cresce muito, favorecendo a capacidade de generalização do modelo.

## 6.2 Arquitetura da rede

Como mencionado anteriormente, o modelo ideal seria aquele que consegue classificar as imagens com acurácia, se tornando então uma ferramenta confiável de auxílio no diagnóstico das condições nos pulmões de pacientes pós-COVID.

Com essa finalidade, tendo finalizado a implementação do *data augmentation* para aumentar a diversidade das imagens do modelo, passou-se à investigação das arquiteturas de rede, visando a encontrar aquela que tivesse o melhor desempenho para a classificação dos padrões pós-COVID. Foi definida uma inicialização dos parâmetros de rede como default e testadas três arquiteturas de rede com esses parâmetros, utilizando *Transfer Learning* de redes públicas pré-treinadas disponibilizadas no repositório público *Pytorch Hub*<sup>2</sup>.

As arquiteturas investigadas foram: ResNet-18, SqueezeNet e VGG-11. Os parâmetros utilizados para a comparação das redes estão apresentados na Tabela 1, lembrando novamente que o número de épocas foi fixado em 30, com possibilidade de parada antecipada, e o *batch size* de imagens utilizado foi fixado em 16, por limitações de memória do *Google Colaboratory*.

A arquitetura VGG-11 foi apresentada na Seção 5 do trabalho e foi treinada uma rede com os mesmos parâmetros da ResNet e SqueezeNet para comparação. Após avaliação das arquiteturas, decidiu-se pela VGG-11, pois

---

<sup>2</sup> Disponível em <<https://pytorch.org/hub/>>.

apresentou melhor desempenho, tempo de treinamento e parece ter se adequado bem à quantidade de dados.

| Learning Rate | Momentum | Transfer Learning | Criterion    | LR Decay |
|---------------|----------|-------------------|--------------|----------|
| 0.001         | 0.9      | Sim               | CrossEntropy | StepLR   |

**Tabela 1 – Parâmetros utilizados para comparação das redes**

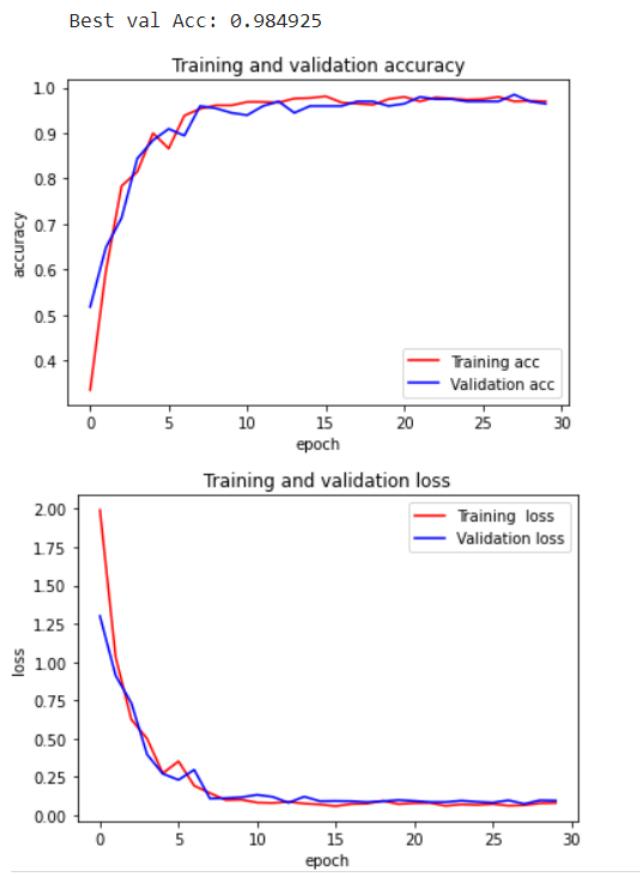
### 6.3 Testes e avaliações

Com a arquitetura de rede VGG-11 selecionada para o algoritmo, foram feitos testes com diferentes valores de hiper-parâmetros, visando a aumentar a acurácia do modelo. Foram testados os otimizadores de rede SGD, Adam e AdamW utilizando diferentes valores para os seguintes parâmetros: *learning rate*, *momentum*, *weight decay*, e *epsilon*. Todas as configurações de teste com as respectivas acurácia de validação e de teste podem ser observadas na Tabela 2, com destaque para o Modelo 11 em negrito que foi selecionado como o melhor modelo. Vale relembrar que todos os modelos contaram com *Transfer Learning* e arquitetura VGG-11, além de utilizarem a função de custo *CrossEntropy* e a função de atualização da taxa de aprendizado *StepLR*, conforme descrito na seção 5 do trabalho.

|                  | Tamanho de Resize         | Optimizer    | Learning Rate | Momentum   | Weight Decay   | Epsilon         | Melhor Accuracy Validação | Accuracy de Teste |
|------------------|---------------------------|--------------|---------------|------------|----------------|-----------------|---------------------------|-------------------|
| Modelo 1         | 70% do Original           | SGD          | 0.001         | 0.9        | Não            | Não             | 80.40%                    | 82.27%            |
| Modelo 2         | Tamanho Original          | SGD          | 0.001         | 0.9        | Não            | Não             | 90.95%                    | 90.45%            |
| Modelo 3         | 16.66% do Original        | SGD          | 0.001         | 0.9        | Não            | Não             | 67.33%                    | 72.73%            |
| Modelo 4         | 33.33% do Original        | SGD          | 0.001         | 0.9        | Não            | Não             | 98.49%                    | 94%               |
| Modelo 5         | Tamanho Original          | SGD          | 0.0001        | 0.9        | 0.001          | Não             | 84.42%                    | 81%               |
| Modelo 6         | Tamanho Original          | SGD          | 0.0001        | 0.9        | Não            | Não             | 82.41%                    | 79%               |
| Modelo 7         | Tamanho Original          | Adam         | 0.001         | Não        | 0.001          | Não             | 92.33%                    | 93%               |
| Modelo 8         | Tamanho Original          | Adam         | 0.0001        | Não        | 0.001          | Não             | 98.49%                    | 97%               |
| Modelo 9         | Tamanho Original          | Adam         | 0.0001        | Não        | 0.00001        | 1,00E-08        | 99.49%                    | 98%               |
| Modelo 10        | 16.66% do Original        | AdamW        | 0.0001        | Não        | 0.00001        | 1,00E-07        | 99.50%                    | 96%               |
| <b>Modelo 11</b> | <b>33.33% do Original</b> | <b>AdamW</b> | <b>0.0001</b> | <b>Não</b> | <b>0.00001</b> | <b>1,00E-07</b> | <b>100%</b>               | <b>99%</b>        |
| Modelo 12        | Tamanho Original          | AdamW        | 0.0001        | Não        | 0.00001        | 1,00E-07        | 100%                      | 98%               |
| Modelo 13        | Tamanho Original          | AdamW        | 0.0001        | Não        | 0.001          | 1,00E-08        | 99.49%                    | 98%               |

**Tabela 2 – Configurações de hiper parâmetros avaliados durante o projeto**

A Tabela 2 apresenta diferentes configurações dos hiper-parâmetros e otimizadores utilizados em diferentes testes. Inicialmente avaliou-se o uso do otimizador SGD, buscando verificar como as modificações de *resize* das imagens, *learning rate* e a aplicação de um *weight decay* influenciavam nos resultados do modelo. Dentre os Modelos 1-6, o melhor modelo obtido foi o Modelo 4 que conseguiu atingir uma acurácia considerável nos testes de 94%, enquanto os outros ficaram por volta de 70%~90%. Esses modelos também apresentaram uma acurácia de validação razoável. A seguir apresenta-se na Figura 10 as acurárias e perdas desse Modelo 4 durante treinamento e validação. A Figura 11 apresenta a matriz de confusão dos testes desse modelo.

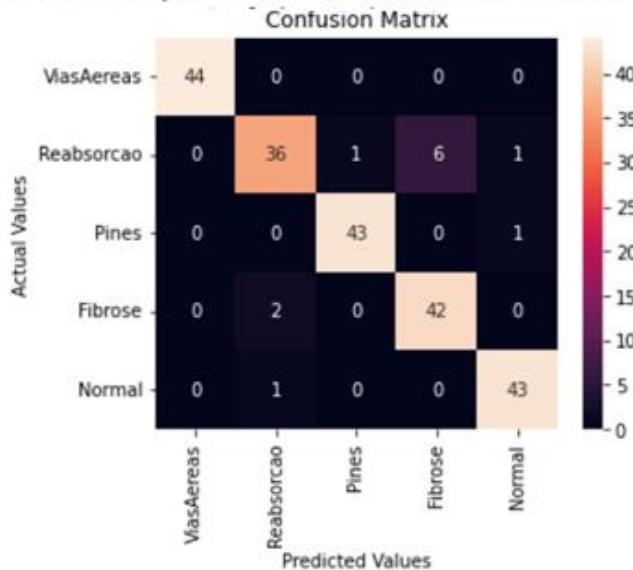


**Figura 6 – Acurácia e perda nas etapas de treino e validação do Modelo 4**

Right: 208

Wrong: 12

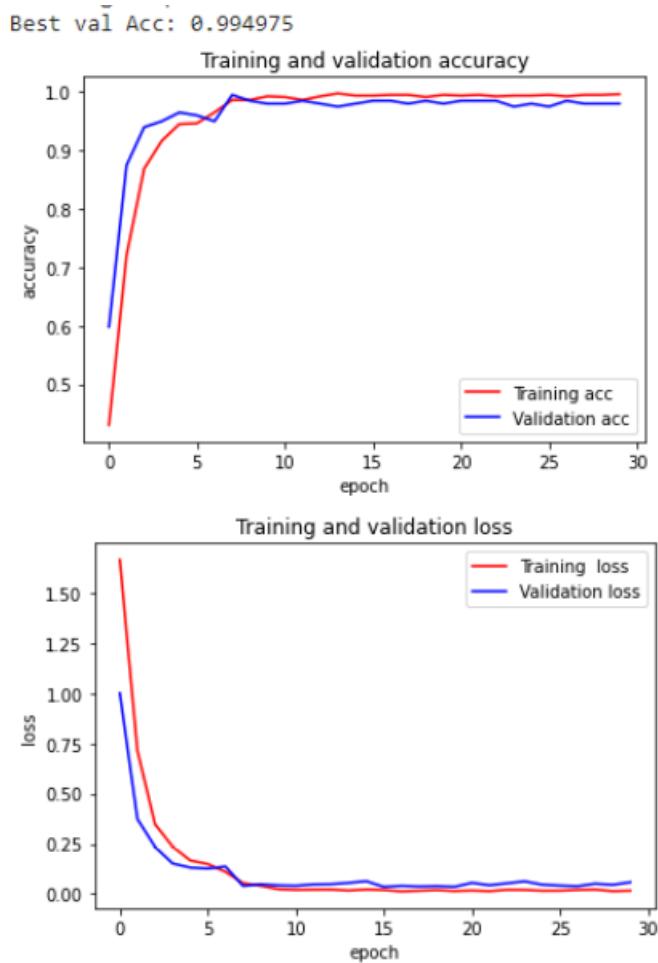
test accuracy over the whole test set is 94 %



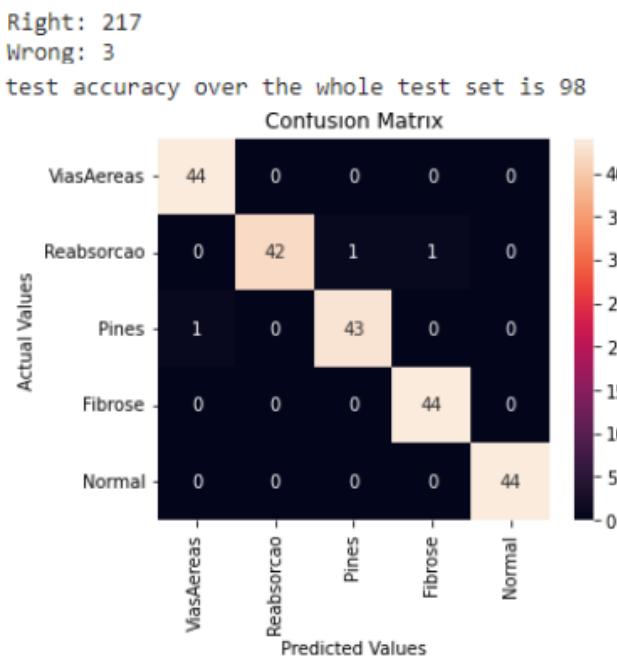
**Figura 7 – Matriz de confusão dos testes do Modelo 4**

Podemos observar na Figura 7 que os maiores problemas do Modelo 4 estiveram no erro da classificação das classes Fibrose e Reabsorção.

Em seguida, passou-se aos testes com outros otimizadores e o Adam já mostrou resultados melhores que o SGD, tanto em acurácia de testes quanto de validação, conforme podemos observar nos Modelos 7-9 da Tabela 2. Também é possível observar que do Modelo 7 para o 8 se verificou uma melhora no resultado devido à utilização de uma *learning rate* menor e que do Modelo 8 para o 9 os resultados foram os mesmos, dando a entender que a modificação no *weight decay* e a utilização do *epsilon* não influenciaram no desempenho do algoritmo de um modelo para o outro. Abaixo temos a Figura 8 com as acurácia e perdas do Modelo 9 durante treinamento e validação e a Figura 9 com a matriz de confusão dos testes desse modelo.



**Figura 8 – Acurácia e perda nas etapas de treino e validação do Modelo 9**

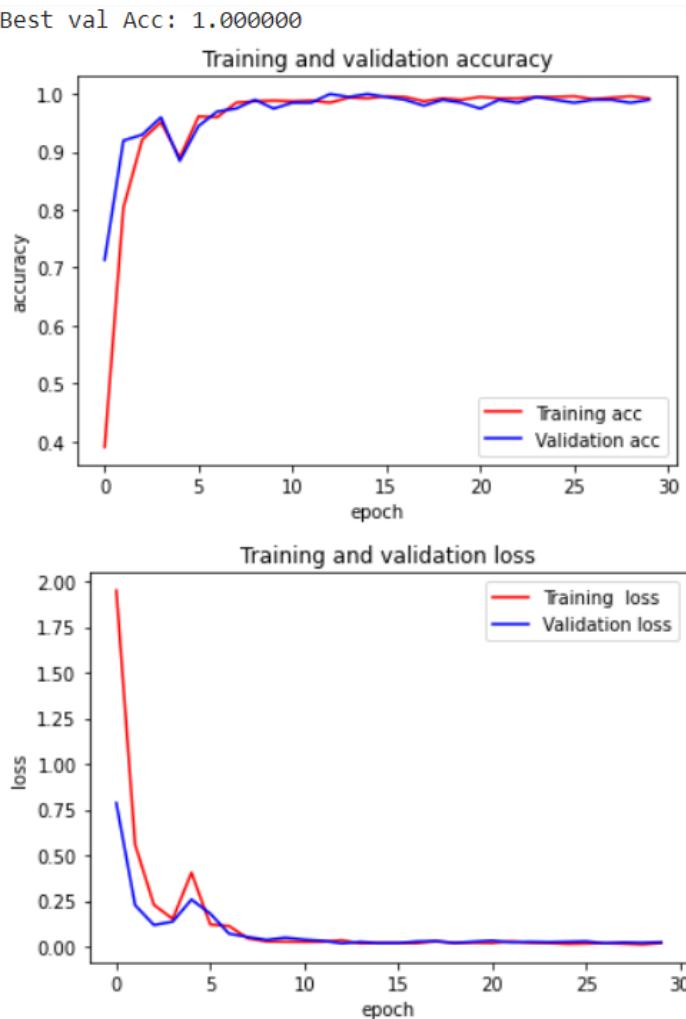


**Figura 9 – Matriz de confusão dos testes do Modelo 9**

Por fim, como o otimizador Adam já tinha obtido resultados excelentes, fez-se testes com o AdamW que é uma versão do Adam com pequena modificação, conforme já exposto anteriormente. A utilização do AdamW melhorou ainda mais a acurácia do algoritmo, conforme pode ser visto nos Modelos 10-13, onde todos obtiveram acurácia superior a 99% na validação e a 96% nos testes. Nesses modelos, foi utilizado a *learning rate* de 0.0001 que obteve os ótimos resultados nos modelos anteriores e foram modificadas configurações de *resize* de imagem, *weight decay* e *epsilon*. O modelo eleito como melhor foi o **Modelo 11** que atingiu 100% de acurácia na validação e 99% de acurácia nos testes, tendo sido selecionado como aquele a ser utilizado no final para o diagnóstico dos pacientes.

#### 6.4 Resultados do modelo escolhido

O Modelo 11 obteve os melhores resultados alcançados dentre aqueles investigados pelo trabalho. Em termos gerais, é possível observar na Figura 10 que ele teve um excelente desempenho, diminuindo progressivamente a perda durante o treino e a validação e aumentando a acurácia até um valor elevado, chegando a conseguir atingir 100% de acerto em algumas das validações. O único outro modelo que atingiu esse valor foi o Modelo 12 que utiliza as mesmas configurações do Modelo 11, mas com um tamanho de *resize* menor nas imagens de treinamento.



**Figura 10 – Acurácia e perda nas etapas de treino e validação do melhor modelo**

Esses resultados foram corroborados na etapa de testes onde o modelo também teve um ótimo desempenho. A matriz de confusão do mesmo é apresentada na Figura 11 onde pode-se observar que obteve uma acurácia de 99%, sendo esse considerado um resultado muito positivo que o levou a ser selecionado como o melhor modelo, pois atingiu esse valor além dos 100% na validação. O fato dele ter errado a classificação de somente 2 imagens nos dá uma visão otimista sobre a capacidade de sua utilização para esse tipo de diagnóstico, apesar da limitada quantidade de imagens de teste disponíveis.

Right: 218

Wrong: 2

The test accuracy over the whole test set is 99 %

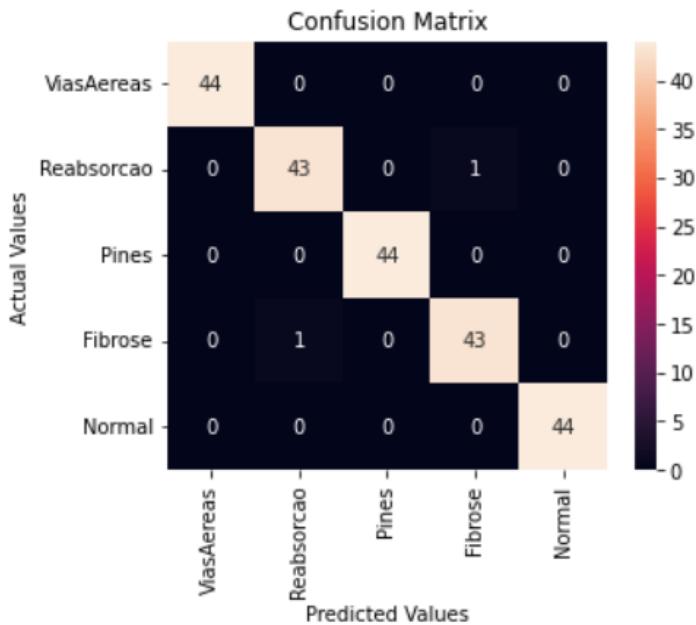


Figura 11 – Matriz de confusão do melhor modelo

## 7 Considerações Finais

Concluindo este trabalho, acredito que os objetivos foram alcançados. Os resultados foram bastante promissores e foi possível desenvolver o modelo proposto utilizando Redes Neurais Convolucionais e atingir uma excelente acurácia durante os testes como observado nos resultados. Esse tipo de modelo foi capaz de identificar as sequelas causadas nos pulmões proeminentes do contexto do pós-COVID.

No entanto, vale ressaltar que o número de imagens disponíveis e utilizado para treino, validação e teste do modelo está longe de ser o ideal, visto que normalmente se usam milhares de imagens. Dito isto, não se pode garantir que o modelo terá um bom desempenho no mundo real e que poderá efetivamente ser utilizado como uma ferramenta de apoio ao diagnóstico no dia a dia dos médicos e que se mantenha com a proporção de erros esperada. Seria necessário um número maior de dados de teste para validar que o modelo é, de fato, confiável para ser utilizado como método de apoio à decisão em exames de TC de pacientes que tiveram COVID-19.

## 8 Referências

Adam D., **15 million people have died in the pandemic, WHO says** , May, 2022.

Disponível em:

<https://www.nature.com/articles/d41586-022-01245-6>

Acesso em: 04/06/2022

Alghamdi H.S., Amoudi G., Elhag S., Saeedi K. & Nasser J., **Deep Learning Approaches for Detecting COVID-19 From Chest X-Ray Images: A Survey**, January, 2021.

Disponível em:

<https://ieeexplore.ieee.org/document/9335581>

Acesso em: 30/09/2021

Ampadu J., **Normalization in deep Learning**, 2021.

Disponível em:

<https://ai-pool.com/a/s/normalization-in-deep-learning#:~:text=Batch%20normalization%20is%20the%20most,training%20of%20deep%20neural%20networks>

Acesso em: 12/02/2022

Araújo F. H. A. & Fernandes L. H. S, **Lighting The Populational Impact of COVID-19 Vaccines In Brazil**, 2022.

Disponível em:

<https://www.worldscientific.com/doi/10.1142/S0218348X22500669>

Acesso em: 07/05/2022

Bollinger R., Ray S. & Maragakis L., **COVID Variants: What You Should Know**, 2022.

Disponível em:

<https://www.hopkinsmedicine.org/health/conditions-and-diseases/coronavirus/a-new-strain-of-coronavirus-what-you-should-know>

Acesso em: 08/05/2022

Brownlee J., **A Gentle Introduction to Transfer Learning for Deep Learning**, December, 2017.

Disponível em:

<https://machinelearningmastery.com/transfer-learning-for-deep-learning/>

Acesso em: 01/10/2021

Brownlee J., **How Do Convolutional Layers Work in Deep Learning Neural Networks?**, April, 2019.

Disponível em:

<https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/>

Acesso em: 14/06/2022

Chen D. S. & Jain R. C., **A robust backpropagation learning algorithm for function approximation**, 1994.

Disponível em:

<https://ieeexplore.ieee.org/document/286917>

Acesso em: 08/04/2022

Corum J. & Zimmer C., **Tracking Omicron and Other Coronavirus Variants**, 2022.

Disponível em:

<https://www.nytimes.com/interactive/2021/health/coronavirus-variant-tracker.html>

Acesso em: 07/05/2022

Cucinotta D. & Vanelli M, **Who declares Covid-19 a pandemic**, Acta Biomed, v. 91, n. 91, p. 157-160, 2020.

Disponível em:

[https://pubmed.ncbi.nlm.nih.gov/32191675/#:~:text=The%20World%20Health%20Organization%20\(WHO,a%20global%20pandemic%20\(1\).](https://pubmed.ncbi.nlm.nih.gov/32191675/#:~:text=The%20World%20Health%20Organization%20(WHO,a%20global%20pandemic%20(1).)

Acesso em: 02/05/2022

Das D., Santosh KC. & Pal U., **Truncated inception net: COVID-19 outbreak screening using chest X-rays**, June 2020.

Disponível em:

<https://link.springer.com/article/10.1007/s13246-020-00888-x>

Acesso em: 30/11/2021

Data Science Academy, **Deep Learning Book**, 2022.

Disponível em:

<https://www.deeplearningbook.com.br/camadas-de-pooling-em-redes-neurais-convolucionais/>

Acesso em: 14/06/2022

Elijah Koech K., **Cross Entropy Loss Function**, 2020.

Disponível em:

<https://towardsdatascience.com/cross-entropy-loss-function-f38c4ec8643e>

Acesso em: 14/03/2022

Fedorenko Y. S., Gapanyuk Y. & Minakova S., **The Analysis of Regularization in Deep Neural Networks Using Metagraph Approach**, 2018. Advances in Neural Computation, Machine Learning, and Cognitive Research.

NEUROINFORMATICS 2017. Studies in Computational Intelligence, vol 736.

Disponível em:

[https://link.springer.com/chapter/10.1007/978-3-319-66604-4\\_1](https://link.springer.com/chapter/10.1007/978-3-319-66604-4_1)

Acesso em: 15/05/2022

Goodfellow I., Bengio Y. & Courville A. ,**Deep learning**, MIT press 2016.

Govardhan J., Deepti M., Daksh T. & Madhup K. M., **A deep learning approach to detect Covid-19 coronavirus with X-Ray images**, October 2020.

Disponível em:

<https://www.sciencedirect.com/science/article/pii/S0208521620301005?via%3Dihub>

Acesso em: 30/11/2021

Haykin S., **Neural Networks and Learning Machines 3rd Edition**, Upper Saddle River, NJ: Pearson Education, 2009.

He K., Zhang X., Ren S. & Sun J., **Deep Residual Learning for Image Recognition**, December 2015.

Disponível em:

<https://arxiv.org/pdf/1512.03385.pdf>

Acesso em: 26/02/2022

Huang L., Yao Q., Gu X., Wang Q., Ren L., Wang Y., et al., **1-year outcomes in hospital survivors with COVID-19: a longitudinal cohort study**, August 2021.

Disponível em:

[https://www.thelancet.com/journals/lancet/article/PIIS0140-6736\(21\)01755-4/fulltext#%20](https://www.thelancet.com/journals/lancet/article/PIIS0140-6736(21)01755-4/fulltext#%20)

Acesso em: 03/05/2022

JavadiMoghaddam S. & Gholamalinejad H., **A novel deep learning based method for COVID-19 detection from CT image**, 2021. *Biomedical Signal Processing and Control*, 70, 102987.

Disponível em:

<https://www.sciencedirect.com/science/article/pii/S174680942100584X>

Acesso em: 04/05/2022

Kessy A., Lewin A. & Strimmer K., **Optimal Whitening and Decorrelation**, 2018. *The American Statistician*, vol. 72, no. 4, pp. 309-314.

Disponível em:

<https://www.tandfonline.com/doi/abs/10.1080/00031305.2016.1277159?journalCode=utas20>

Acesso em: 01/12/2021

Khan A., Sohail A., Zahoor U., Saeed Qureshi A., **A Survey of the Recent Architectures of Deep Convolutional Neural Networks**, *Artificial Intelligence Review*, 2020.

Disponível em:

<https://arxiv.org/abs/1901.06032>

Acesso em: 15/04/2022

Kingma D. P. & Ba J., **Adam: A Method for Stochastic Optimization**, 2015.

Disponível em:

<https://arxiv.org/abs/1412.6980>

Acesso em: 20/04/2022

Krizhevsky A., Sutskever I. & Hinton G. E, **Imagenet classification with deep convolutional neural networks**, 2012 . *Adv. Neural Inf. Process. Syst.*, vol. 25, pp. 1097- 1105.

Disponível em:

<https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68>

[c45b-Paper.pdf](#)

Acesso em: 08/04/2022

Kyla C., Iteen A., Nicolini R. & Auten J., **COVID-19 pneumonia with hemoptysis: Acute segmental pulmonary emboli associated with novel coronavirus infection**, July 2020.

Disponível em:

<https://www.sciencedirect.com/science/article/pii/S0735675720302394>

Acesso em: 28/11/2021

Landora F. N., Han S., Moskewicz M. W., Ashraf K., Dally W. J. & Keutzer K, **SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size**, 2017.

Disponível em:

<https://arxiv.org/abs/1602.07360v4>

Acesso em: 26/02/2022

LeCun Y., Bengio Y. & Hinton G., **Deep Learning**, May 2015.

Disponível em:

<https://www.nature.com/articles/nature14539>

Acesso em: 28/11/2021

LeCun Y., Haffner P., Bottou L. & Bengio Y., **Object recognition with gradient-based learning. InShape, contour and grouping in computer vision** 1999 (pp. 319-345). Springer, Berlin, Heidelberg.

Liou C.-Y., Cheng W.-C., Liou J.-W. & Liou D.-R, **Autoencoder for words** , 2014.

Disponível em:

<https://www.sciencedirect.com/science/article/abs/pii/S0925231214003658>

Acesso em: 02/12/2021

Loshchilov I. & Hutter F., **Decoupled Weight Decay Regularization**, 2019.

Disponível em:

<https://paperswithcode.com/paper/decoupled-weight-decay-regularization>

Acesso em: 22/04/2022

Master's in Data Science, **What Is the Difference Between Bias and Variance?**

Disponível em:

<https://www.mastersindatascience.org/learning/difference-between-bias-and-variance/>

Acesso em: 30/05/2022

Minaee S., Kafieh R., Sonka M., Yazdani S. & Soufi G. J., **Deep-COVID: Predicting COVID-19 From Chest X-Ray Images Using Deep Transfer Learning**, 2021. Medical Image Analysis.

Disponível em:

<https://arxiv.org/pdf/2004.09363.pdf>

Acesso em: 14/03/2022

Mogami R., Costa A., Filho A., Carvalho R., Chantong C., Gianella C., Almeida S., Carlos F., Koifman B., Célia A., Gustavo J., Thiago M., da Silva Bessa H., Peres dos Santos G., Zangerolame de Carvalho B. & da Silva Passos G., **The Importance of Radiological Patterns and Small Airway Disease in Long-Term Follow-Up of Postacute COVID-19: A Preliminary Study**, May 2022.

Disponível em:

<https://www.hindawi.com/journals/rrp/2022/7919033/>

Acesso em: 12/05/2022

Ng A., Ngiam J., Yu Foo C., Mai Y., Suen C., Coates A., Mass A., Hannun A., HuvalB., Wang T. & Tandon S., **Stanford Unsupervised Feature Learning and Deep Learning Tutorial**, 2013. .

Disponível em:

<http://ufldl.stanford.edu/tutorial/>

Acesso em: 27/09/2021

Ortiz A., Trivedi A., Desbiens J., Blazes M., Robinson C., Gupta S., Dodhia R., Bhatraju P. K., Conrad Liles W., Lee A & Lavista Ferres J. M., **Effective deep learning approaches for predicting COVID-19 outcomes from chest computed tomography volumes**, February 2022.

Disponível em:

<https://www.nature.com/articles/s41598-022-05532-0>

Acesso em: 14/03/2022

Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G., Killeen T.,

Lin Z., Natalia Antiga L., Desmaison A., Yang E., DeVitoZ., Raison M., Tejani A., Chilamkurthy S., Steiner B., Lu B. & Junjie Chintala S., **PyTorch: An Imperative Style, High-Performance Deep Learning Library**, 2019.

Disponível em:

<https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf>

Acesso em: 29/11/2021

Poojary R. & Pai A., **Comparative Study of Model Optimization Techniques in Fine-Tuned CNN Models**, 2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA), 2019, pp. 1-4, doi: 10.1109/ICECTA48151.2019.8959681.

Disponível em:

<https://ieeexplore.ieee.org/document/8959681>

Acesso em: 11/05/2022

Ruder S., **An overview of gradient descent optimization algorithms**, 2016. ArXiv Preprint ArXiv:1609.04747.

Disponível em:

<https://arxiv.org/abs/1609.04747>

Acesso em: 21/04/2022

Saeedan F., Weber N., Goesele M. & Roth S., **Detail-Preserving Pooling in Deep Networks**, April 2018.

Disponível em:

<https://arxiv.org/abs/1804.04076>

Acesso em: 01/12/2021

Sedik A., Iliyasu AM., Abd El-Rahiem B., Abdel Samea ME., Abdel-Raheem A., Hammad M., Peng J., Abd El-Samie FE. & Abd El-Latif AA., **Deploying Machine and Deep Learning Models for Efficient Data-Augmented Detection of COVID-19 Infections**, July 2020.

Disponível em:

<https://pubmed.ncbi.nlm.nih.gov/32708803/>

Acesso em: 29/11/2021

Sethi R., Mehrotra M. & Sethi D., **Deep Learning based Diagnosis**

**Recommendation for COVID-19 using Chest X-Rays Images**, July 2020.

Disponível em:

<https://ieeexplore.ieee.org/document/9183278>

Acesso em: 30/11/2021

Shah F., Joy S., Ahmed F., Humaira M., Ami A., Paul S. & Jim M., **A**

**Comprehensive Survey of COVID-19 Detection Using Medical Images**, June, 2020. Disponível em:

[https://www.researchgate.net/publication/343264351\\_A\\_Comprehensive\\_Survey\\_of\\_COVID-19\\_Detection\\_Using\\_Medical\\_Images](https://www.researchgate.net/publication/343264351_A_Comprehensive_Survey_of_COVID-19_Detection_Using_Medical_Images)

Acesso em: 30/09/2021

Shorten C., **Unsupervised Feature Learning**, February, 2019.

Disponível em:

<https://towardsdatascience.com/unsupervised-feature-learning-46a2fe399929>

Acesso em: 27/09/2021

Shorten C. & Khoshgoftaar T., **A survey on Image Data Augmentation for Deep Learning**, 2019.

Disponível em:

<https://jurnalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0>

Acesso em: 05/05/2022

Simonyan K. & Zisserman A., **Very deep convolutional networks for large-scale image recognition**, 2014.

Disponível em:

<https://arxiv.org/abs/1409.1556>

Acesso em: 30/03/2022

Soria Olivas E. & Marin Guerrero J. D., **Handbook of research on machine learning applications and trends**, 2010. Hershey, Pa.: IGI Global (701 E. Chocolate Avenue, Hershey, Pennsylvania, 17033, USA),

Torrey J. & Shavlik J., **Transfer Learning**, 2009.

Disponível em:

<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.146.1515>

Acesso em: 30/11/2021

Varshni D., Thakral K., Agarwal L., Nijhawan R. & Mittal A., **Pneumonia Detection Using CNN based Feature Extraction**, October ,2019. IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2019.

Disponível em:

<https://ieeexplore.ieee.org/abstract/document/8869364>

Acesso em: 01/10/2021