

UNIVERSIDADE FEDERAL DE ALFENAS

INSTITUTO DE CIÊNCIAS EXATAS

Departamento de Ciência da Computação

Luiz Gustavo Akazawa Nora, Pedro Paulo Miranda Afonso

**IDENTIFICAÇÃO E LOCALIZAÇÃO, DE ANOMALIAS
CAUSADAS PELO COVID-19, EM RADIOGRAFIAS DO
TÓRAX UTILIZANDO INTELIGÊNCIA ARTIFICIAL**

Alfenas, 19 de Agosto de 2022

UNIVERSIDADE FEDERAL DE ALFENAS

INSTITUTO DE CIÊNCIAS EXATAS

Departamento de Ciência da Computação

**IDENTIFICAÇÃO E LOCALIZAÇÃO, DE ANOMALIAS
CAUSADAS PELO COVID-19, EM RADIOGRAFIAS DO
TÓRAX UTILIZANDO INTELIGÊNCIA ARTIFICIAL**

Luiz Gustavo Akazawa Nora, Pedro Paulo Miranda Afonso

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Alfenas como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Ricardo Menezes Salgado

Alfenas, 19 de Agosto de 2022

ESPAÇO RESERVADO PARA A FOLHA DE APROVAÇÃO

AGRADECIMENTO

Agradecemos os nossos familiares por nos apoiarem e incentivarem durante nossas vidas, e possibilitarem de alcançarmos mais um objetivo em nossas vidas.

Agradecemos aos nossos professores, que ao longo dos ensinamentos básico, fundamental e médio nos ajudaram a adquirir conhecimento e nos tornar melhores cidadãos.

Agradecemos ao corpo docente do curso Bacharelado da Ciência da Computação da Universidade Federal de Alfenas, que nos ensinaram tanto e sempre ofereceram apoio quando necessário: Ricardo Menezes, Eliseu, Flávio, Mariane, Luiz Eduardo, Paulo Bressan, Nelson, Ângela, Luciana, Humberto Brandão, Felipe Rey, Marcelo Rezende, Rodrigo Pagliares, Diego e Thiago.

Agradecemos os nossos companheiros de graduação que nos fizeram companhia nos momentos de alegria e de dificuldade.

Resumo

Segundo a Organização Mundial de Saúde (OMS) “*Covid-19 é uma doença infecciosa causada pelo coronavírus (SARS-CoV-2) e tem como principais sintomas febre, cansaço e tosse seca*”. Para o combate à pandemia causada pela doença, são recomendadas algumas medidas, como distanciamento social, máscaras faciais em público, vacinação e testagem em massa. A testagem para Covid-19 é dividida entre testes rápidos e testes precisos. Testes precisos, como o PCR, demoram cerca de 24 horas para serem realizados. Em função disso, o objetivo desse trabalho foi desenvolver um método capaz de detectar a doença através de radiografias do tórax, realizadas em cerca de minutos. Na primeira parte são detectadas opacidades e anomalias em radiografias do tórax. Na segunda parte, essas radiografias são classificadas em classes legíveis para os profissionais de saúde. Para isso são usados modelos pré-treinados junto com algumas técnicas como, *data augmentation, cross-validation e ensemble*.

YOLO (*You Only Look Once*) é um algoritmo de detecção e reconhecimento de objetos em tempo real, que se utiliza de redes neurais convolucionais. Através do uso do YOLO com o intuito de prever opacidades como objetos nas radiografias, foi atingida uma acurácia de 75,23% na detecção dessas opacidades nas imagens. Em relação à classificação, que abrange 4 possíveis classes (*Negative for Pneumonia, Typical Appearance, Indeterminate Appearance e Atypical Appearance*), foi alcançada uma acurácia de 76,23%, o que aliado à análise do especialista representa um resultado satisfatório.

Palavras-Chave: Covid-19, Redes Neural Convolucional. *Ensemble*. Radiografia, *Data Augmentation*. Detecção de Objetos. Classificação de Imagens. YOLO.

Abstract

According to the World Health Organization (WHO) “Covid-19 is an infectious disease caused by the corona-virus (SARS-CoV-2) and its main symptoms are fever, tiredness and dry cough”. In order to combat the pandemic caused by the disease, some measures are recommended such as, social distancing, face masks in public, vaccination and mass testing. Testing for Covid-19 is divided between rapid tests and accurate tests. Precise tests, such as PCR, take about 24 hours to perform. As a result, the objective of this work was to develop a method capable of detecting the disease through chest X-rays, performed in about minutes. In the first part, opacities and abnormalities are detected on chest radiographs. In the second part, these images are classified into classes that are readable by health professionals. Pre-trained models are used along with some techniques such as data augmentation, cross-validation and ensemble.

YOLO (You Only Look Once) is a real-time object detection and recognition algorithm, which uses convolutional neural networks. Using YOLO to predict opacities like objects on radiographs, an accuracy of 75.23% was reached in the detection of these opacities in the images. Regarding the classification, which covers 4 possible classes (Negative for Pneumonia, Typical Appearance, Indeterminate Appearance and Atypical Appearance), an accuracy of 76.23% was achieved, which combined with the expert analysis represents a satisfactory result.

Keywords: Covid-19. Convolutional Neural Network. Ensemble. Radiography, Data Augmentation, Object Detection, Image Classification, YOLO.

LISTA DE FIGURAS

Figura 1: Mortes por ano no Brasil, por todas as causas.	13
Figura 2: Variação do PIB brasileiro em relação ao ano anterior.	14
Figura 3: Modelo do neurônio artificial.	25
Figura 4: Arquitetura de uma rede com duas camadas ocultas.	25
Figura 5: Função ReLU.	26
Figura 6: Exemplo de Data Augmentation.	28
Figura 7: Esquema do ensemble.	30
Figura 8: Arquitetura de uma rede neural convolucional.	31
Figura 9: Camada de convolução.	32
Figura 10: Estrutura da Faster RCNR.	34
Figura 11: Amostra de uma imagem no formato DICOM retirada diretamente da base de dados.	38
Figura 12: Amostra de uma imagem, convertida para o formato JPG, com 2 bounding boxes.	39
Figura 13: Amostra de cada uma das possíveis classes do banco de dados.	40
Figura 14: Gráfico que exemplifica o número de imagens em cada classe do study_level. ...	41
Figura 15: Fluxograma que exemplifica a metodologia da detecção.	42
Figura 16: Amostra de imagens da base de dados, com suas opacidades desenhadas por radiologistas.	43
Figura 17: Amostra das mesmas imagens da figura 16, mas com suas opacidades previstas pelo YOLOv5.	43
Figura 18: Fluxograma que exemplifica a metodologia da classificação.	44
Figura 19: Distribuição de classes do conjunto de teste.	45
Figura 20: Gráficos do efficientnetv2 e efficientnetv2-l-21k-ft1k.	48
Figura 21: Gráficos do efficientnetv2-s e efficientnetv2-s-21k.	49
Figura 22: Gráficos do efficientnetv2-s-21k-ft1k e EFFNet7.	49
Figura 23: Gráficos do InceptionResNetV2 e InceptionV3.	49
Figura 24: Gráficos do ResNet-50 e Xception.	50
Figura 25: Matriz de Confusão efficientnetv2-s-21k-ft1k.	52
Figura 26: Matriz de Confusão final.	53
Figura 27: Recall das classes.	54

Figura 28: Recall das classes.....	55
Figura 29: Gráficos de loss do treino.	56
Figura 30: Gráficos de loss do treino.	57
Figura 31: Matriz de confusão da detecção.....	58
Figura 32: Imagens de teste com predições.....	59

LISTA DE TABELAS

Tabela 1: Parâmetros de treinamento.	46
Tabela 2: Resultados dos modelos.....	50

LISTA DE ABREVIações

AUROC	<i>Area Under the Receiver Operating Characteristic curve</i>
BIMCV	Banco de Imagens Médicas da Região de Valência
COCO	<i>Common Objects Dataset</i>
CNN	<i>Convolutional Neural network</i>
DICOM	<i>Digital Imaging and Communications in Medicine</i>
DO	Detecção de Objetos
FISABIO	A Fundação para o Avanço da Saúde e Pesquisa Biomédica da Região de Valencia
GAN	Generative Adversarial Network
GPU	Graphic Processing Unit
IA	Inteligência Artificial
ILSVRC	<i>ImageNet Large Scale Visual Recognition Challenge</i>
JPEG	<i>Joint Photographics Experts Group</i>
PCR	<i>Polimerase Chain Reaction</i>
PNG	<i>Portable Network Graphics</i>
ReLU	<i>Rectified Linear Unity</i>
RICORD	<i>The RSNA International Radiology Database</i>
RNA	Rede Neural Artificial
R-CNN	<i>Region-based Convolutional Neural Network</i>
TCC	Trabalho de Conclusão de Curso
TPU	Tensor Processing Unit
YOLO	<i>You Only Look Once</i>

SUMÁRIO

1.	Introdução	13
1.1.	Problematização	16
1.2.	Objetivos	17
1.2.1.	Gerais	17
1.2.2.	Específicos	17
1.3	Organização da Monografia	18
2.	Revisão Bibliográfica	19
3.	Fundamentação Teórica	24
3.1.	Redes Neurais Artificiais	24
3.1.1	<i>Data Augmentation</i>	27
3.1.2	<i>Ensemble</i>	28
3.2.	Detecção de Objetos	30
3.2.1	Redes Neurais Convolucionais	31
3.2.2.	Rede Neural Convolucional baseada em Região	32
3.2.3.	<i>Fast R-CNN</i> e <i>Faster R-CNN</i>	32
3.2.4.	<i>You Only Look Once (YOLO)</i>	34
3.3.	Classificação de Imagens	35
4.	Metodologia	37
4.1.	Base de dados	37
4.2	Detecção de objetos	41
4.3	Classificação de Imagens	43
4.3.1	Hiper-Parâmetros	46
4.3.2	<i>Ensemble</i>	46
5.	Resultados	48
5.1.	Resultados dos modelos pré-treinados	48

5.2.	Resultados com ensemble	52
5.3.	Resultados do YOLO	55
6.	Análises e Conclusões	60
6.1.	Modelos Pré-Treinados	60
6.2.	<i>Ensemble e Data Augmentation</i>	60
6.3.	Conclusões	60
6.4.	Trabalhos Futuros	61
	REFERÊNCIAS	62

1. Introdução

Este capítulo apresenta a motivação e os objetivos para a realização deste trabalho. O item 1.1 explica de maneira geral os problemas causados pela SARS-CoV-2, algumas maneiras de combatê-los, e qual a proposta deste trabalho nesta conjuntura. No item 1.2, são explicados pontos fortes e fracos das ferramentas atuais, e como este trabalho pode contribuir neste cenário. No ponto 1.3, são descritos os resultados que este trabalho pretende alcançar. No ponto 1.4, é descrita a organização da monografia.

Desde que surgiu, em 2019, o SARS-CoV-2, um coronavírus causador de uma doença respiratória, assolou o mundo. Neste contexto, o Brasil foi um dos países mais afetados, sendo o terceiro país com mais casos totais confirmados e o segundo país com mais mortes totais confirmadas pela doença, apesar de ser apenas o sexto país mais populoso (OUR WORLD IN DATA, 2022).

No Brasil, até a data de 14/08/2022, mais de 34500000 de casos, com 681253 óbitos confirmados. Esses dados configuram uma incidência de aproximadamente 15042,7 por 100 mil habitantes, além de uma mortalidade de 318,2 por 100 mil habitantes e 2,1% de letalidade. (MINISTÉRIO DA SAÚDE, 2022). A figura 1 demonstra como esses números são capazes de impactar o país.

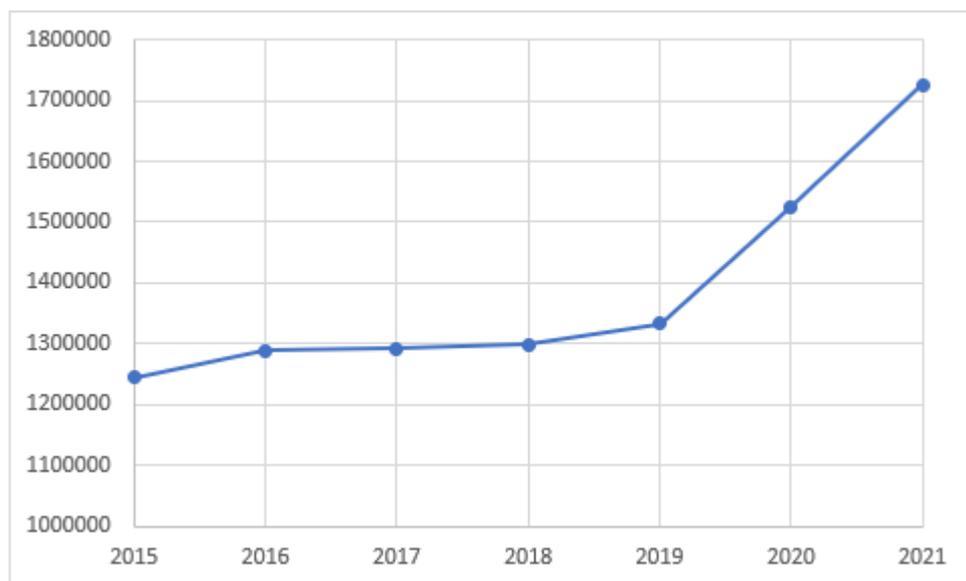


Figura 1: Mortes por ano no Brasil, por todas as causas.

Fonte: (IBGE, 2022).

De 2015 a 2019, houve um crescimento médio de 1,73% no número de mortes por ano no Brasil. Todavia, em 2020, ano do início da propagação nacional da doença, houve um crescimento de 14,45% de mortes em relação ao ano anterior. Em 2021, auge da pandemia, houve um crescimento de 13,21% de mortes em relação ao já elevado número de 2020. O número de mortes causadas pela Covid-19, nome popular da doença causada pelo SARS-CoV-2, em 2021 foi de 412880 (MINISTÉRIO DA SAÚDE, 2022), representando 24% do total de mortes do país, tornando inegável o impacto da doença nesses números.

A doença também gerou consequências na economia brasileira, conforme pode ser observado na figura 2. Este gráfico demonstra a variação de crescimento do PIB (Produto interno bruto) brasileiro em relação ao ano anterior.

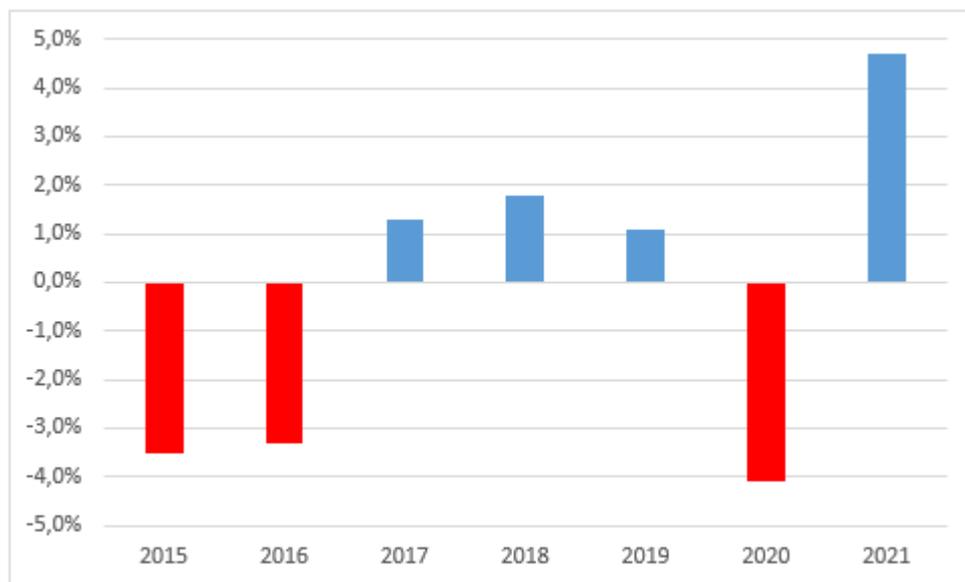


Figura 2: Variação do PIB brasileiro em relação ao ano anterior.

Fonte: (IBGE, 2022).

Após a crise econômica de 2016, a economia brasileira dava lentos sinais de recuperação. Todavia, em 2020, fatores como a insegurança e incertezas causadas pela doença, além de algumas medidas necessárias para conter o vírus e salvar vidas, levaram o produto interno bruto brasileiro a despencar mais de 4% em 2020. Outros indicadores também foram afetados, como o desemprego, que subiu de 11,8% em 2019 para 14,4% em 2020 (IBGE, 2022), e o PIB per capita, que recuou 4,8% no mesmo período (IBGE, 2022). Em 2021, houve uma recuperação do PIB aos patamares pré-pandêmicos, demonstrando recuperação. Todavia, assim como na saúde, o impacto da Covid-19 na economia foi inegável. Essa conjunção de fatores

demonstram a importância do contínuo combate ao vírus, de modo que qualquer fator que ajude a minimizar seus impactos é bem-vindo.

Durante a pandemia, com o intuito de combater o vírus, medidas como distanciamento social, uso de máscaras, filtragem de ar, lavagem das mãos, desenvolvimento e distribuição de vacinas, *lockdowns*, dentre outras foram adotadas (NHSINFORM,2022). Dentre todas as medidas, uma delas, a testagem dos infectados, é o foco deste trabalho.

Para a testagem e consequente diagnóstico da Covid-19, testes foram desenvolvidos. Dentre eles, podem-se destacar o RT-PCR, o teste de antígeno, e o uso de tomografias computadorizadas do tórax. (NHSINFORM,2022).

O RT-PCR, sigla em inglês traduzida como “transcrição reversa seguida de reação em cadeia da polimerase”, é um método laboratorial que consiste em transformar e aumentar o material genético de um vírus. Essa parte do processo consiste em transformar o RNA do vírus em DNA complementar, utilizando da enzima transcriptase reversa, seguida da inserção de dois *primers* que auxiliam na amplificação do material genético. Feito esta parte do processo, é possível observar se o conteúdo da amostra molecular é correspondente a de um agente infeccioso, como por exemplo, o SARS-CoV-2. Com isso, é possível realizar o diagnóstico da Covid-19 (FERREIRA, 2020). O teste PCR tem 86% de sensibilidade quanto ao diagnóstico da Covid-19 (ou seja, do conjunto de diagnósticos positivos, 86% estão corretos) e 96% de especificidade (ou seja, do conjunto de diagnósticos negativos, 96% estão corretos) (FLORIANO et al., 2021).

Os testes de antígeno, também conhecidos como testes rápidos, são realizados por meio da retirada de uma amostra, geralmente nasal, do paciente, e por analisar a reação da amostra com um anticorpo relacionado ao vírus e um anticorpo que serve como controle. No caso da presença do SARS-CoV-2, uma proteína do nucleocapsídeo do vírus irá reagir a ambos os anticorpos, configurando o diagnóstico positivo. O teste de antígeno tem cerca de 80% de sensibilidade e 97% de especificidade. É importante ressaltar que o teste necessita de carga viral suficiente para obter seus melhores resultados, de modo que não é recomendado realizá-lo nos primeiros 4 dias de sintomas, algo que não é problema com o RT-PCR. Por essa razão, o teste é mais ineficiente em assintomáticos, outra desvantagem em relação ao RT-PCR. (OMS, 2021).

Quanto às tomografias do tórax, estudos iniciais foram realizados para a detecção de infecção para a Covid-19 em Wuhan, na China, mas esses testes não podem ser generalizados nem focaram em distinguir a Covid-19 de outras pneumonias. (NHSINFORM,2022). Mais

recentemente, outros estudos, que serão detalhados na revisão bibliográfica, focaram em distinguir, por exemplo, a Covid-19 de outras doenças, como o câncer de pulmão e pneumonias.

Nesta conjuntura, a proposta deste trabalho é criar um método alternativo para a testagem, e consequente diagnóstico da Covid-19, utilizando radiografias do tórax, para detectar e classificar as opacidades presentes nestas, auxiliando radiologistas. Além disso, o algoritmo também é capaz de diferenciar a Covid-19 de outras pneumonias, algo mais complexo que os atuais trabalhos.

1.1. Problematização

Os testes RT-PCR são realizados apenas em laboratórios específicos, de modo que nem todos os hospitais são capazes de realizá-lo, além de demorarem 24 horas, por vezes mais, quando ocorre uma alta demanda de testes. Os testes de antígeno, apesar do curto tempo de 10 a 30 minutos entre a coleta da amostra e resultado (OMS, 2021), baixo custo e demanda de recursos, tem uma taxa de sensibilidade mais baixa diante de cargas virais mais baixas, chegando a cair para 40% em pacientes assintomáticos (IAN W. PRAY et al, 2020), que também podem transmitir a doença (OMS, 2022), de modo que quando é necessário o diagnóstico preciso, é realizado o RT-PCR.

Todavia, radiografias estão disponíveis em minutos, e são realizadas em mais locais do que o RT-PCR. Além disso, uma radiografia é mais barata do que o RT-PCR. Enquanto uma radiografia do tórax custa entre 40 e 60 reais (PROEXAME,2022) para ser realizada, um RT-PCR custa entre 120 e 150 reais (LARGUI,2022). Esse conjunto de fatores leva a conclusão de que a existência de um algoritmo que pudesse diagnosticar a Covid-19 em radiografias iria ajudar a sociedade de modo geral, em especial regiões mais vulneráveis e com menos acesso a recursos. O diagnóstico rápido e preciso da Covid-19 auxilia no tratamento correto e isolamento dos infectados, ajudando a sociedade. (KAGGLE, 2022).

Construir um algoritmo para a classificação de radiografias não é simples. A Covid-19 é similar a outras pneumonias nos pulmões, causando inflamações e acúmulo de fluídos, tornando-a de difícil diagnóstico (KAGGLE, 2022). Além disso, um sistema simples de classificação geralmente é incapaz de classificar a doença de modo aceitável, muitas vezes não conseguindo diferenciá-las de outras. Nesse sentido, são necessários algoritmos inteligentes, de difícil compreensão e que muitas vezes requerem grande capacidade computacional para

atingirem melhores resultados. Segundo (FRANCO, 2010) “*A dificuldade em capturar e processar uma grande quantidade de dados, além das mudanças contínuas de padrões, faz com que a tomada de decisão seja um processo complexo, principalmente pela quantidade de características que o problema pode apresentar*”.

Diante das limitações da tomada de decisão de uma máquina, ao fim, a tomada final da decisão do diagnóstico é do radiologista. Todavia, ferramentas como a proposta por este trabalho podem servir como um norte para o profissional. E isso leva a uma pergunta. *É possível construir uma ferramenta para auxiliar os radiologistas a realizarem o diagnóstico da Covid-19 de forma rápida e precisa?* Essa pergunta é respondida por meio dos resultados obtidos neste trabalho.

1.2. Objetivos

Esta seção apresenta os objetivos a serem atingidos por este trabalho.

1.2.1. Gerais

Este trabalho foi proposto com o objetivo de criar um modelo inteligente que possa servir como suporte na decisão de médicos para o diagnóstico de Covid-19, fazendo uso de inteligência artificial. A ideia foi classificar corretamente a situação do paciente através da radiografia de seu tórax, identificando e localizando uma possível anormalidade que possa haver no seu pulmão. Foram trabalhadas, baseada nas anomalias, a possibilidade da Covid-19 ou de outras pneumonias, diferenciando-as. É importante reforçar a ideia de que o objetivo não é eliminar a necessidade da análise de um especialista, mas sim oferecer uma ferramenta para que seja rápida e precisa no diagnóstico, atuando como um auxílio para o resultado final.

1.2.2. Específicos

Os objetivos específicos deste trabalho foram:

- Gerar um algoritmo capaz de:
 - Detectar opacidades em radiografias.
 - Classificar as radiografias em determinadas classes.
- Encontrar uma forma viável de utilizar um combinador de algoritmos (*ensemble*) para atingir melhores resultados.

- Assegurar que o algoritmo aprenda e não decore, e que seja capaz de detectar e classificar dados além do conjunto que foi treinado.
- Verificar, através de métricas estatísticas e gráficos, a precisão dos resultados. Eles devem ser entregues com maior sensibilidade que o teste rápido e mais rápido que o teste RT-PCR.

1.3 Organização da Monografia

O trabalho encontra-se organizado de acordo com a seguinte sequência: No Capítulo 2 se encontra a revisão bibliográfica deste trabalho, trazendo alguns artigos científicos que trabalham com diagnóstico de doenças através de imagens, em uma abordagem usando inteligência artificial.

A fundamentação teórica é abordada no Capítulo 3, que traz uma explicação de o que são e como funcionam redes neurais artificiais, junto com algumas técnicas úteis na busca de melhores resultados, como *data augmentation* e *ensemble*. Além disso, explora dois ramos importantes de IA para este trabalho, detecção de objetos e classificação de objetos.

O Capítulo 4 detalha a base de dados e explicitar a metodologia utilizada para a resolução do problema proposto neste trabalho.

O Capítulo 5 tem o objetivo de apresentar e analisar os resultados obtidos através da metodologia utilizada.

Por fim, as análises finais e conclusões feitas durante todo o processo de realização deste trabalho estão descritas no Capítulo 6.

2. Revisão Bibliográfica

A revisão bibliográfica deste trabalho é focada em artigos científicos que abordam o uso de inteligência artificial, especialmente para a detecção e classificação de imagens em diferentes linhas de atuação. Parte considerável é referente à área médica, no auxílio ao diagnóstico de doenças, incluindo pneumonias, cânceres e Covid-19.

(CHOUHAN et al., 2020) trabalha com a classificação de imagens para classificar se determinada radiografia é de um pulmão saudável, pneumonia bacteriana ou pneumonia viral. Os autores utilizam um *dataset* do *Guangzhou Women and Children's Medical Center* contando com 5232 imagens no total e trabalham com modelos pré-treinados, combinado estes em um *ensemble* no final. Os modelos escolhidos foram AlexNet, DenseNet121, InceptionV3, resNet18 e GoogLeNet e os resultado obtidos após o *ensemble* foi de 96,39% de acurácia.

(ZHANG et al., 2021) propõe um modelo composto por uma rede de detecção de anomalia e uma rede de previsão de confiança, que compartilham um extrator de características. Utilizando os *datasets* X-VIRAL (apenas com possíveis casos de pneumonia) e X-COVID, este último com apenas 213 imagens. Alcançou 87% no X-VIRAL e 83% no X-COVID.

Usando um *dataset* de 33676 imagens contendo 4 classes (normal, câncer, pneumonia e Covid-19) (IBRAHIM; ELSHENNAWY; SARHAN, 2021) conseguiu alcançar uma acurácia de 98,05%, *recall* de 98,05% e *F1 score* 98,24 ao combinar o modelo pré-treinado VGG19 com uma rede neural convolucional (CNN).

(HAGHANIFAR et al., 2022) propõe a COVID-CXNet que é uma rede baseada no modelo CheXNet pré-treinado no *dataset* Covid-19 CXR e conta com 431 camadas e aproximadamente 7 milhões de parâmetros. Após testar diferentes variações do COVID-CXNet o melhor resultado obtido foi COVID-CXNet v1 com 99,04% de acurácia em um *dataset* de 3,628 imagens.

Com o intuito de classificar Covid-19 em radiografias do tórax, (DASH; MOHAPATRA, 2022) propôs o uso de uma CNN, o VGG-16, devidamente adaptado às características da base de dados, que contém 1419 imagens. Ao fim da classificação, foi obtida uma acurácia de 97,12%.

(ARIELA, 2022) propôs o uso de uma rede neural convolucional para a classificação de Covid-19 em tomografias computadorizadas. Foi feito um *ensemble* entre o VGG-19 e o Inception V3, gerando uma acurácia final de 99,5%. A base de dados continha 1227 imagens.

(ZAIDI et al., 2021) realizou uma análise de diferentes modelos de *Deep Learning* para a detecção de objetos, comparando-os em 4 diferentes bases de dados, com respectivamente: 5717, 118.287, 455.567 e 1,743,042 imagens. Dentre os modelos analisados, destacaram-se EfficientDet-D7, DetectoRS e Swin-L, com respectivas acurácias médias de 52,65%;53,95% e 57,70%.

Na área de *Neural Architecture Search* (NAS), buscando trazer um método mais eficiente computacionalmente, (LU et al., 2021) propuseram um algoritmo evolutivo multiobjetivo para fazer a pesquisa de arquiteturas neurais, através de uma cuidadosa redução de escala durante a pesquisa, unido a um reforço dos padrões comumente compartilhados entre arquiteturas de sucesso anteriores por meio do aprendizado de modelos bayesianos. A rede foi chamada de NSGANetV1 e quando usada com o *dataset* ChestX-Ray14, contendo 112120 radiografias do tórax com diversos exemplos de doenças comuns, alcançou 84,6% na métrica AUROC (*Area Under the Receiver Operating Characteristic curve*).

(ZHOU et al., 2020) propõe um método rápido, preciso e independente de máquina que automaticamente consegue segmentar e quantificar as regiões de infecção de imagens de tomografia computadorizada de diferentes fontes. Fizeram seus testes em 201 imagens de 140 pacientes de Covid-19 vindos de 4 hospitais diferentes, que utilizavam diferentes aparelhos de tomografia. Os resultados mostraram um F1-score de 0,783 usando *data augmentation*.

Fast Covid-19 Detector (FCOD) é um modelo de aprendizagem profundo baseado na arquitetura Inception e usa 17 camadas de convolução separáveis por profundidade o que diminui o custo computacional de tempo e memória e ainda pode diminuir o número de parâmetros da rede. Para avaliar seu modelo (PANAHI; RAFIEI; REZAEI, 2020) usaram o *covid-chestxray-dataset* que conta com 914 imagens públicas de radiografias do tórax, obtendo uma acurácia de 96%.

Em seu trabalho (WANG et al., 2020) propõe um método composto de uma rede 3D-ResNet que é treinada como um classificador binário usando imagens de pulmões para destacar áreas lesionadas. Simultaneamente, em uma outra ramificação, blocos PARL (*prior-attention residual learning*) são usados para gerar mapas que servem para auxiliar a primeira ramificação a aprender representações mais distintas para a classificação da pneumonia e outras doenças. Usando um total de 4657 imagens seus resultados variam de 76,9% a 93,8% de acurácia para casos de Covid-19.

(HSU; ZHUANG; LEE, 2020) propuseram o uso de uma DenseNet modificada para o *pairwise learning* para combater as Generative Adversarial Networks (GANs), redes neurais que podem ser utilizadas maleficamente para a criação de imagens falsas foto-realistas. Obtiveram, no mínimo, 90% de precisão e 85% de *recall* em 3 diferentes bases de dados com 600 mil imagens cada.

(CHAUHAN; GHANSHALA; JOSHI, 2018) demonstraram a capacidade das redes neurais convolucionais na detecção e classificação de imagens. Seus algoritmos tiveram seus desempenhos avaliados em 2 bases de dados, MNIST, uma base de dados composta por 70000 imagens, e CIFAR-10, base de dados composta por 60000 imagens. Obtiveram 99,6% de acurácia na primeira base e 80,17% na segunda.

Para a detecção de nódulos no pulmão em imagens 3D, (WANG et al., 2020) desenvolveram um novo método supervisionado denominado FocalMix. Utilizaram 2 bases de dados, LUNA16, com 888 imagens, e NLST, com 75000 imagens. Usaram como métrica o CPM. Obtiveram 87,2% na primeira base e 90,7% na segunda.

Para a detecção de câncer de mama em imagens termográficas (REDDY ALLUGUNTI, 2022) propôs o uso de diferentes algoritmos de *Machine Learning*. Sua base de dados continha mais de 1000 imagens de 150 pacientes. Dentre os algoritmos propostos, as redes neurais convolucionais obtiveram o melhor resultado, com uma acurácia de 99,65%.

O uso da U-NET, uma rede neural que provém de customizações da CNN, foi utilizada por (BOUSLAMA; LAAZIZ; TALI, 2020) para o diagnóstico e localização de cardiomegalias em radiografias. Obtiveram uma acurácia entre 93 e 94% na detecção da doença. Para o trabalho, foi utilizada uma base de dados com 1010 imagens.

(MANOHARAN, 2019) propôs o uso das *Least-squares support-vector machines* (LS-SVM), um método supervisionado de classificação, para a detecção de vazamentos em ar condicionados de automóveis. Utilizando um total de 2218 imagens, obtiveram 98% de acurácia na classificação dos vazamentos.

(HASOON et al., 2021) baseia seu trabalho nos algoritmos de classificação *K-Nearest Neighbor* (KNN) e *Support Vector Machine* (SVM) combinados a um pré-processamento de imagens, segmentação e detecção por Região de Interesse e técnicas de extração de características. Dessa forma gerando 6 diferentes modelos, são eles LBP-KNN, HOG-KNN,

Haralick-KNN, LBP-SVM, HOG-SVM e Haralick-SVM. O resultado alcançado varia de 89,2% até 98,66% de acurácia.

(RAJARAMAN et al., 2020) faz uso de alguns modelos pré-treinados (VGG-16, VGG-19, Inception-V3, Xception, InceptionResNet-V2, MobileNet- V2, DenseNet-201 e NasNet-mobile) e uma rede neural convolucional customizada para aprender a classificar radiografias em normal, pneumonia bacteriana ou Covid-19. Os modelos que tiveram melhores resultados são iterativamente podados para reduzir complexidade e deixar o uso de memória mais eficiente. Então, as saídas dos modelos podados de melhor *performance* são combinadas usando diferentes estratégias de *ensemble*. O melhor resultado alcançado foi de 99%, em um conjunto de mais de 10000 imagens vindas de diferentes fontes.

(GAO et al., 2021) divide seu método em 3 partes. Primeiro, uma rede baseada na U-net responsável por extrair regiões críticas dos pulmões. Na segunda parte uma nova rede proposta chamada *Dual-branch combination network* (DCN) que consegue simultaneamente fazer classificação e segmentação das imagens de tomografia computadorizada. Na parte final, os segmentos da imagem que saem da DCN são integrados com um mapa de probabilidades para obter as classes de resultado. Validando em um *dataset* de 1202 imagens o resultado atingido foi de 92,87% de acurácia.

(MACHADO et al., 2022) propuseram o uso de um algoritmo Interpretable Model-Agnostic Explanations (LIME), com o uso de *Deep Learning*, para a identificação de sintomas da Covid-19 usando 907 relatos de usuários da rede social Twitter. Obtiveram 97% de acurácia e 0.95 na curva AUC-ROC.

(LV; WANG, 2020) fez a comparação do desempenho de vários classificadores na classificação de imagens hiperespectrais. Os resultados obtidos pelos métodos que utilizam *Deep Learning* foram melhores que os classificadores ordinários, dos quais as *Deep Belief Network* (DBN) e SAE, ambas redes baseadas em *Restricted Boltzmann machines* (RBMs), obtiveram melhores acurácias, respectivamente 95,1% e 96,5%.

(ABBAS; ABDELSAMEA; GABER, 2021) propôs o uso de uma CNN adaptada, nomeada DeTraC, para a classificação de Covid-19 em radiografias. Com uma base de dados de 196 imagens, obtiveram 93% de acurácia, e uma sensibilidade de 100%, resultado obtido ao aplicar o DeTraC ao VGG19.

Para a detecção e classificação de pneumonias por radiografias (GARCEZ; SOUSA; LUÍS -MA, [s.d.]) propôs o uso de uma rede neural convolucional adaptada para o problema. A base de dados continha 5232 imagens, e foi obtida uma acurácia de 95,5% na detecção e 83,1% na classificação.

3. Fundamentação Teórica

Primeiramente, este capítulo tem o objetivo de explicar o funcionamento e a estrutura básica de uma Rede Neural Artificial (RNA) através do item 3.1. Além de trazer também duas técnicas importantes para a melhoria dos resultados, que foram usadas neste trabalho. Depois, no item 3.2 são apresentados uma visão geral e os objetivos de uma das áreas mais importantes da computação gráfica que é a detecção de objetos, junto com alguns modelos eficientes em executar tal tarefa. Por fim, o último item 3.3 traz uma apresentação de outra abordagem importante neste trabalho, que é a classificação de imagens.

3.1. Redes Neurais Artificiais

Nas palavras de (KUMAR; SHARMA, 2014) “*Redes Neurais Artificiais (RNA) são modelos computacionais ou matemáticos inspirados pelo sistema nervoso central humano (particularmente o cérebro) que é capaz de realizar aprendizado de máquina assim como reconhecimento de padrões*”. De maneira resumida, um neurônio é uma célula eletricamente excitável que se comunica com outras células através de conexões chamadas sinapses. É importante lembrar que o funcionamento do cérebro humano é bem mais complexo e ainda precisa ser mais profundamente estudado. Dito isto, a ideia por trás da RNA é construir camadas interconectadas de neurônios artificiais buscando alcançar uma certa capacidade de aprendizado (WANKER, 2014).

Em RNAs existem dois tipos de aprendizado (supervisionado e não supervisionado), porém, neste trabalho foi utilizado somente aprendizado supervisionado que se trata de um caso onde os rótulos dos dados são conhecidos. Resumidamente, nessa abordagem de aprendizado, os dados de entrada são alimentados na rede obtendo-se o valor de saída, este valor é comparado com o valor alvo e um erro é calculado. O erro é propagado na direção inversa na rede e os valores dos pesos são atualizados. Esse processo de ajuste de pesos se repete um número de vezes definido pelo programador ou até que o resultado alvo seja atingido. O aprendizado acontece à medida em que os pesos da rede são atualizados buscando uma melhoria no resultado de saída.

Como dito anteriormente os neurônios artificiais são a base para o funcionamento de uma RNA, a figura 3 apresenta a estrutura de um desses neurônios.

Uma rede neural é composta de camada de entrada, camada oculta e camada de saída. Caso exista apenas uma camada oculta, a rede é chamada de rede rasa, no caso de múltiplas camadas ocultas chama-se de rede profunda. Um exemplo deste tipo de rede é mostrado na figura 4 onde existem duas camadas ocultas. No mundo real os modelos são construídos com inúmeras camadas ocultas, tirando proveito de algoritmos de aprendizado profundo.

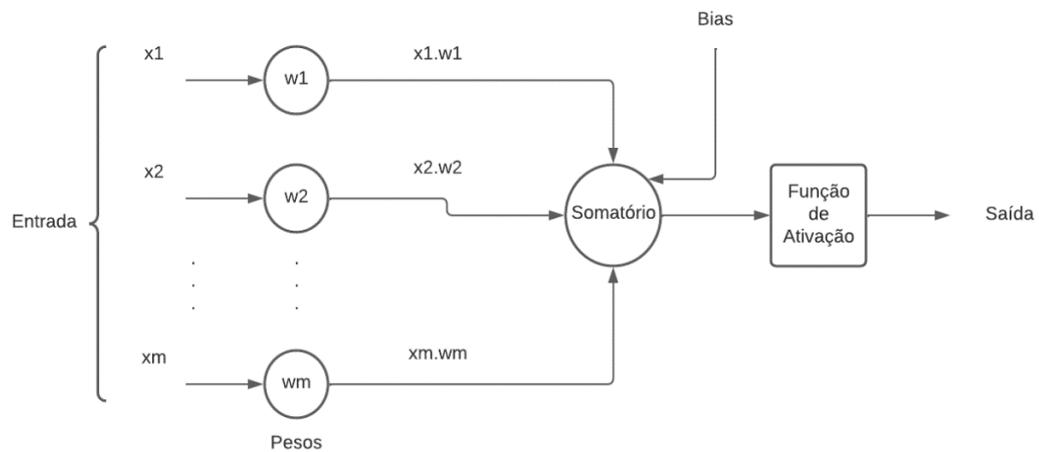


Figura 3: Modelo do neurônio artificial.

Fonte: Elaborada pelos Autores (2022).

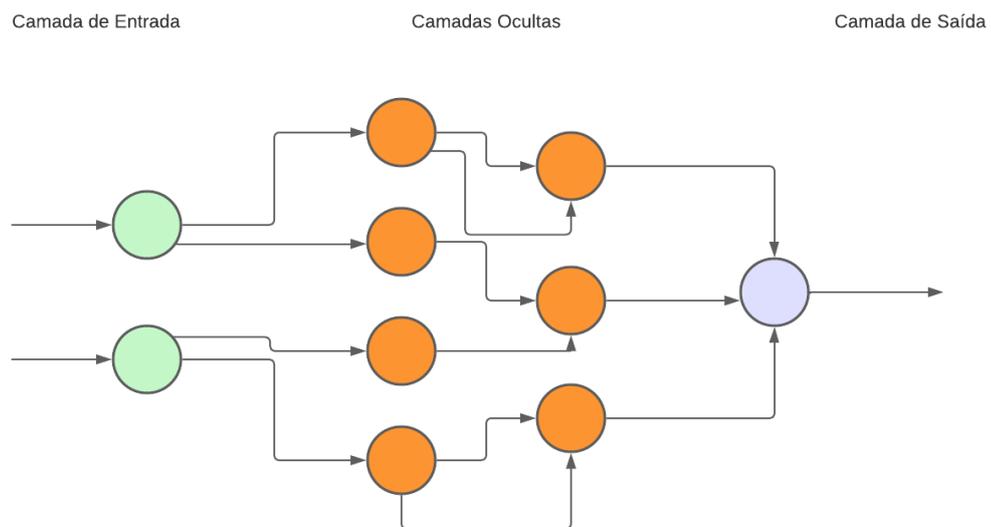


Figura 4: Arquitetura de uma rede com duas camadas ocultas.

Fonte: Elaborada pelos Autores (2022).

Os valores de entrada x_m entram no que seriam as sinapses e são multiplicados pelos pesos w_m , então são todos somados e um valor de bias é adicionado. No final, este valor passa por uma função de ativação, gerando a saída do neurônio.

O valor de saída do neurônio se dá pelas equações (1) e (2):

$$u = \sum_{i=0}^m (w_i x_i) + b \quad (1)$$

$$saida = S(u) \quad (2)$$

Sendo S a função de ativação e b o valor do bias. Dentre as várias funções de ativação, as duas mais importantes para esse trabalho são ReLU (Rectified Linear Unit) e Softmax. A função ReLU é uma função não linear que traz um ganho de eficiência em relação a algumas outras funções, já que os neurônios não são todos ativados ao mesmo tempo, mas apenas parte deles (SHARMA; SHARMA; ANIDHYA, 2020). Isso implica que um neurônio só é desativado quando a saída é 0. A função é dada pela equação (3) e seu comportamento é mostrado na figura 5.

$$f(x) = (0, x) \quad (3)$$

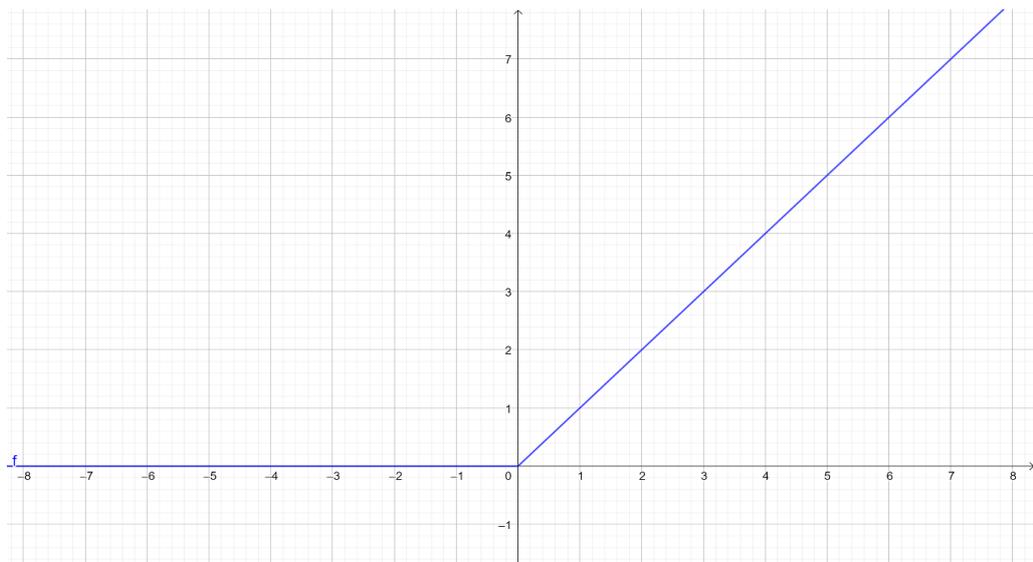


Figura 5: Função ReLU.

Fonte: Elaborada pelos Autores (2022).

Já a função Softmax é principalmente usada como a última função de ativação para normalizar a saída da rede neural em uma distribuição de probabilidades das classes a serem preditas. A função pode ser expressa pela fórmula (4):

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^K e^{z_k}} \quad (4)$$

Onde z é o vetor de entrada, sendo z_i cada elemento deste vetor. E o vetor K representa o número de classes do problema. Quando a rede é construída para classificação de n classes, a camada de saída é configurada para ter n neurônios e a função softmax é selecionada para esta camada.

Como base na apresentação resumida da estrutura e funcionamento de uma rede neural, os próximos itens trazem algumas técnicas que buscam melhorar a performance e eficiência ao trabalhar com essas redes.

3.1.1 *Data Augmentation*

A técnica de *Data Augmentation* busca aumentar o tamanho do *dataset* através de uma geração artificial de novas imagens podendo usar dois métodos diferentes (SHORTEN; KHOSHGOFTAAR, 2019):

- *Data Warping*: transformar os dados de forma que o rótulo das imagens seja mantido. Isso abrange transformações geométricas e de cor, apagamento aleatório e transferência de estilo neural.
- *Oversampling*: cria instâncias sintéticas e adiciona-as no conjunto de dados de treino. Por exemplo, mistura de imagens.

Resumindo, as diferenças de maneira básica, *data warping* aplica transformações no espaço dos dados e *oversampling* aplica transformações no espaço das características (WONG et al., 2016).

A figura 6 é um exemplo de como a técnica de *data augmentation* pode ser usada na busca de gerar uma variação maior a partir de uma imagem.

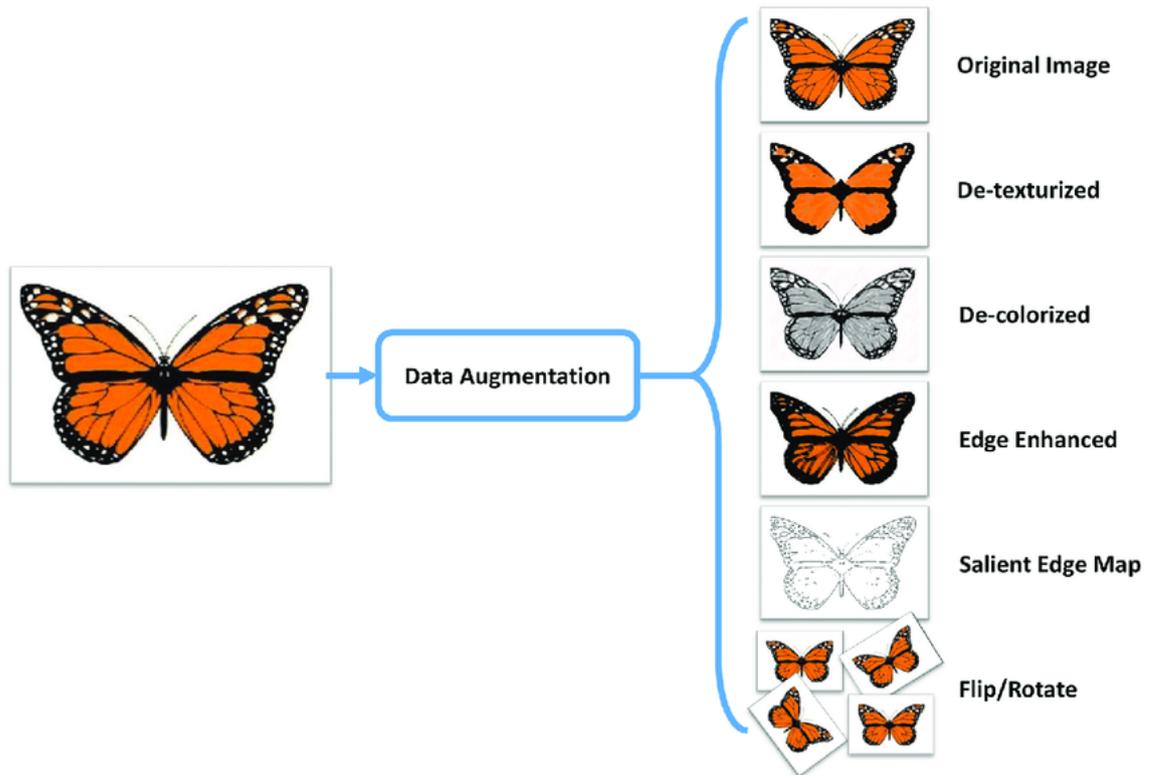


Figura 6: Exemplo de Data Augmentation.

Fonte: (KUMAR, 2019).

Uma característica importante é que a *data augmentation* aborda o *overfitting* na raiz do problema. De acordo com a definição de (YING, 2019) “*Overfitting é uma questão fundamental no aprendizado de máquina supervisionado que impede que os modelos atinjam uma generalização perfeita dos dados observados, assim como dos dados inéditos do conjunto de testes*”. Os principais motivos para o acontecimento deste fenômeno são presença de ruído, tamanho limitado do conjunto de dados de treino e complexidade dos classificadores. Essa técnica é usada neste trabalho visando evitar o *overfitting* causado pelo desbalanceamento dos dados.

3.1.2 Ensemble

A técnica de *Ensemble* consiste em combinar as decisões de diferentes modelos de aprendizagem buscando aumentar o resultado geral do sistema (MENZIES et al., 2015). Segundo (MACLIN, 2016) pesquisas demonstraram que um bom *ensemble* é aquele onde os classificadores individuais apresentam bons resultados e cometem seus erros em partes

diferentes do espaço de entrada. Os modelos são treinados de maneira independente, os resultados de cada um desses modelos são combinados buscando elevar a taxa de acertos que foi obtida nos modelos individuais. Porém, em uma situação na qual poucos dos modelos conseguem boa precisão em determinada classe, e a maioria apresenta resultados com baixo desempenho no mesmo caso, a saída combinada pode ter um resultado pior, pois o resultado dos modelos de performance inferior é amplificado na combinação. Portanto, o *ensemble* não garante que a combinação final melhore o resultado do sistema.

O *ensemble* possui diferentes estratégias e maneiras em como é aplicado. Neste trabalho, o treinamento é feito em cada um dos modelos de maneira isolada gerando suas respectivas saídas como é mostrado na figura 7. Essas saídas são combinadas de forma que as respostas que repetem com maior frequência são selecionadas, para exemplificar de uma maneira simples considere as saídas dos seguintes modelos:

- Modelo 1: A
- Modelo 2: B
- Modelo 3: B
- Modelo 4: C

A saída selecionada seria a classe B, pois acontece com mais frequência entre os modelos analisados. Caso aconteça um empate, algum critério deve ser definido, podendo ser por exemplo escolher a classe com menor taxa de acerto entre os modelos ou a classe com maior número de imagens.

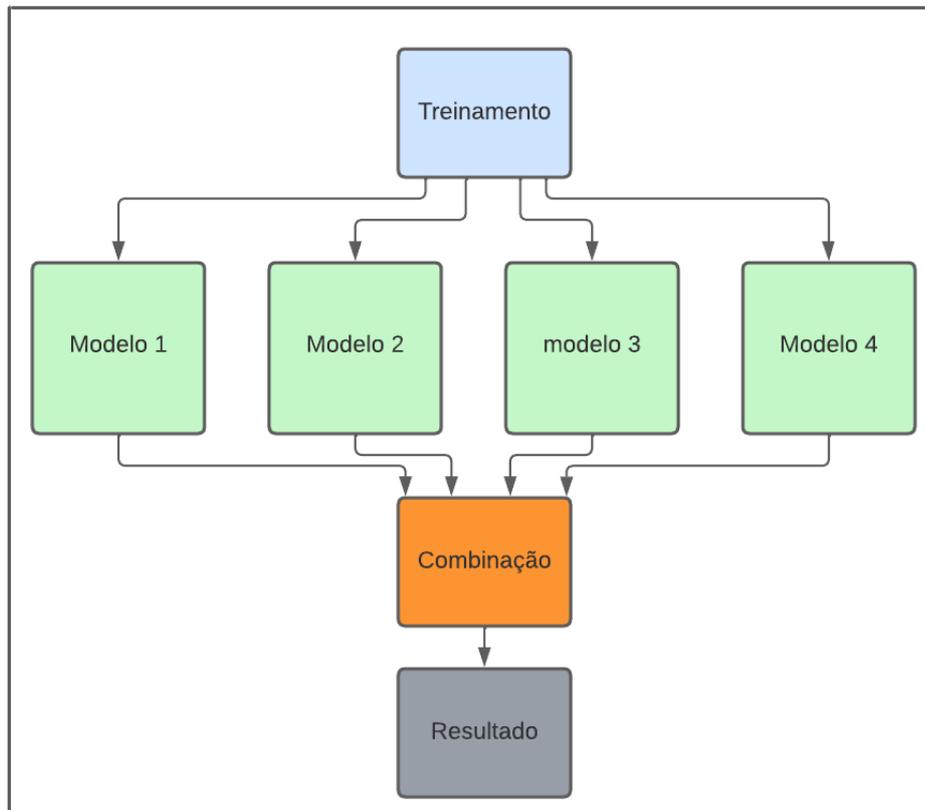


Figura 7: Esquema do ensemble.

Fonte: Elaborada pelos Autores (2022).

3.2. Detecção de Objetos

A detecção de objetos (DO) é uma tarefa que envolve o desenvolvimento de modelos computacionais e técnicas para detectar instâncias de objetos visuais que pertencem a determinadas classes (como animais, construções, carros ou árvores). O objetivo principal é trazer a informação de qual objeto está na imagem e onde ele se encontra (ZOU et al., 2019). A evolução dos recursos computacionais junto com o rápido desenvolvimento das técnicas de aprendizagem profunda, trouxeram um ânimo e um aumento nas possibilidades com o estudo dessa área. Cada vez mais a detecção de objetos tem sido aplicada a problemas do mundo real, isso inclui carros autônomos, visão de robôs, sistemas de segurança, detecção de veículos e pessoas, entre outros.

Um conceito importante usado neste trabalho é o termo *bounding box*, basicamente uma caixa que é desenhada sobre a imagem analisada que delimita a localização do objeto de

interesse e o identifica. As principais arquiteturas usadas para DO, assim como em outras áreas da computação visual, são baseadas nas redes neurais convolucionais.

3.2.1 Redes Neurais Convolucionais

Segundo (SULTANA; SUFIAN; DUTTA, 2018) “*Convolutional Neural Network (CNN)* ou *Rede Neural Convolucional* é um tipo especial de rede neural multicamada inspirada no mecanismo óptico de alguns seres vivos”. Tipicamente uma CNN possui blocos múltiplos de convolução e camadas de *pooling*, camadas totalmente conectadas e uma camada de saída como é mostrado na figura 8.

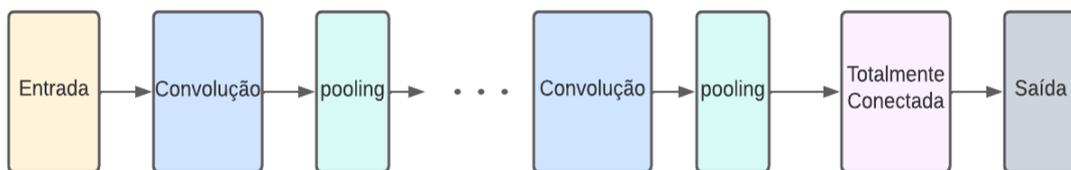


Figura 8: Arquitetura de uma rede neural convolucional.

Fonte: Elaborada pelos Autores (2022).

Cada camada funciona da seguinte maneira:

- **Camadas de Convolução:** é uma aplicação de um filtro chamado *kernel* na imagem de entrada que gera um mapa de ativação. Diferentes *kernels* geram diferentes mapas de ativação, portanto podendo extrair diferentes características da imagem. Essas camadas constituem a maioria dos blocos de uma CNN. A figura 9 ilustra a aplicação de um kernel em uma imagem de entrada.
- **Camadas de Pooling:** a ideia é diminuir o número de fatores considerados fazendo uma amostragem da matriz da camada anterior. Funciona de maneira parecida com a camada de convolução, pois se trata também de um filtro que anda pela matriz gerando uma saída reduzida. Existem diferentes tipos, como *max pooling*, que pega o maior valor da região e *average pooling* que pega o valor médio da região.
- **Camada Totalmente Conectada:** essa camada é ligada com todos os neurônios da camada anterior e “achata” essa entrada antes de passar para a camada de saída.

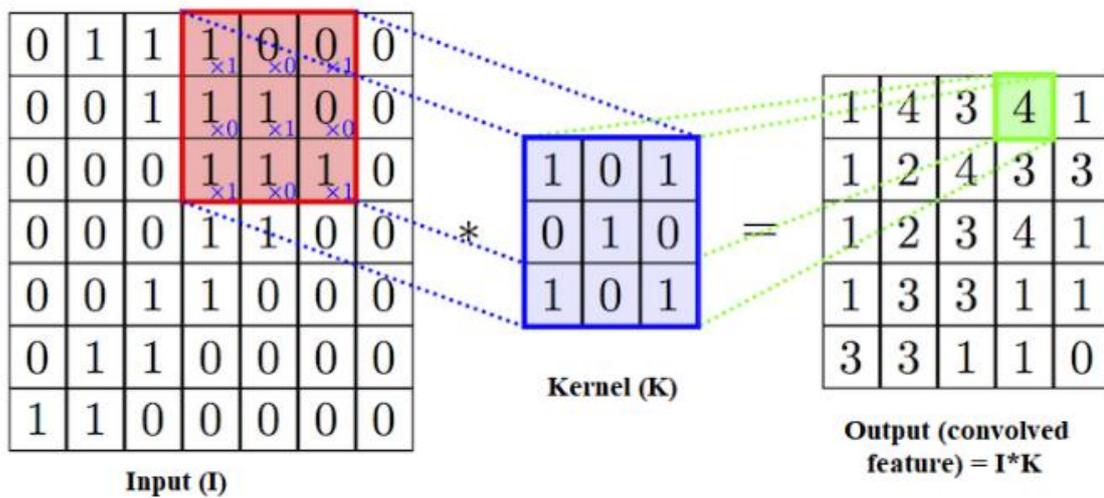


Figura 9: Camada de convolução.

Fonte: (SULTANA; SUFIAN; DUTTA, 2018).

3.2.2. Rede Neural Convolucional baseada em Região

Rede Neural Convolucional baseada em Região, em inglês *Region-based Neural Network* (R-CNN), trabalha com a imagem de entrada dividida em múltiplas regiões, então aplica convolução a cada região individualmente. Normalmente, essas redes são divididas em três fases. A primeira fase trabalha com a categorização das regiões da imagem, o mais comum para isso é usar o método da busca seletiva. A segunda fase é a extração de características, onde se configura os pesos e o número de camadas para cada rede convolucional. A última fase é quando a rede é treinada com imagens já rotuladas. Na fase de treino, inicialmente é feito um treino supervisionado usando uma CNN, depois é feito um ajuste fino nessa rede e por último a própria classificação do objeto da imagem (SHAH; KAPDI, 2017). Apesar de apresentar bons resultados, essa arquitetura ainda apresentava problemas, por isso algumas melhorias foram propostas.

3.2.3. Fast R-CNN e Faster R-CNN

A *Fast R-CNN* tem uma performance melhor do que a R-CNN padrão, pois ao invés de entrar com as regiões da imagem na CNN, a CNN é alimentada com a própria imagem para gerar um mapa. As regiões propostas são identificadas, transformadas em quadrados e seus

tamanhos são ajustados para servirem de entrada em uma camada totalmente conectada (GIRSHICK, 2015). Basicamente, a ideia é que não é necessário alimentar a CNN toda vez com milhares de regiões, mas sim realizar a convolução apenas uma vez por imagem, já que um mapa de recursos é gerado.

Ambas R-CNN e *Fast R-CNN* usam o algoritmo de busca seletiva para fazer a categorização das regiões da imagem, porém este método é lento e tende a prejudicar a performance da rede. Assim como na *Fast R-CNN*, a *Faster R-CNN* usa a imagem na CNN para gerar o mapa, porém ao invés de usar o algoritmo de busca seletiva, é usada uma nova rede especial com a tarefa de fazer a predição das regiões, esta rede é chamada de *Region Proposal Network*. Para gerar essas regiões um filtro passa pelo mapa na saída da CNN e gera uma matriz de dimensões menores, que serve de entrada para duas camadas fortemente conectadas, que são uma camada de *box-regression* e outra de *box-classification* (REN et al., 2017). No final são geradas k caixas delimitadoras para o mapa de características em questão e para cada uma dessas caixas é feito um cálculo para definir a probabilidade de ter um objeto dentro de seus limites. Caso a probabilidade de uma caixa seja maior que um certo valor definido, ela passa a ser considerada uma proposta de região. Essas propostas de região são alimentadas em uma *Fast R-CNN* que com a adição de outras camadas, como uma camada de *pooling*, camadas totalmente conectadas, uma camada *softmax* e um *regressor* finalizam a classificação dos objetos da imagem (HERRMANN, 2019). A figura 11 mostra a arquitetura básica de uma *Faster R-CNN*.

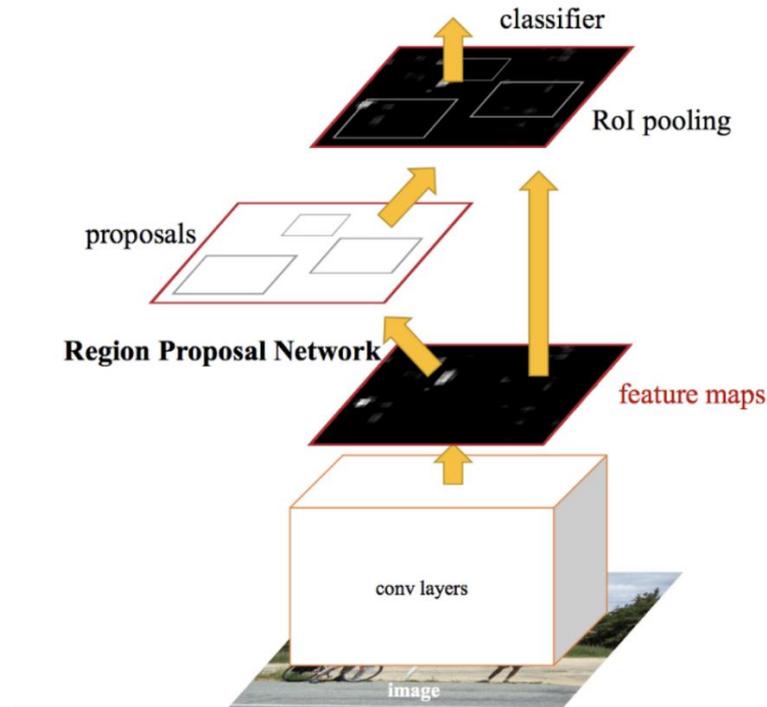


Figura 10: Estrutura da Faster RCNN.

Fonte: (REN et al., 2017).

3.2.4. *You Only Look Once* (YOLO)

Comumente chamado de YOLO, se trata de um algoritmo de detecção de objetos em tempo real e apresenta uma boa performance. O YOLO funciona de maneira diferente dos modelos apresentados anteriormente, ele não trabalha com a divisão da imagem em regiões para achar o objeto alvo. No YOLO uma única CNN faz a predição das *bounding boxes* e usa probabilidade para classificá-las (REDMON et al., 2016). Basicamente, a ideia do algoritmo é dividir a imagem em uma matriz $N \times N$ e gerar um certo número de *bounding boxes*, para cada uma destas caixas a rede define uma probabilidade de pertencer à determinada classe. Se o valor de probabilidade daquela *bounding box* estiver acima de um valor mínimo, ela é selecionada para localizar o objeto dentro da imagem. O YOLO foi criado por Joseph Redmon em maio de 2016 e com o passar dos anos foi recebendo atualizações no seu código com a ajuda de outros desenvolvedores. A versão mais recente lançada é o YOLOv5 que criou uma certa controvérsia pelo seu nome porque este novo modelo baseado no Pytorch é muito diferente da versão inicial no YOLO. O YOLOv5 é uma família de arquiteturas de detecção de objetos e modelos pré-treinados em conjunto com o COCO dataset, um conjunto de dados de larga escala de detecção, segmentação e legendagem de objetos (LIN et al., 2014). É um modelo rápido e fácil de se

utilizar e possui uma boa interface de visualização de resultados, métricas, pesos e outros parâmetros da rede.

3.3. Classificação de Imagens

Como o nome sugere o objetivo principal, dado uma imagem de entrada, é fazer a previsão correta de qual classe aquela imagem pertence, a partir de um conjunto de classes pré-definidas. Por mais que seja uma tarefa fácil para o ser humano, ensinar uma máquina a fazer isso pode ser bem difícil (LORENTE; RIERA; RANA, 2021). Assim como em toda área de visão computacional, com o avanço de técnicas de aprendizado profundo se tornou possível construir modelos mais robustos e mais eficientes, de forma que seja possível extrair características e padrões representativos das imagens.

Para tratar a classificação de imagens neste trabalho alguns modelos pré-treinados famosos na literatura foram testados:

- Xception: É uma arquitetura proposta por François Chollet, que trabalha com a ideia de uma rede neural convolucional inspirada pela Inception, onde os módulos Inception foram substituídos por convoluções separáveis em profundidade. Apresenta um ganho de performance em relação à Inception V3, pois faz uso mais eficiente dos parâmetros (AVERY et al., 2014).
- ResNet-50: Essa arquitetura foi proposta por Kaiming He, que percebeu que com o aumento da profundidade da rede, a acurácia se satura e então degrada rapidamente. Visando facilitar o treinamento de redes consideravelmente mais profundas que aquelas usadas anteriormente, ele propôs uma estrutura de aprendizagem residual e forneceu evidências empíricas de que estas redes são mais fáceis de otimizar (HE et al., 2016).
- Inception-ResNet-v2: É uma combinação entre a estrutura das redes Inception com as conexões das redes residuais. Além de evitar o problema de degradação de redes profundas, essa arquitetura ainda traz um treinamento mais rápido (SZEGEDY et al., 2017).
- EfficientNet: Essa família de arquiteturas foi proposta por Mingxing Tan e Quoc V. Le que, após estudar sistematicamente dimensionamento de modelos, identificaram que fazer um balanceamento cuidadoso de profundidade, largura e resolução, através do que eles chamam de coeficiente composto, pode resultar em um ganho de performance

(TAN; LE, 2019). Escalar essas três dimensões na prática significa aumentar a largura adicionando mais *feature maps* em cada camada, aumentar a profundidade aumentando o número de camadas e aumentar a resolução da imagem de entrada. A família *EfficientNet* vai da arquitetura *EfficientNet-B0* até a *EfficientNet-B7*, essa última sendo a principal para este trabalho.

- *EfficientNetV2*: Buscando ser um aprimoramento de sua versão anterior essa família de modelos consegue gerar modelos menores e realizar um treinamento mais rápido e eficiente. Isso através de uma técnica chamada Pesquisa de Arquitetura Neural combinada com dimensionamento (TAN; LE, 2021). Segundo Mingxing Tan e Quoc V. Le, esses modelos são treinados de maneira mais rápida do que modelos *state-of-art*, além de serem até 6,8 vezes menores.

Todos esses modelos foram inicialmente treinados no *ImageNet*. O projeto *ImageNet* é uma enorme base de dados que conta com mais de 1 milhão de imagens de acordo com o site oficial. Todas foram rotuladas à mão para indicar quais objetos estão presentes na imagem. De acordo com (RUSSAKOVSKY et al., 2015) o *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) acontece anualmente desde 2010 e acabou se tornando referência no reconhecimento de objetos em larga escala e, por isso, é tão amplamente usado para fazer o treinamento de modelos relacionados a computação visual.

4. Metodologia

Este capítulo apresenta a análise da base de dados do problema, trazendo estatísticas que exemplificam a complexidade do problema e a distribuição de suas classes. Também traz a abordagem deste trabalho diante do problema, e os raciocínios por trás dela.

A seção 4.1 analisa a base de dados utilizada por este trabalho. A 4.2 demonstra como foi realizada a detecção de anomalias nas radiografias, enquanto a seção 4.3 demonstra como as imagens da base de dados foram classificadas nas 4 classes descritas no tópico 4.1.

4.1. Base de dados

A base de dados utilizada neste trabalho foi publicada originalmente por uma colaboração entre o Banco de Imagens Médicas da Região de Valência (BIMCV), a Fundação para o Avanço da Saúde e Pesquisa Biomédica da Região de Valencia (FISABIO) e o Ministério de Inovação, Universidades, Ciência e Sociedade Digital (Generalitat Valenciana). Ela é uma junção de duas outras bases de dados, o BIMCV COVID-19+ e o *The RSNA International Radiology Database* (RICORD). A primeira, conforme analisada por (VAYÁ et al., 2020) é um banco de dados que contém imagens raio-x no formato DICOM, obtidas de 1311 pacientes não identificados. Para o diagnóstico da covid-19 nas radiografias, foram utilizados exames RT-PCR, e o auxílio de radiologistas. O segundo *dataset*, conforme analisado por (TSAI et al., 2021) contém 1000 radiografias do tórax, também no formato DICOM. Radiologistas realizaram diferentes classificações em relação às radiografias, que para o intuito deste trabalho e da base de dados principal, foram restritas à classificação da COVID-19.

Diante das bases de dados, profissionais de diversas universidades trabalharam em conjunto para adicionar valor a ambas as bases de dados. *Bounding boxes* foram desenhadas sobre as opacidades das radiografias de ambas as bases de dados e as imagens foram classificadas em 4 diferentes classes, unindo as bases de dados e padronizando-as (PARAS et al., 2021). Ambos os conceitos serão abordados mais à frente neste capítulo. A união de ambas as bases, fruto do trabalho destes profissionais e também objeto de estudo deste trabalho foi disponibilizada ao público pelo (KAGGLE, 2020).

A base de dados da plataforma *Kaggle*, por fim, é formada por 6054 radiografias do tórax, vindas de diferentes pacientes, porém com a possibilidade de 1 ou mais radiografias do

mesmo paciente. As imagens foram disponibilizadas no formato DICOM, um tipo de imagem usado por vários hospitais e laboratórios ao redor do mundo. O formato foi criado com o intuito de padronizar imagens, com radiografias, tomografias e ressonâncias, geradas pelos mais diversos equipamentos, possibilitando troca de informações entre médicos e pesquisadores. O formato DICOM é bem mais pesado em relação a outros formatos de imagem como o PNG e o JPG. Por exemplo, uma das imagens da base de dados, “3019399c31f4”, tem 18,63 megabytes de tamanho no formato DICOM, enquanto tem 0,11 megabytes no formato PNG e apenas 0,02 megabytes no formato JPG. Além disso, as imagens têm diferentes tamanhos, tornando-as difíceis de serem lidas em conjunto por uma máquina.

O *dataset* providenciado pelo Kaggle e utilizado por este trabalho, é dividido em 2 conjuntos. O primeiro deles é o *train_image_level*, que visa a detecção das anomalias. Ele é composto por 6334 radiografias no formato DICOM. Um exemplo pode ser visto na figura 12. As imagens são acompanhadas por duas *Strings* que as classificam, *boxes*, que traz a localização das anomalias, e *Label*, que traz a classificação delas. Cada imagem pode conter nenhuma, uma, ou múltiplas anomalias, e suas respectivas classificações.



Figura 11: Amostra de uma imagem no formato DICOM retirada diretamente da base de dados.

Fonte: Elaborada pelos Autores (2022).

O campo *boxes* indica a posição e dimensões de uma espécie de “caixa”, que contém a anomalia identificada pelo radiologista. É definida pelos parâmetros *x*, *y*, largura e altura. Um exemplo retirado do próprio *dataset* é “[{‘x’: 2729, ‘y’: 2181.33331, ‘width’: 948.00012, ‘height’:

604}]]”. Os campos x e y marcam as coordenadas iniciais como num plano cartesiano que parte do canto inferior esquerdo da imagem. Largura e altura são definidas a partir desse ponto. Imagens sem anomalias retornam *boxes* vazias. O campo *label* contém a classe correta daquela imagem para a *bounding box* correspondente. Retorna *Opacity* para cada uma das *bounding box*, e *none* para a ausência delas. A figura 13 exemplifica uma imagem com 2 *bounding boxes*, desenhadas com o auxílio do YOLOv5.

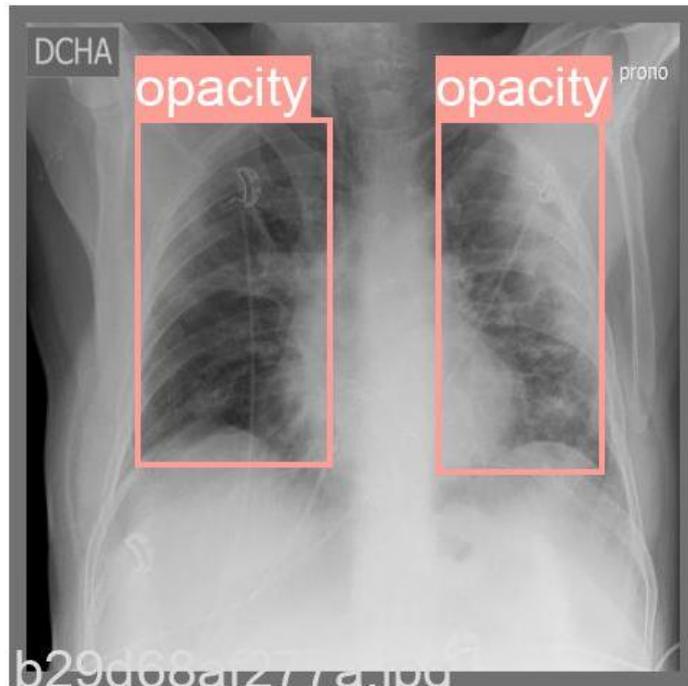


Figura 12: Amostra de uma imagem, convertida para o formato JPG, com 2 bounding boxes.

Fonte: Elaborada pelos Autores (2022).

O outro arquivo é o *train_study_level*, formado pela classificação de 6054 imagens do primeiro conjunto. Ele é dividido nos seguintes campos:

- Id;
- *Negative for Pneumonia*;
- *Typical Appearance*;
- *Indeterminate Appearance*;
- *Atypical Appearance*;

Cada um dos campos é binário e excludente, sendo classificado em 1 caso seja correspondente à imagem, e 0 caso contrário. Por exemplo, 0 1 0 0 exemplifica que a imagem tem uma *Typical Appearance* para Covid-19. O campo *Negative for Pneumonia* indica que o

paciente correspondente à radiografia está saudável, ao menos em relação a pneumonias. O campo *Typical Appearance* denota uma aparência típica e provável diagnóstico da Covid-19. O campo *Atypical Appearance* denota uma aparência atípica em relação à Covid-19, indicando que o paciente tem anomalias causadas por outra pneumonia. O campo *Indeterminate Appearance* denota uma aparência indeterminada, indicando que nem mesmo radiologistas profissionais foram capazes de classificar se as anomalias da imagem são causadas ou não pela Covid-19. Na figura 14, é possível ver algumas amostras de cada uma dessas classificações em imagens, com suas respectivas *bounding boxes*. Em geral, é possível notar que imagens classificadas como “*Negative for Pneumonia*” não tem *bounding boxes*, enquanto as imagens com elas se dividem entre as 3 outras classificações. Também é interessante notar a diferença do tamanho entre as imagens.

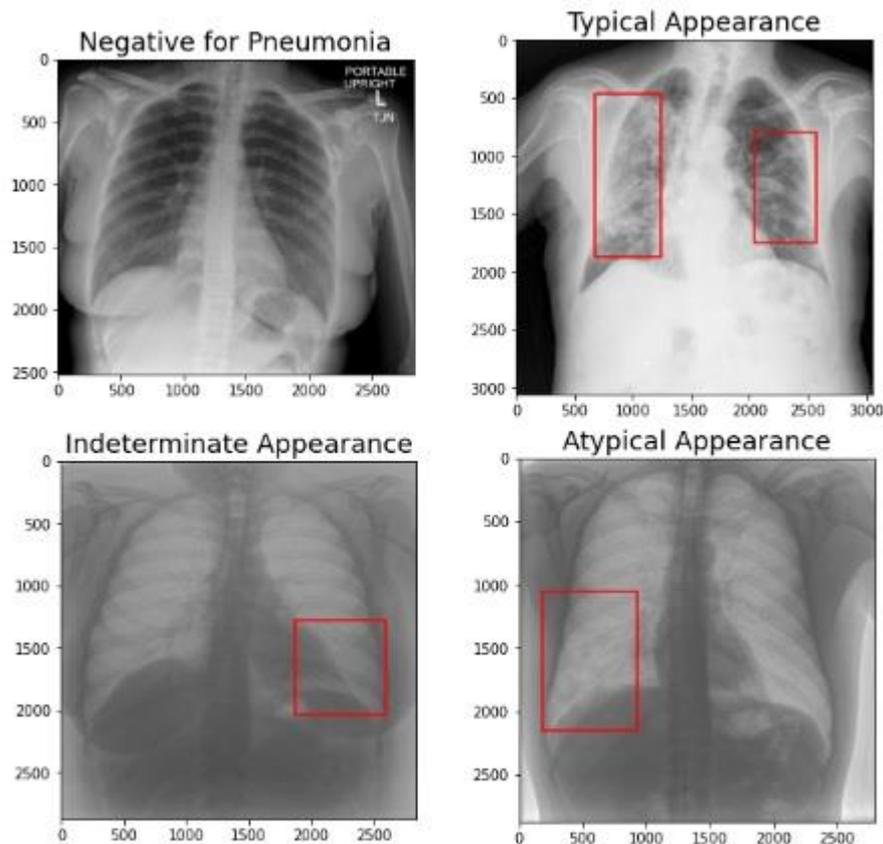


Figura 13: Amostra de cada uma das possíveis classes do banco de dados.

Fonte: Elaborada pelos Autores (2022).

Quanto à distribuição das classes, ela é exemplificada pela figura 15. Nela, pode-se notar que 2885 radiografias são correspondentes à *Typical Appearance*, cerca de 47% da base de dados, enquanto apenas 474 imagens são correspondentes ao *Atypical Appearance*, cerca de

8%. Essa pequena quantidade desta última classe, aliado à sua natureza errática, aumenta muito a complexidade do problema.

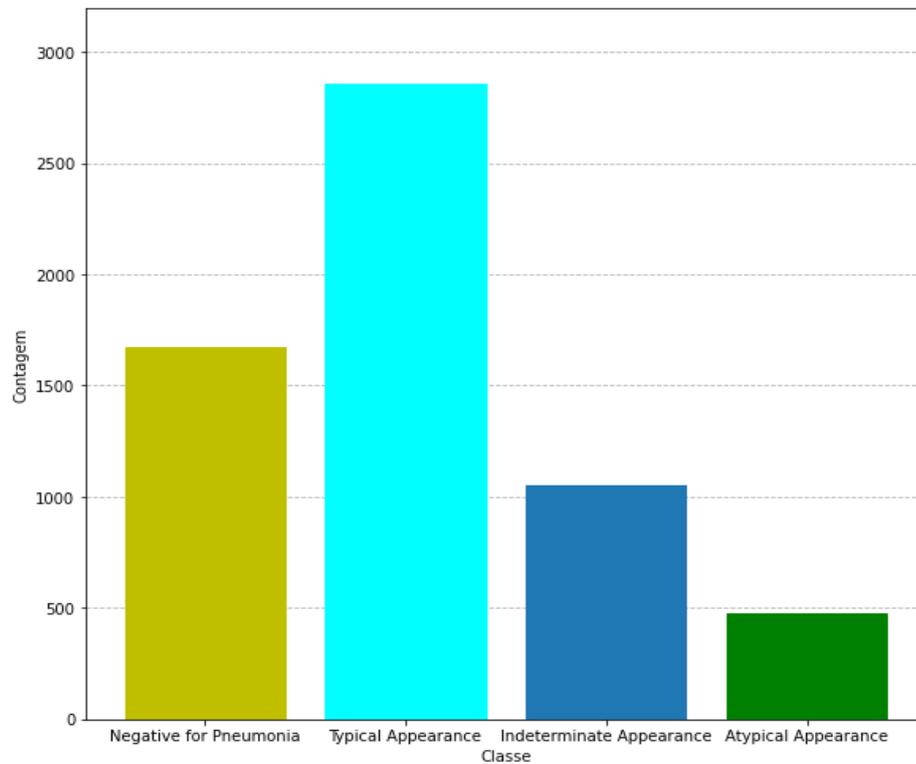


Figura 14: Gráfico que exemplifica o número de imagens em cada classe do study_level.

Fonte: Elaborada pelos Autores (2022).

4.2 Detecção de objetos

A detecção de objetos, conforme já exemplificado no referencial teórico, é uma técnica de visão computacional que permite identificar, localizar, e classificar objetos em uma imagem ou vídeo. Para a detecção de objetos deste trabalho, foi utilizado o YOLOv5, quinta versão do algoritmo de detecção de objetos YOLO. O YOLO emprega redes neurais convolucionais para detectar objetos em tempo real. Conforme a tradução de seu nome, apresenta vantagens em relação aos sistemas de classificação por requerer apenas uma propagação direta através da rede neural para detectar objetos, (Karimi, 2021) sendo dentre 100 e 1000 vezes mais rápido. E dentro dos sistemas de detecção, a quinta versão do YOLO se destaca.

Em comparação ao R-CNN, por exemplo, o algoritmo YOLOv5 tem uma velocidade de inferência maior, mais que o dobro, além de uma melhor precisão em detectar pequenos ou

distantes objetos (DWIVEDI, 2020). Novamente, em virtude da velocidade, as capacidades computacionais são aproveitadas de forma melhor, aumentando a precisão final. Em virtude das características, o YOLOv5 foi escolhido por esse trabalho para realizar a detecção dos objetos. A figura 15 exemplifica o fluxograma que demonstra os passos tomados para a metodologia da detecção, que serão detalhados a seguir.

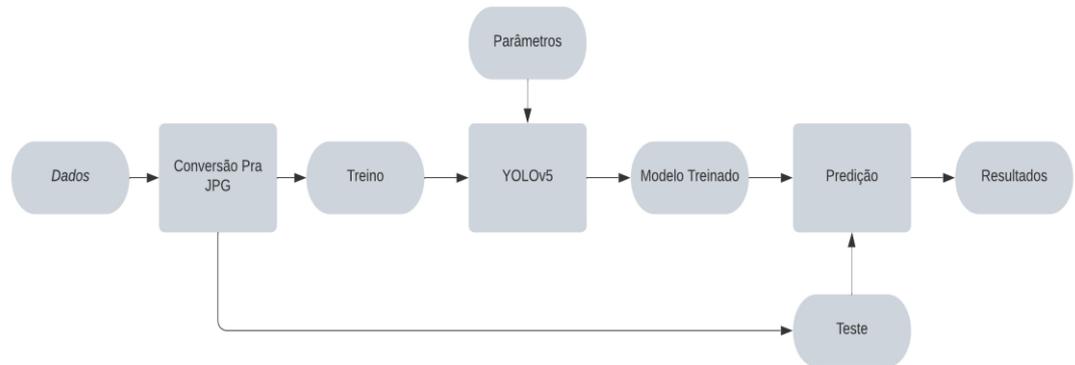


Figura 15: Fluxograma que exemplifica a metodologia da detecção.

Fonte: Elaborada pelos Autores (2022).

Para a aplicação do YOLOv5, foi realizada a conversão das imagens do formato DICOM para o formato JPEG, pelas razões descritas no ponto 5.1. Também foi realizado o redimensionamento destas mesmas imagens. Foram realizados teste com imagens de tamanhos de 256 e 512 pixels, com as últimas apresentando melhores resultados. Após o redimensionamento das imagens, foi realizada a divisão destas. No início, foi realizada uma divisão separando 10% das 6334 imagens para o conjunto de teste. Depois, foi realizada uma divisão separando os outros 90% em 80% para treino e 20% para validação. Ao fim, restou uma divisão de 4560 imagens para o conjunto de treino, 1140 para o conjunto de validação e 634 para o conjunto de teste. Depois, foi realizada a conversão das *Boxes* para o formato do YOLO. Como o detector utiliza valores mínimos e máximos de x e y, ao invés de um x e y auxiliados por altura e largura como a base de dados, é necessária a conversão.

Com o pré-processamento realizado, foi aplicado o algoritmo YOLOv5 no conjunto de treino. Ele foi rodado com um *Batch_Size* de 32, e 50 *Epochs*. O algoritmo foi processado com o auxílio de GPUs. Na figura 16, podemos ver amostras de algumas imagens do banco de dados, com suas opacidades desenhadas. Na figura 17, vemos estas mesmas amostras com suas

opacidades previstas pelo YOLOv5, ao lado de seus valores de confiança. O valor de confiança define a partir de qual porcentagem o algoritmo define a existência ou não existência da opacidade. Por exemplo, se o algoritmo acredita que haja uma chance de 40% de que uma imagem contenha uma opacidade, ele retorna com a classificação. Caso haja uma chance de 20%, ela é descartada, e assim sucessivamente.

Batch_Size é um termo que se refere ao número de amostras de treinamento usadas à cada iteração. *Epochs* é um valor referente ao número de iterações realizadas no treinamento.

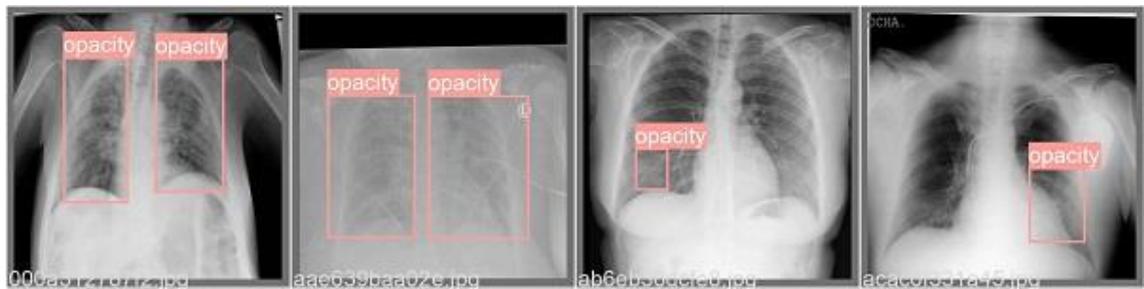


Figura 16: Amostra de imagens da base de dados, com suas opacidades desenhadas por radiologistas.

Fonte: Elaborada pelos Autores (2022).

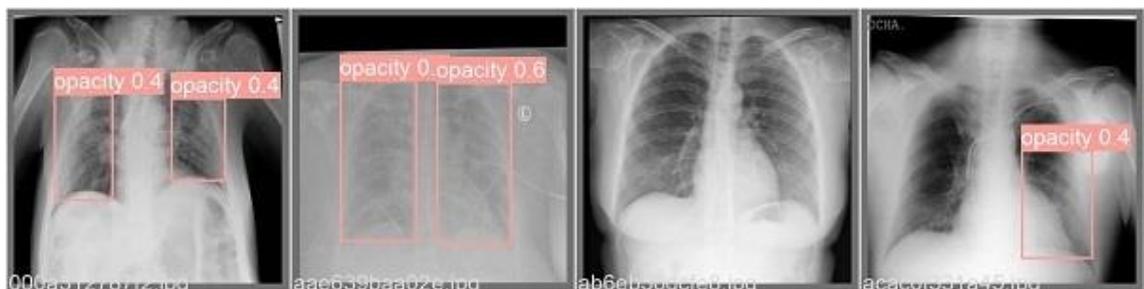


Figura 17: Amostra das mesmas imagens da figura 16, mas com suas opacidades previstas pelo YOLOv5.

Fonte: Elaborada pelos Autores (2022).

4.3 Classificação de Imagens

Diferente da classificação de objetos, na qual cada imagem pode ter vários objetos com várias classes diferentes, cada imagem pode ter apenas uma classe. Uma classe é essencialmente uma palavra. Aplicando esse conceito ao problema deste trabalho, torna-se necessário classificar cada uma das radiografias nas 4 classes descritas anteriormente.

Para a classificação das imagens foram utilizados modelos pré-treinados. Conforme analisado por (HAN et al., 2021), modelos pré-treinados no banco de imagens *ImageNet* têm sido utilizados em diversas áreas da visão computacional, incluindo classificação de imagens e detecção de objetos, apresentando bons resultados. Esses modelos, em comparação com uma rede neural feita do zero, têm a vantagem de serem treinados em ambientes com grande capacidade computacional, permitindo um bom ponto de partida para o treinamento, em comparação a um ponto de partida aleatório. O fluxograma da figura 18, que será descrito a seguir, exemplifica a metodologia utilizada na classificação das imagens.

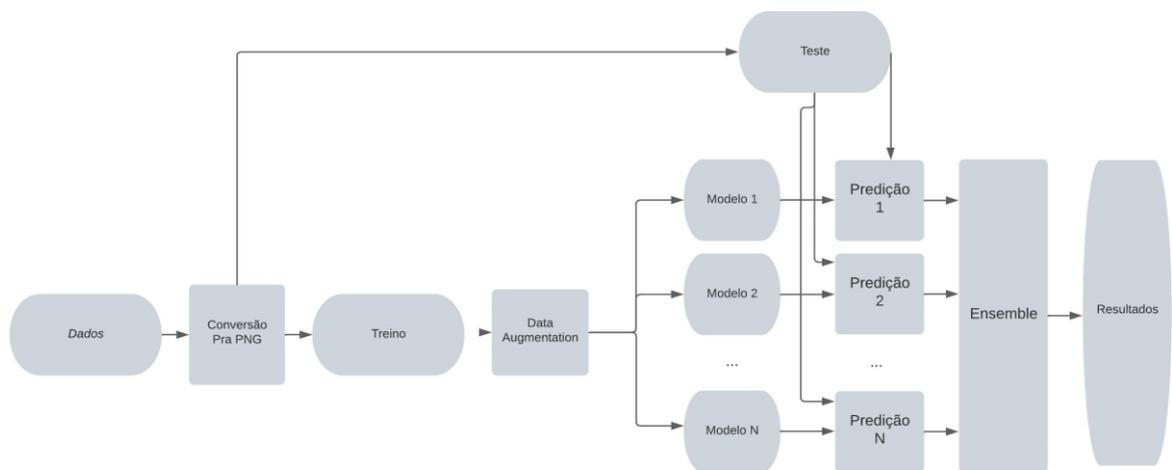


Figura 18: Fluxograma que exemplifica a metodologia da classificação.

Fonte: Elaborada pelos Autores (2022).

Como pré-processamento foi realizada a conversão das imagens *study_level* para o formato PNG, pelas razões descritas no ponto 4.1. Em seguida a base foi dividida novamente em 3 conjuntos, sendo 606 imagens para o conjunto de testes, enquanto as demais 5448 imagens foram divididas em 5 grupos, sendo 3 deles com 1090 imagens e 2 deles com 1089 imagens. Na figura 19, é possível notar como ficou a divisão entre as classes no conjunto de testes, mantendo uma proporção similar à figura 15. Dentre os 5 grupos, 1 subconjunto é selecionado para validação enquanto os demais para o treino. Isso é feito para cada um dos grupos, sendo este processo denominado *cross-validation*.

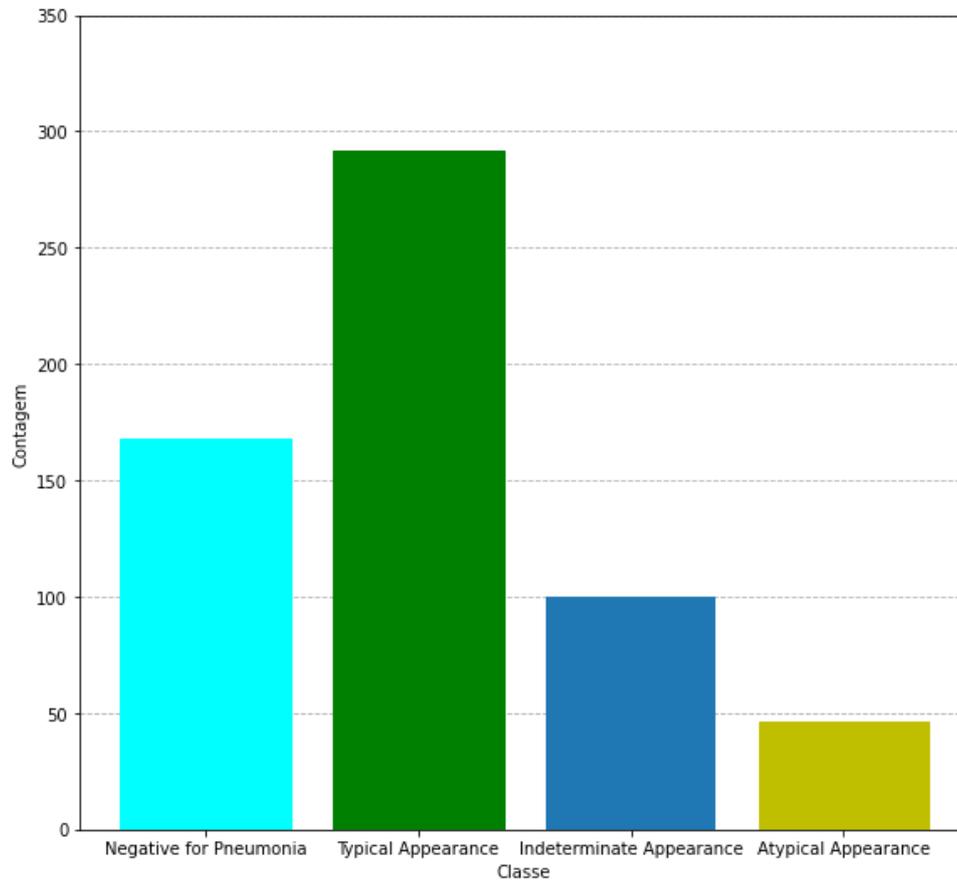


Figura 19: Distribuição de classes do conjunto de teste.

Fonte: Elaborada pelos Autores (2022).

Em seguida, é realizada a conversão das imagens para o tamanho específico de entrada de cada um dos modelos. Após a divisão dos conjuntos, foram gerados modelos pré-treinados diferentes. A cada passo, 10 modelos eram gerados, sendo que 1 desses modelos era escolhido com base na acurácia do conjunto de validação. Esse processo foi repetido para cada grupo de imagens, sendo escolhidos 5 modelos a cada execução. O processo total também foi repetido para cada modelo nomeado. Em cada um dos modelos, foi utilizada a técnica de *data augmentation*, descrita no capítulo anterior, com as imagens do próprio banco de dados, se utilizando do *flipping*, técnica que consiste em aumentar as imagens do banco de dados por replicá-las, adicionando rotações múltiplas de 90 graus. Em cada um dos modelos, também foram usados hiper-parâmetros responsáveis por ditar o modo como cada um dos modelos utilizados seriam treinados e gerados. Esses hiper-parâmetros serão detalhados na seção a seguir. Ao fim do processo, foi utilizado um *ensemble* dentre os modelos, que foi detalhado na seção 4.3.2. O algoritmo teve o auxílio de TPUs na execução de treinamentos e GPUs no *ensemble*.

4.3.1 Hiper-Parâmetros

Todos os modelos foram treinados de modo similar, à exceção do EffNet7, que teve seu número de *batches* reduzido para 32 devido à capacidade computacional. Um resumo dos hiper-parâmetros usados se encontra na tabela 1. Além disso, todos os modelos utilizaram os pesos provindos do *ImageNet*.

Tabela 1: Parâmetros de treinamento.

Fonte: Elaborada pelos Autores (2022).

Parâmetro	Valor
<i>batch size</i>	64
<i>learning rate</i>	0,001
<i>optimizer</i>	Adam
<i>loss function</i>	<i>categorical crossentropy</i>
<i>epochs</i>	20

Adam é o algoritmo de otimização utilizado para se atualizar os pesos, inicialmente carregados através do *ImageNet*, durante o treinamento dos modelos. O *Adam* mantém uma taxa de aprendizado por parâmetro que melhora o desempenho em problemas com gradientes esparsos, tornando-se adequado para problemas de visão computacional, (BROWNLEE,2017) como os deste trabalho.

Categorical_crossentropy é uma função de minimização usada para problemas de multi-classificação, como o deste trabalho, que retorna um valor decimal entre 0 e 1. Esse valor serve como a confiança do algoritmo em sua própria classificação ao classificar a imagem. (KUMAR,2020).

4.3.2 Ensemble

Um *ensemble* consiste em um conjunto de classificadores treinados individualmente (como redes neurais ou árvores de decisão) cujas previsões são combinadas ao classificar novas instâncias. Pesquisas anteriores mostraram que um conjunto geralmente é mais preciso do que qualquer um dos classificadores únicos no conjunto (OPITZ,1999).

O *Ensemble* é um componente importante deste trabalho. Ao realizar a combinação dos algoritmos baseado no histórico de acertos de cada modelo, é possível aumentar a acurácia geral dos modelos. Sua construção é mais simples, e sua execução é mais rápida, mas ainda assim, é um componente prioritário e fundamental neste trabalho.

Para a construção do *Ensemble* deste trabalho, foram utilizados os algoritmos EffNet7 e InceptionResNetV2. Para sua elaboração, foi realizado um algoritmo de força bruta se utilizando de todos os modelos gerados, descritos na abordagem. O algoritmo foi livre para remover ou duplicar modelos, quando necessário.

Como a construção do *Ensemble* independe da construção do modelo, é possível combinar outros modelos conforme o avanço da tecnologia ou aumento da base de dados. Para os resultados desse trabalho, foram utilizados apenas 4 componentes, pois combinações maiores não foram capazes de aumentar a acurácia do problema.

5. Resultados

Este capítulo tem o objetivo de apresentar e analisar os resultados dos diferentes modelos usados para classificação, constando também a aplicação do *ensemble*, e os resultados relacionados ao YOLO, se tratando da detecção.

5.1. Resultados dos modelos pré-treinados

Seguindo a divisão de dados que já foi detalhada anteriormente, os modelos usados foram avaliados no conjunto de testes. Todos estes modelos foram treinados com os mesmos parâmetros que foram descritos na tabela 1.

No total 10 modelos foram treinados e testados na base de dados sendo necessário adaptar nessas redes a camada de entrada, que possui o formato das imagens (512, 512, 3) sendo a última dimensão os valores RGB. E a camada de saída que é uma camada totalmente conectada com 4 neurônios de saída e usando a função de ativação *softmax*. As figuras 20, 21, 22, 23 e 24 mostram os gráficos de treinamento de cada um dos 10 modelos.

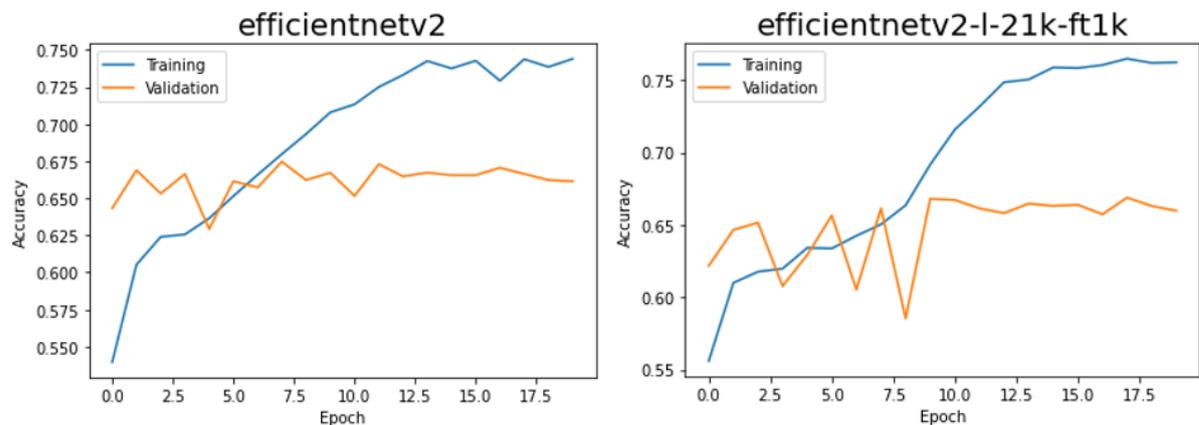


Figura 20: Gráficos do *efficientnetv2* e *efficientnetv2-l-21k-ft1k*.

Fonte: Elaborada pelos Autores (2022).

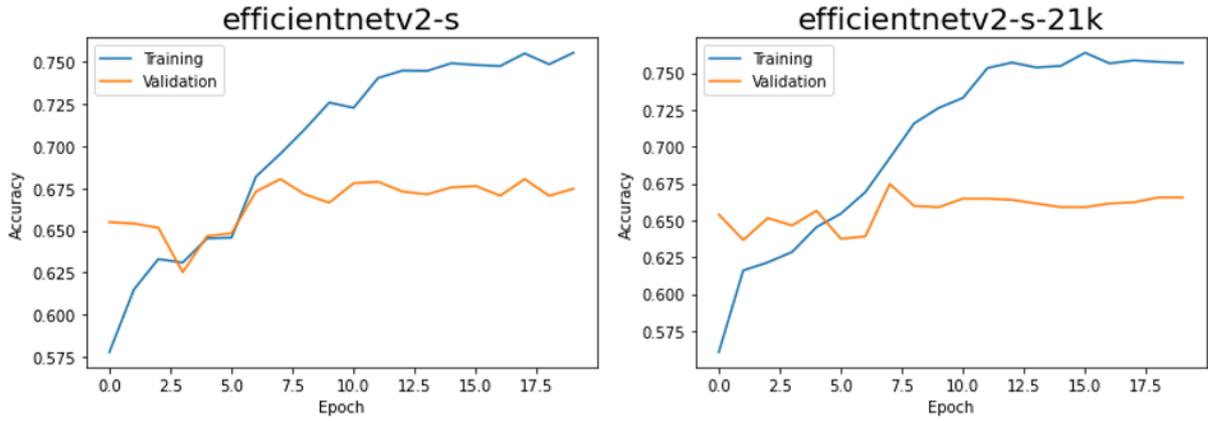


Figura 21: Gráficos do efficientnetv2-s e efficientnetv2-s-21k.

Fonte: Elaborada pelos Autores (2022).

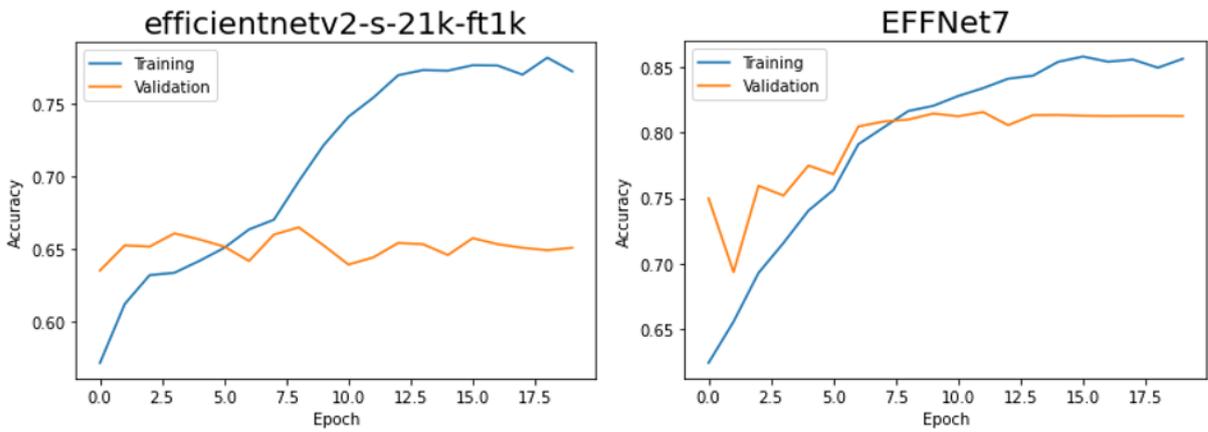


Figura 22: Gráficos do efficientnetv2-s-21k-ft1k e EFFNet7.

Fonte: Elaborada pelos Autores (2022).

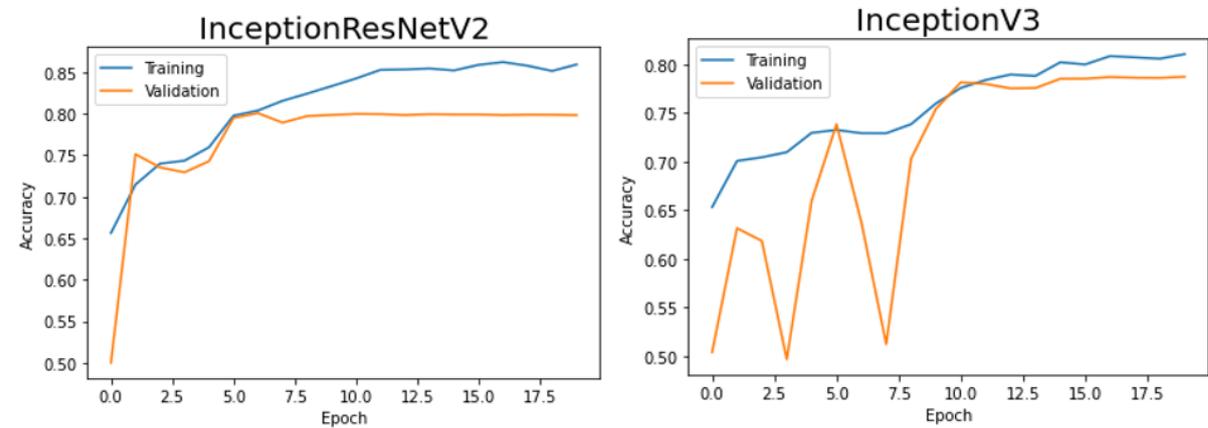


Figura 23: Gráficos do InceptionResNetV2 e InceptionV3.

Fonte: Elaborada pelos Autores (2022).

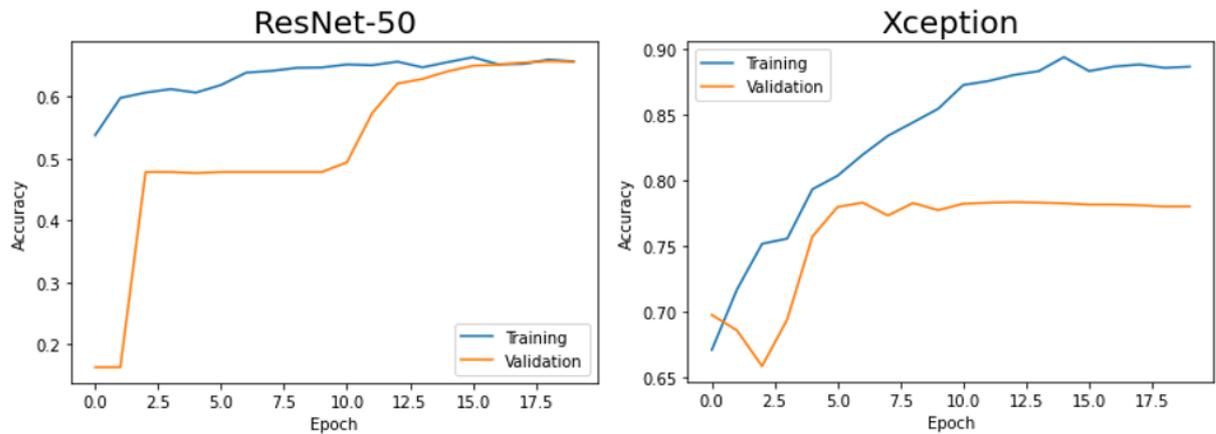


Figura 24: Gráficos do ResNet-50 e Xception.

Fonte: Elaborada pelos Autores (2022).

Após o treinamento completo, a melhor versão de cada modelo foi salva e utilizada para fazer a predição dos dados de teste. A tabela 2 mostra os resultados obtidos, contando a acurácia total e o tamanho de cada modelo gerado, onde a acurácia total é a razão dos verdadeiros positivos pelo número total de predições.

Tabela 2: Resultados dos modelos.

Fonte: Elaborada pelos Autores (2022).

Modelo	Acurácia	Tamanho (MB)
Resnet50	68,81%	283
efficientnetv2-l-21k	72,11%	1410
efficientnetv2-l-21k-ft1k	71,45%	1410
efficientnetv2-s	71,62%	244
efficientnetv2-s-21k	71,94%	244
efficientnetv2-s-21k-ft1k	72,94%	244
EfficientNetB7	67,98%	769
InceptionV3	68,81%	262
Xception	68,31%	250
InceptionResNetV2	67,32%	654

Um comportamento em comum em todos modelos é a dificuldade de classificar corretamente as classes em menor quantidade (*indeterminate appearance* e *atypical appearance*). Isso porque o desbalanceamento dos dados causou uma dificuldade de aprendizado nessas classes. De maneira geral, as arquiteturas da família *efficientnetv2* obtiveram os melhores resultados. Através da matriz de confusão, mostrada na figura 25, é possível analisar de maneira mais detalhada os resultados obtidos pelo modelo *efficientnetv2-s-21k-ft1k* que conseguiu a maior acurácia entre todos. Considera-se um número para cada classe da seguinte maneira:

- 0: *Negative for Pneumonia*;
- 1: *Typical Appearance*;
- 2: *Indeterminate Appearance*;
- 3: *Atypical Appearance*;

A coluna principal contabiliza os acertos do modelo, deixando claro como as classes 0 e 1 foram as mais corretamente previstas, também levando em conta que são as que mais aparecem na base de dados. Por outro lado, o pior resultado está na classificação da classe 2 com apenas 5 acertos. Fazendo uma leitura das linhas e colunas é possível perceber como as classes 0 e 1 tiveram poucos falsos negativos e uma quantidade maior de falsos positivos. Esse é um comportamento causado pelo desbalanceamento dos dados, que faz com que o modelo tenha uma tendência a classificar mais vezes os rótulos mais frequentes na base de dados. De maneira geral este comportamento é refletido em todos modelos, que sofrem para classificar as classes 2 e 3.

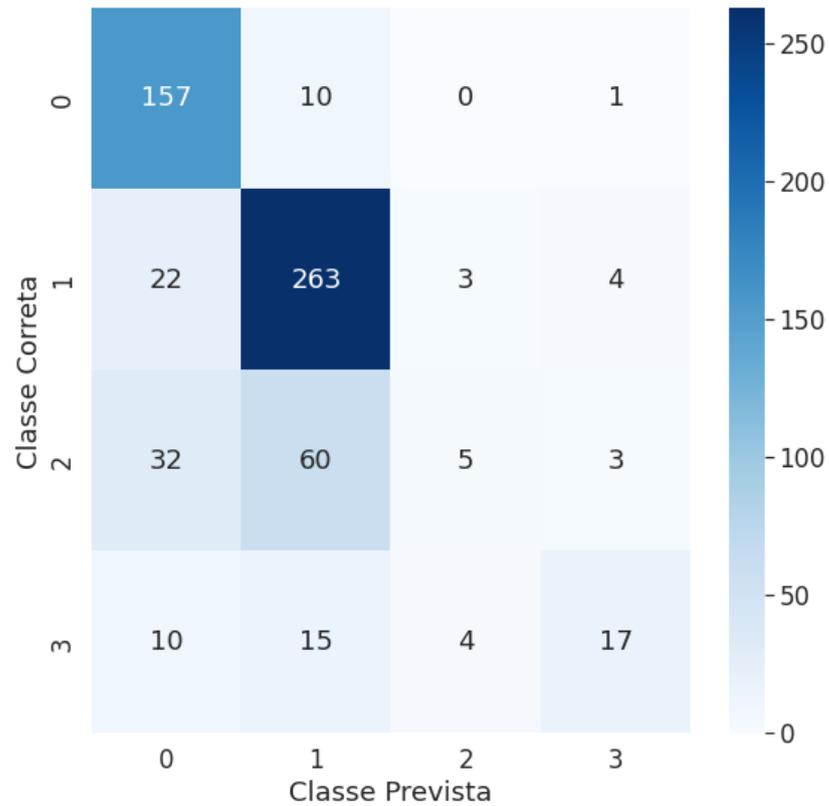


Figura 25: Matriz de Confusão efficientnetv2-s-21k-ft1k.

Fonte: Elaborada pelos Autores (2022).

5.2. Resultados com ensemble

Após testar diferentes combinações entre os modelos, como explicado na metodologia, o melhor resultado foi obtido em uma combinação do EfficientNetB7 com InceptionResNetV2, atingindo uma acurácia de 76,23%. A figura 26 é a matriz de confusão após uso do ensemble. Apesar de um aumento nos acertos da classe 2, a dificuldade de acertar as classes 2 e 3 se manteve.

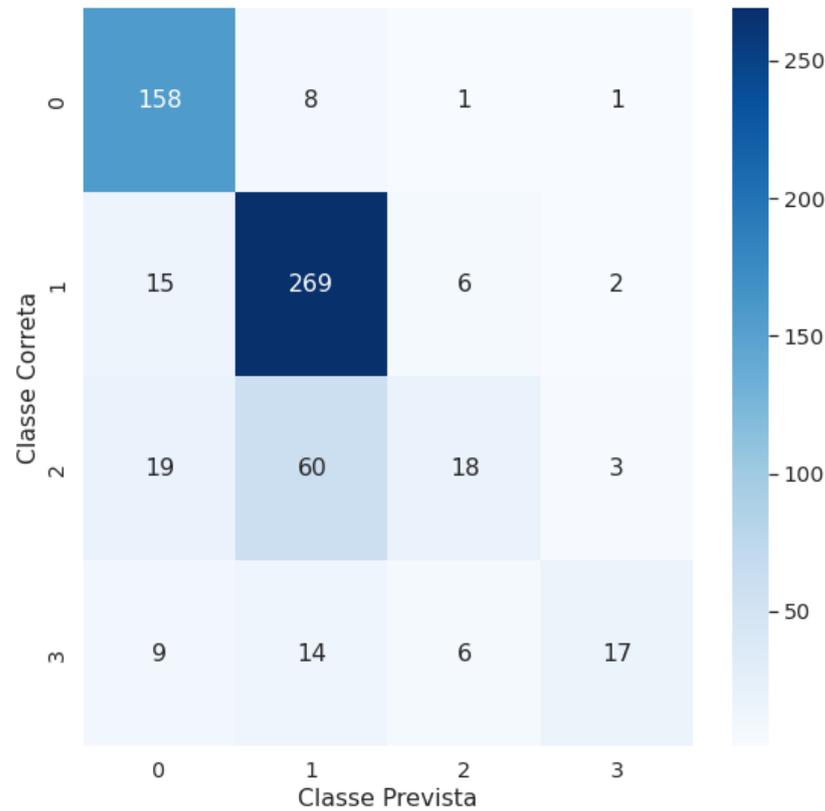


Figura 26: Matriz de Confusão final.

Fonte: Elaborada pelos Autores (2022).

Outras métricas podem ser úteis para entender melhor os resultados de cada classe, como *recall* e precisão, que podem ser definidos pelas equações (5) e (6) respectivamente. O *recall* mede, dentre as imagens que pertencem a determinada classe, quantas foram corretamente classificadas. É uma medida importante principalmente por se tratar de um trabalho relacionado à detecção de doenças. A classe *Typical Appearance*, por exemplo, indica que o paciente está infectado com Covid-19, por isso é importante ter um valor alto de *recall*.

$$recall = \frac{VP}{VP + FN}$$

(5)

Já a medida de precisão trabalha com o peso dos falsos positivos nas classes. Os valores dessa métrica foram mais próximos entre as classes.

$$precisão = \frac{VP}{VP + FP}$$

(6)

As figuras 27 e 28 são representações gráficas dos valores de *recall* e precisão dos resultados gerados com o *ensemble*.

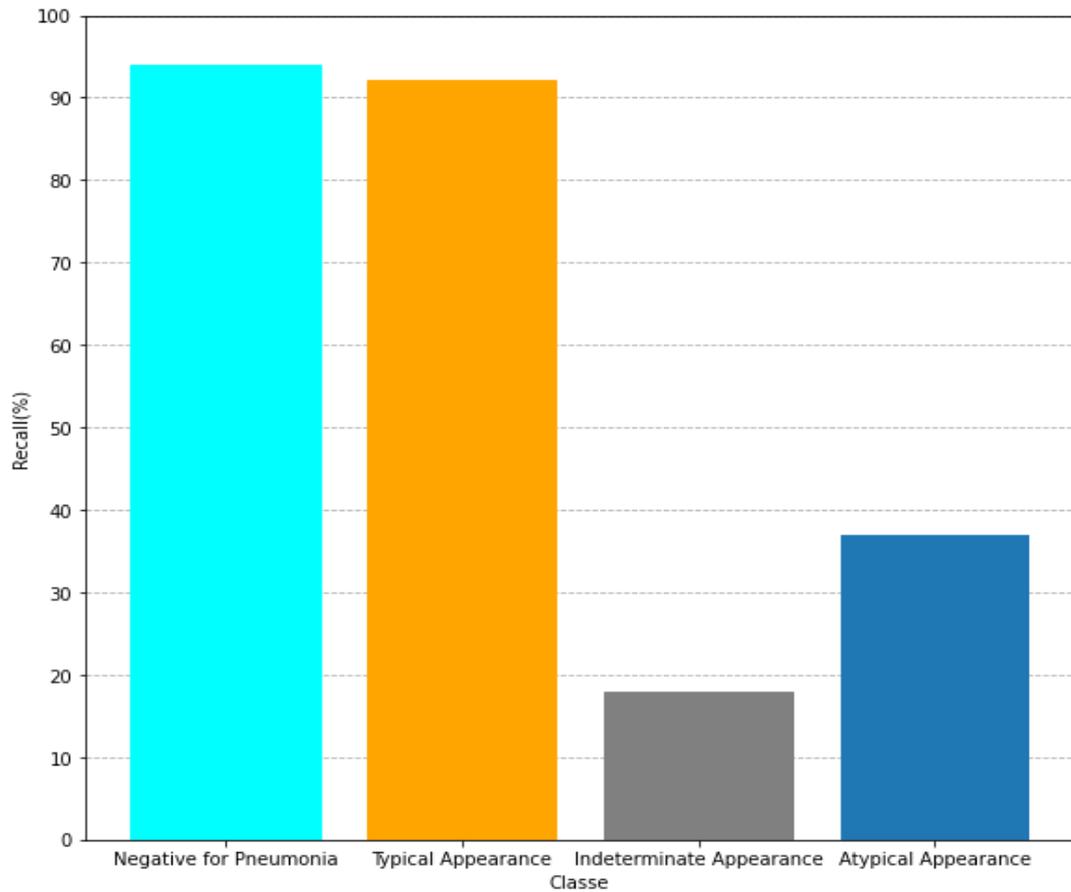


Figura 27: *Recall* das classes.

Fonte: Elaborada pelos Autores (2022).

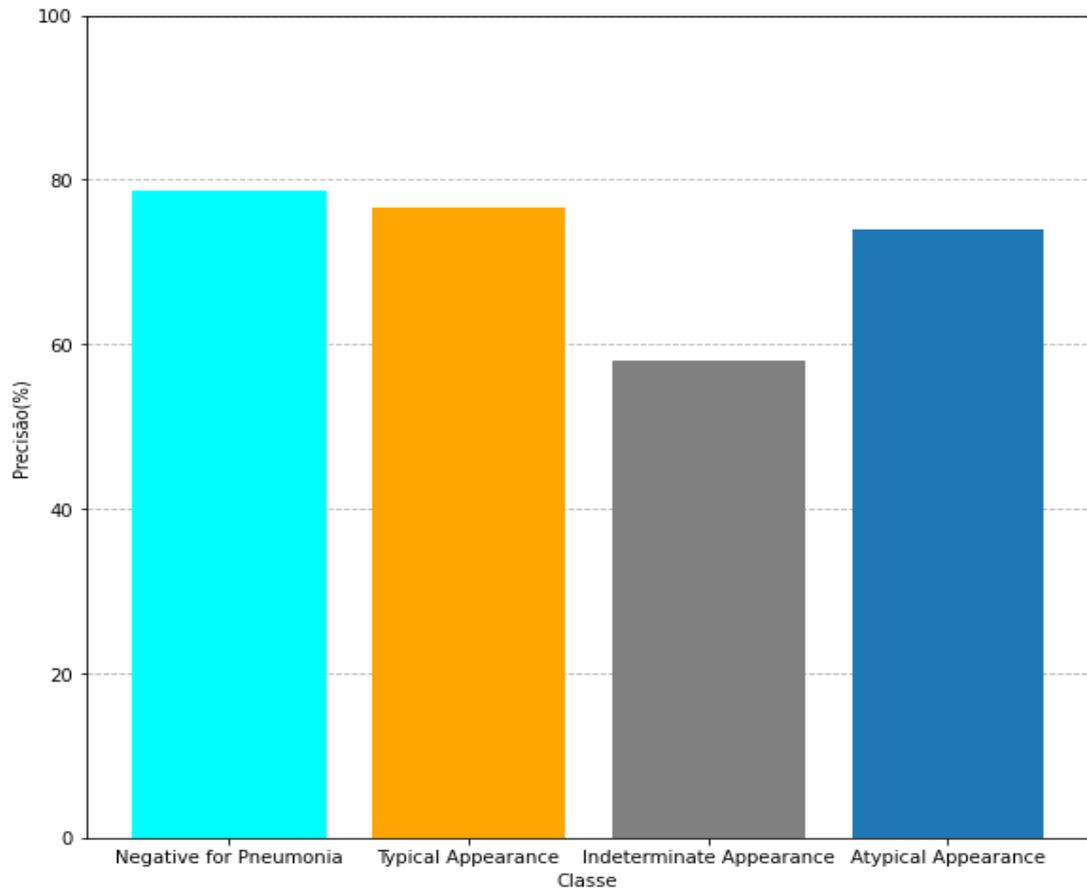


Figura 28: *Recall* das classes.

Fonte: Elaborada pelos Autores (2022).

Os valores de *recall* acima de 90% nas classes 0 e 1 são ótimos resultados, indicando uma quantidade baixa de falsos negativos. O contrário, porém, ocorre com as classes 2 e 3 que apresentam valores bem baixos de *recall*. Se tratando da precisão, não há uma variação tão grande entre as classes, ficando entre 80% e 58%.

5.3. Resultados do YOLO

Como explicado na metodologia a parte de detecção de opacidades trabalha com dois tipos de imagens, aquelas que possuem alguma opacidade detectada e as que não possuem. As figuras 29 e 30 trazem os gráficos gerados durante o treinamento usando o YOLO.

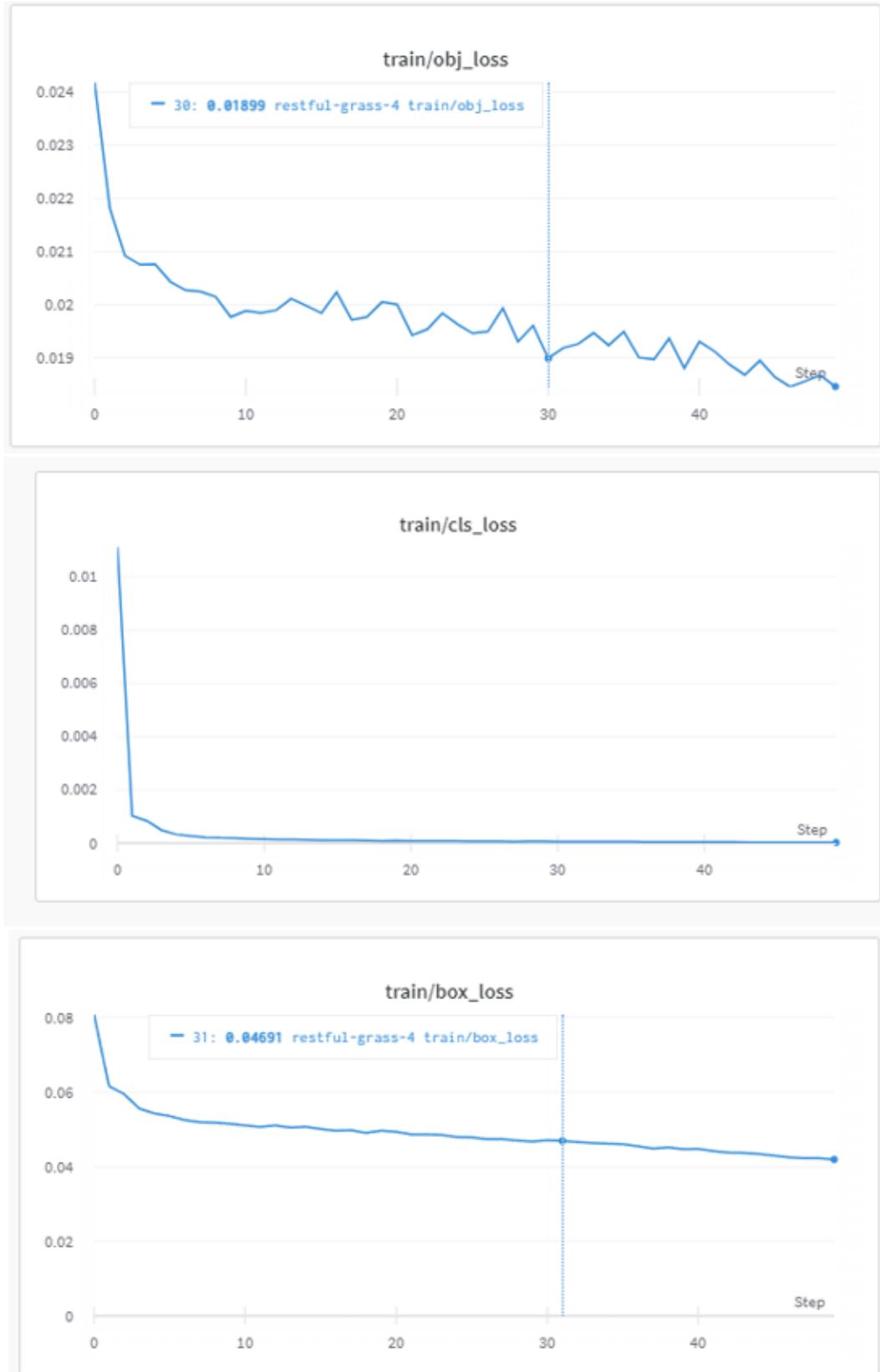


Figura 29: Gráficos de *loss* do treino.

Fonte: Elaborada pelos Autores (2022).

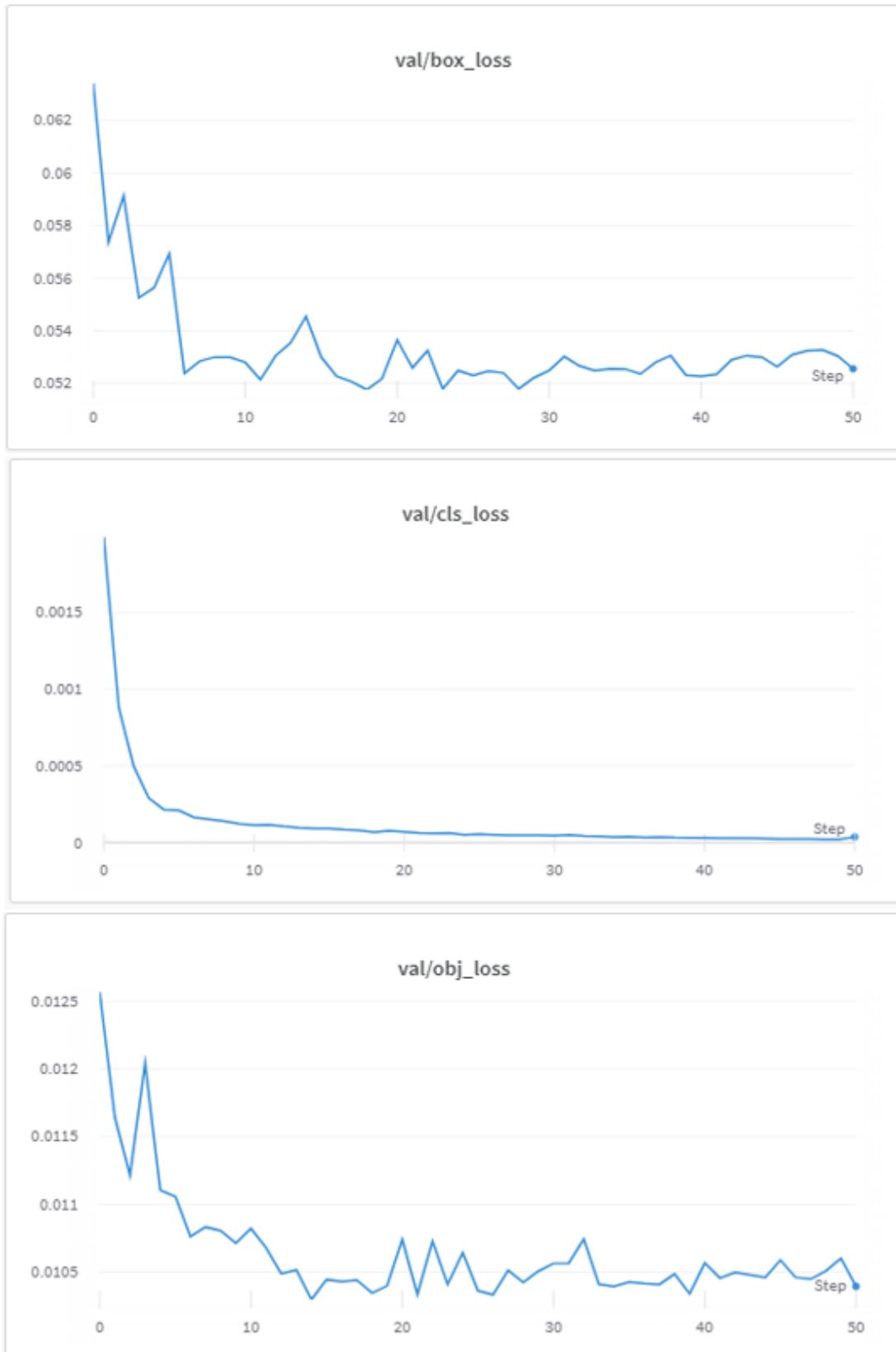


Figura 30: Gráficos de *loss* do treino.

Fonte: Elaborada pelos Autores (2022).

Apesar de se tratar de um problema de detecção, é possível analisar os resultados como uma classificação, já que os rótulos das imagens são conhecidos. Se uma ou mais opacidades é detectada, a imagem é um exemplo da classe *opacity*, caso contrário considera-se classe *none*. A matriz de confusão representada na figura 31 mostra os acertos e erros do uso do YOLO nos dados de teste. A acurácia final obtida foi de 75,23%, contabilizando 477 acertos em um total de 634 imagens.

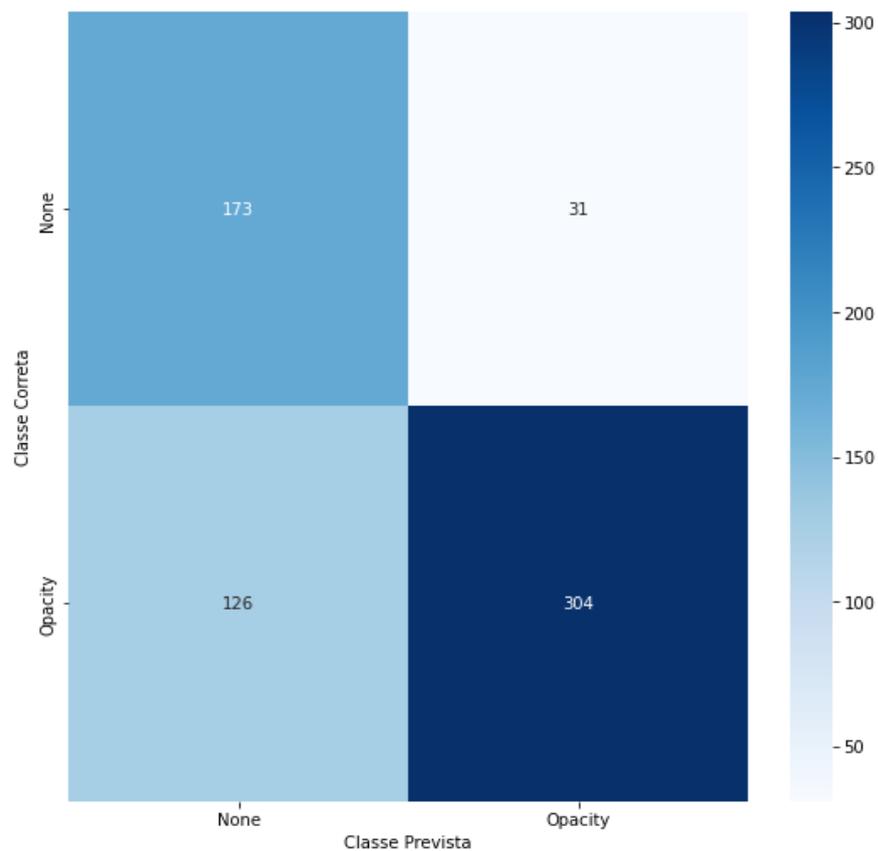


Figura 31: Matriz de confusão da detecção.

Fonte: Elaborada pelos Autores (2022).

A próxima imagem, figura 32, traz 4 exemplos de predições. Uma imagem onde nenhuma opacidade foi detectada, e casos onde 1, 2 e 3 opacidades foram localizadas em imagens de teste.

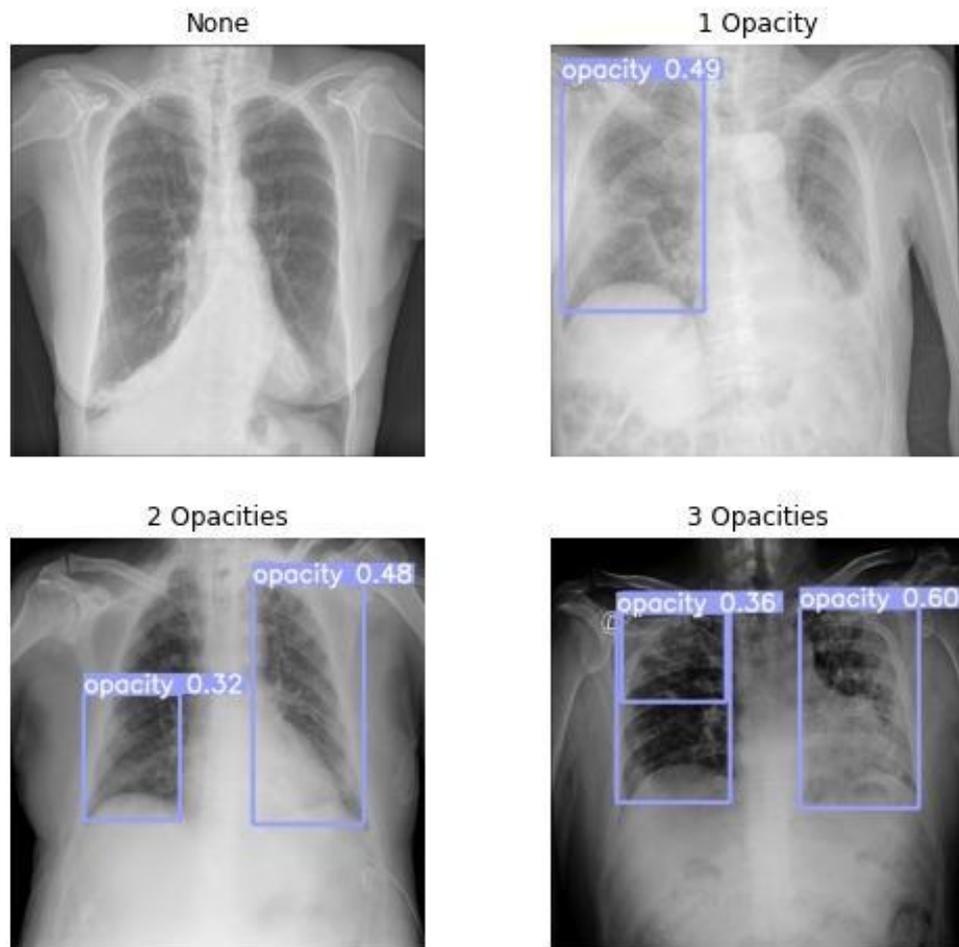


Figura 32: Imagens de teste com predições.

Fonte: Elaborada pelos Autores (2022).

A *bounding box* localiza na imagem uma possível opacidade, que pode ser uma anomalia encontrada no pulmão do paciente. Os valores que aparecem nas imagens são os graus de confiança de cada detecção. Resultados visuais são bem úteis, pois facilitam a análise do especialista que vai ser responsável por chegar a uma conclusão.

6. Análises e Conclusões

Este capítulo traz algumas análises finais sobre os modelos usados e as técnicas aplicadas, finalizando com uma conclusão sobre os resultados obtidos e sugestões para trabalhos futuros.

6.1. Modelos Pré-Treinados

Em média os modelos levavam em torno de 2 horas para serem treinados o que dificultava em ter uma agilidade maior para rodar diferentes modelos com diferentes configurações e técnicas, considerando também que o tempo de computação era limitado. Entre os modelos testados os pertencentes à família *efficientnetv2* tiveram os melhores resultados, como foi mostrada na tabela 2 comparando seus valores de acurácia. De maneira geral, o grande diferencial entre os modelos era a capacidade de classificar corretamente as classes mais incomuns.

6.2. Ensemble e Data Augmentation

A combinação de todos os modelos gerou um resultado abaixo do esperado, porém ainda assim trouxe um aumento na acurácia final. Dessa forma, o *ensemble* foi uma técnica importante para a melhoria do resultado.

A utilização do *data augmentation* melhorou a acurácia de boa parte dos modelos, porém em alguns casos específicos o contrário ocorreu, isso provavelmente depende de como o modelo foi projetado. Um fenômeno que esta técnica pareceu minimizar, após uma análise dos resultados, é o *overfitting*. Modelos quando rodados sem *data augmentation* apresentaram uma queda maior da acurácia dos dados de testes em relação à acurácia da validação, isso porque o modelo se adaptou demais aos dados de treino e tem dificuldades de generalização para novas imagens.

6.3. Conclusões

Os resultados alcançados não foram os esperados, porém são satisfatórios dadas as condições. Levando em conta que a ideia é que sirva de apoio ao médico responsável, uma acurácia de 76,23% combinada à análise profissional é um resultado expressivo, contando também que a inferência de uma nova imagem leva aproximadamente 1 segundo. O fato da base de dados ser desbalanceada acabou sendo uma questão que dificultou alcançar taxas de

acerto mais altas. Uma tentativa de nivelar a quantidade de imagens de cada classe com a classe de menor quantidade foi feita, porém não resultou em uma melhora justamente porque diminui consideravelmente o número de imagens para se treinar o modelo. Uma abordagem que poderia ter sido seguida era trabalhar com 3 classes, removendo as imagens rotuladas como *indeterminate appearance*, o que facilitaria atingir resultados um pouco melhores. Porém essa não foi a forma decidida para se tratar o problema neste trabalho. No caso do uso do YOLO o resultado obtido dificilmente seria melhor, comparando o resultado deste trabalho com os resultados de alguns participantes da competição que forneceu o *dataset* de estudos pode-se notar que os níveis de acurácia encontrados foram inferiores.

Em linhas gerais, é possível alcançar resultados melhores através de alguns fatores:

- Ter disponível um maior número de imagens, já que de forma análoga ao aprendizado humano quanto mais dados de qualidade para treinar, maior a tendência do modelo de melhorar seus resultados.
- Uma base mais balanceada na sua distribuição de classes.
- Ter disponível maior capacidade de computação para possibilitar testes de modelos mais complexos em tempo hábil sem que esgote os recursos de *hardware*.

6.4. Trabalhos Futuros

Algumas sugestões para trabalhos futuros que os autores deste trabalho propõem são:

- Construir uma interface amigável e intuitiva para facilitar o uso do modelo por usuários da área da saúde.
- Ampliar o sistema para que possa fornecer dados históricos e a plotagem de gráficos para facilitar análises específicas.

REFERÊNCIAS

ABBAS, A.; ABDELSAMEA, M. M.; GABER, M. M. Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network. **Applied Intelligence**, v. 51, n. 2, p. 854–864, 1 fev. 2021.

AVERY, K. R. et al. Fatigue Behavior of Stainless Steel Sheet Specimens at Extremely High Temperatures. **SAE International Journal of Materials and Manufacturing**, v. 7, n. 3, p. 560–566, 2014.

BOUSLAMA, A.; LAAZIZ, Y.; TALI, A. Diagnosis and precise localization of cardiomegaly disease using U-NET. **Informatics in Medicine Unlocked**, v. 19, 1 jan. 2020.

BROWNLEE, J.. Gentle Introduction to the Adam Optimization Algorithm for Deep Learning. **Machine Learning Mastery**, 2017. Disponível em: <<https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/#:~:text=Adam%20is%20a%20replacement%20optimization,sparse%20gradients%20on%20noisy%20problems.>>>. Acesso em: 6 Jul. 2022.

CHAUHAN, R.; GHANSHALA, K. K.; JOSHI, R. C. **Convolutional Neural Network (CNN) for Image Detection and Recognition**. ICSCCC 2018 - 1st International Conference on Secure Cyber Computing and Communications. **Anais...Institute of Electrical and Electronics Engineers Inc.**, 2 jul. 2018

CHOUHAN, V. et al. A novel transfer learning based approach for pneumonia detection in chest X-ray images. **Applied Sciences (Switzerland)**, v. 10, n. 2, 2020.

Coronavirus (COVID-19). **NHSINFORM**. 2022. Disponível em: <<https://www.nhsinform.scot/illnesses-and-conditions/infections-and-poisoning/coronavirus-covid-19/coronavirus-covid-19/>> Acesso em 10 de Agosto de 2022.

DASH, A. K.; MOHAPATRA, P. A Fine-tuned deep convolutional neural network for chest radiography image classification on COVID-19 cases. **Multimedia Tools and Applications**, v. 81, n. 1, p. 1055–1075, 1 jan. 2022.

FERREIRA, NICOLA. Saiba como funciona o PCR, o exame que detecta o novo coronavírus **UOL**. 17 de março de 2020. Disponível em

<<https://www.uol.com.br/vivabem/noticias/redacao/2020/03/17/saiba-como-funciona-o-pcr-o-exame-que-detecta-o-novo-coronavirus.htm>> Acesso em 9 de Agosto de 2022.

FLORIANO, I. et al. Accuracy of the polymerase chain reaction (PCR) test in the diagnosis of acute respiratory syndrome due to coronavirus: A systematic review and meta-analysis. **Revista da Associação Médica Brasileira**, v. 66, n. 7, p. 880–888, 2021

GAO, K. et al. Dual-branch combination network (DCN): Towards accurate diagnosis and lesion segmentation of COVID-19 using CT images. **Medical Image Analysis**, v. 67, p. 101836, 2021.

GARCEZ, G.; SOUSA, B.; LUÍS -MA, S. **Deep Learning para a Detecção e Classificação de Pneumonia por Radiografias do Tórax**. [s.l: s.n.].

GESSER, C.. O que é DICOM e quais são os seus benefícios? Píxeon, 2019. Disponível em: <<https://www.píxeon.com/blog/o-que-e-dicom-e-quais-sao-seus-beneficios/>>. Acesso em: 14 Out. 2021.

GIRSHICK, R. Fast R-CNN. **Proceedings of the IEEE International Conference on Computer Vision**, v. 2015 Inter, p. 1440–1448, 2015.

HAGHANIFAR, A. et al. COVID-CXNet: Detecting COVID-19 in frontal chest X-ray images using deep learning. **Multimedia Tools and Applications**, 2022.

HAN, X. et al. Pre-trained models: Past, present and future. **AI Open**, v. 2, p. 225–250, 2021.

HASOON, J. N. et al. COVID-19 anomaly detection and classification method based on supervised machine learning of chest X-ray images. **Results in Physics**, v. 31, n. December 2020, p. 0–7, 2021.

HE, K. et al. Deep residual learning for image recognition. **Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, v. 2016-Decem, p. 770–778, 2016.

HERRMANN, G. C. Análise e aplicação de redes R-CNN para identificação de cobras peçonhentas a partir de imagens. **Universidade Estadual de Londrina**, 2019.

HSU, C. C.; ZHUANG, Y. X.; LEE, C. Y. Deep fake image detection based on pairwise learning. **Applied Sciences (Switzerland)**, v. 10, n. 1, 1 jan. 2020.

IBGE. Instituto Brasileiro de Geografia. IBGE, 2022. Disponível em

<<https://www.ibge.gov.br/>>. Acesso em 9 de agosto de 2022.

IBRAHIM, D. M.; ELSHENNAWY, N. M.; SARHAN, A. M. Deep-chest: Multi-classification deep learning model for diagnosing COVID-19, pneumonia, and lung cancer chest diseases. **Computers in Biology and Medicine**, v. 132, p. 104348, 2021.

SIIM-FISABIO-RSNA COVID-19 Detection. **Kaggle**, 2021. Disponível em: <<https://www.kaggle.com/c/siim-covid19-detection>>. Acesso em: 9 de agosto de 2022.

KUMAR, A.. Keras – Categorical Cross Entropy Loss Function. Vitalflux, 2020. Disponível em: <<https://vitalflux.com/keras-categorical-cross-entropy-loss-function/>>. Acesso em: 8 Jul. 2022.

KUMAR, E. P.; SHARMA, E. P. Artificial Neural Networks-A Study. v. 2, n. 2, p. 143–148, 2014.

LAKHANI, P. et al. The 2021 SIIM-FISABIO-RSNA Machine Learning COVID-19 Challenge: Annotation and Standard Exam Classification of COVID-19 Chest Radiographs. Disponível em: <osf.io/532ek>. LIN, T. Y. et al. Microsoft COCO: Common objects in context. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 8693 LNCS, n. PART 5, p. 740–755, 2014.

LARGUI, Nathália. PCR, antígeno... Qual teste de covid-19 fazer? Saiba preços, diferenças e indicações de cada um. **Valor Investe**. São Paulo, 14/01/2022. Disponível em <<https://valorinveste.globo.com/mercados/brasil-e-politica/noticia/2022/01/14/qual-teste-de-covid-19-fazer-saiba-precos-diferencas-e-indicacoes-de-cada-um.ghtml>>. Acesso em 16 de agosto de 2022.

LORENTE, Ò.; RIERA, I.; RANA, A. Image Classification with Classic and Deep Learning Techniques. 2021.

LU, Z. et al. Multiobjective Evolutionary Design of Deep Convolutional Neural Networks for Image Classification. **IEEE Transactions on Evolutionary Computation**, v. 25, n. 2, p. 277–291, 2021.

LV, W.; WANG, X. **Overview of Hyperspectral Image Classification** **Journal of Sensors** Hindawi Limited, , 2020.

MACHADO, VÍ. et al. Using Deep Learning Transformer Networks to Identify Symptoms Associated with COVID-19 on Twitter. **Notas Técnicas**, v. 12, n. 1, p. 1–10, abr. 2022.

MACLIN, R. Popular Ensemble Methods : An Empirical Study Popular Ensemble Methods : An Empirical Study. **Journal of artificial intelligence research**, v. 11, n. July, p. 169–198, 2016.

MANOHARAN, S. IMAGE DETECTION, CLASSIFICATION AND RECOGNITION FOR LEAK DETECTION IN AUTOMOBILES. **Journal of Innovative Image Processing**, v. 1, n. 02, p. 61–70, 4 dez. 2019.

MENZIES, T. et al. Ensembles of Learning Machines. **Sharing Data and Models in Software Engineering**, n. December 2013, p. 239–265, 2015.

OPITZ, D.; SHAVLIK, J. Actively Searching for an Effective Neural-Network Ensemble. *Connection Science*, 8, pp. 337-353.

Painel Coronavírus. **CORONAVÍRUS // BRASIL**, 2022. Disponível em <<https://covid.saude.gov.br/>>. Acesso em 9 de agosto de 2022.

PANAHI, A. H.; RAFIEI, A.; REZAEI, A. Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information. n. January, 2020.

PICOLLI, D., 2022. O que é DICOM e quais são os seus benefícios para radiologia. **Pixeon**, 2022. Disponível em: <<https://www.pixeon.com/blog/o-que-e-dicom-e-quais-sao-seus-beneficios/>> Acesso em 7 de agosto de 2022.

PRAY, Ian et al. Performance of an Antigen-Based Test for Asymptomatic and Symptomatic SARS-CoV-2 Testing at Two University Campuses - Wisconsin, September-October 2020. **PubMed**. 1 de Janeiro de 2021. Disponível em: <<https://pubmed.ncbi.nlm.nih.gov/33382679/>> Acesso em 9 de Agosto de 2022.

CAETANO, Simone. **Pro Exame**, 2022. Disponível em <<https://proexame.com/tabela-preco/>>. Acesso em 16 de Agosto de 2022.

RAJARAMAN, S. et al. Iteratively Pruned Deep Learning Ensembles for COVID-19 Detection in Chest X-Rays. **IEEE Access**, v. 8, p. 115041–115050, 2020.

REDDY ALLUGUNTI, V. Breast cancer detection based on thermographic images using machine learning and deep learning algorithms Healthcare View project Breast cancer detection based on thermographic images using machine learning and deep learning algorithms.

Computer Science, v. 4, n. 1, p. 49–56, 2022.

REDMON, J. et al. You only look once: Unified, real-time object detection. **Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, v. 2016- Decem, p. 779–788, 2016.

REN, S. et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 39, n. 6, p. 1137–1149, 2017.

RITCHIE, Hannah et al. Coronavirus (COVID-19) Deaths. **Our World in Data**. 14 de agosto de 2022. Disponível em <<https://ourworldindata.org/covid-deaths>>. Acesso em 16 de agosto de 2022.

SHAH, M.; KAPDI, R. Object detection using deep neural networks. **Proceedings of the 2017 International Conference on Intelligent Computing and Control Systems, ICICCS 2017**, v. 2018- Janua, p. 787–790, 2017.

SHARMA, S.; SHARMA, S.; ANIDHYA, A. Understanding Activation Functions in Neural Networks. **International Journal of Engineering Applied Sciences and Technology**, v. 4, n. 12, p. 310–316, 2020.

SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on Image Data Augmentation for Deep Learning. **Journal of Big Data**, v. 6, n. 1, 2019.

STEFANINI, Ariela. CNN Approach for CT Scan Exam Classification. **Sigmae**. Alfnas, 2021. Disponível em <<https://publicacoes.unifal-mg.edu.br/revistas/index.php/sigmae/index>> Acesso em 29 de agosto de 2022.

SULTANA, F.; SUFIAN, A.; DUTTA, P. Advancements in image classification using convolutional neural network. **Proceedings - 2018 4th IEEE International Conference on Research in Computational Intelligence and Communication Networks, ICRCICN 2018**, p. 122–129, 2018.

SZEGEDY, C. et al. Inception-v4, inception-ResNet and the impact of residual connections on learning. **31st AAAI Conference on Artificial Intelligence, AAAI 2017**, p. 4278–4284, 2017.

TAN, M.; LE, Q. V. EfficientNet: Rethinking model scaling for convolutional neural networks. **36th International Conference on Machine Learning, ICML 2019**, v. 2019- June, p. 10691–10700, 2019.

- TAN, M.; LE, Q. V. EfficientNetV2: Smaller Models and Faster Training. 2021.
- TSAI, E. B. et al. The RSNA International COVID-19 Open Radiology Database (RICORD). **Radiology**, v. 299, n. 1, p. E204–E213, 1 abr. 2021.
- VAYÁ, M. DE LA I. et al. BIMCV COVID-19+: a large annotated dataset of RX and CT images from COVID-19 patients. 1 jun. 2020.
- WANG, D. et al. **FocalMix: Semi-Supervised Learning for 3D Medical Image Detection**. [s.l: s.n.].
- WANG, J. et al. Prior-Attention Residual Learning for More Discriminative COVID-19 Screening in CT Images. **IEEE Transactions on Medical Imaging**, v. 39, n. 8, p. 2572–2583, 2020.
- WANKER, M. S. Research Paper on Basic of Artificial Neural Network. **International Journal on Recent and Innovation Trends in Computing and Communication**, v. 2, n. 1, p. 96–100, 2014.
- WONG, S. C. et al. Understanding Data Augmentation for Classification: When to Warp? **2016 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2016**, 2016.
- YING, X. An Overview of Overfitting and its Solutions. **Journal of Physics: Conference Series**, v. 1168, n. 2, 2019.
- ZAIDI, S. S. A. et al. A Survey of Modern Deep Learning based Object Detection Models. 24 abr. 2021.
- ZHANG, J. et al. Viral Pneumonia Screening on Chest X-Rays Using Confidence-Aware Anomaly Detection. **IEEE Transactions on Medical Imaging**, v. 40, n. 3, p. 879–890, 2021.
- ZHOU, L. et al. A Rapid, Accurate and Machine-Agnostic Segmentation and Quantification Method for CT-Based COVID-19 Diagnosis. **IEEE Transactions on Medical Imaging**, v. 39, n. 8, p. 2638–2652, 2020.
- ZOU, Z. et al. Object Detection in 20 Years: A Survey. p. 1–39, 2019.