Estudo comparativo entre modelos de Deep Learning para detecção da Covid-19 utilizando imagens de raios-x do tórax

Vitor Vaz Andrade, Matheus Giovanni Pires
Universidade Estadual de Feira de Santana - UEFS
Engenharia de Computação
vitorandrade@ecomp.uefs.br, mgpires@ecomp.uefs.br

***Resumo*—Com a disseminação da Covid-19 pelo mundo, aliado à sua complexidade de diagnóstico, devido à semelhança com outras doenças virais, como por exemplo a penumonia, tornou-se necessário a utilização de métodos práticos e eficazes para sua detecção. Imagens da radiografia do tórax é um recurso que vem sendo amplamente utilizado para monitorar a doença. Com isso, o fato é que muitos estudos têm focado em utilizar Deep Learning para processar imagens da radiografia e realizar um diagnóstico automático da Covid-19. Dessa forma, a partir da pesquisa em trabalhos nessa área, foi detectado que muitos estudos foram feitos com repositórios desbalanceados e com pouquíssimos casos positivos de Covid-19, pois foram realizados logo após o início da pandemia. Diante deste contexto, o objetivo desse trabalho é realizar um estudo comparativo de diferentes modelos de Redes Neurais Convolucionais para a detecção de Covid-19, a partir de imagens radiográficas do tórax. Mais precisamente, avaliamos as redes VGG-19, Densenet-121 e Xception, além de testar a segmentação da região pulmonar a partir das redes UNet e ResUnet, com o objetivo de aumentar o desempenho das redes. Nos testes sem a utilização da segmentação, a rede VGG-19 foi a que apresentou melhor desempenho no teste, com acurácia de 84%, precisão de 84,89% e *recall* de 82.44%. Por outro lado, quando a segmentação foi utilizada com a rede VGG-19, ocorreu um decréscimo de aproximadamente 33% na acurácia.**

***Keywords*-Redes Neurais Convolucionais, radiografia do tórax, diagnóstico da Covid-19.**

I. Introdução

A Covid-19 se espalhou de forma devastadora pelo mundo nos últimos anos, causando danos inestimáveis à população mundial, já tendo causado mais de 6,9 milhões de mortes e cerca de 760 milhões de casos confirmados [1]. Com isso, várias medidas foram tomadas desde a identificação da pandemia, sendo elas: isolamento social, intensificação da higiene, vacinação, entre outras. No entanto, essas são medidas para apenas evitar a contaminação. A melhor forma de diminuir a taxa de mortes pela doença é detectá-la o mais breve possível e encaminhar os pacientes para o devido tratamento.

Porém, o diagnóstico da doença não é tão simples, afinal, os sintomas são parecidos com os de outras infecções virais [2], [3]. Alguns métodos têm sido utilizados para auxiliar no diagnóstico da doença, como a RT-PCR (*reverse transcriptase-polymerase chain reaction*), tomografia computadorizada e radiografia do tórax. Mas o fato é que estudos mostram que o RT-PCR, além de ter um custo de tempo alto, apresentou sensibilidade baixa, se comparada às outras duas opções [4]. A segunda opção é a tomografia computadorizada, que de fato têm mostrado uma sensibilidade superior ao RT-PCR [5], mas também apresentou certas limitações. Por exemplo, o processo de aquisição da imagem é lento, além de ser um equipamento de alto custo. O que nos leva à terceira opção, a imagem via raios-x, que é uma opção mais barata, rápida e acessível, além de expor o corpo do paciente à radiação por menos tempo [6].

O problema é que mesmo com o auxílio das imagens radiográficas, fazer a distinção entre Covid-19 e outras infecções pulmonares não é uma tarefa fácil, mesmo para um radiologista, e isso é devido a similaridade da radiografia de pacientes com Covid-19 com radiografias de pessoas com outras doenças inflamatórias pulmonares [7]. Essa complexidade para detectar a doença a partir da radiografia, pode levar a diagnósticos errôneos, e consequentemente, a sérias consequências aos pacientes. Assim, os sistemas computacionais podem auxiliar os radiologistas a realizarem um diagnóstico mais rápido e preciso a partir das imagens.

O enorme avanço do Aprendizado Profundo (do inglês, *Deep Learning*) nos últimos anos, tem gerado bons resultados em tarefas de visão computacional, como por exemplo, classificação e segmentação de imagens. Esse avanço levou a um aumento na utilização de soluções baseadas em Inteligência Artificial em diversos campos da ciência, incluindo o domínio dos problemas biomédicos. E quando trata-se da área médica, há um destaque para as Redes Neurais Profundas, em especial as Redes Neurais Convolucionais (RNC), que vêm se provando ser extremamente benéficas para a área médica, com diversas aplicações, tais como, detecção de tumor cerebral, detecção de câncer de mama, classificação de lesão de pele, etc.

Atualmente, existem trabalhos relevantes utilizando RNC para a detecção de Covid-19 a partir da radiografia do tórax, que apresentaram resultados muito promissores. O estudo em [8] utilizou um algoritmo de síntese generativa como estratégia de exploração de *design* para criar uma arquitetura de rede neural profunda, chamada COVID-Net. O processo de síntese generativa é guiado por requisitos de *design*, como sensibilidade para COVID-19 e valor preditivo positivo (VPP) para COVID-19. Essa abordagem permite que a COVID-Net consiga altos índices de detecção de casos

positivos enquanto minimiza o número de falsos positivos, evitando sobrecarregar os locais clínicos com um volume desnecessário de testes.

O trabalho em [9] propôs um modelo contendo duas RNC, uma para segmentação do pulmão e outra para segmentação da região infectada, possibilitando não apenas detectar a Covid, mas também o seu grau de infecção. Já o projeto em [10] propôs um modelo contendo uma RNC para segmentação, que alimentava três classificadores que funcionam em paralelo, onde a saída de cada um dos três tem um peso, e a classificação final é gerada pelo somatório destes pesos.

Há outros trabalhos interessantes como em [11], que propôs um modelo que utiliza uma RNC apenas para extração de características, e em seguida as características alimentam um algoritmo evolutivo de classificação. O trabalho em [12] propôs um modelo de Rede Neural Convolucional para detecção de Covid-19, o qual tem os hiperparâmetros definidos a partir de um algoritmo evolutivo chamado *Boosted Salp Swarm Algorithm*. A classificação das imagens é realizada pelo algoritmo *Suport Vector Machine*. O trabalho [13] propôs um modelo que remove a camada de classificação da ResNet-101 e a substitui por uma rede neural de grafo, aliada ao algoritmo dos K-Vizinhos mais próximos. Por fim, o [14] propôs um modelo contendo uma RNC de segmentação chamada U-Net, cuja saída alimenta a RNC VGG-19, apresentando uma acurácia de aproximadamente 97%.

Diante do contexto apresentado, o objetivo desse trabalho é avaliar a aplicabilidade de RNC na detecção automática da Covid-19. Mais precisamente, três arquiteturas de RNC no estado da arte, especializadas em classificação de imagens, foram experimentadas e comparadas entre si, sendo elas a VGG-19, DenseNet-121 e Xception. Além disso, com o intuito de potencializar o desempenho das classificações realizadas pelas redes, avaliamos o uso de três RNC especializadas em segmentação de imagens, que foram a U-Net, ResUnet e U-Net com codificador pré-treinado.

II. Fundamentação Teórica

A. Arquitetura feedforward com múltiplas camadas

Essa rede se caracteriza por possuir uma ou mais camadas ocultas, cujos nós computacionais são chamados de neurônios ocultos. A função desses neurônios ocultos é intervir entre a entrada externa e a saída da rede, de forma a extrair estatísticas de ordem elevada. Numa rede desse tipo, a ideia é que a saída gerada a partir de uma primeira camada, seja a entrada de uma segunda camada, e assim por diante. Dessa forma, os neurônios de uma camada têm como entrada, unicamente a saída dos neurônios da camada anterior. As saídas da última camada representam a resposta global da rede [15].

Também vale ressaltar que uma rede é dita como totalmente conectada, quando cada um dos nós de uma camada da rede está conectado a todos os nós da camada adjacente seguinte [16]. Entretanto, se alguns dos elos de comunicação estiverem faltando, diz-se que a rede é parcialmente conectada. Um exemplo desse tipo de rede é ilustrado na Figura 1:

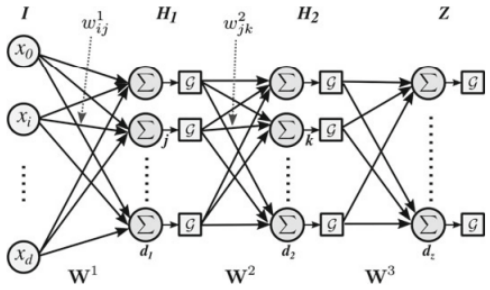

Figura 1. Rede feedforward completamente conectada [16].

B. Redes Neurais Convolucionais

Redes com arquitetura *feedforward propagation* com camadas completamente conectadas (do inglês, *Fully Connected Layer*) são muito utilizadas para extrair características e realizar classificações. Porém, há um grande problema na utilização de redes como o Perceptron Multicamadas no processamento de imagens, que é a necessidade de uma quantidade alta de neurônios, mesmo para imagens simples, tornando sua aplicação quase que impraticável. Com isso, a ideia básica por trás das RNC é conceber uma solução para reduzir o número de parâmetros, permitindo que uma rede seja mais profunda e ao mesmo tempo, tenha menos parâmetros [16].

C. Derivando uma camada convolucional a partir de uma completamente conectada

Suponhamos uma imagem em escala de cinza de dimensões 32x32 pixels, que está conectada a uma camada oculta contendo 7200 neurônios. A imagem pode ser enxergada como um vetor de 1024 elementos. Todos os elementos da camada estão conectados aos 1024 elementos da entrada. Consequentemente. essa camada completamente conectada possui $1024 \cdot 7200 = 7.232.800$ parâmetros distintos. Uma forma de se pensar na solução, seria tentar reduzir a quantidade de neurônios da rede, mas isso afetaria o seu desempenho. Objetivando diminuir a quantidade de parâmetros sem alterar a quantidade de neurônios, rearranjaremos os 7200 neurônios em 50 blocos de $12 \cdot 12$ neurônios [16]. Isso é ilustrado na Figura 2.

Mas a quantidade de parâmetros continua a mesma, então olhemos para a geometria de pixels de uma imagem. O pixel (m, n) de uma imagem é mais relacionado com seus vizinhos do que com pixels distantes. Assumindo que o neurônio (0, 0) de um bloco de neurônios, precisa apenas extrair informações do pixel (2, 2) e de seus vizinhos próximos, objetivando extrair informações apenas dessa região. Considerando que cada neurônio cubra uma região de 5x5 pixels, os neurônios

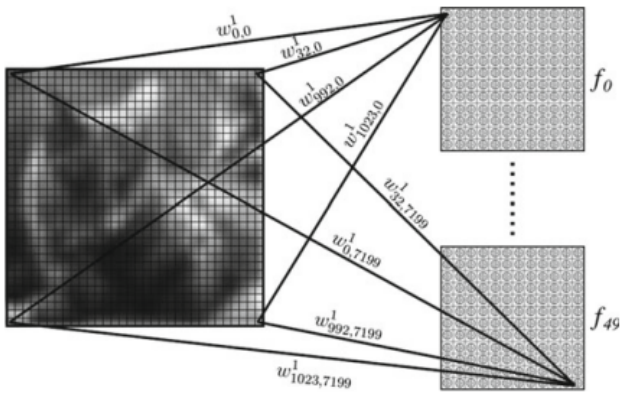

Figura 2. Neurônios rearranjados em 50 blocos, com cada bloco tendo dimensão 12x12 neurônios [16].

do bloco conseguem cobrir toda a imagem de entrada [16]. Isso é ilustrado na figura 3.

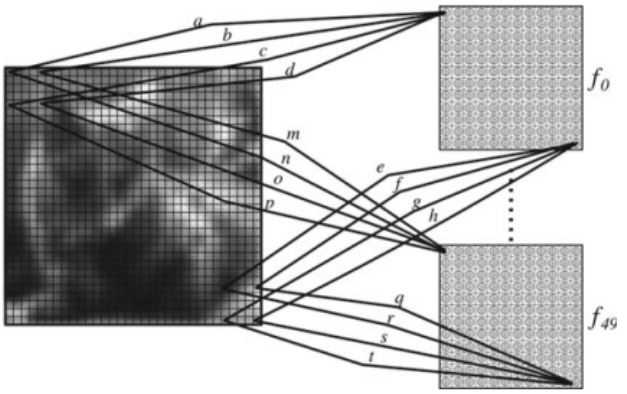

Figura 3. Cada neurônio de um bloco cobre apenas uma área de 5x5 pixels [16].

Realizando essa operação, a quantidade de parâmetros diminui para $(5 \cdot 5) \cdot 50 \cdot 12 \cdot 12 = 180.000$, o que representa uma redução de parâmtros de 97.5%. Mas esse número pode ser reduzido ainda mais. Cada neurônio do mesmo bloco possui um peso diferente um do outro. Para reduzir os parâmetros, podemos assumir que todos os neurônios de um mesmo bloco possuem o mesmo peso, com isso, cada bloco é formado por 25 pesos. Consequentemente, há $5 \cdot 5 \cdot 50 = 1250$ pesos entre a entrada e a camada escondida, ocasionando uma redução de 99.88% no número de parâmetros, se comparado a uma camada completamente conectada [16].

Designando um neurônio (p, q) em um bloco l, a saída desse neurônio é representada por:

$$f^l_{p,q} = (\gamma)(\sum_{i=0}^{4}\sum_{j=0}^{4} im(p+i, q+j)W^l_{i,j}) \quad (1)$$

Onde $W^l_{i,j}$ são os pesos do bloco l e $p, q = 0, ...11$. No exemplo que foi citado acima, o p e q variam de 0 a 4, pois cada neurônio é conectado a uma área de 5x5 pixels. A saída de cada bloco terá a mesma dimensão do bloco, logo, nesse exemplo a saída é uma matriz 12x12. Essa matriz pode ser obtida com a Equação 2 [16]:

$$f(_{p,q})^l = (\gamma)(\sum_{i=0}^{4}\sum_{j=0}^{4} im(p+i, q+j)W^l_{i,j})\forall p, q \in 0, 11 \quad (2)$$

A Equação 2, mencionada anteriormente, se mostra exatamente análoga ao processo de convolução do filtro W 5x5 com a imagem de entrada. Ou seja, o resultado desse processo consiste nas imagens geradas a partir desta convolução, e o volume de imagens produzidas equivale à quantidade de filtros da camada em questão. Em seguida, a função de ativação é aplicada em cada imagem separadamente. Em geral, se o tamanho da imagem for W x H e a camada de convolução for composta por L filtros de tamanho M x N, a saída da camada de convolução será L imagens de tamanho W - M + 1 x H - N + 1 onde cada imagem é obtida pela convolução do filtro correspondente com a imagem de entrada [16].

D. Camada de Pooling

O principal objetivo de uma camada de *pooling* é realizar uma diminuição na dimensão da saída da camada convolucional, por essa razão é chamada de *downsampling* ou subamostragem [16].

O fator pelo qual será realizado a subamostragem da imagem é chamado *stride*. Como exemplo, tendo um vetor $x = [1, 10, 8, 2, 3, 6, 7, 0]$. Ao aplicar uma subamostragem com um *stride* $s = 2$, a ideia seria ignorar 1 elemento a cada 2, resultando no novo vetor $x = [1, 8, 3, 7]$. Observe que a dimensionadade foi dividida por 2, indo de 8 para 4. No entanto, nesse processo são descartados alguns dados sem nenhum critério, o que pode gerar a perda de informações relevantes na imagem [16]. E é por isso que a abordagem de subamostragem do Pooling leva em consideração todos os pixels para descartar os prováveis menos relevantes. A proposta é fazer um max pooling, onde além do parâmetro *stride*, há também o tamanho d da região a aplicar o pooling. Observe o cálculo de subamostragem do mesmo vetor x, utiliando o maxpooling na Equação 3:

$$x = [max(1, 10), max(8, 2), max(3, 6), max(7, 0)] \quad (3)$$

Dessa forma, ao contrário da subamostragem comum, o obhetivo é reduzir o vetor de forma inteligente, considerando os vizinhos mais próximos do elemento atual, onde o tamanho da vizinhança é determinado por d. Da mesma forma, é computado o *pooling* nas imagens de saída da camada convolucional, onde a imagem é completamente percorrida a cada dxd, sendo calculado o máximo valor a cada passo [16].

E. VGG-19

As arquiteturas VGG são um grupo de RNC que se diferenciam pelo número de camadas da rede, tendo como principal característica a profundidade. Sua principal contribuição na época foi substituir os filtros convolucionais de alta dimensão, por vários pequenos filtros de dimensão 3x3, o que possibilitou o aumentado da profundidade da rede. Afinal, um empilhamento múltiplo de *kernels* de tamanho menor, é melhor do que aquele com um *kernel* de tamanho maior, porque múltiplas camadas não-lineares aumentam a profundidade da rede, que permite com que ela aprenda características mais complexas, com um custo menor [17].

A arquitetura VGG-19 possui 16 camadas convolucionais e 3 camadas completamente conectadas [17]. A Tabela I exibe a quantidade de camadas e profundidade (número de filtros da camada convolutiva) da rede:

Tabela I
ARQUITETURA VGG-19.

Camada	Profundidade
b1_conv1	64
b1_conv2	64
max_pool	
b2_conv1	128
b2_conv2	128
max_pool	
b3_conv1	256
b3_conv2	256
b3_conv3	256
b3_conv4	256
max_pool	
b3_conv4	512
b3_conv4	512
b3_conv4	512
b3_conv4	512
max_pool	
b3_conv4	512
b3_conv4	512
b3_conv4	512
b3_conv4	512
max_pool	
FC-4096	
FC-4096	
FC-1000	
Softmax	

F. Densenet-121

Com o passar do tempo, as RNC foram se tornando cada vez mais profundas, com isso, surge um problema muito popular entre essas redes: como a informação oriunda da entrada da rede ou o gradiente passa por inúmeras camadas, pode desaparecer antes de atingirem o final da rede, esse problema é conhecido como *vanishing gradient* [18].

Assim, a Densenet busca permitir que o máximo de informações possível flua entre as camadas, interconectando-as. Cada camada obtém informação das camadas anteriores e passa para as subsequentes, isso é feito através de operações de concatenação [18], ou seja, os mapas de característica obtidos nas camadas convolutivas são concatenados como mostrado na Figura 4.

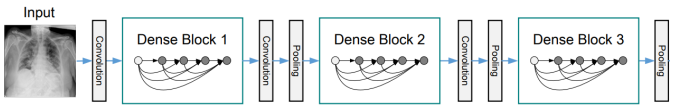

Figura 4. Exemplo de uma Densenet com 3 dense blocks [18].

Observe que a rede é dividida em blocos, essa divisão é necessária pois as concatenações só podem ocorrer dentro de cada bloco, visto que toda RNC precisa realizar operações de *pooling*, mas como o *pooling* reduz a dimensão dos dados pela metade, seria impossível realizar a concatenação entre matrizes com largura e altura distintos [18].

A Figura 5 detalha a arquitetura dos principais modelos Densenet:

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	[1 × 1 conv; 3 × 3 conv] × 24	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 48	[1 × 1 conv; 3 × 3 conv] × 64
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	[1 × 1 conv; 3 × 3 conv] × 16	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 48
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

Figura 5. Arquitetura das redes Densenet [18].

G. Xception

Para falar sobre Xception é necessário falar sobre um conceito chamado *módulo inception*, que foi inicialmente criado com a rede profunda Inception. A ideia desse conceito é realizar operações convolutivas em paralelo, onde cada filtro teria dimenões diferentes, objetivando fazer o mapeamento das características da entrada de forma independente, conforme a Figura 6 [19].

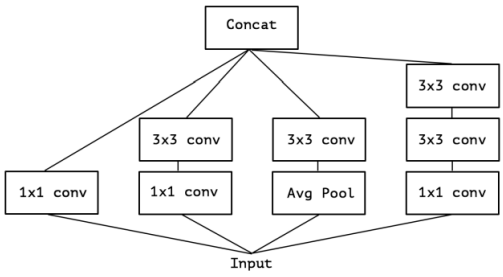

Figura 6. Módulo inception de uma rede Inception V3 [19].

Com isso, a rede Xception proposta por François Chollet, é uma melhoria desse conceito. O objetivo é fazer uma versão

"extrema"do bloco inception, aplicando um conceito chamado *depthwise separable convolutions*. A versão extrema de um bloco inception seria aplicar um filtro convolutivo 1x1 nos dados de entrada para primeiramente mapear as relações entre os canais (uma imagem RGB tem 3 canais por exemplo) e em seguida separadamente realizar o mapeamento das correalações espaciais (altura e largura) de cada canal de saída, conforme na Figura 7 [19].

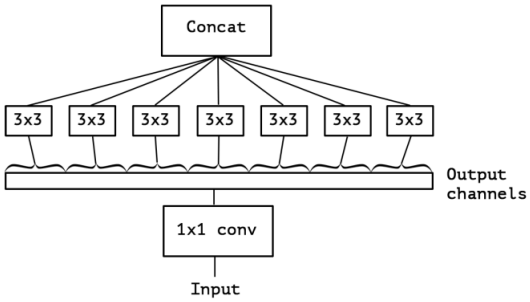

Figura 7. Versão "extrema"de um módulo inception [19].

H. U-net

U-net é uma RNC que tem como função realizar segmentação em imagens biomédicas. A arquitetura dessa rede consiste em duas partes, a primeira é chamada de caminho de contração dos dados ou *encoder* e o segundo de caminho de expansão dos dados ou *decoder*. O caminho de contração segue o típico padrão de uma RNC, que consiste da repetida aplicação de duas convoluções com filtros 3x3, seguido pela ativação com a ReLU e em seguida um max pooling 2x2 com stride 2, onde em cada operação de subamostragem é dobrado o número de canais de características . Já no caminho de expansão, cada passo consiste em um *upsampling* seguido por uma convolução 2x2 que divide o número de canais de características pela metade. A saída de um bloco de expansão é concatenada à saída do correnponde bloco de contração. Por fim, na camada final é feita uma convolução com kernel 1x1 [17]. O algoritmo pode ser melhor vizualizado na Figura 8.

III. Metodologia

A metodologia deste trabalho consistiu na realização de dois experimentos. O primeiro foi a implementação de modelos de RNC responsáveis pela classificação das imagens radiográficas. As redes implementadas foram a VGG-19, Densenet-121 e Xception. Por outro lado, no segundo experimento, foram implementadas as redes UNet, ResUnet e UNet com um *encoder* pré-treinado, que é conhecido como *Backbone*, para realizar a segmentação da região pulmonar. Sendo assim, após a segmentação das imagens, a melhor rede do primeiro experimento será usada novamente para a classificação das imagens, com o objtetivo de avaliar se há ganho de desempenho ao utilizar segmentação.

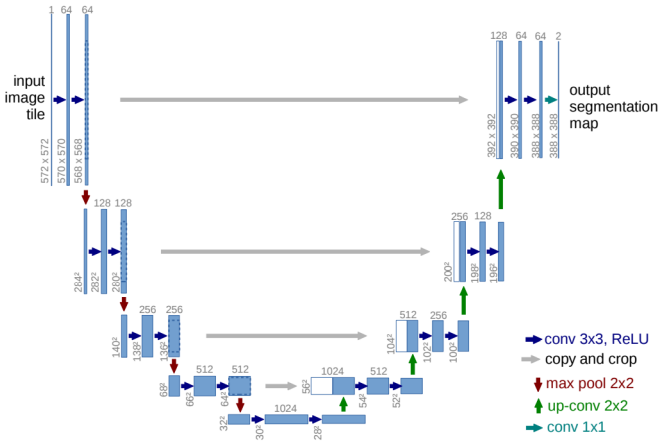

Figura 8. Arquiterua de uma rede U-net [17].

A. Conjunto de imagens (dataset)

O *dataset* utilizado nesse trabalho foi desenvolvido em 2021 por pesquisadores da Qatar University, ganhando o prêmio de melhor *dataset* do ano, eleito pela comunidade do Kaggle. Ele possui 33.920 imagens radiográficas do tórax, sendo distribuídas em três diferentes classes da seguinte forma:

- Covid-19: Essa é a classe de imagens radiográficas de pacientes diagnósticados com Covid-19 e possui um total de 11.956 imagens;
- *Non-COVID infections*: Essa classe é caracterizada por possuir imagens radiográficas de pacientes que estão infectadas por doenças com sintomas parecidos com os da Covid-19, como infecções virais ou pneumonia bacteriana. Essa classe é de extrema importância, pois além dos sintomas, os exames radiográficos também são muito parecidas com o Covid. É essencial que o modelo seja capaz de identificar os padrões que distinguem as duas classes. O total de imagens dessa classe é 11.263;
- Normal: Essa é a classe das imagens radiográficas de pessoas que não possuem nenhuma infecção pulmonar, contendo 10.701 imagens.

O *dataset* é balanceado, visto que as três classes possuem aproximadamente 11 mil imagens. Além disso, o *dataset* possui um subconjunto com todos os *ground-truths* da segmentação da região do pulmão de todas as 33.920 imagens.

B. Experimento 1

No primeiro experimento foram implementadas e testadas as redes VGG-19, DenseNet-121 e Xception, com o objetivo de escolher o melhor modelo, o qual será usado novamente no segundo experimento. Vale ressaltar que os três modelos foram configurados com pesos pré-treinados usando o *ImageNet*, que é um banco de imagens amplamente utilizado, criado para a área de processamento de imagens com *deep learning*. O nome dessa técnica é *transfer learning* e ela garante um treinamento mais eficiente, já que a maior parte

da rede já está treinada. A Figura 9 ilustra a estrutura desse experimento.

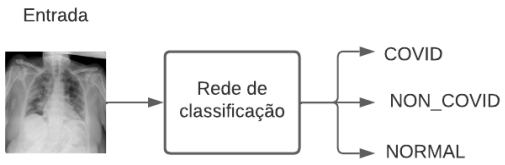

Figura 9. Sequência de passos do experimento 1.

Para a criação e treinamento dos modelos foi utilizada a biblioteca Keras do Tensor Flow, tendo como ambiente de desenvolvimento o Google Colab. Ademais, o *dataset* utilizado nesse trabalho foi construído como sendo um *subdataset* do original. Essa redução foi necessária devido a duas limitações computacionais da versão gratuita do Colab. A primeira é a baixa quantidade de memória RAM disponibilizada, impossibilitando que as imagens e os milhões de pesos das redes fossem carregados simultaneamente na etapa de treinamento, impedindo a execução do código, e a segunda, a baixa capacidade de processamento da máquina. No entanto, mesmo com essa redução, as redes cujo treinamento é mais longo, como a VGG-19, Unet com *backbone* e ResUnet, duraram entre 8 a 15 horas para serem treinadas. Além disso, também foram testadas variações na arquitetura e na taxa de aprendizado das redes, tornando inviável a utilização de uma grande quantidade de imagens. Portanto, das mais de 30 mil imagens do *dataset* original, foram utilizadas apenas 3 mil, contendo mil imagens de cada classe. As imagens foram separadas seguindo a proporção de 70% para treinamento, 15% para validação e 15% para teste, totalizando 2,1 mil imagens para treinar os três modelos, 450 para validar e 450 para testar.

A função de ativação utilizada nas camadas da rede foi a *Rectified Linear Unit* (ReLU), enquanto que na última camada completamente conectada foi utilizada a função de ativação *Softmax*. Para a atualização dos pesos foi utilizado o algoritmo de otimização *ADAM*, que é uma extensão do *stochastic gradient descent*, que recentemente vem sendo amplamente utilizado em aplicações de aprendizado profundo com visão computacional. As redes foram testadas com três variações de taxas de aprendizado, sendo elas 0.001, 0.0005 e 0.0001, e termo de *momentum* com valor padrão do Keras, $\beta_1 = 0.9$ e $\beta_2 = 0.999$. Em relação ao pré-processamento das imagnes, foram feitos apenas alguns procedimentos básicos, como por exemplo, normalização das imagens.

Por fim, para a implementação dessas redes, a abordagem mais utilizada é utilizar apenas as camadas convolucionais, pois tais camadas funcionam como um extrator de características. As camadas posteriores, chamadas *Dense layers*, *Top layers* ou *Fully connected layers*, são as responsáveis por utilizar essas características extraídas para realizar a classificação. Como essas redes são utilizadas para resolver problemas com mil classes, a ideia é que as *dense layers* originais do modelo sejam removidas e novas camadas, ainda não treinadas, sejam adicionadas. Assim, a topologia da rede é ajustada para o contexto deste problema, que contém apenas três classes. Para a determinação da topologia adequada das novas camadas, foram treinadas e testadas quatro variações, onde a menor topologia contém apenas uma camada oculta de 32 neurônios e a maior contém 4 camadas: 256-128-64-32. A ideia é sempre adicionar uma camada com o dobro de neurônios, e a última camada, a camada de saída contendo três neurônios e função de ativação *Softmax*.

C. Experimento 2

O objetivo desse experimento é verificar se a segmentação das imagens contribui ou não para o aumento do desempenho da classificação da rede, a qual será escolhida a partir dos resultados do primeiro experimento. Esta ideia é baseado no trabalho de [14], que usou a rede U-Net para segmentar as imagens e utilizou a VGG-19 como classificador, e de acordo com os autores, a segmentação melhorou os resultados de classificação, apresentando uma acurácia de aproximadamente 97%.

A segmentação da região pulmonar da imagem radiográfica consiste em remover pixels que não possuem relevância para o contexto do problema, tais como, a região do ombro, pescoço, etc. Ao eliminar estas informações das imagens, o treinamento da rede é facilitado, pois a rede não necessitará aprender padrões desnecessários, e consequentemente, espera-se que sua acurácia seja aumentada. Vale destacar também, que algumas informações textuais, provavelmente geradas pelos aparelhos utilizados nos exames, e que são desnecessárias para o problema de classificação, também são eliminadas. A Figura 10 ilustra a estrutura desse experimento.

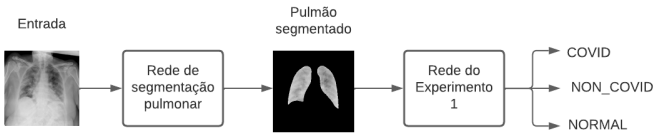

Figura 10. Sequência de passos do experimento 2.

As redes utilizadas para realizar a segmentação das imagens foram UNet, ResUnet e UNet com um *encoder* pré-treinado, que é conhecido como *Backbone*. Para a implementação da Unet com *Backbone*, foi utilizada a rede pré-treinada Densenet-121, e isso deve-se ao fato dela ter um alto desempenho em trabalhos semelhantes, os quais foram citados na Seção I, e a um baixíssimo tempo de treinamento, conforme descrito nos resultados da Seção IV-A.

Para treinar as redes foram utilizadas as mesmas imagens do primeiro experimento, no entanto, há uma diferença, que é a saída desejada. No primeiro experimento, a saída desejada era uma das três classes possíveis, ou seja, paciente com Covid-19, paciente com outra doença e paciente normal. Porém, a saída desejada para as redes que farão a segmentação

Tabela II
DEFINIÇÃO DOS BLOCOS ENCODER E DECODER.

Bloco/camada	Filtro	Saída do bloco/camada
Input		256x256x3
Encoder 1	3x3/16	128x128x16
Encoder 2	3x3/32	64x64x32
Encoder 3	3x3/64	32x32x64
Encoder 4	3x3/128	16x16x128
Bridge 1	3x3/256	8x8x256
Bridge 2	3x3/256	8x8x256
Decoder 1	3x3/128	16x16x128
Decoder 2	3x3/64	32x32x64
Decoder 3	3x3/32	64x64x32
Decoder 4	3x3/16	128x128x16
Output	1x1	256x256x1

são os *ground truths* das imagens, que são disponibilizados pelo próprio *dataset*. Os *ground truths* é a resposta esperada após a segmentação da imagem.

Todas as três redes foram implementadas com cinco blocos *encoder* e cinco blocos *decoder*. Não existe um número correto de blocos, mas foi selecionado uma quantidade que garantisse um tempo de treinamento menor do que 15 horas. Pelas mesmas razões citadas para a quantidade de blocos, a quantidade de filtros foi estabelecida e é detalhada na Tabela II.

A partir da Tabela II, pode-se notar que todos os filtros têm dimensão 3x3. O que muda de um bloco para outro é a quantidade de filtros, que começa com 16 e dobra a cada bloco, sendo que o quinto bloco contém 256 filtros. Além disso, as dimensões dos dados são reduzidos pela metade nos codificadores, visto que é aplicado uma operação de *Max Pooling* com *stride* de 2. Nos decodificadores ocorre o processo inverso, os dados vão dobrando de largura e altura, pois é realizado um *upsampling*. Por último, é importante notar que diferente do que é descrito na Seção II, essa implementação não recorta as saídas dos blocos de codificação para a altura e largura se adequarem às saídas dos blocos de decodificação, antes de realizar a concatenação dos mesmos, porque as linhas e colunas reduzidas são preenchidas com zero, para voltar à sua dimensão anterior, ou seja, antes de passarem pela convolução.

A partir do momento que as três redes foram treinadas e avaliadas, a rede com maior *Intersection Over Union (IoU)* (ver Seção III-D) é selecionada para segmentar as imagens. Essa rede é utilizada para realizar o processo de segmentação das 3 mil imagens utilizadas no experimento 1, de forma a criar um novo *dataset* de 3 mil imagens, mas dessa vez, todas segmentadas.

D. Métricas de avaliação de desempenho

Para avaliar o desempenho das redes neurais utilizadas nesse trabalho foram aplicadas as métricas, acurácia, precisão, *recall* e *intersection over union* (IoU), que são definidas pelas Equações 4, 5, 6 e 7, respectivamente. Essas são as métricas mais utilizadas na área médica e em diversas áreas de aplicação de inteligência artificial.

$$\text{Acurácia} = \frac{VP + VN}{VP + FN + FP + VN} \tag{4}$$

$$\text{Precisão} = \frac{VP}{VP + FP} \tag{5}$$

$$\text{Recall} = \frac{VP}{VP + FN} \tag{6}$$

$$\text{IoU} = \frac{VP}{VP + FP + FN} \tag{7}$$

onde VP, VN, FP e FN significam, respectivamente, verdadeiro positivo, verdadeiro negativo, falso positivo e falso negativo.

IV. RESULTADOS E DISCUSSÕES

Nesta seção serão apresentados os resultados e discussões dos experimentos 1 e 2.

A. Experimento 1

1) Arquitetura dos modelos: Como já mencionado na Seção III, para cada uma das três redes foram treinadas e testadas quatro variações de arquiteturas para as camadas completamente conectadas. A Figura 11 mostra a melhor configuração obtida das redes. A rede Exception e a VGG-19 apresentaram melhor desempenho com três camadas ocultas, com 128, 64 e 32 neurônios, respectivamente. A rede Densenet-121 apresentou melhor desempenho com duas camadas ocultas, contendo 64 e 32 neurônios, respectivamente. Além disso, para as três redes, antes das camadas escondidas, foi adicionado uma camada *Flatten*, que basicamente transforma a saída do extrator de características em um vetor unidimensional, cujos comprimentos são ilustrados na Figura 11. Também foi citado na Seção III que para cada rede foram testadas três variações de taxas de aprendizado. Os melhores valores para cada rede foram: VGG-19 = 0.0005, Densenet-121 = 0.0001 e Exception = 0.001.

2) Comparação dos modelos: O desempenho dos modelos foram mensurados pelas métricas acurácia, precisão, *recall*, *loss* e tempo de treinamento. As Tabelas III, IV e V detalham o desempenho das redes no treinamento, validação e teste, respectivamente. Os melhores resultados estão destacados em cinza.

Tabela III
MÉTRICAS OBTIDAS COM AS IMAGENS DE TREINAMENTO.

Rede	Acurácia	Precisão	Recall	Loss
VGG-19	94.09	94.68	93.38	0.41
Densenet-121	99.95	99.95	99.95	0.46
Xception	89.76	92.33	86.32	0.27

Na Tabela III, que apresenta o desempenho dos modelos durante o treinamento, a rede Densenet-121 se destacou, apresentando acurácia, *recall* e precisão próximas a 100%.

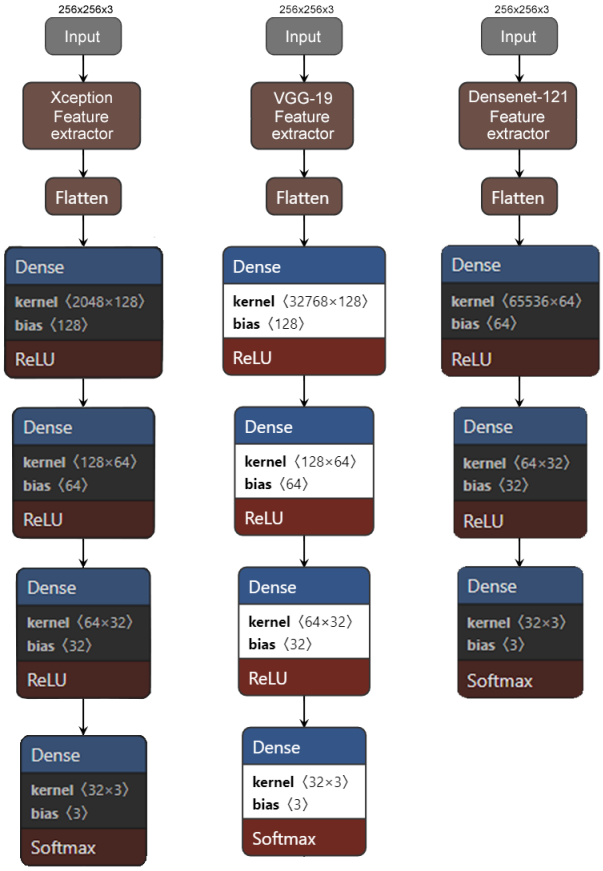

Figura 11. Melhores topologias das camadas totalmente conectadas de cada rede

Tabela IV
MÉTRICAS OBTIDAS COM AS IMAGENS DE VALIDAÇÃO.

Rede	Acurácia	Precisão	Recall	Loss
VGG-19	82.44	83.39	81.33	0.52
Densenet-121	74.66	76.44	73.55	0.73
Xception	75.99	78.86	71.33	0.70

Tabela V
MÉTRICAS OBTIDAS COM AS IMAGENS DE TESTE.

Rede	Acurácia	Precisão	Recall	Loss
VGG-19	84.00	84.89	82.44	0.51
Densenet-121	78.84	80.13	77.11	0.78
Xception	74.44	77.34	71.33	0.70

No entanto, na Tabela IV, onde estão descritos os resultados utilizando as imagens do conjunto de validação, ou seja, imagens inéditas para as redes, nota-se que a rede Densenet-121 sofreu a maior queda de desempenho, diminuindo em mais de 20% na acurária. Este decréscimo também ocorreu com a rede Xception, com menor intensidade, mas suficiente para reduzir o desempenho para valores menores que 80%. Por fim, a rede VGG-19 se destacou, apresentando acurácia, precisão e *recall* superiores a 80%, com *loss* consideravelmente menor do que as outras, o que confirma a boa generalização da rede.

A Tabela V exibe os resultados do teste, e nela é observado um desempenho semelhante ao apresentado na validação, ou seja, a rede VGG-19 novamente com desempenho superior em todas as métricas, em relação às demais redes. Neste caso, a rede Densenet-121 no teste foi um pouco superior a Xception.

Em relação ao tempo de treinamento, a Tabela VI exibe quanto cada modelo gastou (em segundos) para concluir uma época de treinamento.

Tabela VI
TEMPO DE 1 ÉPOCA DE TREINAMENTO.

VGG-19	Densenet-169	Xception
2714s	670s	813s

Observa-se que o VGG-19 possui um tempo de treinamento muito alto, cerca de quatro vezes maior do que o Densenet-121, que apresentou o melhor tempo. O Xception também tem um bom tempo, apenas 143 segundos a mais que o Densenet-121. Entretanto, apesar do alto tempo de treinamento, a rede selecionada para esse experimento foi a VGG-19, devido a sua superiodade na classificação das imagens, conforme apresentado nas Tabelas IV e V.

B. *Experimento 2*

Nesse segundo experimento, foi necessário realizar duas avaliações. A avaliação das redes de segmentação e da rede do Experimento 1, retreinada com o novo *dataset* das imagens segmentadas. Para a rede de classificação retreinada com novas imagens foram utilizadas as mesmas métricas de desempenho do experimento 1, enquanto que para as redes de segmentação, foi necessário acrescentar a métrica *Interception Over Union* (IoU), por ser a métrica mais utilizada para esse tipo de rede.

1) Modelos de segmentação: As Tabelas VII, VIII e IX, exibem os resultados obtidos com as redes de segmentação UNet, UNet com *Backbone* (Unet B.) pré-treinada e ResUnet, nos processos de treinamento, validação e teste, respectivamente.

Tabela VII
MÉTRICAS OBTIDAS COM AS IMAGENS DE TREINAMENTO.

Rede	IoU	Acurácia	Precisão	Recall
UNet	88.42	98.23	96.59	96.10
UNet B.	93.24	98.96	98.53	97.15
ResUnet	87.92	97.91	97.17	87.92

Na Tabela VII percebe-se que as três redes apresentaram um desempenho muito elevado, com acurácia acima de 97%, além de precisão e *recall* com valores muito próximos também, mas o IoU evidencia uma perceptível vantagem da

Tabela VIII
MÉTRICAS OBTIDAS COM AS IMAGENS DE VALIDAÇÃO.

Rede	IoU	Acurácia	Precisão	Recall
Unet	88.70	98.43	96.80	96.60
Unet B.	93.37	98.93	98.37	97.13
ResUnet	89.26	98.24	97.52	95.08

Tabela IX
MÉTRICAS OBTIDAS COM AS IMAGENS DE TESTE.

Rede	IoU	Acurácia	Precisão	Recall
UNet	88.54	98.31	96.24	96.59
UNet B.	92.52	98.82	98.02	96.91
ResUnet	88.11	98.12	96.98	94.95

Unet com *Backbone*, sendo a única com valor acima de 90%. A Tabela VIII mostra que as três redes conseguiram generalizar bem para imagens ainda desconhecidas pelas redes. Os resultados são praticamente idênticos aos do treinamento, o que é excelente. A UNet com *Backbone* continua sendo a melhor em todas as métricas, principalmente na IoU. Por fim, com o conjunto de teste, mais uma vez o melhor modelo foi a UNet com *Backbone*. Portanto, diante dos resultados obtivos, a rede Unet B. foi escolhida para segmentar as imagens e gerar o novo *dataset*.

As Figuras 12 e 13 colocam lado-a-lado uma imagem de entrada, sua respectiva máscara *ground truth* e a máscara gerada pelas redes Unet e Unet com *Backbone*, respectivamente.

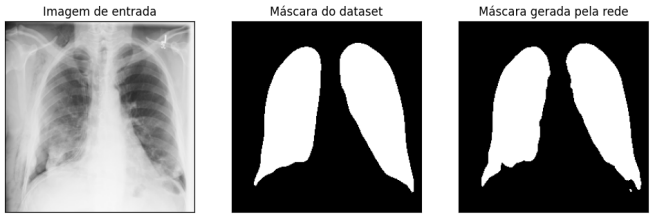

Figura 12. Máscara gerada pelo UNet.

Figura 13. Máscara gerada pelo UNet com *Backbone* pré-treinado.

Na figura 12, observa-se que a máscara gerada pela Unet é bem similar ao seu *ground truth*, mas há alguns ruídos nas bordas da máscara. Já na Figura 13, a máscara gerada pela UNet com *Backbone* é praticamente idêntica, não contendo ruídos. Esse padrão se repetiu para a maioria das imagens segmentadas, o que mostra que a acurácia não é uma métrica ideal para avaliar modelos de segmentação, afinal o valor da acurácia das duas é muito próximo, ao contrário do IoU, que destaca a superioridade da Unet com *Backbone*.

No processo de segmentação das imagens, a máscara é utilizada para segmentar o pulmão. Um exemplo de imagem gerada pode ser observado na Figura 14. Essa imagem foi gerada realizando uma operação AND entre a imagem de entrada e a máscara da Figura 13.

Figura 14. Imagem do pulmão segmentado a partir da rede UNet com *Backbone* pré-treinado

2) Avaliando a VGG-19 retreinada com o novo dataset: A Tabela X apresenta os resultados obtidos com a melhor rede do Experimento 1, a VGG-19, mas dessa vez treinada com o *dataset* contendo as imagens radiográficas segmentadas, ou seja, contento apenas a região do pulmão.

Tabela X
MÉTRICAS DA VGG-19 TREINADA COM IMAGENS SEGMENTADAS.

Etapa	Acurácia	Precisão	Recall	Loss
Treino	65.66	72.79	52.00	0.76
Validação	51.77	60.70	38.44	0.98
Teste	53.77	60.59	40.46	0.98

Os valores apresentados mostram que há uma queda de desempenho enorme em relação a rede treinada com o *dataset* original, pois sofreu uma redução de mais de 30% em quase todas as métricas. Isso indica que provavelmente o modelo não extraiu as características apenas da região pulmonar para realizar a classificação, o que é estranho, já que a doença é localizada especificamente no pulmão. Com isso, esse experimento mostra ser inviável aplicar segmentação nas imagens desse projeto, pois foi extremamente nocivo ao desempenho do modelo, indo na direção contrária ao que foi relatado em trabalhos semelhantes citados na Seção I, que relataram melhora no desempenho.

V. Conclusão

O objetivo desse trabalho foi analisar o desempenho de três Redes Neurais Convolucionais aplicacadas no problema do diagnóstico da Covid-19, a partir de imagens de exames radiográficos do tórax. Mais especifiamente, foram comparadas três RNC pré-treinadas, amplamente utilizadas para classificação de imagens, sendo elas VGG-19, Xception e Densenet-121. Em um primeiro cenário de avaliações, todas estas redes foram treinadas, validadas e testadas com diferentes topologias para as camadas completamente conectadas das três redes. A rede que obteve os melhores resultados neste primeiro cenário foi a VGG-19.

Em seguida, um novo experimento foi realizado, utilizando a VGG-19, que foi a melhor rede do primeiro experimento, com um novo conjunto de imagens segmentadas. O objetivo deste experimento foi analisar se a segmentação das imagens poderia melhorar o desempenho de classificação da rede VGG-19. A segmentação das imagens foi realizada pela rede Unet com *Backbone*, pois foi a melhor em comparação com as redes UNet e ResUnet. No final da execução do experimento 2, obtivemos um resultado inesperado, pois a rede de segmentação acabou prejudicando o desempenho da rede. Esse resultado foi inesperado, pois a ideia de criar um módulo de segmentação foi utilizado nos trabalhos [14] e [10], e os autores relataram melhoria nas métricas.

Para futuras melhorias, pode ser realizado novos experimentos removendo as camadas totalmente conectadas da VGG-19 do experimento 1, substituindo-as por outros algoritmos de classificação, tais como, *Support Vector Machine*, *Árvore de decisão* ou *Random Forest Classifier*.

Referências

[1] World Health Organization, "Who coronavirus (covid-19) dashboard," https://covid19.who.int/?gclid=Cj0KCQjwtZH7BRDzARIsAGjbK2ZXWRpJROEl97HGmSOx0_ydkVbc02Ka1FlcysGjEI7hnaIeR6xWhr4aAu57EALw_wcB, 2023.

[2] T. Singhal, "A review of coronavirus disease-2019 (covid-19)," *Elsevier*, 2020.

[3] C. Sohrabi and et al., "World health organization declares global emergency: A review of the 2019 novel coronavirus (covid-19)," *The Indian Journal of Pediatrics*, 2020.

[4] J. Xia MM and et al., "Evaluation of coronavirus in tears and conjunctival secretions of patients with sars-cov-2 infection," *Wiley Online library*, 2020.

[5] A. Tao and et al., "Correlation of chest ct and rt-pcr testing for coronavirus disease 2019 (covid-19) in china: A report of 1014 cases," *Radiological Society of North America*, 2020.

[6] B. David J. and et al., "Computed tomography an increasing source of radiation exposure," *The New England Journal of Medicine*, 2007.

[7] H. Chaolin and et al., "Clinical features of patients infected with 2019 novel coronavirus in wuhan, china," *Elsevier*, 2020.

[8] L. Wang, Z. Q. L., and A. Wong, "Covid-net: a tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images," *Nature - Scientific Reports*, 2020.

[9] A. M. Tahir, M. E. Chowdhury, A. Khandakar, T. Rahman, Y. Qiblawey, U. Khurshid, S. Kiranyaz, N. Ibtehaz, M. S. Rahman, S. Al-Maadeed, S. Mahmud, M. Ezeddin, K. Hameed, and T. Hamid, "Covid-19 infection localization and severity grading from chest x-ray images," *Elsevir - Computers in Biology and Medicine*, 2021.

[10] R. Hertel and R. Benlamri, "A deep learning segmentation-classification pipeline for x-ray-based covid-19 diagnosis images," *Elsevier - Biomedical Engineering Advances*, 2022.

[11] N. Azouji, A. Sami, M. Taheri, and H. Müller, "A large margin piecewise linear classifier with fusion of deep features in the diagnosis of covid-19," *Elsevier - Computers in Biology and Medicine*, 2021.

[12] S. Ahmadian, S. Mohammad, J. Jalali, S. M. S. Islam, A. Khosravi, E. Fazli, and S. Nahavandi, "A novel deep neuroevolution-based image classification method to diagnose coronavirus disease (covid-19)," *Elsevier - Computers in Biology and Medicine*, 2021.

[13] X. Yu, S. Lu, L. Guo, S.-H. Wang, and Y.-D. Zhang, "Resgnet-c: A graph convolutional neural network for detection of covid-19," *Elsevier - Neurocomputing*, 2021.

[14] D. Arias-Garzón, J. A. A., S. Orozco-Arias, H. B. A., M. A. Bravo-Ortiz, A. Mora-Rubio, J. M S., J. Á. M. S., M. de la I. V., O. Cardona-Morales, and R. Tabares-Soto, "Covid-19 detection in x-ray images using convolutional neural networks," *Elsevier - Machine Learning with Applications*, 2021.

[15] S. Haykin, *Redes Neurais - Princípios e Prática*. Bookman, 2001.

[16] H. H. A. and E. J. H., *Guide to Convolutional Neural Networks - A Practical Application to Traffic-Sign Detection and Classification*. Springer, 2017.

[17] L. Bezerra Maia, "Aprendizagem profunda aplicada ao diagnóstico de melanona," *Dissertação (Programa de Pós-Graduacção em Ciência da Computação / CCET) - Universidade Federal do Maranhao, São Luís.*, 2019.

[18] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[19] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.