

Capítulo

4

Classificação e Segmentação de COVID-19 em Imagens de Tomografia Computadorizada Usando Aprendizado Profundo

Júlio V M Marques, José F C Ferreira, Rodrigo M S Veras and Romuere R V Silva

Abstract

In developing the disease COVID-19, various means for diagnosis have been widely studied, so we arrive at medical imaging tests, where the chest tomography exam has the best level of detail among other imaging exams. However, the analysis of these images generates a repeatable and tiring workload, requiring a team of specialists familiar with the indicative findings of pneumonia caused by COVID-19. To reduce this manual work and collaborate with the specialists, this chapter presents pre-processing methods and techniques for classification and segmentation of these images, reaching 99% accuracy for classification and 87% Dice for segmentation. All this to collaborate with a theoretical and practical basis for future works.

Resumo

Com o desenvolvimento da doença COVID-19 vários meios para o diagnóstico vem sendo amplamente estudados, assim chegamos nos exames médicos por imagens, onde o exame de Tomografia de Tórax detém o melhor nível de detalhamento entre os demais exames por imagens. Porém a análise dessas imagens gera uma carga de trabalho repetível e cansativa, precisando também de uma equipe de especialistas que estejam familiarizados com os achados indicativos da pneumonia causada pela COVID-19, afim de reduzir esse trabalho manual e colaborar com os especialistas, esse capítulo apresenta métodos de pré-processamento, técnicas para classificação e segmentação dessas imagens, podendo chegar a 99% acurácia para classificação e 87% de Dice para segmentação. Tudo isso afim de colaborar com uma base teórica e prática para futuros trabalhos.

4.1. Introdução

A COVID-19 é uma doença causada pelo novo coronavírus (SARS-COV-2), essa doença é altamente infecciosa com diversos sintomas que são facilmente confundidos com outras doenças, tais como a pneumonia viral ou bacteriana [Samir et al. 2021]. A COVID-19 apresenta diversos sintomas que podem variar em cada paciente, podendo apresentar quadros clínicos leves e recuperações rápidas. Porém, em outros casos podem apresentar problemas de saúde graves, com pneumonia, insuficiência respiratória, falência de órgãos e outras complicações [Kazimierczuk and Jozwik 1990]. O número global de casos e mortes para essa doença somam mais de 500 milhões e mais de 6 milhões em maio de 2022, respectivamente [Organization 2022]. Em meio a tantos casos e pacientes com diversos tipos de sintomas diferentes, o diagnóstico em estágio inicial é imprescindível para conseguir aumentar a taxa de sobrevivência dos pacientes e evitar a contaminação de mais pessoas.

Para realizar o diagnóstico em estágio inicial é utilizado o exame da reação em cadeia da polimerase com transcrição reversa em tempo real (RT-PCR) com cotonetes faríngeos [Wang et al. 2020], esse teste foi adotado como padrão em todo o mundo. Porém, com o grande número de casos registrados diariamente a capacidade laboratorial de muitos municípios e estados ficam sobrecarregados, causando a demora no diagnóstico inicial e dificultando o acesso a cuidados, podendo contribuir para o agravamento clínico dos pacientes [Dananché et al. 2022].

Para apoiar o trabalho dos especialistas responsáveis por diagnosticar a doença COVID-19, outros métodos de prevenção, avaliação, análises e diagnóstico têm sido extensivamente explorados e estudados [Martínez Chamorro et al. 2021]. Com isso o uso de métodos como os exames de radiografia e Tomografia Computadorizada (TC) estão sendo amplamente utilizados para acompanhar a progressão dessa doença, com isso é possível realizar observações de pacientes com COVID-19 que estão com pioras no estado respiratório e em ambientes com poucos recursos na triagem médica de pacientes com suspeita de ter COVID-19 [Rubin et al. 2020]. Esses testes podem contribuir com evidências para a identificação da COVID-19 [Sverzellati et al. 2021].

A radiografia geralmente tem a maior disponibilidade, sendo a mais indicada para avaliações de regiões como o tórax, além de ser o exame mais indicado para detectar a pneumonia viral causada pelo mesmo vírus da gripe. No entanto, para a COVID-19, a radiografia apresenta baixa sensibilidade e especificidade comparada com a Tomografia de Tórax [Stephanie et al. 2020]. Portanto, a Tomografia Computadorizada, apresenta-se como um método eficaz, sendo um meio alternativo e confiável para detectar lesões causadas pela COVID-19 devido ao seu nível de detalhes [Axiaq et al. 2021]. Porém, a análise das imagens geradas pelo exame da TC de tórax requer um grande esforço manual devido a quantidade de imagens geradas durante esse exame, o que torna o trabalho e a análise de muitos exames desgastante, exigindo também uma equipe que esteja familiarizada com os achados nas imagens que são sugestivas a características de pneumonia viral compatível com a COVID-19 [Rosa et al. 2020].

Pensando nisso, o presente capítulo tem como objetivo classificar e segmentar as regiões das lesões causadas pela COVID-19. Com a classificação das imagens os especialistas podem focar apenas nas análises das imagens classificadas como positivas

para COVID-19, diminuindo assim a quantidade de imagens a serem analisadas. Com a segmentação das lesões, pretendemos aliviar o esforço manual dos especialistas, sendo possível contribuir com as marcações automática das lesões para facilitar as análises, além de dar apoio na visualização da progressão dessa doença nos pulmões, podendo assim acompanhar o quadro clínico a partir das marcações feitas.

Para a tarefa de segmentação, iremos utilizar a U-Net [Ronneberger et al. 2015b], juntamente com técnicas para o pré-processamento proposto que é capaz de trabalhar com vários tipos de imagens provenientes da TC de tórax de várias fontes, aparelhos e aquisições diferentes. Com a união da U-Net e o método de pré-processamento das imagens pretendemos segmentar as regiões das lesões causadas pela COVID-19. Para a classificação a VGG-19 [Simonyan and Zisserman 2014] é utilizada com transferência de aprendizado, juntamente com um modelo baseado na arquitetura LeNet-5 [Lecun et al. 1998]. Para a classificação os modelos também utilizam as imagens com o método de pré-processamento proposto e as imagens originais.

A metodologia proposta nesse capítulo apresenta as seguintes contribuições: 1) Metodologia de pré-processamento para trabalhar com imagens de várias fontes de TC de tórax, aparelhos e aquisições diferentes; 2) Avaliação do modelo U-Net para o problema proposto; 3) Avaliação do modelo de transferência de aprendizado com o modelo baseado na LeNet;

O restante do capítulo está organizado da seguinte forma. A Seção 1.2 descreve as imagens médicas. A Seção 1.3 descreve a aquisição de imagens. A Seção 1.4 descreve as técnicas de pré-processamento utilizadas. A Seção 1.5 descreve a área da inteligência artificial, juntamente com as técnicas de aprendizado profundo utilizadas. A Seção 1.6 apresenta os experimentos. A Seção 1.7 descreve os resultados obtidos. A Seção 1.8 temos uma discussão sobre os resultados. A Seção 1.9 apresenta as conclusões desse capítulo, e por fim na Seção 2.0 temos os agradecimentos finais desse capítulo.

4.2. Imagens Médicas

Os exames por imagens são uma especialidade médica que faz uso das tecnologias de imagens com o intuito de realizar o diagnóstico de vários tipos de doenças. Esses exames são utilizados para obter imagens do interior do corpo humano e realizar a análise clínica, com isso essas técnicas vem sendo aprimoradas, podendo mostrar com particularidades a estrutura do corpo humano e também são capazes de exibir os tecidos da parte interna do corpo humano [Clifford 2006].

Podemos citar alguns dos exames por imagens mais utilizados no diagnóstico de doenças por imagem, como os exames de radiografias (Raios-X), ultrassonografia (USG), mamografia, Tomografia Computadorizada (TC), ressonância magnética, tomografia de coerência óptica (OCT), dermatoscopia, mapeamento de retina e o exame histopatológico. Na Figura 4.1, apresentamos alguns exemplos desses exames.

Com as constantes evoluções, os exames por imagens estão sendo cada vez mais semelhantes à anatomia real do corpo humano, as tecnologias como TC e USG foram bastante usadas e aprimoradas, apesar disso a radiologia convencional é constantemente utilizada para auxiliar em diagnósticos e nas escolhas das maneiras de tratamentos a serem

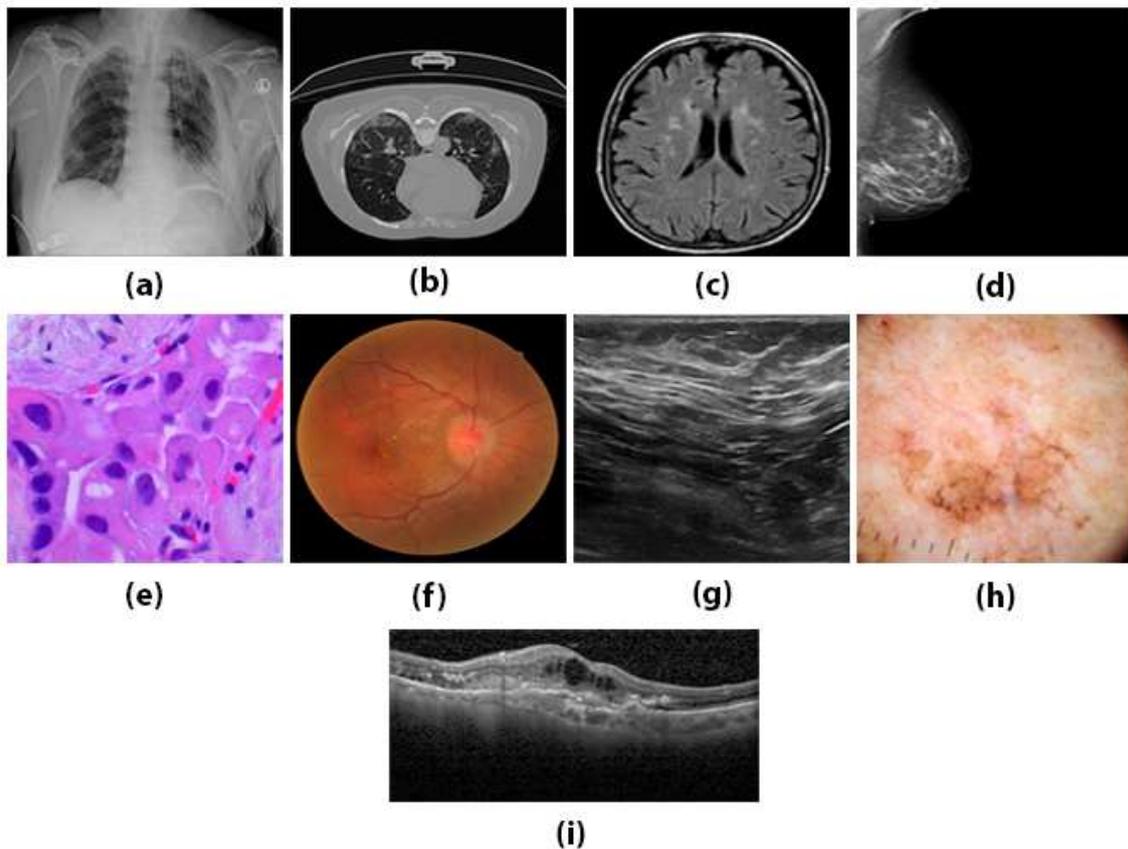


Figura 4.1. Representações dos exames mais utilizados e difundidos. Em (a), temos a radiografia, em (b) Tomografia Computadorizada do tórax, em (c) ressonância magnética, em (d) mamografia, em (e) exame histopatológico, em (f) mapeamento de retina, em (g) ultrassonografia, em (h) dermatoscopia e em (i) temos a coerência óptica.

utilizadas, principalmente em momentos de urgência [Moreira et al. 2017]. Com base na literatura, os exames por imagem mais utilizados para a doença da COVID-19 são a radiografia do tórax e a TC de tórax.

4.2.1. Radiografia de Tórax

A Radiografia (Raio-X) é o método de aquisição de imagens médicas mais antigo e possivelmente o mais utilizado até os dias de hoje, devido principalmente pelo seu custo e disponibilidade. Para formar a imagem, o equipamento lança feixes de Raio-X que irão atravessar o local do corpo em que se deseja ser examinado. As diversas estruturas e órgãos realizam a absorção em diferentes quantidades, variando de sua intensidade e composição. Assim, os feixes que não são absorvidos atingem um filme sensível aos Raio-X, que produz uma imagem [Moreiran 2017].

Os padrões da doença pulmonar COVID-19 podem ser identificados na radiografia torácica convencional. A comunidade médica frequentemente se utiliza da radiografia devido à sua alta disponibilidade e sua facilidade de acesso, além de ser um método barato [Jacobi et al. 2020].

Na Figura 4.2, podemos observar as imagens geradas por esse exame, nela temos imagens de pacientes saudáveis, com COVID-19 e com pneumonia. Podemos observar pela imagem, que essas condições apresentadas pelos pacientes podem ser facilmente confundidas principalmente sem uma equipe especializada nos achados da COVID-19 em imagens médicas.

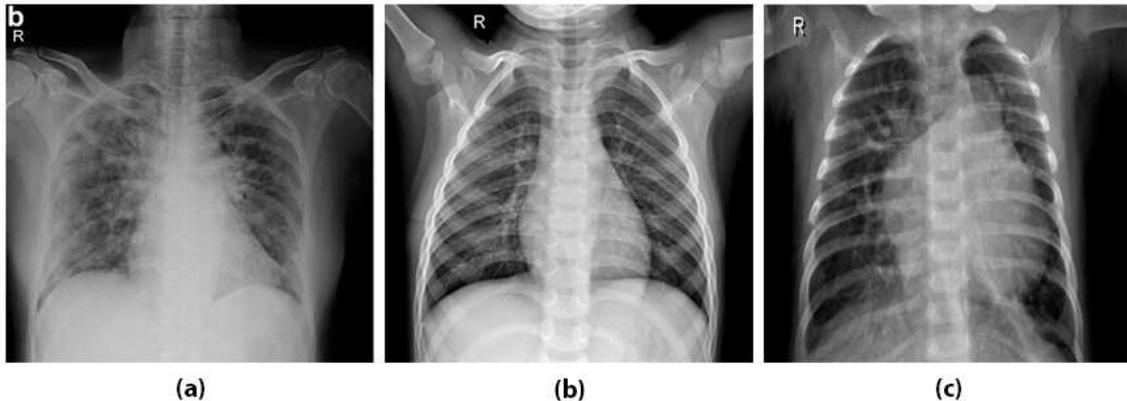


Figura 4.2. Visualização dos exames de raio-x do tórax, em (a), temos um pulmão infectado pela COVID-19, em (b), temos um pulmão saudável e em (c) temos um pulmão com pneumonia viral

4.2.2. Tomografia de Tórax

A Tomografia Computadorizada (TC) é uma técnica baseada em raio-x, ela utiliza uma fonte de raio-x e uma unidade de detecção que realiza movimentos circulares ao redor do paciente e recebe a radiação na área a ser estudada. No lado oposto da fonte, ficam localizados vários detectores que convertem a radiação em um sinal elétrico que logo é transformado em imagem digital [Júnior and Yamashita 2001].

A Radiografia de tórax é extremamente útil para avaliar a progressão da doença, pois permite a detecção de consolidações extensas, além de ser um método rápido e barato. Porém, a Tomografia Computadorizada é mais sensível para detectar alterações pulmonares nos estágios iniciais da doença e também é útil para avaliar a progressão da doença. A opacidade em vidro fosco são os principais achados relacionados ao COVID-19 e são mais facilmente identificados nas imagens de Tomografia de tórax [de Carvalho et al. 2021].

Na Figura 4.3, podemos visualizar que o pulmão saudável não apresenta manchas como podem ser visto no pulmão de um paciente com COVID-19 e no com pneumonia. Após uma análise na literatura, podemos identificar vários trabalhos que se utilizam de imagens de Tomografia de Tórax para detectar a COVID-19 com resultados robustos. Podemos destacar [Castiglione et al. 2021], [Ardakani et al. 2020] e [Zhou et al. 2021], onde os autores conseguiram atingir métricas como sensibilidade e especificidade acima dos 90%. Todos esses trabalhos contêm métodos que utilizam aprendizado profundo com pré-processamento de imagens que mostram resultados promissores, conseguindo detectar e classificar pacientes com COVID-19.

Assim como na classificação, vários trabalhos abordam a tarefa de segmentação das lesões causadas pela COVID-19, e para isso os trabalhos também se utilizam de

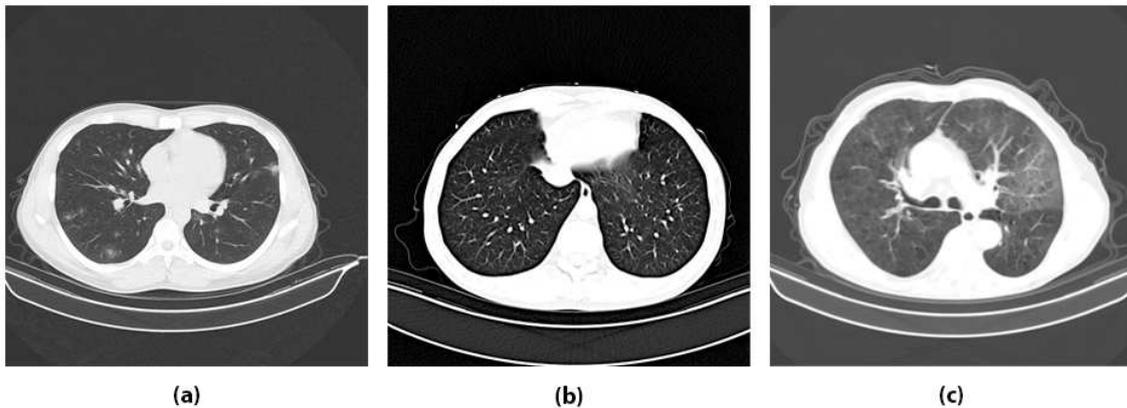


Figura 4.3. Visualização dos exames de tomografia do tórax, em (a), temos um pulmão infectado pela COVID-19, em (b), temos um pulmão saudável e em (c) temos um pulmão com pneumonia viral

imagens de TC, podemos destacar os trabalhos de [Cohen et al. 2020] e [Zhao et al. 2021], que conseguiram atingir métricas de 87,06% em Dice, 90,85% em Sensibilidade, 99,59% em Especificidade e 78,67% em Dice, respectivamente.

Após esses fatos e trabalhos da literatura pesquisados, as imagens de Tomografia de Tórax são utilizadas neste capítulo, tanto na tarefa de classificação, quanto na tarefa de segmentação.

4.3. Aquisição de Imagens

Para a tarefa de classificação ou segmentação com aprendizado profundo no contexto do novo coronavírus em imagens de Tomografia Computadorizada, um grande problema é a escassez de bases disponíveis de forma pública. Principalmente para a tarefa de segmentação, onde a maioria dos datasets disponíveis não possuem as marcações das lesões feitas por especialistas, o que dificulta mais ainda essa tarefa. Durante esse trabalho o COVID-19-CT-SEG [Ma et al. 2021] é utilizado para a tarefa de segmentação. Para a tarefa de classificação o dataset COVIDxCT dos autores [Gunraj et al. 2020], é utilizado. Nesse capítulo chamaremos os dataset COVID-19-CT-SEG e COVIDxCT de Dataset 1 e Dataset 2, respectivamente.

Para a tarefa de classificação o Dataset 2 é utilizado, e nele podemos encontrar um conjunto de dados de acesso aberto que é gerado a partir de vários conjuntos de dados abertos, compreende 194,922 slices de Tomografia Computadorizada no formato de *Portable Network Graphics* (PNG) de 3,745 pacientes. Esse conjunto de dados consiste em casos com diagnósticos confirmados (ou seja, RT-PCR, confirmados pelo radiologista, etc.). Nele é possível encontrar imagens para pacientes com COVID-19, pacientes com pneumonia e pacientes saudáveis. Nessa base de dados, as imagens já estão previamente divididas em treinamento, validação e teste, com 143,778, 25,486 e 25,658 imagens, respectivamente. Na Figura 4.4, podemos observar algumas dessas imagens que estão presentes dentro desse conjunto de dados.

Como esse dataset é construído a partir de várias outras fontes de dados, é bastante comum imagens de exames de TC de tórax com características e formas de aquisição



Figura 4.4. Visualização de três imagens presentes no Dataset 2, é possível observar a diferença do método de aquisição dos exames de TC do tórax entre cada uma delas.

diferentes estarem presentes no mesmo dataset. Podemos observar essa diferença na Figura 4.4, essas imagens são adquiridas de fontes de dados diferentes, com máquinas e métodos diferentes, gerando vários tipos de exames de TC de tórax presente neste dataset.

O Dataset 1, está disponível publicamente na plataforma Zenodo [Jun et al. 2020]. Possuindo um total de 20 exames de TC do tórax com marcações das lesões proveniente da COVID-19. Para cada exame temos as máscaras de segmentação dos pulmões, e as regiões de infecções. Essas marcações foram feitas por dois radiologistas e verificadas por um terceiro radiologista. Os exames têm quantidades de imagens diferentes, não são todos os exames que possuem o mesmo padrão referente ao aparelho utilizado para gerar tais imagens, ao todo a base de imagens possui 3,520 imagens.

Assim como no Dataset 2, as imagens do Dataset 1 também são provenientes de vários métodos de aquisições diferentes, os arquivos desse dataset estão no formato de dados *Neuroimaging Informatics Technology Initiative* (NIfTI) que foram convertidos previamente para imagens do tipo *Portable Network Graphics* (PNG). Na Figura 4.5, podemos observar os tipos de imagens que estão presente nesse Dataset, assim como suas respectivas máscaras.

4.4. Pré-processamento de Imagens

O pré-processamento de imagens é utilizado para a remoção de ruídos e características dos exames que não são úteis para a identificação das lesões e classificação das imagens, assim como partes que não compõe a região do pulmão, além de deixar as imagens mais semelhantes entre si. Tudo isso é utilizado para que o modelo seja capaz de não só fazer boas previsões para um único tipo de imagem presente em um único dataset, mas como fazer boas previsões para outras imagens oriundas de outros datasets que podem conter exames de diferentes máquinas de Tomografia Computadorizada.

Como podemos observar nas Figuras 4.4 e 4.5, os datasets possuem imagens de exames da TC de tórax diferentes, que podem ser, devido ao método de aquisição dessas imagens. Essas imagens passam por uma etapa de pré-processamento que pode ser observada na Figura 4.6, onde são utilizadas diversas técnicas com o propósito de melhorar as características dessas imagens. Nessa mesma Figura podemos observar as

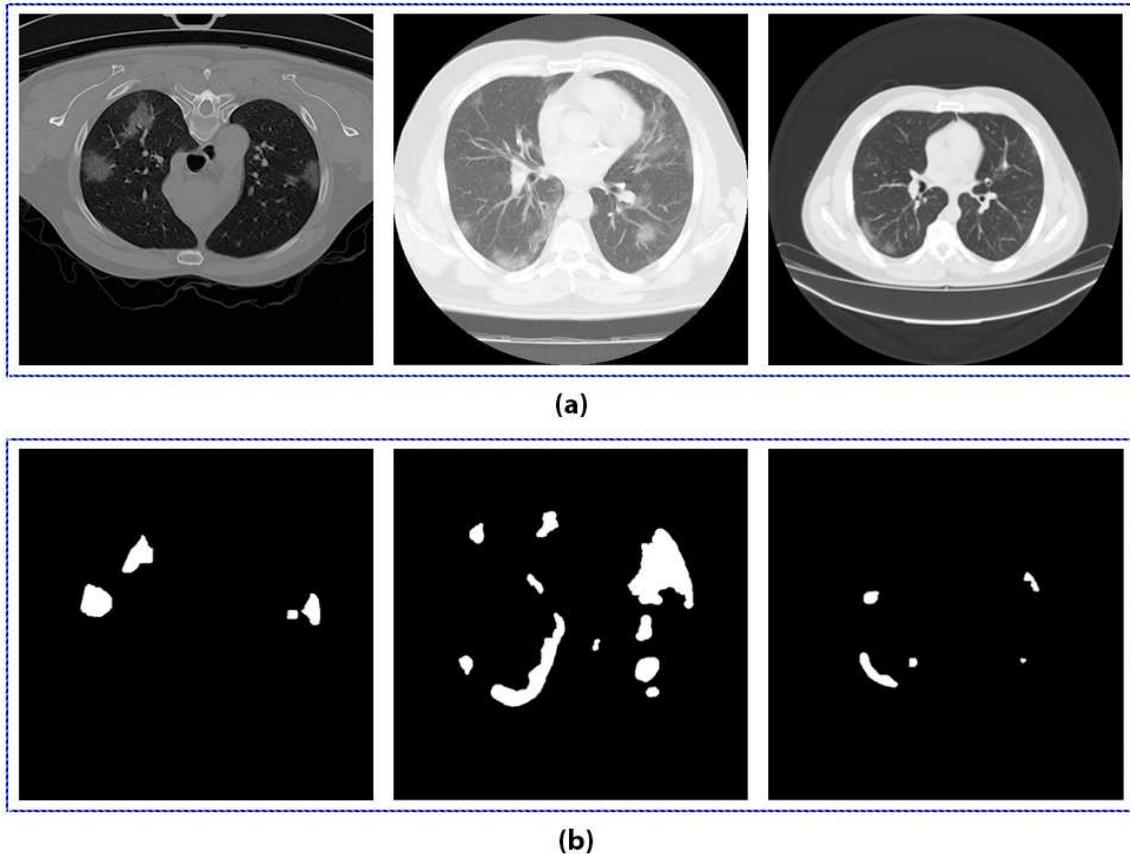


Figura 4.5. Visualização de algumas imagens do Dataset 1 com características diferentes proveniente dos métodos de aquisições diferentes, em (a) temos as imagens e em (b) suas respectivas máscaras.

técnicas utilizadas nas etapas da metodologia de pré-processamento passo a passo.

Primeiramente é utilizada uma máscara de nitidez para melhorar o contraste da imagem. Nessa etapa os detalhes nítidos são identificados usando a diferença entre a imagem original e sua versão borrada. Então os detalhes são adicionados de volta na imagem original. Utilizamos o filtro gaussiana, com *kernel* de 1×1 e sigma de 1,0. Esses parâmetros foram utilizados para que as imagens não sofram modificações bruscas. Os parâmetros foram definidos a partir de testes, onde era possível notar que valores mais altos apresentavam mudanças bruscas e valores menores não apresentava melhoras na etapa seguinte. Na Figura 4.6(b), temos a saída dessa etapa.

Após a etapa anterior é realizada uma verificação da média de intensidade dos pixels para cada imagem, traçando um limiar onde é possível analisar quais as imagens apresentam altas e baixas médias de intensidades dos pixels, com essa informação é possível traçar um tratamento diferente para imagens com alto e baixo contraste, essa etapa é encontrada na Figura 4.6(c).

Após a análise descrita na etapa anterior, as imagens que apresentam alto contraste passam por uma equalização de alongamento de contraste de histograma [Jagatheeswari et al. 2009], com os percentis 2 e 98, os valores dos percentis é definido para as imagens com alto contraste, com o objetivo de fazer leves modificações. Para as imagens com

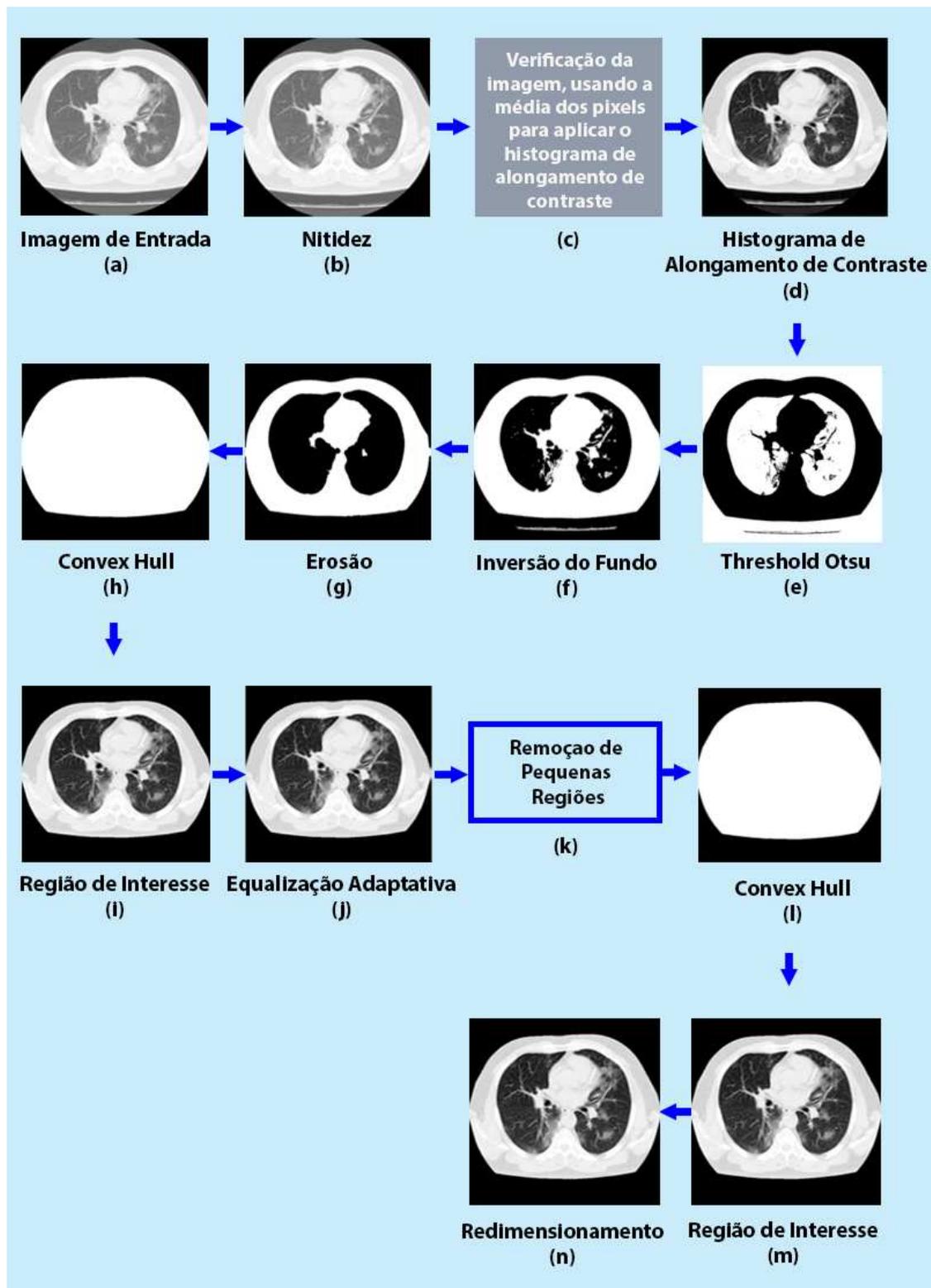


Figura 4.6. Representação da metodologia de pré-processamento e as técnicas empregadas durante as etapas que foram realizadas.

baixo contraste os valores dos percentis utilizados são, 25 e 98, esses valores são definidos para aumentar o constaste dessas imagens, sem fazer mudanças bruscas, valores maiores

podem descartar pequenas lesões presentes nos pulmões, e valores menores podem não funcionar para aumentar o contraste. O resultado dessa etapa pode ser observado na Figura 4.6(d). Esses parâmetros foram definidos para que a imagem resultante pudesse servir de entrada para a próxima etapa.

Na etapa seguinte o Threshold Otsu [Otsu 1979] é aplicado para binarizar as imagens e logo em seguida é feita uma inversão do fundo da imagem binarizada para servir de entrada na próxima etapa. As saídas dessas etapas podem ser observadas na Figura 4.6(e) e (f).

Na próxima etapa, é aplicada uma erosão com o elemento estruturante do tipo disco com tamanho 5, para removermos regiões que fazem partes do exame da TC de tórax e que estão fora da região do pulmão, esse valor é definido pois foi possível observar que valores menores não removiam essas regiões e valores maiores acabavam danificando a região dos pulmões, prejudicando a imagem. Na Figura 4.6(g) é possível ver o resultado da erosão na imagem. Então na etapa seguinte o *Convex Hull* é utilizado para incluir o conjunto de pixels no menor polígono convexo que circunda os pixels brancos, após isso obtivemos a região dos pulmões em destaque. Na Figura 4.6(h), temos o resultado do *Convex Hull*.

O resultado do *Convex Hull* é utilizado para separar a região de interesse da imagem original, onde estão presentes os pulmões, para isso é utilizado a multiplicação dos resultados da Figura 4.6(d) com a Figura 4.6(h). Na Figura 4.6(i) é possível visualizar o resultado da multiplicação, onde temos a região de interesse em destaque.

Então, utilizamos a Equalização Adaptativa, para melhorar as imagens que ainda possui baixo contraste, pois essa técnica distribui os valores de intensidade que são mais frequentes, melhorando assim o seu contraste. O valor utilizado no *clip limit* é 0,003 esse valor foi utilizado para que tenhamos uma imagem com poucas mudanças, para evitar que características sejam perdidas. Na Figura 4.6(j), temos a imagem resultante após essa técnica.

Após todas as etapas anteriores, utilizamos uma nova verificação, o *Threshold Otsu* é utilizado novamente para binarizar as imagens e verificarmos se as imagens ainda possuem regiões que não fazem parte da área dos pulmões, então se a imagem apresentar mais de uma região, utilizamos a etapa da Figura 4.6(k), que consiste na utilização do *Threshold Otsu* para removermos as regiões com menos de 10,000 pixels. Esse valor foi escolhido pois valores maiores podem remover áreas da região dos pulmões e valores menores podem não remover as pequenas regiões fora dos pulmões. Caso isso não remova as regiões, então é utilizado a erosão com o elemento estruturante do tipo disco com tamanho 5. Se na verificação a imagem não apresentar pequenas regiões então esta etapa é pulada. Todo esse processo representa a etapa mostrada na Figura 4.6(k).

Após a verificação, o *Convex Hull* é utilizado novamente na imagem retornada, para incluir os pixels no polígono convexo, e a multiplicação é realizada novamente. Os resultados dessas etapas podem ser observados na Figura 4.6(l) e (m). Por fim, temos a etapa do redimensionamento que é utilizado nas imagens para que todas possuam as mesmas dimensões de 512×512 pixels. Na Figura 4.6(m) temos o resultado do redimensionamento e o final da metodologia de pré-processamento.

Após o pré-processamento temos então imagens padronizadas, contendo apenas a região de interesse para a tarefa proposta, o que pode impedir que o modelo reconheça padrões que não são do nosso interesse, e acabe enviesando os modelos. As imagens após o pré-processamento possuem apenas a região de interesse em destaque. Podemos ver os resultados do pré-processamento nos datasets na Figura 4.7.

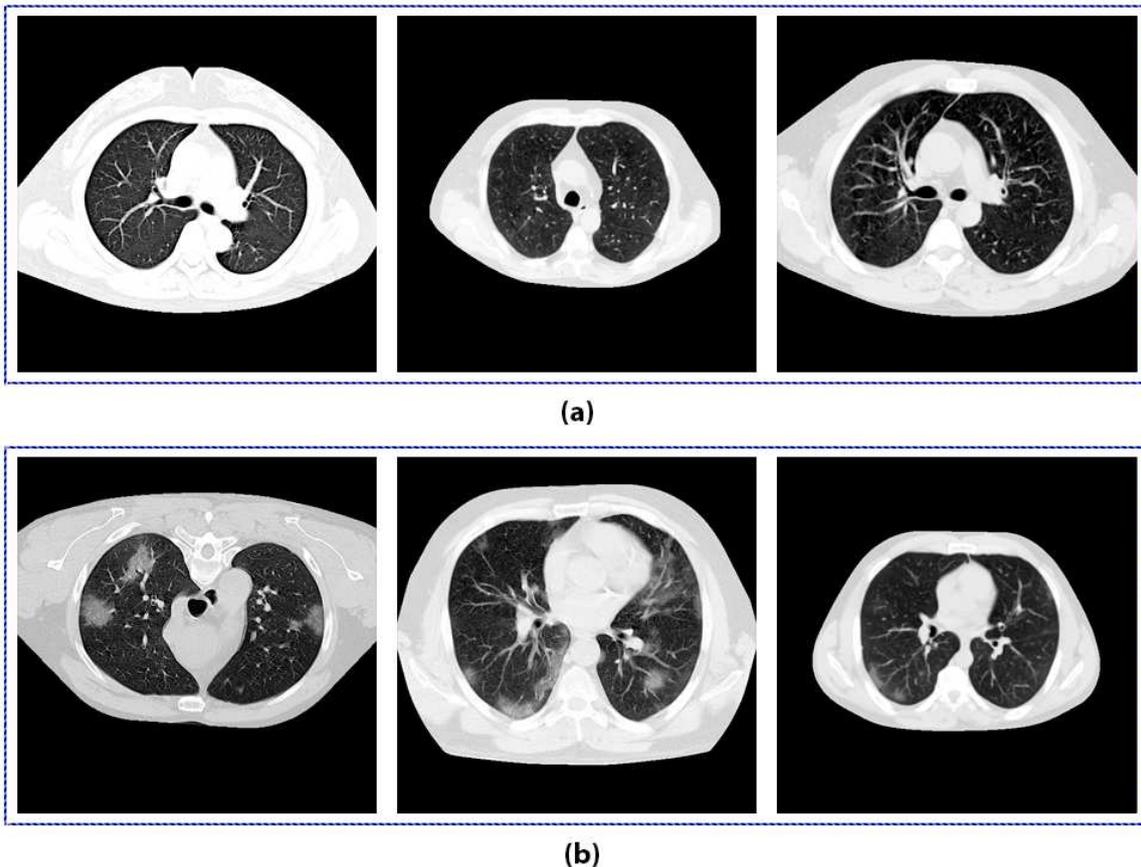


Figura 4.7. Resultado da metodologia de pré-processamento. Em (a), temos os resultados para as imagens presentes no Dataset 2 e em (b), temos os resultados do pré-processamento para as imagens do Dataset 1.

Podemos observar que as imagens agora possuem semelhanças entre si, mesmo sendo de fontes de imagens diferentes, agora temos datasets com imagens de aparência semelhantes. Podemos notar a diferença entre as imagens originais e seus resultados, após a etapa de pré-processamento quando olhamos para as Figuras 4.4, 4.5 e então comparamos com as novas imagens geradas após o pré-processamento na Figura 4.6.

4.5. Inteligência Artificial

A Inteligência Artificial (IA), está cada vez mais presente em nossos ambientes pessoais e de trabalho. As aplicações de IA usadas todos os dias incluem assistentes pessoais alimentados por voz, algoritmos comportamentais aplicados a conversas telefônicas em tempo real, recomendações de compras alimentadas por análises preditivas e veículos autônomos [Tang et al. 2018]. A IA envolve várias tecnologias, como o aprendizado de máquina, aprendizado profundo, entre outros que conseguem simular a capacidades que

os seres humanos possuem.

Com isso, o aprendizado de máquina é uma subárea da IA capaz de analisar dados, identificando seus padrões e conseguindo tomar decisões com o mínimo de intervenção humana. Dentro dessa área é possível encontrar um vasto número de técnicas, algoritmos e metodologias.

Por fim, temos o aprendizado profundo, sub área do aprendizado de máquina, responsável por algoritmos capazes de realizar tarefas complexas, assim como os seres humanos, uma dessas tarefas é a identificação de imagens [Ponti and da Costa 2017]. O aprendizado profundo permite também que os modelos computacionais compostos por múltiplas camadas de processamento aprendam representações de dados com múltiplos níveis de abstração e é evidenciado que esses métodos melhoraram drasticamente o estado da arte no reconhecimento da fala, reconhecimento visual de objetos, e muitos outros domínios, como descoberta de remédios [LeCun et al. 2015]. Na Figura 4.8 podemos observar os domínios da inteligência artificial e suas subáreas.



Figura 4.8. Representação do domínio da inteligência artificial, suas subáreas e suas características principais.

4.5.1. Classificação

Para a tarefa de classificação em imagens, a literatura mostra que o uso do aprendizado profundo está sempre presente, com isso as redes neurais convolucionais (CNN) são utilizadas. Esse tipo de rede possui uma arquitetura profunda e hierárquica.

As CNNs são responsáveis por extrair as características das imagens usando suas camadas de convoluções e representá-las como informações, essas camadas compostas por convoluções são formadas a partir de um conjunto de filtros, que durante o processo de treinamento os valores desses filtros são ajustados para que sejam ativados na presença de características importantes. Nessa etapa é realizada a convolução entre os filtros convolucionais e o volume de entrada, em seguida os valores resultantes passam por uma função de ativação. Na Figura 4.9, temos um exemplo de convolução.

As informações que foram extraídas pelas camadas convolucionais passam pela

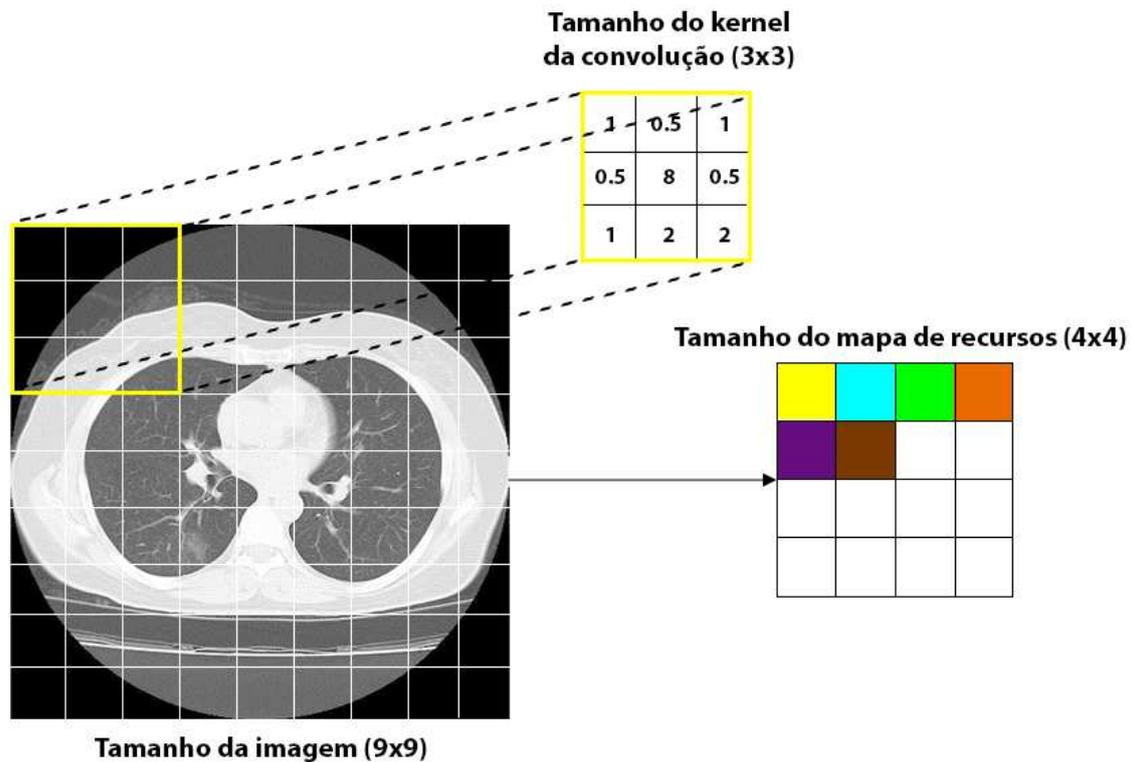


Figura 4.9. Exemplo de convolução usando um filtro com *kernel* 3×3 , que passa por toda a extensão da imagem, e por fim os resultados geram o mapa de recursos.

camada *Pooling*, que é responsável por reduzir a dimensionalidade do atributo envolvido. Com isso, é possível diminuir o custo computacional necessário para processar os dados. Além disso, é útil para extrair características dominantes das imagens. Por fim, a saída das camadas convolucionais e de *Pooling* representam as características extraídas das imagens de entradas, as camadas totalmente conectadas usam essas características para classificar as imagens em uma classe pré-determinada. Na Figura 4.10, podemos observar como é feito o *Pooling* e seu tipo, já na Figura 4.11, podemos observar as etapas realizadas por uma rede neural convolucional.

Após entendermos como funciona a estrutura de uma rede neural convolucional, temos então as etapas de treinamento de um modelo, na primeira etapa a rede neural convolucional começa com todos os filtros e pesos inicializados de forma aleatória, na segunda etapa a rede recebe as imagens de treinamento como entrada e realiza o processo de propagação, com isso são obtidos valores de probabilidade da imagem pertencer a uma determinada classe. Na terceira etapa, o valor do erro total obtido na camada de saída é calculado, na quarta etapa a *backpropagation* é utilizado para calcular os valores do gradiente do erro, em seguida os valores dos filtros e pesos são atualizados e ajustados na proporção que eles contribuíram para o erro total. Na quinta etapa, ocorre a repetição da segunda e quarta etapa, para todas as imagens do conjunto de treinamento. Com isso, o erro obtido pela rede é menor a cada vez que uma mesma imagem passa pela rede, demonstrando então que a rede está aprendendo a classificar corretamente as imagens de treinamento.

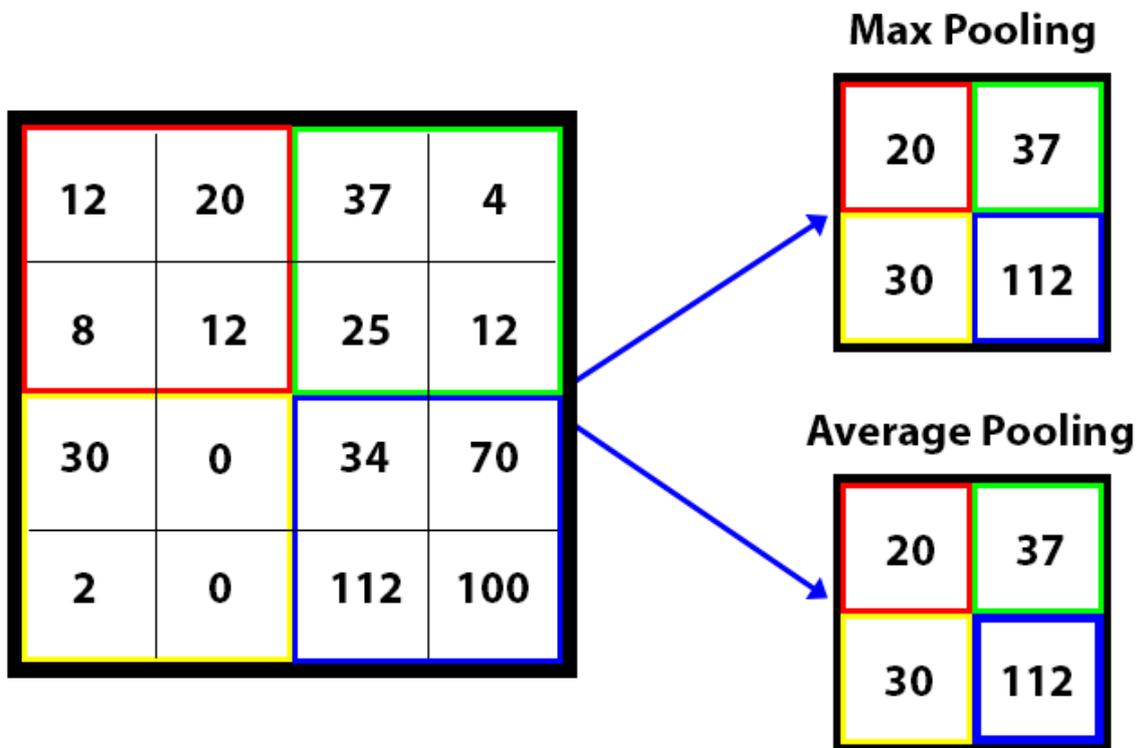


Figura 4.10. Representação da etapa de *Pooling*, nela é possível observar o *Max Pooling* e *Average Pooling*, ambos são um tipo de redução de dimensionalidade, porém suas utilidades são diferentes. O *Max Pooling* mantém as características mais proeminentes do mapa de características, enquanto o *Average Pooling* mantém os valores médios de características do mapa de recursos.

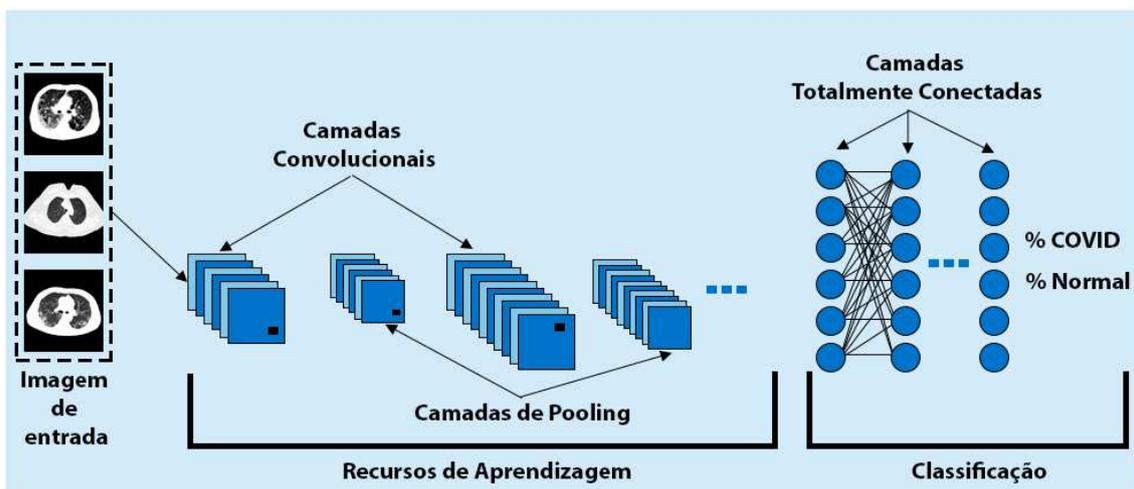


Figura 4.11. Representação da estrutura de uma rede neural convolucional. Temos a etapa de extração dos recursos de aprendizagem, definida pelas camadas de convoluções e *pooling* e temos a etapa de classificação, definida pelas camadas totalmente conectadas.

Para a tarefa de classificação duas redes foram escolhidas, sendo elas a LeNet-5 e a VGG-19, esses modelos foram selecionados após revisões [Marques et al. 2021]

e [Marques et al. 2022], que mostram que mesmo possuindo arquiteturas mais simples possuem bons resultados. A LeNet-5 foi uma das primeiras redes neurais convolucionais e teve um grande papel para a propagação do aprendizado profundo. Essa arquitetura foi utilizada para reconhecer caracteres escritos à mão e impressos por máquinas. Na Figura 4.12, podemos observar sua arquitetura original, proposta pelos autores em seu trabalho.

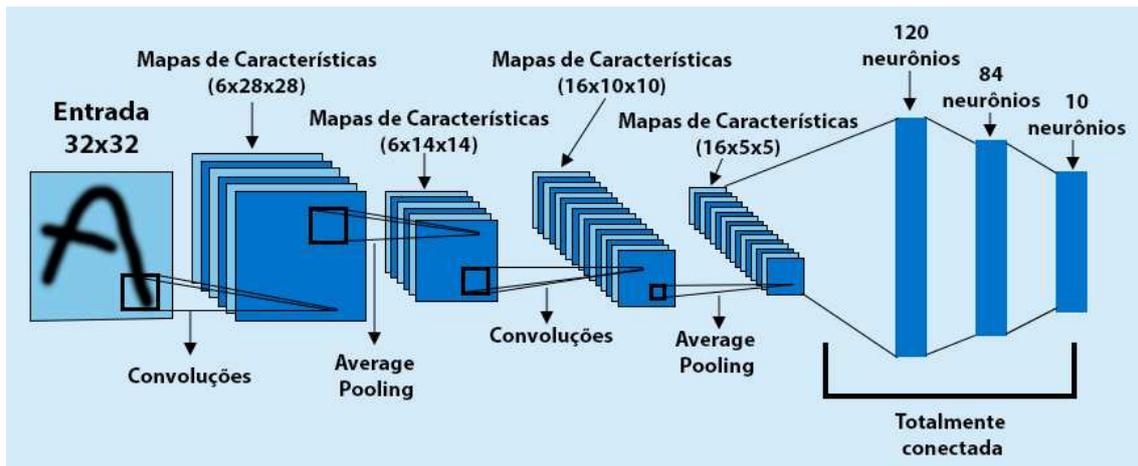


Figura 4.12. Representação da arquitetura da LeNet-5, podemos observar suas etapas de convoluções, *Pooling* e totalmente conectadas, assim como seus valores originais propostos pelos autores. A imagem foi adaptada do trabalho de [Lecun et al. 1998].

Para o primeiro modelo implementado, é utilizado a estrutura da LeNet-5 como base, camadas convolucionais e totalmente conectadas são adicionadas e modificadas, assim como seus hiper-parâmetros, ao fim temos um modelo baseado na LeNet-5 com três blocos, o primeiro bloco consiste em duas camadas convolucionais com 8 filtros de características cada e uma camada de *MaxPooling*, o segundo bloco possui 4 camadas convolucionais com 16 filtros de características, seguida de *MaxPooling* e por fim o terceiro bloco possui 4 camadas convolucionais com 32 filtros de características seguida de *MaxPooling*. O tamanho de entrada dessa arquitetura são imagens de 256×256 . Na Figura 4.13, podemos observar a ilustração dessa arquitetura.

Nessa arquitetura o tamanho do *kernel* de convolução e *pooling* é de 3×3 , com isso a arquitetura será capaz de trabalhar na detecção de pequenas lesões, que é uma situação comum para a COVID-19. Após a etapa de convolução e *pooling*, uma etapa de *flatten* é utilizada para transformar o mapa de características em um vetor que servirá de entrada para as camadas densas. Após o *flatten* uma cama de *dropout* com valor de 0,4 é utilizada para evitar que o modelo seja sensível de mais aos dados de treinamento, evitando problemas como o *overfitting*, então temos 3 camadas densa com 32, 16 e 8 neurônios, respectivamente, por fim temos a ultima camada densa que serve como a saída do modelo, onde será gerada a probabilidade de uma imagem pertencer a uma classe. Na Figura 4.13, podemos observar que essa camada possui 2 neurônios, porém para a classificação multi-classes a ultima camada densa recebe 3 neurônios um para cada classe. Em todas as camadas é utilizada a função de ativação *ReLU* e na camada de saída é utilizada a *softmax*.

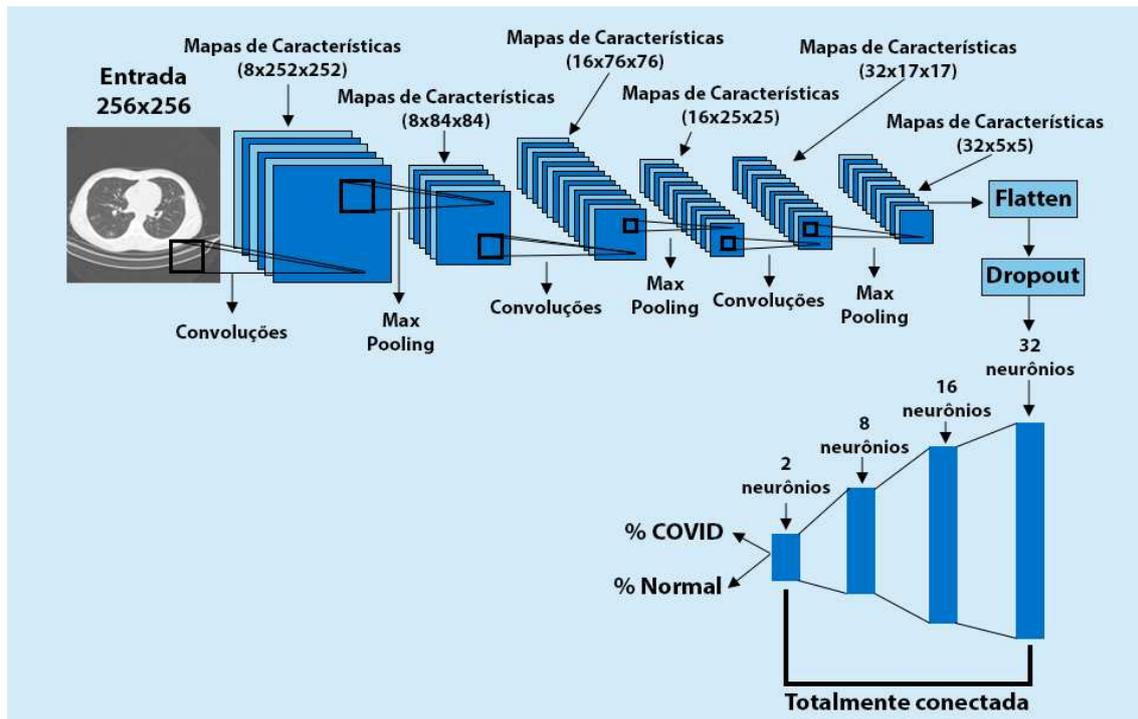


Figura 4.13. Representação da arquitetura baseada na LeNet-5, utilizada nesse trabalho.

Para a compilação do modelo é utilizado o otimizador Adam com uma taxa de aprendizagem com o valor de 0,00001. Com a função de perda *binary crossentropy*, para o treinamento em múltiplas classes é utilizado a função de perda *categorical crossentropy*, para avaliar o modelo durante o treinamento em ambas as situações a métrica acurácia é utilizada.

Para o treinamento é utilizado um *batch size* de 8, esse valor é utilizado para que o modelo tenha uma boa performance sem ficar suscetível a ruídos com *batch* menores. O treinamento é realizado por 100 épocas.

Durante o treinamento é utilizados alguns *callbacks* para executar ações durante o treinamento do modelo. A primeira estratégia utilizada é a *Early Stopping*, onde o treinamento é monitorado e caso o valor da perda em validação não caia durante 15 épocas, então o treinamento é encerrado. A redução da taxa de aprendizado durante o treinamento também é utilizada para retirar o modelo de um platô. Durante o treinamento a perda em validação é monitorada, caso o modelo durante 7 épocas apresente um platô, a taxa de aprendizado é reduzida em 0,000001 automaticamente. Por fim também é utilizado um ponto de verificação do modelo, para salvar o melhor modelo com base na menor perda em validação. No fim, denominamos o modelo proposto com o nome de LeTEN.

Outro modelo para classificação utilizado é a VGG-19, esse modelo é escolhido após análises da literatura, onde muitos trabalhos atingem resultados promissores com essa rede, além da VGG-19 possuir uma estrutura mais simples, de fácil compreensão. A VGG-19 possui 19 camadas treináveis, divididas em cinco blocos e três camadas

totalmente conectadas. Existe uma camada *MaxPooling* entre dois blocos convolucionais, duas camadas totalmente conectadas com 4096 unidades e uma camada de saída com a função de ativação *softmax*. A rede VGG-19 possui 143 milhões de parâmetros. Na Figura 4.14, podemos observar a arquitetura da VGG-19.

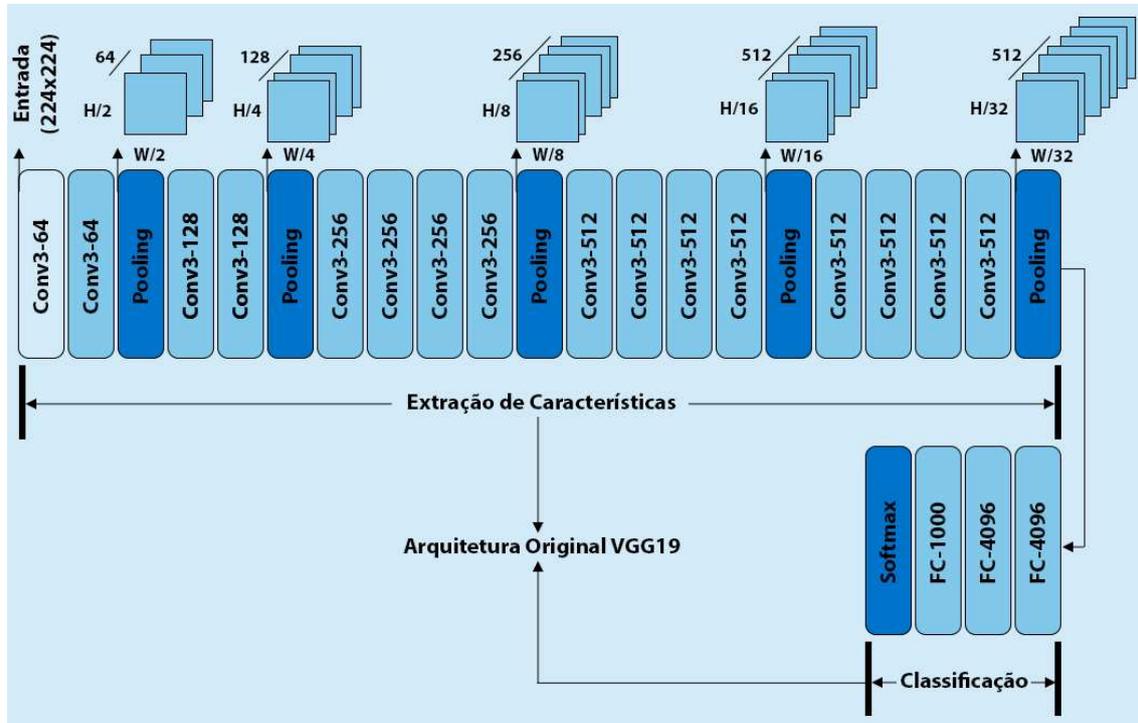


Figura 4.14. Representação da arquitetura da VGG-19, podemos observar suas camadas convolucionais, filtros, *Pooling* e as camadas totalmente conectadas, assim como seus valores originais dos parâmetros. Imagem adaptada do trabalho de [Simonyan and Zisserman 2014].

Para a VGG-19 utilizamos a técnica de transferência de aprendizado, com isso a rede utiliza os pesos da rede já treinada para uma outra base de imagem, nesse caso a ImageNet [Deng et al. 2009] é utilizada, essa base possui mais de 1 milhão de imagens com 1000 classes, com isso os pesos podem ser utilizados para iniciar e treinar a rede novamente para as classes do problema em questão. Com isso, podemos resolver um problema bastante comum que é a falta de uma grande quantidade de imagens e também é possível diminuir o tempo de treinamento que podem exigir *hardwares* mais potentes. Essa técnica é bastante utilizada quando se trabalha com redes com bastantes camadas.

Para a VGG-19 implementada nesse capítulo, a arquitetura é carregada sem as camadas densas. A entrada da arquitetura também é modificada para que seja possível trabalhar com imagens de dimensões (256×256) , então os pesos da rede que foi treinada com a base de dados da ImageNet são utilizados nas camadas convolucionais para iniciar o treinamento. A técnica de transferência de aprendizagem é utilizada nesse trabalho para evitar uma necessidade de treinamentos com longas durações necessários para treinar um modelo do zero.

Após a configuração das camadas convolucionais, uma camada *global average pooling2D* é adicionada ao final da etapa de extração de características. Na Figura 4.15,

temos a representação do funcionamento dessa camada. Essa camada é responsável por calcular o valor médio de todos os valores em toda a matriz de características, esses valores serão utilizados nas camadas totalmente conectadas.

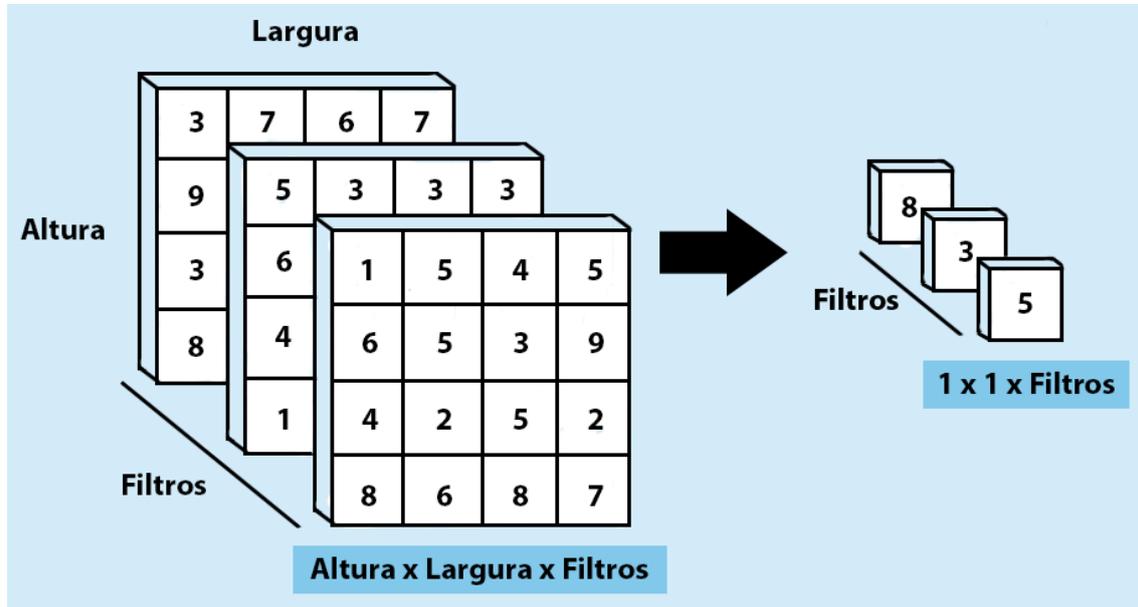


Figura 4.15. Operação *global average pooling*2D

Após essa camada é adicionado um *dropout* com 0,4, para evitar que o modelo se adapte de mais aos dados de treinamento Então novas camadas totalmente conectadas são adicionadas ao modelo. Como essa arquitetura gera muitas informações, uma camada com apenas 8 neurônios são adicionadas na primeira camada densa, seguida pela camada densa de saída com 2 neurônios, para classificação com mais classes, a camada densa de saída recebe um neurônio para cada classe. Na camada densa é utilizada a função de ativação *ReLU* e na camada densa de saída a *softmax* é utilizada. Essa metodologia é utilizada para que o modelo seja adaptado para o problema e as classes trabalhadas nesse capítulo.

Para a compilação do modelo o otimizador Adam é utilizado com uma taxa de aprendizado de 0,0000001, essa taxa é utilizada pelo motivo de o modelo já possuir pesos treinados, e com essa taxa de aprendizado, garantimos que o modelo irá aprender lentamente as características do problema proposto, com isso a dificuldade do modelo de se adaptar de mais aos dados de treinamento do que as características em geral é aumentado. Para verificar o desempenho durante o treinamento a métrica acurácia é utilizada.

Para o treinamento é utilizado um *batch size* de 8. O treinamento é realizado por 100 épocas. Durante esse treinamento também é utilizada *callbacks*, para realizar ações durante o treinamento do modelo. A *Early Stopping*, é utilizada da mesma forma que no treinamento do modelo LeTEN proposta nesse trabalho, assim como o salvamento do modelo, utilizando um ponto de verificação durante o treinamento com base na menor perda em validação, assim podemos salvar sempre o melhor modelo. A redução da taxa de aprendizado durante o treinamento também é utilizada, porém aqui a taxa de aprendizado

é reduzida em 0,00000001. O modelo apresenta muitos parâmetros e devemos tomar cuidados para que o modelo aprenda as características do problema em questão, além de ser capaz de generalizar seu aprendizado para imagens que ainda não foram vistas por esse modelo. Na Figura 4.16, temos a arquitetura VGG-19 modificada utilizada nesse capítulo.

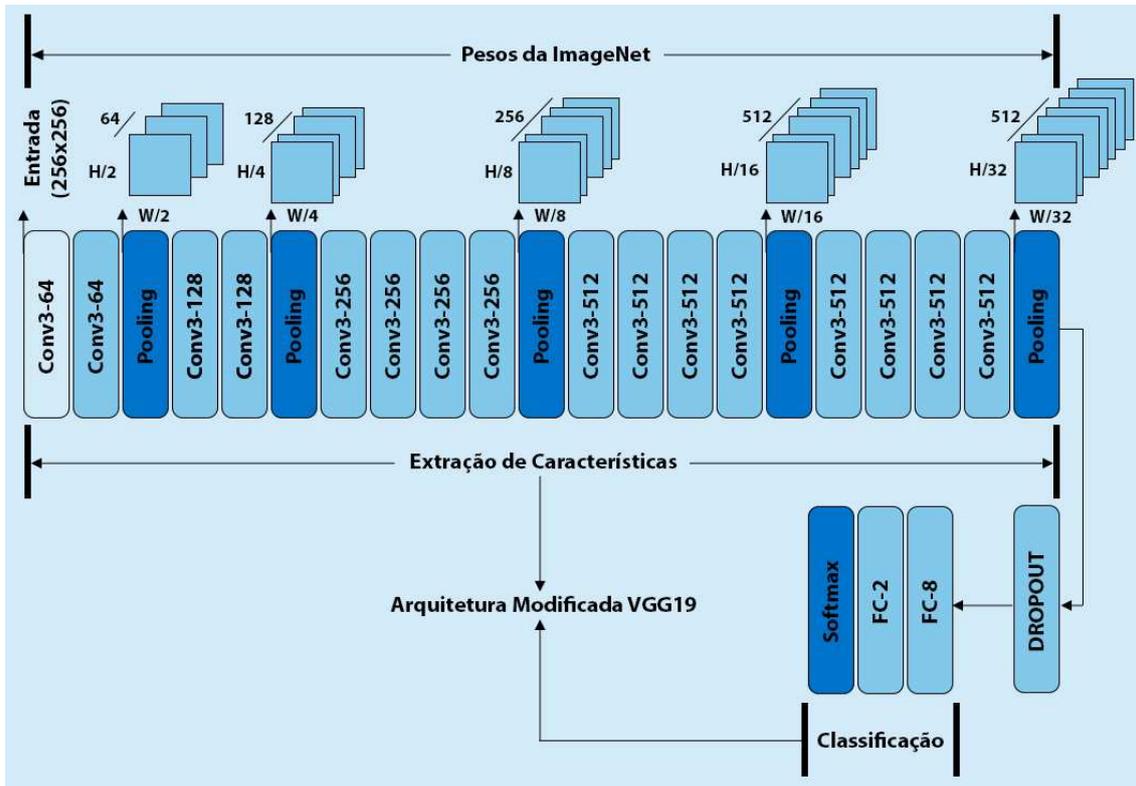


Figura 4.16. Representação do modelo VGG-19 modificada.

4.5.2. Segmentação de Lesões

Com a implementação da tarefa de classificação podemos obter bons resultados, porém é necessário considerar as necessidades que um especialista possui. Para o especialista é necessário a identificação das regiões de lesões e não apenas saber se um paciente possui ou não essa doença. Pensando nisso, o método de classificação pode não ser suficiente para colaborar de uma forma que possa sanar as necessidades desses especialistas. Assim, temos os métodos de segmentação, onde é possível marcar as lesões nas imagens de Tomografia de Tórax, possibilitando ao especialista uma análise e avaliação da gravidade da infecção nos pulmões dos pacientes. Essas informações contribuem não só para o diagnóstico, mas também para o acompanhamento dos pacientes.

Com isso, temos a U-Net, uma rede com uma arquitetura de rede convolucional para segmentação de imagens [Ronneberger et al. 2015c], essa rede foi desenvolvida com um enfoque maior na segmentação de imagens biomédicas. Após analisar os trabalhos relatados na literatura, notamos uma tendência no uso de redes baseadas na U-Net. Essa técnica mostra excelentes resultados em diversos trabalhos. Na Figura 4.17, podemos observar os resultados que podem ser obtidos utilizando essa rede.

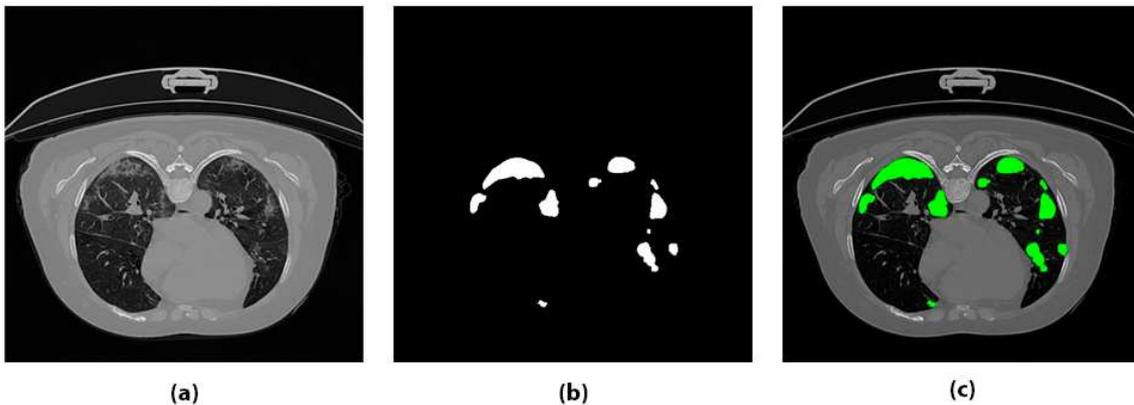


Figura 4.17. Resultados que podem ser obtidos utilizando a rede U-Net, em (a), temos a imagem que ira servir de entrada para a rede, em (b), temos a imagem de saída da rede, essa imagem de saída é a região onde a rede detectou lesões, por fim, em (c), temos uma representação da união da imagem de saída com a imagem de entrada, onde podemos observar as regiões com lesões afetadas em destaque.

A arquitetura da U-Net consiste em um caminho de contração e um caminho expansivo. O caminho de contração da arquitetura é um típico caminho de uma rede convolucional. Este caminho é composto por uma aplicação repetida de duas convoluções 3×3 , cada uma seguida por uma função de ativação *ReLU* e uma operação de max pooling 2×2 com duas etapas para *downsampling*. Em cada etapa de *downsampling* os números de filtros de características são dobrados.

Cada passo no caminho expansivo consiste em um *upsampling* dos filtros de características seguido de uma convolução 2×2 que reduz pela metade o número dos filtros de características, depois temos uma concatenação com o mapa de características correspondentemente cortado do caminho de contração, e duas convoluções 3×3 , cada uma seguida por uma função *ReLU*.

Na camada final, é utilizada uma convolução 1×1 para mapear todos os vetores de característica para o número desejado de classes [Ronneberger et al. 2015a]. Na Figura 4.18, podemos observa a arquitetura original proposta dessa rede, e seus caminhos de contração e expansão.

A U-Net implementada nesse trabalho usa como entrada imagens com dimensões de 512×512 pixel e 32 filtros iniciais que podem ser visualizados na Figura 4.18. Para o treinamento é utilizado um *batch size* de 6, esse valor é utilizado pois é o maior valor que a gpu utilizada consegue trabalhar com as imagens nessas dimensões.

Para o treinamento é utilizada a dice loss como função de perda, ela é dada pela Equação 8, o otimizador Adam é utilizado com uma taxa de aprendizagem com valor de 0,0003 e as métricas Dice, dada pela Equação 7 e *Intersection over Union* (IoU) dada pela Equação 9 são utilizadas durante o treinamento para acompanhar a performance do modelo no conjunto de treino e validação.

Como estratégia para fazer ações durante o treinamento do modelo, os *callbacks*, são utilizados assim como na tarefa de classificação. Primeiramente a *Early Stopping*, é

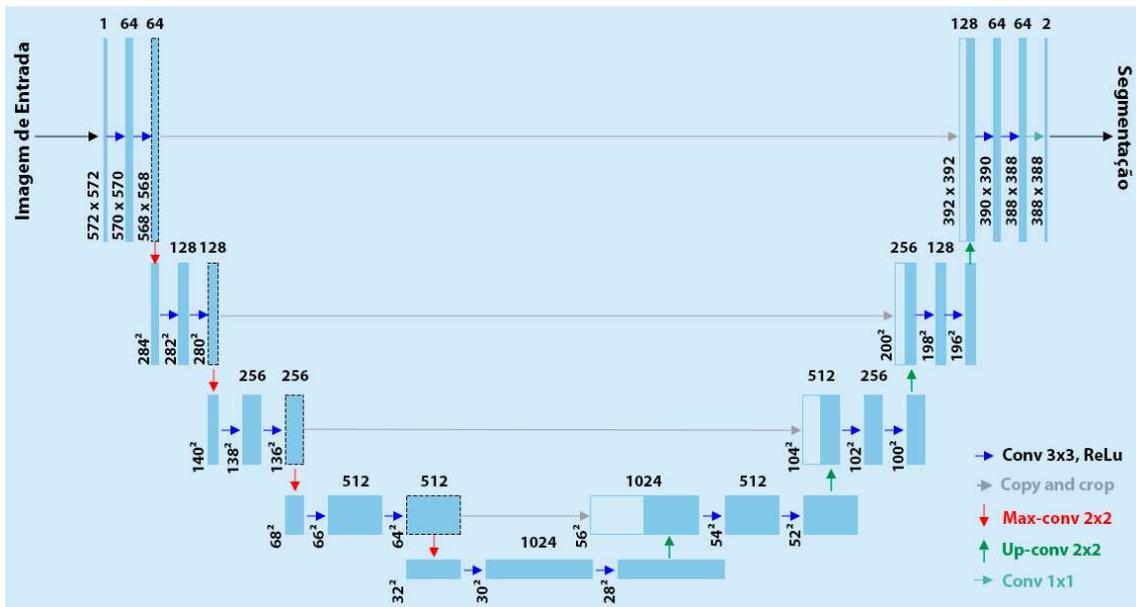


Figura 4.18. Representação da arquitetura original da U-Net, ao lado mais a esquerda temos o caminho de contração e no lado direito temos o caminho expansivo. Imagem adaptada do trabalho de [Ronneberger et al. 2015b].

utilizada. Diferente da tarefa de classificação, aqui monitoramos a perda em validação por 10 épocas, e caso essa perda não caia por 10 épocas o treinamento é encerrado. A redução da taxa de aprendizado durante o treinamento também é utilizada, nessa tarefa a taxa é reduzida em 0,00003 quando a perda em validação não cai durante 5 épocas, e por fim o modelo é salvo com base na menor perda em validação. O número de épocas do treinamento do modelo é definida em 100. Na Figura 4.19, temos a representação do modelo U-Net usado nesse trabalho.



Figura 4.19. Representação da arquitetura U-Net implementada nesse trabalho.

4.6. Avaliação dos Resultados

4.6.1. Classificação

Para avaliar a metodologia proposta, são utilizados alguns experimentos difundidos na literatura, onde o primeiro experimento proposto consiste na divisão das bases de imagens entre treinamento, validação e teste. Para a tarefa de classificação, usamos o Dataset 2, devido a esse dataset ser muito grande, e haver limitações de *hardware*, separamos amostras de forma aleatória para cada conjunto de dados presente nesse dataset, por fim para tarefa de classificação, são utilizadas 9.000 imagens para treinamento, sendo 4,500 imagens para cada classe: COVID-19, Normal. Para a validação temos 2,250 imagens, sendo 1,125 imagens para cada classe e para o teste do modelo temos um total de 2,600 imagens, 1,300 imagens para cada classe.

Para o segundo experimento, adicionamos mais uma classe, para observar como o modelo irá se comportar, então para esse experimento são utilizadas 13,500 imagens para treinamento, sendo 4,500 imagens para cada classe, COVID-19, Normal, Pneumonia. Para a validação temos 3,375 imagens, sendo 1,125 imagens para cada classe e para o teste do modelo temos um total de 3,900 imagens, 1,300 imagens para cada classe. Nesse experimento o pré-processamento não é utilizado, pois o objetivo desse experimento é verificar a capacidade do modelo em classificar mais classes.

Para avaliar os resultados de cada experimento são utilizadas algumas métricas que são amplamente difundidas nos trabalhos do estado da arte, onde estão presente em quase todos os trabalhos pesquisados, temos então as seguintes métricas: Acurácia; Kappa; Precisão; Especificidade; Sensibilidade, F1-Score e Auc.

Para calcular essas métricas é necessário utilizar uma matriz de confusão, com isso é possível analisar as previsões feitas pelo modelo, e com essa matriz, podemos extrair as informações como:

Verdadeiros Positivos (VP), quando o modelo prever que aquela imagem é de uma determinada classe e essa imagem realmente é dessa classe.

Verdadeiros Negativos (VN), quando o modelo prever que a imagem é classificada como não pertencente a uma classe e ela não é dessa classe.

Falso Positivos (FP), quando o modelo classifica uma imagem não pertencente a uma determinada classe mas ela pertence a essa classe.

Falso Negativo (FN), quando o modelo classifica uma imagem pertencente a uma classe, mas ela não é dessa classe. Então podemos calcular as demais métricas, abaixo podemos visualizar sua equação e seu conceito.

Na Figura 4.20, podemos observar o exemplo de uma matriz de confusão. Com as informações de VP, VN, FP e FN, então podemos calcular as métricas abaixo:

Acurácia: é a média global dos acertos para o modelo ao classificar as classes. A Acurácia se dá pela Equação 1.

$$Acc = \frac{(VN + VP)}{(VP + FN + VN + FP)} \quad (1)$$

		Classe Detectada	
		Sim	Não
Classe Real	Sim	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Figura 4.20. Representação da matriz de confusão.

Kappa: a acurácia pode não ser uma boa métrica de avaliação para dados que possam conter classes com quantidade de amostras desbalanceados, então para resolver isso podemos utilizar a métrica Kappa, dada pela Equação 2.

$$Kappa = \frac{2(VP \times VN - FN \times FP)}{(VP + FP) \times (FP + VN) + (VP + FN) \times (FN + VN)}. \quad (2)$$

Precisão: essa métrica verifica quantas observações o modelo classificou de forma correta como pertencente a uma classe. Ou seja, de todas as imagens que o modelo previu como COVID-19, quantas realmente possuem COVID-19. A Precisão se da pela Equação 3.

$$Prec = \frac{VP}{(VP + FP)}. \quad (3)$$

F1-Score: a métrica F1-Score é a média harmônica entre as métricas, Recall e Precisão, ou seja, ela resume as informações dessas duas métricas. A formula dessa métrica é dada pela Equação 4.

$$F1 = 2 \times \frac{Prec \times Sens}{Prec + Sens}. \quad (4)$$

Especificidade: a especificidade avalia a capacidade do modelo de classificar resultados negativos. Podemos calculá-la usando a equação 5.

$$Espec = \frac{VN}{(FP + VN)}. \quad (5)$$

Sensibilidade: essa métrica avalia a capacidade do modelo de classificar com sucesso resultados classificados como positivos. Ela pode ser obtida pela equação 6.

$$Sens = \frac{VP}{VP + FN}. \quad (6)$$

Auc: a curva ROC pode ser avaliada pela métrica AUC, essa métrica indica a probabilidade de duas previsões serem corretamente ranqueadas. A AUC será um valor entre 0 e 1. Quanto maior esse valor, melhor a capacidade do modelo em separar classes.

4.6.2. Segmentação

Para avaliarmos o desempenho do modelo para a tarefa de segmentação, utilizamos o Dataset 1, fazendo uma divisão aleatória das imagens, onde é separado 20% das imagens desse dataset para o teste desse modelo, então as imagens restantes são divididas em 80% para Treinamento e 20% para validação, as imagens que estão presente em um desses conjuntos não se repetem em nenhum outro conjunto. Após essa divisão temos então, 2,252 imagens de treinamento, 564 imagens de validação e 704 imagens para teste.

Para avaliar o desempenho do método proposto são utilizadas algumas métricas da tarefa de classificação, como a sensibilidade e a especificidade, e temos a adição de mais duas métricas, que são a Dice e IoU. Essas métricas são utilizadas especialmente para a tarefa de segmentação. Utilizamos também a Dice Loss que é a função de perda utilizada na U-Net proposta.

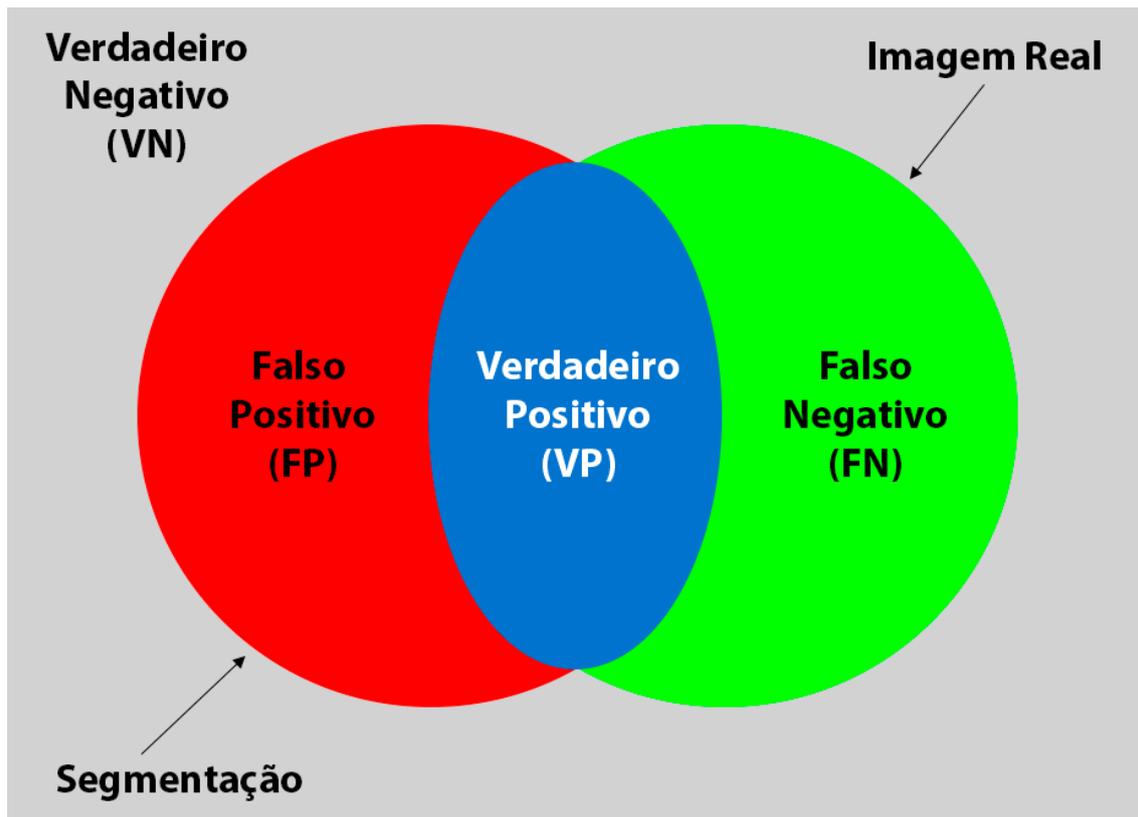


Figura 4.21. Representação da análise das informações VP, VN, FP e FN. A Imagem Real na imagem representa a máscara feita pelo especialista, enquanto a segmentação representa a máscara feita pelo modelo.

Podemos observar que assim como a tarefa de classificação também é utilizado as informações de VP, VN, FN e FP, porém a análise dessas informações na tarefa de segmentação é diferente, podemos observar como funciona na Figura 4.21. Com essas

informações podemos calcular as métricas a baixo:

Dice: é usada para quantificar o desempenho dos métodos de segmentação de imagens. O dice é uma estatística usada para medir a similaridade de duas amostras. o Dice é calculado pela Equação 7.

$$Dice = \frac{2VP}{2VP + FP + FN} \quad (7)$$

Dice Loss: utilizando o Dice para obter valores próximos a 1 para avaliar o quão a segmentação prevista é similar a segmentação verdadeira, utilizamos a função de perda Dice Loss para obter valores próximo a 0, fazendo com que o valor da perda caia quando o valor do dice subir. A tarefa de otimização dessa função de perda é essencial para obter um modelo generalista. A Dice Loss se da pela Equação 8.

$$DiceLoss = 1 - Dice. \quad (8)$$

IoU: assim como o Dice a métrica IoU também é uma métrica para quantificar o desempenho dos métodos de segmentação. A IoU mede a semelhança entre conjuntos amostrais, e é definido como o tamanho da intersecção dividida pelo tamanho da união dos conjuntos amostrais. A Equação 9 é dada por:

$$IoU = \frac{VP}{VP + FP + FN} \quad (9)$$

4.7. Resultados

Para a implementação das tarefas realizadas nesse capítulo, são utilizadas as seguintes bibliotecas: numpy [Harris et al. 2020] e matplotlib [Hunter 2007], para manipulação e visualização de dados; OpenCv [Bradski 2000] e skimage [Van der Walt et al. 2014], para manipulação de imagens; tensorflow [Abadi et al. 2015] e keras [Gulli and Pal 2017], para trabalhar com aprendizado profundo; e sklearn [Pedregosa et al. 2011] para cálculo de métricas e outras tarefas. Para acelerar o treinamento é utilizada a GPU RTX 2080 ti de 11GB.

4.7.1. Classificação

O Experimento 1, que consistiu na utilização do modelo proposto, LeTEN e a VGG-19 com transferência de aprendizado, apresentaram resultados promissores, os modelos fora utilizados para classificar as classes COVID-19 e Normal. Para realizar os testes o modelo é avaliado utilizando as métricas definidas anteriormente: Acurácia (Acc); Kappa (Kap); Precisão (Prec); F1-Score (F1); Especificidade (Esp); Sensibilidade (sen) e Auc. Na Tabela 4.1, temos os resultados das métricas obtidas para o conjunto de teste. Os modelos foram avaliados com as imagens com e sem pré-processamento. Podemos observar que as métricas entre os modelos estão bem próximas, mesmo quando comparamos a LeTEN e a VGG-19.

Nas Figuras 4.22 e 4.23, podemos observar as curvas de aprendizado do modelo LeTEN sem pré-processamento e com pré-processamento, assim como suas matrizes de

Tabela 4.1. Resultados das métricas obtidas no Experimento 1 de classificação.

Modelo	Acc(%)	Kap(%)	Prec(%)	F1(%)	Esp(%)	Sen(%)	Auc(%)
LeTEN	99.00	99.00	99.00	99.00	99.00	100	99.00
LeTENP	99.00	99.00	99.00	99.00	99.00	100	99.00
VGG19	100	100	100	100	100	100	100
VGG19P	100	100	100	100	100	99.00	100

confusão, respectivamente.

Podemos observar pela matriz de confusão como as classes foram previstas pelo modelo e verificar o comportamento da classificação. O modelo apresenta um resultado promissor para a classificação de duas classes, principalmente por causa que as classes: COVID-19 e Normal, são bem diferentes entre si. Podemos notar que os resultados dos modelos com e sem pré-processamento não apresenta diferença entre si em suas métricas e pouquíssima diferença entre as matrizes de confusão.

Com as curvas de aprendizado ainda podemos analisar como foi o treinamento do modelo, e verificar se o modelo está sofrendo de *overfitting*, que é o sobre-ajuste do modelo, ou seja ele aprende o conjunto de dados de treinamento, mas não consegue generalizar para o conjunto de dados de validação, ou até mesmo não consegue fazer novas previsões para um novo conjunto de imagens. Podemos também verificar se o modelo sofre de *underfitting*, que é quando o modelo não consegue aprender.

Assim como foi descrito anteriormente, temos também as curvas de aprendizagem e matrizes de confusão para o modelo VGG-19, com e sem pré-processamento. Na Figura 4.24 e 4.25, podemos observar os gráficos e as matrizes. Podemos notar que a VGG-19 apresenta um resultado pouco melhor que a LeTEN, porém essa diferença é irrisória. Além de que a VGG-19 possui uma arquitetura mais complexa que a LeTEN, além de ser inicializada com os pesos da ImageNet.

Podemos observar também que as matrizes de confusão geradas pelo modelo VGG-19 apresenta pouquíssimos erros nas classificações das classes. O que demonstra um melhor desempenho quando comparamos com o modelo LeTEN.

No Experimento 2, que é definido por uma classificação multi-classes, com as classes: COVID-19, Normal e Pneumonia, e então os modelos são avaliados utilizando as mesmas métricas do experimento anterior. Nesse experimento podemos avaliar como o modelo se comportar quando existe uma classe semelhante. Já que a Pneumonia viral pode ser confundida com a Pneumonia causada pela COVID-19.

Na Tabela 4.2, podemos observar as métricas que os modelos foram capazes de alcançar. Nesse experimento temos a VGG-19 sendo capaz de atingir métricas superiores ao modelo LeTEN.

Tabela 4.2. Resultados das métricas obtidas no Experimento 2 de classificação.

Modelo	Acc(%)	Kap(%)	Prec(%)	F1(%)	Esp(%)	Sen(%)	Auc(%)
LeTEN	94.00	93.00	93.00	95.00	85.00	98.00	91.00
VGG19	99.00	99.00	99.00	100	98.00	100	99.00

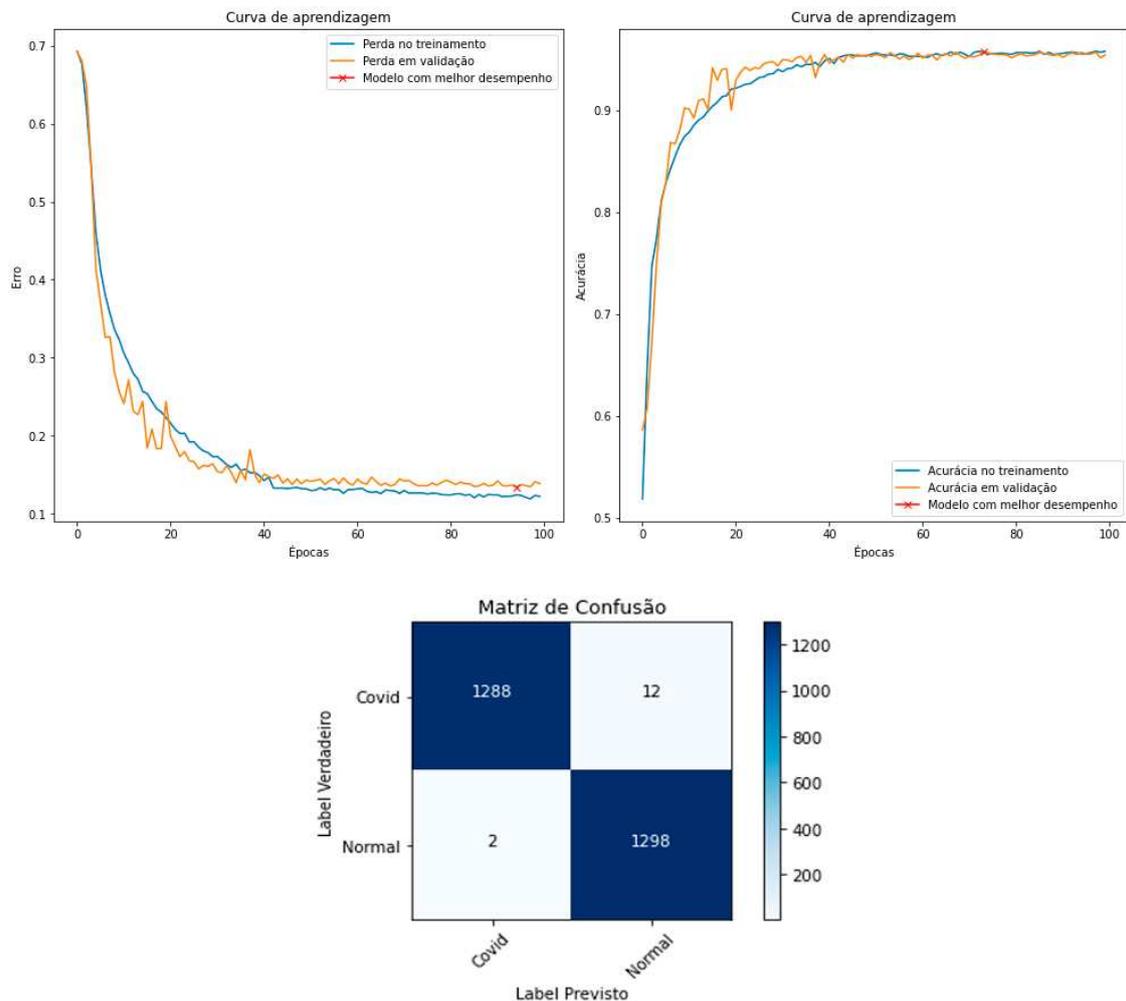


Figura 4.22. Curvas de aprendizagem para o modelo LeTEN do Experimento 1, sem pré-processamento, e sua matriz de confusão. Nas curvas de aprendizagem, observamos tanto a curva de perda como a curva da acurácia.

Assim como no experimento anterior, as curvas de aprendizagem e a matriz de confusão também são utilizadas para verificar o comportamento do modelo durante a etapa de treinamento e principalmente para verificar as classificações realizadas pelos modelos.

Na Figura 4.26, temos as curvas de aprendizagem e a matriz de confusão para o modelo LeTEN. Podemos notar que as curvas de aprendizagem mesmo não sendo tão suaves como as anteriores, podemos notar que o modelo continua aprendendo a cada época que passa. Vemos na matriz de confusão que o modelo confunde bastante quando se trata das classes COVID-19 com Pneumonia o que já era de se esperar, devido a semelhanças das imagens.

Na Figura 4.27, temos as curvas de aprendizagem e matriz de confusão para o modelo VGG-19. Podemos notar que o treinamento é mais suave que a LeTEN, podemos notar também que apesar do gráfico, o modelo continua a aprender com o passar das épocas, na matriz de confusão, temos resultados mais promissores quando comparadas

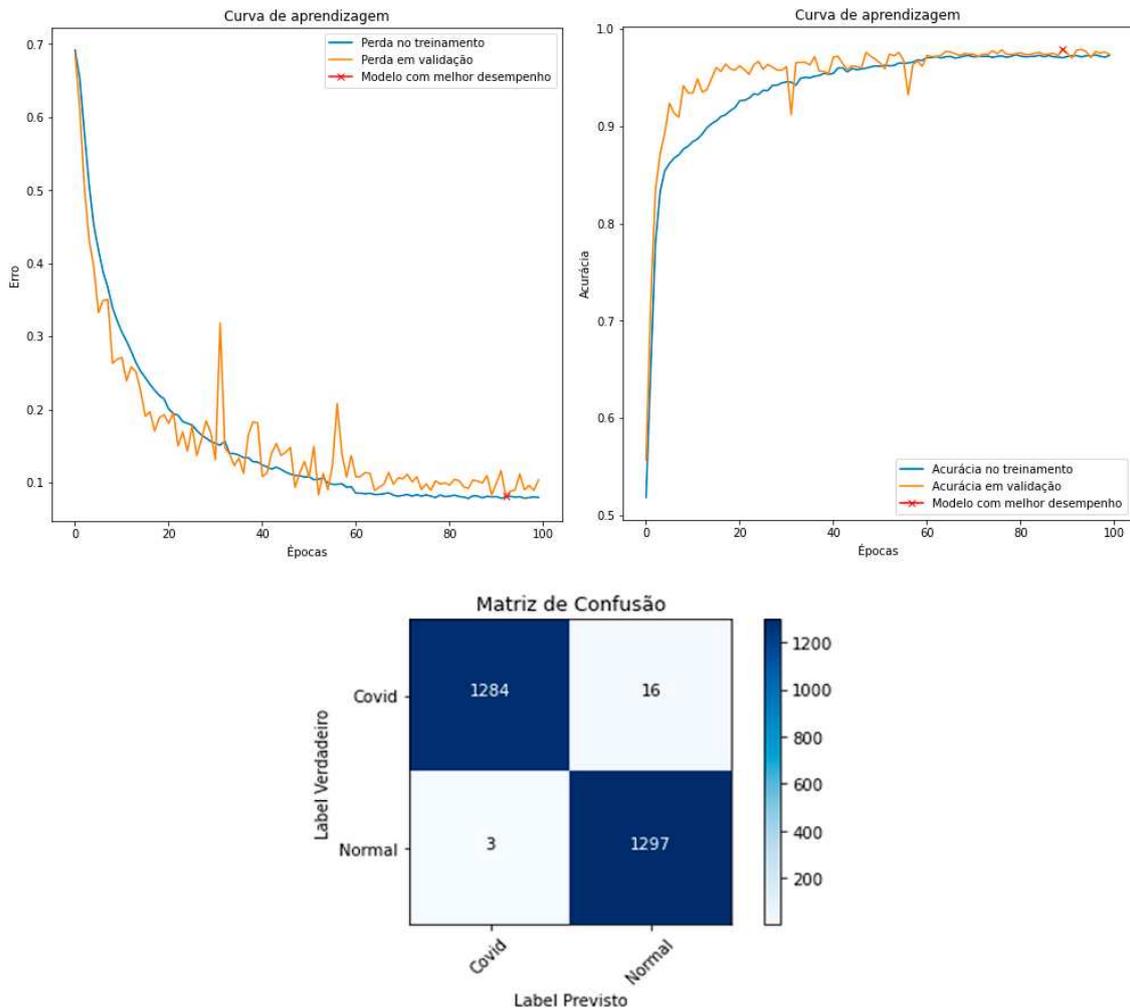


Figura 4.23. Representação das curvas de aprendizagem para o modelo LeTEN do Experimento 1, com pré-processamento, e sua matriz de confusão. Nas curvas de aprendizagem, observamos tanto a curva de perda como a curva da acurácia.

com a LeTEN. Podemos notar que o modelo não comete tantos erros na classificação da COVID-19 e Pneumonia.

4.7.2. Segmentação

Esse experimento foi baseado em outras aplicações utilizadas por autores da literatura. Para os testes o modelo é avaliado utilizando as métricas definidas anteriormente: Dice, IoU, Especificidade (Esp) e Sensibilidade (Sens). No experimento 1 que consistiu na divisão do Dataset 1 em treinamento, validação e teste, obtemos os seguintes resultados que podem ser observados na Tabela 4.3.

Na Figura 4.28 podemos observar alguns dos melhores resultados gerados pelo modelo, na Figura 4.28(a) e (b) temos as previsões feitas pelo modelo para o Dataset 1 com pré-processamento e sem, respectivamente. Assim como nas métricas podemos notar observando as imagens que as previsões são bastante semelhantes e o ganho de métricas

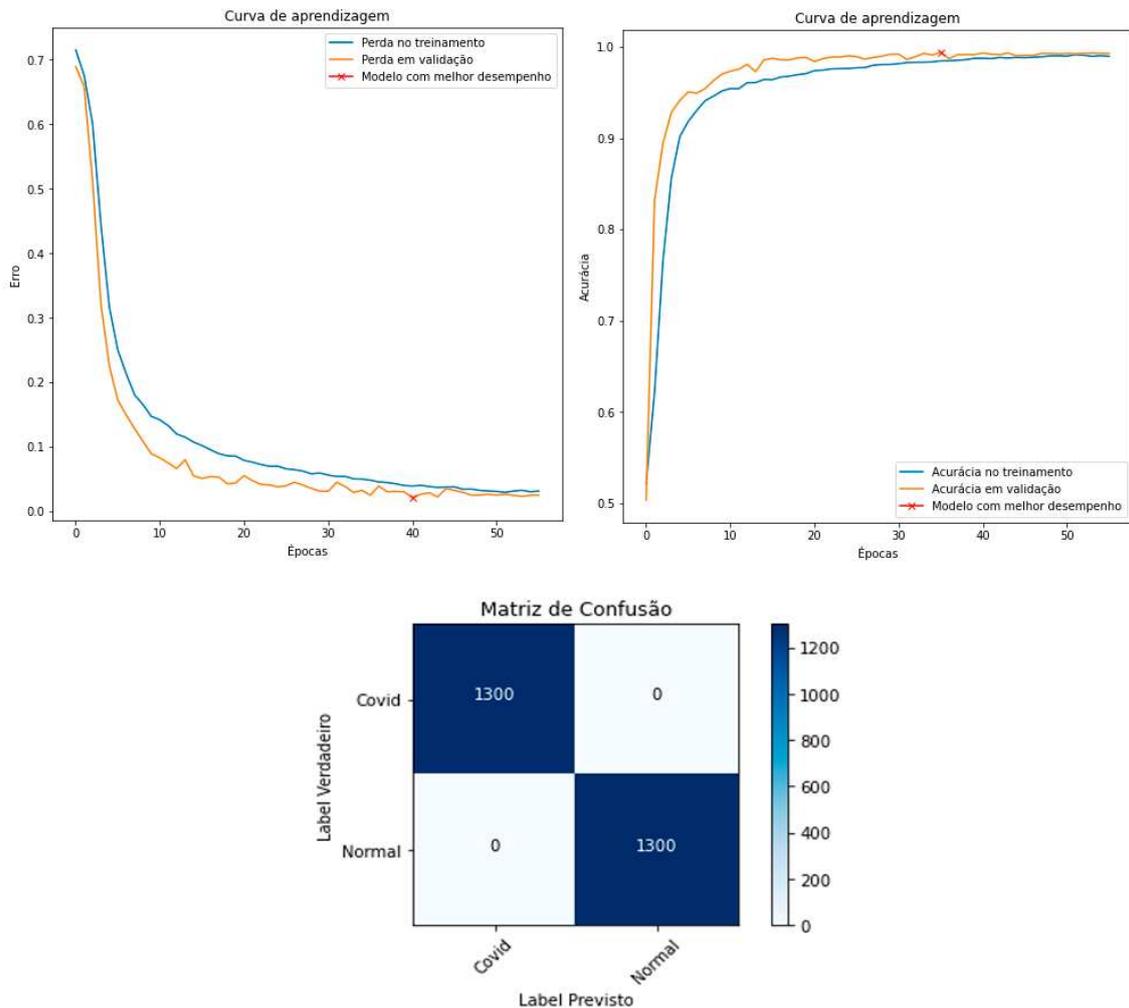


Figura 4.24. Representação das curvas de aprendizagem para o modelo VGG-19 do Experimento 1, sem pré-processamento, e sua matriz de confusão. Nas curvas de aprendizagem, observamos a curva de perda e curva da acurácia.

Tabela 4.3. Resultados das métricas obtidas no experimento 1 de segmentação. O dataset foi avaliado com e sem o método de pré-processamento.

Datasets	Dice(%)	IoU(%)	Sensibilidade(%)	Especificidade(%)
Dataset1	86.00	73.00	86.00	100
Dataset1PRE	87.00	77.00	86.00	100

mostrada na Tabela 4.3 se dá pelo fato do modelo com pré-processamento se adaptar melhor com pequenas lesões.

Podemos destacar também a especificidade que pode chegar a 100% devido há essa métrica representa o fundo das imagens, e o *Threshold* é definido por maior que 5, ou seja o modelo utilizado não possui previsões menores de 50% para o fundo.

Já na Figura 4.29, podemos observar as previsões com menores IoU. Podemos notar que os modelos sofrem para identificar lesões pequenas que ficam localizadas nas laterais dos pulmões, a diferença entre os métodos é a frequência com que isso acontece,

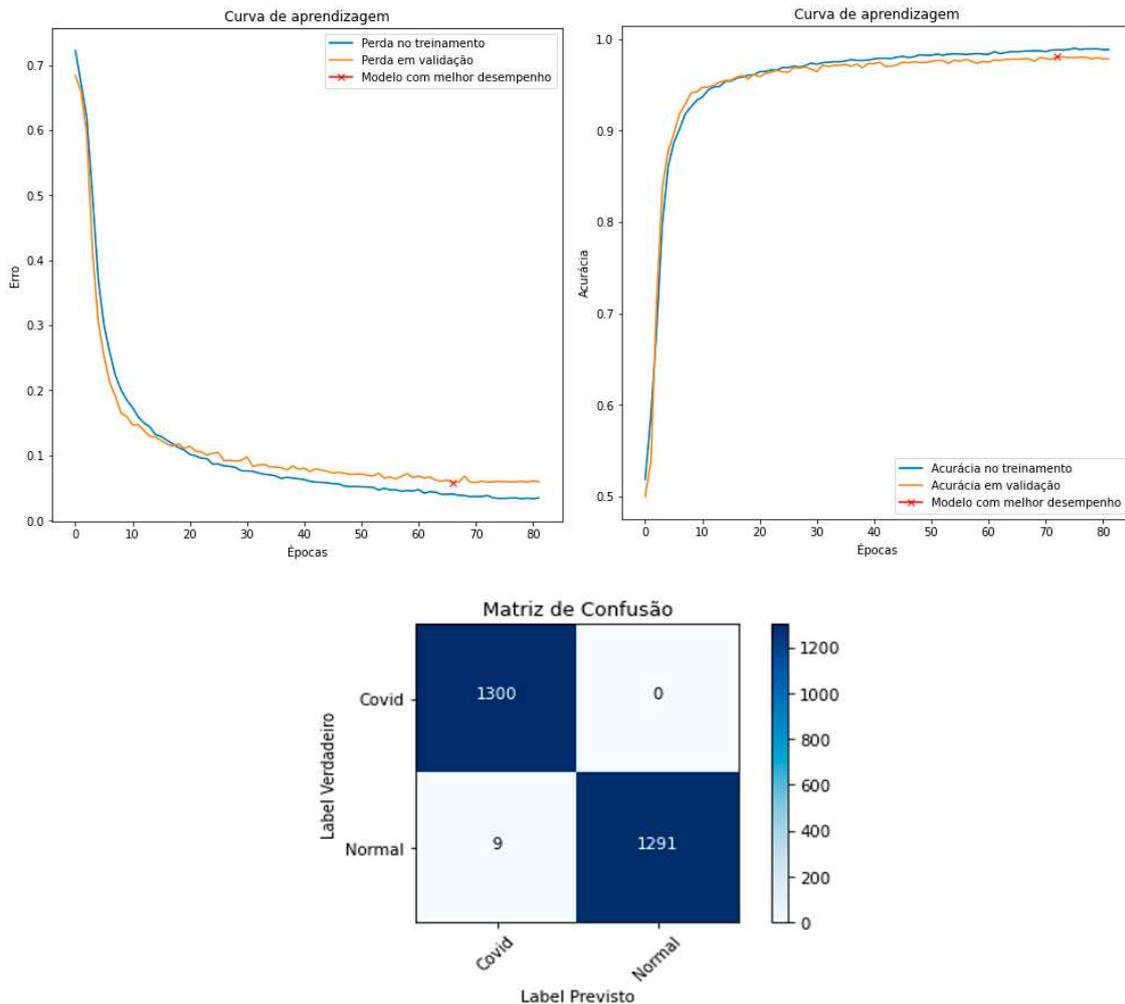


Figura 4.25. Curvas de aprendizagem para o modelo VGG-19 do Experimento 1, com pré-processamento, e sua matriz de confusão. Para as curvas de aprendizagem é observada a curva de perda e a curva da acurácia.

sendo mais comum de acontecer nos métodos sem as imagens processadas.

Na Figura 4.30, podemos observar as curvas de aprendizado dos modelos com e sem pré-processamento assim como na tarefa de classificação, porém aqui apenas a perda é monitorada.

4.8. Discussão

Neste capítulo resolvemos implementar um modelo baseado na LeNET, que chamamos de LeTEN, com o intuito de utilizar um modelo com arquitetura simples, para a classificação de COVID-19, além de testar também o desempenho de um modelo amplamente usado e difundido na literatura como a VGG-19 com transferência de aprendizagem.

Para a tarefa de classificação temos resultados promissores, podemos notar que mesmo com uma rede com poucas camadas e parâmetros como a LeTEN, foi possível classificar corretamente a maior parte das imagens, no Experimento 1, podemos ver que o modelo foi capaz de alcançar métricas de 99% em várias métricas, já no Experimento

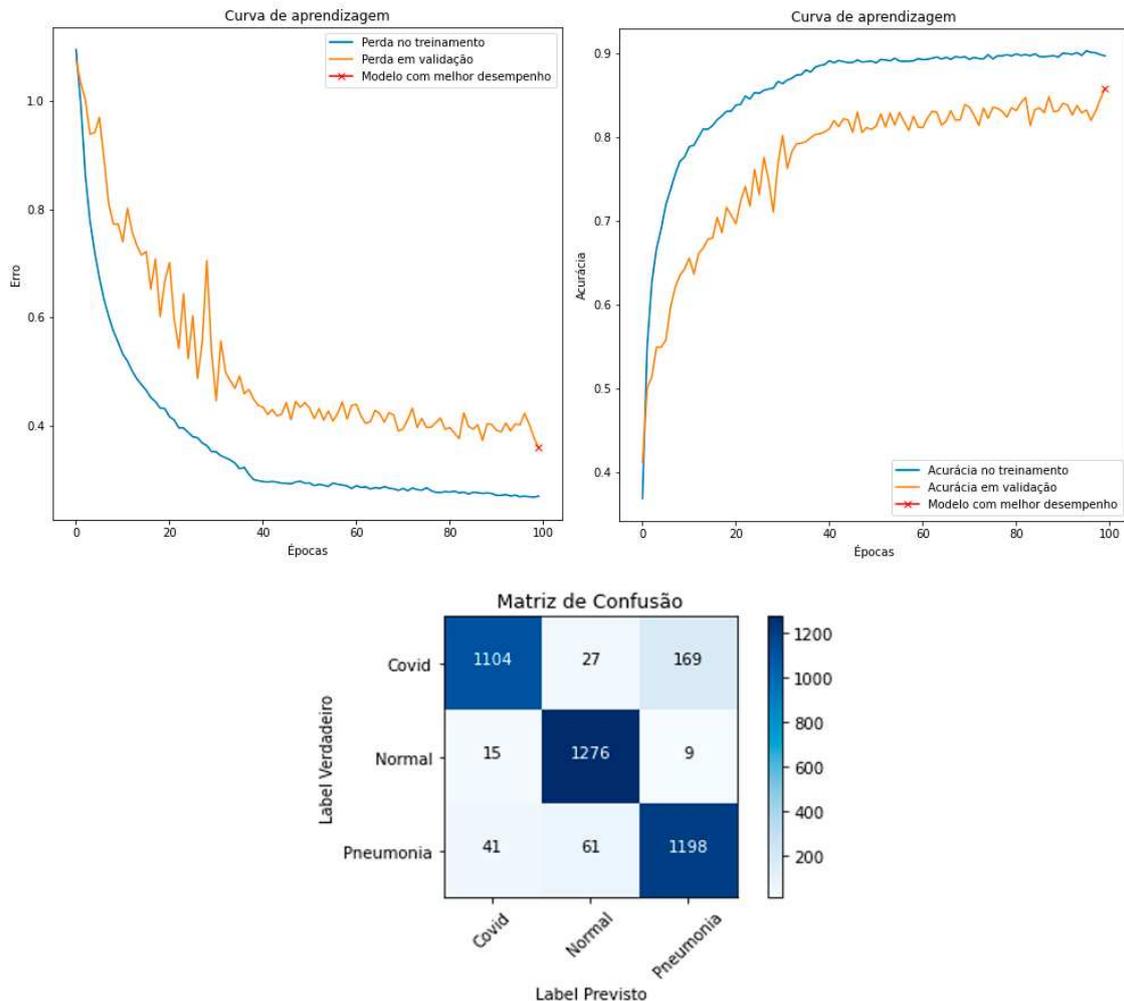


Figura 4.26. Curvas de aprendizagem para o modelo LeTEN do Experimento 2, e sua matriz de confusão. Nas curvas de aprendizagem, tanto a curva de perda quanto a curva da acurácia é observada.

2, onde a dificuldade era maior, ainda sim, conseguimos alcançar métricas acima de 90%. Enquanto que a VGG-19 se sai ainda melhor em ambos os experimentos, principalmente para tarefas com múltiplas classes, como podemos ver a VGG-19 comete pouquíssimos erros na classificação de COVID-19 e Pneumonia.

Nesse capítulo também resolvemos testar a U-Net, a fim de obtermos resultados promissores com uma arquitetura simples, mais rápida e mais leve em comparação com as demais arquiteturas presentes na literatura, como: Attention U-Net [Oktay et al. 2018]; R2U-Net [Alom et al. 2018]; Attention R2U-Net (integração de R2U-Net e Attention U-Net), e 3D U-Net [Çiçek et al. 2016].

Para a tarefa de segmentação o método proposto apresenta resultados promissores, mesmo com a limitação no banco de dados e lesões extremamente pequenas, o que dificulta muito o desempenho da U-Net. Mesmo com uma arquitetura mais simples, podemos obter resultados robusto, principalmente quando utilizada com o método de pré-processamento proposto. A configuração da U-Net também é uma limitação nesse

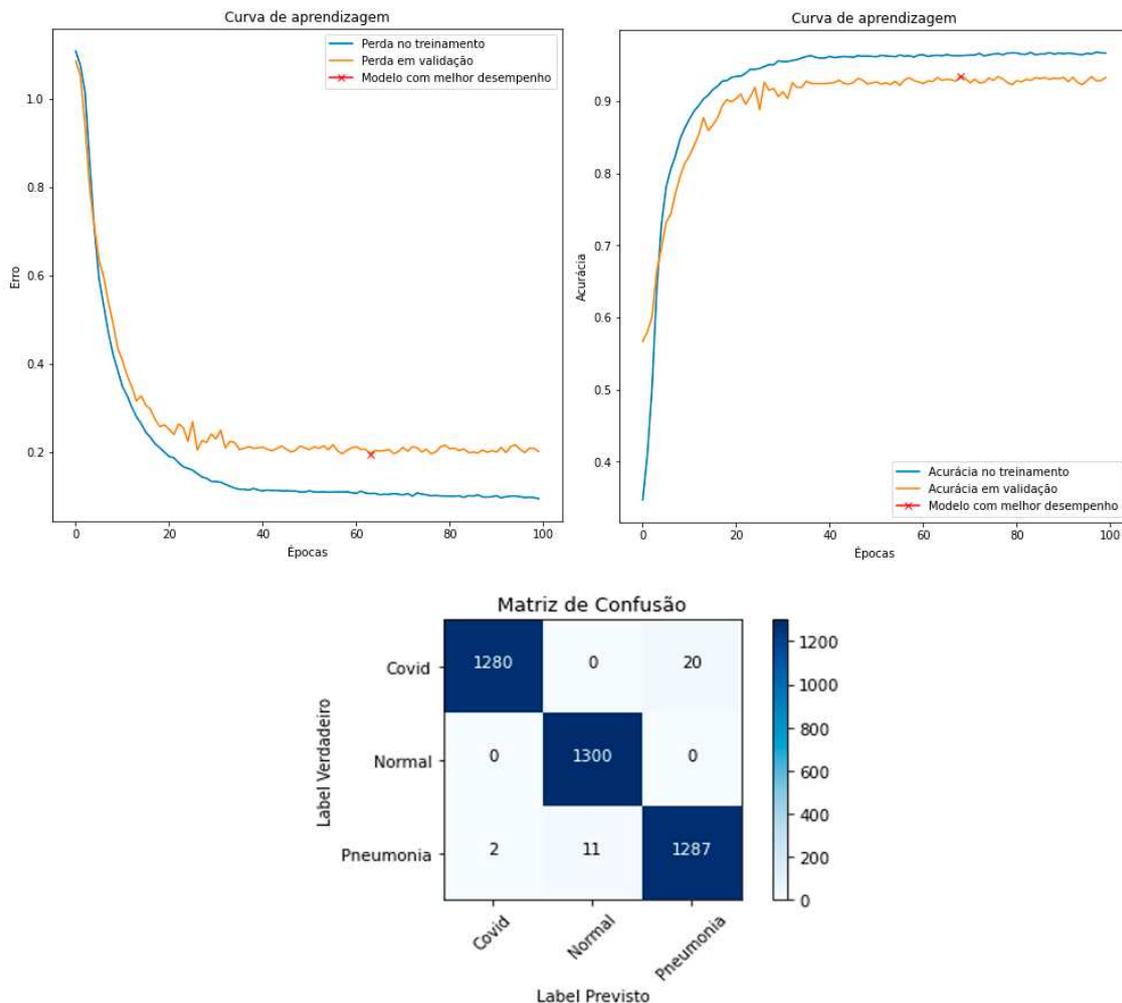


Figura 4.27. Curvas de aprendizagem para o modelo VGG-19 do Experimento 2, e sua matriz de confusão. Nas curvas de aprendizagem, tanto a curva de perda quanto a curva da acurácia é observada.

capítulo, principalmente quando trabalhamos com imagens com grandes resoluções, onde a limitação de *hardware* pode limitar a implementação. Além de que as pequenas lesões são bastantes complexas de serem identificadas e os principais trabalhos tratam dessa limitação com outras versões mais complexas da U-Net e funções de perdas diferentes do convencional.

4.9. Conclusão

Na tarefa de classificação podemos concluir que redes simples com poucas camadas igual a LeTEN e redes mais profundas como a VGG-19 que se utiliza também de transferência de aprendizagem, podemos alcançar métricas promissoras para a classificação de exames médicos de TC. Para essa tarefa as redes ainda podem ser melhoradas, assim como a classificação, podendo ser utilizado aumento de dados, um conjunto de imagens com mais exemplos para cada classe, os parâmetros das redes podem ser ajustados e melhorados. Assim como a utilização de outras redes.

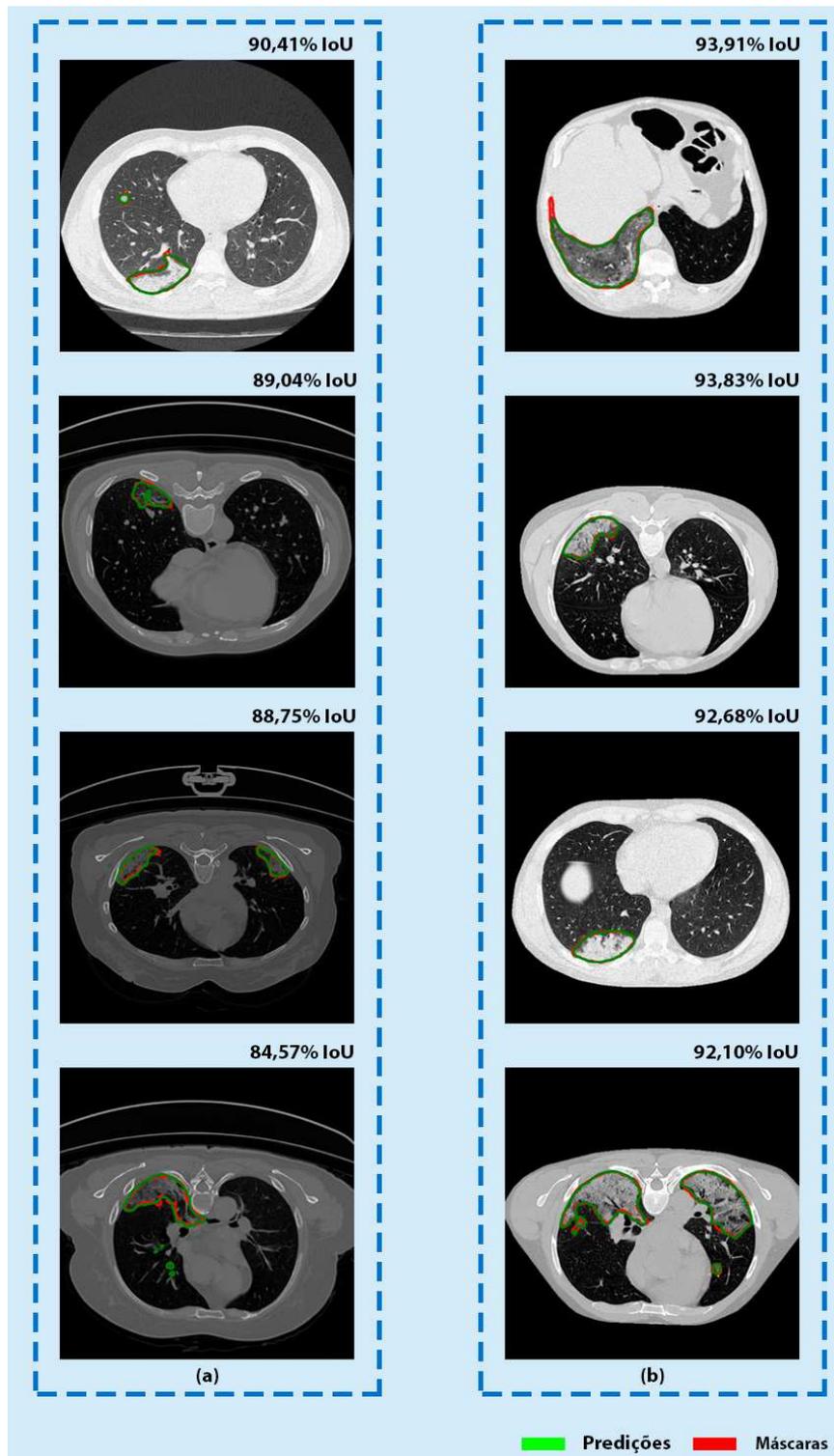


Figura 4.28. Representação das melhores previsões do modelo, a métrica IoU é definida para representar as melhores segmentações. Na coluna (a) temos as melhores segmentações para o Dataset 1 sem pré-processamento, e a coluna (b), apresenta os melhores resultados para o Dataset 1 com pré-processamento.

Para a tarefa de segmentação podemos concluir que mesmo a U-Net sofrendo

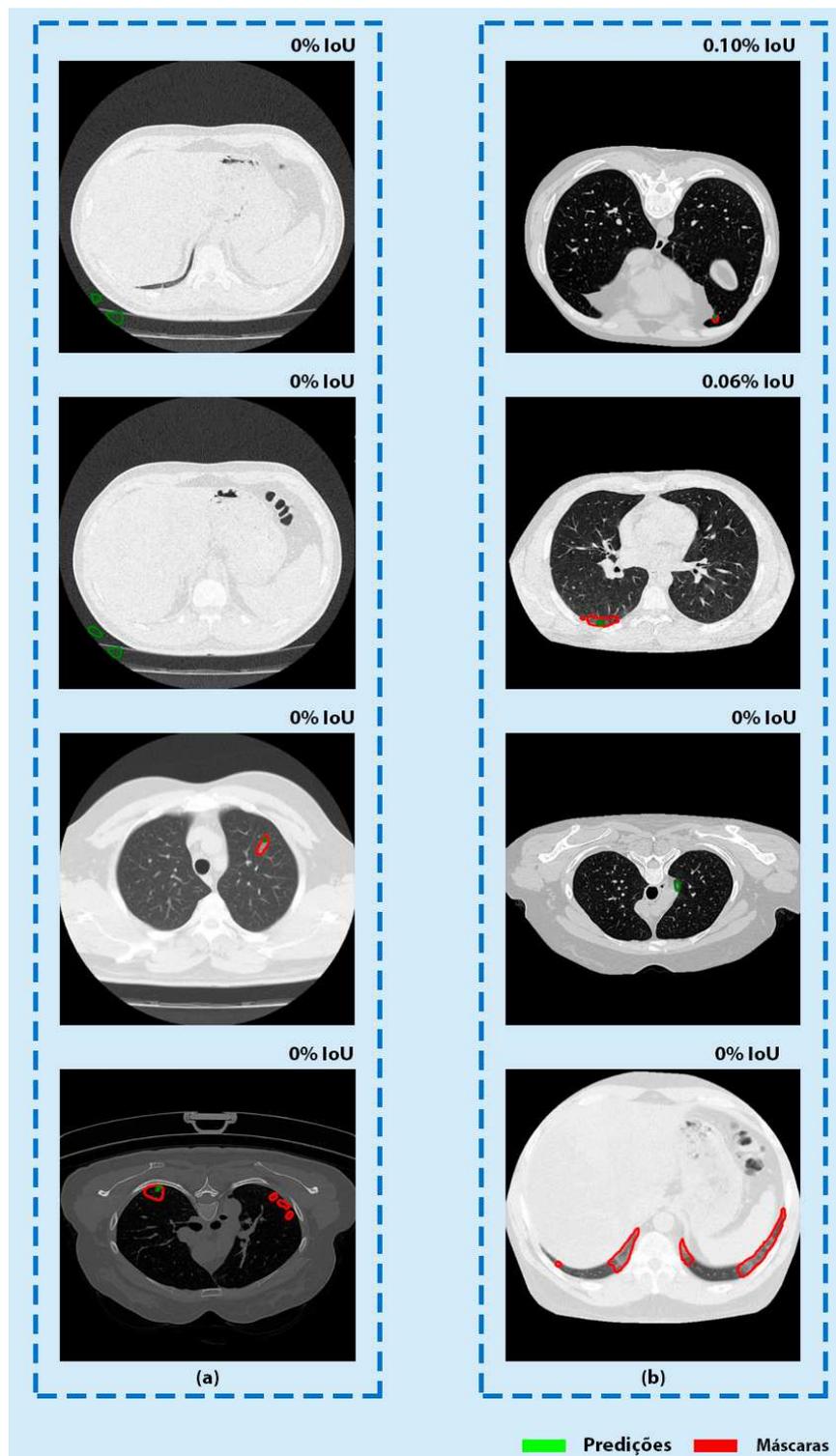


Figura 4.29. Representação das piores previsões feitas pelo modelo, a métrica IoU é definida para calcular as piores segmentações. Na coluna (a) temos as piores segmentações para o Dataset 1 sem pré-processamento, e a coluna (b), apresenta os piores resultados para o Dataset 1 com pré-processamento.

com algumas limitações, ela pode ser utilizada para esse problema e obter resultados

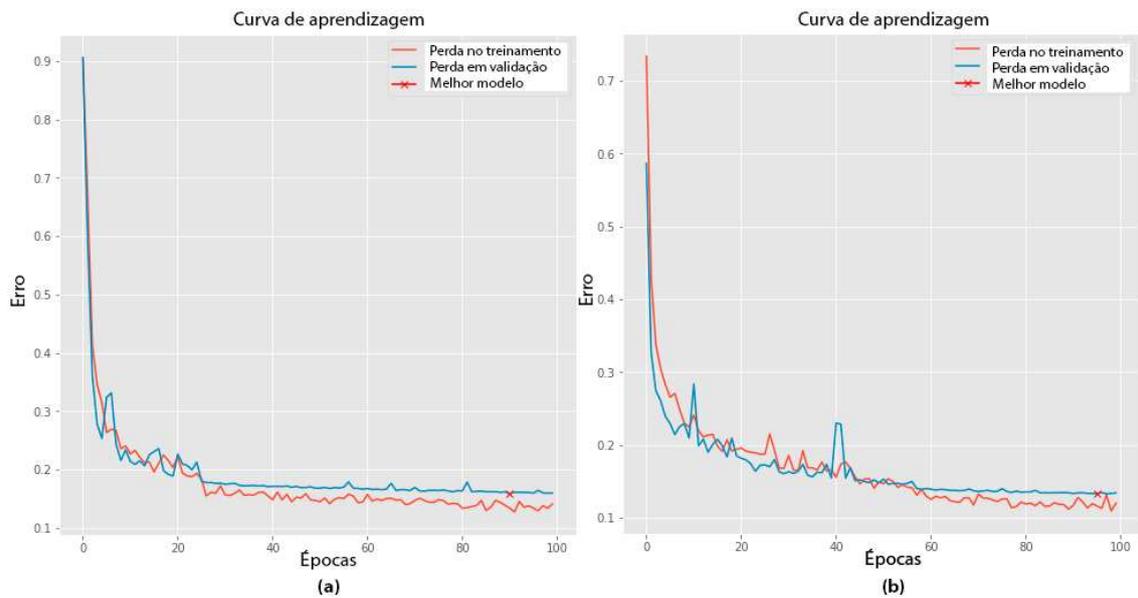


Figura 4.30. Representação das curvas de aprendizado, observando o erro do modelo de segmentação, em (a), temos a curva do erro durante o treinamento e validação do modelo sem pré-processamento, já em (b), temos a curva do erro para o modelo com pré-processamento.

promissores, também podemos validar que o método proposto de pré-processamento é bastante útil para melhorar os resultados, além de fazer com que a U-Net possa identificar pequenas lesões mesmo com a limitação de arquitetura.

A rede U-Net ainda pode ser melhorada utilizando outras funções de perdas para tornar o modelo mais robusto e sensível a pequenas lesões, pode ser utilizado também a técnica de aumento de dados para obtermos um dataset maior, com mais variações de imagens, outros datasets podem ser utilizados para compor um maior número de exames e então utilizar alguma U-Net 3D para obter resultados com outros tipos de segmentações.

Agradecimentos

Este capítulo foi realizado com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código Financeiro 001. Agradecemos também o apoio da Fundação de Amparo à Pesquisa do Estado do Piauí (FAPEPI).

Referências

[Abadi et al. 2015] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

- [Alom et al. 2018] Alom, M. Z., Hasan, M., Yakopcic, C., Taha, T. M., and Asari, V. K. (2018). Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation. *CoRR*, abs/1802.06955.
- [Ardakani et al. 2020] Ardakani, A. A., Kanafi, A. R., Acharya, U. R., Khadem, N., and Mohammadi, A. (2020). Application of deep learning technique to manage covid-19 in routine clinical practice using ct images: Results of 10 convolutional neural networks. *Computers in Biology and Medicine*, 121:103795.
- [Axiaq et al. 2021] Axiaq, A., Almohtadi, A., Massias, S. A., Ngemoh, D., and Harky, A. (2021). The role of computed tomography scan in the diagnosis of covid-19 pneumonia. *Current Opinion in Pulmonary Medicine*, 27(3).
- [Bradski 2000] Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- [Castiglione et al. 2021] Castiglione, A., Vijayakumar, P., Nappi, M., Sadiq, S., and Umer, M. (2021). Covid-19: Automatic detection of the novel coronavirus disease from ct images using an optimized convolutional neural network. *IEEE Transactions on Industrial Informatics*, 17(9):6480–6488.
- [Çiçek et al. 2016] Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., and Ronneberger, O. (2016). 3d u-net: Learning dense volumetric segmentation from sparse annotation. In Ourselin, S., Joskowicz, L., Sabuncu, M. R., Unal, G., and Wells, W., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, pages 424–432, Cham. Springer International Publishing.
- [Clifford 2006] Clifford, G. D. (2006). Shortliffe edward h, cimino james j: "biomedical informatics; computer applications in health care and biomedicine". *BioMedical Engineering OnLine*, 5(1):61.
- [Cohen et al. 2020] Cohen, J. P., Morrison, P., Dao, L., Roth, K., Duong, T. Q., and Ghassemi, M. (2020). Covid-19 image data collection: Prospective predictions are the future.
- [Dananché et al. 2022] Dananché, C., Elias, C., Hénaff, L., Amour, S., Kuczewski, E., Gustin, M.-P., Escuret, V., Saadatian-Elahi, M., and Vanhems, P. (2022). Baseline clinical features of COVID-19 patients, delay of hospital admission and clinical outcome: A complex relationship. *PLoS One*, 17(1):e0261428.
- [de Carvalho et al. 2021] de Carvalho, L. S., da Silva Júnior, R. T., Oliveira, B. V. S., de Miranda, Y. S., Rebouças, N. L. F., Loureiro, M. S., Pinheiro, S. L. R., da Silva, R. S., Correia, P. V. S. L. M., Silva, M. J. S., Ribeiro, S. N., da Silva, F. A. F., de Brito, B. B., Santos, M. L. C., Leal, R. A. O. S., Oliveira, M. V., and de Melo, F. F. (2021). Highlighting COVID-19: What the imaging exams show about the disease. *World J. Radiol.*, 13(5):122–136.
- [Deng et al. 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

- [Gulli and Pal 2017] Gulli, A. and Pal, S. (2017). *Deep learning with Keras*. Packt Publishing Ltd.
- [Gunraj et al. 2020] Gunraj, H., Wang, L., and Wong, A. (2020). Covidnet-ct: A tailored deep convolutional neural network design for detection of covid-19 cases from chest ct images. *Frontiers in Medicine*, 7:1025.
- [Harris et al. 2020] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.
- [Hunter 2007] Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.
- [Jacobi et al. 2020] Jacobi, A., Chung, M., Bernheim, A., and Eber, C. (2020). Portable chest X-ray in coronavirus disease-19 (COVID-19): A pictorial review. *Clinical Imaging*, 64:35–42.
- [Jagatheeswari et al. 2009] Jagatheeswari, P., Kumar, S. S., and Rajaram, M. (2009). Contrast stretching recursively separated histogram equalization for brightness preservation and contrast enhancement. In *2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies*, pages 111–115.
- [Jun et al. 2020] Jun, M., Cheng, G., Yixin, W., Xingle, A., Jiantao, G., Ziqi, Y., Mingqing, Z., Xin, L., Xueyuan, D., Shucheng, C., Hao, W., Sen, M., Xiaoyu, Y., Ziwei, N., Chen, L., Lu, T., Yuntao, Z., Qiongjie, Z., Guoqiang, D., and Jian, H. (2020). COVID-19 CT Lung and Infection Segmentation Dataset. *Zenodo*.
- [Júnior and Yamashita 2001] Júnior, E. A. and Yamashita, H. (Maio 2001). Aspectos básicos de tomografia computadorizada e ressonância magnética. *Brazilian Journal of Psychiatry*, 23.
- [Kazimierczuk and Jozwik 1990] Kazimierczuk, M. and Jozwik, J. (1990). Analysis and design of class e zero-current-switching rectifier. *IEEE Transactions on Circuits and Systems*, 37(8):1000–1009.
- [LeCun et al. 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.
- [Lecun et al. 1998] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- [Ma et al. 2021] Ma, J., Wang, Y., An, X., Ge, C., Yu, Z., Chen, J., Zhu, Q., Dong, G., He, J., He, Z., Cao, T., Zhu, Y., Nie, Z., and Yang, X. (2021). Toward data-efficient learning: A benchmark for covid-19 ct lung and infection segmentation. *Medical Physics*, 48(3):1197–1210.

- [Marques et al. 2021] Marques, J., Veras, R., and Silva, R. (2021). A literature review: Detection of covid-19 in computed tomography images using deep learning. In *Anais do XIV Encontro Unificado de Computação do Piauí e XI Simpósio de Sistemas de Informação*, pages 9–16, Porto Alegre, RS, Brasil. SBC.
- [Marques et al. 2022] Marques, J. V., Veras, R., and Silva, R. R. (2022). Detection of covid-19 in computed tomography images using deep learning: A literature review. *Revista de Sistemas e Computação-RSC*, 12(1).
- [Martínez Chamorro et al. 2021] Martínez Chamorro, E., Díez Tascón, A., Ibáñez Sanz, L., Ossaba Vélez, S., and Borrueal Nacenta, S. (2021). Diagnóstico radiológico del paciente con covid-19. *Radiología*, 63(1):56–73.
- [Moreira et al. 2017] Moreira, F. A., De Almeida, L. A., and Galvão, A. (2017). *Guia de Diagnóstico por Imagem: O passo a passo que todo médico deve saber*. Elsevier Brasil.
- [Moreiran 2017] Moreiran, F. (2017). *Guia de diagnóstico por imagem: O passo a passo que todo médico deve saber*. GEN Guanabara Koogan.
- [Oktay et al. 2018] Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M. C. H., Heinrich, M. P., Misawa, K., Mori, K., McDonagh, S. G., Hammerla, N. Y., Kainz, B., Glocker, B., and Rueckert, D. (2018). Attention u-net: Learning where to look for the pancreas. *CoRR*, abs/1804.03999.
- [Organization 2022] Organization, W. H. (2022). Weekly epidemiological update on covid-19 - 1 march 2022. Acessado: 18-05-2022.
- [Otsu 1979] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66.
- [Pedregosa et al. 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.
- [Ponti and da Costa 2017] Ponti, M. A. and da Costa, G. B. P. (Tópicos em Gerenciamento de Dados e Informações, 2017). Como funciona o deep learning. *SBC, 1a ed.*, 23.
- [Ronneberger et al. 2015a] Ronneberger, O., Fischer, P., and Brox, T. (2015a). U-net: Convolutional networks for biomedical image segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham. Springer International Publishing.
- [Ronneberger et al. 2015b] Ronneberger, O., P.Fischer, and Brox, T. (2015b). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer. (available on arXiv:1505.04597 [cs.CV]).

- [Ronneberger et al. 2015c] Ronneberger, O., P.Fischer, and Brox, T. (2015c). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer. (available on arXiv:1505.04597 [cs.CV]).
- [Rosa et al. 2020] Rosa, M. E. E., Matos, M. J. R. d., Furtado, R. S. O. d. P., Brito, V. M., Amaral, L. T. W., Beraldo, G. L., Fonseca, E. K. U. N., Chate, R. C., Passos, R. B. D., Teles, G. B. d. S., Silva, M. M. A., Yokoo, P., Yanata, E., Shoji, H., Szarf, G., and Funari, M. B. d. G. (2020). COVID-19 findings identified in chest computed tomography: a pictorial essay. *Einstein (Sao Paulo)*, 18:eRW5741.
- [Rubin et al. 2020] Rubin, G. D., Ryerson, C. J., Haramati, L. B., Sverzellati, N., Kanne, J. P., Raoof, S., Schluger, N. W., Volpi, A., Yim, J.-J., Martin, I. B. K., Anderson, D. J., Kong, C., Altes, T., Bush, A., Desai, S. R., Goldin, o., Goo, J. M., Humbert, M., Inoue, Y., Kauczor, H.-U., Luo, F., Mazzone, P. J., Prokop, M., Remy-Jardin, M., Richeldi, L., Schaefer-Prokop, C. M., Tomiyama, N., Wells, A. U., and Leung, A. N. (2020). The role of chest imaging in patient management during the covid-19 pandemic: A multinational consensus statement from the fleischner society. *Radiology*, 296(1):172–180. PMID: 32255413.
- [Samir et al. 2021] Samir, A., Naguib, N. N. N., Elnekeidy, A., Baess, A. I., and Shawky, A. (2021). Covid-19 versus h1n1: challenges in radiological diagnosis—comparative study on 130 patients using chest hrct. *Egyptian Journal of Radiology and Nuclear Medicine*, 52(1):77.
- [Simonyan and Zisserman 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*.
- [Stephanie et al. 2020] Stephanie, S., Shum, T., Cleveland, H., Challa, S. R., Herring, A., Jacobson, F. L., Hatabu, H., Byrne, S. C., Shashi, K., Araki, T., Hernandez, J. A., White, C. S., Hossain, R., Hunsaker, A. R., and Hammer, M. M. (2020). Determinants of chest x-ray sensitivity for COVID- 19: A multi-institutional study in the united states. *Radiol Cardiothorac Imaging*, 2(5):e200337.
- [Sverzellati et al. 2021] Sverzellati, N., Ryerson, C. J., Milanese, G., Renzoni, E. A., Volpi, A., Spagnolo, P., Bonella, F., Comelli, I., Affanni, P., Veronesi, L., Manna, C., Ciuni, A., Sartorio, C., Tringali, G., Silva, M., Michieletti, E., Colombi, D., and Wells, A. U. (2021). Chest radiography or computed tomography for COVID-19 pneumonia? comparative study in a simulated triage setting. *Eur. Respir. J.*, 58(3):2004188.
- [Tang et al. 2018] Tang, A., Tam, R., Cadrin-Chênevert, A., Guest, W., Chong, J., Barfett, J., Chepelev, L., Cairns, R., Mitchell, J. R., Cicero, M. D., Poudrette, M. G., Jaremko, J. L., Reinhold, C., Gallix, B., Gray, B., Geis, R., and for the Canadian Association of Radiologists (CAR) Artificial Intelligence Working Group (2018). Canadian association of radiologists white paper on artificial intelligence in radiology. *Canadian Association of Radiologists Journal*, 69(2):120–135. PMID: 29655580.

- [Van der Walt et al. 2014] Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., and Yu, T. (2014). scikit-image: image processing in python. *PeerJ*, 2:e453.
- [Wang et al. 2020] Wang, W., Xu, Y., Gao, R., Lu, R., Han, K., Wu, G., and Tan, W. (2020). Detection of SARS-CoV-2 in Different Types of Clinical Specimens. *JAMA*, 323(18):1843–1844.
- [Zhao et al. 2021] Zhao, Q., Wang, H., and Wang, G. (2021). Lcov-net: A lightweight neural network for covid-19 pneumonia lesion segmentation from 3d ct images. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 42–45.
- [Zhou et al. 2021] Zhou, T., Lu, H., Yang, Z., Qiu, S., Huo, B., and Dong, Y. (2021). The ensemble deep learning model for novel covid-19 on ct images. *Applied Soft Computing*, 98:106885.