

TRABALHO DE GRADUAÇÃO

**DETECÇÃO DA COVID-19 EM IMAGENS DE RAIO-X:
CONSTRUINDO UM NOVO MODELO DE APRENDIZADO
PROFUNDO UTILIZANDO AUTOML**

Igor Raposo Oliveira Beduin

Brasília, maio de 2021



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

**DETECÇÃO DA COVID-19 EM IMAGENS DE RAIOS-X:
CONSTRUINDO UM NOVO MODELO DE APRENDIZADO
PROFUNDO UTILIZANDO AUTOML**

Igor Raposo Oliveira Beduin

*Relatório submetido como requisito parcial de obtenção
de grau de Engenheiro de Controle e Automação*

Banca Examinadora

Prof. Flávio de Barros Vidal, CIC/UnB
Orientador

MSc. Ana Paula Gonçalves Soares de Almeida,
PPMEC/UnB
Coorientadora

Prof. Marcelino Monteiro de Andrade,
FGA/UnB
Examinador interno

Prof. Marcus Vinícius Chaffim Costa,
FGA/UnB
Examinador interno

Brasília, maio de 2021

FICHA CATALOGRÁFICA

IGOR, RAPOSO OLIVEIRA BEDUIN

Detecção da Covid-19 em imagens de Raio-X: Construindo um novo modelo de aprendizado profundo utilizando AutoML.

[Distrito Federal] 2021.

x, 78p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2021). Trabalho de Graduação – Universidade de Brasília. Faculdade de Tecnologia.

1. COVID-19

2. Aprendizado Profundo

3. AutoML

4. ResNet

I. Mecatrônica/FT/UnB

II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

RAPOSO, IGOR. O. B., (2021). Detecção da Covid-19 em imagens de Raio-X: Construindo um novo modelo de aprendizado profundo utilizando AutoML. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-*n*°005, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 78p.

CESSÃO DE DIREITOS

AUTOR: Igor Raposo Oliveira Beduin

TÍTULO DO TRABALHO DE GRADUAÇÃO: Detecção da Covid-19 em imagens de Raio-X: Construindo um novo modelo de aprendizado profundo utilizando AutoML.

GRAU: Engenheiro

ANO: 2021

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

Igor Raposo Oliveira Beduin.

Quadra 8 Conjunto 2 Lote 2, Casa G, Bairro Park Way.

71740-802 Brasília – DF – Brasil.

Dedicatória

Este trabalho é dedicado à todas as vítimas da COVID-19 e todos aqueles que sofreram de alguma forma por essa doença.

Igor Raposo Oliveira Beduin

Agradecimentos

Agradeço primeiramente à minha família e amigos que tanto me deram forças ao longo de minha graduação. Não chegamos a lugar nenhum sozinhos e nenhum sucesso faz sentido se não podemos compartilhá-lo com aqueles que nos dão suporte. À todos os membros e ex-membros da equipe UnBall que tive o prazer de trabalhar, sou grato por tudo que aprendi e por todos os momentos de riso e tristeza que passamos juntos. Fazer parte dessa equipe foi uma das melhores experiências que vivenciei na UnB e levarei para sempre o orgulho de ter participado de sua história. Aos meus colegas de curso, da turma 35 de Mecatrônica e todos aqueles que tive o prazer de conviver ao longo de minha jornada na UnB. Agradeço pelas incontáveis ajudas, estudos, conversas e conselhos: Manoel Vieira, Samuel Venzi, Caio Campos, Raphael Barbosa, Marcus Dutra, Pedro Perruci, André Mariano, Lucas Schiavini, Thúlio Noslen, Tiago Gallo, Sofia Disegna, Anderson Barros, Ian Moura, Bruno Andregretti, Vitor Duarte, Pedro Matheus e tantos outros que me ajudaram a chegar até aqui e tive o prazer de dividir tantos momentos maravilhosos. Não posso deixar de agradecer também aos meus amigos de longa data que a tantos anos me acompanham e me inspiram a ser uma pessoa melhor: Henrique Barbosa, Isaac Marques, José Eduardo e Luísa Guedes. Por fim, mas não menos importante, agradeço ao Prof. Flávio Vidal e a Ana Paula Almeida pelas puxadas de orelha e correções sim, mas também por sempre acreditaram em mim, mesmo quando nem mesmo eu acreditava.

Igor Raposo Oliveira Beduin

RESUMO

No início de 2020 o mundo foi surpreendido pela nova pandemia global de COVID-19. Sendo uma doença altamente transmissível, a testagem em massa da população ainda é uma das medidas mais importantes na contenção da disseminação da doença. Baseado então na recente onda de Redes Neurais Convolucionais com excelentes resultados em tarefas de classificação de imagens, este trabalho propõe um novo modelo residual gerado pelo AutoKeras, o AutoResCovidNet, para classificação multi-classe baseado em imagens de Raio-X de pessoas saudáveis, com pneumonia e diagnóstico confirmado de COVID-19, comparando ao fim com outros resultados da literatura. O trabalho apresentado neste manuscrito tem como objetivo mostrar que um modelo funcional base pode ser obtido a partir de métodos de AutoML. Assim, este trabalho descreve todas as etapas de projeto de um modelo de Aprendizado Profundo. Foi reunido um conjunto de dados com mais 12 mil imagens classificadas em "Normal", "Pneumonia" e "COVID-19". Este conjunto completo foi utilizado para busca de modelos com AutoKeras na família das ResNets, utilizando técnicas de pré-processamento a fim de se melhorar a generalização das características. O melhor modelo obtido da saída do AutoKeras foi então utilizado para treinamento do zero aplicando o processo de validação cruzada com *split* de 80%/20%. Os treinos de modelos na validação cruzada foram feitos com os otimizadores Adam e SGD, utilizando tanto imagens *raw* quanto imagens com Aumento de Dados. Os modelos foram analisados qualitativamente e quantitativamente respectivamente por curvas ROC e Matrizes de Confusão. Nas duas análises os modelos treinados com conjunto de dados *raw* tiveram desempenho superior, sendo que o modelo SGD_*raw* foi o melhor possuindo maior *F1-Score*. Foi observado também que o pré-processamento para Aumento de Dados foi bastante prejudicial dificultando a generalização das características, indicando que as operações foram demasiadamente agressivas. Apesar de não conseguir alcançar os modelos da literatura, os bons resultados de classificação de duas das três classes indicam que ajustes finos de treinamento e pré-processamento das imagens podem colocar o modelo gerado no patamar do estado-da-arte.

Palavras Chave: COVID-19, Raio-X, Aprendizado Profundo, AutoML, Redes Neurais Convolucionais

ABSTRACT

In early 2020 the world was surprised by the new global pandemic of COVID-19. Being a very contagious disease, mass testing of the population is still one of the most important measures to contain this disease spreadness. Hence, based on the recent wave of Convolutional Neural Networks achieving great results in image classification tasks, this work proposes a new residual model generated by AutoKeras, AutoResCovidNet, for multi-class classification of healthy, pneumonia and COVID-19 positively diagnosed people. At the end the AutoResCovidNet's results are compared with the reference in bibliography. This presented work aims to show that it is possible to obtain a functional base model with AutoML techniques. Thus this work describes all stages of developing a Deep Learning model. It was obtained a dataset with more than 12 thousand images labeled as "Normal", "Pneumonia" and "COVID-19". This whole dataset was used in the model search with AutoKeras using preprocessing techniques to improve features generalization. The best model obtained with AutoKeras was found with a restricted search on the ResNets family and then used for a from-scratch cross-validation training with a 80%/20% split. The cross-validation models were trained with Adam and SGD optimizers using both raw and augmented images. These models were evaluated doing a qualitative and quantitative analysis by ROC curves and Confusion Matrices. In both analyses the raw trained models achieved better results, being the SGD_raw the best one with higher F1-Score. It was also observed that the preprocessing used for Data Augmentation was not beneficial for the trainings, indicating that the preprocessing operations were too aggressive. Despite of not being able to achieve the best academic models, the good classification results in two of three classes indicates that fine tunings in training and preprocessing might put the generated model on the state-of-art level.

Keywords: COVID-19, X-Ray, Deep Learning, AutoML, Convolutional Neural Networks

SUMÁRIO

1	Introdução	1
1.1	CONTEXTUALIZAÇÃO	1
1.2	DESCRIÇÃO DO PROBLEMA E OBJETIVOS	2
1.3	PÚBLICO ALVO	3
1.4	APRESENTAÇÃO DO MANUSCRITO	3
2	Conceitos Fundamentais e Revisão Bibliográfica	4
2.1	FUNDAMENTAÇÃO TEÓRICA	4
2.1.1	APRENDIZADO PROFUNDO	4
2.1.2	REDES NEURAS CONVOLUCIONAIS	6
2.1.3	REDES RESIDUAIS	7
2.1.4	AUTOML	9
2.2	MÉTRICAS DE ANÁLISE	12
2.2.1	MEDIDAS	12
2.2.2	MATRIZ DE CONFUSÃO	13
2.2.3	CURVA ROC	13
2.3	TRABALHOS RELACIONADOS	14
2.3.1	RAIO-X	15
2.3.2	TOMOGRAFIA COMPUTADORIZADA (TC)	19
3	Materiais e Métodos	21
3.1	CONJUNTO DE DADOS	21
3.1.1	COVID-19 IMAGE DATA COLLECTION	22
3.1.2	FIGURE 1 COVID-19 CHEST X-RAY DATASET INITIATIVE	22
3.1.3	ACTUALMED COVID-19 CHEST X-RAY DATASET INITIATIVE	23
3.1.4	COVID-19 CHEST X-RAY DATABASE	23
3.1.5	RSNA PNEUMONIA DETECTION CHALLENGE	24
3.1.6	CONJUNTO FINAL	24
3.2	AUTOML	25
3.2.1	PRÉ-PROCESSAMENTO	26
3.2.2	AUMENTO DE DADOS	26
3.2.3	RESTRIÇÃO DO ESPAÇO DE BUSCA	27
3.2.4	TREINO DOS MODELOS DURANTE A BUSCA	27

3.2.5	MODELO FINAL.....	28
3.3	TREINAMENTO	28
3.3.1	PRÉ PROCESSAMENTO.....	28
3.3.2	DESBALANCEAMENTO DE DADOS.....	28
3.3.3	TREINAMENTO E VALIDAÇÃO CRUZADA	28
3.4	AVALIAÇÃO	30
3.5	<i>HARDWARE</i> E <i>SOFTWARE</i> UTILIZADOS	30
4	Resultados.....	31
4.1	BUSCA E TREINAMENTO NO AUTOKERAS.....	31
4.2	MODELO OBTIDO	33
4.3	TREINAMENTO DE VALIDAÇÃO CRUZADA	35
4.4	ANÁLISE QUALITATIVA	38
4.4.1	CONJUNTO DE DADOS <i>RAW</i>	38
4.4.2	CONJUNTO DE DADOS COM AD	38
4.4.3	VALIDAÇÃO.....	43
4.5	ANÁLISE QUANTITATIVA	44
4.5.1	CONJUNTO DE DADOS <i>RAW</i>	44
4.5.2	CONJUNTO DE DADOS COM AD	44
4.5.3	VALIDAÇÃO.....	50
4.6	COMPARAÇÃO COM OUTROS MODELOS E ABORDAGENS.....	52
4.7	DISCUSSÃO	53
5	Conclusões.....	56
5.1	TRABALHOS FUTUROS.....	57
	REFERÊNCIAS BIBLIOGRÁFICAS	58
	Anexos.....	64
I.1	IMAGENS DE TREINAMENTOS.....	65
I.1.1	BUSCA DE MODELOS NO AUTOKERAS	65
I.1.2	TREINAMENTOS FINAIS COM VALIDAÇÃO CRUZADA	72

LISTA DE TERMOS E ACRÔNIMOS

Acc	Acurácia
AD	Aumento de Dados
AM	Aprendizado de Máquina
AUC	<i>Area Under Curve</i>
BAN	Busca de Arquitetura Neural
F1	<i>F1-Score</i>
FN	Falsos Negativos
FP	Falsos Positivos
FPN	<i>Feature Pyramid Network</i>
IA	Inteligência Artificial
MLP	<i>Multilayer Perceptron</i>
Prec	Precisão
RNC	Redes Neurais Convolucionais
RNCs	Redes Neurais
RNP	Rede Neural Profunda
RT-PCR	<i>Reverse transcription polymerase chain reaction</i>
Sens	Sensibilidade
SGD	<i>Stochastic Gradient Descent</i>
TC	Tomografia Computadorizada
VN	Verdadeiros Negativos
VP	Verdadeiros Positivos

LISTA DE FIGURAS

2.1	À esquerda, um conjunto de dados em representação cartesiana. À direita, o mesmo conjunto de dados em representação polar. Esta imagem ilustra como a representação dos dados influencia diretamente na definição das áreas de classificação em algoritmos de Aprendizado de Máquina. Figura original extraída do livro de Goodfellow (2016) [1].	5
2.2	Diagrama de Venn que mostra como Aprendizado Profundo é um tipo de Aprendizado Representacional, que por sua vez é um tipo de Aprendizado de Máquina e este é apenas uma sub-área da Inteligência Artificial. Imagem adaptada do trabalho original de Goodfellow (2016) [1].	6
2.3	Representação de como uma classificação é feita baseada em abstrações complexas que são baseadas em características mais simples de uma imagem. Imagem adaptada do trabalho original de Goodfellow (2016) [1].	8
2.4	Bloco de aprendizado residual. Figura adaptada da publicação original de He <i>et al.</i> (2015) [2].	8
2.5	Diagrama de representação de um <i>pipeline</i> típico de AutoML. Imagem extraída do artigo original de He <i>et al.</i> (2021) [3].	9
2.6	Diagrama de representação de um processo de Busca de Arquitetura Neural, dividido em dois estágios principais de fluxo bilateral: geração do modelo e estimação do modelo. Estas etapas são iterativas e podem se repetir diversas vezes. Imagem extraída do artigo original de He <i>et al.</i> (2021) [3].	10
2.7	Exemplo dos processos do AutoKeras em cada um dos módulos. Seguindo o fluxo das setas: API, Buscador, módulo de Grafos, Treinador e módulo de Armazenamento de Modelos. Imagem adaptada do artigo original de Jin <i>et al.</i> (2019) [4].	11
2.8	Exemplo de Matriz de Confusão. No eixo Y nós vemos as classes verdadeiras e no eixo X as classes inferidas pelo modelo. Imagem adaptada do artigo original de Fawcett [5].	13
2.9	Exemplo de curva ROC. No eixo Y nós vemos a taxa de VP e no eixo X a taxa de FP. Imagem adaptada do artigo original de Fawcett [5].	14
2.10	Grafo de artigos relacionados que citam o trabalho de Wang <i>et al.</i>	15
3.1	Fluxograma da metodologia.	21
3.2	Imagens de Raio-X frontais extraídas do conjunto de dados original COVID-19 image data collection [6].	22

3.3	Imagens de Raio-X frontais extraídas do conjunto de dados original Figure 1 COVID-19 Chest X-ray Dataset Initiative [7].	23
3.4	Imagens de Raio-X frontais extraídas do conjunto de dados original Actualmed COVID-19 Chest X-ray Dataset Initiative [8].	23
3.5	Imagens de Raio-X frontais extraídas do conjunto de dados original COVID-19 Chest X-ray Database [9].	24
3.6	Imagens de Raio-X frontais extraídas do conjunto de dados original RSNA Pneumonia Detection Challenge [10].	25
3.7	Exemplos de imagens do COVIDx-m com diferentes níveis de rotação e de <i>zoom</i>	27
3.8	Fluxo de separação do conjunto de dados de treino para sub-conjuntos de validação cruzada, e teste para validação dos modelos.	29
4.1	Gráfico de acurácia em validação dos 20 modelos testados pelo AutoKeras ao longo das épocas de treinamento.	32
4.2	Gráfico de <i>loss</i> em validação dos 20 modelos testados pelo AutoKeras ao longo das épocas de treinamento.	32
4.3	Resumo do modelo final obtido pelo AutoKeras.	33
4.4	Representação das primeiras camadas do modelo final obtido.	34
4.5	Representação das últimas camadas do modelo final obtido.	35
4.6	Gráficos de validação dos 5 <i>folds</i> treinados no modelo <i>Adam_raw</i> ao longo das épocas de treinamento.	36
4.7	Gráficos de validação dos 5 <i>folds</i> treinados no modelo <i>SGD_raw</i> ao longo das épocas de treinamento.	36
4.8	Gráficos de validação dos 5 <i>folds</i> treinados no modelo <i>Adam_augmented</i> ao longo das épocas de treinamento.	37
4.9	Gráficos de validação dos 5 <i>folds</i> treinados no modelo <i>SGD_augmented</i> ao longo das épocas de treinamento.	37
4.10	Curvas ROC de cada um dos <i>folds</i> treinados para Adam_raw . As classes nas cores verde, roxo e vermelho são respectivamente "Normal", "Pneumonia" e "COVID-19"..	39
4.11	Curvas ROC de cada um dos <i>folds</i> treinados para SGD_raw . As classes nas cores verde, roxo e vermelho são respectivamente "Normal", "Pneumonia" e "COVID-19"..	40
4.12	Curvas ROC de cada um dos <i>folds</i> treinados para Adam_augmented . As classes nas cores verde, roxo e vermelho são respectivamente "Normal", "Pneumonia" e "COVID-19".	41
4.13	Curvas ROC de cada um dos <i>folds</i> treinados para SGD_augmented . As classes nas cores verde, roxo e vermelho são respectivamente "Normal", "Pneumonia" e "COVID-19".	42
4.14	Matrizes de Confusão de cada um dos <i>folds</i> treinados para Adam_raw . Quanto mais escuro, maior quantidade de dados, quanto mais claro, menor quantidade de dados. As figuras são: a) <i>fold</i> 1; b) <i>fold</i> 2; c) <i>fold</i> 3; d) <i>fold</i> 4; e) <i>fold</i> 5.	46

4.15	Matrizes de Confusão de cada um dos <i>fold</i> s treinados para SGD_raw . Quanto mais escuro, maior quantidade de dados, quanto mais claro, menor quantidade de dados. A As figuras são: a) <i>fold</i> 1; b) <i>fold</i> 2; c) <i>fold</i> 3; d) <i>fold</i> 4; e) <i>fold</i> 5.....	47
4.16	Matrizes de Confusão de cada um dos <i>fold</i> s treinados para Adam_augmented . Quanto mais escuro, maior quantidade de dados, quanto mais claro, menor quantidade de dados. A As figuras são: a) <i>fold</i> 1; b) <i>fold</i> 2; c) <i>fold</i> 3; d) <i>fold</i> 4; e) <i>fold</i> 5.....	48
4.17	Matrizes de Confusão de cada um dos <i>fold</i> s treinados para SGD_augmented . Quanto mais escuro, maior quantidade de dados, quanto mais claro, menor quantidade de dados. A As figuras são: a) <i>fold</i> 1; b) <i>fold</i> 2; c) <i>fold</i> 3; d) <i>fold</i> 4; e) <i>fold</i> 5.....	49
1	Gráficos de acurácia e <i>loss</i> do modelo 1 testado pelo Autokeras.	65
2	Gráficos de acurácia e <i>loss</i> do modelo 2 testado pelo Autokeras.	65
3	Gráficos de acurácia e <i>loss</i> do modelo 3 testado pelo Autokeras.	65
4	Gráficos de acurácia e <i>loss</i> do modelo 4 testado pelo Autokeras.	66
5	Gráficos de acurácia e <i>loss</i> do modelo 5 testado pelo Autokeras.	66
6	Gráficos de acurácia e <i>loss</i> do modelo 6 testado pelo Autokeras.	66
7	Gráficos de acurácia e <i>loss</i> do modelo 7 testado pelo Autokeras.	67
8	Gráficos de acurácia e <i>loss</i> do modelo 8 testado pelo Autokeras.	67
9	Gráficos de acurácia e <i>loss</i> do modelo 9 testado pelo Autokeras.	67
10	Gráficos de acurácia e <i>loss</i> do modelo 10 testado pelo Autokeras.....	68
11	Gráficos de acurácia e <i>loss</i> do modelo 11 testado pelo Autokeras.....	68
12	Gráficos de acurácia e <i>loss</i> do modelo 12 testado pelo Autokeras.....	68
13	Gráficos de acurácia e <i>loss</i> do modelo 13 testado pelo Autokeras.....	69
14	Gráficos de acurácia e <i>loss</i> do modelo 14 testado pelo Autokeras.....	69
15	Gráficos de acurácia e <i>loss</i> do modelo 15 testado pelo Autokeras.....	69
16	Gráficos de acurácia e <i>loss</i> do modelo 16 testado pelo Autokeras.....	70
17	Gráficos de acurácia e <i>loss</i> do modelo 17 testado pelo Autokeras.....	70
18	Gráficos de acurácia e <i>loss</i> do modelo 18 testado pelo Autokeras.....	70
19	Gráficos de acurácia e <i>loss</i> do modelo 19 testado pelo Autokeras.....	71
20	Gráficos de acurácia e <i>loss</i> do modelo 20 testado pelo Autokeras.....	71
21	Gráficos de acurácia e <i>loss</i> da <i>fold</i> 1 de validação cruzada do modelo Adam_raw. ...	72
22	Gráficos de acurácia e <i>loss</i> da <i>fold</i> 2 de validação cruzada do modelo Adam_raw. ...	72
23	Gráficos de acurácia e <i>loss</i> da <i>fold</i> 3 de validação cruzada do modelo Adam _{raw}	72
24	Gráficos de acurácia e <i>loss</i> da <i>fold</i> 4 de validação cruzada do modelo Adam_raw. ...	73
25	Gráficos de acurácia e <i>loss</i> da <i>fold</i> 5 de validação cruzada do modelo Adam_raw. ...	73
26	Gráficos de acurácia e <i>loss</i> da <i>fold</i> 1 de validação cruzada do modelo SGD_raw.....	73
27	Gráficos de acurácia e <i>loss</i> da <i>fold</i> 2 de validação cruzada do modelo SGD_raw.....	74
28	Gráficos de acurácia e <i>loss</i> da <i>fold</i> 3 de validação cruzada do modelo SGD_raw.....	74
29	Gráficos de acurácia e <i>loss</i> da <i>fold</i> 4 de validação cruzada do modelo SGD_raw.....	74
30	Gráficos de acurácia e <i>loss</i> da <i>fold</i> 5 de validação cruzada do modelo SGD_raw.....	75

31	Gráficos de acurácia e <i>loss</i> da <i>fold</i> 1 de validação cruzada do modelo Adam_augmented.	75
32	Gráficos de acurácia e <i>loss</i> da <i>fold</i> 2 de validação cruzada do modelo Adam_augmented.	75
33	Gráficos de acurácia e <i>loss</i> da <i>fold</i> 3 de validação cruzada do modelo Adam_augmented.	76
34	Gráficos de acurácia e <i>loss</i> da <i>fold</i> 4 de validação cruzada do modelo Adam_augmented.	76
35	Gráficos de acurácia e <i>loss</i> da <i>fold</i> 5 de validação cruzada do modelo Adam_augmented.	76
36	Gráficos de acurácia e <i>loss</i> da <i>fold</i> 1 de validação cruzada do modelo SGD_augmented.	77
37	Gráficos de acurácia e <i>loss</i> da <i>fold</i> 2 de validação cruzada do modelo SGD_augmented.	77
38	Gráficos de acurácia e <i>loss</i> da <i>fold</i> 3 de validação cruzada do modelo SGD_augmented.	77
39	Gráficos de acurácia e <i>loss</i> da <i>fold</i> 4 de validação cruzada do modelo SGD_augmented.	78
40	Gráficos de acurácia e <i>loss</i> da <i>fold</i> 5 de validação cruzada do modelo SGD_augmented.	78

LISTA DE TABELAS

2.1	Resultados obtidos pelos trabalhos que fizeram classificação multi-classe "Normal", "Não-COVID-19", "COVID-19". O se refere aos autores que utilizaram de <i>undersampling</i> para balanceamento de dados em seus projetos.....	16
3.1	Distribuição de dados por classe do conjunto COVIDx-m, separados em "treino" e "teste".....	25
3.2	Composição dos modelos treinados.....	29
4.1	Tabela dos Hiperparâmetros variáveis para busca de melhor modelo do AutoKeras. .	31
4.2	Tempo de busca e pré treinamento de cada modelo durante o processamento no AutoKeras.	32
4.3	Tempo de treinamento de cada modelo testado durante o processo de validação cruzada.	35
4.4	Tabela de resultados de AUC de validação cruzada do treino Adam_ <i>raw</i>	38
4.5	Tabela de resultados de AUC de validação cruzada do treino SGD_ <i>raw</i>	38
4.6	Tabela de resultados de AUC de validação cruzada do treino Adam_ <i>augmented</i>	38
4.7	Tabela de resultados de AUC de validação cruzada do treino SGD_ <i>augmented</i>	38
4.8	Resumo dos valores de AUC por classe para cada modelo.	43
4.9	Tabela de comparação de AUCs dos modelos. Podemos ver que a melhor combinação de treino é Adam_ <i>raw</i> , seguido de SGD_ <i>raw</i> , SGD_ <i>augmented</i> e por último Adam_ <i>augmented</i>	43
4.10	Tabela de resultados das métricas de validação cruzada do treino Adam_ <i>raw</i>	44
4.11	Tabela de resultados das métricas de validação cruzada do treino SGD_ <i>raw</i>	45
4.12	Tabela de resultados das métricas de validação cruzada do treino Adam_ <i>augmented</i>	45
4.13	Tabela de resultados das métricas de validação cruzada do treino SGD_ <i>augmented</i>	50
4.14	Resumo dos valores de F1-Score médio entre as classes para cada modelo.	51
4.15	Tabela de comparação de F1-Scores dos modelos. Podemos ver que a melhor combinação de treino é SGD_ <i>raw</i> , seguido de Adam_ <i>raw</i> , SGD_ <i>augmented</i> e por último Adam_ <i>augmented</i>	51
4.16	Tabela Comparativa do modelo AutoResCovidNet (em negrito) com outros da literatura apresentados na Seção 2.3. O se refere aos autores que utilizaram de <i>undersampling</i> para balanceamento de dados em seus projetos.....	53
4.17	Tabela ordenada com as melhores combinações do modelo AutoResCovidNet. Em negrito a melhor combinação.	53

Capítulo 1

Introdução

Neste capítulo será apresentada a contextualização dos problemas a serem resolvidos juntamente com o escopo de trabalho planejado e os objetivos que se espera a cumprir.

1.1 Contextualização

No início do ano de 2020, o mundo foi surpreendido com a nova pandemia global de COVID-19. Surgida na China, no Mercado Atacadista de Frutos do Mar de Huanan na cidade de Wuhan, essa doença é causada por um coronavírus nomeado SARS-CoV-2. A fonte original do vírus ainda é desconhecida, mas todos os indícios indicam que ele fez o salto para a comunidade local através de algum animal intermediário entre o morcego e o ser humano. Os coronavírus são uma grande família de vírus conhecida por ser responsável por uma série de doenças respiratórias em humanos, como a SARS e a MERS [11]. Apesar de possuir uma taxa de mortalidade menor que a dessas doenças, a alta transmissibilidade do SARS-CoV-2 fez com que a doença se espalhasse rapidamente e causasse um grande impacto nas infraestruturas de saúde mundiais.

No momento da escrita deste texto, mais de 150 milhões de pessoas já foram infectadas em todo o mundo e mais de 3 milhões já morreram em decorrência de complicações da doença [12]. Mesmo com as vacinações avançando em ritmo global, a forma mais efetiva de combate à doença ainda é o distanciamento social e **acompanhamento e testes efetivos de casos**. Dessa forma, não só a capacidade de contenção do vírus pela vacinação em massa é amplificada como torna-se mais fácil o controle de regiões com maior número de infecções, podendo ser implementadas medidas restritivas localizadas e possivelmente prevenir o surgimento de novos focos epidêmicos uma vez que uma região já foi controlada [13, 14]. Outro aspecto importante da testagem é que até que a fonte original do vírus seja identificada e controlada, existe um risco potencial da reintrodução do vírus na população humana, e assim a possibilidade de novos surtos.

A testagem em massa da população, então, é fundamental para que os governos e instituições responsáveis possam ser guiados em ações públicas no combate da COVID-19.

Segundo Vandenberg *et al.* [15], o fim desta atual pandemia e a forma como lidaremos com

possíveis futuras ondas e surtos envolve diretamente a velocidade que desenvolvemos diagnósticos e testes e como podemos aplicá-los rapidamente em escala global. A pandemia de COVID-19 então evidenciou a necessidade e estimulou o estudo e desenvolvimento de métodos complementares de testagem em surtos de doenças.

Muitos desses novos testes utilizam Inteligência Artificial (IA) e, mais especificamente, sistemas de Aprendizado Profundo, para análise e extração de características e informações de bases de dados. Mesmo que não sejam suficientemente precisos para diagnóstico direto de COVID-19, esses testes podem auxiliar no desafogamento de sistemas de saúde como uma segunda via de diagnóstico em países com dificuldade de obtenção de testes clínicos mais precisos ou ainda servindo como triagem primária.

1.2 Descrição do problema e objetivos

O Aprendizado de Máquina vem sendo cada vez mais utilizado para resolução de problemas complexos em diversas áreas humanas. Especialmente as tarefas necessitadas de avaliação inteligente estão sendo aprimoradas com a implementação de sistemas baseados em Aprendizado Profundo, já provados capazes de atingir a capacidade humana [2]. O projeto desses sistemas, porém, possui algumas barreiras intrínsecas à sua natureza que atrapalham seu rápido desenvolvimento. Entre eles: 1) é um processo altamente iterativo, baseado em tentativa e erro; 2) longos períodos de tempo necessários para treinamento; e 3) acessibilidade e disponibilidade de bases de dados.

Então, baseado no contexto apresentado na Seção 1.1 acima, este trabalho tem como objetivo *projetar um modelo fim-a-fim de uma arquitetura de Rede Neural Profunda para diagnóstico de COVID-19 em imagens de Raio-X, utilizando o auxílio do AutoKeras para otimização de tempo de busca de arquiteturas*. Para isso então temos os seguintes **objetivos específicos**:

- Reunir um conjunto de dados para classificação multi-classe (Normal, Pneumonia, COVID-19);
- A partir de um processo de AutoML, obter um modelo otimizado para a classificação;
- Treinar o modelo obtido do zero variando o otimizador e conjunto de dados utilizando validação cruzada;
- Obter curvas ROC e Matrizes de confusão para cada um dos *folds* de validação cruzada para cada modelo;
- Obter métricas qualitativas e quantitativas das curvas ROC e Matrizes de Confusão para escolha de melhor modelo;
- Validar desempenho do modelo comparando com outros modelos da literatura.

Como será visto no Capítulo 2, há duas abordagens principais na literatura para diagnóstico de COVID-19 baseado em imagens e recursos auxiliares: utilizar imagens de Raio-X ou de Tomografia

Computadorizada como conjunto de dados para classificação. No contexto de sistemas de saúde com poucos recursos, é natural considerarmos que a disponibilidade de equipamentos e profissionais para obtenção de imagens de TC seja rara. Dado a proposta de um diagnóstico o mais acessível possível, foi escolhido então imagens de **Raio-X**, estas sendo mais baratas e mais fáceis de serem obtidas e compartilhadas.

As imagens TC podem ser utilizadas no futuro com aprimoramento natural dos sistemas quando os modelos utilizando Raio-X se mostrarem saturados.

1.3 Público Alvo

Este trabalho utiliza conceitos e termos típicos de Aprendizado Profundo e destina-se a um público que já possua familiaridade com este tema e suas áreas afins. Não serão explicados conceitos bases como *backpropagation*, *overfitting* e otimizadores, fundamentais para total compreensão deste trabalho. Recomenda-se ao leitor que chegue a este texto sem uma bagagem prévia no assunto um estudo da bibliografia base apresentada no Capítulo 2.

1.4 Apresentação do Manuscrito

Este manuscrito é organizado em 5 capítulos. São eles:

- Capítulo 1 - Introdução: o presente capítulo, contendo uma contextualização do cenário em que o problema visa ser resolvido;
- Capítulo 2 - Conceitos Fundamentais e Revisão Bibliográfica: neste capítulo é apresentado uma base teórica para os conceitos trabalhados neste documento e a revisão bibliográfica dos trabalhos relevantes que serão comparados posteriormente;
- Capítulo 3 - Metodologia: neste capítulo são apresentadas todas as técnicas e os procedimentos utilizados durante o projeto desenvolvido;
- Capítulo 4 - Resultados: neste capítulo é apresentado os resultados e discussões referentes à arquitetura e avaliação do modelo obtido juntamente informações relevantes dos processos de treinamento;
- Capítulo 5 - Conclusões: este capítulo final contém um breve resumo do trabalho realizado e apresenta perspectivas futuras para melhores resultados.

Capítulo 2

Conceitos Fundamentais e Revisão Bibliográfica

Este capítulo apresenta toda a fundamentação teórica necessária para compreensão da metodologia apresentada no Capítulo 3. Ao fim, também é apresentado um estudo sobre as referências mais relevantes no meio acadêmico que motivaram e guiaram este projeto. Esses trabalhos serão utilizados como base de comparação e avaliação dos resultados no Capítulo 4.

2.1 Fundamentação Teórica

Quando falamos de Inteligência Artificial, um dos tópicos mais expressivos da atualidade é o Aprendizado Profundo. Dominante em diversas áreas, este paradigma de Aprendizado de Máquina possui excelência em diversos campos, com uma menção especial a tarefas de Visão Computacional com as Redes Neurais Convolucionais (RNCs) [16, 17, 18, 19]. Esta Seção aborda os fundamentos por trás desse paradigma assim como algumas tecnologias que têm dominado as pesquisas e implementações tanto no mundo acadêmico quanto na indústria.

2.1.1 Aprendizado Profundo

Segundo Haykin (1999) [20], "um sistema de IA deve ser capaz de fazer três coisas: (1) armazenar conhecimento, (2) aplicar o conhecimento armazenado para resolver problemas, e (3) adquirir novos conhecimentos através da experiência". Temos então três componentes chave: representação, racionalização e aprendizado. A seguir, é apresentada uma breve contextualização das áreas de estudo de IA e será visto como dois destes conceitos são fundamentais para o seu desenvolvimento.

2.1.1.1 Contextualização

Uma maneira rudimentar de implementação de IA é a abordagem de Base de Conhecimento, ou *Knowledge Base*. Nesta abordagem, o conhecimento é codificado de forma literal e a máquina pode fazer inferências automáticas a partir de regras lógicas [1]. Nenhum dos trabalhos que seguiram este paradigma alcançaram sucessos relevantes. Segundo Goodfellow (2016) [1], "as dificuldades encontradas por estes sistemas [...] sugerem que sistemas de IA precisam da habilidade de adquirir seu próprio conhecimento, extraindo padrões de dados crus". Temos então o surgimento do Aprendizado de Máquina.

Um exemplo simples de Aprendizado de Máquina é o algoritmo de regressão logística, ou *logistic regression*, usado em 1990 por Mor-Yosef *et al.* [21] para decisão de recomendação de cesariana. Esses algoritmos possuem uma característica importante que é a forte dependência da representação dos dados que são apresentados para treinamento [1]. Em algumas tarefas isso pode ser um grande problema já que nem sempre é intuitivo quais são as características relevantes para serem extraídas de um conjunto de dados. Uma maneira, então, de se resolver esse problema é utilizar o próprio algoritmo de Aprendizado de Máquina para se aprender a melhor representação. Essa abordagem é chamada de Aprendizado Representacional, ou *Representation Learning*.

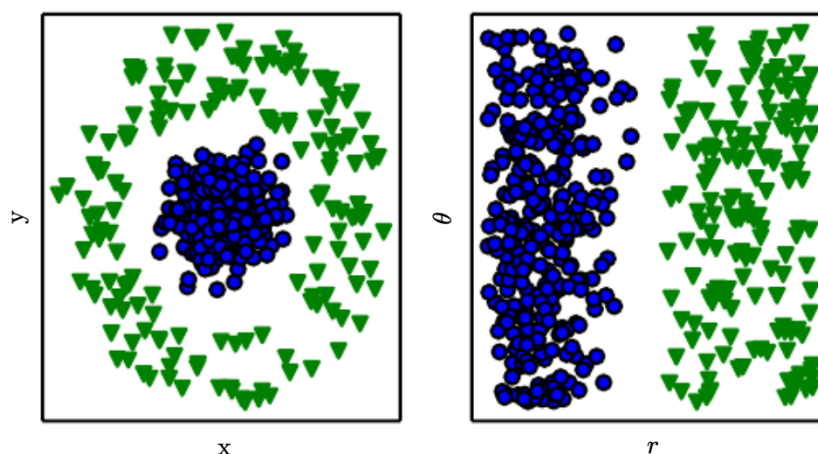


Figura 2.1: À esquerda, um conjunto de dados em representação cartesiana. À direita, o mesmo conjunto de dados em representação polar. Esta imagem ilustra como a representação dos dados influencia diretamente na definição das áreas de classificação em algoritmos de Aprendizado de Máquina. Figura original extraída do livro de Goodfellow (2016) [1].

Os algoritmos de Aprendizado Representacional costumam ter melhores performances que aqueles com representações projetadas manualmente, encontrando boas representações de dados rapidamente e permitindo maior flexibilidade para o sistema se adaptar a novos dados [1]. O principal exemplo de algoritmo de Aprendizado Representacional é o *autoencoder*. Ainda resolvendo a representação dos dados de entradas, existem outros problemas que complicam os algoritmos de aprendizado. O principal deles se refere aos **Fatores de Variação**. Os Fatores de Variação são quantidades que influenciam diretamente os dados observados mas que são difíceis de serem separados pois são inerentes à origem desses dados. Por exemplo, a forma da silhueta de um

carro depende do ângulo do observador[1]. Mudanças na representação dos dados da imagem de um carro podem não ser suficientes para um sistema avaliar se aquilo é de fato a silhueta de um carro. O Aprendizado Profundo resolve este problema do aprendizado representacional fazendo representações que são combinações de outras representações mais simples [1].

No Aprendizado Profundo, conhecimentos complexos são baseados em conhecimentos mais simples, dessa forma, é possível se construir uma pilha de características relevantes, onde cada estágio da pilha armazena o conhecimento de uma abstração. O maior exemplo deste paradigma são as redes neurais baseadas no perceptron de multicamadas, ou *multilayer perceptron* (MLP).

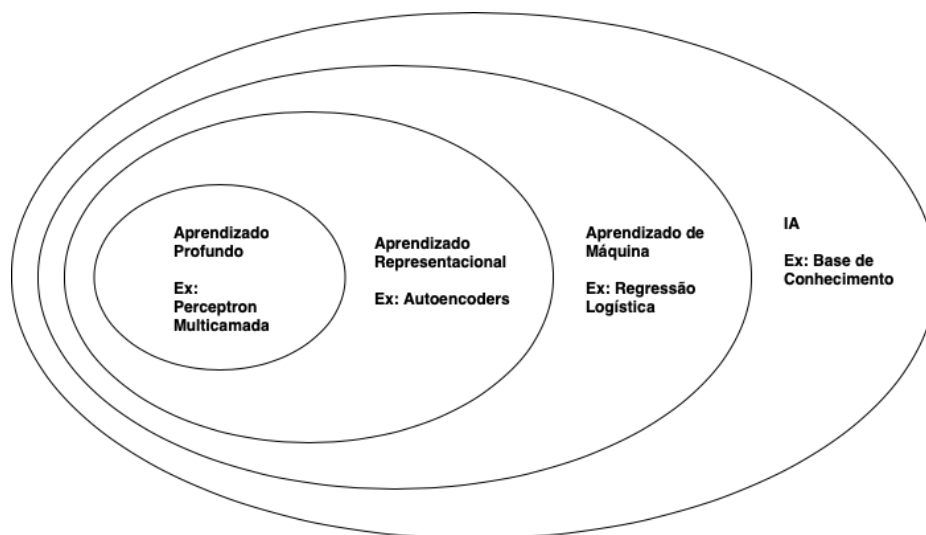


Figura 2.2: Diagrama de Venn que mostra como Aprendizado Profundo é um tipo de Aprendizado Representacional, que por sua vez é um tipo de Aprendizado de Máquina e este é apenas uma sub-área da Inteligência Artificial. Imagem adaptada do trabalho original de Goodfellow (2016) [1].

2.1.1.2 Definição

A partir do breve histórico apresentado acima, definimos então as Redes Neurais Profundas (RNPs) como um modelo flexível e treinável capaz de representar abstrações complexas de um conjunto de dados como abstrações mais simples, sem a extração ou mudança de representação manual de características. Ao longo deste trabalho, essas redes serão referidas apenas como Redes Neurais (RN), ficando subentendido o termo "Profundas".

2.1.2 Redes Neurais Convolucionais

Segundo Goodfellow (2016) [1], Redes Neurais Convolucionais (RNC) são Redes Neurais que implementam a operação de convolução, ao invés da multiplicação matricial tradicional em pelo menos uma de suas camadas. A operação de convolução discreta pode ser vista a seguir na Equação 2.1

$$s(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a)da = (x * w)(t). \quad (2.1)$$

A partir da equação acima, pode-se entender a convolução como uma média móvel ponderada pelas suas proximidades, onde $x(a)$ é um número real e w é uma função de probabilidade de densidade.

Estendendo este conceito para duas dimensões, obtemos Equação 2.2, em que $I(m, n)$ é a matriz (imagem) de entrada em duas dimensões, m e n são respectivamente os eixos das abscissas e ordenadas da imagem, $K(i, j)$ é a matriz do *kernel* da convolução, isto é, a janela de pesos que aplica a operação na imagem de modo invertido e deslizante, e i e j são respectivamente os eixos das abscissas e ordenadas da região de aplicação da convolução. Tanto no caso unidimensional quanto bidimensional, S é o vetor (ou matriz) resultado da operação de convolução de dimensão igual à dimensão de aplicação da convolução i, j .

$$S(i, j) = \sum_m \sum_n I(m, n)K(i-m, j-n) = (I * K)(i, j). \quad (2.2)$$

A saída da convolução de duas dimensões é um novo vetor de dimensões i, j , com características próximas enaltecidas ao longo da região de aplicação da operação. Especialmente para aplicação de Visão Computacional, isso se torna extremamente útil, já que não só permite o melhor agrupamento de características próximas em uma imagem, possibilitando o empilhamento das abstrações de regiões de interesse nas camadas das RNCs, como o processo de cálculo multi-dimensional se torna mais rápido que operações singulares, pixel por pixel [1].

2.1.3 Redes Residuais

Apesar das RNCs serem utilizadas com sucesso em tarefas de Visão Computacional desde o início dos anos 2000 [22, 23, 24, 25], elas se mantiveram relativamente esquecidas até 2012, com a publicação da Alexnet [16]. Com um uso eficiente de GPU, funções de ativação do tipo Unidade Linear Retificada (ReLU), aumento de dados e uma nova técnica de regularização, o *Dropout*, esta RNC atingiu resultados superiores na popular competição de classificação de imagens ImageNet [26, 27].

Após a publicação da Alexnet [16] e a decorrente popularização das Redes Neurais Convolucionais, a ideia de que RNCs aprendiam melhor e alcançavam melhores resultados quanto mais profundas fossem foi bastante difundida. Não de todo errada, a interpretação de que essas redes conseguiam abstrair e extrair características e conceitos mais complexos com mais camadas fazia bastante sentido e encorajava essa ideia. No entanto, logo alguns problemas de se empilhar camadas indiscriminadamente começaram a ser percebidos. Entre eles, o 'desaparecimento' e 'explosão' dos gradientes [28, 29], centrais no treinamento da rede. Mesmo que esses problemas pudessem ser contornados [30, 31], logo se percebeu que com modelos ainda mais profundos, em algum momento uma saturação da métrica de acurácia era percebida, degradando o gradiente [32, 33]. Esses problemas então foram resolvidos com a introdução das ResNets [2].

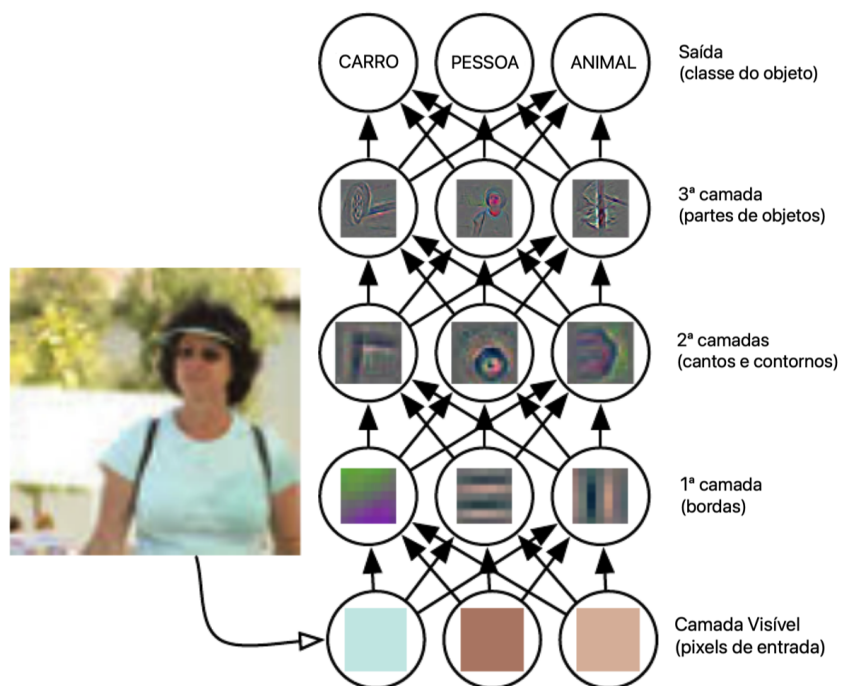


Figura 2.3: Representação de como uma classificação é feita baseada em abstrações complexas que são baseadas em características mais simples de uma imagem. Imagem adaptada do trabalho original de Goodfellow (2016) [1].

Essas redes resolvem o problema de degradação do gradiente introduzindo o bloco residual, como mostrado na Figura 2.4.

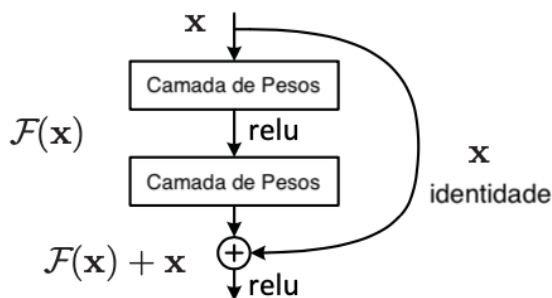


Figura 2.4: Bloco de aprendizado residual. Figura adaptada da publicação original de He *et al.* (2015) [2].

A ideia por trás deste bloco é que uma solução para a construção de modelos mais profundos seria adicionar camadas *identidade*, fazendo um atalho entre as conexões, e as outras camadas são copiadas dos modelos mais rasos [2]. Essa solução se mostrou bastante eficiente e marcou um novo surgimento de RNCs com centenas de camadas, extremamente profundas e sem enfrentar os problemas de gradiente anteriores. As ResNets, assim chamadas essas redes, se tornaram referência em eficiência e são utilizadas em várias aplicações diferentes de Visão Computacional como avaliação e comparação de *benchmarks* [34, 35, 36].

2.1.4 AutoML

Quando falamos de Aprendizado de Máquina, normalmente falamos de um *pipeline* complexo e iterativo que envolve várias etapas: obtenção e tratamento de dados, projeto, avaliação e ajuste do modelo, etc. Muitas vezes esses processos são recursivos, sendo necessários vários especialistas para revisar e aprimorar cada etapa. Mesmo assim, nada garante que o problema proposto seja de fato resolvido. Essas condições particulares impõem grandes restrições de custo e tempo aos pesquisadores e desenvolvedores. Por esse motivo, grandes incentivos têm sido dados à pesquisas de tecnologias capazes de propor de forma automática modelos eficientes de Aprendizado de Máquina [3, 37]. O AutoML é o resultado disto.

Obtido a partir da junção das palavras em inglês *automatic* (automático) e *Machine Learning* (aprendizado de máquina), o termo AutoML significaria literalmente "Aprendizado de Máquina Automático". AutoML então, dá nome ao conjunto de técnicas e processos computacionais responsáveis por automatizar todo o *pipeline* de projeto e treinamento de um modelo capaz de gerar uma saída a partir de um conjunto de dados de entrada.

Segundo He *et al.* (2021) [3], as etapas do *pipeline* de AutoML são: preparação de dados, engenharia de características, geração de modelo e avaliação do modelo. Essas etapas estão representadas na Figura 2.5.

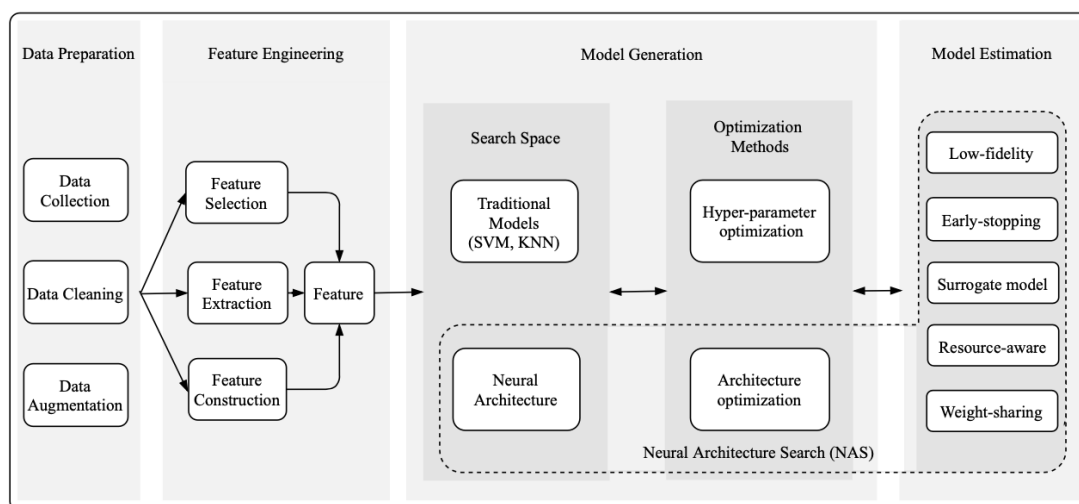


Figura 2.5: Diagrama de representação de um *pipeline* típico de AutoML. Imagem extraída do artigo original de He *et al.* (2021) [3].

Difícil de se automatizar a etapa de preparação de dados, devido a necessidade de um especialista ter de fazer a curadoria de dados corretamente, a geração do modelo talvez seja a etapa mais importante do processo. A geração do modelo pode ser dividida em *espaço de busca* e *métodos de otimização*. Espaço de busca se refere aos princípios de projeto dos modelos de Aprendizado de Máquina, podendo ser tradicional ou neural. Métodos de otimização podem ser dos tipos *otimização de parâmetros* (taxa de aprendizado e tamanho de batch, por exemplo) e/ou *otimização de arquitetura* (quantidades e tipos de camadas, por exemplo). Definimos então o processo de Busca de Arquitetura Neural, ou BAN, como as etapas de: 1) busca no espaço de arquiteturas de

redes neurais; 2) métodos de otimização (parâmetros e arquitetura); e 3) métodos de estimação de modelos. A etapa de geração do modelo é representada com mais detalhes na Figura 2.6.

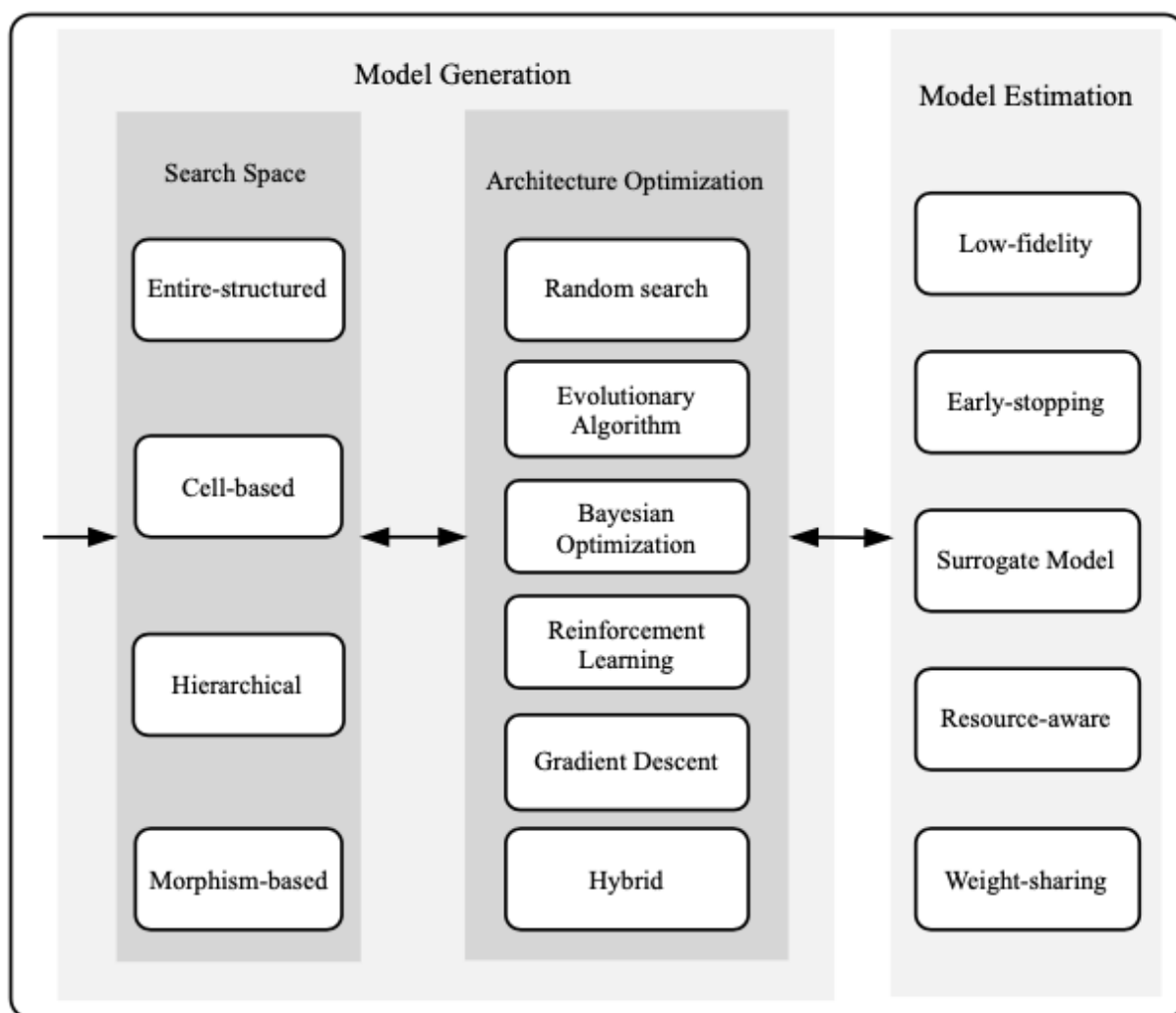


Figura 2.6: Diagrama de representação de um processo de Busca de Arquitetura Neural, dividido em dois estágios principais de fluxo bilateral: geração do modelo e estimação do modelo. Estas etapas são iterativas e podem se repetir diversas vezes. Imagem extraída do artigo original de He *et al.* (2021) [3].

Ainda segundo He *et al.* (2021) [3] sobre BANs, podemos classificar os espaços de busca mais comuns em *estrutura-completa*, *baseada em células*, *hierárquica*, *baseada em morfismos*. Os métodos mais comuns de otimização de arquitetura são *aprendizado por reforço*, *algoritmos baseados em evolução*, *gradiente descendente*, *baseada em modelo substituto* e outros modelos híbridos.

2.1.4.1 AutoKeras

Em 2019, Jin, Song e Hu [38] propuseram uma nova ferramenta de AutoML, o AutoKeras. Utilizando BAN com espaço de busca baseado em morfismo de rede e Otimização Bayesiana e Processo Gaussiano para seleção de operações, o AutoKeras é uma ferramenta gratuita de código

livre poderosa que compete diretamente com os serviços de AutoML das grandes plataformas de computação em nuvem.

O AutoKeras possui sua estrutura dividida em módulos especiais, cada um responsável por uma parte do processo. Na parte mais alto nível, temos a API chamada pelo usuário, responsável por chamar os outros métodos. Temos o Buscador, o módulo de BAN que contém o Otimizador Bayesiano e o Processo Gaussiano, que rodam na CPU. O módulo de Grafos, que é controlado pelo Buscador e processa as operações de morfismo das redes na RAM para rápido acesso. O módulo Treinador de Modelos, que treina as Redes Neurais nos conjuntos de dados utilizando de paralelismo na GPU. E por último, o módulo para Armazenamento de Modelos, que nada mais é que um tanque de modelos em dispositivos de armazenamento, visto que os modelos são grandes e não poderiam todos serem guardados em memória. O AutoKeras possui então, uma arquitetura modularizada projetada para fazer uso eficiente dos recursos de CPU e GPU.

Na Figura 2.7 é representado um fluxo de trabalho típico no AutoKeras: 1) é dado o início da busca para melhor arquitetura neural para o conjunto de dados, a API recebe a chamada, pré-processa os dados, e chama o buscador para início de busca; 2) o Otimizador Bayesiano gera uma nova arquitetura neural e chama o módulo de Grafo para construir a rede neural gerada na RAM; 3) o modelo é copiado para a GPU, onde será feito o treinamento pelo Treinador. Quando o treino é finalizado, o modelo é salvo em disco e usado para realimentar o Buscador a fim de atualizar o processo Gaussiano.

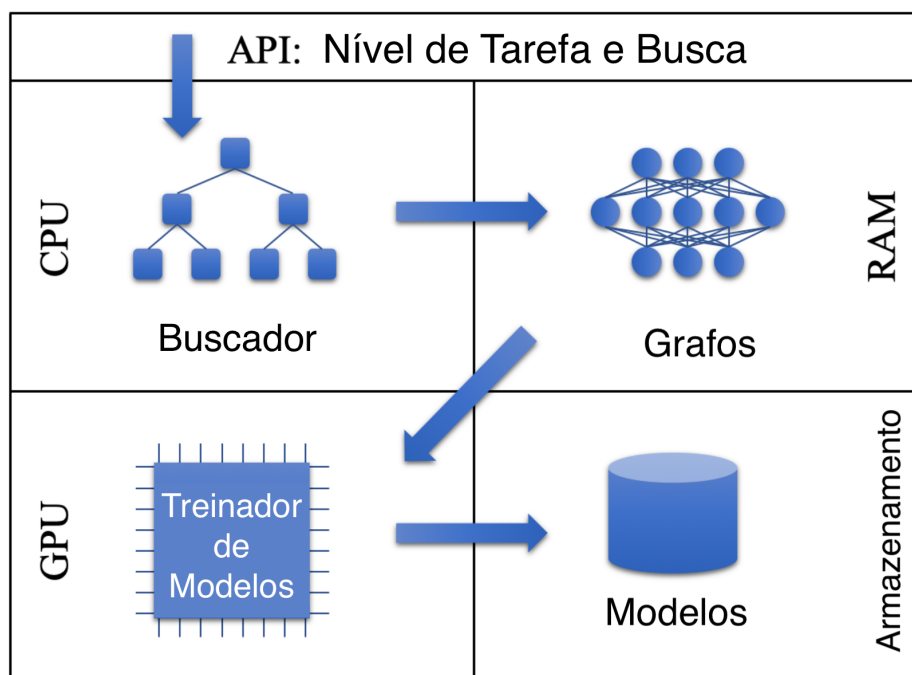


Figura 2.7: Exemplo dos processos do AutoKeras em cada um dos módulos. Seguindo o fluxo das setas: API, Buscador, módulo de Grafos, Treinador e módulo de Armazenamento de Modelos. Imagem adaptada do artigo original de Jin *et al.* (2019) [4].

Uma das grandes vantagens do uso do AutoKeras é não estar preso às restrições e condições

dos serviços de AutoML das plataformas de computação em nuvem. A maioria destes serviços não são gratuitos, exigem configurações de ambiente que podem ser complexas para pesquisadores com pouca bagagem de computação, e por último mas não menos importante, não garantem segurança e privacidade dos dados. Sendo assim, o AutoKeras ganha facilmente esse espaço, sendo de fácil uso e acesso, gratuito e poderoso.

2.2 Métricas de Análise

Em tarefas de classificação temos algumas métricas estabelecidas na literatura como padrão para avaliar a qualidade do modelo. Nesta seção vamos descrever as métricas que serão utilizadas para análise dos resultados no Capítulo 4.

2.2.1 Medidas

Sendo, VP, ou *verdadeiros positivos*, referente ao número de dados verdadeiramente positivos classificados como positivos, FP, ou *falsos positivos*, referente ao número de dados verdadeiramente negativos classificados como positivos, VN, ou *verdadeiros negativos*, referente ao número de dados verdadeiramente negativos classificados como negativos e FN, ou *falsos negativos*, referente ao número de dados verdadeiramente positivos classificados como negativos [39], temos:

- Acurácia

$$Acc = \frac{VP + VN}{VP + VN + FP + FN} \quad (2.3)$$

Esta medida é simplesmente a quantidade de classificações corretas em relação à todas classificações. Podemos observar que a Acurácia é uma métrica facilmente enganável por conjuntos de dados enviesados. Por exemplo, um conjunto em que 99% dos dados são de uma classe hipotética A, se um modelo classificar sempre todos os dados como A, ele ainda teria 99% de acurácia, enganando a performance de um modelo ineficiente. Por este motivo não é aconselhável utilizar esta métrica como referência de qualidade.

- Precisão

$$Prec = \frac{VP}{VP + FP} \quad (2.4)$$

Esta medida é uma representação direta da quantidade de *falsos positivos* em uma classificação. Se FP = 0, a Precisão é máxima e igual a 1. Se FP for muito alto, a Precisão diminui.

- Sensibilidade

$$Sens = \frac{VP}{VP + FN} \quad (2.5)$$

Esta medida é uma representação direta da quantidade de *falsos negativos* em uma classificação. Se FN = 0, a Sensibilidade é máxima e igual a 1. Se FN for muito alto, a Sensibilidade diminui.

- *F1-Score*

$$F1 = 2 * \frac{Prec * Sens}{Prec + Sens} \quad (2.6)$$

Esta medida também pode ser interpretada como a *média harmônica* da Precisão e da Sensibilidade. É uma ótima forma de se considerar a qualidade do classificador já que considera ao mesmo tempo ambas quantidades de classificações errôneas (*falsos negativos e positivos*).

2.2.2 Matriz de Confusão

A matriz de Confusão é a matriz obtida traçando as classes verdadeiras em um eixo e no outro as classes classificadas pelo modelo [5]. Cruzando esses valores, temos as quantidades das medidas de VP, FP, FN e VN. Um exemplo de uma Matriz de Confusão pode ser visto na Figura 2.8 a seguir.

Classes Verdadeiras	P	VP	FN
	N	FP	VN
		P	N
		Classes Inferidas	

Figura 2.8: Exemplo de Matriz de Confusão. No eixo Y nós vemos as classes verdadeiras e no eixo X as classes inferidas pelo modelo. Imagem adaptada do artigo original de Fawcett [5].

Essa matriz pode ser expandida para classificação multi-classe aumentando o número de espaços para *falsos negativos*. Isso será utilizado posteriormente para avaliação no Capítulo 4.

2.2.3 Curva ROC

A curva ROC é um gráfico que tem como objetivo a avaliação da qualidade de um modelo de classificação. Ela é obtida traçando os valores de VP no eixo Y e os valores de FP no eixo X de um plano variando o valor do limiar de classificação [40]. Um exemplo de uma curva ROC pode ser visto na Figura 2.9 a seguir.

Uma característica interessante desse gráfico é que o classificador **aleatório** pode ser facilmente representado traçando a diagonal do início do gráfico na posição (0,0) até a posição (1,1). Dessa forma nós mantemos sempre o mesmo número de FP e VP. E assim, independente do limiar de classificação, o modelo sempre possui taxas iguais de classificação para a classe positiva e

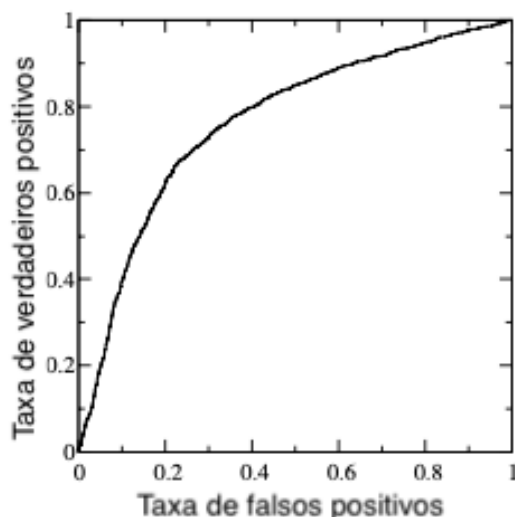


Figura 2.9: Exemplo de curva ROC. No eixo Y nós vemos a taxa de VP e no eixo X a taxa de FP. Imagem adaptada do artigo original de Fawcett [5].

negativa. É possível, então, analisar se um modelo é minimamente melhor que o classificador aleatório observando se a curva se encontra acima da diagonal do gráfico.

A curva ROC também nos dá uma métrica importante de avaliação de modelo que é a *Area Under Curve*, ou AUC. Esta métrica é dada simplesmente pela integral da curva ROC ou literalmente a área abaixo dela [5]. Esta métrica varia de 0 a 1, sendo respectivamente referentes aos classificadores ideal-negado e ideal. Sendo o classificador aleatório a diagonal que corta o gráfico em dois, este possui AUC de 0.5, e portanto nenhum classificador real deve possuir AUC menor que este valor, ou seja, performance pior que o aleatório.

Assim, conseguimos reduzir a informação da curva ROC é uma única métrica numérica.

2.3 Trabalhos Relacionados

Como visto na Seção 2.1, as RNCs atingem resultados excelentes em tarefas de classificação de dados. Assim, considerando o contexto apresentado no Capítulo 1, desde o início da pandemia em 2020, diversos trabalhos foram publicados visando utilizar esse paradigma para propor soluções que auxiliem no combate deste problema. Nesta Seção serão apresentados os principais trabalhos publicados até o presente momento referentes à diagnóstico de COVID-19 envolvendo RNCs e Aprendizado Profundo.

Para se propor um modelo para diagnóstico de COVID-19, deve-se primeiro decidir em qual conjunto de dados a classificação deve ser feita. Ignorando informações adicionais provenientes de exames fisiológicos (como RT-PCR, por exemplo), a tarefa se resume a um problema de Visão Computacional. Assim, as principais propostas na literatura têm como entrada do sistema imagens de **Raio-X** e/ou de **Tomografia Computadorizada** da região do tórax do paciente. A partir

dessas imagens, então, a classificação deve ser feita baseada nas características encontradas.

Na Tabela 2.1 a seguir são apresentados alguns resultados e estudos relevantes que fizeram classificação multi-classe com "Normal", "Não-COVID-19", "COVID-19" que serão utilizados como base para comparação no Capítulo 4. Estes trabalhos serão descritos nas Seções 2.3.1 e 2.3.2. Nesta Seção, será referido à "Não-COVID-19" como a classe referente à Pneumonias genéricas que não são COVID-19. Os trabalhos marcados com na Tabela 2.1 fizeram forçaram balanceamento de dados a fim de melhorar performance do modelo. Isso será levado em conta na comparação feita no Capítulo 4.

Na Figura 2.10 vemos o grafo gerado no ConnectedPapers¹ onde é representada a relação de referências dos artigos relevantes com o artigo de Wang *et al.* [41].

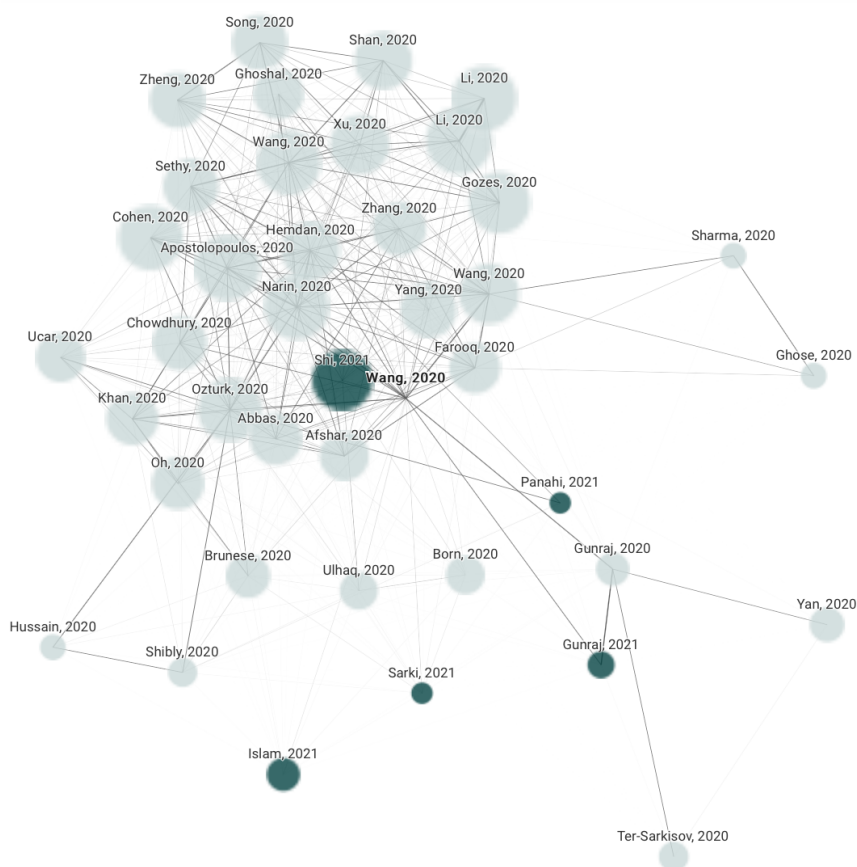


Figura 2.10: Grafo de artigos relacionados que citam o trabalho de Wang *et al.*.

2.3.1 Raio-X

O trabalho de Wang *et al.* (2020) [41] se tornou a principal referência para estudo e métrica desse problema, inspirando uma série de outros trabalhos. Eles propõem não só uma inédita arquitetura de rede neural para diagnóstico de COVID-19, a COVID-Net, como também apresentam um conjunto de dados de código aberto para treino e avaliação de outros modelos, o COVIDx.

¹<https://www.connectedpapers.com/>

Tabela 2.1: Resultados obtidos pelos trabalhos que fizeram classificação multi-classe "Normal", "Não-COVID-19", "COVID-19". O se refere aos autores que utilizaram de *undersampling* para balanceamento de dados em seus projetos.

Estudo	Tipo de Imagem	Classes	Acurácia (%)	Precisão (%)	Sensibilidade (%)	F1-Score (%)
Wang <i>et al.</i> [41]	Raio-X	Global	93.3	-	-	-
		<i>Normal</i>	-	90.5	95	-
		<i>Pneumonia</i>	-	91.3	94	-
		<i>COVID-19</i>	-	98.9	91	-
Tulin <i>et al.</i> * [42]	Raio-X	Global	87.02	89.96	85.35	87.37
Khan <i>et al.</i> * [43]	Raio-X	Global	95	95	96.9	95.6
Chowdhury <i>et al.</i> * [44]	Raio-X	Global	97.94	97.95	97.94	97.94
Gunraj <i>et al.</i> [45]	TC	Global	99.1	-	-	-
		<i>Normal</i>	-	99.4	100	-
		<i>Pneumonia</i>	-	98.4	99	-
		<i>COVID-19</i>	-	99.7	97.3	-
Misztal <i>et al.</i> [46]	Raio-X	Global	58	74	100	85
	TC	Global	73	82	83	82
Li <i>et al.</i> * [47]	TC	Global	-	-	-	-
		<i>Normal</i>	-	-	94	-
		<i>Pneumonia</i>	-	-	87	-
		<i>COVID-19</i>	-	-	90	-

Este agrupa 5 outros conjuntos de dados para se obter um total de 13.975 imagens de Raio-X de 13.870 pacientes. Os conjuntos de dados utilizados foram: COVID-19 Image Data Collection [6], Figure 1 COVID-19 Chest X-ray Dataset Initiative [7], ActualMed COVID-19 Chest X-ray [8] Dataset Initiative [8], RSNA Pneumonia Detection Challenge conjunto de dados [10] e COVID-19 radiography database [9].

Implementado para classificação multi-classe baseado em Raio-X, "Normal", "Não-COVID-19" e "COVID-19", a COVID-Net atingiu respectivamente para cada uma das classes: acurácia global de 93.3%; sensibilidade de 95%, 94% e 91%; e precisão 90.5%, 91.3% e 98.9%. Para projeto da arquitetura, fizeram uso de uma abordagem humano-máquina, descobrindo estruturas de rede eficientes com auxílio da máquina e utilizando do conhecimento dos projetistas para se montar a arquitetura final. A rede então possui dois ramos principais, com mais de 60 camadas e totalizando 11.75 milhões de parâmetros.

Também com a ideia de uma projeto completo de RNC, Tulin *et al.* (2020) [42] utilizam o conjunto de dados de Cohen (2020) [6] para construir uma rede própria, a DarkCovidNet, baseada na Darknet-19. Como principais características, esta se destaca como a rede base da YOLOv2 [48], possuindo apenas 19 camadas e fazendo uso de Leaky ReLU [49] com o objetivo de se prevenir do problema de neurônios mortos (ativação nula). Ela também se utiliza bastante de normalização de *batches* e *maxpools* [16], com a intenção de reduzir o tempo de treinamento e aumento da robustez do modelo. Tulin *et al.* também utilizou da técnica de *undersampling*, isto é, retirar dados aleatoriamente de modo a minimizar o efeito de desbalanceamento dos dados aproximando a quantidade de amostras de cada classe. Essa proposta atingiu 98.08% de acurácia em classificação binária (Normal/COVID-19) e 87.02% em multi-classe (Normal/Pneumonia/COVID-19) com um modelo com 1,164 milhões de parâmetros (quase 10 vezes menor que o de Wang *et al.* (2020) [41]

).

Farooq e Hafeez (2020) [34] utilizaram como base a ResNet50[2] para montar uma arquitetura própria mais leve e rápida de se treinar que a COVID-Net[41], utilizando o mesmo conjunto de dados, o COVIDx [41]. Este modelo, Covid-Resnet, foi treinado para classificação de 4 classes utilizando transferência de aprendizado de pesos da ImageNet [26] para o corpo da ResNet50 e técnicas de intervenção para calibração fina de parâmetros, como redimensionamento de imagens e seleção de taxas de aprendizado com *Discriminative Learning Rate* [50] e *Cyclical Learning Rate* [51]. O modelo foi capaz de atingir 96% de acurácia global e 96.58%, 97.15%, 93.96%, 100% em acurácias para as classes Normal, Pneumonia Bacteriana, Pneumonia Viral e COVID-19.

Assim como Tulin *et al.* [42], Khan *et al.* (2020) [43] também utilizaram uma RNC conhecida, Xception [52], como base para seu projeto, a CoroNet. Utilizaram também transferência de aprendizado e *undersampling* a fim de contornar o problema da falta de conjunto de dados. Com 33 milhões de parâmetros, a CoroNet atingiu 89.6% de acurácia global para classificação de 3 classes (Normal, Pneumonia e COVID-19) e 99% para binária (Normal/COVID19).

Já no trabalho de Hemdan *et al.* (2020) [53], foi utilizado um conjunto de dados pequeno de apenas 50 imagens, com 25 destas de COVID-19 confirmadas. Foram treinados 7 modelos conhecidos de RNC. As redes utilizadas foram VGG19 [54], DenseNet121 [55], InceptionV3 [56], ResNetV2 [57], Inception-ResNet-V2 [58], Xception [52] e MobileNetV2 [59]. Os melhores resultados foram da VGG19 e DenseNet com *F1-Score* respectivamente de 0.89 e 0.91 na classificação binária em Normal/COVID-19.

Chowdhury *et al.* (2020) [44] utilizaram de transferência de aprendizado e *undersampling* para comparar 8 tipos diferentes de arquitetura de rede para classificação binária e multi-classe. As redes testadas foram MobileNetv2 [59], SqueezeNet [60], ResNet18 [2], ResNet101 [2], DenseNet201 [55], CheXNet [61], Inceptionv3 [56] e VGG19 [54] treinadas com aumento de dados (AD), consistindo em: rotação de 5, 10 e 15 graus e translação de 5%. Foram feitas duas classificações: binária e multi-classe. Para classificação binária, os melhores resultados foram obtidos com a DenseNet201 e CheXNet, ambas com AD e respectivamente com acurácia 99.70% e 99.69%. Para classificação multi-classe, essas mesmas redes obtiveram os melhores resultados mas agora com acurácia de 97.74% para CheXNet sem AD e 97.94% para DenseNet201 com AD.

Um outro trabalho bastante notório é o de Narin *et al.* (2020) [62]. Neste caso, a fim de se superar os desafios dos estágios iniciais da pandemia quando não havia dados em abundância, foi utilizado transferência de aprendizado em 5 redes diferentes para 3 tipos de classificação binária baseada em Raio-X: "COVID-19/Normal", "COVID-19/Pneumonia Bacteriana" e "COVID-19/Pneumonia Viral". Os modelos base utilizados para a transferência de aprendizado foram treinados no banco de dados ImageNet [26]. As redes testadas por eles foram InceptionV3 [56], ResNet50 [2], ResNet101 [2], ResNet152 [2] e InceptionResNetV2 [58]. O conjunto de dados utilizado para retreinamento foi construído a partir da junção de três outros ([6], [63] e [64]), contendo imagens diagnosticadas com doenças respiratórias como ARDS, MERS, pneumonia, SARS e não-detectadas (saudáveis ou "Normal"). Desta forma foram construídos os três conjunto de dados binários, com o objetivo de se avaliar não só o diagnóstico de COVID-19 em um paciente sau-

dável mas também diferenciá-la de outras formas de anomalias respiratórias. O melhor modelo encontrado foi da ResNet50, atingindo respectivamente 96.1%,99.5% e 99.7% de acurácia nas classificação binárias de COVID-19/"Normal", COVID-19/Pneumonia Viral e COVID-19/Pneumonia Bacteriana.

Misztal *et al.* (2020) [46] juntaram os conjuntos de dados do Cohen *et al.* [6] de Raio-X e o reunido por Yang *et al.* [65] de TC para fazer diversos testes com diferentes arquiteturas. Aqui serão apresentados apenas os resultados da classificação multi-classe ("Normal", "Não-COVID-19", "COVID-19") de Raio-X. Os resultados de TC serão apresentados na Subseção seguinte. Em Raio-X, a arquitetura com melhor resultado foi a DenseNet-169 [55] com métricas globais de: 0.58 de acurácia; 0.74 de precisão; 1 de sensibilidade; 0.85 de *F1-Score*.

Sethy e Behra (2020) [66], utilizaram de uma abordagem diferente dos trabalhos apresentados até agora. Ao invés de utilizar de modelos de aprendizagem profunda para a classificação fim-a-fim, estes modelos, pré-treinados para classificação de doenças de arroz, foram utilizados apenas para extração de características profundas das imagens. Estas características então foram classificadas por SVMs [67] em COVID-19/Normal. Dentre as diversas redes testadas, a que obteve melhor resultado foi a ResNet50 + SVM, com 95% de acurácia, 97% de sensibilidade e 93% de especificidade.

Seguindo essa linha de abordagens diferentes, temos ainda dois trabalhos notórios: em Zhang *et al.* (2020) [68], ao invés de se utilizar de uma rede profunda para classificação fim-a-fim, foi desenvolvido um sistema de detecção de anomalia (classificação unitária) com critério de confiança *Confidence-Aware Anomaly Detection*, focada não só em COVID-19 mas também em diagnóstico geral de pneumonia viral. Para isso, utilizaram uma rede profunda, a EfficientNet [69], como extrator de características e dois outros módulos *head*: 1) módulo de detecção de anomalia com saída de pontuação de anomalia; 2) módulo de predição de confiança da rede. O módulo de predição de confiança é importante uma vez que a sensibilidade para diagnósticos médicos é um fator crítico e portanto, se ter um mecanismo que aumenta a confiabilidade do modelo é essencial. Apesar dos diversos mecanismos de aprimoramento de confiança e do sistema, com um resultado de 83.61% AUC (*Area Under Curve*) e acurácia máxima de 79%, este sistema não é capaz de bater os resultados publicados por outros autores com acurácia de no mínimo 90% em classificações binárias.

Como pode ser visto nos diversos trabalhos acima, a falta de conjuntos de dados grandes o suficiente é um problema a ser solucionado para uso extensivo de técnicas de Aprendizado Profundo para COVID-19. Então, sabendo que RNCs não são invariáveis à características espaciais em imagens sem o uso de Aumento de Dados, Afshar (2020) [70] propõe a COVID-CAPS, um sistema baseado em CapsNets [71], redes estas capazes de captar informações espaciais. Com esse sistema, fizeram classificação binária entre as classes: 1) Não-Covid: Normal + pneum. bacterial + pneum. viral; 2) Covid-19. Foram feitos testes também a fim de se verificar o impacto de pré treinamento para a rede. Neste caso, para a transferência de aprendizado foi feito um pré-treinamento com imagens de tórax, ao invés de pesos gerais da ImageNet, como feito por outros. Foi obtido então 98.3% de acurácia com pré treinamento e 95.7% de acurácia sem pré-treinamento.

2.3.2 Tomografia Computadorizada (TC)

Como descrito na Subseção 2.3.1, um dos principais desafios encontrados no uso de RNCs no início da pandemia foi a falta de dados. Como será descrito em cada caso, todos os trabalhos a seguir envolveram coleta e reunião de conjunto de dados do tipo TC.

Yang *et al.* (2020) construíram um conjunto de dados de TC com 349 imagens de pacientes confirmados com COVID-19 e 463 imagens de pessoas saudáveis [65]. Este conjunto de dados possui a particularidade de não utilizar de toda a estrutura 3D das imagens de TC, ele utiliza apenas de uma fatia por imagem. A fim de confirmar a validade dos dados reunidos, também foram feitos testes de classificação binária com dois modelos de redes diferentes: DenseNet-169 [55] e ResNet-50 [2], fazendo a classificação na imagem de segmentada por uma U-Net [72]. A validade de uso do conjunto de dados foi confirmada atingindo com a DenseNet-169: Acurácia de 89%, AUC de 0.98 e *F1-Score* de 0.9. Esta rede foi treinada com transferência de aprendizado e *contrastive self-supervised learning*, uma técnica de ajuste fino de aprendizado [73].

Em fevereiro de 2020, Ying *et al.* reuniram um conjunto de dados de 88 pacientes diagnosticados com COVID-19, 101 pacientes infectados com pneumonia bacteriana e 86 pessoas saudáveis. Com este conjunto de dados, um modelo baseado em Aprendizado Profundo foi proposto, o DeepPneumonia [74]. Neste conjunto, cada imagem de TC possui 15 fatias. O modelo proposto, baseado em *Details Relation Extraction* [75], foi construído e pré-treinado na ResNet50 [2], com adição de uma *Feature Pyramid Network* (FPN) [76]. Este modelo extrai os melhores *top-K* detalhes das imagens de TC e obtêm as predições em nível de imagem. Então, para cada pessoa, as predições de cada fatia são agregadas com *mean pooling* e é feito o diagnóstico à nível de paciente. Este modelo obteve: AUC de 0.95; *Recall* de 0.96; *Precisão* de 0.79; *F1-Score* de 0.87; e Acurácia de 0.86.

Em março de 2020, Gozes *et al.* coletam diversos conjuntos de dados internacionais para aplicar modelos de aprendizado profundo 3D e 2D na detecção de COVID-19 e avaliação do avanço da doença, *Corona score*. O sistema proposto possui duas partes principais: 1) análise em 3D das imagens de TC, analisando nódulos e opacidades; 2) análise em 2D de cada fatia para detecção e localização de anomalias maiores. As imagens do conjunto de dados são previamente segmentadas com uso de uma U-Net [72]. A classificação binária foi feita entre COVID-19 e Não-COVID-19, obtendo AUC de 0.996.

Ainda em março de 2020, Zheng *et al.* reuniram imagens de TC de 542 pacientes, sendo 313 diagnosticados com COVID-19 e 229 sem diagnóstico de COVID-19 (Não-COVID-19) para propor um modelo de classificação binária [77]. O modelo proposto, DeCoVNet, faz operações comuns em 3D das RNCs com blocos residuais, a fim de se aproveitar as características espaciais das fatias da TC. Aumento de Dados foi utilizado no treinamento para melhor generalização e o modelo atingiu AUC de 0.959.

No trabalho de Wang *et al.* (2020) [78], também foi construído um conjunto de dados próprio com 325 imagens de TC de COVID-19 confirmada e 740 de outras pneumonias virais. Número de fatias de TC utilizada não informada. Para teste do conjunto de dados, foi utilizado um modelo

modificado da Inception [58] com transferência de aprendizado, a **M-Inception**. Este modelo modificado obteve 0.93% de AUC, 89.5% de acurácia, sensibilidade de 0.88, especificidade de 0.87 e *F1-Score* de 0.77.

Gunraj *et al.*(2020) [45] continuaram os trabalhos do COVIDx e COVID-Net [41], apresentados na Subseção anterior, ainda fazendo classificação multi-classe (Normal, Não-COVID-19, COVID-19) mas mudando a base de diagnóstico de Raio-X para TC. Eles também organizaram seu conjunto de dados próprio, o COVIDx-CT. Este conjunto possui 104.009 fatias de imagens TC da região do tórax de 1.489 pacientes e é de código aberto. O trabalho seguiu uma metodologia semelhante à apresentada no caso de Raio-X [41], utilizando de um projeto com exploração automática de micro e macro estruturas de arquiteturas para construção da RNC. Como descrito em Wang *et al.* (2020) [41], esta abordagem permite maior flexibilidade durante o projeto. O modelo final, COVIDNet-CT, obteve respectivamente para as classes Normal, Não-COVID-19 e COVID-19: sensibilidade de 100%, 99% e 97.3%; precisão de 99.4%, 98.4% e 99.7%; especificidade de 99.5%, 99.2% e 99.9%.

Misztal *et al.* (2020) [46] juntaram os conjuntos de dados do Cohen *et al.* [6] de Raio-X e o reunido por Yang *et al.* [65] de TC para fazer diversos testes com diferentes arquiteturas. Aqui serão apresentados apenas os resultados da classificação multi-classe ("Normal", "Não-COVID-19", "COVID-19") de TC. Os resultados de Raio-X foram apresentados na Subseção anterior. Em TC, a arquitetura com melhor resultado foi a ResNet-18 com métricas globais de: 0.73 de acurácia; 0.82 de precisão; 0.83 de sensibilidade; 0.82 de *F1-Score*.

Li *et al.* (2020) [47] coletaram um conjunto de dados aproximadamente balanceado próprio com 1292 imagens de TC de COVID-19, 1735 de "Não-COVID-19" e 1325 "Normal". Propuseram então um *framework*, o CONVNet, que: 1) recebe uma série de fatias de TC como entrada em uma ResNet50 [2]; 2) as características extraídas das fatias são combinadas com uma operação de *max-pooling*; 3) o mapeamento final de características é realizado classificado por camada totalmente conectada + função *softmax*. O CONVNet obteve, respectivamente para as classes "COVID-19", "Não-COVID-19" e "Normal": sensibilidade de 0.90, 0.87, e 0.94; e especificidade de 0.96, 0.92 e 0.96.

Capítulo 3

Materiais e Métodos

No desenvolvimento de aplicações de Aprendizado de Máquina, não só é necessário para o projetista saber implementar as técnicas e algoritmos corretamente como também é imprescindível que ele conheça em profundidade todo o processo fim-a-fim. Se tratando de um processo altamente iterativo, primeiramente é definido o *pipeline* do projeto juntamente com as métricas a se utilizar para validação de resultados. Desta forma, sabendo responder às saídas do problema, o projetista poderá gradualmente aprimorar cada uma das etapas, seja recolhendo mais dados, depurando as implementações de *software* ou ainda reavaliando a arquitetura do modelo. Na **Figura 3.1** a seguir, vemos o fluxograma do *pipeline* desenvolvido neste trabalho. As etapas propostas serão descritas nas Seções ao longo deste capítulo juntamente com o material utilizado.

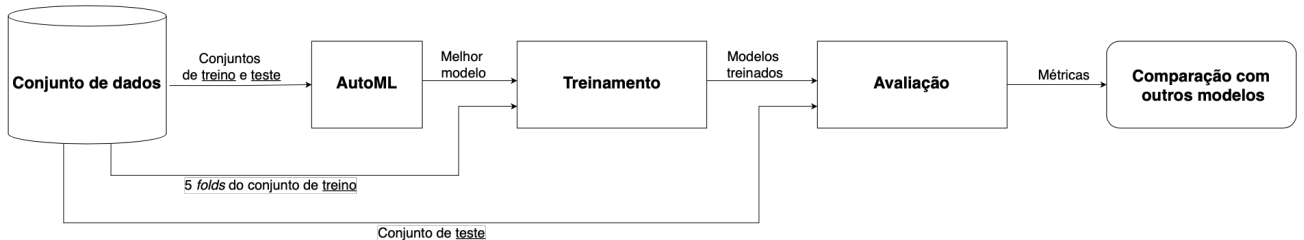


Figura 3.1: Fluxograma da metodologia.

3.1 Conjunto de Dados

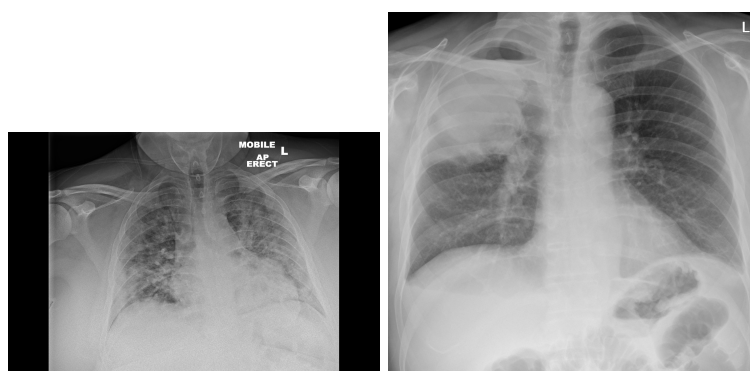
A aquisição e organização dos dados é uma das partes mais importantes no Aprendizado de Máquina. Considerando os tipos de dados, definimos aspectos importantes da arquitetura do modelo. Seguindo a referência da COVID-Net e do COVIDx [41] apresentado na Subseção 2.3.1, foi realizado um agrupamento de 5 conjuntos de dados diferentes [6, 7, 8, 9, 10] de imagens de Raios-X a fim de se obter um COVIDx [41] modificado. Este conjunto será tratado como **COVIDx-m**. Cada um dos conjuntos geradores possui características e organizações diferentes.

3.1.1 COVID-19 image data collection

O primeiro e principal conjunto de dados para diagnóstico de COVID-19 baseado em Raio-X [6] reúne centenas de imagens de pacientes de mais de 26 países, com diferentes tipos de pneumonia, além da COVID-19, separados em diferentes vistas.

Fora o diagnóstico da patologia encontrada e a vista da imagem, há também diversas informações suplementares, como: *offset*, que indica o número de dias desde o início dos sintomas ou hospitalização; sexo e idade do paciente; *survival*, indicando se o paciente sobreviveu ou não, *temperature*, sendo a temperatura em graus Celsius do paciente no momento da imagem, entre outras. Estas informações não foram utilizadas neste trabalho mas podem ser úteis em trabalhos futuros, seja para ajudar na classificação da doença baseado na gravidade ou até um estudo de chance de sobrevivência.

Para a criação do nosso conjunto de dados, foram reunidas todas as imagens de vistas **frontal** classificadas como "COVID-19" e outras pneumonias.



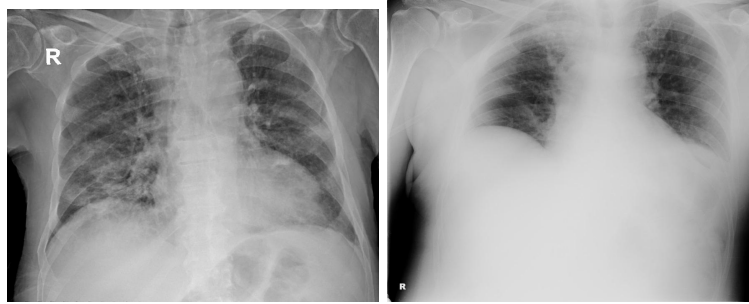
(a) Exemplo de Raio-X com diagnóstico de COVID-19. (b) Exemplo de Raio-X com diagnóstico de pneumonia (Não-COVID-19).

Figura 3.2: Imagens de Raio-X frontais extraídas do conjunto de dados original COVID-19 image data collection [6].

3.1.2 Figure 1 COVID-19 Chest X-ray Dataset Initiative

Este conjunto de dados foi organizado pelo time da COVID-Net em colaboração com a plataforma Figure 1 [7]. Este conjunto reúne 55 imagens de diagnóstico de pneumonia/COVID-19/não-encontrado, todas de vista frontal, também com informações auxiliares dos pacientes. [7]

Todas as imagens de "COVID-19" e "Pneumonia" deste conjunto de dados foram utilizadas no **COVIDx-m**.



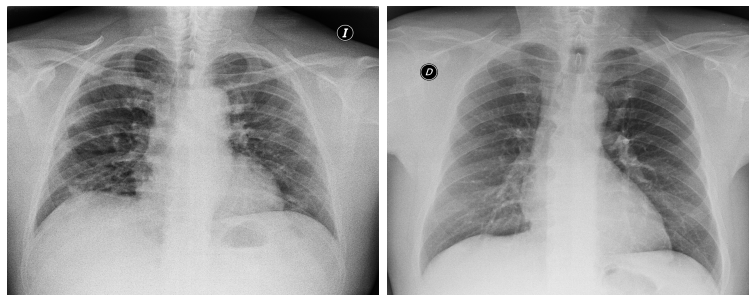
(a) Exemplo de Raio-X com diagnóstico de COVID-19. (b) Exemplo de Raio-X com diagnóstico de pneumonia (Não-COVID-19).

Figura 3.3: Imagens de Raio-X frontais extraídas do conjunto de dados original Figure 1 COVID-19 Chest X-ray Dataset Initiative [7].

3.1.3 Actualmed COVID-19 Chest X-ray Dataset Initiative

Também organizado pela equipe da COVID-Net, este conjunto foi reunido com auxílio da plataforma de gestão de imagens médicas ActualMed [8] em colaboração com a Universitat Jaume I. Este conjunto reúne 238 imagens no mesmo formato dos dois conjuntos anteriores, com visão frontal e informações auxiliares dos pacientes.

As imagens deste conjunto estão classificadas em "COVID-19" e "não encontrado", sendo apenas as de "COVID-19" utilizadas no **COVIDx-m**.



(a) Exemplo de Raio-X com diagnóstico de COVID-19. (b) Exemplo de Raio-X sem diagnóstico de COVID-19.

Figura 3.4: Imagens de Raio-X frontais extraídas do conjunto de dados original Actualmed COVID-19 Chest X-ray Dataset Initiative [8].

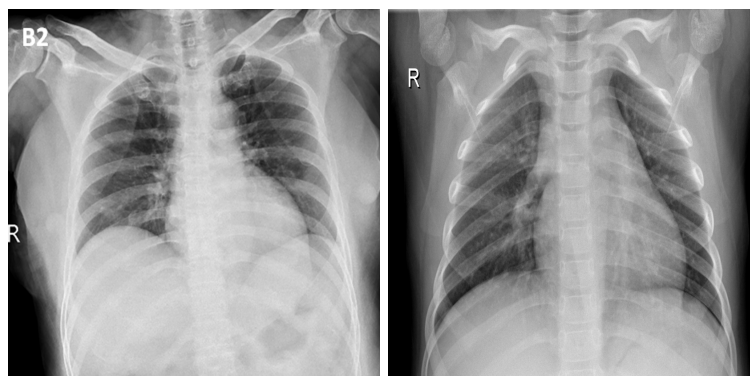
3.1.4 COVID-19 Chest X-ray Database

Já este conjunto de dados foi divulgado e publicado na plataforma Kaggle ¹ por um grupo de pesquisadores da Universidade do Qatar e Universidade de Dhaka, Bangladesh em colaboração

¹<https://www.kaggle.com/>

com pesquisadores do Paquistão e Malásia [9]. Os dados estão organizados em 3 classes: 219 de "COVID-19", 1341 da classe "Normal" e 1345 de "Pneumonia-Viral", reunindo imagens de uma série de outras 43 publicações.

As imagens são todas frontais e apenas as diagnosticadas como "COVID-19" foram utilizadas no **COVIDx-m**. Quando disponível, as imagens de COVID-19 possuem *link* da página de origem de onde foram retiradas. Desta forma, pode-se fazer uma filtragem de imagens duplicadas no conjunto final.



(a) Exemplo de Raio-X com diagnóstico de COVID-19. (b) Exemplo de Raio-X sem diagnóstico de doença.

Figura 3.5: Imagens de Raio-X frontais extraídas do conjunto de dados original COVID-19 Chest X-ray Database [9].

3.1.5 RSNA Pneumonia Detection Challenge

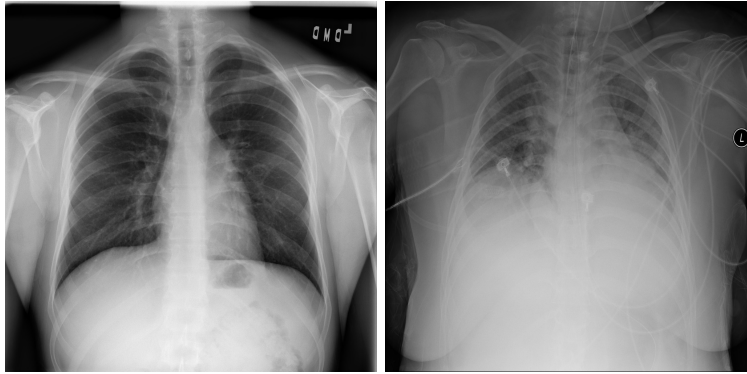
Este conjunto de dados [10] foi publicado na plataforma Kaggle como base para competição de algoritmos de detecção de sinais de pneumonia em imagens médicas. O conjunto possui mais de 30 mil imagens frontais com diagnóstico: Normal, anomalia detectada no pulmão e pneumonia confirmada.

Apenas as imagens normais e de pneumonia confirmada foram utilizadas na construção do **COVIDx-m**.

3.1.6 Conjunto Final

Reunindo todos os dados, o **COVIDx-m** possui um total de 15540 imagens categorizadas em 3 classes: Normal, Pneumonia (Não COVID-19) e COVID-19. As distribuições de amostras de cada classes estão mostradas na Tabela 3.1.

A partir deste total, o conjunto total foi separado em dois subconjuntos: treino e testes. A escolha foi de uma proporção de 80%/20%. A fim de se manter uniformidade com o conjunto COVIDx original, tentou-se separar os mesmos dados nos conjuntos em todos os conjuntos, com exceção do "RSNA Pneumonia Detection Challenge". Para este último, devido a seu tamanho,



(a) Exemplo de Raio-X sem diagnóstico de doença. (b) Exemplo de Raio-X com diagnóstico de pneumonia (Não-COVID-19).

Figura 3.6: Imagens de Raio-X frontais extraídas do conjunto de dados original RSNA Pneumonia Detection Challenge [10].

preferiu-se fazer uma separação básica de 20% para testes. Ao final, temos 12.460 imagens no conjunto de treino e 3.080 no conjunto de testes. Pode-se ver que a proporção da separação de dados de fato se aproxima de 80%/20%.

Na Tabela 3.1 a seguir, encontra-se uma descrição mais detalhada da separação do conjunto juntamente com a quantidade de imagens de cada classe.

Tabela 3.1: Distribuição de dados por classe do conjunto COVIDx-m, separados em "treino" e "teste".

Imagens no COVIDx-m		
Classe / Subconjunto	Treino	Teste
Normal	7084	1767
Pneumonia	4872	1210
COVID-19	504	103
Total	12.460	3.080

3.2 AutoML

Essa etapa consiste em apresentar o conjunto de dados obtido na etapa anterior à nossa ferramenta de AutoML. É feita então uma busca de arquiteturas de modelos com parâmetros especificamente escolhidos para classificação dos nossos dados. A saída final desta etapa é o modelo a ser utilizado na etapa posterior de treinamento.

A nossa ferramenta de AutoML, é o AutoKeras. Apesar de AutoML se tratar de um processo automático, há alguns parâmetros que devemos ajustar e trabalhar antes de se iniciar a busca pelo modelo.

3.2.1 Pré-processamento

Características morfológicas de uma imagem influenciam diretamente na arquitetura de um modelo de RNC. Logo, decidido como será feito o diagnóstico, deve-se entregar ao AutoKeras o conjunto de dados mais próximo possível de como será feita a classificação, para que os parâmetros de rede e os modelos sejam coerentes com os dados.

Assim, foram feitas alguns pré processamentos nas imagens de entrada:

1. Redimensionamento para 240x240

Este valor foi escolhido de forma arbitrária, mas partindo do valor utilizado por Wang *et al.*(2020) [41]. Este tamanho de imagem também é típico na literatura para RNCs.

2. Mudança de espaço de cores para tons de cinza

Devido à natureza das imagens de Raios-X, espaços de cores de mais de uma dimensão não são necessários, portanto as imagens são todas lidas em tons de cinza, com valores de pixel entre 0 e 255.

3. Normalização de valores de pixel para 1/255

Uma técnica muito comum para desviar do problema de explosão de gradiente é a mudança de escala para valores entre 0 e 1. Desta forma, a informação da relação entre as intensidades de pixels se mantém porém os cálculos de treinamento são facilitados com valores menores.

3.2.2 Aumento de Dados

RNCs não aprendem bem informações espaciais, tendo dificuldade de generalizar padrões em orientações diferentes. Uma técnica bastante simples para resolver esse problema é a utilização de Aumento de Dados. Isto é, aplicar transformações semi-aleatórias às imagens de entrada que aumentem a variabilidade do conjunto a ser ensinado e permita uma melhor generalização.

Neste trabalho foram utilizadas 3 transformações:

1. Rotação de imagem de 30°

Para entradas aleatórias, foi aplicada uma rotação de valor e sentido aleatório entre 0 e 30°.

2. *Zoom* de 30%

Para entradas aleatórias, foi aplicado um *zoom* com valor aleatório entre 0 e 30%.

3. Inversão Horizontal

Aleatoriamente algumas imagens são invertidas horizontalmente a fim de se retirar vieses de visão (frontal, etc).

Na Figura 3.7 a seguir são apresentados 6 exemplos de imagens do COVIDx-m após aplicação das transformações de Aumento de Dados.

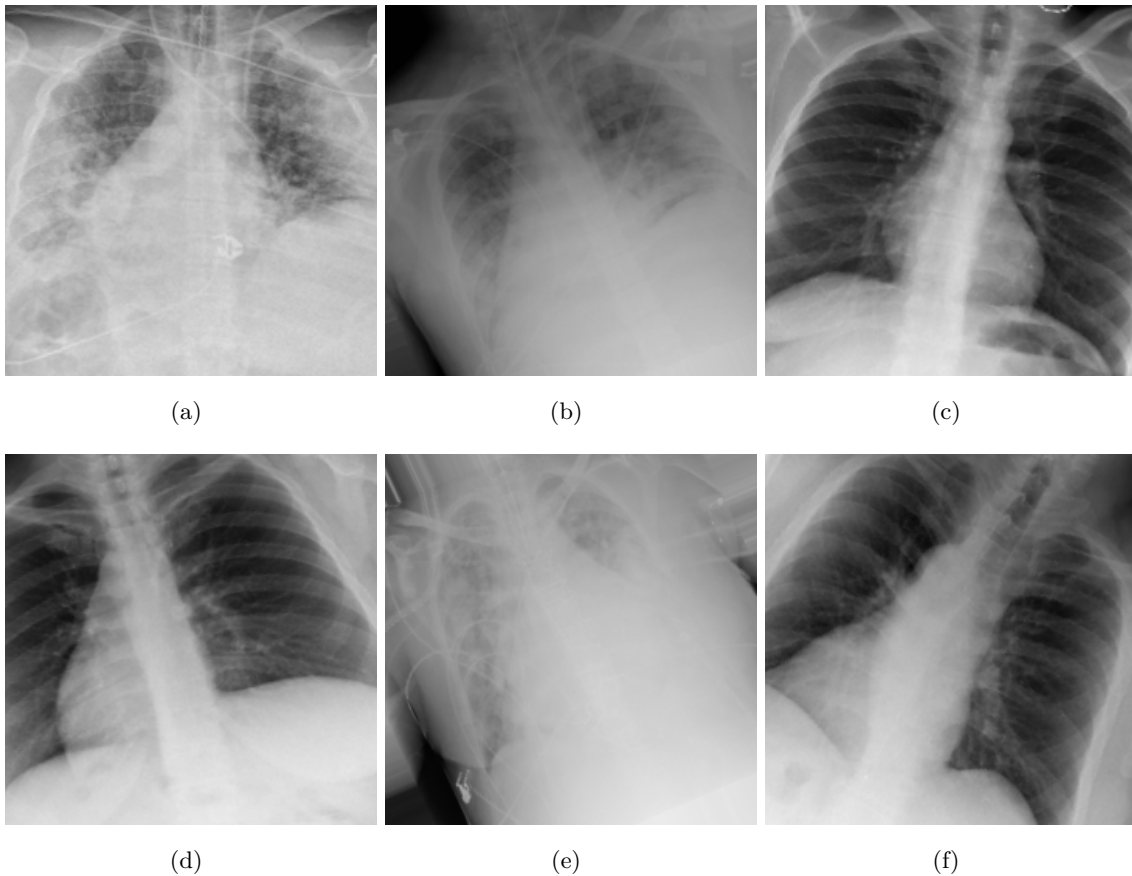


Figura 3.7: Exemplos de imagens do COVIDx-m com diferentes níveis de rotação e de *zoom*.

3.2.3 Restrição do Espaço de Busca

Sabendo dos bons resultados obtidos nos diversos trabalhos apresentados na Seção 2 e a fim de se otimizar o processo, o espaço de busca dos modelos foi restringido à família de ResNets. Ainda assim, o AutoML continuaria buscando otimizações infinitamente, por isso também foi definido um número máximo de 20 modelos. Assim, conseguimos acompanhar a evolução com clareza, salvando sempre o melhor modelo.

3.2.4 Treino dos Modelos durante a Busca

Durante o processo de busca de modelo, é feita uma etapa de treinamento do modelo para que as métricas sejam calculadas e o algoritmo possa tomar decisões de otimizações. Para esse treinamento, foi definido um período de **100** épocas com 32 imagens por *batch*. Estes valores foram escolhidos de forma arbitrária mas se mostraram coerentes, sendo suficientes para estabilização. A fim de prevenir também significativos *overfittins*, também foi utilizado um protocolo de parada prematura, *early stopping*, com paciência de 10 épocas na métrica de *loss*. Isto é, se nenhuma melhora é detctada no intervalo de 10 épocas o treinamento é interrompido.

Também foram passados todos os dados do conjunto COVIDx-m, os de treino sendo utilizados

para treino efetivo da rede e os de teste para avaliação das métricas após cada época. Informações da rotina de treino serão apresentadas no Capítulo 4.

3.2.5 Modelo Final

Depois de todas essas etapas anteriores, o AutoKeras salva o melhor modelo encontrado em relação à todas as outras tentativas. Este modelo agora será utilizado para treinamento da rede do zero.

O resultado do modelo obtido pode ser visto no Capítulo 4.

3.3 Treinamento

Neste ponto, temos o conjunto de dados completamente construído e o modelo obtido pelo AutoKeras pronto para ser treinado. Como explicado na Seção 3.2, este modelo já foi pré treinado durante o processo de busca de modelos. Como desejamos validar que o modelo encontrado é capaz de resolver o problema proposto, os pesos da rede foram removidos e o modelo inteiro foi treinado do zero.

3.3.1 Pré Processamento

Os mesmos processos de pré processamento utilizados na Subseção 3.2.1 foram feitos neste ponto. Assim, foi garantido que o modelo receberia como entrada imagens com as mesmas características morfológicas, sem discrepâncias de performance.

3.3.2 Desbalanceamento de dados

A fim de se lidar com a diferença na proporção de dados referentes a cada classe no conjunto de treino, um parâmetro de peso é considerado na rotina de treino de modo a se ponderar o desbalanceamento. Isso não resolve o problema em sua totalidade, mas ajuda a mitigá-lo.

Foi decidido não utilizar *undersampling* para balanceamento forçado do conjunto de dados, pois foi entendido que a utilização do conjunto total representa com maior veracidade os desafios de treinamento que o modelo encontraria em casos reais de classificação.

Os pesos foram calculados com auxílio da função `compute_class_weight()` da biblioteca SciKitLearn ².

3.3.3 Treinamento e Validação Cruzada

Foram feitas duas rotinas principais de teste, cada uma para um otimizador diferente: Adam [79] e SGD [80]. Para cada otimizador, foram feitas duas sub-rotinas de treino: uma com Aumento

²<https://scikit-learn.org/>

de Dados e outra com os dados puros. Dessa forma, temos então 4 tipos de modelos: Adam + Aumento de Dados, Adam + *raw*, SGD + Aumento de Dados, SGD + *raw*.

A fim de se confirmar a performance dos modelos e garantir que o conjunto de dados utilizado é capaz de representar as características das classes, foi utilizado o método de validação cruzada para cada um dos quatro tipos de modelo. Assim, o processo de treinamento não foi realizado com o conjunto completo de treino, mas sim com uma fatia desse modelo. O parâmetro para separação das fatias foi também 80%/20%, gerando cinco subconjuntos de treino/teste. Os dados para validação por época foram as fatias de 20%. Ao final então, foram realizados **20** rotinas de treinamento: 5 *folds* para cada tipo de modelo. Os parâmetros de treinamento utilizados se mantiveram os mesmos para todas as rotinas: 1000 épocas máximas e 32 imagens por *batch*. Para os treinamentos também foi utilizado um protocolo de parada prematura, *early stopping*, com paciência de 50 épocas na métrica de *loss*.

A Figura 3.8 a seguir apresenta o fluxo de treinamento com validação cruzada e a Tabela 3.2 apresenta a separação dos modelos treinados por otimizados conjunto de dados.

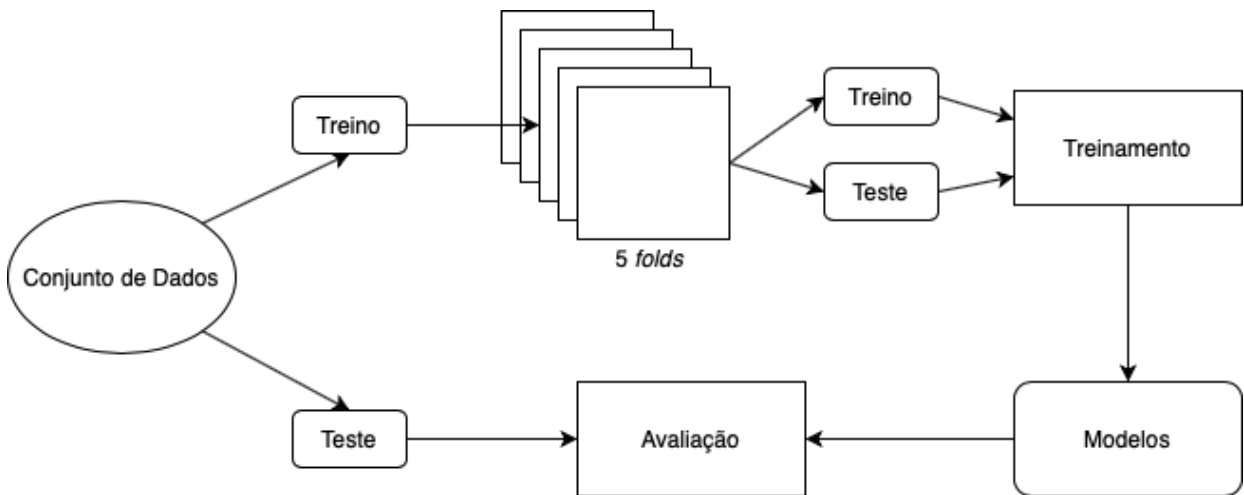


Figura 3.8: Fluxo de separação do conjunto de dados de treino para sub-conjuntos de validação cruzada, e teste para validação dos modelos.

Tabela 3.2: Composição dos modelos treinados.

Modelos Treinados		
Otimizador	Conjunto de Dados	Modelos
Adam	Aumento de Dados	5
	Raw	5
SGD	Aumento de Dados	5
	Raw	5
Total		20

3.4 Avaliação

Para a avaliação dos modelos, foi utilizado o conjunto completo de testes original, garantindo que nenhuma das imagens nunca tivesse sido vista por nenhum dos modelos. As métricas obtidas foram: precisão, sensibilidade e *F1-score*, estes apresentados na Seção 2.2, para cada uma das classes, além das acurácias, tanto global quanto ponderadas pelas classes.

A partir destes dados, como métrica qualitativa, foi gerada a curva ROC multi-classe do modelo. Como métrica quantitativa, foram geradas matrizes de confusão.

Com os dados de cada um dos modelos foi possível obter a média das métricas e se decidir qual dos 4 tipos possui a melhor performance. Essas métricas então foram comparadas com os valores da Tabela 2.1 e os resultados podem ser encontrados na Seção 4.

3.5 *Hardware e Software* utilizados

Para busca de modelos com AutoKeras foi utilizado um computador com as seguintes especificações:

- Processador **AMD Ryzen Threadripper**: 32 cores e 3.7GHz de *clock*;
- 2x GPUs **NVIDIA GeForce RTX 2080**: 8Gb de memória.

Para treinamento em validação cruzada, foi utilizada a plataforma *Google Colab*³. Esta plataforma nos permite executar códigos *Python* no formato *Jupyter Notebook* com aceleração de GPU. Esta GPU utilizada é uma **NVIDIA Tesla V100** com 16Gb de memória.

Todo o código de treinamento foi desenvolvido na linguagem *Python3* no formato *Jupyter Notebook* com auxílio de suas bibliotecas científicas e *frameworks* de aprendizado profundo: SciKitLearn⁴, NumPy⁵, Tensorflow⁶, Keras⁷, além do AutoKeras já mencionado na Seção 2.1.4.1.

³<https://colab.research.google.com/>

⁴<https://scikit-learn.org/>

⁵<https://numpy.org>

⁶<https://www.tensorflow.org>

⁷<https://keras.io>

Capítulo 4

Resultados

Neste capítulo serão apresentados os resultados da metodologia descrita no Capítulo anterior e em seguida é comparado os resultados obtidos com os da literatura, apresentados no Capítulo 3. Ao fim é discutido perspectivas futuras e possíveis pontos de melhoria.

4.1 Busca e Treinamento no AutoKeras

Com as configurações de busca descritas no Capítulo 3, o espaço de busca resultante pode ser visto na Tabela 4.1. Cada modelo testado é uma combinação de cada um desses hiperparâmetros juntamente com configurações de otimizadores e seus parâmetros de aprendizado.

Tabela 4.1: Tabela dos Hiperparâmetros variáveis para busca de melhor modelo do AutoKeras.

Espaço de Busca do AutoKeras	
Blocos ResNet	ResNet50
	ResNet101
	ResNet152
	ResNet50_v2
	ResNet101_v2
	ResNet152_v2
Redução Espacial	flatten
	global_max
	global_avg
Dropout	0
	0.25
	0.5

Os gráficos dos valores de *loss* e acurácia de validação dos 20 modelos testados encontram-se nas Figuras 4.1 e 4.2 a seguir.

Podemos ver que alguns modelos alcançaram estabilidade de *loss* em torno de 20 épocas en-

quanto outros perduraram até o limite de 100.

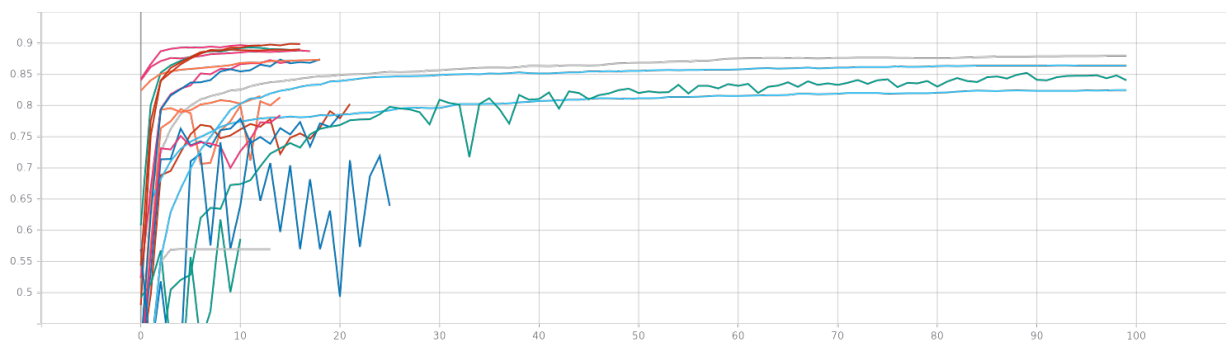


Figura 4.1: Gráfico de acurácia em validação dos 20 modelos testados pelo AutoKeras ao longo das épocas de treinamento.

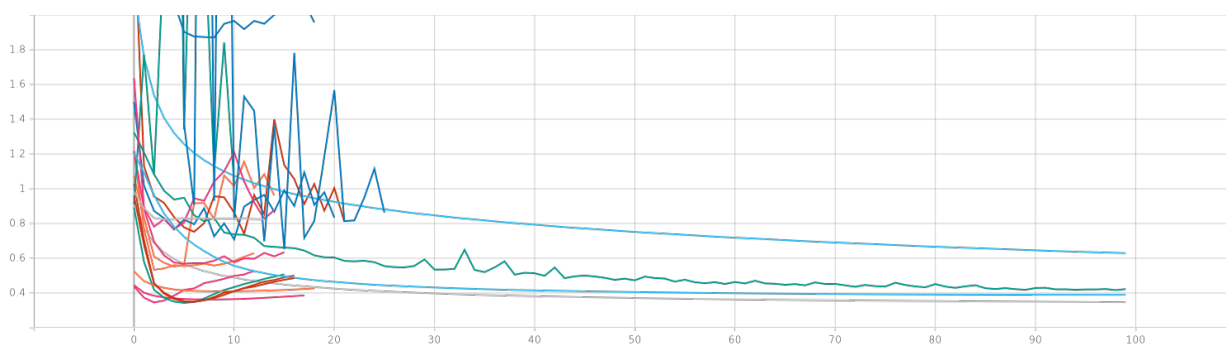


Figura 4.2: Gráfico de *loss* em validação dos 20 modelos testados pelo AutoKeras ao longo das épocas de treinamento.

O tempo total de busca e os tempos de cada modelo podem ser vistos na Tabela 4.2.

Tabela 4.2: Tempo de busca e pré treinamento de cada modelo durante o processamento no AutoKeras.

Tempo de Busca no AutoKeras			
	Tempo		Tempo
Modelo 1	37min	Modelo 11	24min
Modelo 2	18min	Modelo 12	29min
Modelo 3	36min	Modelo 13	19min
Modelo 4	26min	Modelo 14	23min
Modelo 5	20min	Modelo 15	1h 40min
Modelo 6	17min	Modelo 16	20min
Modelo 7	18min	Modelo 17	27min
Modelo 8	1h 39min	Modelo 18	2h 47min
Modelo 9	1h 39min	Modelo 19	42min
Modelo 10	26min	Modelo 20	24min
Total		14h 30min	

4.2 Modelo Obtido

Uma síntese do modelo final obtido pela AutoKeras pode ser visto na Figura 4.3. A partir deste momento chamaremos este modelo **AutoResCovidNet**.

Analisando esta imagem vemos que o modelo é formado por uma estrutura principal de ResNet50V2 com adição de uma sequência de entrada de camadas como preparação de dados e uma sequência final de camadas formando um *head* de classificação. A estrutura completa da ResNet50 pode ser conferida nos artigos originais [2] com as modificações das redes residuais V2 [57].

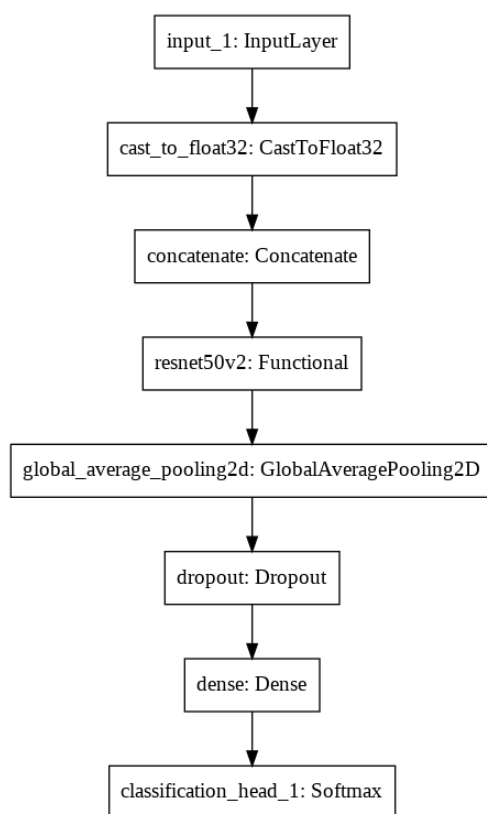


Figura 4.3: Resumo do modelo final obtido pelo AutoKeras.

Na Figura 4.4 vemos as camadas de preparação adicionadas ao início da estrutura e na Figura 4.5, as camadas de *head* adicionadas ao fim da estrutura.

As camadas de entrada adicionadas ao início são respectivamente: 1) `cast_to_float32` e 2) `concatenate`. Essas camadas convertem o tipo de dado dos tensores de entrada para `float32` e em seguida os concatena retornando um único tensor. Essas modificações têm por objetivo preparar os dados dos tensores para que eles possam ser processados pelas camadas seguintes.

As camadas de saída adicionadas ao fim são respectivamente: 1) `global_average_pooling2d`; 2) `dropout`; 3) `dense`; e 4) `classification_head`. Esse conjunto de camadas é muito comum como saída de redes neurais convolucionais. Primeiramente, é feito um ***AveragePooling2D***, uma operação de filtragem de características médias em um espaço 2D. Essa saída então passa por um ***Dropout***, ou seja, unidades de entrada são aleatoriamente definidas para 0, de forma a regularizar

a rede e evitar *overfitting*. Em seguida, esses valores passam pela camada *Dense*, onde todos os neurônios compartilham pesos e estão totalmente conectados com a camada seguinte de saída, no caso, o número de neurônios é igual ao número de classes a serem classificadas. Essa camada final de classificação então serve como entrada pro *ClassificationHead*, que é uma simples função matemática *softmax* que tem como saída a porcentagem de ativação dos neurônios referentes às classes a serem inferidas. A classificação realizada pelo modelo é a classe que apresenta o maior valor de ativação. Mais informações sobre essas camadas e operações podem ser encontradas no livro original de Goodfellow (2016) [1] e na documentação da biblioteca Keras¹.

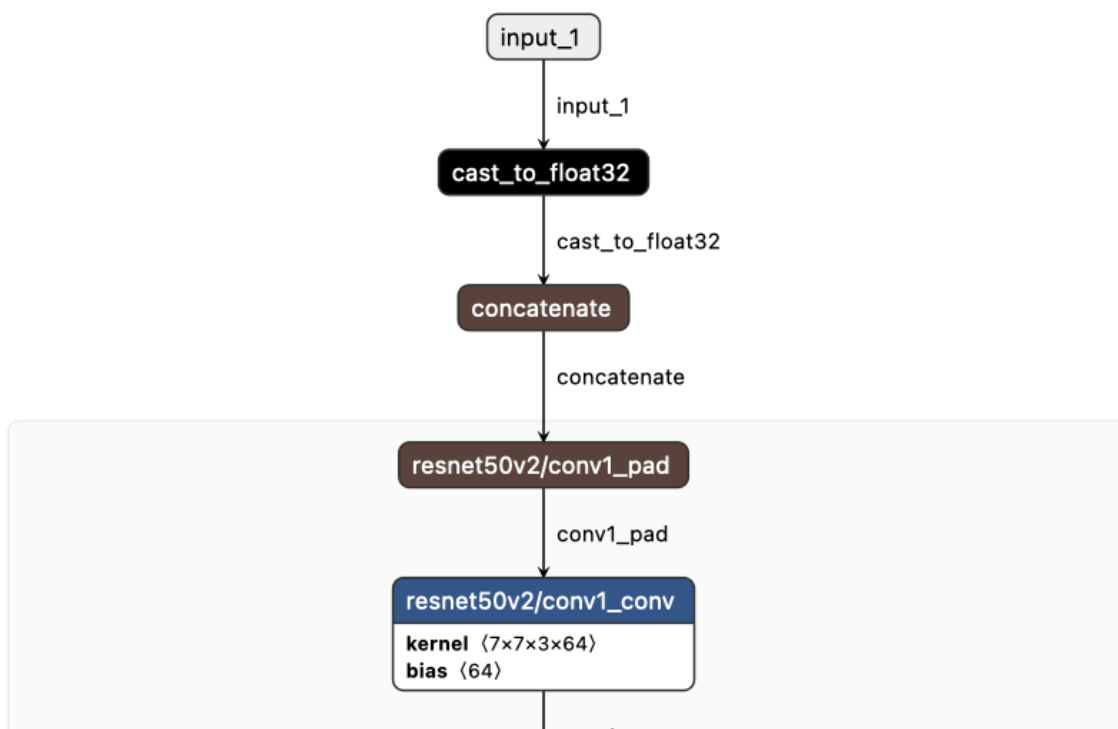


Figura 4.4: Representação das primeiras camadas do modelo final obtido.

O modelo final possui 23.570.947 de parâmetros com um total de 197 camadas.

Nas Seções a seguir serão feitas as análises qualitativas e quantitativas dos resultados dos treinamentos com otimizador Adam e otimizador SGD para o AutoResCovidNet. Para cada otimizador, os resultados estão separados em dois grupos: conjunto de dados *raw*, e conjunto de dados com AD. Para cada modelo é apresentado uma tabela de comparação com os resultados de treinamento de cada *fold* juntamente com as médias das métricas.

A fim de se facilitar a leitura, será utilizado como nomenclatura de referência para os modelos e combinações de treinamento: [otimizador]_[conjunto de dados]_[número do *fold*], por exemplo, **Adam_raw_1**. Quando o número do *fold* não for explicitado a referência é sobre o treinamento completo referente àquela configuração de otimizador + conjunto de dados.

¹<https://keras.io>

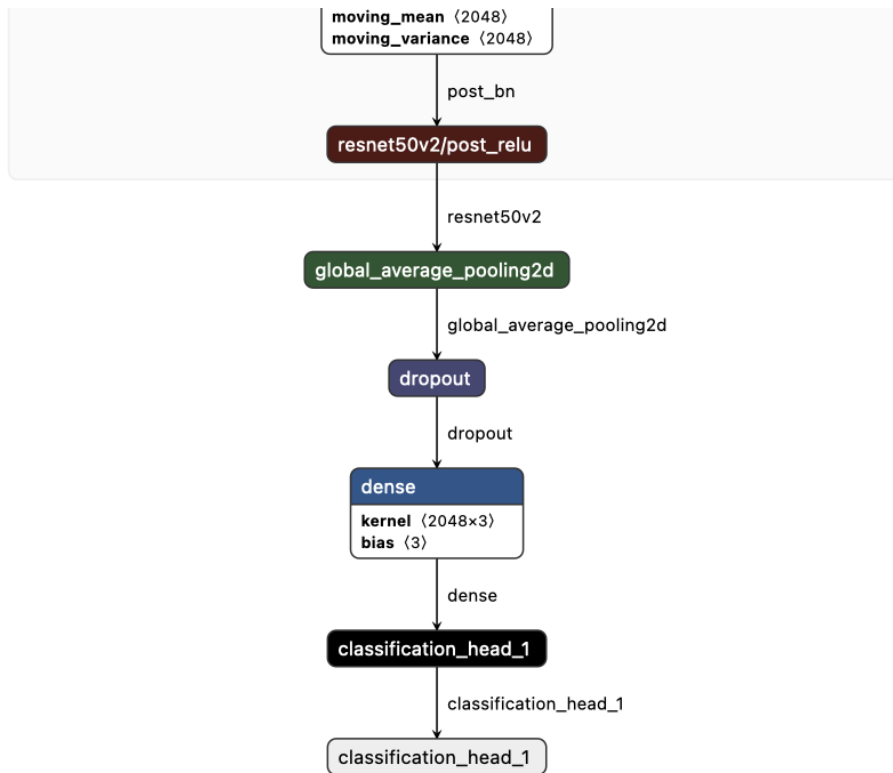


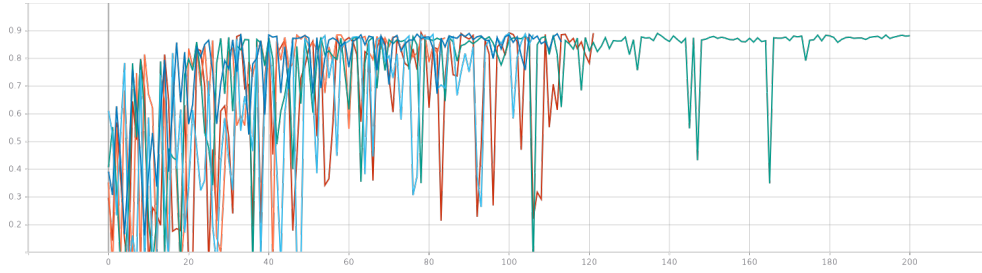
Figura 4.5: Representação das últimas camadas do modelo final obtido.

4.3 Treinamento de Validação Cruzada

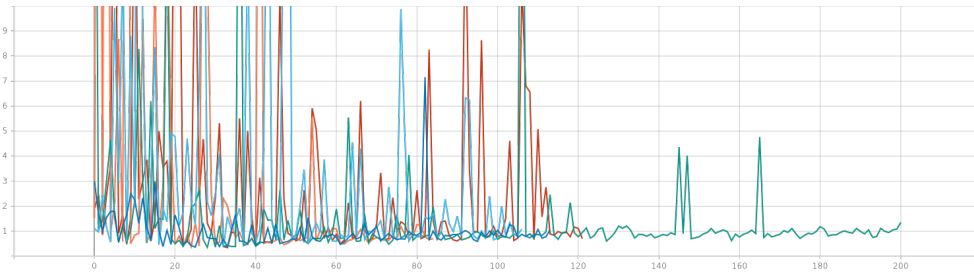
Assim como mostrado na Seção 4.1, nas Figuras 4.6, 4.7, 4.8 e 4.9 a seguir encontram-se os gráficos de acurácia e *loss* de validação de cada combinação de treinamento. Todas as *folds* estão no mesmo gráfico a fim de se facilitar comparação e visualização. Imagens individuais de cada modelo podem ser encontradas em Anexo. O tempo total de treinamento (somados os tempos de todas as *folds*) de cada um dos modelos encontra-se na Tabela 4.3.

Tabela 4.3: Tempo de treinamento de cada modelo testado durante o processo de validação cruzada.

Tempo de Treino de Validação Cruzada	
Modelos	Tempo Total
Adam_raw	10h
SGD_raw	8h 18min
Adam_augmented	18h 9min
SGD_augmented	14h 13min
Total	50h 40min

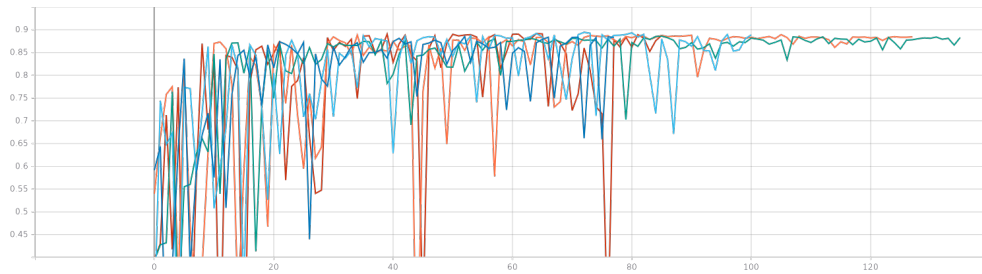


(a) Acurácia

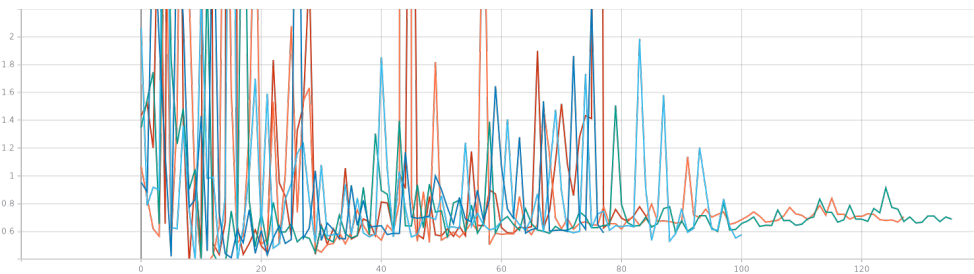


(b) loss

Figura 4.6: Gráficos de validação dos 5 *fold*s treinados no modelo *Adam_raw* ao longo das épocas de treinamento.

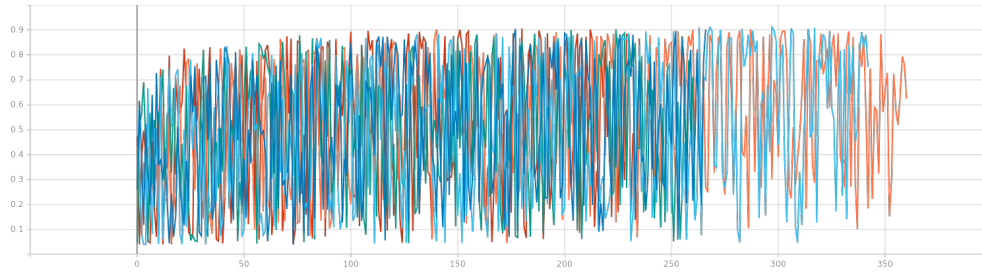


(a) Acurácia

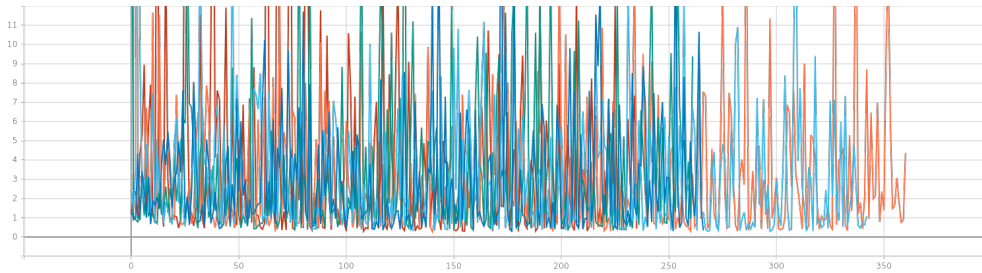


(b) loss

Figura 4.7: Gráficos de validação dos 5 *fold*s treinados no modelo *SGD_raw* ao longo das épocas de treinamento.

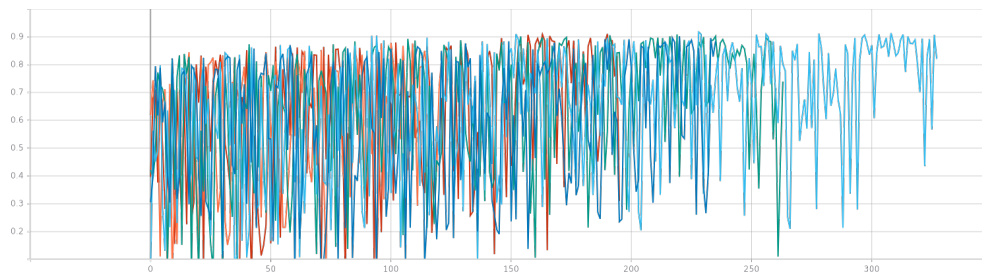


(a) Acurácia

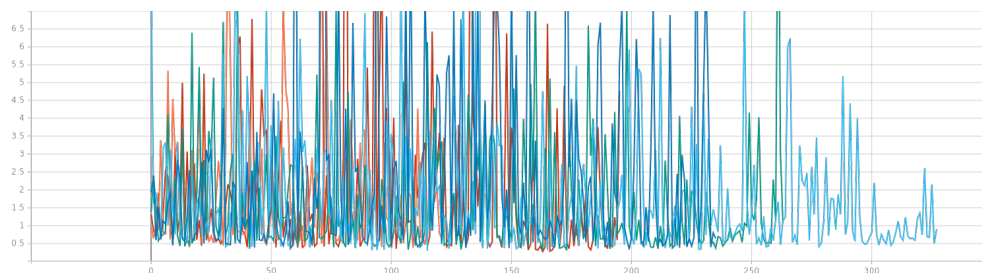


(b) loss

Figura 4.8: Gráficos de validação dos 5 *folds* treinados no modelo Adam_augmented ao longo das épocas de treinamento.



(a) Acurácia



(b) loss

Figura 4.9: Gráficos de validação dos 5 *folds* treinados no modelo SGD_augmented ao longo das épocas de treinamento.

4.4 Análise Qualitativa

Nas subseções a seguir, vemos as Tabelas 4.4, 4.5, 4.6 e 4.7 com os resultados de AUC dos sub-treinos **Adam_raw**, **Adam_augmented**, **SGD_raw** e **SGD_augmented**, respectivamente. Nas Figuras 4.10, 4.11, 4.12 e 4.13 encontram-se as curvas ROC de cada um dos *fold*s.

4.4.1 Conjunto de Dados *Raw*

Tabela 4.4: Tabela de resultados de AUC de validação cruzada do treino *Adam_raw*.

Area Under Curve - <i>Adam_raw</i>			
	Classe		
	Normal	Pneumonia	COVID-19
<i>fold 1</i>	0.9553	0.9486	0.9184
<i>fold 2</i>	0.9465	0.9438	0.9317
<i>fold 3</i>	0.9438	0.9398	0.9500
<i>fold 4</i>	0.9509	0.9450	0.9425
<i>fold 5</i>	0.9472	0.9419	0.9276
Média	0.9487	0.9438	0.9340

Tabela 4.5: Tabela de resultados de AUC de validação cruzada do treino *SGD_raw*.

Area Under Curve - <i>SGD_raw</i>			
	Classe		
	Normal	Pneumonia	COVID-19
<i>fold 1</i>	0.9499	0.9421	0.9143
<i>fold 2</i>	0.9486	0.9412	0.9156
<i>fold 3</i>	0.9453	0.9398	0.9178
<i>fold 4</i>	0.9502	0.9433	0.9081
<i>fold 5</i>	0.9494	0.9428	0.9195
Média	0.9486	0.9418	0.9150

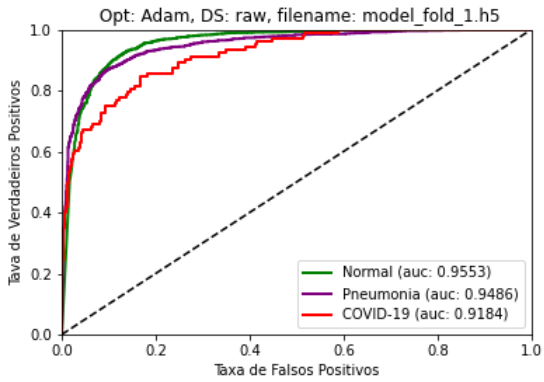
4.4.2 Conjunto de Dados com AD

Tabela 4.6: Tabela de resultados de AUC de validação cruzada do treino *Adam_augmented*.

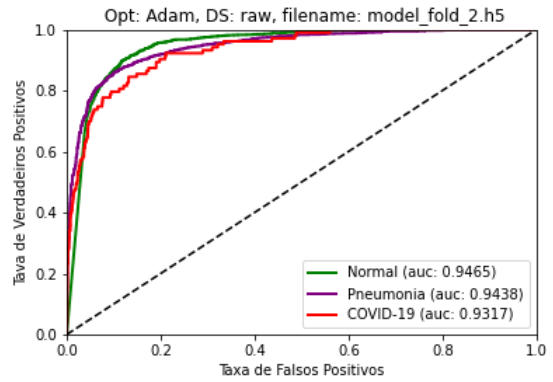
Area Under Curve - <i>Adam_augmented</i>			
	Classe		
	Normal	Pneumonia	COVID-19
<i>fold 1</i>	0.7353	0.6211	0.8792
<i>fold 2</i>	0.9401	0.9211	0.8825
<i>fold 3</i>	0.6982	0.7171	0.6084
<i>fold 4</i>	0.9598	0.9526	0.9544
<i>fold 5</i>	0.7804	0.7025	0.9121
Média	0.8227	0.7828	0.8473

Tabela 4.7: Tabela de resultados de AUC de validação cruzada do treino *SGD_augmented*.

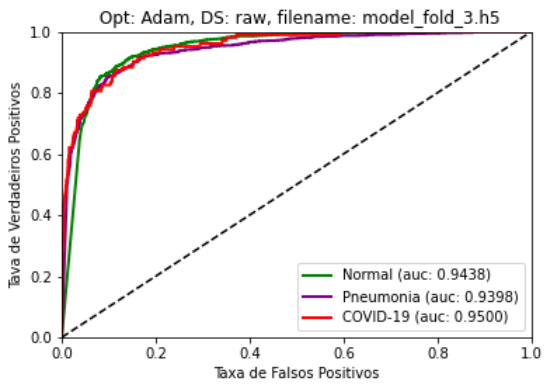
Area Under Curve - <i>SGD_augmented</i>			
	Classe		
	Normal	Pneumonia	COVID-19
<i>fold 1</i>	0.9255	0.9336	0.9206
<i>fold 2</i>	0.9456	0.9390	0.9230
<i>fold 3</i>	0.9249	0.9356	0.9423
<i>fold 4</i>	0.9301	0.9250	0.8792
<i>fold 5</i>	0.8980	0.8681	0.9003
Média	0.9248	0.9202	0.9130



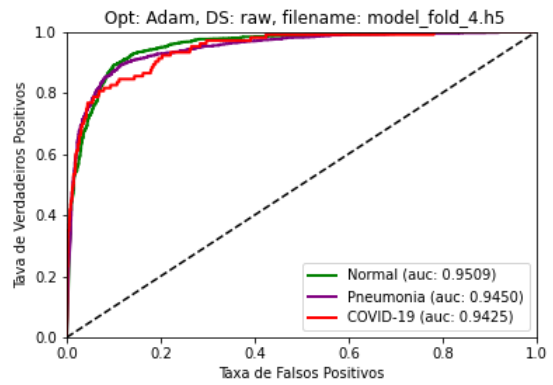
(a) *fold 1*



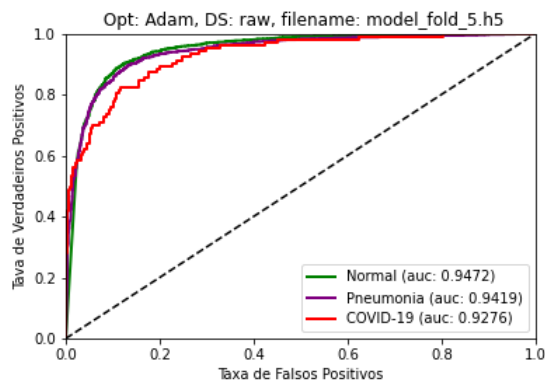
(b) *fold 2*



(c) *fold 3*

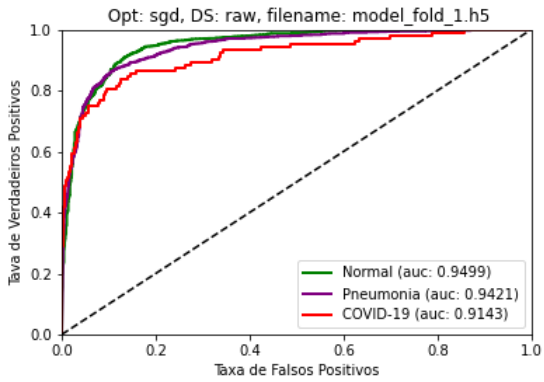


(d) *fold 4*

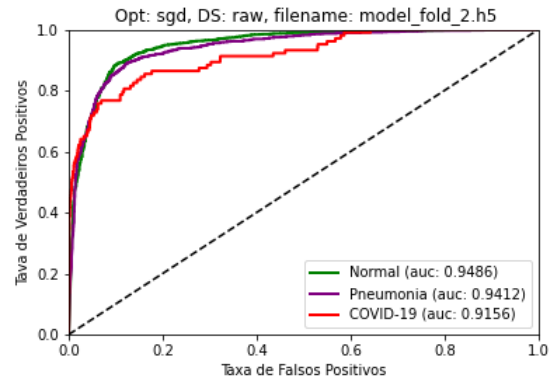


(e) *fold 5*

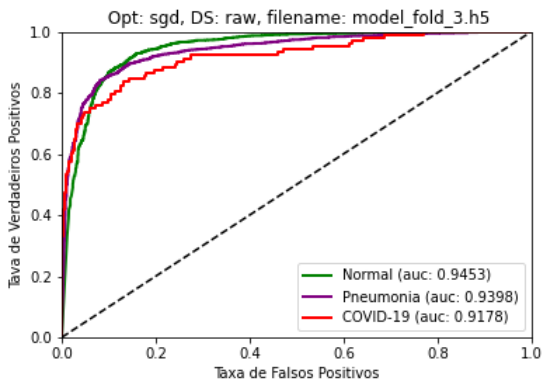
Figura 4.10: Curvas ROC de cada um dos *fold*s treinados para **Adam_{raw}**. As classes nas cores verde, roxo e vermelho são respectivamente "Normal", "Pneumonia" e "COVID-19".



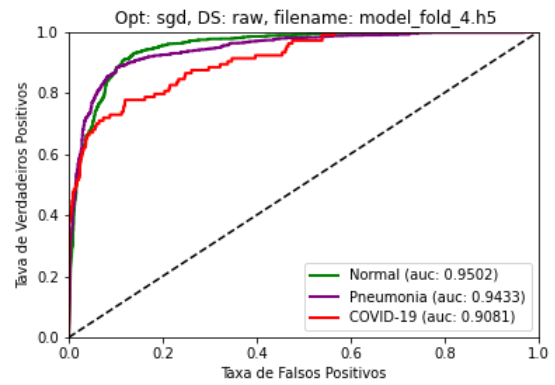
(a) *fold 1*



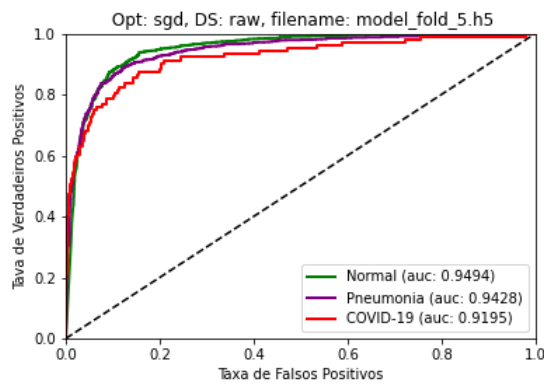
(b) *fold 2*



(c) *fold 3*

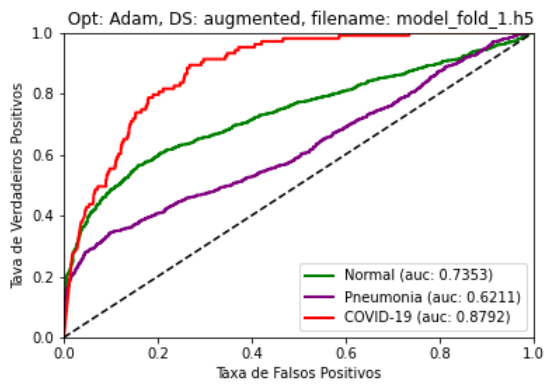


(d) *fold 4*

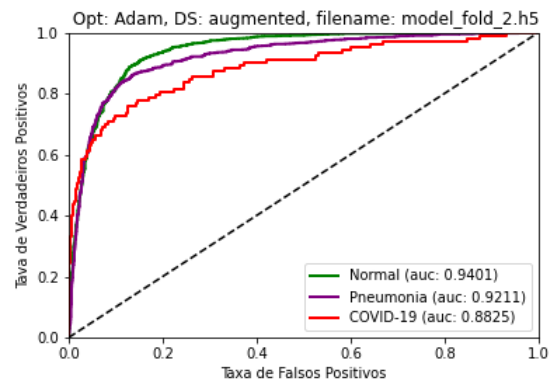


(e) *fold 5*

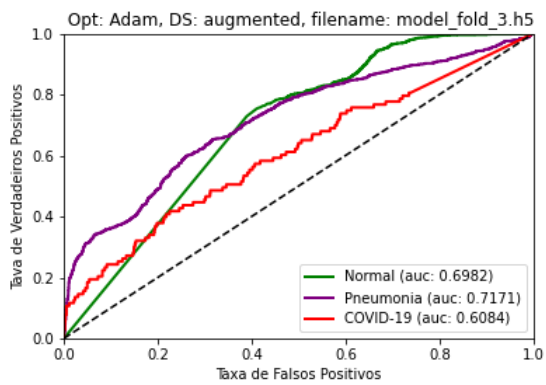
Figura 4.11: Curvas ROC de cada um dos *fold*s treinados para **SGD_{raw}**. As classes nas cores verde, roxo e vermelho são respectivamente "Normal", "Pneumonia" e "COVID-19"..



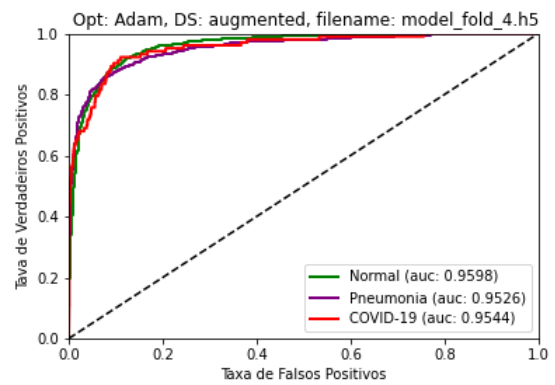
(a) *fold 1*



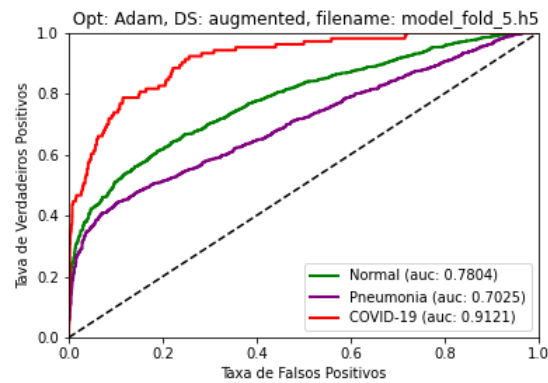
(b) *fold 2*



(c) *fold 3*

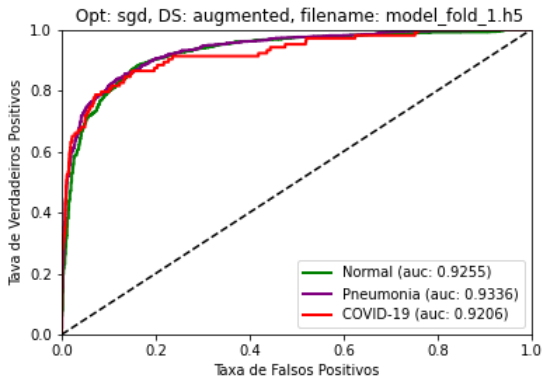


(d) *fold 4*

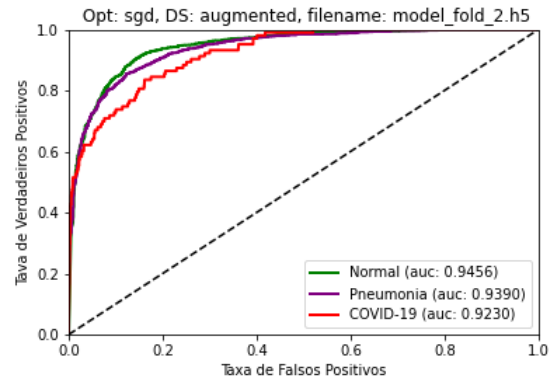


(e) *fold 5*

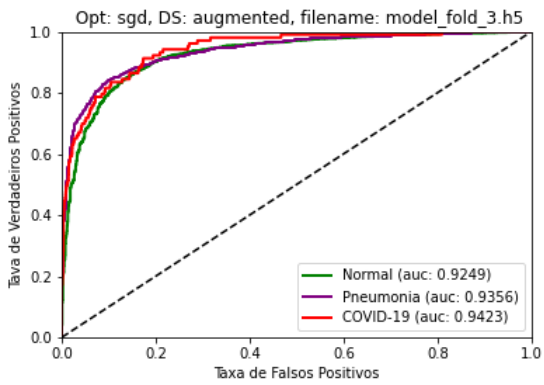
Figura 4.12: Curvas ROC de cada um dos *folders* treinados para **Adam_augmented**. As classes nas cores verde, roxo e vermelho são respectivamente "Normal", "Pneumonia" e "COVID-19".



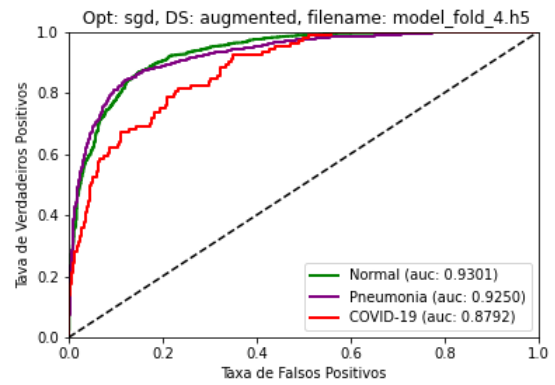
(a) *fold 1*



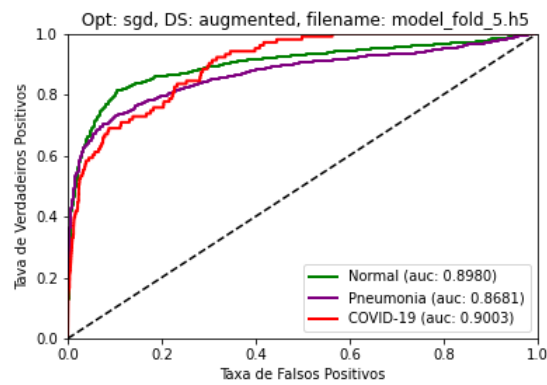
(b) *fold 2*



(c) *fold 3*



(d) *fold 4*



(e) *fold 5*

Figura 4.13: Curvas ROC de cada um dos *fold*s treinados para **SGD_augmented**. As classes nas cores verde, roxo e vermelho são respectivamente "Normal", "Pneumonia" e "COVID-19".

4.4.3 Validação

Para decidir a melhor entre duas combinações de treino, primeiro foram comparado os valores médios das AUCs de cada classe. O melhor modelo é aquele que vence em mais classes. Estes valores de comparação podem ser encontrados na Tabela 4.8.

Tabela 4.8: Resumo dos valores de AUC por classe para cada modelo.

Análise Qualitativa de Modelos			
	AUC por Classe		
	Normal	Pneumonia	COVID-19
Adam_raw	0.9487	0.9438	0.9340
SGD_raw	0.9486	0.9418	0.9150
Adam_augmented	0.8227	0.7828	0.8473
SGD_augmented	0.9248	0.9202	0.9130

Podemos ver que os resultados com conjunto de dados *raw* foram muito similares para os dois otimizadores. Tanto para a classe "Normal" quanto "Pneumonia" a diferença de média de AUC foi menor que 1%. Para a classe de "COVID-19" foi de 2%, favorecendo o otimizador Adam. Assim, a melhor combinação no conjunto de dados puros é com o otimizador Adam.

Agora, com o conjunto de dados com AD, temos uma grande diferença nos resultados. Para todas as classes, os resultados com otimizador SGD foram superiores em pelo menos 7%. Ainda assim, esses resultados não foram capazes de alcançar o caso *Adam_raw*. Concluimos que em geral o Aumento de Dados **prejudicou** o treinamento.

A Tabela 4.9 mostra a comparação de cada combinação de treino. Podemos ordenar as melhores combinações vendo quantas vezes cada uma aparece em uma das matrizes triangulares (a matrizes triangulares superior e inferior são espelhadas).

Tabela 4.9: Tabela de comparação de AUCs dos modelos. Podemos ver que a melhor combinação de treino é *Adam_raw*, seguido de *SGD_raw*, *SGD_augmented* e por último *Adam_augmented*.

Comparação de AUCs				
Combinação de treino	<i>Adam_raw</i>	<i>Adam_augmented</i>	<i>SGD_raw</i>	<i>SGD_augmented</i>
<i>Adam_raw</i>	-	<i>Adam_raw</i>	<i>Adam_raw</i>	<i>Adam_raw</i>
<i>Adam_augmented</i>	<i>Adam_raw</i>	-	<i>SGD_raw</i>	<i>SGD_augmented</i>
<i>SGD_raw</i>	<i>Adam_raw</i>	<i>SGD_raw</i>	-	<i>SGD_raw</i>
<i>SGD_augmented</i>	<i>Adam_raw</i>	<i>SGD_augmented</i>	<i>SGD_raw</i>	-

Vemos que qualitativamente a melhor combinação de treino é o **Adam_raw** (com três aparições), seguido de *SGD_raw* (com duas aparições), *SGD_augmented* (1 aparição) e por último *Adam_augmented* (não ganha de nenhuma outra combinação).

O resultado de *Adam_augmented* como pior combinação pode ser facilmente visto pelas curvas ROC de seus *folders*.

4.5 Análise Quantitativa

Nas Subseções a seguir, vemos as Tabelas 4.10, 4.11, 4.12 e 4.13 com os resultados das métricas estatísticas dos sub-treinos **Adam_raw**, **Adam_augmented**, **SGD_raw** e **SGD_augmented**, respectivamente. Nas Figuras 4.14, 4.15, 4.16 e 4.17 encontram-se as Matrizes de Confusão de cada um dos *fold*s.

4.5.1 Conjunto de Dados *Raw*

Tabela 4.10: Tabela de resultados das métricas de validação cruzada do treino *Adam_raw*.

Métricas Estatísticas - Adam_raw					
	Acurácia Global	Classes	Métricas		
			<i>Precisão</i>	<i>Sensibilidade</i>	<i>F1-Score</i>
<i>fold 1</i>	0.8782	Normal	0.8875	0.9364	0.9113
		Pneumonia	0.8729	0.8355	0.8538
		COVID-19	0.6897	0.3883	0.4969
<i>fold 2</i>	0.8688	Normal	0.8566	0.9602	0.9055
		Pneumonia	0.9048	0.7812	0.8385
		COVID-19	0.6250	0.3398	0.4403
<i>fold 3</i>	0.8708	Normal	0.8950	0.9097	0.9023
		Pneumonia	0.8401	0.8553	0.8476
		COVID-19	0.7692	0.3883	0.5161
<i>fold 4</i>	0.8776	Normal	0.9031	0.9154	0.9092
		Pneumonia	0.8586	0.8586	0.8586
		COVID-19	0.5949	0.4563	0.5165
<i>fold 5</i>	0.8718	Normal	0.8902	0.9165	0.9032
		Pneumonia	0.8457	0.8520	0.8488
		COVID-19	0.8333	0.3398	0.4828
Média	0.8734	Normal	0.8864	0.9276	0.9062
		Pneumonia	0.8643	0.8365	0.8494
		COVID-19	0.7024	0.3825	0.4905

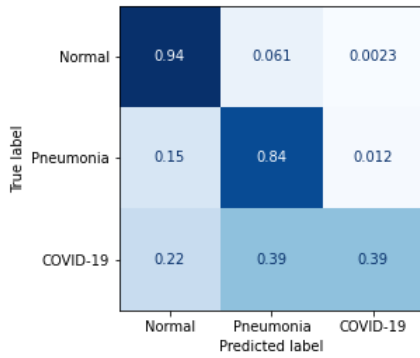
4.5.2 Conjunto de Dados com AD

Tabela 4.11: Tabela de resultados das métricas de validação cruzada do treino SGD_ raw.

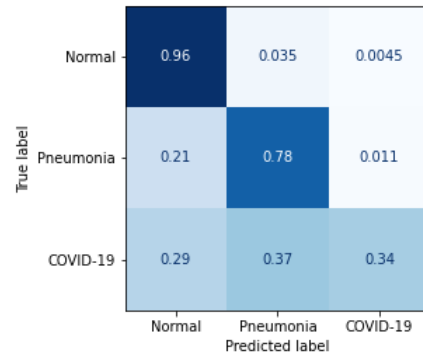
Métricas Estatísticas - SGD_ raw					
	Acurácia Global	Classes	Métricas		
			Precisão	Sensibilidade	F1-Score
<i>fold 1</i>	0.8617	Normal	0.9162	0.8688	0.8919
		Pneumonia	0.8001	0.8890	0.8422
		COVID-19	0.7288	0.4175	0.5309
<i>fold 2</i>	0.8744	Normal	0.9050	0.9029	0.9039
		Pneumonia	0.8392	0.8668	0.8528
		COVID-19	0.7313	0.4757	0.5765
<i>fold 3</i>	0.8734	Normal	0.8714	0.9392	0.9041
		Pneumonia	0.8827	0.8166	0.8484
		COVID-19	0.7544	0.4175	0.5365
<i>fold 4</i>	0.8727	Normal	0.9176	0.8847	0.9008
		Pneumonia	0.8222	0.8939	0.8566
		COVID-19	0.7167	0.4175	0.5276
<i>fold 5</i>	0.8805	Normal	0.8872	0.9381	0.9120
		Pneumonia	0.8788	0.8347	0.8562
		COVID-19	0.7143	0.4369	0.5422
Média	0.8725	Normal	0.8994	0.9067	0.9025
		Pneumonia	0.8446	0.8601	0.8512
		COVID-19	0.7290	0.4330	0.5429

Tabela 4.12: Tabela de resultados das métricas de validação cruzada do treino Adam_ augmented.

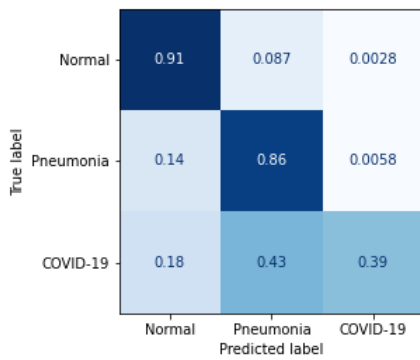
Métricas Estatísticas - Adam_ augmented					
	Acurácia Global	Classes	Métricas		
			Precisão	Sensibilidade	F1-Score
<i>fold 1</i>	0.2990	Normal	0.9459	0.2584	0.4059
		Pneumonia	0.7251	0.2993	0.4237
		COVID-19	0.0486	0.9903	0.0927
<i>fold 2</i>	0.8409	Normal	0.8191	0.9671	0.8870
		Pneumonia	0.8884	0.7072	0.7875
		COVID-19	0.8182	0.2621	0.3971
<i>fold 3</i>	0.6510	Normal	0.6262	0.9903	0.7673
		Pneumonia	0.8878	0.2146	0.3457
		COVID-19	0.0000	0.0000	0.0000
<i>fold 4</i>	0.8640	Normal	0.9310	0.8728	0.9009
		Pneumonia	0.8640	0.8618	0.8629
		COVID-19	0.3519	0.7379	0.4765
<i>fold 5</i>	0.3484	Normal	0.9495	0.3203	0.4790
		Pneumonia	0.8430	0.3355	0.4800
		COVID-19	0.0504	0.9806	0.0960
Média	0.6006	Normal	0.8543	0.6817	0.6880
		Pneumonia	0.8416	0.4837	0.5799
		COVID-19	0.2538	0.5941	0.2124



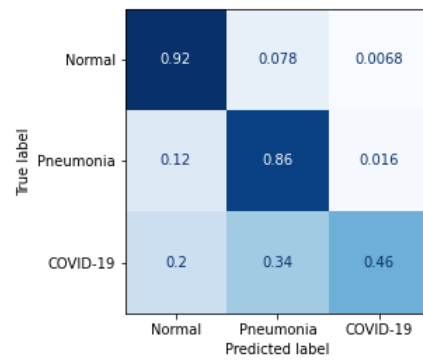
(a)



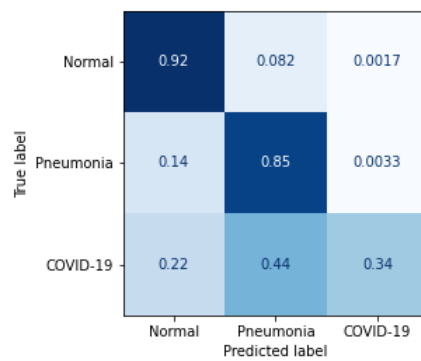
(b)



(c)

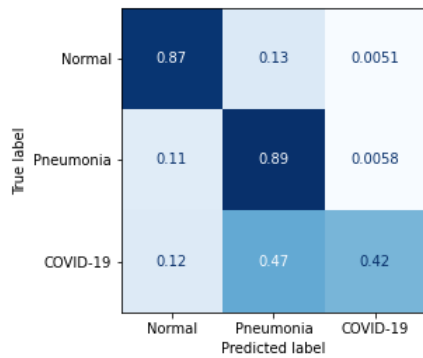


(d)

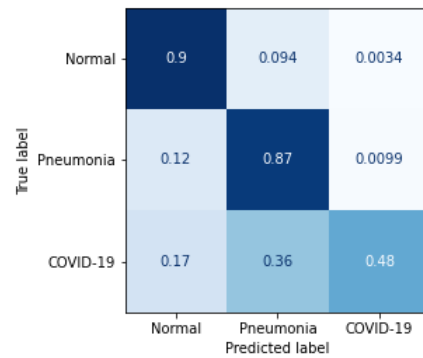


(e)

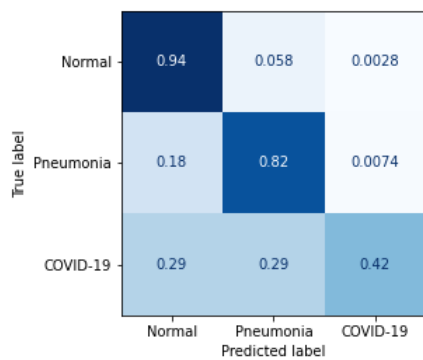
Figura 4.14: Matrizes de Confusão de cada um dos *fold*s treinados para **Adam_raw**. Quanto mais escuro, maior quantidade de dados, quanto mais claro, menor quantidade de dados. As figuras são: a) *fold* 1; b) *fold* 2; c) *fold* 3; d) *fold* 4; e) *fold* 5.



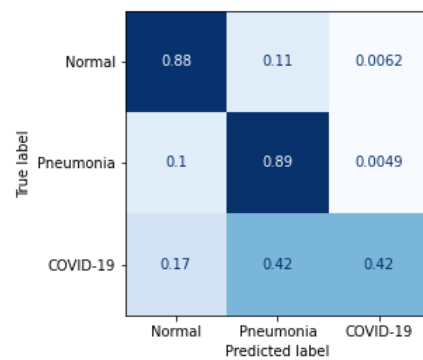
(a)



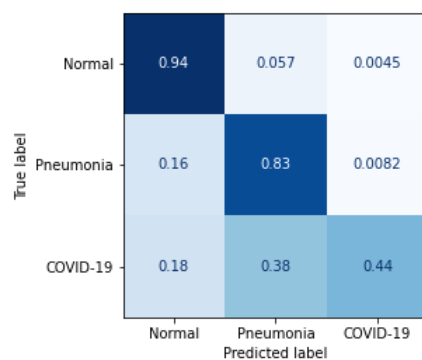
(b)



(c)

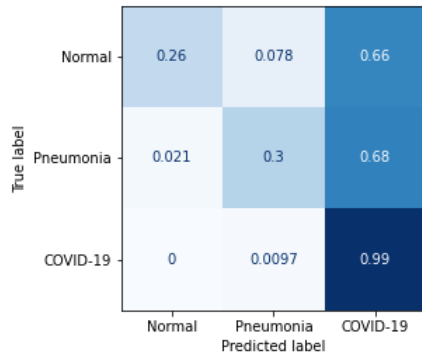


(d)

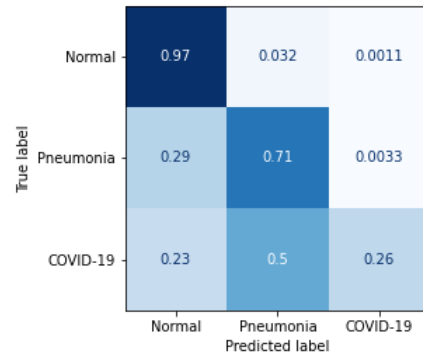


(e)

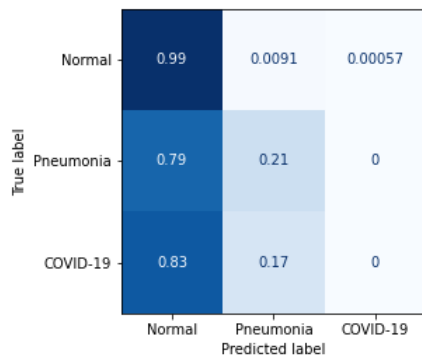
Figura 4.15: Matrizes de Confusão de cada um dos *fold*s treinados para **SGD_raw**. Quanto mais escuro, maior quantidade de dados, quanto mais claro, menor quantidade de dados. A As figuras são: a) *fold* 1; b) *fold* 2; c) *fold* 3; d) *fold* 4; e) *fold* 5.



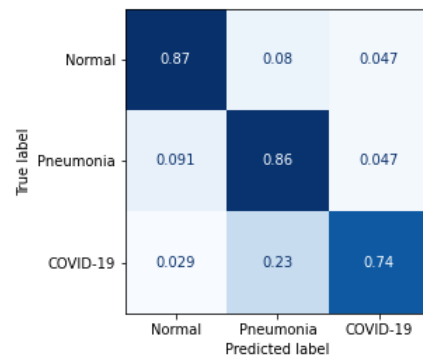
(a)



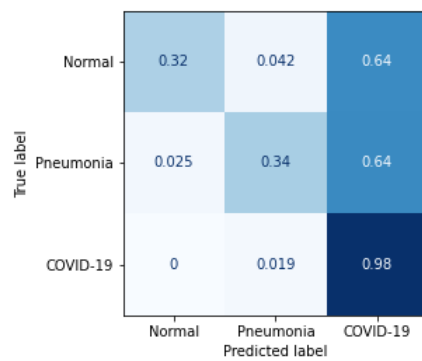
(b)



(c)

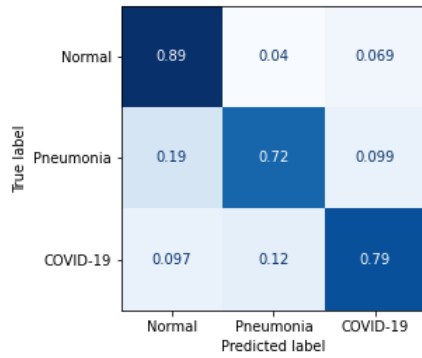


(d)

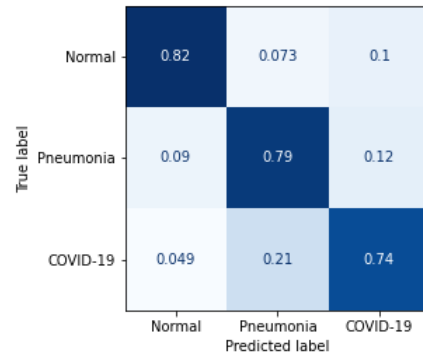


(e)

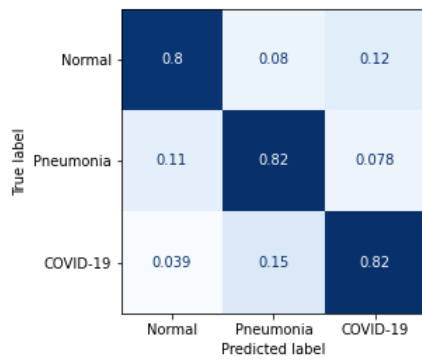
Figura 4.16: Matrizes de Confusão de cada um dos *fold*s treinados para **Adam_augmented**. Quanto mais escuro, maior quantidade de dados, quanto mais claro, menor quantidade de dados. As figuras são: a) *fold* 1; b) *fold* 2; c) *fold* 3; d) *fold* 4; e) *fold* 5.



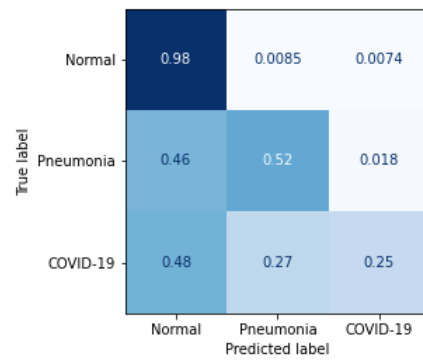
(a)



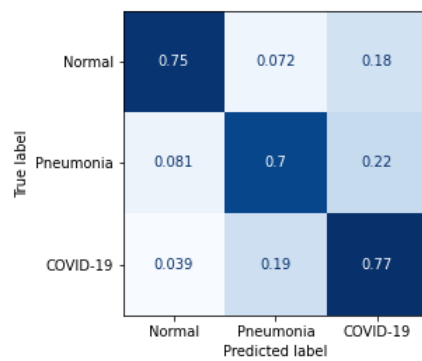
(b)



(c)



(d)



(e)

Figura 4.17: Matrizes de Confusão de cada um dos *fold*s treinados para **SGD_augmented**. Quanto mais escuro, maior quantidade de dados, quanto mais claro, menor quantidade de dados. A As figuras são: a) *fold* 1; b) *fold* 2; c) *fold* 3; d) *fold* 4; e) *fold* 5.

Tabela 4.13: Tabela de resultados das métricas de validação cruzada do treino SGD_augmented.

Métricas Estatísticas - SGD_augmented					
	Acurácia Global	Classes	Métricas		
			Precisão	Sensibilidade	F1-Score
<i>fold 1</i>	0.8182	Normal	0.8693	0.8919	0.8800
		Pneumonia	0.9139	0.7155	0.8026
		COVID-19	0.2508	0.7864	0.3803
<i>fold 2</i>	0.8075	Normal	0.9271	0.8228	0.8718
		Pneumonia	0.8643	0.7911	0.8261
		COVID-19	0.1881	0.7379	0.2998
<i>fold 3</i>	0.8065	Normal	0.9137	0.7995	0.8528
		Pneumonia	0.8649	0.8158	0.8396
		COVID-19	0.2143	0.8155	0.3394
<i>fold 4</i>	0.7760	Normal	0.7390	0.9841	0.8441
		Pneumonia	0.9362	0.5189	0.6677
		COVID-19	0.4262	0.2524	0.3171
<i>fold 5</i>	0.7341	Normal	0.9286	0.7530	0.8316
		Pneumonia	0.8543	0.7039	0.7719
		COVID-19	0.1215	0.7670	0.2098
Média	0.7884	Normal	0.8755	0.8500	0.8560
		Pneumonia	0.8867	0.7090	0.7815
		COVID-19	0.2401	0.6718	0.3092

4.5.3 Validação

Primeiramente, simplesmente vendo as Matrizes de Confusão da combinação Adam_augmented observamos que o Aumento de Dados prejudicou bastante o aprendizado do modelo. Vemos que 3 dos 5 *folds* treinados estão totalmente desbalanceados com um grande viés de classe: nos *folds* 1 e 5 a maioria das imagens são classificadas como "COVID-19" e no *fold* 3 como "Normal". Esse resultado é coerente com o observado nas curvas ROC da análise qualitativa.

Entre as combinações SGD_raw e Adam_raw, também confirmamos os resultados semelhantes observados na análise qualitativa. Observando as médias, agora vemos que eles possuem performance levemente melhor em cada caso. Adam_raw possui melhor Precisão para "Pneumonia", melhor Sensibilidade para "Normal" e melhor F1-Score para "Normal", enquanto em todas as outras métricas SGD_raw é melhor. Por isso, consideraremos SGD_raw como um modelo em média melhor, em desacordo com a análise qualitativa desses dois modelos.

Podemos ver também que essas duas combinações possuem resultados muito bons para classificação "Normal" e "Pneumonia", com poucos falsos positivos (principalmente nessa primeira), demonstrado pela Precisão. A confusão de classificação acontece principalmente em imagens de "COVID-19". Isso é demonstrado pelo valor baixo de Sensibilidade, abaixo de 50%, isto é, mais de 50% das imagens de "COVID-19" foram classificadas erroneamente.

Agora quando comparamos a combinação de SGD_augmented com os dois *raws*, temos uma mudança grande de comportamento: **claramente as Matrizes de Confusão estão mais balanceadas**. Para explicar esse fenômeno, devemos olhar a Precisão dos modelos.

O que acontece é que nos casos *raw* temos bons classificadores com alta Sensibilidade e Precisão

para as classes "Normal" e "Pneumonia" e baixa sensibilidade para "COVID-19". Isso significa que os modelos possuem relativa boa confiança de classificação quando uma imagem é não-"COVID-19", mas quando ela é "COVID-19", a confiança diminui confundindo com as outras classes - especialmente "Pneumonia". Esse comportamento indica altas quantidades de **falsos negativos**. No caso da combinação SGD_augmented, nós temos uma troca de priorização da classificação de "COVID-19": alta Sensibilidade com baixa Precisão. Isto é, imagens das classes não-"COVID-19" são classificadas erroneamente mais vezes como "COVID-19", ao mesmo tempo que as imagens que de fato são "COVID-19" são favorecidas. Esse comportamento indica altas quantidades de **falsos positivos**.

Para decidir qual modelo é melhor então, vamos utilizar a métrica *F1-Score*. Esta métrica leva em consideração tanto a Sensibilidade quanto a Precisão, por isso é uma boa forma de se avaliar o número de falsos negativos e falsos positivos simultaneamente, e por consequência, o melhor modelo geral.

A Tabela 4.15 a seguir compara os modelos com base em suas *F1-Score* média, de forma similar ao feito com os valores de AUC na análise qualitativa. Como temos três valores (um por classe) para cada combinação, foi tirado a média das três classes e o melhor modelo é o que possui a maior *F1-Score* média entre as classes. Esses valores podem ser encontrados na Tabela 4.14.

Tabela 4.14: Resumo dos valores de F1-Score médio entre as classes para cada modelo.

Comparação Quantitativa de Modelos	
F1-Score Médio	
Adam_raw	0.7487
SGD_raw	0.7655
Adam_augmented	0.4934
SGD_augmented	0.6489

Tabela 4.15: Tabela de comparação de F1-Scores dos modelos. Podemos ver que a melhor combinação de treino é SGD_raw, seguido de Adam_raw, SGD_augmented e por último Adam_augmented.

Comparação de F1-Scores				
Combinação de Treino	Adam_raw	Adam_augmented	SGD_raw	SGD_augmented
Adam_raw	-	Adam_raw	SGD_raw	Adam_raw
Adam_augmented	Adam_raw	-	SGD_raw	SGD_augmented
SGD_raw	SGD_raw	SGD_raw	-	SGD_raw
SGD_augmented	Adam_raw	SGD_augmented	SGD_raw	-

Vemos que quantitativamente a melhor combinação de treino é o **SGD_raw** (com três aparições), seguido de Adam_raw (com duas aparições), SGD_augmented (1 aparição) e por último Adam_augmented (não ganha de nenhuma outra combinação).

O resultado de Adam_augmented como pior combinação pode ser facilmente visto pelas Matrizes de Confusão de seus *folds* e está de acordo com o encontrado na análise qualitativa.

4.6 Comparação com outros Modelos e Abordagens

Como a comparação com os modelos apresentados na Tabela 4.16 é de natureza quantitativa, os valores apresentados na Tabela 4.16 de comparação a seguir são relativos ao modelo de combinação SGD_raw, que obteve os melhores resultados nessa análise. Os valores globais das métricas foram calculados como uma média simples entre os valores específicos das classes.

Vemos que o modelo AutoResCovidNet apresentado neste trabalho foi capaz de atingir resultados melhores de Precisão e Acurácia Global que os trabalhos de Misztal *et al.* [46] e de Precisão Global que o trabalho de Tulin *et al.* [42]. Em todos os outros casos os outros modelos da literatura foram superiores, ainda que alguns deles tenham utilizado de *undersampling* para balanceamento de seus dados.

Podemos ver ainda que a métrica mais prejudicial aos resultados é o valor de Sensibilidade de "COVID-19", isto é, o grande número de falsos negativos relativos a esta classe. Trataremos mais disso na Seção 4.7 a seguir.

Tabela 4.16: Tabela Comparativa do modelo AutoResCovidNet (em negrito) com outros da literatura apresentados na Seção 2.3. O se refere aos autores que utilizaram de *undersampling* para balanceamento de dados em seus projetos.

Comparação AutoResCovidNet com Outros Modelos da Literatura						
Estudo	Tipo de Imagem	Classes	Acurácia (%)	Precisão (%)	Sensibilidade (%)	F1-Score (%)
Wang <i>et al.</i> [41]	Raio-X	Global	93.3	-	-	-
		<i>Normal</i>	-	90.5	95	-
		<i>Pneumonia</i>	-	91.3	94	-
		<i>COVID-19</i>	-	98.9	91	-
Tulin <i>et al.</i> * [42]	Raio-X	Global	87.02	89.96	85.35	87.37
Khan <i>et al.</i> * [43]	Raio-X	Global	95	95	96.9	95.6
Chowdbury <i>et al.</i> * [44]	Raio-X	Global	97.94	97.95	97.94	97.94
Gunraj <i>et al.</i> [45]	TC	Global	99.1	-	-	-
		<i>Normal</i>	-	99.4	100	-
		<i>Pneumonia</i>	-	98.4	99	-
		<i>COVID-19</i>	-	99.7	97.3	-
Misztal <i>et al.</i> [46]	Raio-X	Global	58	74	100	85
	TC	Global	73	82	83	82
Li <i>et al.</i> * [47]	TC	Global	-	-	-	-
		<i>Normal</i>	-	-	94	-
		<i>Pneumonia</i>	-	-	87	-
		<i>COVID-19</i>	-	-	90	-
AutoResCovidNet	Raio-X	Global	87.25	82.43	73.32	76.55
		<i>Normal</i>	-	89.94	90.67	90.25
		<i>Pneumonia</i>	-	84.46	86.01	85.12
		<i>COVID-19</i>	-	72.90	43.30	54.29

4.7 Discussão

Observando os resultados da Seção 4.6 com as tabelas dos modelos testados, vemos que em média as combinações *Adam_raw* e *SGD_raw* competem como melhor modelo. A fim de manter consistência com a comparação feita com os modelos da literatura vamos considerar que **o vencedor é o SGD_raw**, sem grandes desconsiderações do resultado qualitativo (diferença de 2% de AUC). Obtemos a Tabela 4.17 a seguir com a ordem dos melhores modelos:

Tabela 4.17: Tabela ordenada com as melhores combinações do modelo AutoResCovidNet. Em negrito a melhor combinação.

AutoResCovidNet - Melhores Combinações	
SGD_raw	Melhor modelo
Adam_raw	2º melhor modelo
SGD_augmented	3º melhor modelo
Adam_augmented	Pior modelo

Apesar do valor de *F1-Score* apontar claramente que ambos modelos *raw* são melhores que os modelos de AD, os resultados da Matriz de Confusão no caso *SGD_augmented* (Figura 4.17) pode nos fornecer um panorama diferente. De fato o Aumento de Dados para o otimizador Adam foi bastante prejudicial, mas para o SGD serviu como um bom balanceador, fazendo com que o modelo acertasse mais entradas de "COVID-19" em troca de mais falsos positivos. Este resultado pode

indicar que há um espaço para melhora nas operações de AD, adicionando novas transformações e/ou fazendo ajuste fino em suas variáveis.

Ainda assim, podemos concluir que o AD foi prejudicial para os experimentos. Isso pode ser explicado por operações excessivamente agressivas que ao invés de ajudar o conjunto de dados a ser mais representativo, fizeram com que ele perdesse identidade, sendo difícil o aprendizado ser generalizado. Essa hipótese pode ser verificada ao analisarmos a grande oscilação nos gráficos de *loss* e acurácia dos modelos *Adam_augmented* e *SGD_augmented* nas Figuras 4.8 e 4.9, respectivamente, indicando que o otimizador não foi capaz de achar mínimos locais para classificação.

Sobre a estrutura do *AutoResCovidNet*, podemos ver que o melhor modelo obtido pelo *AutoKeras* não possui tantos parâmetros a mais que a *ResNet50V2* original, 23.570.947 contra 23.564.800 parâmetros. Pode ser decepcionante pensar que após 20 modelos testados apenas 6.147 parâmetros tenham sido adicionados mas deve-se considerar que modelos **mais profundos** foram testados e não foram selecionados porque não foram capazes de obter melhores resultados. Isso chama atenção para um aspecto importante do Aprendizado Profundo: maior profundidade não implica, necessariamente, em melhor performance. Podemos ver isso comparando *DarkNet-19* de Tulin *et al.* [42] com a *COVID-Net* de Wang *et al.* [41] Concluimos então, juntamente com as comparações feitas na Seção 4.6, que o número de camadas da *ResNet50V2* é suficiente para uma classificação satisfatória.

Assumindo então que o número de camadas não é o responsável pelos resultados inferiores aos modelos da literatura, podemos analisar 4 hipóteses principais que podem explicar a discrepância dos resultados: 1) desbalanceamento de dados ; 2) microestruturas ineficientes da rede; 3) rotina de treinamento simplista; 4) baixa qualidade dos dados.

Primeiramente, quatro dos sete trabalhos comparados [42], [43], [44], [47] utilizaram de *undersampling* para balanceamento dos dados. Foi optado não utilizar dessa técnica pois foi entendido que seu uso minimiza e ignora dificuldades reais relevantes na aplicação de Aprendizado Profundo em problemas novos - como uma nova pandemia mundial. Certamente o modelo obtido neste trabalho teria sua performance melhorada se essa técnica tivesse sido utilizada.

Sobre a segunda hipótese, quantidade de parâmetros brutos influencia na generalização de características mas, como Aprendizado Profundo se baseia no princípio de empilhamento e agrupamento de características, a forma como as abstrações de conhecimento são organizadas influencia diretamente no resultado final. Um exemplo disto é a própria arquitetura residual das *ResNets* que modifica o comportamento total de aprendizagem com uma simples adição de um *bypass*, sem adição de novos parâmetros. Essa hipótese abre espaço para possíveis modificações na estrutura principal do modelo que permitam o aprendizado de características mais sutis e melhor generalização.

Para a terceira hipótese, pode ser visto dos artigos *Universal Language Model Fine-tuning for Text Classification* [50] e *Cyclical Learning Rates for Training Neural Networks* [51] que técnicas de escolha e agendamento da taxa de aprendizado e outras configurações de otimização podem influenciar bastante no processo de treinamento. Para treino da *AutoResCovidNet*, foram testados apenas dois otimizadores, *Adam* e *SGD*, com configurações padrões da biblioteca. Podemos pensar

então que o ajuste fino das variáveis desses algoritmos nos permita achar melhores mínimos locais que resolvam melhor a classificação.

E a quarta e última hipótese, sobre os dados apresentados ao modelo, podemos descartar que eles originalmente não são suficientes, pois Wang *et al.* [41] utilizaram-se deles para atingir os seus resultados superiores com a COVID-Net. Assumimos então que um possível problema esteja no pré-processamento e preparação desses dados. Como já dito, o Aumento de Dados utilizado não foi eficiente e até piorou o comportamento do modelo. Um melhor projeto da etapa de preparo de dados, juntamente com mais transformações e um filtro de Região de Interesse para segmentação da região do pulmão são soluções que podem melhorar bastante a performance.

Capítulo 5

Conclusões

Este trabalho apresenta todo o projeto de uma solução de Aprendizado Profundo facilitado por técnicas de *AutoML* e aborda todas as etapas de seu desenvolvimento, desde a obtenção e preparação dos dados até a avaliação final do modelo.

Mesmo que não tenha sido possível superar os melhores resultados da literatura, o modelo apresenta ainda *F1-Score* médio acima de 75% com acurácia de 87.25%, nível comparável aos modelos estado-da-arte. Vale ressaltar também que, dos três trabalhos que não utilizaram de *undersampling*, um deles foi alcançado na maioria das métricas [46] e os outros dois, realizados pela mesma equipe COVID-Net [41], [45], são as referências mundiais que inspiraram este trabalho e não existia a expectativa de superá-los.

Ainda assim, o insucesso do Aumento de Dados para treinamento da rede foi um grande limitador para obtenção de melhores resultados. O pré-processamento do conjunto de dados não foi suficiente para compensar o desbalanceamento e sutilezas da classe de "COVID-19". Nesses casos, raramente um conjunto de dados brutos é capaz de prover toda representação das características desejadas em todas classes. Isso fica claro quando analisamos os resultados das Matrizes de Confusão dos modelos *raw*. Normalmente, o Aumento de Dados e preparação do conjunto base são as imediatas tentativas de melhoria de performance, mas o seu impacto sendo prejudicial, tornou-se uma tarefa árdua o aprendizado das características de interesse.

Apesar disso, foi visto como AutoML é um grande passo na busca e no desenvolvimento de modelos de Redes Neurais, agilizando o desenvolvimento e permitindo enfoque maior em estudos mais localizados e específicos com resultados bastante satisfatórios. Vemos isso com o AutoKeras utilizado neste trabalho. Tendo sido testado 20 modelos, este foi um grande facilitador de projeto, ficando claro que um grande período de tempo foi reduzido eliminando tentativas de adição de camadas com poucas melhorias. Dessa forma, fica evidenciado os fatores a serem melhorados que podem causar grandes impactos no desempenho – neste caso, preparação de dados e rotina de treinos mais elaborada.

Por fim, historicamente, avanços em Redes Neurais Convolucionais dependem de disponibilidade de tempo, alta capacidade de processamento e grande número de iterações. Sendo todas essas variáveis limitadas neste trabalho e com os resultados apresentados, concluímos que os objetivos

foram alcançados com um modelo de Rede Neural Convolutacional capaz de auxiliar a comunidade de profissionais da saúde na análise de imagens de Raio-X de pacientes com suspeita de "COVID-19", mesmo que seu diagnóstico direto não seja possível.

5.1 Trabalhos Futuros

O insucesso do Aumento de Dados em melhorar a generalização do modelo indica que essa é a principal etapa a ser aperfeiçoada. Mais especificamente, podemos: 1) diminuir o ângulo máximo de rotação da imagem; 2) acrescentar transformações afins (como escala, translação, cisalhamento) e/ou perspectivas; 3) acrescentar processamentos de brilho e contraste; e 4) adicionar um filtro de recorte de Região de Interesse na região do pulmão. Se essas operações não forem suficientes para melhoria de desempenho (especialmente diminuição de falsos negativos), ainda pode-se recorrer a uma estratégia de rotina de treino mais elaborada, como, por exemplo, taxa de aprendizado variável.

Para estudos futuro, vale utilizar ainda de *undersampling* para balanceamento do conjunto de dados e verificar a hipótese de que seus resultados realmente seriam aprimorados.

Uma última tentativa de melhoria que exigiria mais tempo estudo, seriam transformações e modificações de microestruturas da rede. Poderiam ser adicionados novos caminhos de processamento na arquitetura e unidades de operações mais específicas, capazes de extrair características mais granulares das entradas.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.
- [2] HE, K. et al. *Deep Residual Learning for Image Recognition*. 2015.
- [3] HE, X.; ZHAO, K.; CHU, X. Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, Elsevier BV, v. 212, p. 106622, Jan 2021. ISSN 0950-7051. Disponível em: <<http://dx.doi.org/10.1016/j.knosys.2020.106622>>.
- [4] JIN, H.; SONG, Q.; HU, X. Auto-keras: An efficient neural architecture search system. In: ACM. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. [S.l.], 2019. p. 1946–1956.
- [5] FAWCETT, T. An introduction to roc analysis. *Pattern recognition letters*, Elsevier, v. 27, n. 8, p. 861–874, 2006.
- [6] COHEN, J. P.; MORRISON, P.; DAO, L. *COVID-19 Image Data Collection*. 2020.
- [7] WANG ALEXANDER WONG, Z. Q. L. P. M. A. C. H. G. J. L. M. R. B. V. A. E. K.-A. G. A. A.-H. L. *Figure 1 COVID-19 Chest X-ray Dataset Initiative*. may 2020. Disponível em: <<https://github.com/agchung/Figure1-COVID-chestxray-dataset#figure-1-covid-19-chest-x-ray-dataset-initiative>>.
- [8] WANG ALEXANDER WONG, Z. Q. L. P. M. A. C. H. G. J. L. M. R. B. V. A. E. K.-A. G. A. A.-H. L. *Actualmed COVID-19 Chest X-ray Dataset Initiative*. may 2020. Disponível em: <<https://github.com/agchung/Actualmed-COVID-chestxray-dataset>>.
- [9] CHOWDHURY TAWSIFUR RAHMAN, A. K. R. M. M. A. K. Z. B. M. K. R. I. M. S. K.-A. I. N. A.-E. M. B. I. R. M. E. H. *COVID-19 Chest X-ray Database*. apr 2020. Disponível em: <<https://www.kaggle.com/tawsifurrahman/covid19-radiography-database>>.
- [10] AMOROSA VERONICA ARTEAGA, M. G.-A. R. G. M. G. S. H. J. J. A. L. P. S. D. V. C. W. K. Y. T. C. S. H. M. K. L. M. P. A. S. G. S. K. P. A. F. H. C. B. E. A. F. J. K.-C. A. S. P. C. J. M. W. C. J. *COVID-19 Chest X-ray Database*. aug 2018. Disponível em: <<https://www.kaggle.com/tawsifurrahman/covid19-radiography-database>>.

- [11] ORGANIZATION, W. H. *Origin of SARS-CoV-2*. mar 2020. Disponível em: <[https://apps.who.int/iris/bitstream/handle/10665/332197/WHO-2019-nCoV-FAQ-Virus_{origin} - 2020.1 - eng.pdf](https://apps.who.int/iris/bitstream/handle/10665/332197/WHO-2019-nCoV-FAQ-Virus_origin-2020.1-eng.pdf)>.
- [12] ORGANIZATION, W. H. *WHO Coronavirus (COVID-19) Dashboard*. may 2021. Disponível em: <<https://covid19.who.int>>.
- [13] GRUBAUGH, N. D. et al. Public health actions to control new SARS-CoV-2 variants. *Cell*, Elsevier BV, v. 184, n. 5, p. 1127–1132, mar. 2021. Disponível em: <<https://doi.org/10.1016/j.cell.2021.01.044>>.
- [14] MORENS, D. M.; FAUCI, A. S. Emerging pandemic diseases: How we got to COVID-19. *Cell*, Elsevier BV, v. 182, n. 5, p. 1077–1092, set. 2020. Disponível em: <<https://doi.org/10.1016/j.cell.2020.08.021>>.
- [15] VANDENBERG, O. et al. Considerations for diagnostic COVID-19 tests. *Nature Reviews Microbiology*, Springer Science and Business Media LLC, v. 19, n. 3, p. 171–183, out. 2020. Disponível em: <<https://doi.org/10.1038/s41579-020-00461-z>>.
- [16] KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, v. 25, p. 1097–1105, 2012.
- [17] LOPEZ, M. M.; KALITA, J. *Deep Learning applied to NLP*. 2017.
- [18] KARPATHY, A. et al. Large-scale video classification with convolutional neural networks. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2014. p. 1725–1732.
- [19] KHAN, S. et al. A novel deep learning based framework for the detection and classification of breast cancer using transfer learning. *Pattern Recognition Letters*, v. 125, p. 1–6, 2019. ISSN 0167-8655. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0167865519301059>>.
- [20] HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. second. [S.l.]: Pearson Education, 1999.
- [21] MOR-YOSEF, S. et al. Ranking the risk factors for cesarean: logistic regression analysis of a nationwide study. *Obstetrics and gynecology*, v. 75, n. 6, p. 944–947, June 1990. ISSN 0029-7844. Disponível em: <<http://europepmc.org/abstract/MED/2342742>>.
- [22] CIREŞAN, D. et al. Multi-column deep neural network for traffic sign classification. *Neural Networks*, v. 32, p. 333–338, 2012. ISSN 0893-6080. Selected Papers from IJCNN 2011. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0893608012000524>>.
- [23] Feng Ning et al. Toward automatic phenotyping of developing embryos from videos. *IEEE Transactions on Image Processing*, v. 14, n. 9, p. 1360–1371, 2005.

- [24] TURAGA, S. C. et al. Convolutional networks can learn to generate affinity graphs for image segmentation. *Neural Computation*, MIT Press - Journals, v. 22, n. 2, p. 511–538, fev. 2010. Disponível em: <<https://doi.org/10.1162/neco.2009.10-08-881>>.
- [25] Garcia, C.; Delakis, M. Convolutional face finder: a neural architecture for fast and robust face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 26, n. 11, p. 1408–1423, 2004.
- [26] RUSSAKOVSKY, O. et al. ImageNet large scale visual recognition challenge. v. 115, n. 3, p. 211–252. ISSN 1573-1405. Disponível em: <<https://doi.org/10.1007/s11263-015-0816-y>>.
- [27] LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *nature*, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015.
- [28] BENGIO, Y.; SIMARD, P.; FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, v. 5, p. 157–66, 02 1994.
- [29] GLOROT, X.; BENGIO, Y. Understanding the difficulty of training deep feedforward neural networks. *Journal of Machine Learning Research - Proceedings Track*, v. 9, p. 249–256, 01 2010.
- [30] LECUN, Y. et al. Efficient backprop. 08 2000.
- [31] HE, K. et al. *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. 2015.
- [32] HE, K.; SUN, J. *Convolutional Neural Networks at Constrained Time Cost*. 2014.
- [33] SRIVASTAVA, R. K.; GREFF, K.; SCHMIDHUBER, J. *Highway Networks*. 2015.
- [34] FAROOQ, M.; HAFEEZ, A. *COVID-ResNet: A Deep Learning Framework for Screening of COVID19 from Radiographs*. 2020.
- [35] Lu, Z. et al. Multi-object detection method based on yolo and resnet hybrid networks. In: *2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM)*. [S.l.: s.n.], 2019. p. 827–832.
- [36] Chapala, H.; Sujatha, B. Resnet: Detection of invasive ductal carcinoma in breast histopathology images using deep learning. In: *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*. [S.l.: s.n.], 2020. p. 60–67.
- [37] ELSHAWI, R.; MAHER, M.; SAKR, S. *Automated Machine Learning: State-of-The-Art and Open Challenges*. 2019.
- [38] JIN, H.; SONG, Q.; HU, X. *Auto-Keras: An Efficient Neural Architecture Search System*. 2019.
- [39] METZ, C. E. Basic principles of roc analysis. In: ELSEVIER. *Seminars in nuclear medicine*. [S.l.], 1978. v. 8, n. 4, p. 283–298.

- [40] POWERS, D. M. W. *Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation*. 2020.
- [41] WANG, L.; LIN, Z. Q.; WONG, A. COVID-net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest x-ray images. v. 10, n. 1, p. 19549. ISSN 2045-2322. Disponível em: <<https://doi.org/10.1038/s41598-020-76550-z>>.
- [42] OZTURK, T. et al. Automated detection of covid-19 cases using deep neural networks with x-ray images. *Computers in Biology and Medicine*, v. 121, p. 103792, 2020. ISSN 0010-4825. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0010482520301621>>.
- [43] KHAN, A. I.; SHAH, J. L.; BHAT, M. M. CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images. *Computer Methods and Programs in Biomedicine*, Elsevier BV, v. 196, p. 105581, nov. 2020. Disponível em: <<https://doi.org/10.1016/j.cmpb.2020.105581>>.
- [44] CHOWDHURY, M. E. H. et al. Can ai help in screening viral and covid-19 pneumonia? *IEEE Access*, Institute of Electrical and Electronics Engineers (IEEE), v. 8, p. 132665–132676, 2020. ISSN 2169-3536. Disponível em: <<http://dx.doi.org/10.1109/ACCESS.2020.3010287>>.
- [45] GUNRAJ, H.; WANG, L.; WONG, A. Covidnet-ct: A tailored deep convolutional neural network design for detection of covid-19 cases from chest ct images. *Frontiers in Medicine*, v. 7, p. 1025, 2020. ISSN 2296-858X. Disponível em: <<https://www.frontiersin.org/article/10.3389/fmed.2020.608525>>.
- [46] MISZTAL, K. et al. The importance of standardisation – COVID-19 CT & radiograph image data stock for deep learning purpose. *Computers in Biology and Medicine*, Elsevier BV, v. 127, p. 104092, dez. 2020. Disponível em: <<https://doi.org/10.1016/j.combiomed.2020.104092>>.
- [47] LI, L. et al. Using artificial intelligence to detect COVID-19 and community-acquired pneumonia based on pulmonary CT: Evaluation of the diagnostic accuracy. *Radiology*, Radiological Society of North America (RSNA), v. 296, n. 2, p. E65–E71, ago. 2020. Disponível em: <<https://doi.org/10.1148/radiol.2020200905>>.
- [48] REDMON, J.; FARHADI, A. *YOLO9000: Better, Faster, Stronger*. 2016.
- [49] MAAS, A. L.; HANNUN, A. Y.; NG, A. Y. Rectifier nonlinearities improve neural network acoustic models. In: CITESEER. *Proc. icml*. [S.l.], 2013. v. 30, n. 1, p. 3.
- [50] HOWARD, J.; RUDER, S. *Universal Language Model Fine-tuning for Text Classification*. 2018.
- [51] SMITH, L. N. *Cyclical Learning Rates for Training Neural Networks*. 2017.
- [52] CHOLLET, F. *Xception: Deep Learning with Depthwise Separable Convolutions*. 2017.
- [53] HEMDAN, E. E.-D.; SHOUMAN, M. A.; KARAR, M. E. *COVIDX-Net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-Ray Images*. 2020.

- [54] SIMONYAN, K.; ZISSERMAN, A. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015.
- [55] HUANG, G. et al. *Densely Connected Convolutional Networks*. 2018.
- [56] SZEGEDY, C. et al. *Rethinking the Inception Architecture for Computer Vision*. 2015.
- [57] HE, K. et al. *Identity Mappings in Deep Residual Networks*. 2016.
- [58] SZEGEDY, C. et al. *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*. 2016.
- [59] HOWARD, A. G. et al. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. 2017.
- [60] IANDOLA, F. N. et al. *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size*. 2016.
- [61] RAJPURKAR, P. et al. *CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning*. 2017.
- [62] NARIN, A.; KAYA, C.; PAMUK, Z. *Automatic Detection of Coronavirus Disease (COVID-19) Using X-ray Images and Deep Convolutional Neural Networks*. 2020.
- [63] Wang Y. Peng, L. L. Z. L. M. B. R. M. S. X. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2017. p. 3462–3471.
- [64] KERMANY, D. *Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification*. Mendeley, 2018. Disponível em: <<https://data.mendeley.com/datasets/rscbjbr9sj/2>>.
- [65] YANG, X. et al. *COVID-CT-Dataset: A CT Scan Dataset about COVID-19*. 2020.
- [66] SETHY, P. K.; BEHERA, S. K. Detection of coronavirus disease (COVID-19) based on deep features. MDPI AG, mar. 2020. Disponível em: <<https://doi.org/10.20944/preprints202003.0300.v1>>.
- [67] EVGENIOU, T.; PONTIL, M. Support vector machines: Theory and applications. In: *Machine Learning and Its Applications*. Springer Berlin Heidelberg, 2001. p. 249–257. Disponível em: <https://doi.org/10.1007/3-540-44673-7_12>.
- [68] ZHANG, J. et al. *Viral Pneumonia Screening on Chest X-ray Images Using Confidence-Aware Anomaly Detection*. 2020.
- [69] TAN, M.; LE, Q. V. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. 2020.

- [70] AFSHAR, P. et al. Covid-caps: A capsule network-based framework for identification of covid-19 cases from x-ray images. *Pattern Recognition Letters*, v. 138, p. 638 – 643, 2020. ISSN 0167-8655. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0167865520303512>>.
- [71] HINTON, G. E.; SABOUR, S.; FROSST, N. Matrix capsules with em routing. In: *International conference on learning representations*. [S.l.: s.n.], 2018.
- [72] RONNEBERGER, O.; FISCHER, P.; BROX, T. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015.
- [73] HE, K. et al. *Momentum Contrast for Unsupervised Visual Representation Learning*. 2020.
- [74] YING, S. et al. Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images. Cold Spring Harbor Laboratory, fev. 2020. Disponível em: <<https://doi.org/10.1101/2020.02.23.20026930>>.
- [75] WANG, H. et al. *Deep Neural Network Based Relation Extraction: An Overview*. 2021.
- [76] LIN, T.-Y. et al. *Feature Pyramid Networks for Object Detection*. 2017.
- [77] ZHENG, C. et al. Deep learning-based detection for COVID-19 from chest CT using weak label. Cold Spring Harbor Laboratory, mar. 2020. Disponível em: <<https://doi.org/10.1101/2020.03.12.20027185>>.
- [78] WANG, S. et al. A deep learning algorithm using CT images to screen for corona virus disease (COVID-19). *European Radiology*, Springer Science and Business Media LLC, fev. 2021. Disponível em: <<https://doi.org/10.1007/s00330-021-07715-1>>.
- [79] KINGMA, D. P.; BA, J. *Adam: A Method for Stochastic Optimization*. 2017.
- [80] RUDER, S. *An overview of gradient descent optimization algorithms*. 2017.

ANEXOS

I.1 Imagens de Treinamentos

I.1.1 Busca de Modelos no Autokeras

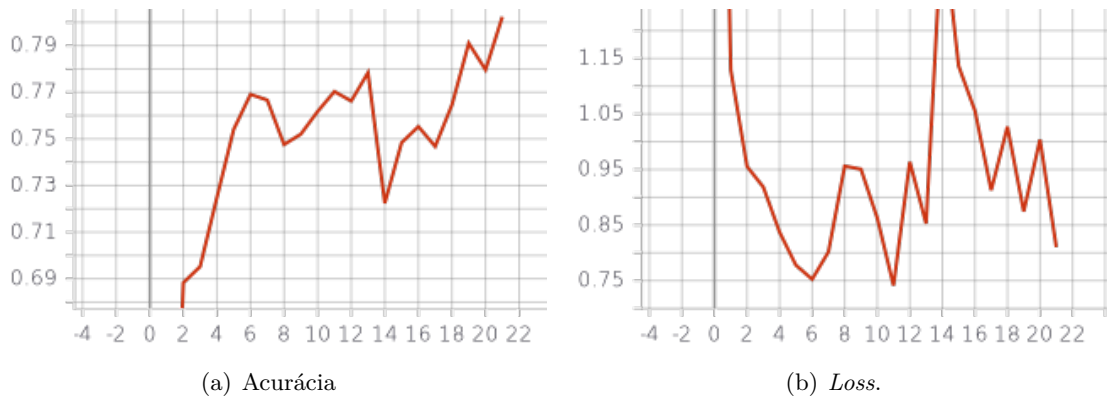


Figura 1: Gráficos de acurácia e *loss* do modelo 1 testado pelo Autokeras.

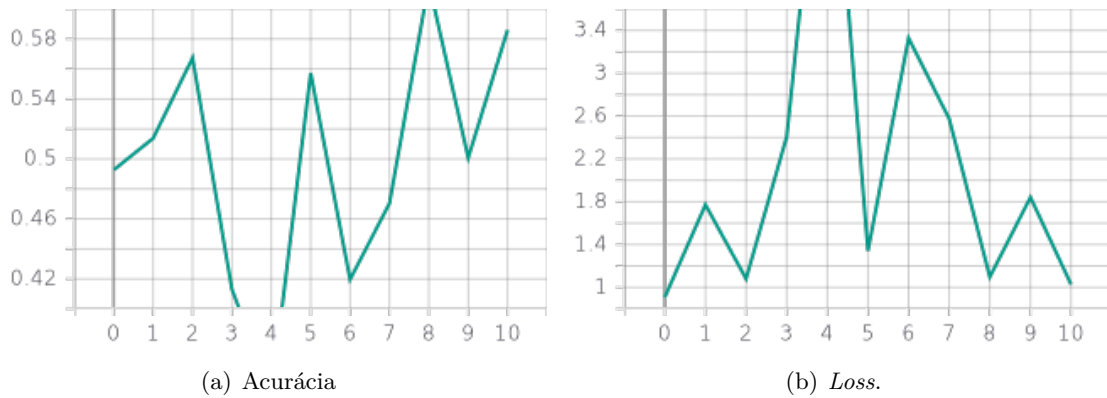


Figura 2: Gráficos de acurácia e *loss* do modelo 2 testado pelo Autokeras.

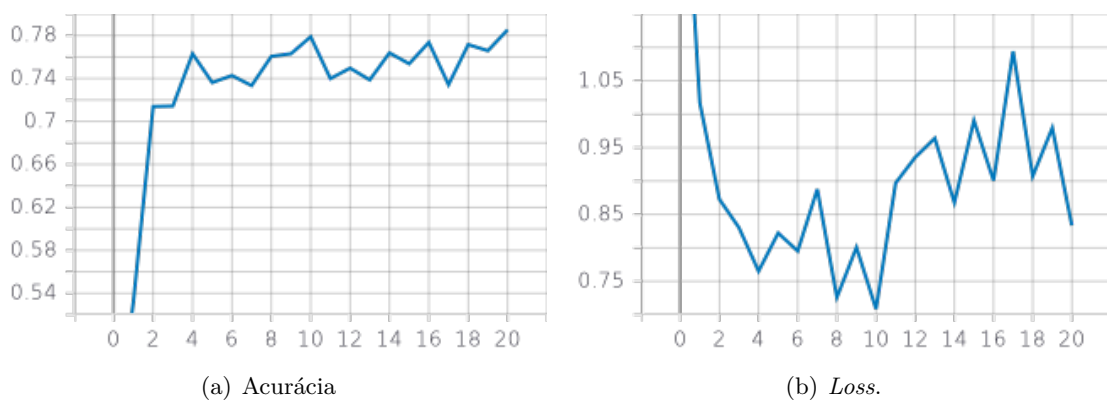


Figura 3: Gráficos de acurácia e *loss* do modelo 3 testado pelo Autokeras.

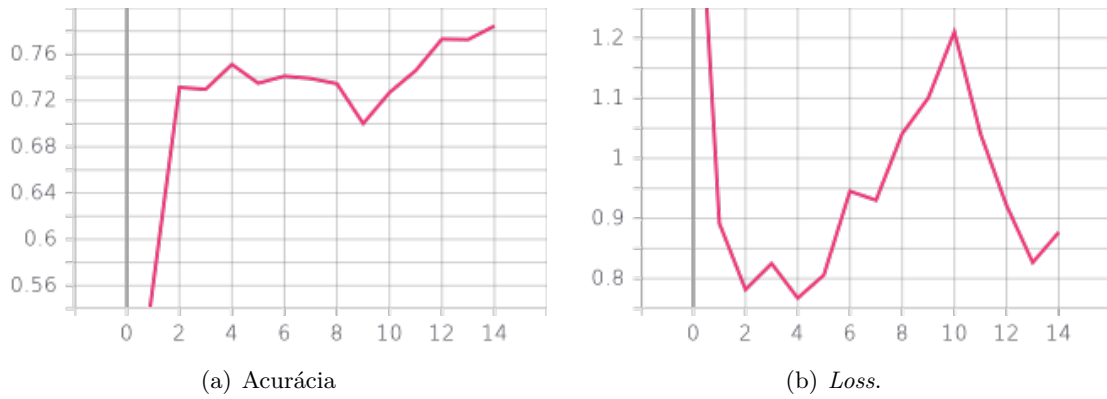


Figura 4: Gráficos de acurácia e *loss* do modelo 4 testado pelo Autokeras.

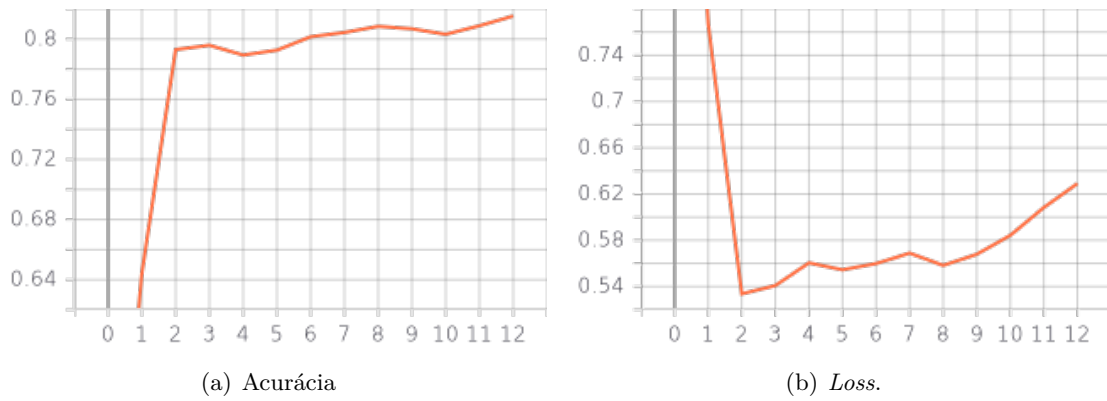


Figura 5: Gráficos de acurácia e *loss* do modelo 5 testado pelo Autokeras.

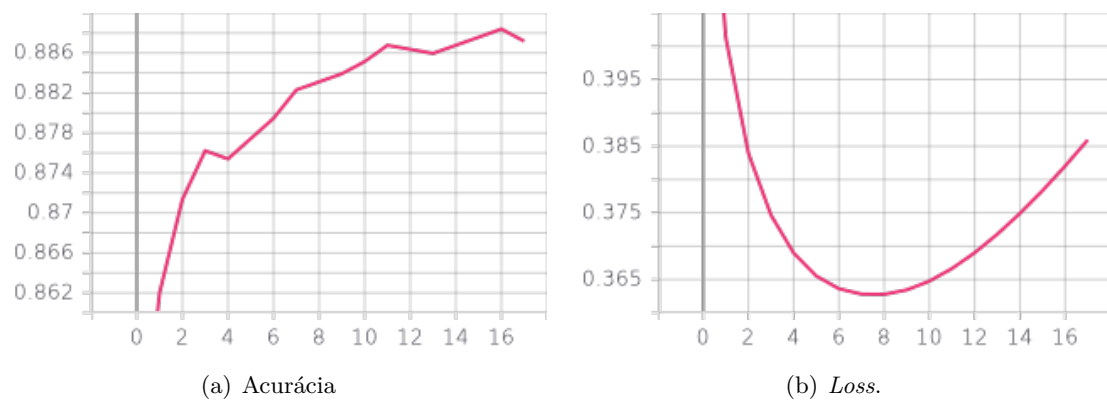


Figura 6: Gráficos de acurácia e *loss* do modelo 6 testado pelo Autokeras.

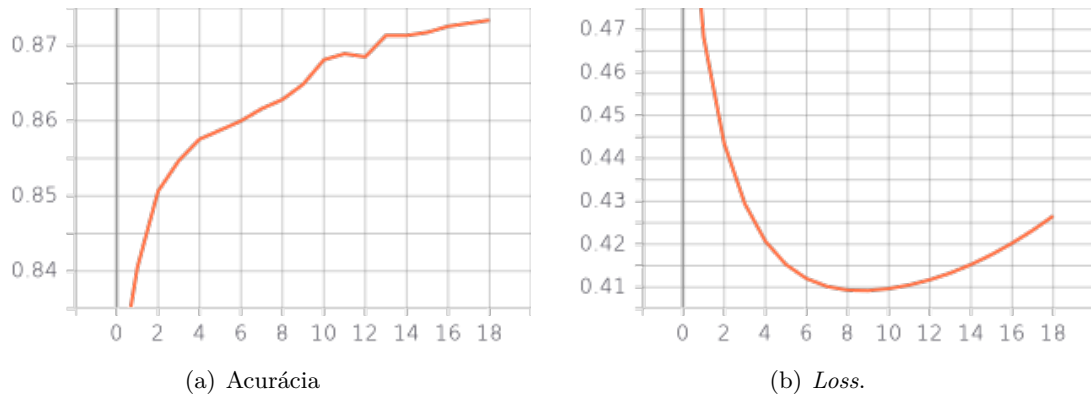


Figura 7: Gráficos de acurácia e *loss* do modelo 7 testado pelo Autokeras.

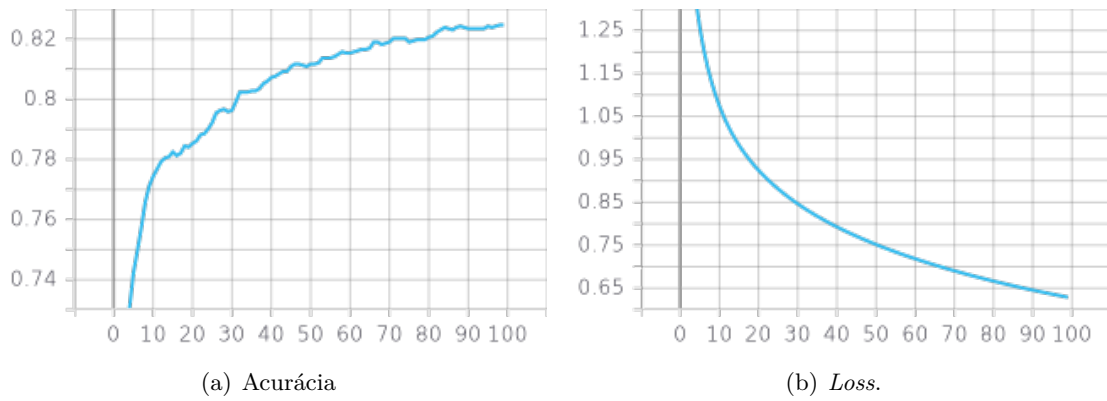


Figura 8: Gráficos de acurácia e *loss* do modelo 8 testado pelo Autokeras.

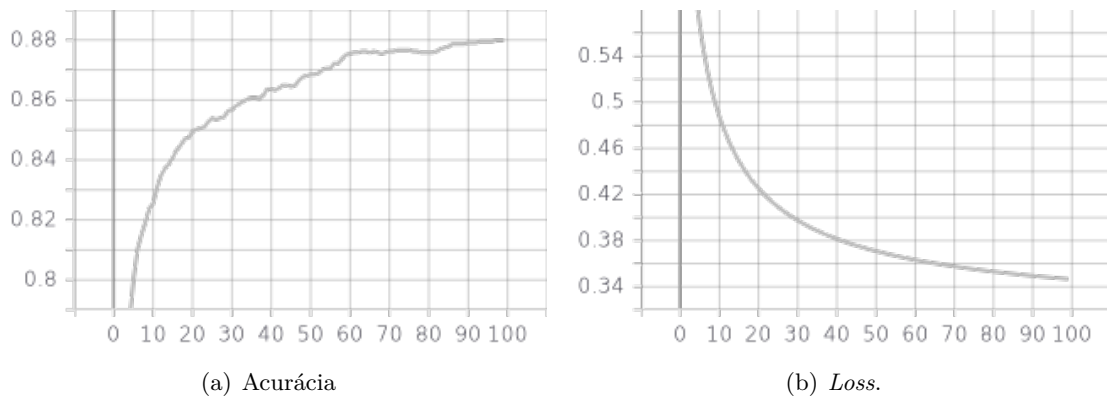


Figura 9: Gráficos de acurácia e *loss* do modelo 9 testado pelo Autokeras.

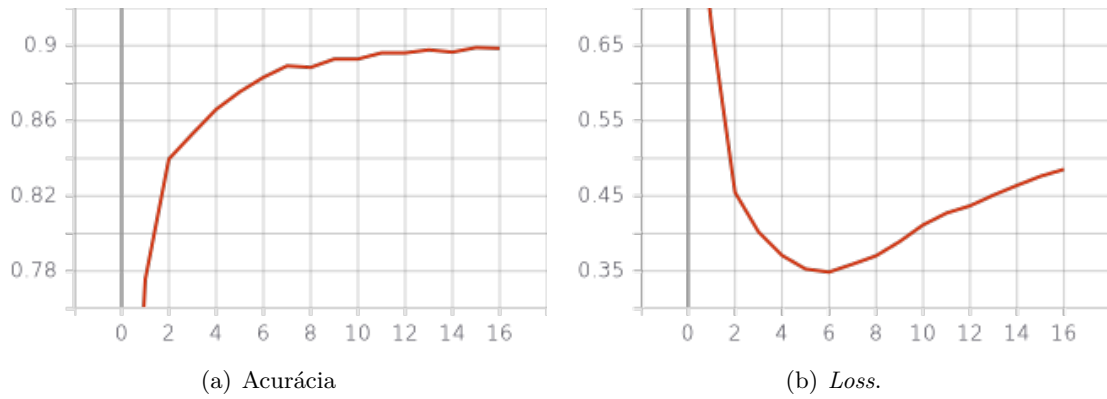


Figura 10: Gráficos de acurácia e *loss* do modelo 10 testado pelo Autokeras.

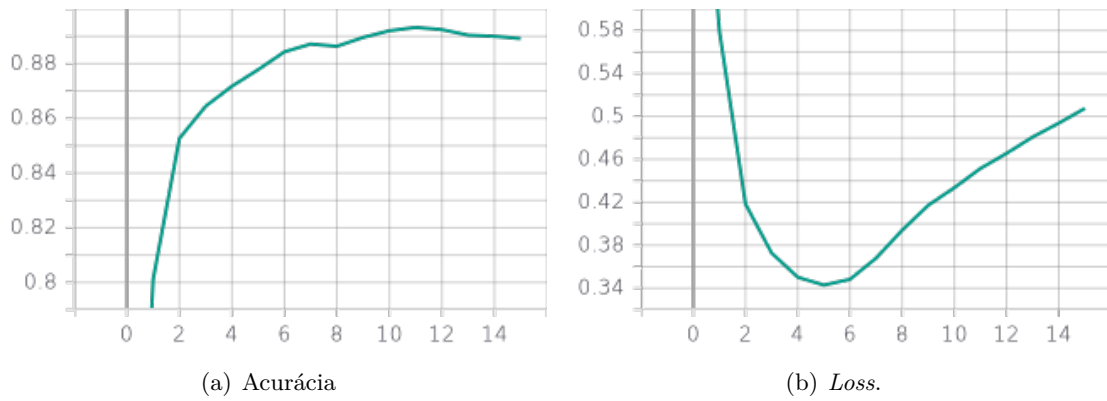


Figura 11: Gráficos de acurácia e *loss* do modelo 11 testado pelo Autokeras.

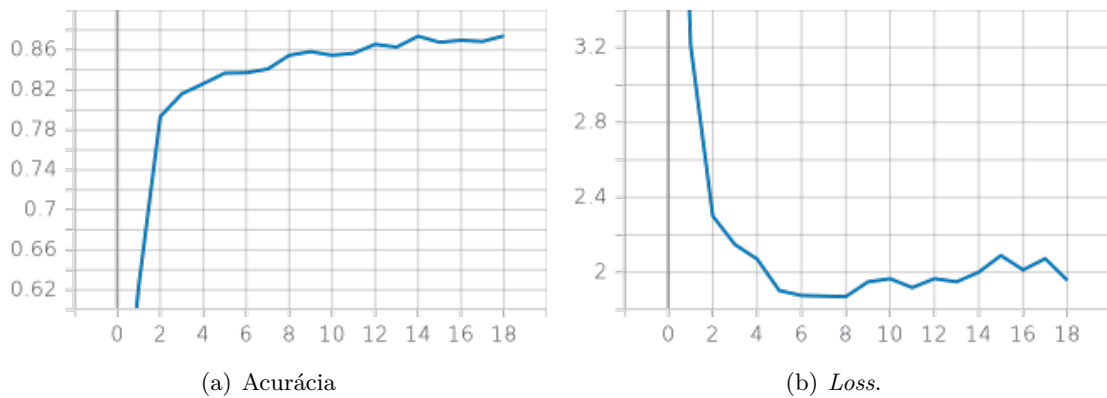


Figura 12: Gráficos de acurácia e *loss* do modelo 12 testado pelo Autokeras.

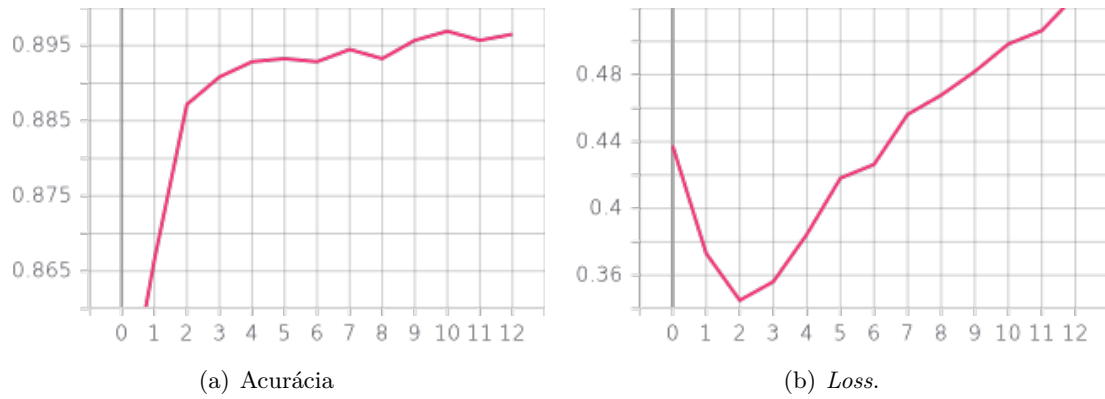


Figura 13: Gráficos de acurácia e *loss* do modelo 13 testado pelo Autokeras.

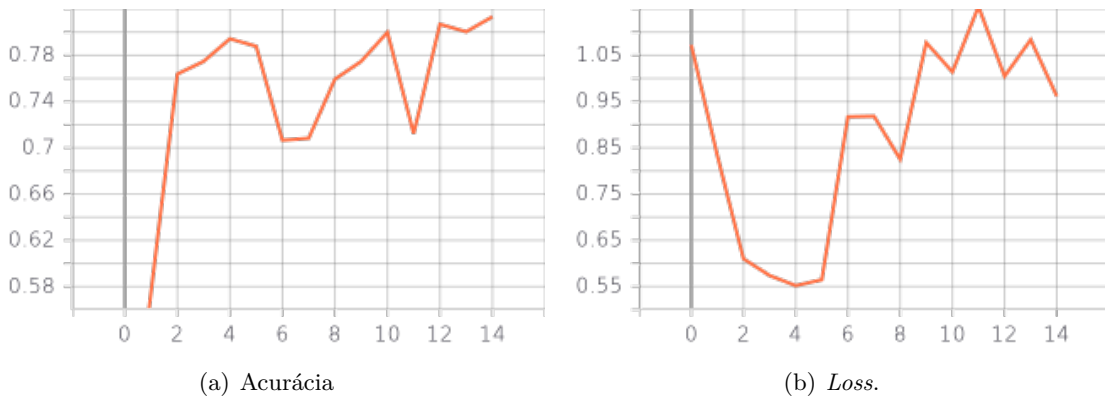


Figura 14: Gráficos de acurácia e *loss* do modelo 14 testado pelo Autokeras.

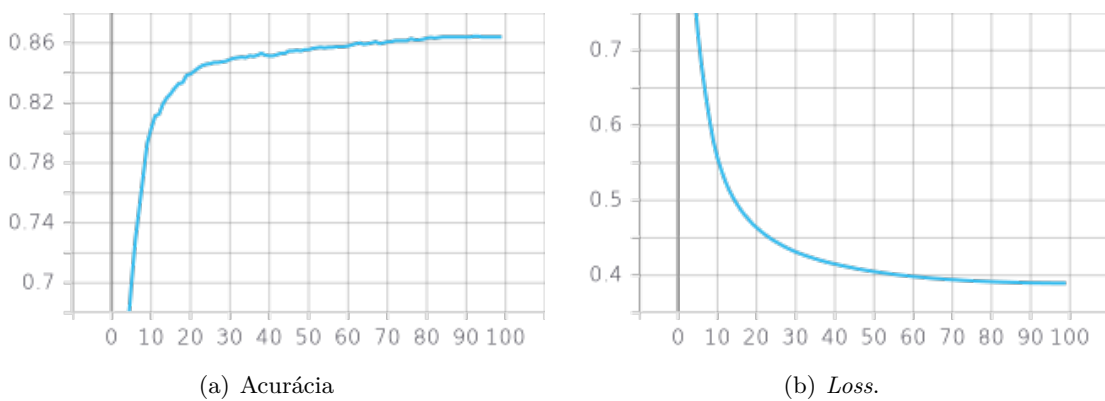


Figura 15: Gráficos de acurácia e *loss* do modelo 15 testado pelo Autokeras.

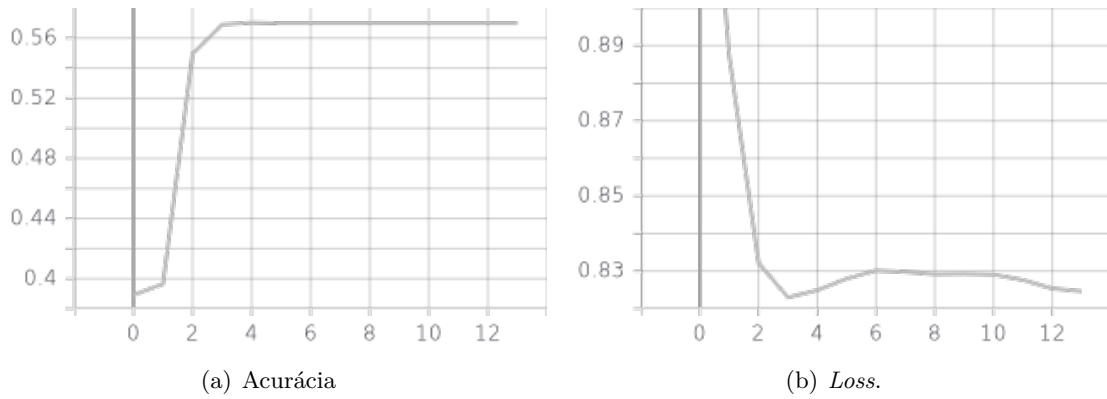


Figura 16: Gráficos de acurácia e *loss* do modelo 16 testado pelo Autokeras.

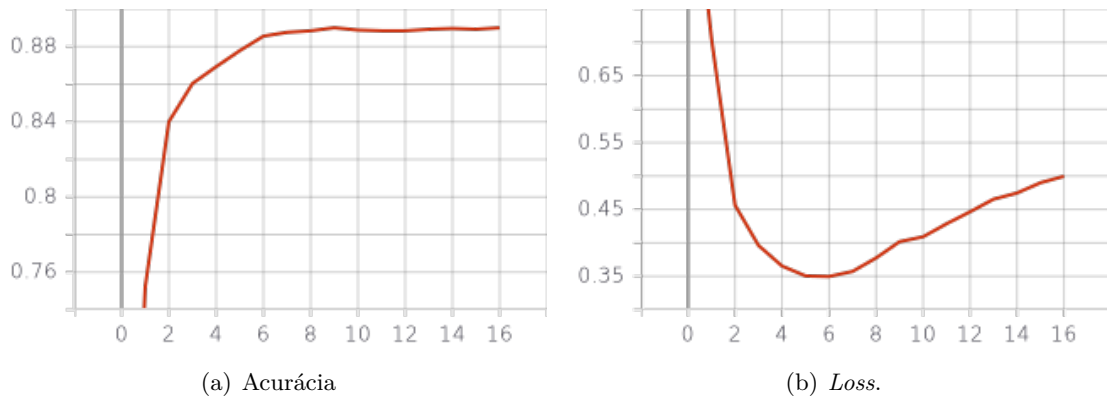


Figura 17: Gráficos de acurácia e *loss* do modelo 17 testado pelo Autokeras.

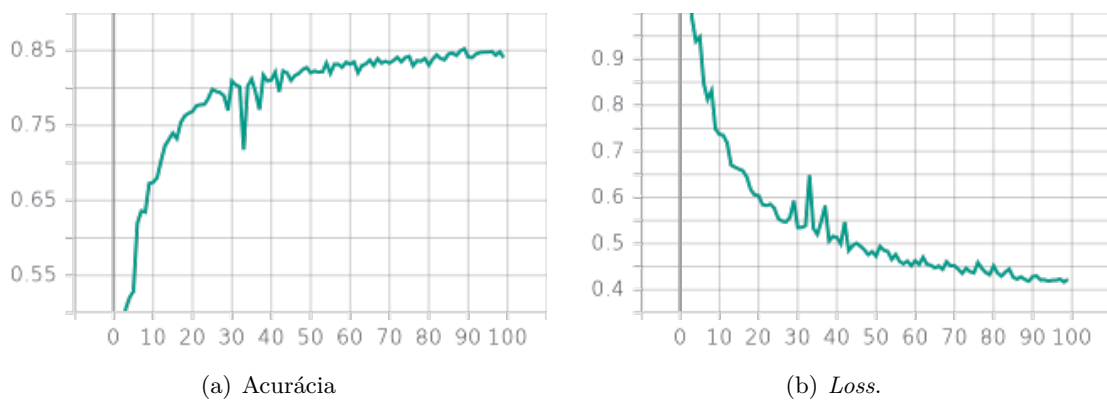


Figura 18: Gráficos de acurácia e *loss* do modelo 18 testado pelo Autokeras.

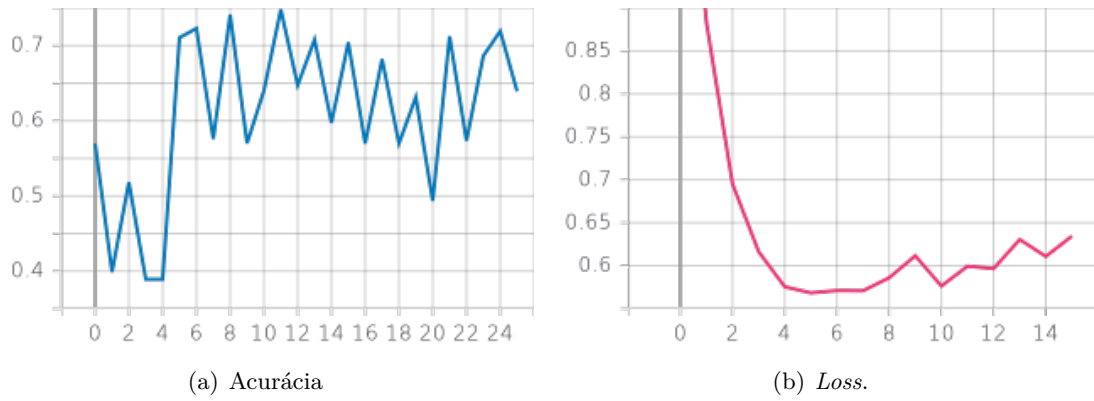


Figura 19: Gráficos de acurácia e *loss* do modelo 19 testado pelo Autokeras.

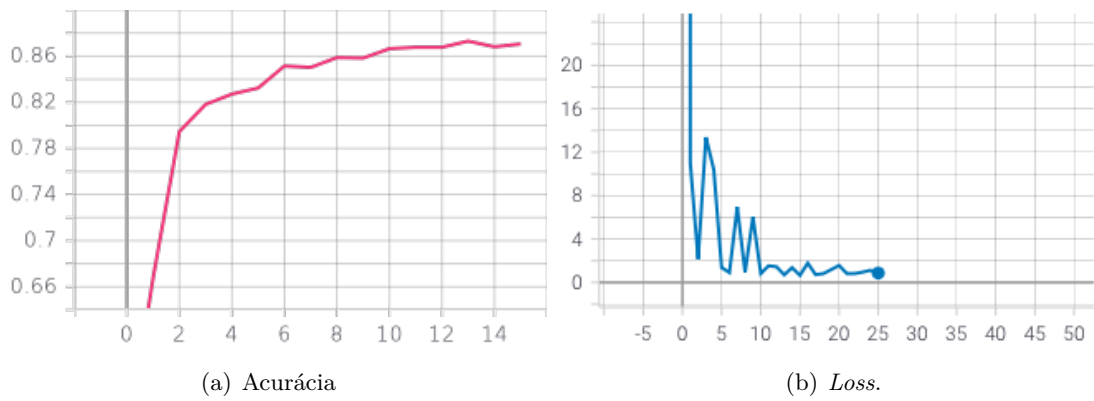


Figura 20: Gráficos de acurácia e *loss* do modelo 20 testado pelo Autokeras.

I.1.2 Treinamentos Finais com Validação Cruzada

I.1.2.1 Adam_raw

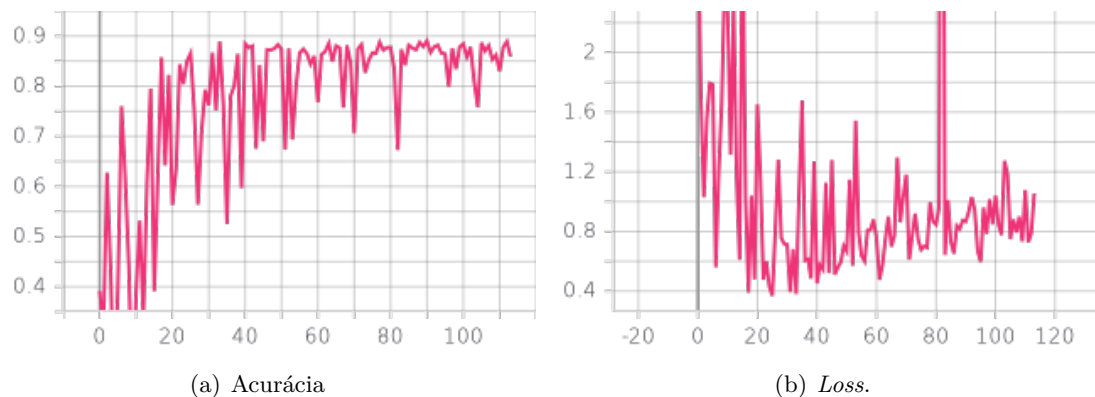


Figura 21: Gráficos de acurácia e *loss* da *fold* 1 de validação cruzada do modelo Adam_raw.

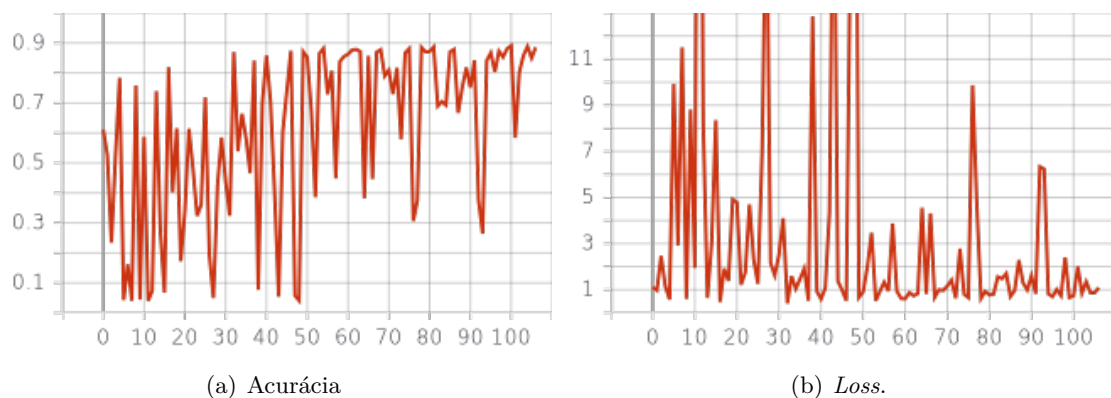


Figura 22: Gráficos de acurácia e *loss* da *fold* 2 de validação cruzada do modelo Adam_raw.

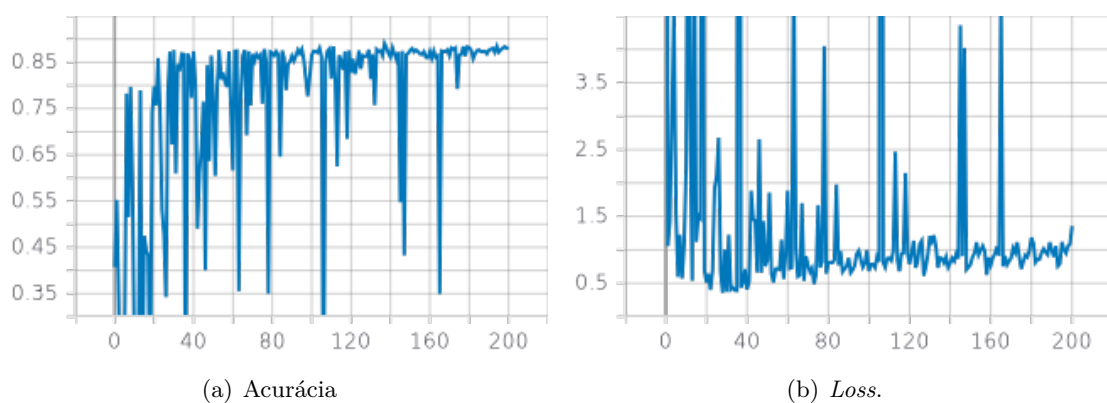


Figura 23: Gráficos de acurácia e *loss* da *fold* 3 de validação cruzada do modelo Adam_raw.

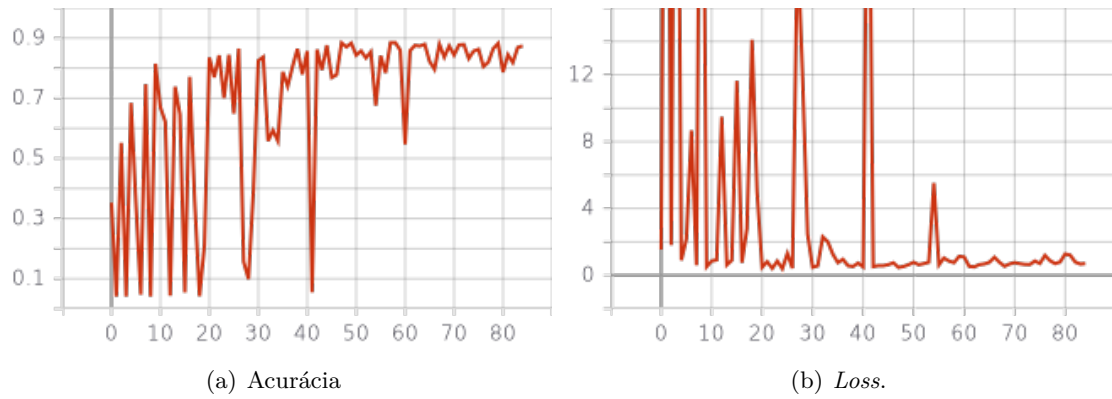


Figura 24: Gráficos de acurácia e *loss* da *fold* 4 de validação cruzada do modelo Adam_raw.

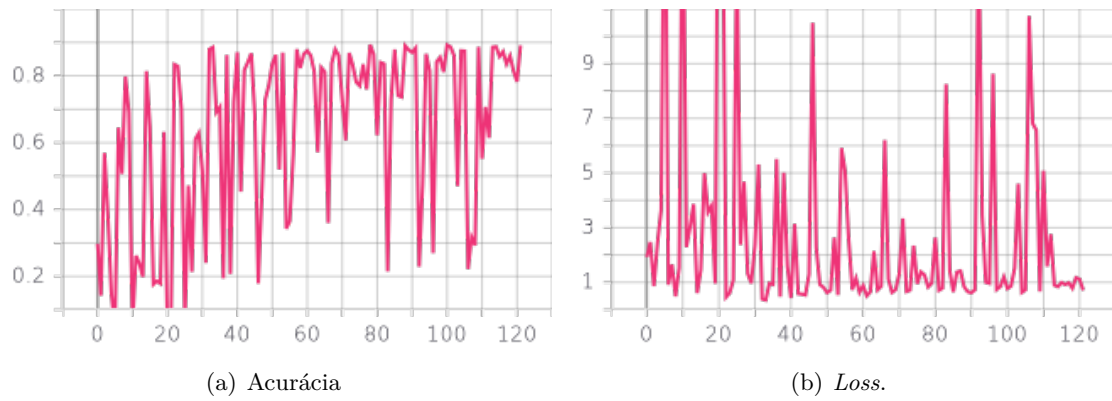


Figura 25: Gráficos de acurácia e *loss* da *fold* 5 de validação cruzada do modelo Adam_raw.

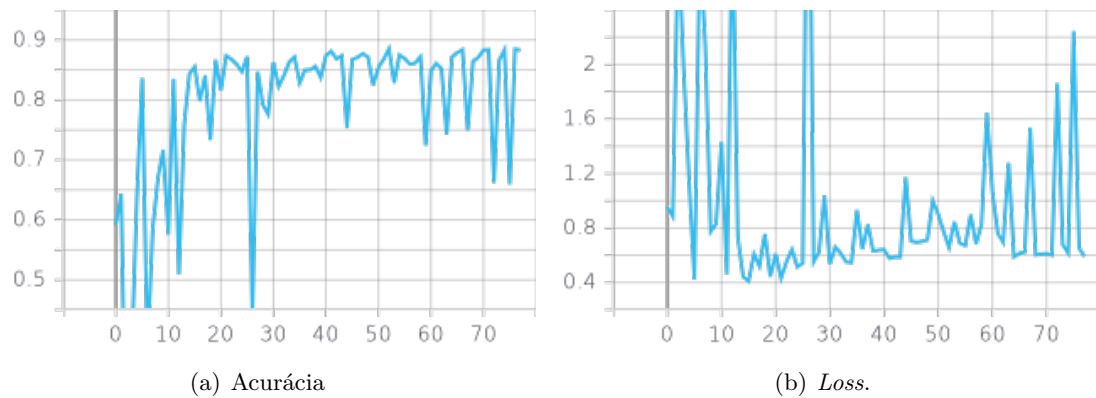


Figura 26: Gráficos de acurácia e *loss* da *fold* 1 de validação cruzada do modelo SGD_raw.

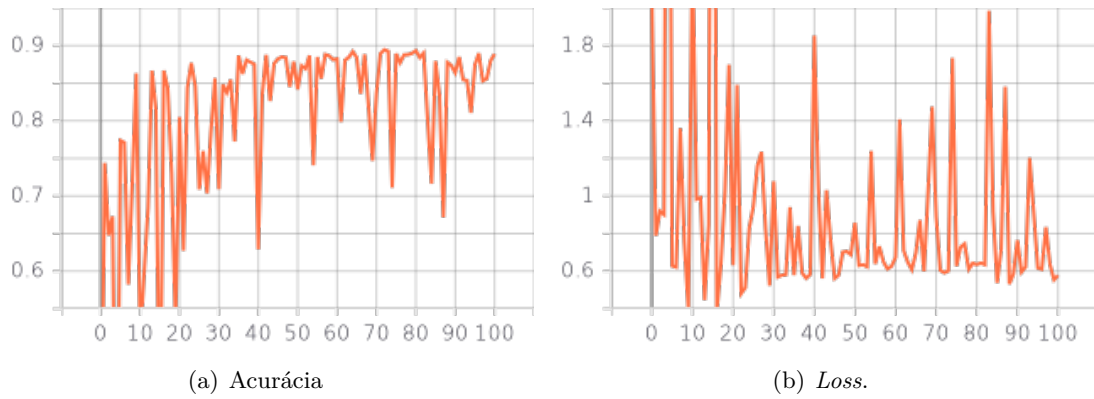


Figura 27: Gráficos de acurácia e *loss* da *fold* 2 de validação cruzada do modelo SGD_raw.

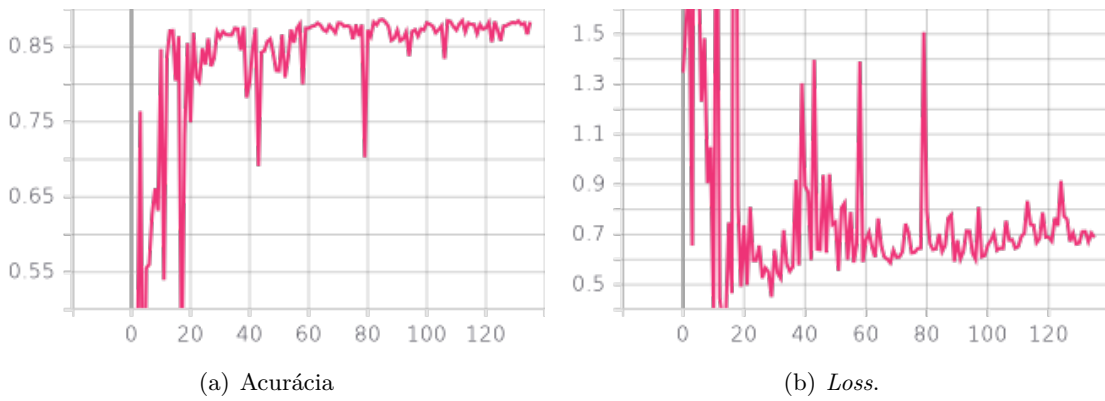


Figura 28: Gráficos de acurácia e *loss* da *fold* 3 de validação cruzada do modelo SGD_raw.

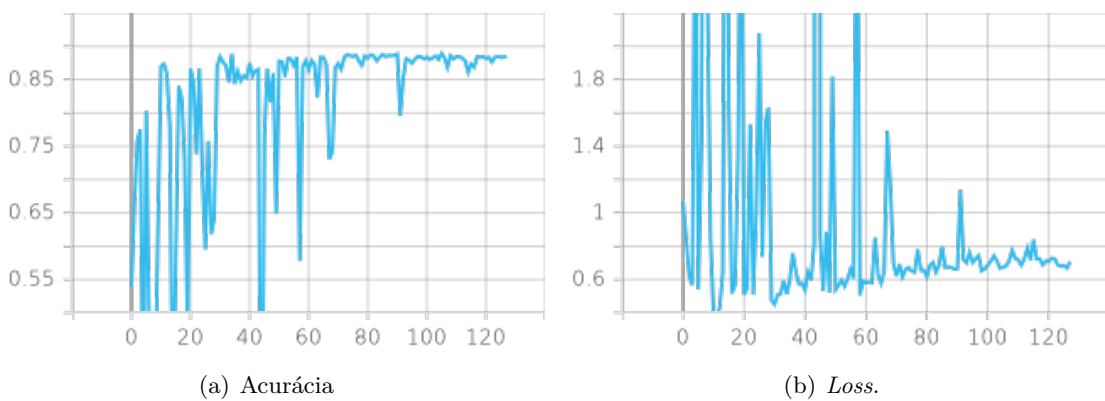


Figura 29: Gráficos de acurácia e *loss* da *fold* 4 de validação cruzada do modelo SGD_raw.

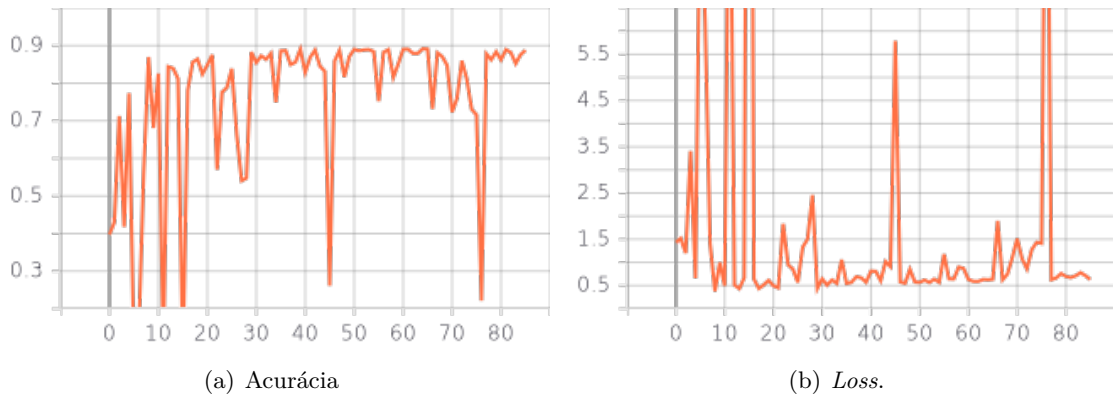


Figura 30: Gráficos de acurácia e *loss* da *fold* 5 de validação cruzada do modelo SGD_raw.

I.1.2.2 SGD_raw

I.1.2.3 Adam_augmented

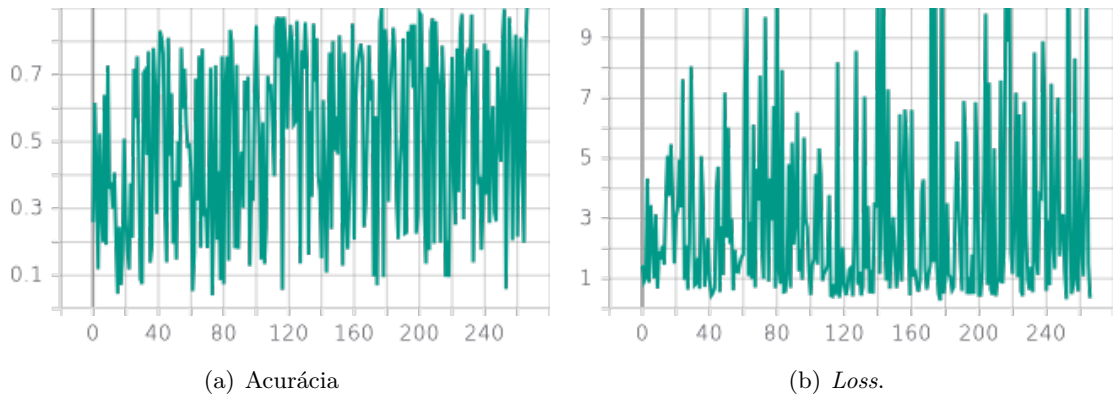


Figura 31: Gráficos de acurácia e *loss* da *fold* 1 de validação cruzada do modelo Adam_augmented.

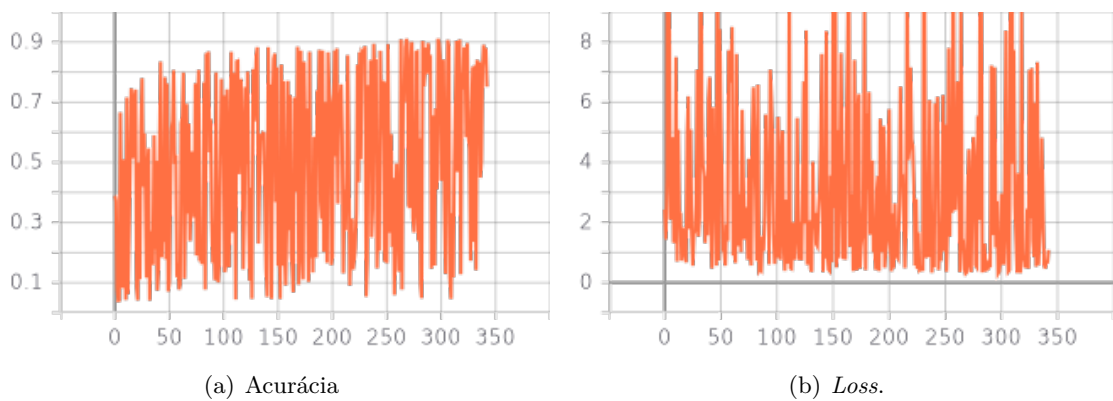


Figura 32: Gráficos de acurácia e *loss* da *fold* 2 de validação cruzada do modelo Adam_augmented.

I.1.2.4 SGD_augmented

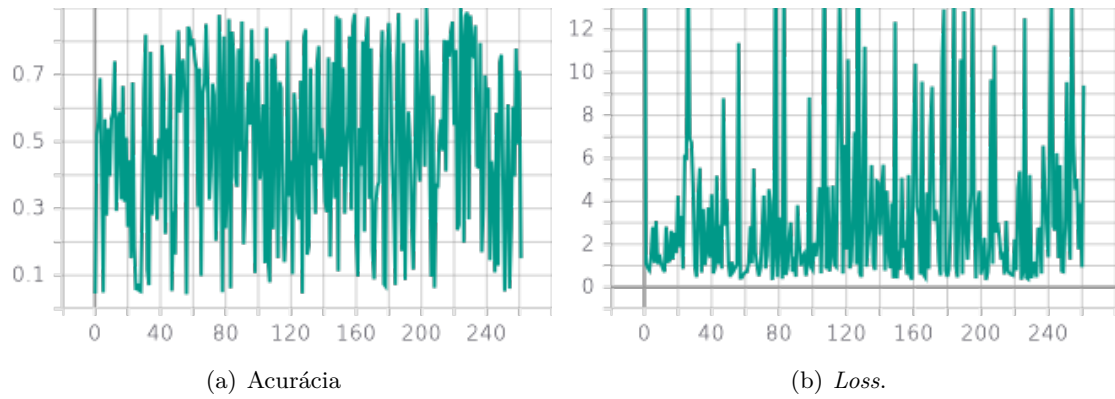


Figura 33: Gráficos de acurácia e *loss* da *fold* 3 de validação cruzada do modelo Adam_augmented.

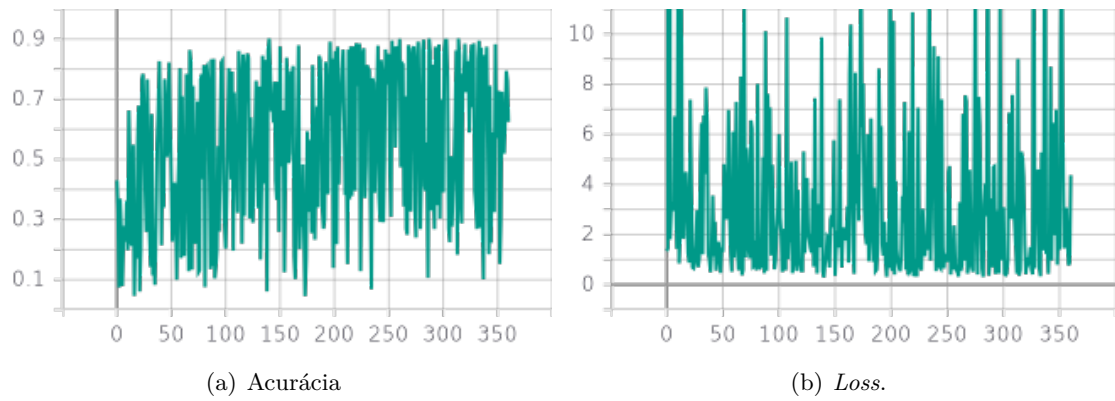


Figura 34: Gráficos de acurácia e *loss* da *fold* 4 de validação cruzada do modelo Adam_augmented.

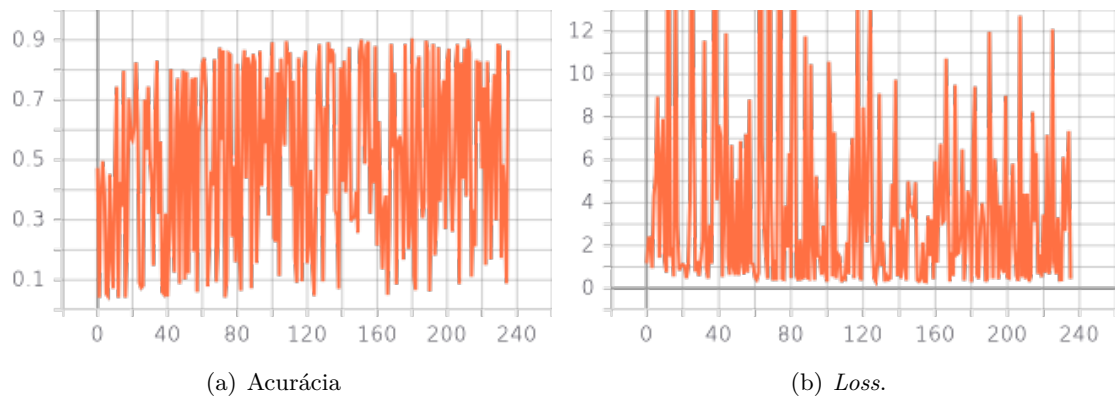


Figura 35: Gráficos de acurácia e *loss* da *fold* 5 de validação cruzada do modelo Adam_augmented.

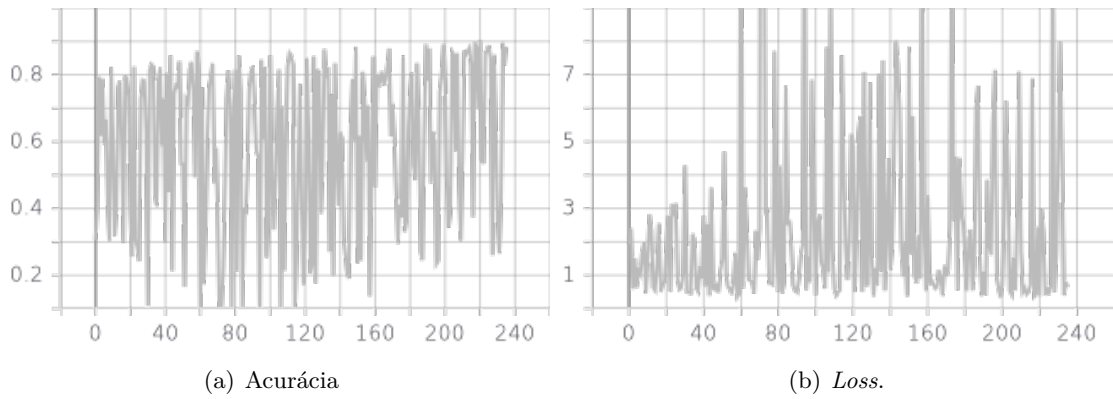


Figura 36: Gráficos de acurácia e *loss* da *fold* 1 de validação cruzada do modelo SGD_augmented.

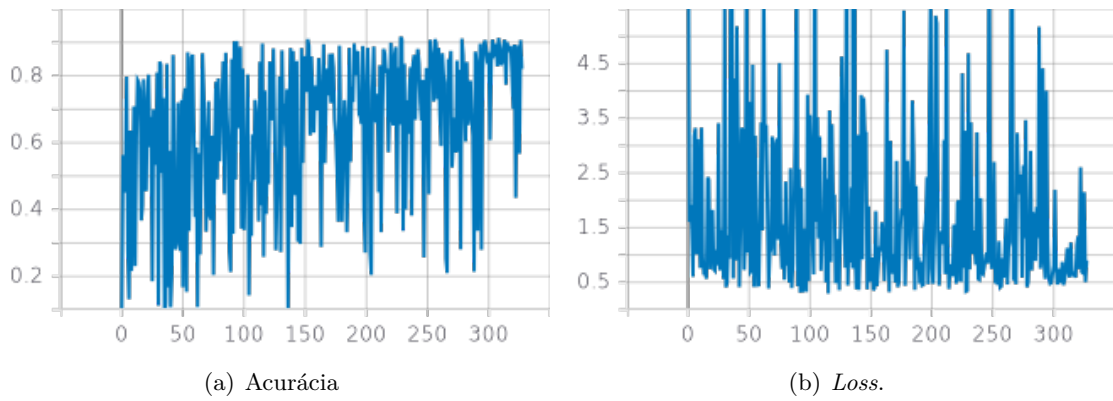


Figura 37: Gráficos de acurácia e *loss* da *fold* 2 de validação cruzada do modelo SGD_augmented.

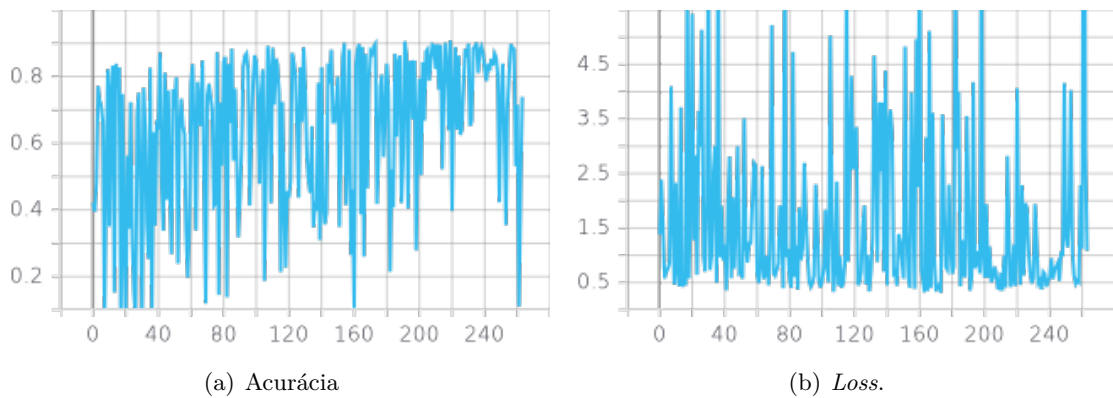


Figura 38: Gráficos de acurácia e *loss* da *fold* 3 de validação cruzada do modelo SGD_augmented.

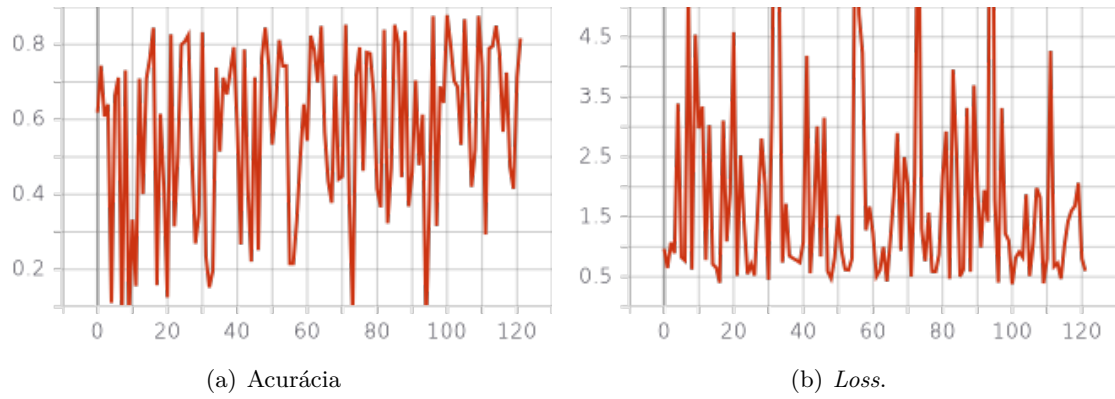


Figura 39: Gráficos de acurácia e *loss* da *fold* 4 de validação cruzada do modelo SGD_augmented.

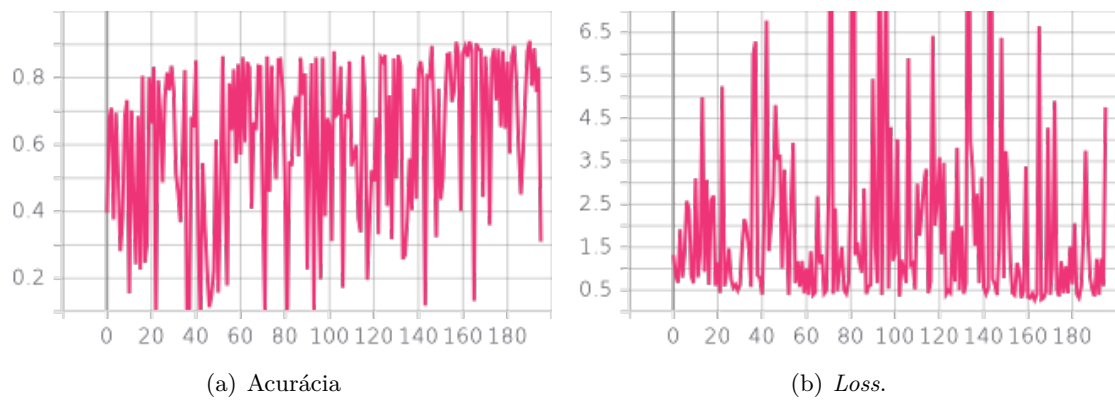


Figura 40: Gráficos de acurácia e *loss* da *fold* 5 de validação cruzada do modelo SGD_augmented.