

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SÃO PAULO**

**GUSTAVO TELES BONIFÁCIO
JEFFERSON DIAS**

**IDENTIFICAÇÃO DE COVID-19 E PNEUMONIA VIRAL BASEADO EM IMAGENS
DIGITAIS DE RAIO-X**

**BARRETOS
2021**

GUSTAVO TELES BONIFÁCIO
JEFFERSON DIAS

**IDENTIFICAÇÃO DE COVID-19 E PNEUMONIA VIRAL BASEADO EM IMAGENS
DIGITAIS DE RAIO-X**

Trabalho apresentado ao Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, campus Barretos como requisito parcial para obtenção do grau de Tecnólogo em Tecnologia em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Dr. Jones Mendonça de Souza

BARRETOS
2021

GUSTAVO TELES BONIFÁCIO

JEFFERSON DIAS

**IDENTIFICAÇÃO DE COVID-19 E PNEUMONIA VIRAL BASEADO EM IMAGENS
DIGITAIS DE RAIOS-X**

Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia - IFSP - *campus* Barretos, como parte das exigências para a obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas.

BANCA EXAMINADORA

23 de março de 2021

Prof. Dr. Jones Mendonça de Souza

Instituto Federal de Educação, Ciência e
Tecnologia de São Paulo – IFSP

Prof. Dr. Tiago Alexandre Dócusse

Instituto Federal de Educação, Ciência e
Tecnologia de São Paulo – IFSP

Prof. Msc. José Ricardo Ferreira Cardoso

Instituto Federal de Educação, Ciência e
Tecnologia de São Paulo – IFSP

LOCAL

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo
Barretos, SP

Aos nossos pais e familiares por suas lutas constantes para nos proporcionar um conforto e uma estabilidade, abrindo portas para executar estudos e trabalhos. Ao grupo de amigos Highlands pelo apoio e explicações sobre conteúdos relacionados a esse trabalho, e ao apoio de Beatriz Santos Francisco.

AGRADECIMENTOS

A nosso orientador, Prof. Dr. Jones Mendonça de Souza, pelo acompanhamento e revisão, apoio, orientação e amizade.

Aos integrantes da banca examinadora por disponibilizar seu tempo e atenção a este trabalho.

RESUMO

Até o momento, cerca de 2 milhões de pessoas no mundo foram vítimas fatais da COVID-19. O Brasil computa mais de 200 mil mortes e 9 milhões de casos positivos. Alguns exames como: tomografia computadorizada e radiografias de tórax são capazes de diagnosticar a doença e contribuir para tomadas de decisões ao longo do tratamento. No entanto, algumas doenças respiratórias como as diversas pneumonias virais podem apresentar exames similares, prejudicando o diagnóstico médico e consequentemente o tratamento. Com base neste problema, este trabalho propõe o uso das redes neurais convolutivas para a classificação da COVID-19 e pneumonia viral utilizando imagens de radiografias de tórax. Para isso, foram realizados experimentos de treinamento e classificação com as redes *InceptionV3* e *ResNet152*, utilizando-se duas bases de dados: *Covid-19 Radiography DataSet* e *Covid-19 Image Dataset*. Os resultados experimentais apontam que a rede *InceptionV3* demonstra ser a mais promissora para o diagnóstico das doenças, atingindo 99.82% de acurácia nos testes com amostras do mesmo grupo da base de dados de treinamento. Na hipótese de *overfitting*, foi realizada uma validação cruzada com amostras de outra base de dados, na qual foi obtida 98.07% de acurácia.

Palavras-chave: COVID-19, Pneumonia Viral, Radiografia de Tórax, Rede Neural Convolutacional.

ABSTRACT

So far, about 2 million people worldwide have been fatal to COVID-19. Brazil has more than 200,000 deaths and 9 million positive cases. Some tests such as CT scans and chest X-rays can diagnose the disease and contribute to decision-making throughout treatment. However, some respiratory diseases such as various viral pneumonias may present similar tests, impairing the medical diagnosis and consequently treatment. Based on this problem, this study proposes the use of convoluted neural networks for the classification of COVID-19 and viral pneumonia using chest X-ray images. For this, training and classification experiments were carried out with inceptionv3 and resnet152 networks, using two databases: Covid-19 Radiography DataSet and Covid-19 Image Dataset. The experimental results indicate that the InceptionV3 network proves to be the most promising for the diagnosis of diseases, reaching 99.82% accuracy in tests with samples from the same group from the training database. In the case of overfitting, a cross-validation was performed with samples from another database, in which 98.07% accuracy was obtained.

Keywords: Covid-19, Viral Pneumonia, Thorax Radiography, Convolutional Neural Network.

LISTA DE FIGURAS

Figura 1 - Radiografia de Tórax de Paciente com Covid-19.....	16
Figura 2 - Amostras de radiografia Pulmonar.....	17
Figura 3 - Exemplo de Rede Neural Profunda.....	25
Figura 4 - Rede Neural Convolucional LeNet-5.....	27
Figura 5 - Exemplo de Max-Pooling.....	29
Figura 6 - Amostras de Radiografias de Tórax	32
Figura 7 - Exemplo de Matriz de Confusão com Classes.....	41
Figura 8 - Treinamento da InceptionV3	42
Figura 9 - Treinamento da ResNet152	43
Figura 10 - Treinamento da ResNet152 Transfer Learning	45
Figura 11 - Treinamento da InceptionV3 Transfer Learning	46

LISTAS DE TABELAS

Tabela 1 - Elementos de um sistema de processamento de imagens.	20
Tabela 2 - Teoriza do Aprendizado	23
Tabela 3 - Configurações de Hardware do Google Colab.	30
Tabela 4 - Ferramentas e Sistemas utilizados juntos ao Google Colab.....	31
Tabela 5 - Matriz de Confusão.....	39
Tabela 6 - Resultado das Métricas.....	41
Tabela 7 - Matriz de Confusão InceptionV3.....	43
Tabela 8 - Tabela de Métricas InceptionV3	43
Tabela 9 - Tabela de Confusão ResNet152	44
Tabela 10 - Tabela de Métricas ResNet152	44
Tabela 11 - Tabela de Confusão ResNet152 Transfer Learning.....	46
Tabela 12 - Tabela de Métricas ResNet152 Transfer Learning.....	46
Tabela 13 - Tabela de Confusão InceptionV3 Transfer Learning.....	47
Tabela 14 - Tabela de Métricas InceptionV3 Transfer Learning.....	47
Tabela 15 - Tabela de Confusão - InceptionV3 - Experimento 3	48
Tabela 16 - Tabela de Métricas - InceptionV3 - Experimento 3	49

LISTA DE EQUAÇÕES

Equação 1 - Equação Convolucional.....	28
Equação 2 - Equação de Accuracy.....	39
Equação 3 - Equação de Misclassification Rate.....	40
Equação 4 - Equação de Precision.....	40
Equação 5 - Equação de Recall	40

LISTA DE SIGLAS

AI	<i>Artificial intelligence</i>
CNN	<i>Convolutional Neural Network</i>
CPU	<i>Central Process Unit</i>
CUDA	<i>Compute Unified Device Architecture</i>
GCC	<i>GNU Compiler Collection</i>
GPU	<i>Graphics Processing Unit</i>
IA	Inteligência Artificial
ILSVRC	<i>Large Scale Visual Recognition Challenge</i>
MLP	<i>Multilayer perceptron</i>
RN	Rede Neural
RNA	Rede Neural Artificial
TPU	<i>Tensor Processing Unit</i>

SUMÁRIO

1 INTRODUÇÃO	15
1.1 Justificativa	17
1.2 Objetivo Gerais	17
1.3 Objetivos Específico	18
2 REFERENCIAL TEÓRICO	19
2.1 Processamento de Imagens	19
2.1.1 Um sistema de processamento de imagens e seus componentes	20
2.2 Inteligência Artificial	20
2.2.1 Inteligência da máquina e Aprendizado da máquina	22
2.3 Rede Neural	24
2.4 Transfer Learning	26
2.5 Deep Learning	26
2.6 Rede Convolucional	27
2.6.1 Convolução	27
2.6.2 Pooling	28
3. MATERIAIS E MÉTODOS	30
3.1 Métodos	30
3.2 Tecnologias	30
3.2.1 Bases de Dados	31
3.2.2 Ubuntu	32
3.2.3 Google Colaboratory	32
3.2.4 Python	32
3.2.5 TensorFlow	33
3.2.6 Keras	33
3.2.7 Gdown.pl	34
3.2.8 InceptionV3	34

3.2.9 ResNet152	34
3.3 Desenvolvimento	34
3.3.1 Passo 1: Escolha das ferramentas.	35
3.3.2 Passo 2: Definição das Redes Convolucionais	35
3.3.3 Passo 3: Compilador.	36
3.3.4 Passo 4: Definir Callback.....	36
3.3.5 Passo 5: Definir Parâmetros para Treinamento	37
3.3.6 Passo 6: Treinamento	38
3.3.7 Passo 7: Método de Avaliação das Redes.....	38
4. RESULTADO E DISCUSSÃO	42
4.1 Resultado e Discussões do Experimento 1.....	42
4.2 Resultado e Discussões do Experimento 2.....	45
4.3 Resultado e Discussões do Experimento 3.....	48
5. CONCLUSÃO	50
5.1 Trabalhos Futuros	50
REFERÊNCIAS	52

1 INTRODUÇÃO

A constante evolução da inteligência artificial (conhecido como AI, Artificial Intelligence, em inglês) e de seus métodos de aprendizagem de máquina proporciona um aumento de áreas em que máquinas podem desenvolver trabalhos, sejam elas aquelas que podem facilitar o trabalho de um ser humano (LOPES R. LINO, 2004).

O Aprendizado de Máquina pode ser utilizado para diversos trabalhos, independentemente do nível de complexidade. Basta determinar os limites, e assim pode-se utilizar para classificação de imagens, ou até análise de um conjunto muito maior de dados (HAYKIN, 2001).

Com a evolução das redes neurais, e até o uso de ferramentas gratuitas, como o Google Colaboratory, a utilização de redes neurais tem sido muito mais acessível, ainda mais com toda a busca de usuários para a utilização de redes neurais em diversas áreas, o termo se popularizou em diversos países do mundo.

O fácil uso se deve a criação de bibliotecas em Python que ajudam no desenvolvimento de técnicas de aprendizado automatizado, em especial: O TensorFlow que é uma biblioteca de algoritmo aberto para o Aprendizado de Máquina, contém uma extensa variedade de tarefas para a criação e treinamento de redes neurais para detectar e decifrar padrões e correlações. E junto ao TensorFlow como front-end é utilizado o Keras. Na ferramenta Keras há diversas ferramentas para o desenvolvimento de redes neurais, com uma série de ferramentas para facilitar o trabalho com dados de imagem e texto para simplificar a codificação necessária para a escrita de código de rede neural profunda (Deep Learning, em inglês).

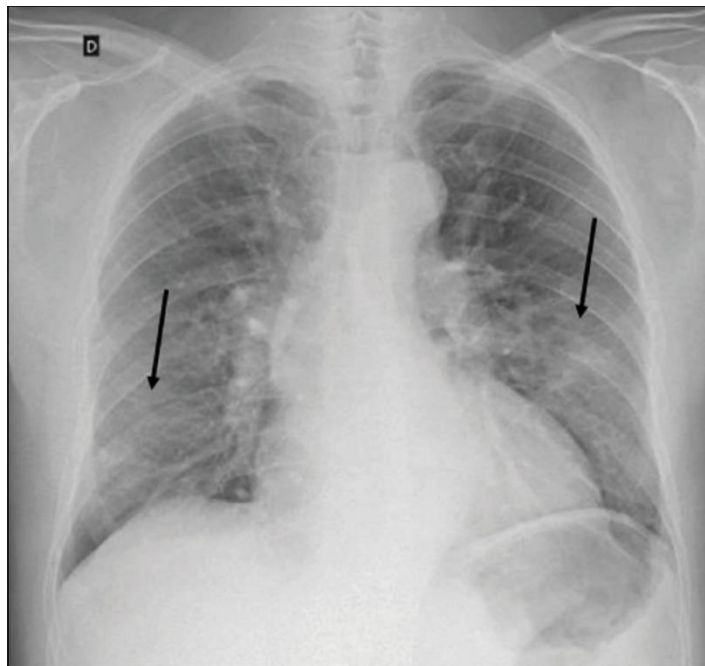
O aprendizado profundo, faz uso de redes neurais, com a diferença das várias camadas, permitindo modelos mais precisos, e ainda uma análise muito mais profunda dos dados. Essa técnica é muito utilizada em aplicações de conhecimento de fala, identificação de imagens e até mesmo previsões (HAYKIN, 2001).

As Redes Neurais Convolucionais (em inglês, Convolutional Neural Networks — CNN) é parte da rede neural artificial, que usa camadas convolucionais para filtrar as informações que podem ser úteis para uma tarefa específica. A operação de convolução envolve a combinação de dados de entrada (mapa de características) com um algoritmo de convolução (filtro) para formar um mapa de características

transformado. Os filtros nas camadas convolucionais são modificados com base nos parâmetros aprendidos para extrair as informações mais úteis. As CNNs se ajustam automaticamente para encontrar o melhor recurso com base nos parâmetros da tarefa. Por meio do treinamento, a CNN filtra informações específicas de um objeto, assim podendo diferenciá-los ou compará-los (HAYKIN, 2001).

Aplicações de CNNs se divergem em vários sistemas para o processamento de imagem, alguns deles são: reconhecimento de imagem e classificação de imagem. Na área médica pode-se desenvolver CNN capazes de classificar doenças que contenham pequenos detalhes, é o caso da radiografia de tórax. De acordo com Meirelles (2020) na Figura 1 é apontado com as setas ao lado esquerdo e direito uma opacidade pulmonar de baixa atenuação, ela se encontra em maior evidência ao lado esquerdo.

Figura 1 - Radiografia de Tórax de Paciente com Covid-19



Fonte: (MEIRELLES, 2020)

Nesse trabalho, foi utilizado duas CNNs para a classificação de imagens de raio-x de pulmões e das seguintes doenças: Covid-19 ou Pneumonia Viral, e verificar a capacidade de ambas as redes em classificar ocorrências de enfermidade no pulmão de um indivíduo.

1.1 Justificativa

Com o crescimento de casos do vírus Covid-19, o coronavírus (SARS-CoV-2) uma das maiores doenças desse século, e em pouco tempo de descobrimento, se tornou uma pandemia. Infelizmente, o Brasil não está preparado para enfrentar um vírus com tal magnitude, países que são mais evoluídos até conseguiram diminuir a taxa de mortes e contaminação, mas infelizmente não conseguiram erradicar o Covid-19 (WERNECK; CARVALHO, 2020).

Uma das formas de identificar o vírus é por meio de uma imagem de raio-x, com essa imagem é possível visualizar locais onde ocorrem uma opacidade pulmonar de baixa atenuação. Um exemplo pode ser visto na Figura 2 (MEIRELLES, 2020).

Figura 2 - Amostras de radiografia Pulmonar



(a) Amostra de Covid-19. Arquivo "COVID (11).png"



(b) Amostra Normal Arquivo "NORMAL (576).png"

Fonte: Imagens retiradas do conjunto de dados desenvolvido por: Chowdhury et al. (2020).

De acordo com Santos (2020) é possível diferenciar amostras de Covid-19 e outras doenças através da radiografia de tórax, mas não pode ser usado como um diagnóstico final.

1.2 Objetivo Gerais

O objetivo geral desse trabalho é utilizar duas CNNs para verificar o potencial que essa tecnologia tem em ajudar no diagnóstico tanto do Covid-19 quanto da Pneumonia Viral no corpo de um indivíduo, por meio de uma imagem de radiografia de tórax.

Com a ajuda de um *dataset* (um banco de dados de imagens), foi idealizado treinar duas CNNs, para que ambas classifiquem Covid-19, pneumonia viral ou a não existência dessas doenças.

1.3 Objetivos Específico

O objetivo específico deste trabalho foi estudar técnicas de CNN e implementar uma CNN com condições de classificar amostras de (Covid-19, Pneumonia Viral, pulmão sem doenças). Para isso, foram utilizadas amostras de um *dataset* (banco de dados de imagens) para o treinamento apropriado da rede neural, e por fim executar o treino e avaliar o comportamento da rede neural. Contudo, foi executado todos os passos acima, com duas redes neurais diferentes: InceptionV3 e a ResNet152.

Após as etapas de treinamento e teste, foi realizado comparações entre as amostras e obtido uma métrica de desempenho da rede, que no caso desse trabalho, avaliando imagens de pulmões, estipulado da seguinte forma: Covid-19, Pneumonia Viral ou Normal. E enfim, podendo auxiliar médicos a diagnosticar ambas as doenças: Covid-19 e Pneumonia Viral.

2 REFERENCIAL TEÓRICO

Neste capítulo, são definidos conceitos capitais para a compreensão deste trabalho. As seções a seguir abordam a inteligência Artificial e suas definições, apresentam a estrutura e o funcionamento das CNNs, e determinadas tecnologias relacionadas a sua utilização. Acoplado a compreensão do funcionamento do treino e da inteligência de uma máquina. Além disso, é feita uma introdução ao processamento de imagens.

2.1 Processamento de Imagens

O Processamento Digital de Imagens tem sido amplamente utilizado em diversas aplicações. Essas aplicações podem ser separadas em dois conceitos distintos: o primeiro é o aprimoramento da interpretação humana com a ajuda de imagens; e a segunda é a análise automática e quase espontânea do computador, que consegue analisar e obter informações, e até detalhar uma imagem de determinada cena (MARQUES; VIEIRA, 1999).

Segundo Marques e Vieira (1999), alguns exemplos dessas inúmeras aplicações estão: na medicina, a utilização de imagens no diagnóstico médico, que vem permitindo tanto o desenvolvimento de novos aparelhamentos, quanto ajuda na interpretação de imagens produzidas por aparelhamentos antigos; em biologia, a capacidade de processar automaticamente imagens adquiridas de microscópios, por exemplo, calculando o número de células presentes na imagem, que facilita o processo de tarefas em laboratórios, necessitam de um alto grau de precisão e repetitividade; e em geografia não seria diferente, já que o processamento a interpretação automatizada de imagens capturadas por satélites, e citando exemplos que se explora: Sensoriamento Remoto, Geoprocessamento e na Meteorologia, entre outras áreas da geografia; Na área de arqueologia, com a ajuda de métodos de restauração de imagens, que auxiliam arqueólogos a recuperar fotografias ofuscadas e borradas de itens raros que já foram extintos.

Existem outras áreas que distingam muito das já mencionadas, como: Astronomia, Segurança, Publicidade e Direito vêm sendo favorecidas com as melhorias nas áreas de processamento digital de imagens e visão por computador. O

uso de inteligências dotadas de visão artificial em empreitadas tais como: controle de qualidade em linhas de produção majora a cada ano. Num panorama de crescente automatização industrial (MARQUES; VIEIRA, 1999).

2.1.1. Um sistema de processamento de imagens e seus componentes

O conjunto de aparelhos de processamento de imagens de uso comum são indicados na Tabela 1. Os aparelhos descritos abaixo, conseguem simular desde aparelhamentos de baixo custo até terminais de alto custo utilizadas em aplicações que envolvem uso de imagens. Ele abrange as principais operações que se pode efetuar sobre uma imagem, a saber: aquisição, armazenamento, processamento e exibição. Todas estas operações são descritas a seguir (MARQUES; VIEIRA, 1999).

Tabela 1 - Elementos de um sistema de processamento de imagens.

Aquisição das Imagens	Processamento das Imagens	Saída	Armazenagem
Câmeras	Computador	Monitores de Vídeo	Discos ópticos
		Impressoras	Discos Magnéticos

Fonte: Adaptado de Marques e Vieira (1999).

2.2 Inteligência Artificial

De acordo com Russell e Norving (2013) a Inteligência artificial possui 4 categorias: Máquinas que pensam como humanos; Máquinas pensam racionalmente; Máquinas que se comportam como seres humanos; Máquinas que se comportam racionalmente.

Alan Turing em 1950 propôs um teste para fornecer uma definição operacional satisfatória de inteligência. Um interrogador deve propor algumas perguntas por escrito e receber as respostas de suas perguntas de forma escrita e identificar se as respostas vêm de uma pessoa ou um computador. A interação física direta de uma pessoa é desnecessária para a inteligência, porém, o teste de Turing total inclui um sinal de vídeo, para que o interrogador possa testar as habilidades de percepção do

indivíduo. Deve-se se oferecer também a opção de o interrogador repassar objetos físicos para ao indivíduo.

Para ser aprovado o computador precisará de alguns requisitos: processamento de linguagem natural em que o permite a comunicação de forma natural; representação de conhecimento para armazenar o que sabe ou ouve; raciocínio automatizado para utilização das informações armazenadas com o objetivo de responder as perguntas e tirar novas conclusões e assim armazena-las novamente; aprendizado de máquina para a adaptação em todas as circunstâncias detectando padrões; visão computacional para perceber objetos; robótica para movimentos. Estas seis funcionalidades definem uma grande parte da Inteligência Artificial, apesar disso os pesquisadores de IA, dedicam poucos esforços para a aprovação no teste de Turing, acreditando que seja mais importante estudar os princípios básicos da inteligência do que reproduzir um exemplar. O teste de Alan Turing se refere a máquinas que se comportam como seres humanos (RUSSELL; NORVING, 2013).

Para determinar que uma máquina pensa como um ser humano primeiro deve-se identificar como um ser humano pensa, existem três maneiras de fazer isso: introspecção que é capturar nossos pensamentos na medida em que eles se desenvolvem, também é possível identificar através de experimentos psicológicos observando pessoas em ação, e através de imagens reais do cérebro identificando todo seu comportamento. Se os comportamentos da máquina coincidirem com o comportamento humano usando um desses três métodos temos a evidência de que alguns dos mecanismos do programa estão funcionando como seres humanos (RUSSELL; NORVING, 2013).

De acordo com Russell e Norving (2013) o filósofo Aristóteles ao tentar codificar o “pensamento correto” em que é um processo de raciocínios irrefutáveis. Seus silogismos oferecem padrões para argumento que sempre resultam em conclusões corretas ao receberem as premissas corretas, como exemplo: “Sócrates é um homem; todos os homens são mortais; então, Sócrates é mortal”. Esse estudo deu início a uma área, chamada: lógica. Os lógicos do século XIX desenvolveram uma notação precisa para declarações sobre todos os tipos de coisas no mundo e sobre as relações entre elas (compare isso com a notação aritmética básica, que fornece apenas declarações a respeito de números). Em 1965, existiam programas que, em princípio, podiam resolver qualquer problema solucionável descrito em notação lógica (contudo,

se não houver solução, o programa poderá entrar num laço infinito) (RUSSELL; NORVING, 2013).

A chamada tradição lógica dentro da inteligência artificial espera desenvolver tais programas para criar sistemas inteligentes, isso define máquinas que pensam de forma racional (RUSSELL; NORVING, 2013). Segundo Russell e Norving (2013), há uma diferença entre resolver um problema “em princípios” e resolvê-lo na prática. Para a Inteligência Artificial, problemas com apenas algumas centenas de fatos podem fazer com que os computadores esgotem seus recursos computacionais, pois, a complexidade se torna grande, ao menos que se tenha uma orientação de onde o computador deve começar seus raciocínios.

Todo e qualquer programa realiza algo, mas quando se trata de um agente computacional espera-se que ele opere de forma autônoma, que se adapte ao seu ambiente e a mudanças e seja capaz de criar e seguir metas de modo que alcance o melhor resultado, esse é o exemplo de uma máquina que se comporta racionalmente (RUSSELL; NORVING, 2013).

2.2.1 Inteligência da máquina e Aprendizado da máquina

Para abstrair e entender o Aprendizado de Máquina, precisa-se de determinar e entender os seguintes itens: inteligência e aprendizado. Pode parecer ser temas simples de definir, mas veremos um pouco mais nos assuntos a seguir (OSÓRIO; BITTENCOURT, 2000).

2.2.1.1 Teoria do Aprendizado

O Aprendizado tem uma ampla importância para que possa desenvolver um sistema inteligente com a capacidade de aprender.

Aprender é o ato de se adaptar, melhorar e talvez se modificar para melhorar sua resposta, e até sua ação e reação. Então é uma característica bem importante para indivíduos que se dizem inteligentes.

Segundo Osório e Bittencourt (2000) a capacidade de aprender é a definição junto dos seguintes itens: adaptação, correção, otimização e interação. Pode se obter uma ideia, mais vastas desses itens na Tabela 2.

Tabela 2 - Teoriza do Aprendizado

Adaptação	Adaptação é melhorar e modificar o comportamento dependendo do critério. Um sistema ou ser, que não tem o poder de melhorar ou se modificar perante novas circunstâncias e barreiras que lhe propõem é um ser/sistema sem inteligência.
Correção	Correção dos erros cometidos no passado, de modo que não se repita posteriormente. O sistema precisa se modificar, caso o desempenho atual não satisfaça a um determinado tipo de exigência.
Otimização	Otimização é arquitetar a melhora do desempenho do sistema, o aprendizado pode exigir uma mudança do comportamento que procure algo mais além do que se alterar-se apenas para chegar à conclusão de um trabalho, mas também pode se modificar a fim de buscar: economia de energia gasta a fim de concluir um trabalho, a redução do tempo gasto etc. Quando se pensa em otimização, deve se levar em consideração quesitos que podem ser contraditórios, onde deve-se elevar ao máximo e minimizar custos de acordo com algum critério, e ainda temos o contato com o mundo, que será explicado melhor na próxima clausula: Interação.
Interação	Interação se baseia em trocar experiencias com o mundo a fora, na ideia de adquirir novos conhecimentos. Em um sistema, quando falamos de interação: é buscar pela internet ou no mundo que nos cerca, por aquilo que se precisa e adquirindo novos aprendizados e conhecimentos, e melhorar desde: respostas até a velocidade de resposta.

Fonte: Osório e Bittencourt (2000)

A principal característica da rede neural, é exatamente a possibilidade de se adaptar, aprender e filtrar todo conhecimento proposto.

2.2.1.2 Teoria da Inteligência

Desenvolver sistemas inteligentes é a idealizar a reprodução da inteligência humana em uma máquina, e alocar trabalhos e tarefas através de implementações que automatizem esse tipo de processo (OSÓRIO; BITTENCOURT, 2000).

O termo Artificial Intelligence ou Inteligência Artificial, em português, foi usado pela primeira vez em 1956 por McCarthy, que é uma tentativa de formalizar e idealizar um “cérebro eletrônico” (termo frequentemente usado na ficção científica e mesmo na época inicial do desenvolvimento dos computadores).

Segundo Osório e Bittencourt (2000), pode-se especificar alguns termos para ajudar na definição de um ser inteligente. Os significados podem ter contradições, onde pode ser dizer: inteligência distingue o homem do animal. Entretanto, isso não é muito conciso, já que animais também podem aprender, conhecer e compreender, por exemplo: aprender a distinguir pessoas, conhecer o caminho de volta para a casa e compreender um comando que foi ensinado. Essas tarefas podem ser classificadas de seres inteligentes, já que nem os computadores na atualidade, conseguem um desempenho satisfatório.

Uma pessoa com limitações, seja: deficiente, analfabeta ou com algum problema intelectual. Poderá ser classificada como inteligente? Independente da resposta, a questão é, essa pessoa já possui aptidões superiores à de um computador. Já que um computador não vai até o mercado e faz sua lista de compras. Mas deve-se pensar bem, para realizar tal tarefa é necessário: planejamento da trajetória, identificar o caminho e passar por imprevistos etc. Logo o conceito de inteligência, indica que ser humano ou não, o ser inteligente deve ser capaz de: se adaptar, aprender, evoluir e corrigir seus erros (OSÓRIO; BITTENCOURT, 2000).

2.3 Rede Neural

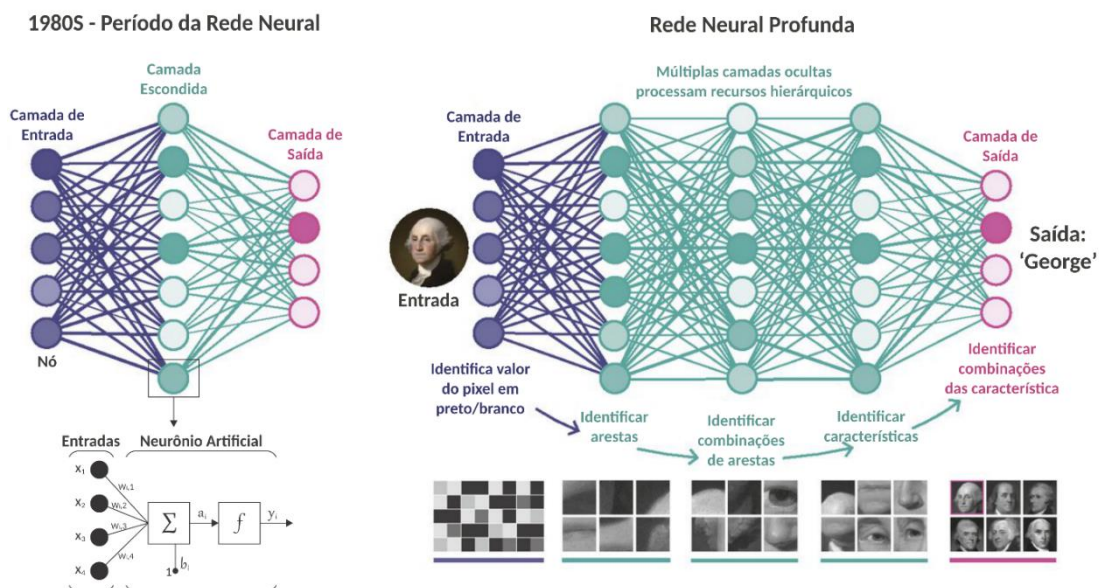
Redes neurais artificiais também chamadas de “redes neurais” são motivadas pela forma em que um cérebro humano processa informações. Desta forma o cérebro é visto como um sistema de processamento de informações (entradas e saídas) altamente complexo, não-linear e paralelo. Ele pode organizar seus neurônios de forma em que seja possível realizar certos processamentos, dentre eles temos o reconhecimento de padrões, percepção e o controle do corpo, e o cérebro consegue

efetuar essas atividades mais rápido de qualquer computador digital existente (HAYKIN, 2001).

Redes neurais geralmente é implementada usando componentes eletrônicos (*hardware*) ou programada em computadores digitais (*software*) que tem como objetivo emular a maneira em que o cérebro realizar tarefas ou funções em particular. Entre todas as topologias de redes neurais artificiais o *Perceptron* Multicamadas é a topologia mais usada, um momento em que o *perceptron* é capaz de resolver inúmeros problemas, mas tem uma capacidade especial em reconhecimento de padrão e regressão numérica. Como mostra ao lado esquerdo inferior da Figura 3, a saída de um neurônio artificial é representada por y_i que se dá pela sua ponderação das suas n entradas ($x_1, x_2, x_3 \dots, x_n$) pelos seus pesos sinápticos ($w_{i,1}, w_{i,2}, \dots, w_{i,n}$) mais a soma de seu limiar de disparo (b_i) que se dá o resultado a_i que é passado para função de ativação (f) e temos o resultado de y_i (HAYKIN, 2001).

Uma rede neural deve-se antes passar por um treinamento em que é dado uma entrada N com uma resposta R , onde a rede neural deve receber a entrada N e dar uma resposta X , a depender da distância das respostas R e X é feito a calibração de seus pesos para que possa chegar o mais perto da resposta R .

Figura 3 - Exemplo de Rede Neural Profunda



Fonte: Adaptado de Waldrop (2019)

2.4 Transfer Learning

De acordo com Pan e Yang (2010) a transferência de aprendizagem (do inglês *transfer learning*) é útil quando se tem poucos dados de treinamento e é necessário um treinamento curto. Um exemplo dado por Pan e Yang (2010) em que o aprendizado de uma rede que classifica maçãs pode ser usado no início do treinamento de uma rede que classifica peras através da transferência de aprendizagem.

Pan e Yang (2010) define a transferência de aprendizagem da seguinte forma: se tem o domínio D_s e a tarefa de aprendizagem T_s , um domínio alvo D_t e a tarefa de aprendizagem T_t , transferência de aprendizagem visa melhorar a aprendizagem de T_t usando inicialmente T_s .

Geralmente a transferência por aprendizado pode ser utilizada quando uma rede A possui o mesmo tipo de entrada de uma rede B; quantidade de dados de A é maior que a quantidade de dados em B; algumas características de A podem ser utilizadas em B.

2.5 Deep Learning

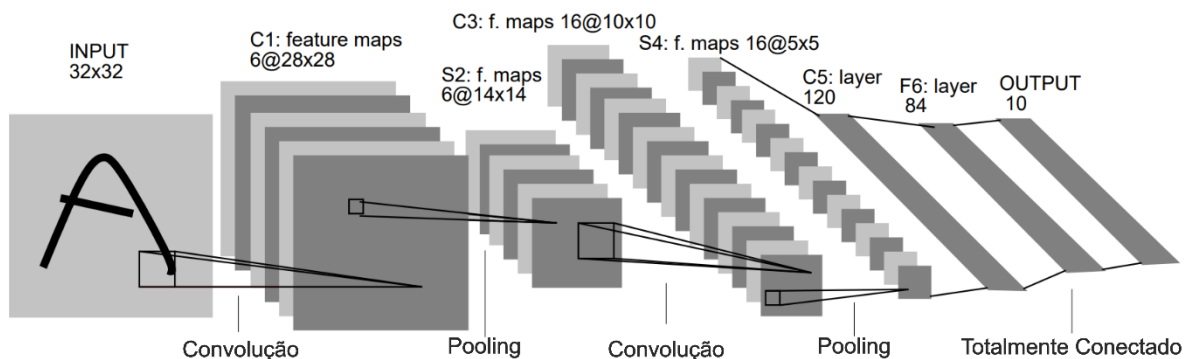
Deep Learning em português chamado como aprendizagem profunda tem recebido uma maior atenção e notoriedade, pois, existem resultados bastantes positivos na área de inteligência artificial e visão computacional, contribuindo assim no desenvolvimento de sistemas inteligentes. O aprendizado de representações em camadas (*layered representations learning*) e o hierárquico (*hierarchical representation learning*) também podem ser chamados de *deep learning* devido sua abordagem em aprendizado com camadas consecutivas a partir de dados de entrada (ROSA, 2019).

Redes neurais profundas são diferentes de uma rede neural comum, pois, os procedimentos de cálculos são feitos em muitas camadas, e cada camada subsequente se torna mais abstrata como mostra a Figura 2. Ao utilizar esta técnica pode-se perceber que os nós amplificamos a relevância de recursos essenciais e ignore os menos relevantes (ROSA, 2019).

2.6 Rede Convolucional

A rede convolutiva é uma rede *perceptron* com múltiplas camadas desenvolvida inicialmente por Lecun et al. (1998) para o reconhecimento de matrizes bidimensionais com muita invariância (translação, escalonamento, inclinação, cores e outras formas). Ao utilizar redes neurais convolutivas para resolver problemas de classificação se tem uns bons resultados quanto a uma rede *perceptron* multicamadas comuns (RODRIGUES, 2018). Apesar da rede neural convolucional conter resultados gratificantes ela possui uma desvantagem ao realizar o treinamento, pois, ela necessita de uma grande quantidade de dados rotulados para a extração de *features* (padrões). As redes neurais convolucionais básicas são compostas por três componentes básicos chamados: convolução, *pooling* e rede totalmente conectada (RODRIGUES, 2018). A rede criada por Lecun et al. (1998), a LeNet-5 pode ser vista na Figura 4.

Figura 4 - Rede Neural Convolucional LeNet-5



Fonte: Adaptado de Lecun et al. (1998)

2.6.1 Convolução

A convolução é iniciada utilizando filtros convolucionais que ao passar por toda a entrada em largura, altura e profundidade realizando uma operação de multiplicação de matrizes se tem como resultado uma *feature*. Na matemática, a convolução é vista como um operador linear que tem como entrada duas funções: função de entrada e a função *kernel*, e retorna uma terceira função resultante do somatório da multiplicação

da função *kernel* na região superposta na função de entrada pelo deslocamento do *kernel* sobre ela. A Equação 1 é a convolução no domínio discreto onde f e g são matrizes de tamanhos iguais ou variados e se tem como retorno o k -ésimo elemento do somatório da multiplicação:

Equação 1 - Equação Convolucional

$$(f * g)(k) = h(k) = \sum_{i=0}^k f(i) * g(k - i)$$

Fonte: Rosa (2019).

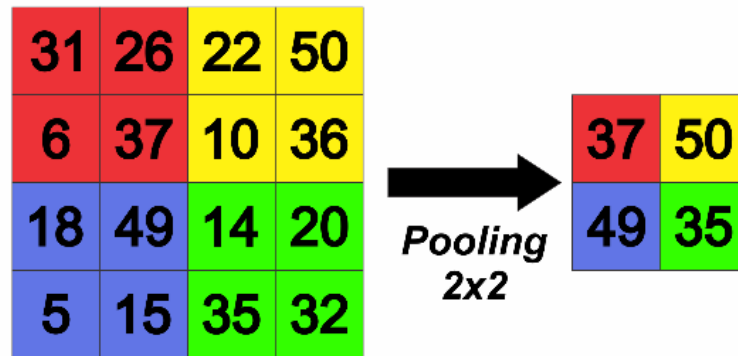
Durante o treinamento da rede os filtros se ajustam de modo a identificar arestas, cores, círculos, retângulos entre outras características. Quanto mais filtros convolucionais, mais resultados de extração de *feature* é possível obter, porém, o uso de uma quantidade alta de filtros pode resultar em perda de desempenho (ROSA, 2019).

2.6.2 Pooling

O *pooling* consiste em reduzir o tamanho da entrada para a próxima camada da rede, é comum o uso do *pooling* após uma convolução, pois ao utilizá-lo se tem uma saída que representa de forma diferente a sua entrada. Ao passar para a próxima camada de convolução se tem a possibilidade da rede aprender novas formas de filtros e evitar o *overfitting* (CS231N..., 2020).

Max-pooling é uma técnica bastante utilizada em redes neurais convolucionais, o seu funcionamento é basicamente reduzir subpartes dos dados originais pelo maior valor encontrado na sub região de $m \times n$ (ROSA, 2019). Podemos ver um exemplo na Figura 5.

Figura 5 - Exemplo de Max-Pooling



Fonte: Elaborado pelo Autor

2.6.3 Camada totalmente conectada

Na camada totalmente conectada as *features* extraídas da convolução/*pooling* são utilizadas para a classificação da rede. Comumente a camada totalmente conectada é uma *perceptron* multicamadas após o último *pooling* no final da rede convolutiva (ROSA, 2019).

3. MATERIAIS E MÉTODOS

Neste capítulo será discutido os métodos científicos presente nesse trabalho, em seguida as especificações das tecnologias utilizadas e o desenvolvimento do projeto.

3.1 Métodos

De acordo com Prodanov e Freitas (2013) uma pesquisa aplicada tem como o objetivo gerar um conhecimento para uma aplicação prática dirigida a soluções de problemas específicos. Esta pesquisa tem como natureza aplicada, pois, irá gerar conhecimento sobre como uma rede neural convolucional é capaz de classificar imagens de raio-x de Covid-19 e Pneumonia Viral.

A pesquisa como forma de abordagem pode se classificar como quantitativa, pois, como diz Prodanov e Freitas (2013) uma pesquisa quantitativa é que tudo pode ser quantificável em que se pode trazer números para analisá-los e classificá-los.

3.2 Tecnologias

Este trabalho foi desenvolvido utilizando o Google Colab para o treinamento e para os procedimentos dos resultados. O hardware do Google Colab pode ser visto na Tabela 2.

Tabela 3 - Configurações de Hardware do Google Colab.

Processador	Intel® Xenon® CPU @ 2.20GHz
Memória Ram	RAM 26GB
Placa de vídeo	GPU Tesla P100

Fonte: Elaborado pelo Autor.

O sistema operacional e outras especificações de software podem ser vistos na Tabela 3.

Tabela 4 - Ferramentas e Sistemas utilizados juntos ao Google Colab

Sistema Operacional	Outras Ferramentas Utilizadas
Ubuntu 18.04.5 LTS	GCC 8.4.0
Kernel Linux 4.19.112+	Python: 3.6.9
	Tensorflow 2.4.1
	Keras 2.4.3
	Cuda 10.1
	Gdown.pl 1.4

Fonte: Elaborado pelo Autor.

3.2.1 Bases de Dados

Como já mencionado neste trabalho, para o treinamento da CNN, foi selecionado um banco de dados de imagens, que é conhecido em inglês como *dataset*. A base de dados *Covid-19 Radiography Data Set* (CHOWDHURY et al., 2020) foi utilizado para o desenvolvimento do trabalho, essa base de dados, contém classes, que é o rótulo que classifica as imagens. Existem três classes com os seguintes padrões: na primeira classe possui inúmeras imagens de raio-x de pulmões saudáveis com suas variantes, na segunda classe contém inúmeras imagens de raio-x de pulmões com Covid-19 e variantes, e na terceira e última classe, possui imagens de raio-x de pulmões com Pneumonia Viral e variantes. Todas as imagens estão em formato PNG, ao todo são: 1200 imagens de Covid-19; 1345 imagens de Pneumonia Viral; 1341 imagens de pulmões sem essas determinadas doenças. Foi selecionado aleatoriamente 70% de cada classe do conjunto de dados para o treinamento das redes neurais convolucionais, e os outros 30% de cada classe para os testes finais e seus resultados.

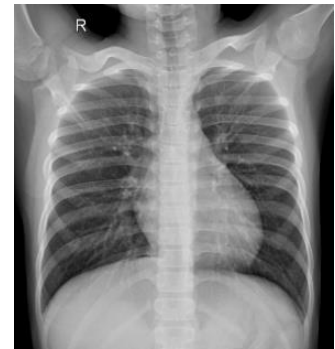
Figura 6 - Amostras de Radiografias de Tórax



(a) Amostra de Covid-19
"COVID (11).png"



(b) Amostra Normal
"NORMAL (576).png"



(c) Amostra Viral Pneumonia
"Viral Pneumonia (770).png"

Fonte: Chowdhury et al. (2020).

3.2.2 Ubuntu

Ubuntu é um sistema operacional de código aberto, desenvolvido a partir do Kernel do Linux, baseado em Debian, onde não exige assinatura para obtenção da aplicação, e isento de qualquer taxa (CANONICAL, 2004). Segundo Canonical (2004), Ubuntu oferece um sistema que qualquer pessoa possa aproveitar sem dificuldades, independente da nacionalidade, cultura e nível de conhecimento.

Ainda se aproveita do drive de aceleração GPU para placas da NVIDIA, onde pode se obter um desempenho mais satisfatório para o treinamento e teste de uma Rede Neural Convolucional.

3.2.3 Google Colaboratory

Google Colab ou Google Colaboratory é um ambiente gratuito que permite a execução de códigos em Python no navegador de forma arbitrária e é especialmente adequado para o aprendizado de máquina, análise de dados e educação, podendo utilizar aceleração GPU ou não (CARNEIRO et al., 2018).

3.2.4 Python

De acordo com Perkovic (2016) Python é uma linguagem de programação de código aberto, desenvolvida inicialmente por Guido van Rossum em 1980 e lançado

oficialmente em 1991. Python é uma linguagem de uso geral, criada para um desenvolvimento de códigos mais legíveis, suas construções linguísticas e abordagem orientada a objetos visam ajudar a escrever códigos claros. A programação orientada a objetos e a programação estruturada são suportadas, e a abstração de algumas características que suporta programação funcional e programação orientada a aspectos, e muitos outros paradigmas que não funcionam tradicionalmente, é possível utilizá-los através de extensões e bibliotecas, incluindo programação lógica. A linguagem possui uma enorme biblioteca facilitando assim a criação de aplicações.

3.2.5 TensorFlow

De acordo com Abadi et al. (2016) TensorFlow é uma biblioteca de tensores de baixo nível desenvolvida em linguagem como C/C++, Golang e Python. É a principal biblioteca de código aberto para desenvolver modelos de aprendizado de máquinas, já que há inúmeras ferramentas para tornar o desempenho mais significativo, um exemplo é a unidade processadora de tensores, ou TPU (*tensor processing unit*). TPU é basicamente um acelerador de inteligência artificial programável desenvolvido para amparar e fornecer uma alta percentual de transferência para aritmética de baixa precisão, e encaminhada para o uso ou o implemento de modelos, ao invés de treino. TensorFlow ainda possui suporte tanto para CPU quanto para GPUs da NVIDIA que utilizam o CUDA, que inclusive uma das melhores GPUs para a detecção de objetos é a Pascal Titan X (ABADI et al., 2016).

3.2.6 Keras

O Keras é uma biblioteca exclusiva do Python que se apresenta como *frontend* do TensorFlow em python. Ele segue a ideia de reduzir a curva de aprendizagem no desenvolvimento de modelos de aprendizagem de máquina, contendo uma documentação completa com seus métodos e como utilizá-los sendo assim uma biblioteca que pode ser utilizada tanto para profissionais quanto por estudantes.

O Keras permite que os usuários desenvolvam modelos profundos em dispositivos móveis, como smartphone, e vai além, permitindo disseminar o treinamento dos algoritmos em unidades de processamento gráfico (GPU) e unidades

de processamento tensorial (TPU) especialmente unidos com a CUDA da NVIDIA (CHOLLET, 2015).

3.2.7 Gdown.pl

A biblioteca Gdown.pl é usada para downloads diretos ao Google Drive por linha de comando. É compatível em Windows e Linux, ferramenta disponível em: <https://github.com/circulosmeos/gdown.pl>

3.2.8 InceptionV3

A InceptionV3 é uma rede neural convolucional que auxilia na análise de imagens e detecção de objetos, e que começou inicialmente com a GoogLeNet, e que se provou ter altas taxas de acertos em objetos que possuem mínimos detalhes para classificação (WOJNA et al., 2015). Além disso, a InceptionV3 é uma rede que se encontra implementada no *Keras* e possui um treinamento prévio com o *dataset* do *ImageNet*.

3.2.9 ResNet152

A ResNet152 é uma rede convolucional que venceu a competição *Large Scale Visual Recognition Challenge* (ILSVRC) em 2015, um de seus adversários eram a InceptionV3 (HE et al., 2015). ResNet152 se encontra implementada no *Keras* com um pré treinamento com o *dataset* do *ImageNet*. ResNet152 possui um total de 152 camadas.

3.3 Desenvolvimento

Nesta sessão, será apresentada passo a passo dos métodos realizados para o desenvolvimento do trabalho. Abaixo, pode ser vistos os procedimentos realizados em passos:

- Passo 1: Escolhas das ferramentas;

- Passo 2: Definição das Redes Convolucionais;
- Passo 3: Escolha de Compilador;
- Passo 4: Definir Callback;
- Passo 5: Definir Parâmetros para Treinamento;
- Passo 6: Treinamento;
- Passo 7: Métodos de Avaliação das Redes.

3.3.1 Passo 1: Escolha das ferramentas.

Nesse trabalho foi escolhido o ambiente do Google Colaboratory já que ele possui todas as bibliotecas necessárias instaladas e atualizadas, e a execução de códigos é por blocos o que promove facilidade no momento da execução. É possível reiniciar todo o ambiente em caso de um erro não esperado e assim iniciar os testes novamente. É uma ferramenta gratuita, porém em sua versão paga possui acesso a GPUs mais potentes que resulta em execuções mais rápidas.

As versões dos *softwares* como o Tensorflow e Keras são as versões em que atualmente são consideradas estáveis. O Keras nesta versão possui algumas redes implementadas como Xception, VGG16, ResNet152, InceptionV3, MobileNet, DenseNet121, EfficientNetB0, entre outras. As redes já implementadas possuem um treinamento prévio, que agiliza o tempo de treinamento para soluções de novos problemas.

3.3.2 Passo 2: Definição das Redes Convolucionais

Neste trabalho foi usado duas redes neurais convolucionais, uma das escolhidas é a InceptionV3 pois de acordo com Wojna et al., (2015) a primeira versão a InceptionV1 conhecida como GoogleNet venceu em primeiro lugar a competição da ILSVRC em 2014, e a InceptionV3 como sucessora das suas anteriores versões desempenham um resultado igual ou superior a elas. De acordo com seus criadores a InceptionV3 possui resultados melhores quando se trata de pequenos detalhes na classificação, e isso se encaixa no problema, pois analisar características de pulmões não é uma atividade como classificar peras e maçãs, as redes devem se atentar as diferenças pequenas entre pulmões com pneumonia viral e covid-19.

A segunda rede escolhida se trata da ResNet152 vencedora da competição ILSVRC em 2015 na qual se saiu melhor que a InceptionV3. A diferença de resultados entre as duas redes não são altas, e esse trabalho não tem o objetivo de julgar as que foram utilizadas na etapa de experimentos, a hipótese do trabalho é avaliar o desempenho das CNNs na área médica.

Para alcançar resultados adequados foi utilizado a transferência de aprendizado, onde ambas as redes são iniciadas com o treinamento prévio feito com o conjunto de dados do *ImageNet*. Também foi utilizado as redes iniciadas com pesos aleatórios. No entanto, a camada de saída é inicializada com mil classes e para a solução do problema precisa-se de três classes, logo a camada de saída é destruída completamente e é criada uma camada com três classes.

3.3.3 Passo 3: Compilador.

A compilação de uma rede é a verificação em que o keras passa por sua rede validando suas camadas e é o local onde se escolhe o otimizador, a função de perda e a métrica a ser alcançada.

O otimizador utilizado será o Adam, pois, de acordo com as pesquisas de Kingma e Ba (2014) é averiguado os resultados de outros compiladores comparados ao Adam onde Adam tem um desempenho tanto qualitativo como quantitativo por iterações.

Com base nas funções de percas implementadas no keras foi utilizar a *categorical_crossentropy* pois ao lidar com um problema com três classes a *crossentropy* tem bons resultados quando se trata de múltiplas classes (CHOLLET, 2015).

A métrica usada foi a *accuracy*, que calcula a frequência em que as previsões são iguais aos rótulos, em outras palavras, se divide o total do conjunto por total de acertos (CHOLLET, 2015).

3.3.4 Passo 4: Definir Callback

Callbacks são métodos que podem ser executados em vários estágios do treinamento como, por exemplo, no começo ou no final de cada época, antes ou depois de um lote, entre outros (CHOLLET, 2015).

O único *Callback* utilizado será o *ModelCheckpoint*, esse modelo de *Callback* salva o estado do treinamento de acordo com o parâmetro passado, suas configurações são as seguintes:

- Pode-se salvar apenas os pesos do modelo ou todo o modelo;
- Salvar o modelo que alcançou o melhor resultado de acordo com a métrica passada por parâmetro;
- Frequência na qual deve ser executada: final de cada época ou após um número fixo de lotes.

Foi utilizado o salvamento da rede de acordo com os acertos do conjunto de dados de teste após cada época. Durante o treinamento quando a rede alcançar o maior valor foi efetuado o salvamento do seu estado e os resultados apresentados neste trabalho é o estado da rede no momento do salvamento.

3.3.5 Passo 5: Definir Parâmetros para Treinamento

Para etapa de treinamento deve-se separar um conjunto de dados de treino e de teste. Como mencionado anteriormente o conjunto de dados permanecem separados, 70% dos dados para treino e 30% para os testes, alocados em pastas diferentes respectivamente *train* e *test*. Para transformar o conjunto de dados em dados compatíveis com TensorFlow foi utilizado o *ImageDataGenerator* do Keras, a funcionalidade dele vai além de converter os dados já existente, ele permite criar mais dados ao rotacionar, dar zoom, inverter em horizontal, vertical, entre outros. Foi utilizado apenas para conversão dos dados passando o parâmetro *reescale* como 1/255, significa que todos os valores contidos em cada imagem no conjunto de dados será abaixo ou igual a 1.

O objeto *ImageDataGenerator* possui um método chamado *flow_from_directory*, a sua funcionalidade é apontar onde está o conjunto de dados que será usado no objeto e passar os seguintes parâmetros: *directory* (local onde estão os dados), *target_size* (o tamanho que os dados devem ser redimensionados) e *batch_size* (tamanho de cada lote de dados). Deve-se criar dois objetos *ImageDataGenerator* um para os dados de treino e outro para os dados de teste e apontar o *flow_from_directory* para seus respectivos conjuntos de dados. O parâmetro *directory* é o local onde estão os dados (*train* e *test*), *target_size* é de acordo com a

entrada que a rede recebe, no caso InceptionV3 recebe uma imagem 299x299 *pixels*, e ResNet152 recebe 224x224 *pixels*. *Batch_size* é o parâmetro com dificuldade de se encontrar um valor correto, leva-se em conta o tempo do treinamento, o tamanho do conjunto de dados, a quantidade de classes para rede aprender, entre outros. Neste trabalho o valor escolhido de *batch_size* foi 64, após testes serem efetuados teve-se um resultado em que um *batch_size* maior proporciona melhores resultados porem um longe treinamento.

3.3.6 Passo 6: Treinamento

No treinamento da rede convolucional a especificação da forma como o treinamento deve se comportar é bem ampla, mas neste trabalho é utilizado o objeto *ImageDataGenerator*, ele também faz o trabalho de rotular todos dados para o treinamento e para o teste por época. Todo e qualquer modelo de rede convolucional do *keras* possui métodos de treinamento e um deles é o *Fit*. *Fit* possui muitos parâmetros dentre eles alguns obrigatórios e outros opcionais. Os parâmetros passados para o método *Fit* neste trabalho são os objetos *ImageDataGenerator* de treino e o de teste, o *callback ModelCheckpoint* e a quantidade de épocas que é a quantidade de iterações efetuadas após o método *Fit* passar por todo conjunto de dados, o valor escolhido de épocas é 250, o que resulta em um treinamento rápido, em outras palavras as duas redes foram treinadas utilizando o método *Fit* com os mesmos parâmetros em um total de 250 épocas com o ajuste de pesos de 64 em 64 imagens. O treinamento de cada CNN levou aproximadamente três horas e vinte minutos.

3.3.7 Passo 7: Método de Avaliação das Redes

No método de avaliação dos modelos propostos neste trabalho foi utilizado inicialmente uma matriz de confusão como pode ser visto no exemplo da Matriz de Confusão da Tabela 5.

Tabela 5 - Matriz de Confusão

	Covid-19	Pneumonia Viral	Normal
Covid-19	Resultado	Resultado	Resultado
Pneumonia Viral	Resultado	Resultado	Resultado
Normal	Resultado	Resultado	Resultado

Fonte: Elaborado pelo Autor.

De acordo com as métricas propostas por Mohajon (2020), deve-se avaliar os dois modelos de redes convolucionais com as seguintes variáveis:

- Verdadeiro positivo (do inglês True Positive) sigla TP. Refere-se ao valor onde o classificador prediz corretamente a classe positiva como positiva.
- Verdadeiro negativo (do inglês True Negative) sigla TN. Refere-se ao valor onde o classificador prediz corretamente a classe negativa como negativa.
- Falso positivo (do inglês False Positive) sigla FP. Refere-se ao valor que o classificador classifica incorretamente a classe negativa como positiva.
- FN Falso negativo (do inglês False Negative). Refere-se ao valor que o classificador classifica incorretamente a classe positiva como negativa.

Com as seguintes métricas:

- *Accuracy* do português acurácia denota a exatidão total da rede, em outras palavras se dá como resultado a fração total das amostras que foram classificadas corretamente, a sua equação pode ser vista na Equação 2.

Equação 2 - Equação de Accuracy

$$\frac{TP + TN}{TP + TN + FP + FN}$$

Fonte: Mohajon (2020).

- *Misclassification Rate* do português taxa de classificação incorreta se destaca por informar a fração de classificações incorretas utilizando a Equação 3.

Equação 3 - Equação de Misclassification Rate

$$\frac{FP + FN}{TP + TN + FP + FN}$$

Fonte: Mohajon (2020).

- Precision do português precisão é utilizada para informar a fração em que a classificação acertou corretamente a devida classe.

Equação 4 - Equação de Precision

$$\frac{TP}{TP + FP}$$

Fonte: Mohajon (2020).

- Recall do português podemos chamar também de sensibilidade retorna à fração em que uma classe é classificada corretamente pela rede se dá pela Equação 5:

Equação 5 - Equação de Recall

$$\frac{TP}{TP + FN}$$

Fonte: Mohajon (2020).

De acordo com Mohajon (2020), as variáveis são extraídas a partir da matriz de confusão, e cada classe possui valores diferente das variáveis, sendo assim as métricas devem ser calculadas por classes.

Figura 7 - Exemplo de Matriz de Confusão com Classes

		Classes Corretas		
		Maçã	Laranja	Manga
Classes Prevista	Maçã	7	8	9
	Laranja	1	2	3
	Manga	3	2	1

Fonte: Adaptado de Mohajon (2020)

Utilizando a Figura 7 como referência, deve-se encontrar os valores de TP, TN, FP e FN de cada classe. Uma forma simples é começar pelo valor de TP ele é o valor que está no encontro da classe, no caso o TP da classe Laranja é 2 pois é o valor que se encontra na linha laranja e na coluna laranja (linha 2 coluna 2). O próximo valor é FP ele é a soma de todos os valores na linha da classe laranja retirando o valor de TP, então FP é $1 + 3 = 4$. FN se dá com a soma de todos os valores da coluna laranja retirando o valor de TP logo FN é $8 + 2 = 10$. TN é considerado a soma de todo o restante dos valores da matriz, ou seja, TN é $7 + 9 + 3 + 1 = 20$. Todos os valores das variáveis junto de suas métricas podem ser vistas na Tabela 6.

Tabela 6 - Resultado das Métricas

Classe	TP	TN	FP	FN	Accuracy	Misclassification	Precision	Recall
Maçã	7	8	17	4	0,41	0,58	0,29	0,64
Laranja	2	20	4	10	0,61	0,38	0,33	0,17
Manga	1	18	5	12	0,52	0,47	0,17	0,08

Fonte: Elaborado pelo Autor.

4. RESULTADO E DISCUSSÃO

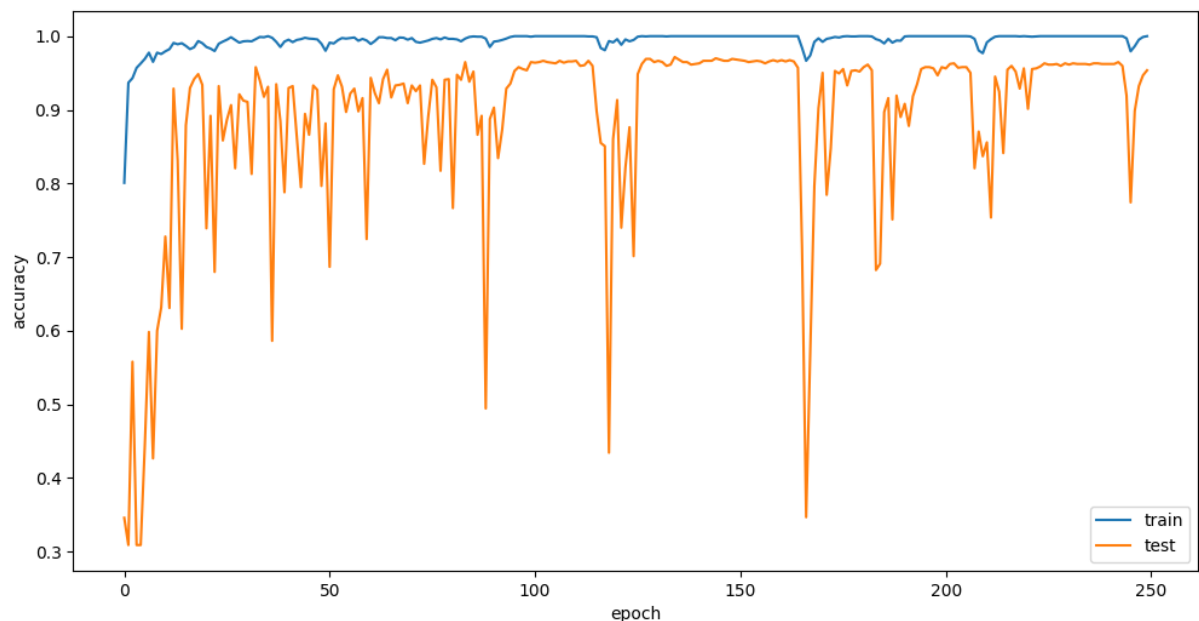
Nesse capítulo, onde discutiremos os resultados obtidos das redes neurais Convolucionais e foi dividido em 4 passos:

- Resultado e Discussões do Experimento 1;
- Resultado e Discussões do Experimento 2;
- Resultado e Discussões do Experimento 3;

4.1 Resultado e Discussões do Experimento 1

Com base no desenvolvimento proposto neste trabalho as duas Redes Neurais Convolucionais, que são: InceptionV3 e ResNet152 foram iniciadas com os pesos aleatórios, ou seja, não foi utilizado o aprendizado por transferência (*Transfer Learning*), lembrando que as métricas utilizadas são de Mohajon (2020). O treinamento da InceptionV3 pode ser visto na Figura 8, a matriz de confusão na Tabela 7 e os resultados das métricas na Tabela 8, já o treinamento da ResNet152 está presente na Figura 9, matriz de confusão na Tabela 9 e as métricas na Tabela 10.

Figura 8 - Treinamento da InceptionV3



Fonte: Elaborado pelo Autor.

Tabela 7 - Matriz de Confusão InceptionV3

	Covid-19	Pneumonia Viral	Normal
Covid-19	353	5	2
Pneumonia Viral	4	385	14
Normal	0	8	394

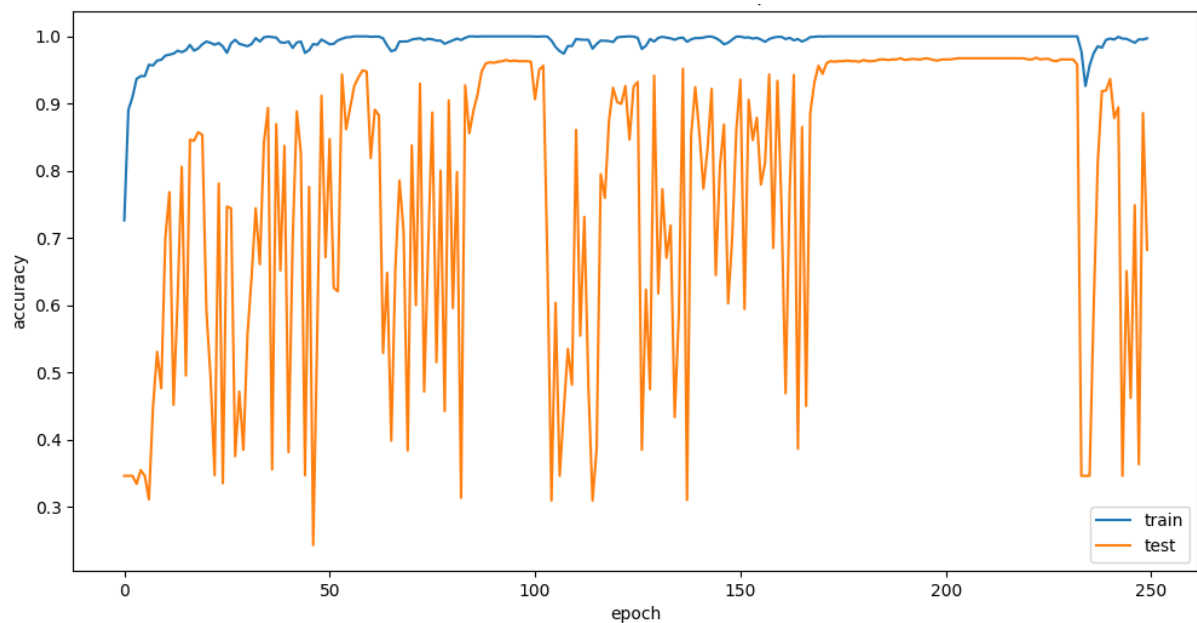
Fonte: Elaborado pelo Autor.

Tabela 8 - Tabela de Métricas InceptionV3

Classe	TP	TN	FP	FN	Accuracy	Misclassification	Precision	Recall
Covid-19	353	801	7	4	99.05%	0.009%	98.05%	98.87%
Pneumonia Viral	385	749	18	13	97.33%	0.026%	95.53%	96.73%
Normal	394	747	8	16	97.93%	0.020%	98.00%	96.09%

Fonte: Elaborado pelo Autor.

Figura 9 - Treinamento da ResNet152



Fonte: Elaborado pelo Autor.

Tabela 9 - Tabela de Confusão ResNet152

	Covid-19	Pneumonia Viral	Normal
Covid-19	355	3	2
Pneumonia Viral	4	377	22
Normal	1	5	396

Fonte: Elabora pelo Autor.

Tabela 10 - Tabela de Métricas ResNet152

Classe	TP	TN	FP	FN	Accuracy	Misclassification	Precision	Recall
Covid-19	355	800	5	5	99.14%	0.008%	98.61%	98.61%
Pneumonia Viral	377	754	26	8	97.08%	0.029%	93.54%	97.92%
Normal	396	739	6	24	97.42%	0.025%	98.50%	94.28%

Fonte: Elabora pelo Autor.

Como já mencionado, ambas as Redes Neurais Convolucionais foram testadas sem *Transfer Learning*, utilizando o total de 3 classes de imagens de pulmão em raio-x, que são: Covid-19, Pneumonia Viral e Normal.

A Figura 8 representa todo o treinamento da InceptionV3, a linha em azul é a taxa de acerto no conjunto de dados de treino e a linha laranja representa a taxa de acerto no conjunto de teste, durante o treinamento o *Callback ModelCheckpoint* foi executado em todas as épocas, porem a melhor época salva foi a 134 onde foi atingido 97.16% de acerto no conjunto de dados de teste. No começo do treinamento a taxa de acerto no conjunto de treino é baixa, pois o aprendizado das características das imagens está a ser ajustado, mas pode-se notar que logo após a décima época a linha azul que representa o conjunto de treino diminuiu a ociosidade se mantendo acima de 95%. O *Modelcheckpoint* é bem útil quando se busca o melhor resultado, salvar o melhor estado da rede durante o treinamento faz com que os resultados sejam satisfatórios, mas também pode ocasionar em *overfitting*. *Overfitting* é quando a rede acerta apenas dados do seu conjunto que foi treinado/testado, mas quando apresentado dados de outros conjuntos a rede atinge baixas taxas de acerto. As

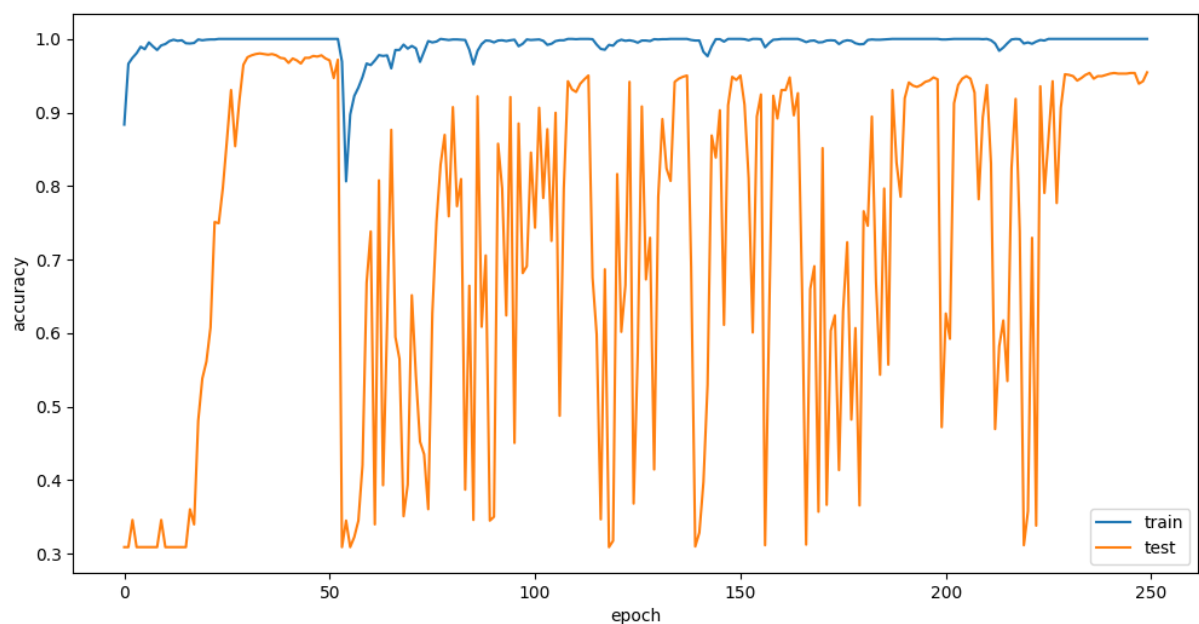
métricas obtidas da InceptionV3 sem a transferência de aprendizagem está presente na Tabela 8, pode-se notar que mesmo iniciando os pesos da InceptionV3 de forma aleatória, os resultados das métricas apresentadas são altas.

A ResNet152 tem resultados baixos no começo do treinamento, podemos ver na Figura 9 que no início a validação do conjunto de dados de teste se mantém ociosa de forma não positiva, porém ao contrário do treinamento da InceptionV3 pode-se notar que a partir da época 175 a 230 ela se manteve estável com resultados próximos de 96.00% de acerto, o *Modelcheckpoint* salvou a rede na época 222 onde ela obteve o seu maior desempenho sendo 96.82% de taxa de acerto no conjunto de dados teste. Nas métricas localizadas na Tabela 10 é notável que são altos valores de *accuracy*, *precision* e *recall*, e um baixo valor na *misclassification*.

4.2 Resultado e Discussões do Experimento 2

As redes usadas são as mesmas do primeiro experimento, mas diferente de antes foi utilizado a inicialização com os pesos do treinamento prévio do ImageNet disponível no próprio Keras.

Figura 10 - Treinamento da ResNet152 *Transfer Learning*



Fonte: Elaborado pelo Autor.

Tabela 11 - Tabela de Confusão ResNet152 *Transfer Learning*

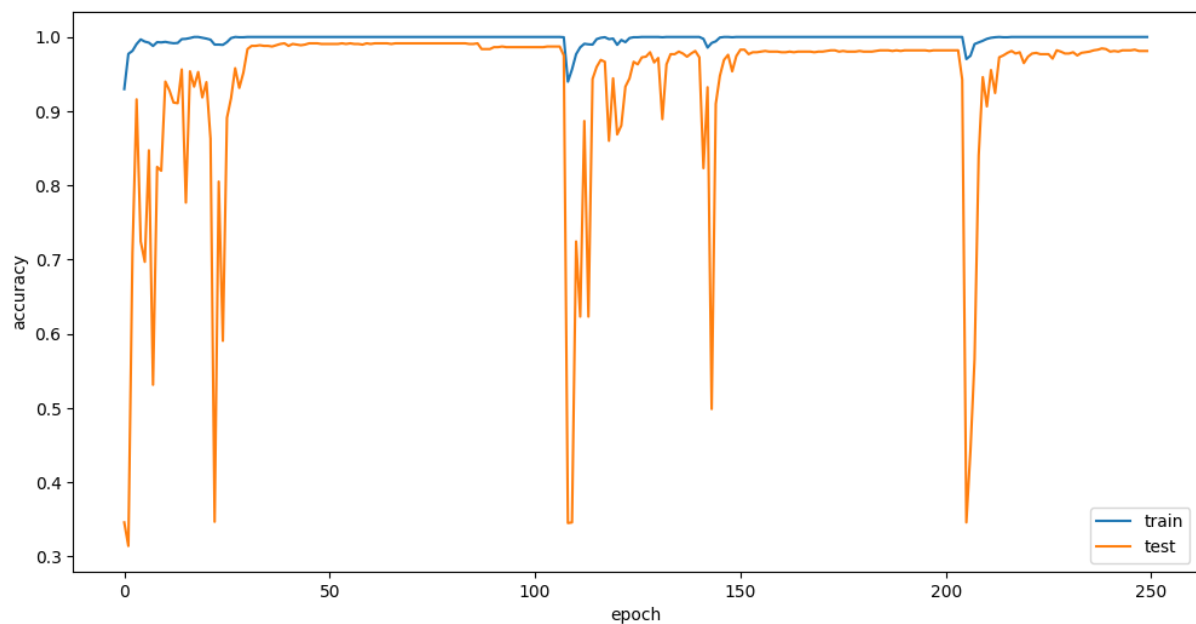
	Covid-19	Pneumonia Viral	Normal
Covid-19	353	5	2
Pneumonia Viral	4	385	14
Normal	0	8	394

Fonte: Elaborado pelo autor.

Tabela 12 - Tabela de Métricas ResNet152 *Transfer Learning*

Classe	TP	TN	FP	FN	Accuracy	Misclassification	Precision	Recall
Covid-19	359	803	1	2	99.74%	0.002%	99.72%	99.44%
Pneumonia Viral	384	758	19	4	98.02%	0.019%	95.28%	98.96%
Normal	399	746	3	17	98.28%	0.017%	99.25%	95.91%

Fonte: Elaborado pelo Autor.

Figura 11 - Treinamento da InceptionV3 *Transfer Learning*

Fonte: Elaborado pelo Autor.

Tabela 13 - Tabela de Confusão InceptionV3 *Transfer Learning*

	Covid-19	Pneumonia Viral	Normal
Covid-19	360	0	0
Pneumonia Viral	2	394	7
Normal	0	1	401

Fonte: Elaborado pelo Autor.

Tabela 14 - Tabela de Métricas InceptionV3 *Transfer Learning*

Classe	TP	TN	FP	FN	<i>Accuracy</i>	<i>Misclassification</i>	<i>Precision</i>	<i>Recall</i>
Covid-19	360	803	0	2	99.82%	0.001%	100%	99.44%
Pneumonia Viral	394	761	9	1	99.14%	0.008%	97.76%	99.74%
Normal	401	756	1	7	99.31%	0.006%	99.75%	98.28%

Fonte: Elaborado pelo autor.

A transferência de aprendizado é muito utilizada para treinamentos rápidos de redes convolucionais e para treinar redes que possuem uma quantidade baixa de amostras para o treinamento. Comparando a Figura 9 e a Figura 10 é notável a diferença da ResNet152 entre a inicialização aleatória e a transferência de aprendizado, na Figura 10 pode-se concluir que do início do treinamento até a vigésima época o treino estava muito abaixo, mas da vigésima época até a época 53 ela se manteve muito alta, entre 95% e 98% de taxa de acerto do conjunto de teste, mas a diante entrou em ociosidade até o final do seu treino onde se recuperou e permaneceu estável. O *ModelCheckpoint* salvou a rede na época 33 na qual a taxa de acerto do conjunto de teste era 98.02% um resultado acima da ResNet152 sem a transferência de aprendizagem que foi de 96.82% de acerto. A transferência de aprendizagem proporcionou um aumento na taxa de acerto em uma menor quantidade de épocas, confirmando que para o problema proposto o *transfer learning* trouxe benefícios. Observando a Tabela 10 da ResNet152 com a Tabela 12, pode-se notar

também um aumento nas métricas *accuracy*, *precision* e *recall* de Covid-19, e o *misclassification* obteve uma queda significativa.

A InceptionV3 contém o melhor resultado deste trabalho, na Figura 11 nota-se um melhor treinamento que o primeiro experimento, há estabilidades por maiores tempos, e um resultado bem acima de todos os experimentos, o *ModelCheckpoint* salvou a rede na época 39 com uma taxa de acerto de 99.14% do conjunto de teste. Em suas métricas na Tabela 14 a *accuracy* de todas as classes estão acima de 99.14%, em *precision* uma taxa de 100% para Covid-19, porem na Pneumonia Viral a rede se confundiu abaixando sua *precision* para 97.76%. De todas as amostras de Covid-19 a InceptionV3 acertou todas, mas se confundiu duas vezes nas amostras de Pneumonia Viral, afirmando ser Covid-19 e se confundiu sete vezes com um pulmão normal.

4.3 Resultado e Discussões do Experimento 3

Após os resultados do segundo experimento com a CNN InceptionV3 veio-se a dúvida de um possível *overfitting*, logo foi decidido o experimento 3, que consiste em encontrar um novo conjunto de dados, e com a InceptionV3 classificar todas as amostras do novo conjunto de dados.

O novo conjunto de dados selecionados foi desenvolvido por Raikote (2020) contendo um total de 317 imagens separadas em duas pastas, *train* e *test*, foi efetuado a junção da pasta *train* com a *test* totalizando 3 classes: Covid-19 com 132 imagens, Pneumonia Viral com 90 e Normal com 90. A matriz de confusão pode ser vista na Tabela 15, e as métricas na Tabela 16.

Tabela 15 - Tabela de Confusão - InceptionV3 - Experimento 3

	Covid-19	Pneumonia Viral	Normal
Covid-19	126	2	4
Pneumonia Viral	0	84	6
Normal	0	7	83

Fonte: Elaborado pelo autor.

Tabela 16 - Tabela de Métricas - InceptionV3 - Experimento 3

Classe	TP	TN	FP	FN	<i>Accuracy</i>	<i>Misclassification</i>	<i>Precision</i>	<i>Recall</i>
Covid-19	126	180	6	0	98.07%	0.019%	95.45%	100%
Pneumonia Viral	84	213	6	9	95.19%	0.048%	93.33%	90.32%
Normal	83	212	7	10	94.55%	0.054%	92.22%	89.24%

Fonte: Elaborado pelo autor.

Os resultados das métricas da Tabela 16 representa a InceptionV3 com transferência de aprendizagem após a classificação do conjunto de dados de Raikote (2020), pode-se concluir que o treinamento da InceptionV3 do segundo experimento não sofreu pela possibilidade de um *overfitting*, pelo fato de que as métricas *accuracy*, *precision* e *recall* da Tabela 16 pode-se considera-las altas, apesar da métrica *Misclassification* ser mais alta que os experimentos anteriores ela ainda é considerada como um valor adequado.

5. CONCLUSÃO

Este trabalho teve como objetivo demonstrar o potencial das CNNs em conjunto a radiografias pulmonares, classificando doenças pulmonares como: Covid-19 e Pneumonia Viral, ou a não existência dessas enfermidades.

De acordo com Santos (2020) a radiografia do tórax não é o único exame para o diagnóstico da Covid-19, mas é um exame que auxilia o médico a conceder um diagnóstico, logo essa pesquisa contribui em classificar radiografias pulmonares para auxiliar os médicos a definirem um diagnóstico a seus pacientes com suspeitas de Covid-19.

Os resultados dessa pesquisa colaboram diretamente com o uso de CNNs nas áreas médicas. Os experimentos propostos neste trabalho mostram resultados adequados de classificações de radiografia de tórax. No primeiro experimento apresentado já é notado um resultado de classificação positiva, mas no segundo experimento pode-se dizer que o uso de *transfer learning* fez uma diferença nos treinamentos e nos resultados. Mas com apenas os dois experimentos foi levantado a hipótese de um possível *overfitting*, essa possível hipótese é descartada no terceiro experimento, onde é posto à prova a rede InceptionV3, rede na qual obteve resultados de 99.82% de *accuracy* na classe Covid-19 nas métricas propostas.

A eficiência de uma CNN é afetada diretamente pelo seu *dataset* de treino/teste, baixas quantidades de imagens de treino podem levar a um resultado nada satisfatório. Um conjunto de dados completo com uma quantidade alta de imagens, definitivamente apresentará resultados mais satisfatórios. Em um mundo ideal, onde se é acreditado no potencial das CNNs para classificações de imagens médicas, eventualmente o mundo vivenciaria um caminho diferente quando se trata dessa adversidade pandêmica.

Todo o código desenvolvido e os conjuntos de dados podem ser visualizados no GitHub: github.com/lockade/TCC_Gustavo_Jefferson.

5.1 Trabalhos Futuros

As sugestões propostas aqui para trabalhos futuros, é a criação de um *dataset* composto por uma grande quantidade de imagens de radiografias de tórax, com o

objetivo de generalizar a classificação em qualquer *dataset*. Sendo possível a oportunidade de se colocar um sistema de classificação em situações reais.

REFERÊNCIAS

ABADI, Martín et al. **TensorFlow: A System for Large-Scale Machine Learning**. 12th USENIX Symposium on Operating Systems Design and Implementation, USENIX, p. 265 - 283, 2 nov. 2016. DOI ISBN 978-1-931971-33-1. Disponível em: <https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf>. Acesso em: 14 fev. 2021.

BUSSON, Antonio; FIGUEIREDO, Lucas; SANTOS, Gabriel Noronha Pereira; DAMASCENO, Andre; COLCHER, Sérgio; MILIDIÚ, Ruy Luiz. **Desenvolvendo Modelos de Deep Learning para Aplicações Multimídia no Tensorflow**. In: MINICURSOS do XXIV Simpósio Brasileiro de Sistemas Multimídia e Web. 978-85-7669-455-7. ed. [S. l.]: SBC, 2018. p. 67-116. Disponível em: https://www.researchgate.net/publication/328341764_Desenvolvendo_Modelos_de_Deep_Learning_para_Aplicacoes_Multimidia_no_Tensorflow. Acesso em: 13 fev. 2021.

CANONICAL (org.). **Ubuntu**. 2004. Disponível em: <https://ubuntu.com/>. Acesso em: 15 fev. 2021.

CARNEIRO, TIAGO et al. Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications. **SPECIAL SECTION ON TRENDS, PERSPECTIVES AND PROSPECTS OF MACHINE LEARNING APPLIED TO BIOMEDICAL SYSTEMS IN INTERNET OF MEDICAL THINGS**, IEEE, 8 out. 2018. DOI 10.1109/ACCESS.2018.2874767. Disponível em: <https://ieeexplore.ieee.org/document/8485684>. Acesso em: 14 fev. 2021.

CHOLLET, Francois. **Keras**. 2015. Disponível em: <https://github.com/fchollet/keras>. Acesso em: 13 jan. 2021.

CHOWDHURY, Muhammad E. H. et al. **Can AI Help in Screening Viral and COVID-19 Pneumonia?** Ieee Access, [S.L.], v. 8, p. 132665-132676, 2020. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/access.2020.3010287>.

HE, Kaiming; SUN, Jian; REN, Shaoqing; ZHANG, Xiangyu. **Deep Residual Learning for Image Recognition**. Computer Vision and Pattern Recognition (cs.CV), Cornell University, 10 dez. 2015. Disponível em: <https://arxiv.org/abs/1512.03385>. Acesso em: 3 fev. 2021.

KINGMA, Diederik P.; BA, Jimmy Lei. **Adam: A Method for Stochastic Optimization**. International Conference on Learning Representations - ICLR, San Diego, Califórnia, 26 dez. 2014. Paper Científico, p. 1 - 15. Disponível em: <https://arxiv.org/abs/1412.6980v9>. Acesso em: 8 fev. 2021.

LECUN, Yann; BOTTOU, Léon; BENGIO, Yoshua; HAFFNER, Patrick. **Gradient-Based Learning Applied to Document Recognition**. Proceedings of the IEEE, Estados Unidos, novembro 1998. DOI 10.1109/5.726791. Disponível em: <http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf>. Acesso em: 19 nov. 2020.

LOPES R. LINO, Tiago Alexandre. **INTELIGÊNCIA ARTIFICIAL, HUMANA E A EMOÇÃO**. 2004. 6 p. Paper científico (3º ano de Psicologia) - Universidade Autónoma de Lisboa, Lisboa, Portugal, 2004. Disponível em: <https://www.psicologia.pt/artigos/textos/TL0012.PDF>. Acesso em: 2 fev. 2021.

MARQUES, Ogê FILHO; VIEIRA, Hugo NETO. **Processamento Digital de Imagens**. Rio de Janeiro: Brasport, 1999. 307 p. v. 1. ISBN 8574520098. DOI ISBN 8574520098. Disponível em: <https://www.ogemarques.com/wp-content/uploads/2014/11/pdi99.pdf>. Acesso em: 23 dez. 2020.

MEIRELLES, Gustavo de Souza Portes. **COVID-19: uma breve atualização para radiologistas**. Radiologia Brasileira, São Paulo, v. 53, n. 5, p. 320-328, 02 out. 2020. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/0100-3984.2020.0074>.

MOHAJON, Joydwip. **Confusion Matrix for Your Multi-Class Machine Learning Model**. Medium.com, [s. l.], 29 maio 2020. Disponível em: <https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826>. Acesso em: 11 fev. 2021.

OSÓRIO, Fernando Santos; BITTENCOURT, João Ricardo. **Sistemas Inteligentes baseados em Redes Neurais Artificiais aplicados ao Processamento de Imagens**. I WORKSHOP DE INTELIGÊNCIA ARTIFICIAL UNISC – Universidade de Santa Cruz do Sul Departamento de Informática, Universidade de Santa Cruz do Sul, junho 2000.

PAN, Sinno Jialin; YANG, Qiang. **A Survey on Transfer Learning**. IEEE Computer Society, [S. l.], v. 22, n. 10, p. 1345-1359, 10 out. 2010.

PASSOS, Rafael de Campos. **Uso de técnicas de deep learning como ferramenta de criação de datasets para análise de evidências forenses digitais**. Brasília: IFB, 2019. 86 f.: il.

PERKOVIC, Ljubomir. **Introdução à Computação Usando Python - Um Foco no Desenvolvimento de Aplicações**. 1ª. ed. [S. l.]: LTC, 1 de abril 2016. ISBN 978-8521630814.

PRODANOV, Cleber Cristiano; FREITAS, Ernani Cesar. **Metodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico**. 2ª. ed. rev. e atual. [S. l.]: Editora Feevale, 2013. 277 p. ISBN 8577171582, 9788577171583.

RAIKOTE, Pranav. Covid-19 Image Dataset. 2020. Disponível em: <https://www.kaggle.com/pranavraikokte/covid19-image-dataset>. Acesso em: 13 fev. 2021.

RODRIGUES, Diego Alves. **Deep Learning e Redes Neurais Convolucionais: Reconhecimento Automático de Caracteres em Placas de Licenciamento Automotivo**. Orientador: Prof. Dr. Leonardo Vidal Batista. 2018. Trabalho de conclusão de curso (Bacharel em Ciência da Computação) - Universidade Federal da Paraíba, Paraíba, 2018.

ROSA, Nícolas dos Santos. **Estimação monocular de profundidade por aprendizagem profunda para veículos autônomos: influência da esparsidade dos mapas de profundidade no treinamento supervisionado**. Orientador: Prof. Dr. Valdir Grassi Junior. 2019. Dissertação (Mestrado em Sistemas Dinâmicos) - Escola de Engenharia de São Carlos da Universidade de São Paulo, São Carlos, 2019. Disponível em: <https://www.teses.usp.br/teses/disponiveis/18/18153/tde-13092019-150449/pt-br.php>. Acesso em: 2 fev. 2021.

RUSSELL, Stuart; NORVING, Peter. **Inteligência Artificial**. 3. ed. Rio de Janeiro: Elsevier, 2013.

SANTOS, Maria Tereza. **Coronavírus: como raio-x e tomografia ajudam no diagnóstico e no controle: Exames de imagem também podem participar da detecção e do acompanhamento de casos de Covid-19. Entenda como e em quais situações eles são indicados...** Veja Saúde, [s. l.], 24 ago. 2020. Disponível em: <https://saude.abril.com.br/medicina/coronavirus-como-raio-x-e-tomografia-ajudam-no-diagnostico-e-no-controle/>. Acesso em: 2 fev. 2021.

SILVA, Luan; FERREIRA, Victor; ARAÚJO, Leandro. **Aplicação de Deep Learning no pré-diagnóstico da COVID-19 através de imagens de raio-x**. 2020. 11 f. Monografia (Especialização) - Curso de Sistemas de Informação, Universidade Federal do Sul e Sudeste do Pará, Marabá, 2020.

WALDROP, M. Mitchell. **News Feature: What are the limits of deep learning?** Proceedings of the National Academy of Sciences, [s. l.], 17 jan. 2019. DOI 10.1073/pnas.1821594116. Disponível em: <https://www.pnas.org/content/116/4/1074>. Acesso em: 12 fev. 2021.

WERNECK, Guilherme Loureiro; CARVALHO, Marília Sá. **A pandemia de COVID-19 no Brasil: crônica de uma crise sanitária anunciada**. Cad. Saúde Pública, Rio de Janeiro, v.36, n.5, e00068820, 2020. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-311X2020000500101&lng=en&nrm=iso. Acesso em: 16 Fev. 2021. Epub maio 08, 2020. <https://doi.org/10.1590/0102-311x00068820>.

WOJNA, Zbigniew; SHLENS, Jonathon; SZEGEDY, Christian; VANHOUCKE, Vincent; IOFFE, Sergey. **Rethinking the Inception Architecture for Computer Vision. Computer Vision and Pattern Recognition (cs.CV)**, Cornell University, 2 dez. 2015. Disponível em: <https://arxiv.org/abs/1512.00567>. Acesso em: 3 fev. 2021.