



Universidade do Estado do Rio de Janeiro
Centro de Tecnologia e Ciências
Faculdade de Engenharia

COSTA, I.G.G.

**Uso de Rede Neural Artificial para auxiliar no diagnóstico de
COVID-19 através de exame de Raio-X de Tórax**

Rio de Janeiro

2022

COSTA, I.G.G.

**Uso de Rede Neural Artificial para auxiliar no diagnóstico de COVID-19
através de exame de Raio-X de Tórax**

Projeto de graduação apresentado, como requisito parcial para obtenção do título de Engenheiro Eletricista, ênfase Sistemas e Computação, à Faculdade de Engenharia, da Universidade do Estado do Rio de Janeiro.



Orientador: Profa. Dra. Karla Tereza Figueiredo Leite
Coorientador: Profa. Dra. Luiza de Macedo Mourelle

Rio de Janeiro
2022

CATALOGAÇÃO NA FONTE

UERJ / REDE SIRIUS / BIBLIOTECA CTC/B

C837 Costa, Ian Germano Guimarães.

Uso de rede neural artificial para auxiliar no diagnóstico de COVID-19 através do exame de raio-x de tórax / Ian Germano Guimarães Costa. – 2022.

43f.

Orientadora: Karla Tereza Figueiredo Leite.

Coorientadora: Luiza de Macedo Mourelle

Projeto final (Graduação) – Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia.

1. Engenharia elétrica - Monografias. 2. Redes neurais (Computação) - Monografias. 3. Aprendizado do computador - Monografias. 4. Diagnóstico - Monografias. 5. Exame pelo raio X - Monografias. I. Leite, Karla Tereza Figueiredo. II. Mourelle, Luiza de Macedo. III. Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia. IV. Título.

CDU 621.3

COSTA, I.G.G.

**Uso de Rede Neural Artificial para auxiliar no diagnóstico de COVID-19
através de exame de Raio-X de Tórax**

Projeto de graduação apresentado, como requisito parcial para obtenção do título de Engenheiro Eletricista, ênfase Sistemas e Computação, à Faculdade de Engenharia, da Universidade do Estado do Rio de Janeiro.

Aprovada em 26 de Abril de 2022.

Banca Examinadora:

Profa. Dra. Karla Tereza Figueiredo Leite (Orientadora)
IME – UERJ

Profa. Dra. Luiza de Macedo Mourelle (Coorientadora)
DESC – UERJ

Prof. Dr. João Araujo Ribeiro
DESC - UERJ

Rio de Janeiro
2022

DEDICATÓRIA

Pelo carinho, afeto, dedicação e cuidado que meus pais me deram durante toda a minha existência, dedico esta monografia a eles. Com muita gratidão.

AGRADECIMENTOS

A todos os meus professores, pelas correções e ensinamentos que me permitiram apresentar um melhor desempenho no meu processo de formação profissional ao longo do curso.

A persistência realiza o impossível.

RESUMO

COSTA, I.G.G. *Uso de Rede Neural Artificial para auxiliar no diagnóstico de COVID-19 através de exame de Raio-X de Tórax.* 2022. 43 f. Projeto Final (Graduação em Engenharia Elétrica, ênfase em Sistemas e Computação) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2022.

Com a crise sanitária global, iniciada no ano de 2019 e originada pela proliferação do vírus batizado de COVID-19, tem-se procurado métodos mais efetivos para o combate e redução de danos causados pela pandemia do COVID-19. Concomitantemente, o avanço da tecnologia da informação possibilitou a exploração e aplicação mais amplas de tecnologias inspiradas no cérebro humano como as redes neurais. Tendo em vista explorar as novas aplicações tecnológicas para resolver problemas atuais, nesse trabalho propõe-se um modelo baseado em arquiteturas de aprendizagem profunda, capaz de classificar casos de COVID-19 utilizando imagens de exames de raios-X de tórax de pacientes com e sem a doença a fim de automatizar o diagnóstico.

Palavras-chave: Redes neurais. Aprendizagem profunda. COVID-19.

ABSTRACT

COSTA, I.G.G. *Use of Artificial Neural Network to aid in the diagnosis of COVID-19 through Chest X-Ray examination.* 2022. 43 f. Projeto Final (Graduação em Engenharia Elétrica, ênfase em Sistemas e Computação) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2022.

With the global health crisis that began in 2019, caused by the proliferation of the virus named COVID-19, more effective methods have been sought to combat and reduce the damage caused by the COVID-19 pandemic. Concomitantly, the advance of information technology has enabled the exploration and broader application of technologies inspired by the human brain, such as neural networks. With a view to exploring new technological applications to solve current problems, this work proposes a model based on Deep Learning architectures, capable of classifying COVID-19 cases using images from chest X-ray exams of patients with and without the disease in order to automate the diagnosis.

Keywords: Neural network. Deep learning. COVID-19.

LISTA DE ILUSTRAÇÕES

Figura 1 - Exemplo de imagens RXT: (A) sem COVID-19	13
Figura 2 - Modelo básico de neurônio	15
Figura 3 - Rede neural totalmente conectada genérica	18
Figura 4 - Rede neural convolucional genérica	19
Figura 5 - Rede neural residual genérica	19
Figura 6 - Arquitetura ResNet50	22
Figura 7 - Arquitetura proposta em Wang <i>et al</i>	24
Figura 8 - Arquitetura proposta em Jain <i>et al</i>	25
Figura 9 - Arquitetura proposta em Haque <i>et al</i>	26
Figura 10 - Arquitetura completa da rede implementada	29
Figura 11 - Acurácia do modelo	31
Figura 12 - Perda (função erro) do modelo	32
Figura 13 - Matriz de confusão	33

SUMÁRIO

	INTRODUÇÃO	10
1	TIPOS DE DIAGNÓSTICOS COVID-19	12
1.1	Vantagens do Raio-X de Tórax (RXT)	13
2	REDES NEURAIS ARTIFICIAIS	15
2.1	Neurônio Artificial	15
2.1.1	<u>Neurônios de entrada</u>	16
2.1.2	<u>Neurônios escondidos</u>	16
2.1.3	<u>Neurônios de saída</u>	16
2.2	Camadas de processamento	16
2.2.1	<u>Camada de achatamento (<i>flatten layer</i>)</u>	17
2.2.2	<u>Camada de <i>dropout</i></u>	17
2.3	Tipos de redes neurais	17
2.3.1	<u>Redes Neurais Totalmente Conectadas</u>	17
2.3.2	<u>Redes Neurais Convolucionais</u>	18
2.3.3	<u>Redes Neurais Residuais</u>	19
2.4	Etapas do treinamento de redes neurais	20
2.4.1	<u>Inicialização da rede</u>	20
2.4.2	<u>Alimentação para frente (<i>feed foward</i>)</u>	20
2.4.3	<u>Alimentação para trás (<i>feed backward</i>)</u>	20
2.4.4	<u>Definição dos Parâmetros e Hiper-parâmetros da rede</u>	20
2.5	Transferência de aprendizado em redes neurais	21
3	TRABALHOS RELACIONADOS	23
3.1	COVID-Net: rede neural convolucional	23
3.2	Transferência de aprendizado ResNet50 e ResNet101	24
3.3	Aprendizado profundo para detecção de COVID-19	25
4	IMPLEMENTAÇÃO	27
4.1	Coleta e pré-processamento dos dados	27
4.2	Modelagem da arquitetura da rede neural	28
4.3	Definição de hiper-parâmetros e parâmetros do modelo	28
4.4	Treinamento da rede neural	29
5	ANÁLISE DE RESULTADOS	31
6	CONCLUSÃO	34
	REFERÊNCIAS	35
	GLOSSÁRIO	37
	ANEXO A – Códigos em Python do projeto e repositório GitHub	38

INTRODUÇÃO

Em 2019, surgiu uma nova doença em diversas regiões do mundo que rapidamente se espalhou por todo o planeta, tornando-se uma pandemia. Descobriu-se que o agente patológico responsável era um vírus identificado como SARS-CoV-2 da família dos coronavírus [1]. Coronavírus são uma grande família de vírus que se assemelham geneticamente. Um vírus dessa família sofreu mutação e conseguiu infectar seres humanos, designado, então SARS-CoV-2. A família coronavírus também está presente em diferentes espécies de animais como os camelos, gados, cachorros, gatos e morcegos. Raramente, os coronavírus que infectam animais podem infectar pessoas. O primeiro caso notificado foi em dezembro de 2019. Em Wuhan, na China, houve a transmissão de um novo coronavírus SARS-CoV-2 que ocasionou a doença COVID-19, sendo em seguida disseminada e transmitida pessoa a pessoa.

A COVID-19 é uma doença causada pelo vírus da família coronavírus, denominado SARS-CoV-2. A COVID-19 apresenta um espectro clínico variado, apresentando-se desde infecções assintomáticas a quadros graves que podem envolver severas dificuldades respiratórias. As estatísticas gerais sobre a COVID-19, conforme a Organização Mundial de Saúde (OMS), são que a maioria (cerca de 80%) dos pacientes com COVID-19 podem ser assintomáticos ou oligossintomáticos (poucos sintomas), e a minoria (cerca de 20%) dos casos detectados, requerem atendimento hospitalar por apresentarem dificuldades respiratórias, podendo precisar de suporte ventilatório em casos mais graves [1].

O objetivo desse trabalho de conclusão de curso é automatizar o diagnóstico por RXT de pacientes com suspeita de COVID-19. Para tal, uma grande quantidade de dados, gerados pelos exames de RXT, será utilizada para ajustar um modelo baseado em redes neurais profundas. Isto permitirá a classificação de imagens de COVID-19 e não COVID-19.

A metodologia de pesquisa utilizada é a de pesquisa exploratória, que visa explorar e ampliar os conhecimentos nas áreas de redes neurais e aprendizado profundo a fim de possibilitar a implementação do projeto proposto. As etapas da implementação do projeto são as seguintes: coleta e pré-processamento dos dados utilizados; modelagem da arquitetura da rede neural; escolha de hiper-parâmetros e parâmetros da rede neural; treinamento da rede neural e a análise de desempenho da rede.

A estrutura desse trabalho de conclusão de curso consiste nos seguintes capítulos: Capítulo 1 apresenta diferentes tipos de diagnósticos usados para detectar COVID-19; Capítulo 2 consiste em uma apresentação geral dos conceitos teóricos estudados para a resolução do problema proposto; Capítulo 3 faz um apanhado geral dos trabalhos relacionados na área dando destaque a três artigos publicados escolhidos para apresentação e estudo; Capítulo 4 é abordado o passo a passo da implementação; Capítulo 5 é feita

uma análise dos resultados obtidos na implementação do trabalho; Capítulo 6 finaliza o trabalho trazendo as conclusões obtidas.

1 TIPOS DE DIAGNÓSTICOS COVID-19

Até o presente momento existem vários métodos para se realizar o diagnóstico de COVID-19, tais como: Diagnóstico clínico; Diagnóstico laboratorial; Teste rápido; Raio-X. Uma breve descrição de cada um é apresentada a seguir.

- Diagnóstico clínico

COVID-19 pode ocasionar sintomas como febre em casos iniciais da doença. O diagnóstico clínico investiga o quadro inicial do paciente para ver se se encaixa com os sintomas da doença. O quadro inicial de COVID-19 pode ser classificado como síndrome gripal [2-4].

- Diagnóstico laboratorial

No diagnóstico laboratorial, coletam-se amostras das mucosas dos pacientes para serem analisadas para a detecção de genoma viral do SARS-CoV-2 [2-4].

- Teste rápido

Nesse tipo de diagnóstico, coletam-se amostras como soro, plasma ou sangue do paciente para detectar a presença de anticorpos especializados em combater o vírus SARS-CoV-2 na amostra. O teste rápido serve como ferramenta auxiliar para um diagnóstico final [2-4].

- Raio-X

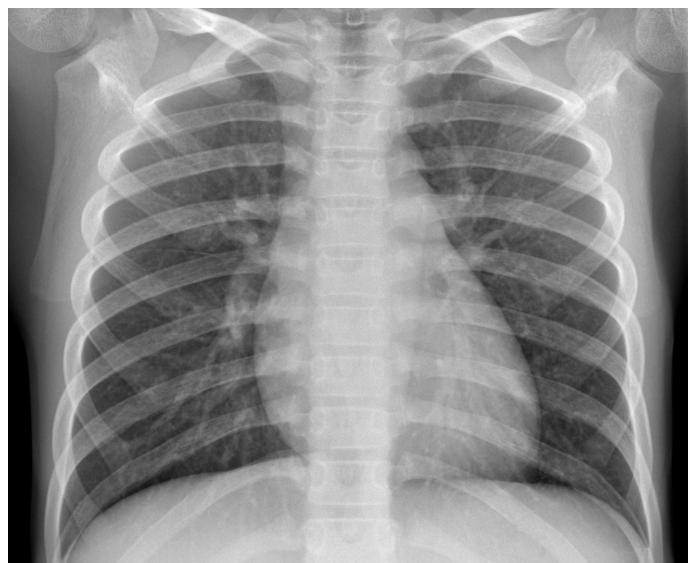
A radiografia desempenha papel fundamental no gerenciamento da pandemia COVID-19 [2-4]. A figura 1 mostra dois exemplos de Raio-X de tórax, com e sem COVID-19. Os radiologistas, que estão na linha de frente do surto, descobriram que a radiografia de tórax é um complemento útil para diagnosticar pacientes com COVID-19 e avaliar seu quadro atual.

Como o Raio-X gera uma imagem em uma chapa, essa chapa pode ser digitalizada. Em termos de informação, as imagens em si são grandes volumes de dados. Logo, o Raio-X torna-se o diagnóstico mais suscetível à automação por redes neurais. Além disso, possuem as seguintes vantagens citadas abaixo.

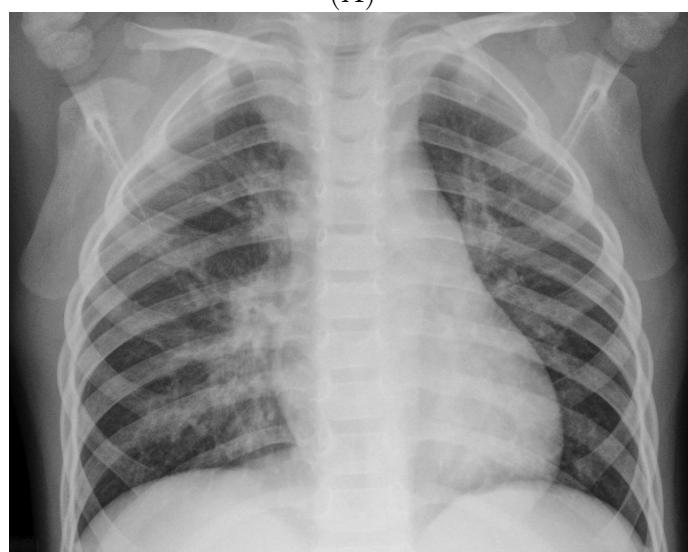
1.1 Vantagens do Raio-X de Tórax (RXT)

- Facilidade na triagem de pacientes

O exame RXT pode ser aplicado aos pacientes suspeitos de COVID-19 em paralelo com os testes de laboratório que, em geral, levam mais tempo para serem concluídos. Em caso de detecção de lesões compatíveis com padrões de COVID-19 no RXT, já se pode isolar o paciente e esperar pela confirmação do exame de laboratório. Além disso, os RXTs realizados quando os pacientes se apresentam na sala de emergência com sintomas de COVID-19, podem ajudar a prever a gravidade da doença [2-4].



(A)



(B)

Figura 1 - Exemplo de imagens RXT: (A) sem COVID-19, (B) com COVID-19

- Baixo custo e versatilidade

Os exames de Raio-X estão disponíveis na maioria das unidades de saúde e requerem equipamentos mais baratos e com menor custo de manutenção, tornando-os mais acessíveis do que os outros tipos de exames. Os exames de RXT fazem parte do procedimento padrão de pacientes com suspeitas de doenças pulmonares, logo esses exames podem ser usados também para detectar COVID-19, já que esta predominantemente se manifesta no aparelho respiratório [2-4].

- Portabilidade

O Raio-X móvel permite que a imagem seja levada ao leito do paciente, o que confere enormes vantagens de versatilidade para o paciente. Isso mantém os pacientes infectados fora do departamento de radiologia do hospital e evita a necessidade de higienizar as salas de radiologia após o uso [2-4].

- Diagnóstico por Redes Neurais Artificiais

As Redes Neurais Artificiais (RNA) são uma excelente ferramenta usada para auxiliar o diagnóstico médico. A tecnologia consegue discriminar e identificar diferentes padrões em imagens, contribuindo para diagnósticos mais acurados. Com elas é possível reconhecer padrões em dezenas de milhares de imagens, com uma precisão comparada a de um médico especialista, porém em um menor tempo de avaliação. Assim, pode ser usada como uma ferramenta que auxilia o profissional, não dispensando a avaliação do médico para assegurar que as alterações apresentadas nas imagens condizem com as condições clínicas do paciente.

Desse modo, entende-se que o potencial para aplicação de redes neurais na área da saúde é gigantesco e deve trazer ainda mais benefícios para a área. O principal objetivo tem sido priorizar o atendimento ao paciente, proporcionando maior efetividade na entrega de resultados, precisão no diagnóstico, segurança para as informações dos pacientes, entre outros recursos. Otimizando o trabalho nas instituições de saúde, o paciente será o primeiro a ser beneficiado. Motivado pela necessidade de desenvolver soluções práticas e eficientes para ajudar no combate a pandemia do COVID-19, e tendo em vista o grande número de dados gerados pelos exames médicos, tem-se um cenário ideal para o desenvolvimento de aplicações de alta tecnologia que se aproveitam de grandes volumes de dados como as arquiteturas de aprendizagem profunda.

2 REDES NEURAIS ARTIFICIAIS

As redes neurais artificiais são modelos computacionais inspirados na estrutura do cérebro animal. Os cérebros biológicos são compostos por redes de neurônios biológicos capazes de processar a informação do ambiente a sua volta, reconhecer padrões, armazenar informações e aprender com as experiências vivenciadas. Os cérebros biológicos também utilizam certos grupos de neurônios especializados para realizarem tarefas específicas, por exemplo, os neurônios do córtex visual para a interpretação de imagens.

2.1 Neurônio Artificial

As redes neurais artificiais são compostas de neurônios artificiais, capazes de processar informações de um ambiente virtual, reconhecer padrões em informações digitais e aprender a realizar tarefas específicas equivalentes a um processamento de informação de alta complexidade. Existe uma diversidade de tipos de neurônios artificiais, mas geralmente cada neurônio artificial possui um valor numérico associado, uma função de ativação e ligações com outros neurônios. Essas ligações também podem ser chamadas pesos e possuem valor numérico associado. A figura 2 ilustra um modelo básico de neurônio com seus cálculos de processamento.

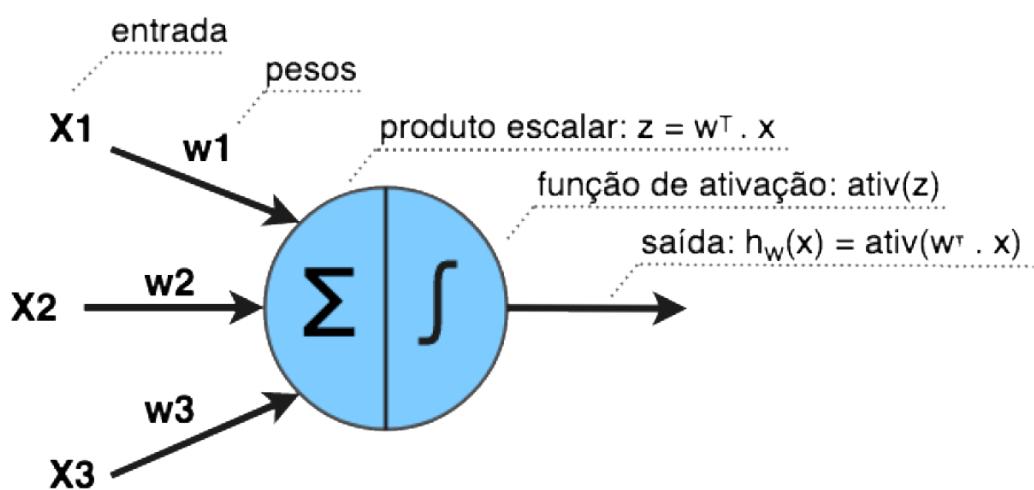


Figura 2 - Modelo básico de neurônio

Para a execução de um processamento básico de um neurônio artificial é feita uma série de cálculos: na primeira etapa, é feito um produto escalar entre a matriz transposta de pesos e a matriz de valores de entrada; em seguida, o resultado da operação matricial é submetido a uma função de ativação e o valor gerado pela função de ativação é a saída

do neurônio. Os neurônios podem ser agrupados e organizados em camadas, sendo que, devido à natureza das arquiteturas utilizadas nesse trabalho, não serão incluídas aquelas que permitem conexão entre neurônios na mesma camada. Os seguintes tipos de neurônios artificiais serão abordados no trabalho.

2.1.1 Neurônios de entrada

Pertencem à camada de entrada (*input layer*) das redes neurais os neurônios que recebem as informações de entrada. Essas informações de entrada tanto podem ser saídas de uma outra rede ou as informações de entrada do problema. Possuem um valor numérico associado, função de ativação e se ligam com outros neurônios da rede para mandar informações, só recebem as informações da entrada da rede.

2.1.2 Neurônios escondidos

Pertencem às camadas escondidas (*hidden layer*) das redes neurais os neurônios caracterizados tanto por receber como mandar informações para outros neurônios. Possuem valor numérico associado e função de ativação.

2.1.3 Neurônios de saída

Pertencem à camada de saída (*output layer*) das redes neurais os neurônios que recebem informações de outros neurônios, fazem o processamento final e mandam o resultado final para a saída do programa ou para a camada de entrada de outra rede. Possuem valor numérico associado e função de ativação.

2.2 Camadas de processamento

Certos tipos de camadas não possuem neurônios, mas executam funções específicas sobre outras camadas. Nesse trabalho foram usadas dois tipos dessas camadas, sucintamente apresentados nas próximas subseções.

2.2.1 Camada de achatamento (*flatten layer*)

Recebe um tensor n-dimensional como entrada e transforma esse tensor em um único vetor de uma dimensão. Usada para conectar redes convolucionais com redes totalmente conectadas.

2.2.2 Camada de *dropout*

A camada de *dropout* define aleatoriamente valores de entrada para 0 com uma certa probabilidade durante o tempo de treino, ajudando a evitar o *overfitting* ou sobreajuste, que significa que o modelo utilizado se ajusta muito bem ao conjunto de dados anteriormente treinado, mas se mostra ineficaz para prever novos resultados. As entradas não ajustadas a 0 são aumentadas em $1/(1 - \text{probabilidade})$ de modo que a soma sobre todas as entradas permaneçam inalteradas.

2.3 Tipos de redes neurais

Existem diferentes tipos de estruturas de redes neurais, todos os tipos de redes neurais existentes usam a estrutura básica do neurônio, mas se diferenciam na forma como as ligações entre os neurônios ocorrem e como são processadas essas ligações. Nesse trabalho foram abordados três tipos de estruturas de redes neurais, são elas.

2.3.1 Redes Neurais Totalmente Conectadas

Redes Neurais Totalmente Conectadas (RNTC) são compostas por uma camada de entrada, uma ou mais camadas escondidas e uma camada de saída. Sendo que todo neurônio de cada camada escondida é conectado a todos os neurônios das camadas anteriores e posteriores [8-9]. A figura 3 ilustra uma RNTC genérica, onde os neurônios em amarelo pertencem à camada de entrada, os neurônios em verde pertencem às camadas escondidas e os neurônios em laranja pertencem à camada de saída.

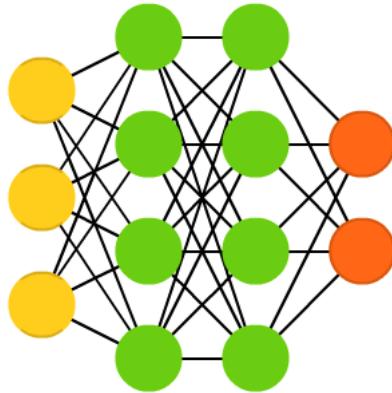


Figura 3 - Rede neural totalmente conectada genérica [14]

2.3.2 Redes Neurais Convolucionais

Redes Neurais Convolucionais (RNC) são redes que simulam o comportamento dos neurônios ligados ao córtex visual, no quesito de encontrar padrões através de cálculos convolucionais. Nas RNCs cada neurônio recebe um estímulo somente da sua área de estímulo, ou seja, cada neurônio é conectado a um conjunto de neurônios específico, e sua ativação é executada a partir de um cálculo convolucional envolvendo o grupo de neurônios específicos conectados. Nas RNC os pesos (ligações entre os neurônios) são chamados de filtros [8-9]. A figura 4 ilustra uma RNC genérica, onde os neurônios em amarelo pertencem à camada de entrada, os neurônios da camada convolucional aparecem em rosa com circulo preto, os neurônios em verde pertencem às camadas escondidas e os neurônios em laranja pertencem à camada de saída. Pode-se notar que as camadas convolucionais reduzem a quantidade de informação filtrando essas informações para as camadas escondidas como observado na figura 4.

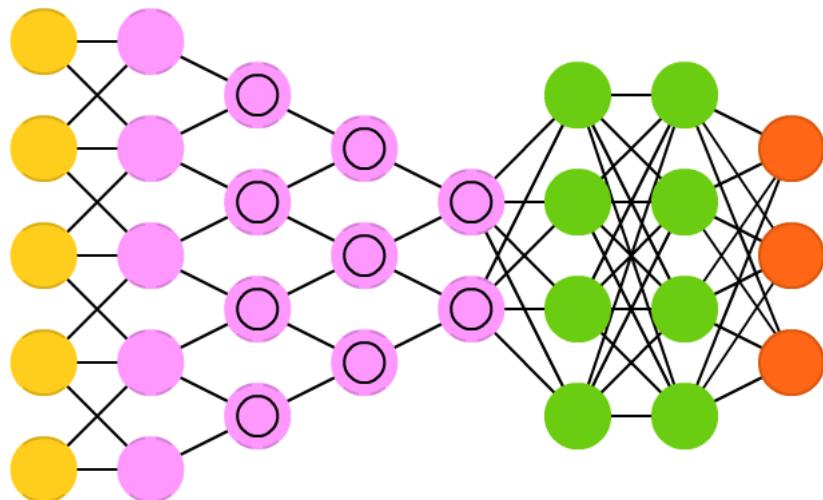


Figura 4 - Rede neural convolucional genérica [14]

2.3.3 Redes Neurais Residuais

Redes Neurais Residuais (RNR) são redes neurais construídas com inspiração em células piramidais do córtex cerebral, as redes residuais apresentam conexões internas adicionais entre camadas de neurônios não adjacentes, essas conexões adicionais criam conexões “atalho” entre camadas que pulam uma ou mais camadas intermediárias. Uma das razões para se usar as redes neurais residuais é que elas evitam o problema do desaparecimento do gradiente e mitigam o problema de saturação de acurácia, por outro lado, é necessário mais dados para treina-las. A figura 5 ilustra uma RNR genérica, onde os neurônios em amarelo pertencem à camada de entrada, os neurônios em verde pertencem às camadas escondidas que possuem conexões “atalho” e os neurônios em laranja pertencem à camada de saída.

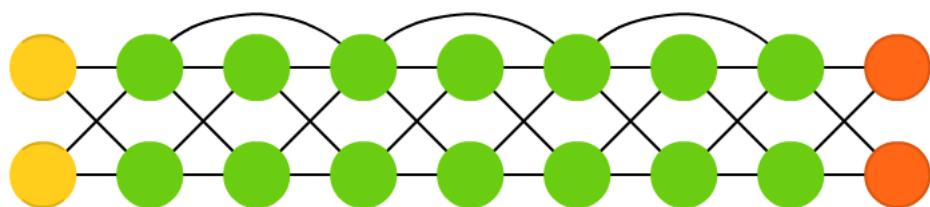


Figura 5 - Rede neural residual genérica [14]

2.4 Etapas do treinamento de redes neurais

No treinamento de redes neurais existem várias formas de treinamentos como o treinamento supervisionado, treinamento não supervisionado e treinamento por reforço, neste trabalho será abordado o treinamento supervisionada. O treinamento supervisionado possui diversas fases são elas.

2.4.1 Inicialização da rede

Nesta etapa os valores iniciais dos neurônios e de seus pesos são atribuídos com valores aleatórios.

2.4.2 Alimentação para frente (*feed forward*)

Um conjunto dos dados de entrada, também chamado de lote, é submetido à rede e calculado suas saídas essas saídas obtidas são comparadas com as saídas desejadas, essa comparação é feita por uma função de erro.

2.4.3 Alimentação para trás (*feed backward*)

O algoritmo de otimização usa os valores da função erro e recalcula os pesos dos neurônios da última camada até a primeira camada de modo a ajustá-los. Esse processo é feito para todos os dados de treinamento e validação, após a conclusão do processo pode-se dizer que a rede foi treinada. Em resumo, no treinamento supervisionado, um conjunto de dados de entradas e saídas desejados são submetidos à rede, de forma que os erros nas saídas ajustem os pesos para que as saídas obtidas sejam próximas às saídas desejadas. O objetivo é fazer a rede aprender a mapear as entradas em saídas, sem deixar de ser capaz de generalizar, ou seja, aprender a responder corretamente a dados não usados para o treinamento. [8-9].

2.4.4 Definição dos Parâmetros e Hiper-parâmetros da rede

Para a estruturação de uma rede neural é preciso que se determine os parâmetros da rede, esses parâmetros são definidos pela própria arquitetura da rede, fazem parte dos parâmetros o número de pesos (ligações entre os neurônios) e as funções de ativação dos

neurônios. É considerado parâmetro dos modelos todas as variáveis que são ajustadas durante o treinamento.

Já os hiper-parâmetros da rede são algoritmos, funções e parâmetros globais que são executados sobre a rede como um todo. Os parâmetros globais seriam o tamanho do lote de treinamento e o número de épocas de treinamento. O algoritmo de otimização seria um hiper-parâmetro tendo como exemplos o gradiente descendente estocástico, gradiente adaptativo, propagação quadrática média. O algoritmo mais famoso de otimização é o gradiente descendente, que busca encontrar o mínimo da função erro. Assim, pode-se interpretar as redes neurais como uma função com múltiplas variáveis, onde se aplica um algoritmo de otimização para ajustar essa função ao ponto de se obter os resultados desejados [8-9].

A função de erro ou função de custo que também é um hiper-parâmetro, é uma função usada para estimar a diferença entre um valor real e um valor previsto. No caso da classificação de imagens por uma rede neural, a função erro mede o quanto errado a rede está classificando as imagens em relação a um valor definido como verdadeiro. À medida que a rede é treinada, os pesos são ajustados e o valor da função de erro é minimizado [8-9].

2.5 Transferência de aprendizado em redes neurais

Algumas estratégias são utilizadas para acelerar o processo de aprendizado de redes neurais e reduzir o conjunto de dados necessários para ajustar os pesos. Uma dessas estratégias conhecidas é a de transferência de aprendizado, que consiste em transferir o “conhecimento” de uma rede neural para uma segunda rede, sendo que ambas devem possuir a mesma arquitetura das camadas de extração de características, ou seja, mesma RNC, pois os pesos ajustados nessas camadas na primeira rede são copiados para a segunda rede neural. Depois da transferência de pesos, a rede que recebeu os pesos já ajustados pode fazer o treinamento (ajustes dos pesos) das camadas totalmente conectadas e da camada de saída a partir dos dados do novo problema [10]. Esse procedimento faz mais sentido nas redes de aprendizado profundo, pois, em geral, possuem muitos pesos, acarretando alto custo computacional e grande quantidade de dados.

Um exemplo de rede convolucional residual é a ResNet50 [11], que possui 50 camadas, tendo alcançado bons resultados para classificar imagens do banco de imagens Imagenet. Este banco de dados, considerado um *benchmark* na área, consiste em 14.197.122 de imagens anotadas de diversas naturezas, criado pelo “ImageNet Large Scale Visual Recognition Challenge (ILSVRC)” para avaliar algoritmos para detecção de objetos e classificação de imagens em grande escala. Ela é utilizada para reconhecer padrões e retirar as principais características das imagens, inspirado no córtex visual de um ser biológico.

A figura 6 ilustra a arquitetura da ResNet50.

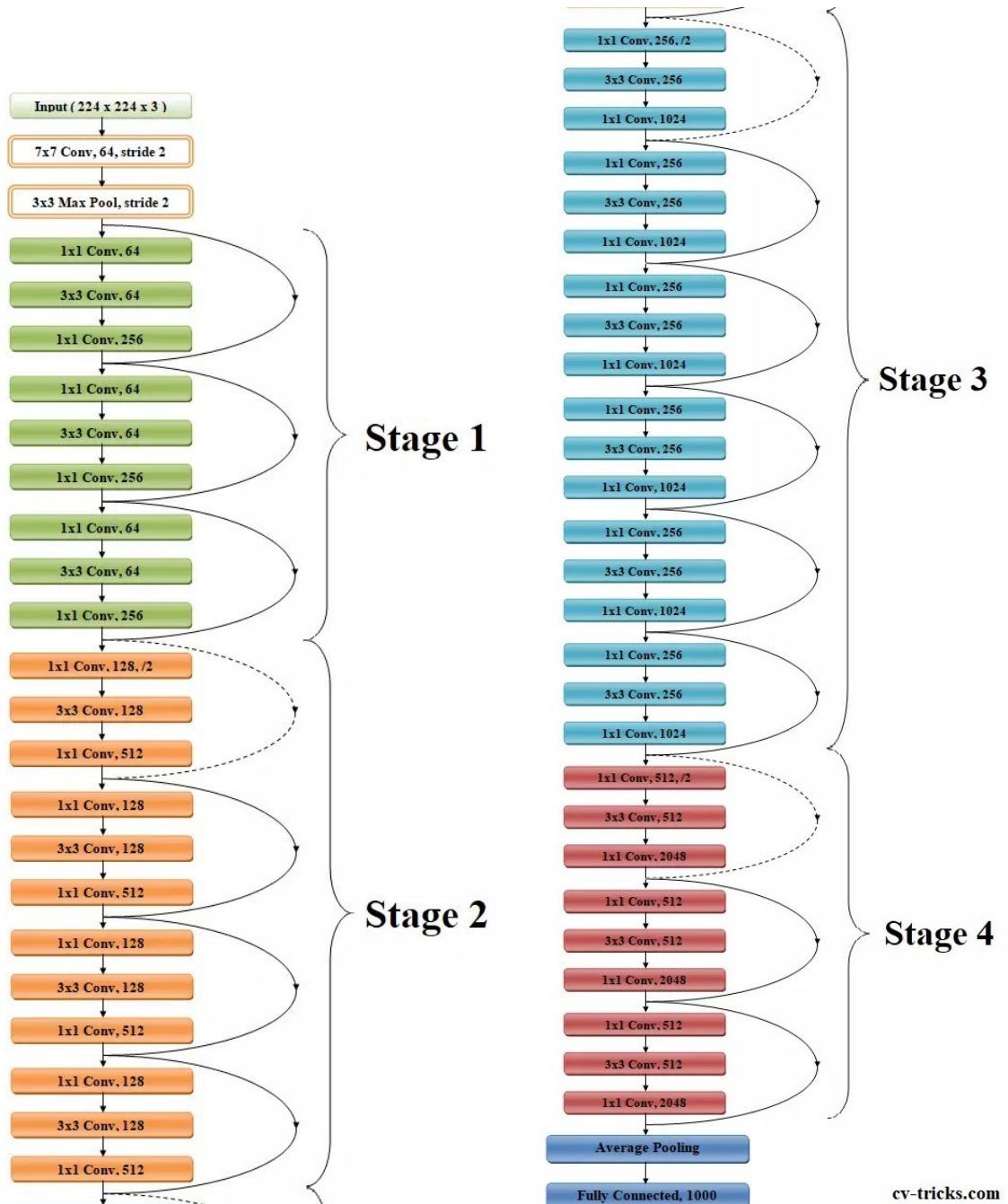


Figura 6 - Arquitetura ResNet50 [11]

3 TRABALHOS RELACIONADOS

Quando se descobriu a relevância das imagens RXT para detecção de COVID-19, vários trabalhos foram desenvolvidos [5-7] especificamente visando a detecção automatizada de COVID-19 e de outras doenças respiratórias usando essas imagens. Para realizar tal tarefa, foram propostas diferentes arquiteturas de redes neurais. Algumas dessas arquiteturas foram projetadas especialmente para detecção de padrões de COVID-19. Além disso, também foram usadas arquiteturas já conhecidas para detecção de imagens em geral, assim como diferentes tamanhos de banco de imagens e diferentes parâmetros e hiper-parâmetros de rede foram usados [5-7].

3.1 COVID-Net: rede neural convolucional

Em Wang *et al* [5], é introduzida uma nova arquitetura de rede neural profunda personalizada, adaptada para a detecção de casos COVID-19 a partir de imagens RXT, usando uma estratégia de projeto colaborativo homem-máquina. COVID-Net é a primeira arquitetura de rede neural projetada para detecção COVID-19 a apresentar um design leve de projeção-expansão-projeção-extensão (PEPX), que permite uma capacidade de representação aprimorada, enquanto reduz muito a complexidade computacional. A COVID-Net proposta foi pré-treinada com o conjunto de dados *ImageNet43* e, em seguida, treinada com o conjunto de dados COVID usando o otimizador Adam e uma política de taxa de aprendizagem, onde a taxa de aprendizagem é reduzida quando o processo de aprendizagem estagna por um período. Os seguintes hiper-parâmetros foram usados para o treinamento desse modelo: taxa de aprendizagem = 2e-4, número de épocas = 22, tamanho do lote = 64, paciência = 5. A paciência é um atributo da função Early stopping, é responsável pela parada antecipada do treinamento. Esse interrupção ocorre quando o algoritmo detecta que não houve redução do erro nos resultados da validação, após determinado número de ciclos de treinamento. Este período é definido pelo parâmetro de "paciente" da rede. O protótipo inicial da COVID-Net foi construído e avaliado usando a biblioteca de aprendizado profundo *Keras* com um *back-end TensorFlow*. A COVID-Net foi usada para detectar COVID-19, não COVID-19 e Pneumonia. Pode-se observar que COVID-Net atinge boa capacidade de diagnóstico ao atingir 93,3% de precisão de teste de validação, destacando assim a eficácia de alavancar uma estratégia de *design* colaborativo homem-máquina para criar arquiteturas de rede neural profunda altamente personalizadas de uma maneira acelerada. A figura 7 mostra a arquitetura proposta em Wang *et al* [5].

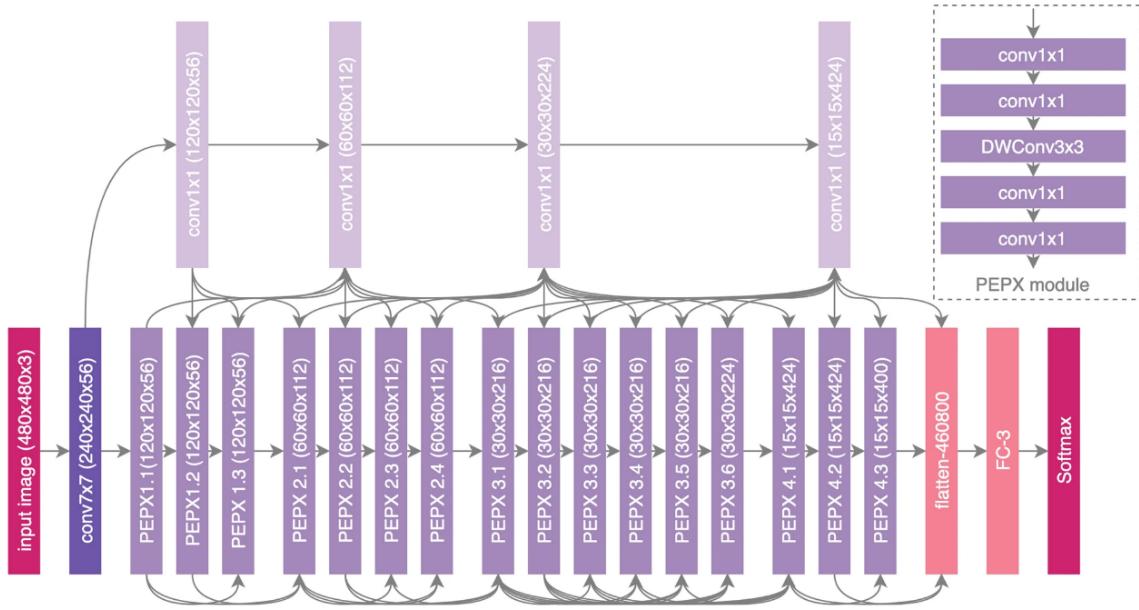


Figura 7 - Arquitetura proposta em Wang *et al* [5]

3.2 Transferência de aprendizado ResNet50 e ResNet101

Em Jain *et al* [6], é usada uma metodologia que consiste em quatro fases: (i) pré-processamento de imagem; (ii) calibragem dos dados; (iii) treinamento da rede ResNet50 de aprendizagem profunda para diferenciar pneumonia induzida por vírus, pneumonia induzida por bactérias e casos normal e (iv) treinamento da rede ResNet-101 para detectar a presença de COVID-19 em casos positivos de pneumonia induzida por vírus usando imagens de RXT. Neste estudo é proposta uma estratégia promissora de solução em dois estágios para detectar casos de COVID-19, diferenciando-a de pneumonia bacteriana, pneumonia viral e pessoas saudáveis com as imagens de RXT usadas em redes de aprendizagem residuais profundas (ResNet50 e ResNet-101). O modelo do primeiro estágio mostra um bom desempenho com uma acurácia de 93,01% na diferenciação de pneumonia induzida por vírus, pneumonia induzida por bactérias e pessoas saudáveis. As imagens de RXT de pneumonia induzida por vírus foram posteriormente analisadas, no segundo estágio, para detectar a presença de COVID-19. O modelo do segundo estágio, que detecta a presença ou ausência de COVID-19, mostra um desempenho ainda melhor, com acurácia de 97,22%. A figura 8 mostra a arquitetura proposta em Jain *et al* [6].

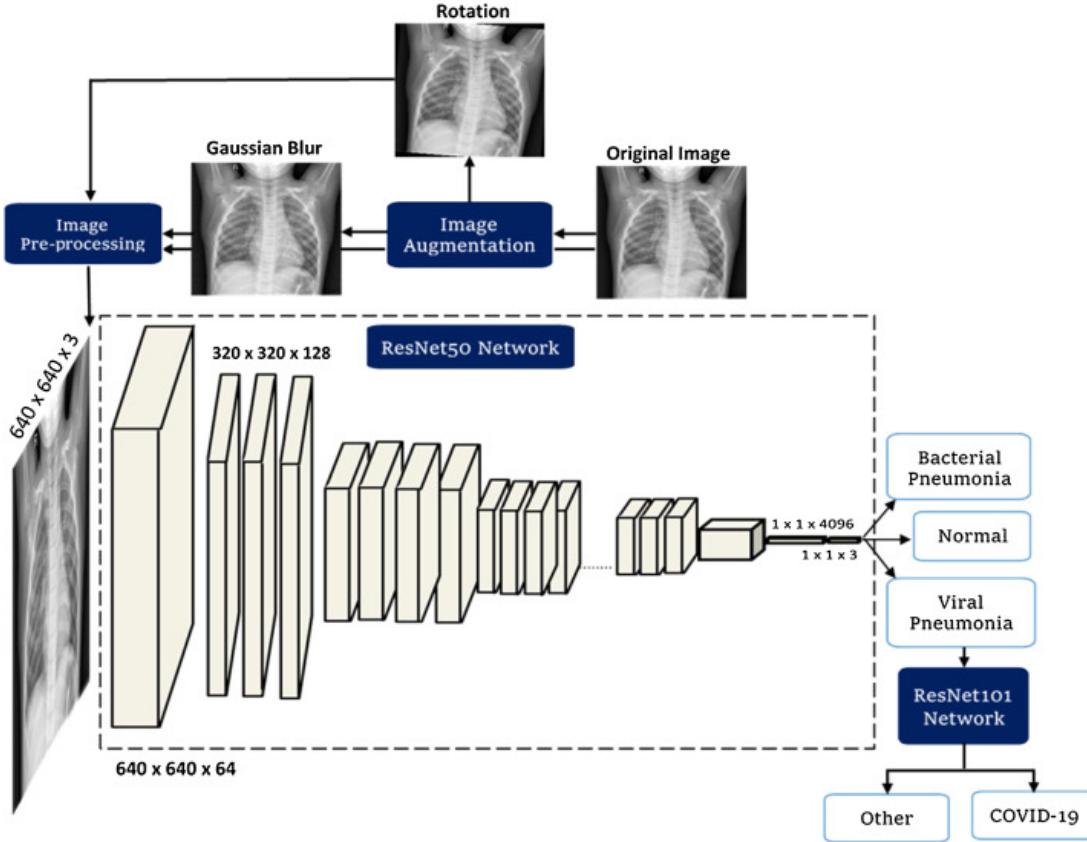


Figura 8 - Arquitetura proposta em Jain *et al* [6]

3.3 Aprendizado profundo para detecção de COVID-19

Em Haque *et al* [7], é proposta uma arquitetura personalizada com quatro camadas convolucionais: a primeira é uma camada convolucional 2D com núcleos 3×3 e função de ativação da Unidade Linear Retificada (ReLU). As próximas três camadas são a camada convolucional 2D, com a função de ativação ReLU e a camada de *pooling* máximo. A próxima etapa do modelo passa pela camada *flatten*, que em seguida alimenta a camada totalmente conectada com *dropout*. Na camada de saída a função de ativação sigmóide toma a decisão de classificação dependendo do rótulo de classificação dos neurônios. Assim, na camada de saída, uma imagem radiográfica apresentada na entrada do modelo é declarada como positiva (paciente possui COVID-19) ou normal (paciente saudável). Este modelo é denominado modelo 1. Para análise comparativa, mais dois modelos de redes neurais convolucionais também são desenvolvidos com 3 e 5 camadas de convolução em vez das 4 camadas do modelo 1. Esses modelos com 3 e 5 camadas de convolução são denominados modelo 2 e modelo 3, respectivamente. O modelo 2 tem uma camada de convolução 3×3 com ReLU com 32 canais e mais duas camadas de convolução 3×3 com ReLU e camadas de agrupamento máximo 2×2 com 64 canais cada. O modelo

3 tem uma camada de convolução 3 x 3 com ReLU e *maxpooling* 2 x 2 com 128 canais como a quinta camada. Para o conjunto de dados, a acurácia geral é de 97,5%, 93,75% e 95% para o modelo 1, modelo 2 e modelo 3, respectivamente, enquanto o modelo que utilizou *transfer learning* alcançou a acurácia de 88,5%, 78,75% e 60%, respectivamente. Isso mostra que o modelo proposto (modelo 1) tem um desempenho melhor do que os outros modelos em acurácia. A figura 9 mostra a arquitetura proposta em Haque *et al* [7].

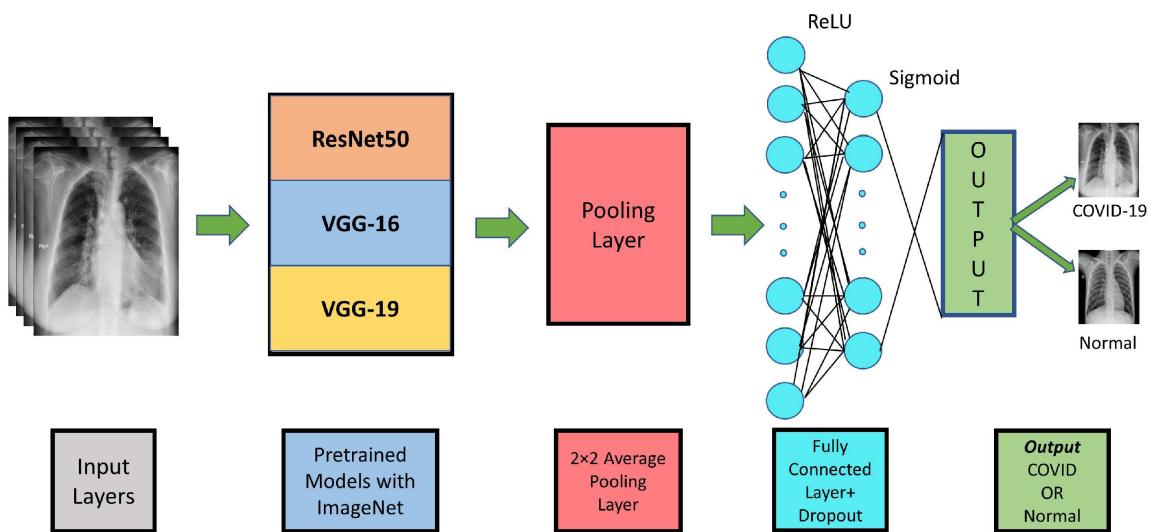


Figura 9 - Arquitetura proposta em Haque *et al* [7]

4 IMPLEMENTAÇÃO

Para iniciar o desenvolvimento de uma rede neural capaz de identificar COVID-19, primeiro é necessário estabelecer quais ferramentas serão usadas. A linguagem de programação escolhida foi *Python*, usando a biblioteca desenvolvida pela *Google* chamada *Keras*. Nesse ambiente de trabalho já estão contidas funções necessárias para acelerar o desenvolvimento de programação de redes neurais, como os vários algoritmos utilizados e as estruturas básicas necessárias para o desenvolvimento de redes neurais. Como o treinamento de redes neurais exige uma quantidade de processamento elevada e uma quantidade de armazenamento considerável, todo o código criado foi executado na plataforma do *Google Colab*. Essa plataforma disponibiliza servidores do *Google* em nuvem para elevadas quantidades de processamento. O código criado usando a biblioteca *Keras* foi otimizado para poder ser executado em GPUs na plataforma do *Google Colab*, reduzindo o tempo de processamento de toda a rede neural, principalmente na parte do treinamento.

Para realizar a implementação do trabalho foram cumpridas diversas etapas, seguindo a metodologia. São elas: coleta e pré-processamento dos dados, modelagem da arquitetura da rede neural, definição de hiper-parâmetros e parâmetros do modelo.

4.1 Coleta e pré-processamento dos dados

É preciso primeiro coletar os dados necessários para realizar o trabalho. Nesse caso os dados são imagens de RXT de pessoas infectadas com COVID-19 e imagens de RXT de pessoas não infectadas com COVID-19, de instituições de saúde que disponibilizam esses dados de forma pública. No caso desse trabalho, as imagens de treino e validação foram extraídas de um *dataset* publicados no site *Mendeley Data* [12] de forma gratuita para pesquisas acadêmicas, obtidas através de instituições de saúde. A partir deste *dataset* [12] foram extraídas 4000 imagens de RXT sem COVID-19 e 4000 imagens de RXT com COVID-19. Para as imagens de teste foi utilizado outro *dataset*, obtido da plataforma *Kaggle*¹⁴ [13]. O *dataset* de teste usado foi feito por uma equipe de pesquisadores das universidades do Qatar, de Doha e de Dhaka, Bangladesh, com seus colaboradores do Paquistão e da Malásia que, em colaboração com médicos, criaram um banco de dados de imagens de raios-X de tórax para casos de COVID-19 positivos, além de imagens de pneumonia viral, opacidade pulmonar e normal (saudável). Deste *dataset* de teste foram extraídas 3000 imagens de pacientes com COVID-19 e 3000 imagens de pacientes sem

¹⁴ Kaggle uma plataforma *open source*, onde são disponibilizados códigos e *datasets*

COVID-19 (saudável). Em seguida, foi feito um pré-processamento das informações, ou seja, foram ajustados os tamanhos das imagens para as dimensões 224x224 pixels. Depois, triplicaram-se os canais de cores repetindo as imagens, já que as imagens só possuem um canal de cor, por serem em escalas de cinza. Após os ajustes, cada imagem tornou-se um tensor de 224x224x3 pixels. Esse processo foi feito tanto na parte do treino e validação quanto no teste.

4.2 Modelagem da arquitetura da rede neural

A modelagem da arquitetura de redes neurais envolve, principalmente, a especificação do tipo camada, números e tipos de filtros a serem utilizadas. A camada de entrada da rede neural foi projetada para receber um tensor de imagens. As dimensões do tensor de imagem são 224x224x3 pixels, ou seja, as imagens são um tensor com 224 pixels de largura por 224 pixel de altura com 3 canais de cores. Para a camada de extração de características foi utilizada uma transferência de aprendizado. Assim, foi transferida parte do conhecimento de uma rede neural convolucional residual ResNet50 [11] para executar a detecção de padrões nas imagens. Logo em seguida, foi adicionada uma camada de achatamento (*flatten*) e, depois, uma rede neural totalmente conectada utilizando a função de ativação *ReLU*, com camadas de *dropout* de 50% entre as camadas totalmente conectadas para evitar o *overfitting*. No final da rede neural totalmente conectada, a camada de saída possui 2 neurônios: um para detectar COVID-19 e o outro para detectar ausência de COVID-19, ambos com a função de ativação *softmax*. A figura 10 mostra a arquitetura completa da rede implementada nesse projeto.

4.3 Definição de hiper-parâmetros e parâmetros do modelo

Essa fase da construção da solução baseada em redes neurais, em geral, envolve a busca e definição de hiper-parâmetros que resultem em maior eficácia do modelo. Para o algoritmo de otimização (hiper-parâmetro) foi utilizado o gradiente descendente estocástico adaptativo (*AdaGrad*), com uma taxa de aprendizagem de 0,001; para a função de erro (hiper-parâmetro) utilizou-se a entropia cruzada categórica (*categorical cross-entropy*); foi utilizado tamanho de lote (*batch size*) de 64 (hiper-parâmetro). No total, foram usadas 8 camadas escondidas (hiper-parâmetro) e determinou-se um número máximo de 200 épocas de treinamento (hiper-parâmetro).

O número total de parâmetros treináveis foi de 421.536.770, além de 23.587.712 parâmetros não treináveis pertencentes a ResNet50 totalizando 445.124.482 parâmetros.

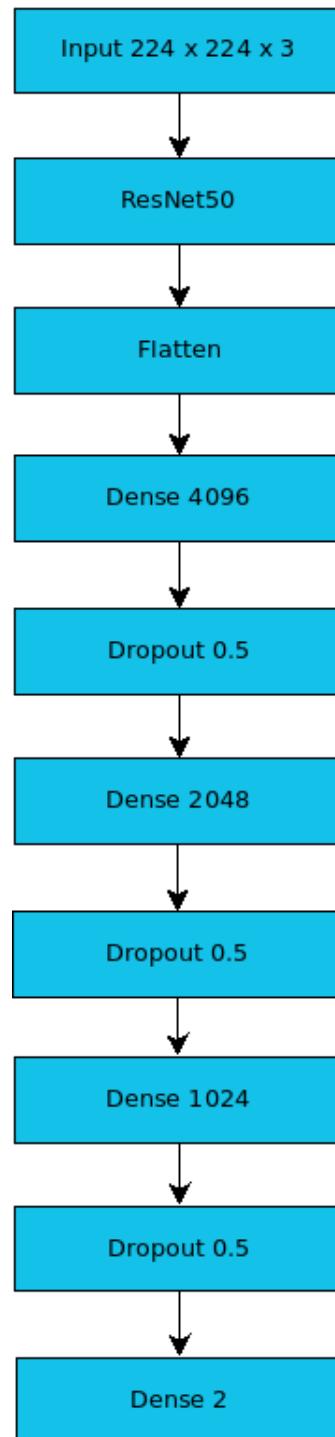


Figura 10 - Arquitetura completa da rede implementada

4.4 Treinamento da rede neural

Como foi utilizado um método de transferência de aprendizado, a única parte da rede treinada foi a parte da rede totalmente conectada. O treino foi processado em lotes de 64 imagens. O *dataset* de treino e validação usado foi dividido de maneira que a parte

do treino corresponde a 80% das imagens do *dataset* equivalente a 6400 imagens, e a de validação corresponde a 20% das imagens do *dataset* equivalente a 1600 imagens, sendo que essas divisões possuem quantidades proporcionais de imagens de RXT com COVID-19 e sem COVID-19. Foi estimado um máximo de 200 épocas de treinamento, com um critério de parada chamado de “parada antecipada” (*early stopping*), onde foi escolhido como monitor o valor da função de erro. Se, depois de 20 épocas sucessivas o valor de erro não diminuir, o treinamento é encerrado, ou seja, foi aplicada uma parada antecipada com paciência de 20 épocas, tendo como monitor o valor da função de erro.

5 ANÁLISE DE RESULTADOS

Duas métricas foram utilizadas para a avaliação do treinamento/validação: a primeira métrica usada foi a acurácia, métrica essa que define a precisão e exatidão dos dados e é obtida através da comparação entre os dados experimentais e os dados considerados fundamentalmente verdadeiros. A segunda métrica utilizada foi a perda (função erro), que mede o quanto distante ou “errado” um dado está em relação ao dado fundamentalmente verdadeiro.

Após o treinamento da rede neural ser executado utilizando as imagens do *dataset* de treino e validação, que são equivalentes a 8000 imagens, a rede obteve seu desempenho máximo após 134 épocas (usando parada antecipada). Foi obtido o seguinte gráfico da acurácia para o treino e validação mostrado na figura 11. O gráfico da perda (função erro) para o treino e validação é mostrado na figura 12:

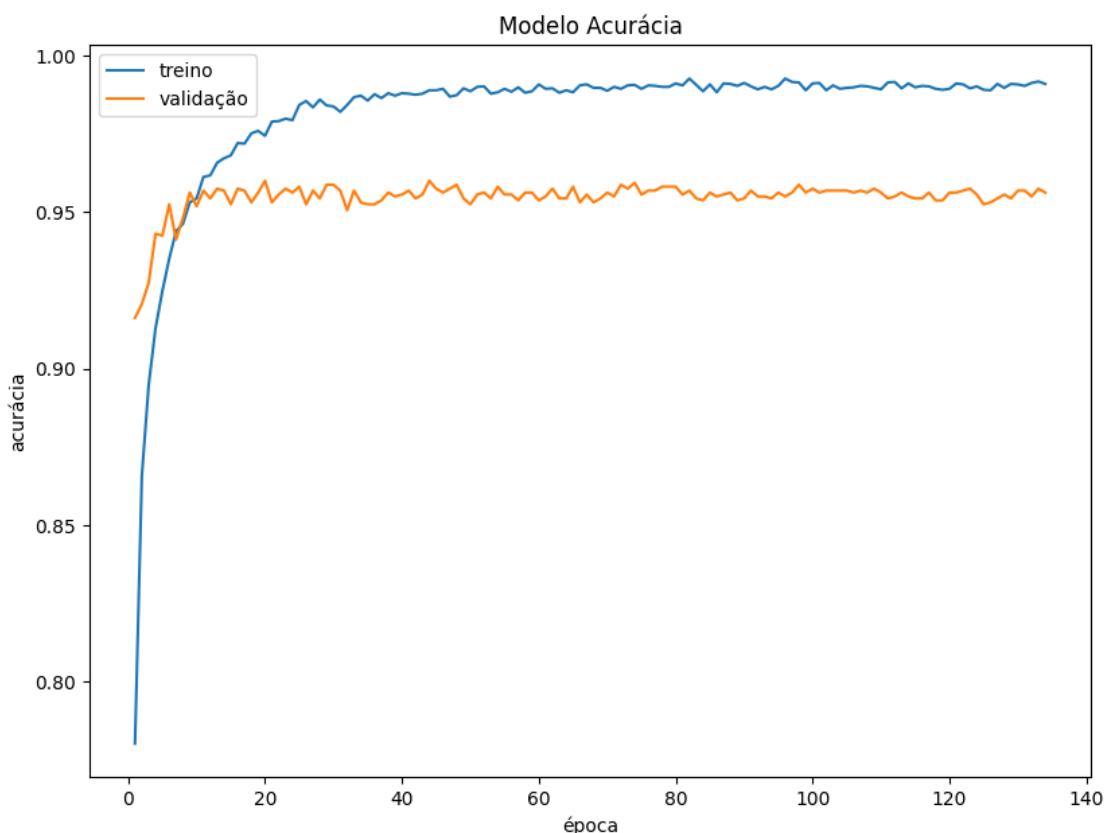


Figura 11 - Acurácia do modelo

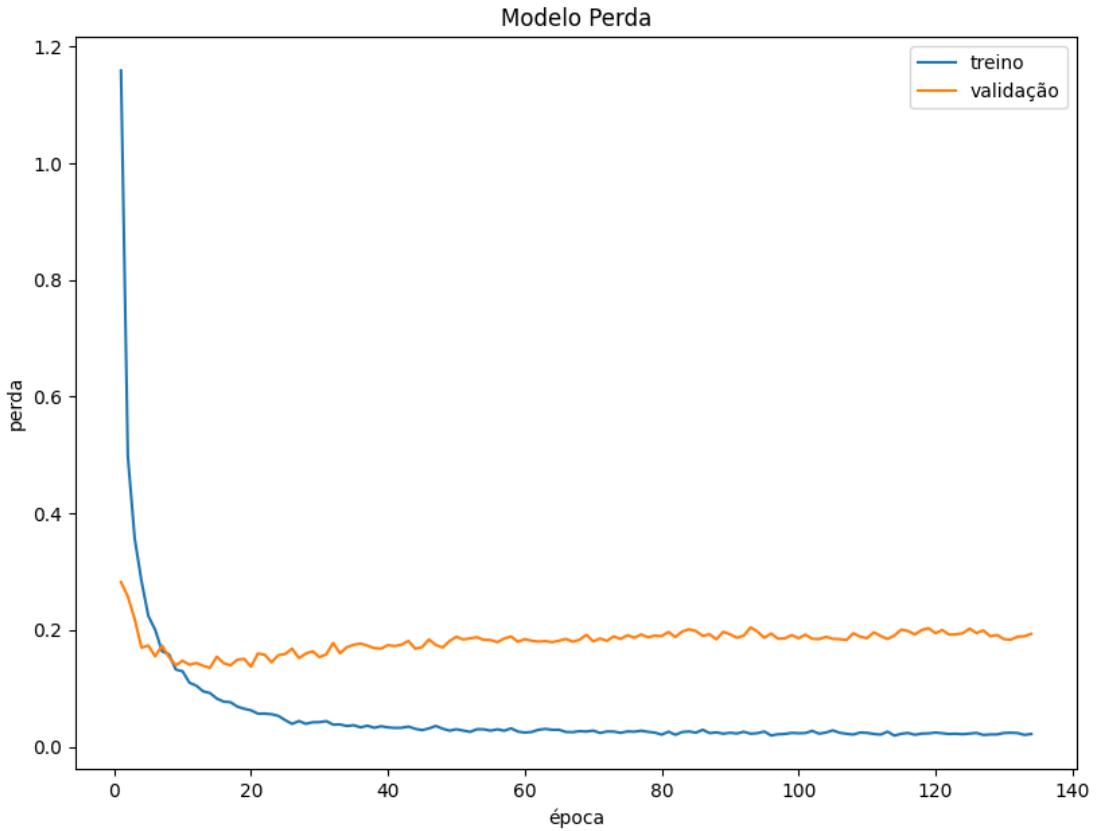


Figura 12 - Perda (função erro) do modelo

O modelo treinado foi usado para avaliar as imagens do *dataset* de teste, que equivalem a 6000 no total. Devido à natureza binária da classificação, um *dataset* balanceado foi utilizado. Sendo assim, metade das imagens do *dataset* são de COVID-19 e a outra metade das imagens do *dataset* são sem COVID-19.

Para uma análise mais detalhada dos resultados foi montada uma matriz de confusão, matriz essa que visa mostrar a quantidade de casos falsos positivos, falsos negativos, verdadeiros positivos e verdadeiros negativos. Os casos falsos positivos são aqueles em que a rede classificou como positivos (com COVID-19), mas eram negativos (sem COVID-19). Já os falsos negativos são os casos classificados como negativos (sem COVID-19), mas eram positivos (com COVID-19). Os casos verdadeiros positivos são aqueles classificados como positivos (com COVID-19) e eram positivos (com COVID-19). Por último, os casos verdadeiros negativos são aqueles classificados como negativos (sem COVID-19), e eram negativos (sem COVID-19). Assim, obteve-se a matriz de confusão apresentada na Figura 13.

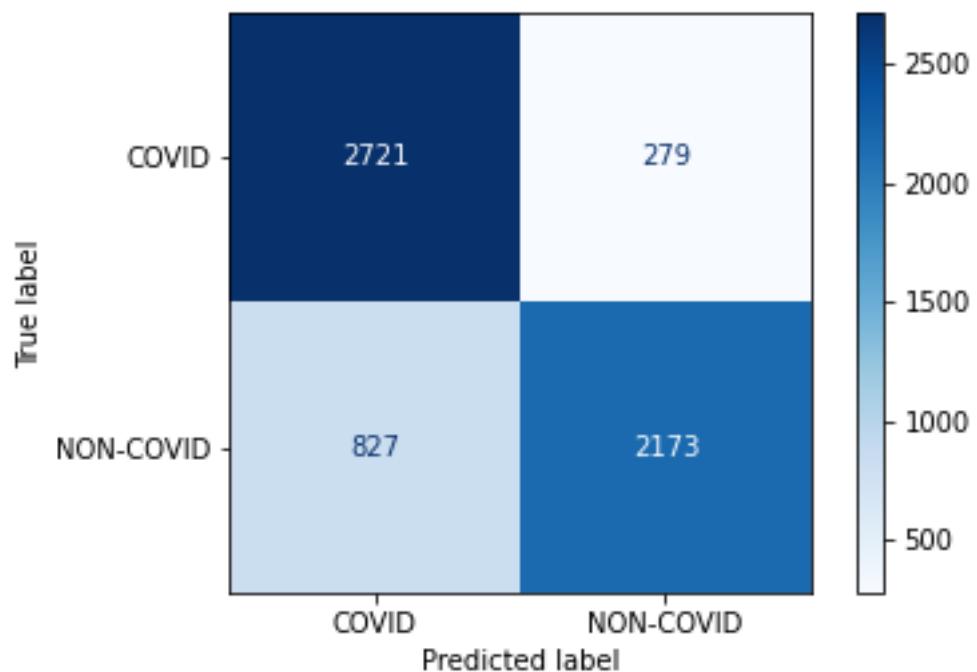


Figura 13 - Matriz de confusão

Apesar de a base de teste ter usado dados com imagens provenientes de outras unidades de saúde, obteve-se uma acurácia de 81,57%. Tais métricas apontam que o modelo apresenta mais falsos positivos para COVID-19 do que falsos negativos. Apresentar erros maiores de falsos positivos do que negativos, costuma ser melhor, pois, em geral, cruza-se esse resultado com outros exames laboratoriais e clínicos para corroborar o resultado.

6 CONCLUSÃO

A descoberta de um novo vírus capaz de infectar humanos, causador de uma nova doença que se espalhou a nível global tornando-se uma pandemia, trouxe mais desafios para toda a sociedade, mas um desafio ainda maior para a área da saúde. Para combater a disseminação de tal vírus e controlar a pandemia, diversas áreas da ciência buscaram contribuir com seus conhecimentos e com tecnologias a fim de amenizar as consequências causadas por essa pandemia. Na área da computação não foi diferente, buscaram-se métodos e ferramentas capazes de ajudar todos aqueles profissionais de saúde que estavam na linha de frente combatendo a pandemia.

O objetivo do trabalho foi implementar uma arquitetura de rede neural artificial capaz de identificar COVID-19 em Raio-X de tórax, buscando automatizar e complementar o diagnóstico da doença. Um trabalho importante, pois se bem implementado poderá ajudar os profissionais da área da saúde no combate a doença da COVID-19. O trabalho exigiu um estudo de aprendizado de máquina, redes neurais e aprendizado profundo, além da utilização da biblioteca *Keras*, um pouco de neurociência, processamento em nuvem e métodos de processamento de imagem.

Os resultados obtidos foram satisfatórios, pois a acurácia alcançada pela rede treinada foi razoavelmente boa, em torno de 81%. Isso demonstra que a rede neural aprendeu a identificar a presença de COVID-19 e a ausência de COVID-19 nas imagens de Raio-X de tórax. No entanto, para ser utilizada em hospitais o modelo precisa garantir uma acurácia maior, pois é preciso resultados mais confiáveis para auxiliar no diagnóstico. Algumas das maneiras de melhorar o desempenho da rede neural em um trabalho futuro, poderia incluir a exploração de outras formas de pré-processamento das imagens, busca de melhores valores de hiper-parâmetros para o modelo, incluindo o algoritmo de ajuste dos pesos, além de outras arquiteturas de redes neurais usadas em aprendizados profundos.

REFERÊNCIAS

- [1] SECRETARIA DE ESTADO DE SAÚDE DE MINAS GERAIS. Coronavírus Secretaria de Estado de Saúde de Minas Gerais, c2021. Perguntas e Respostas. Disponível em:<<https://coronavirus.saude.mg.gov.br/perguntaserespostas>>. Acesso em: 12 set. 2021.
- [2] MENEZES, Maria Elizabeth; LIMA, Lenilza Mattos; MAERINELLO, Flávia. Diagnóstico laboratorial do SARS-CoV-2 por transcrição reversa seguida de reação em cadeia da polimerase em tempo real (RT-PCR). RBAC, 2020. Disponível em:<<https://doi.org/10.21877/2448-3877.20200006>>.
- [3] LIMA, Claudio Márcio Amaral de Oliveira. Information about the new coronavirus disease (COVID-19). SciELO, 2020. Disponível em:<<https://doi.org/10.1590/0100-3984.2020.53.2e1>>.
- [4] PASCARELLA, Giuseppe et al. COVID-19 diagnosis and management: a comprehensive review. Journal of Internal Medicine, v. 288, n. 2, p. 192-206, 2020. Disponível em:<<https://doi.org/10.1111/joim.13091>>.
- [5] WANG, Linda; LIN, Zhong Qiu; WONG, Alexander. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. Scientific Reports, v. 10, n. 1, p. 1-12, 2020. Disponível em:<<https://doi.org/10.1038/s41598-020-76550-z>>.
- [6] JAIN, Govardhan et al. A deep learning approach to detect Covid-19 coronavirus with X-ray images. Biocybernetics and biomedical engineering, v. 40, n. 4, p. 1391-1405, 2020. Disponível em:<<https://doi.org/10.1016/j.bbe.2020.08.008>>.
- [7] HAQUE, Khandaker Foysal; ABDELGAWAD, Ahmed. A deep learning approach to detect COVID-19 patients from chest X-ray images. AI, v. 1, n. 3, p. 418-435, 2020. Disponível em:<<https://doi.org/10.3390/ai1030027>>.
- [8] DE PÁDUA BRAGA, Antônio; DE LEON FERREIRA, André Carlos Ponce; LUDERMIR, Teresa Bernarda. Redes neurais artificiais: teoria e aplicações. Rio de Janeiro, Brazil:: LTC Editora, 2007.
- [9] SEWAK, Mohit; KARIM, Md Rezaul; PUJARI, Pradeep. Practical convolutional neural networks: implement advanced deep learning models using Python. Packt Publishing Ltd, 2018.

- [10] TORREY, Lisa; SHAVLIK, Jude. Transfer learning. In: Handbook of research on machine learning applications and trends: algorithms, methods, and techniques. IGI global, 2010. p. 242-264.
- [11] HE, K., ZHANG, X., REN, S., SUN, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, p. 770-778, 2016. Disponível em:<<https://arxiv.org/abs/1512.03385>>. Acesso em: 11 set. 2021.
- [12] EL-SHAFAI, Walid; ABD EL-SAMIE, Fathi. Extensive COVID-19 X-Ray and CT Chest Images Dataset. Mendeley Data, V3, 2020. Disponível em:<<https://doi.org/10.17632/8h65ywd2jr.3>>.
- [13] RAHMAN, Tawsifur; CHOWDHURY, Muhammad; KHANDAKAR, Amith. COVID-19 Radiography Database. Kaggle, 2021. Disponível em:<<https://www.kaggle.com/tawsifurrahman/covid19-radiography-database>>. Acesso em: 14 set. 2021.
- [14] TCH, ANDREW. The mostly complete chart of Neural Networks. Towards Data Science, 2021. Disponível em:<<https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464>>. Acesso em: 15 set. 2021.

GLOSSÁRIO

<i>Dataset</i>	Conjunto de dados
<i>Flatten</i>	Camada de achatamento
<i>Relu</i>	Função retificadora em unidade linear
<i>Softmax</i>	Função de normalização exponencial
<i>Batch</i>	Lote, quantidade usada para dividir dados
<i>Dropout</i>	Camada para diluir os pesos treinados

ANEXO A – Códigos em Python do projeto e repositório GitHub

Os datasets utilizados estão referenciados no TCC, os códigos dos trabalhos foram copilados e processados na plataforma Google Colab Pro.

A.1 Código em Python do projeto

A.1.1 Código pré-processamento dataset

```
"""Carregar o drive"""
from google.colab import drive
drive.mount('/content/drive')

"""Imports"""
import numpy as np
import os

"""Caminhos para os datasets"""
path_covid_treino = '/content/drive/MyDrive/TCC/COVID-19-Dataset/X-ray/class_COVID'
path_non_covid_treino = '/content/drive/MyDrive/TCC/COVID-19-Dataset/X-ray/class_Non-COVID'
path_covid_teste = '/content/drive/MyDrive/TCC/COVID-19-Dataset/teste-X-ray/class_COVID'
path_non_covid_teste = '/content/drive/MyDrive/TCC/COVID-19-Dataset/teste-X-ray/class_Non-COVID'
count = 1

"""loop para renomear as imagens do treino COVID"""
for file in os.listdir(path_covid_treino):
    nn = "COVID_image_" + str(count) + ".jpg"
    os.rename(path+'/'+file, path+'/'+nn)
    count += 1
count = 1
```

```

"""loop para renomear as imagens do treino Non-COVID"""
for file in os.listdir(path_non_covid_treino):
    nn = "Non-COVID_image_" + str(count) + ".jpg"
    os.rename(path+'/' +file , path+'/' +nn)
    count += 1
count = 1

"""loop para renomear as imagens de teste COVID"""
for file in os.listdir(path_covid_teste):
    nn = "COVID_image_" + str(count) + ".png"
    os.rename(path+'/' +file , path+'/' +nn)
    count += 1
count = 1

"""loop para renomear as imagens de teste Non-COVID"""
for file in os.listdir(path_non_covid_teste):
    nn = "Non-COVID_image_" + str(count) + ".png"
    os.rename(path+'/' +file , path+'/' +nn)
    count += 1

```

A.1.2 Código treino da rede

```

"""Carregar o drive"""
from google.colab import drive
drive.mount('/content/drive')

"""Imports"""
import numpy as np
import tensorflow as tf
from matplotlib import pyplot as plt
from matplotlib.pyplot import figure
from tensorflow import keras
from tensorflow.keras.preprocessing import image_dataset_from_directory
from tensorflow.keras.applications.resnet50 import ResNet50
from keras.models import Sequential
from tensorflow.keras import layers

```

```
"""Carregando o caminho para as imagens"""
path = '/content/drive/MyDrive/TCC/COVID-19-Dataset/X-ray'

"""Carregando as imagens de treino para o dataset de treino"""
ds_training = image_dataset_from_directory(
    path,
    labels="inferred",
    label_mode="categorical",
    class_names=['class_Non-COVID', 'class_COVID'],
    color_mode="rgb",
    batch_size = 64,
    image_size = (224, 224),
    shuffle = True,
    seed = 123,
    validation_split = 0.2,
    subset = "training",
    crop_to_aspect_ratio=True,
)

"""Carregando as imagens de validacao para o dataset de validacao"""
ds_validation = image_dataset_from_directory(
    path,
    labels="inferred",
    label_mode="categorical",
    class_names=['class_Non-COVID', 'class_COVID'],
    color_mode="rgb",
    batch_size = 64,
    image_size = (224, 224),
    shuffle = True,
    seed = 123,
    validation_split = 0.2,
    subset = "validation",
    crop_to_aspect_ratio=True,
)
```

```

"""Montando o modelo"""
base_model = ResNet50(include_top=False,
weights="imagenet", input_shape=(224, 224, 3))
base_model.trainable = False
inputs = keras.Input(shape=(224, 224, 3))
x = base_model(inputs, training=False)
x = layers.Flatten()(x)
x = layers.Dense(4096, activation="relu")(x)
x = layers.Dropout(0.5)(x)
x = layers.Dense(2048, activation="relu")(x)
x = layers.Dropout(0.5)(x)
x = layers.Dense(1024, activation="relu")(x)
x = layers.Dropout(0.5)(x)
outputs = layers.Dense(2, activation="softmax")(x)
model = keras.Model(inputs, outputs)
model.summary()

"""Compilando o modelo"""
model.compile(optimizer="adagrad", loss="categorical_crossentropy",
metrics=['accuracy'])

"""Treinando o modelo"""
callback = tf.keras.callbacks.EarlyStopping(monitor='loss',
patience=20)
history = model.fit(ds_training, validation_data=ds_validation,
epochs=200, callbacks=[callback])
model.save("/content/drive/MyDrive/models/covid19_model")

"""Plot da acuracia do treino e da validacao"""
figure(figsize=(9.6, 7.2), dpi=100)
train_acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
epo = range(1, len(train_acc)+1)
plt.plot(epo, train_acc, label='treino')
plt.plot(epo, val_acc, label='validacao')
plt.title('Modelo_Acuracia')
plt.xlabel('epoca')
plt.ylabel('acuracia')
plt.legend()

```

```

plt.savefig( '/content/drive/MyDrive/accuracy.png' )
plt.show()

"""Plot da loss treino e da validacao"""
figure(figsize=(9.6, 7.2), dpi=100)
train_loss = history.history[ 'loss' ]
val_loss = history.history[ 'val_loss' ]
epo = range(1, len(train_loss)+1)
plt.plot(epo, train_loss, label='treino')
plt.plot(epo, val_loss, label='validacao')
plt.title('Modelo_Perda')
plt.xlabel('epoca')
plt.ylabel('perda')
plt.legend()
plt.savefig( '/content/drive/MyDrive/loss.png' )
plt.show()

```

A.1.3 Código teste da rede

```

"""Carregar o drive"""
from google.colab import drive
drive.mount('/content/drive')

"""Imports"""
import numpy as np
import tensorflow as tf
from matplotlib.pyplot import figure
from matplotlib import pyplot as plt
from tensorflow import keras
from tensorflow.keras.preprocessing import image_dataset_from_directory
from sklearn.metrics import ConfusionMatrixDisplay
from sklearn.metrics import confusion_matrix

"""Carregando o caminho para as imagens de validacao e de teste"""
path teste = '/content/drive/MyDrive/TCC/COVID-19-Dataset/teste-X-ray'

```

```

"""Carregando o dataset das imagens de teste"""
ds_teste = image_dataset_from_directory(
    path_teste ,
    labels="inferred" ,
    label_mode="categorical" ,
    class_names=[ 'class_Non-COVID' , 'class_COVID' ] ,
    color_mode="rgb" ,
    batch_size = 64,
    image_size = (224, 224),
    shuffle = True,
    crop_to_aspect_ratio=True ,
)

"""Carregando o modelo treinado"""
model_path = '/content/drive/MyDrive/models/covid19_model'
model = keras.models.load_model(model_path)

"""Obtendo matriz de confusao das imagens do teste"""
lab = []
pred = []
for images, labels in ds_teste:
    lab.append(tf.argmax(labels, axis=1))
    pred.append(tf.argmax(model.predict(images), axis=1))
true_categories = tf.concat(lab, axis=0)
predicted_categories = tf.concat(pred, axis=0)

"""Montando a matriz de confusao"""
lbl = ["COVID", "NON-COVID"]
cm = confusion_matrix(true_categories, predicted_categories)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=lbl)
disp.plot(cmap=plt.cm.Blues)
plt.savefig('/content/drive/MyDrive/mc.png')
plt.show()

```

A.2 Repositório GitHub

[⟨https://github.com/IanGermano/TCC⟩](https://github.com/IanGermano/TCC)