



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Uma rede UNet modificada para segmentação e identificação de lesões pulmonares em tomografias

Thiago Araujo da Silva

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia de Computação

Orientador
Prof. Dr. Díbio Leandro Borges

Brasília
2021

Dedicatória

Este trabalho é dedicado a minha família, especialmente ao meu pai, que apesar de não estar mais entre nós, sempre me apoiou e batalhou para me ajudar a alcançar meus sonhos.

Agradecimentos

Agradeço primeiramente à minha família, Ronaldo, Raquel e Tayna, por todo o apoio, carinho e suporte em minha vida.

Aos meus amigos, Victor, Felipe, Romeu, Patrick, De Lima, Nicolas, Arthur, Diniz e Igor que sempre me ajudaram e contribuíram diretamente, ou indiretamente, para a formação da pessoa que sou hoje.

A minha namorada e melhor amiga Catarina, pelo companheirismo, conselhos e apoio fornecidos durante esta fase difícil da minha vida.

Ao professor Díbio, pela contribuição e orientação durante todas as etapas de elaboração deste trabalho.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

Com o surgimento e disseminação do novo Coronavírus, engenheiros, virologistas, epidemiologistas e profissionais de outras áreas da saúde estão trabalhando para ajudar a conter essa pandemia o mais rápido possível. No processo de estudo e determinação dos indícios diagnósticos para a doença, notou-se a presença recorrente de achados pulmonares em tomografias computadorizadas de tórax dos pacientes infectados. Assim como na infecção por coronavírus, número significativo de doenças respiratórias do trato inferior, causadas por agentes externos, manifestam-se nas tomografias. Por tais fatos, automatizar o processo de identificação destas anomalias pulmonares, pode agilizar e facilitar o diagnóstico de profissionais da área da saúde, em casos como estes. Este trabalho, propõe o uso de uma arquitetura UNet modificada para realizar a segmentação e identificação das lesões pulmonares nas tomografias de tórax. Foram criados conjuntos de treinamento, validação e teste, utilizando-se dados abertos disponibilizados pela comunidade científica, e aplicando-se técnicas de pré-processamento e aumento de dados para melhoria da qualidade das amostras de entrada. Como resultado, foi comparada e discutida a validação da utilização de um esqueleto UNet para segmentação das imagens e a importância do pré-processamento dos dados para se obter bons resultados em modelos mais simples. A topologia proposta obteve um resultado de 98.01% de acurácia nos dados de teste.

Palavras-chave: Aprendizado Profundo, Redes Neurais Convolucionais, Segmentação Semântica, UNet, Lesões pulmonares

Abstract

With the arrival of the new Coronavirus, engineers, virologists, epidemiologists and people from other medical fields are working to help contain this pandemic as quickly as possible. In the process of studying and determining diagnostic evidence, the recurrent presence of pulmonary findings was noted in chest CT scans of infected patients. Just as coronavirus infection, a significant number of lower tract respiratory diseases, caused by external agents, are manifested on CT scans. For those facts, automating the process of identifying these pulmonary anomalies can speed up and facilitate professional diagnosis, in cases like these. This work proposes the use of a modified UNet architecture to segment and identify lung lesions. Training, validation and testing sets were created, using open published data made available by the scientific community, and applying pre-processing and data augmentation techniques to improve the dataset quality. As a result, the validation of using a UNet backbone in image segmentation tasks is compared and discussed, and the importance of pre-processing the data to obtain good results in simpler models. The proposed topology obtained a result of 98.01 % accuracy in the test data.

Keywords: Deep learning, Convolutional Neural Networks, Semantic Segmentation, UNet, Pulmonary lesions

Sumário

1	Introdução	1
1.1	Objetivos	3
1.2	Estrutura do trabalho	4
2	Fundamentação Teórica	5
2.1	Inteligência artificial	5
2.2	Aprendizado de Máquina	5
2.2.1	<i>Overfitting e Underfitting</i>	6
2.3	Redes Neurais Artificiais	6
2.3.1	Função de Ativação	7
2.3.2	Rede Neural Multicamadas	8
2.4	Aprendizado Profundo	9
2.5	Redes Neurais Convolucionais	10
2.5.1	Camada de convolução	11
2.5.2	Camada de <i>pooling</i>	12
2.5.3	Camada Totalmente Conectada	12
2.6	Segmentação Semântica	13
3	Metodologia	14
3.1	Detalhes de implementação	14
3.2	Banco de dados	14
3.3	Pré-processamento	15
3.3.1	Conversão de formato	15
3.3.2	Normalização	15
3.3.3	Equalização de histograma	16
3.3.4	Recorte pulmonar	17
3.4	Aumento de dados	19
3.5	Dados de treinamento, validação e teste	20

3.6	Arquitetura	21
3.6.1	UNet	21
3.6.2	UNet Modificada	22
4	Resultados	26
4.1	Treinamento e validação	26
4.1.1	UNet Clássica	26
4.1.2	UNet Modificada	29
4.2	Resultados visuais	31
5	Conclusão	33
5.1	Trabalhos Futuros	34

Lista de Figuras

1.1	Tomografia COVID-19	2
1.2	Tomografia Pneumonia Bacteriana	2
1.3	Tomografia H1N1	3
2.1	Neurônio Artificial	6
2.2	Sigmóide	7
2.3	ReLU	7
2.4	MLP com duas camadas ocultas	8
2.5	Modelo DL extraindo conceitos simples e abstratos	9
2.6	Arquitetura de uma Rede Neural Convolutiva	10
2.7	Representação visual da camada de convolução	11
2.8	Preenchimento para um filtro 2×2	12
2.9	Segmentação Semântica	13
3.1	Dados de entrada: Tomografia, Máscara Pulmonar e Máscara das lesões	15
3.2	Tomografia original e seu respectivo histograma	16
3.3	Tomografia equalizada e seu respectivo histograma	17
3.4	Processo de cálculo do contorno pulmonar	18
3.5	Limite pulmonar calculado e imagens recortadas	19
3.6	Resultado do aumento de dados aplicado em duas imagens	20
3.7	Arquitetura UNet clássica	22
3.8	Arquitetura UNet proposta em (KUCHANA et al., 2021)	23
3.9	Arquitetura UNet Modificada	24
3.10	Blocos detalhados	24
4.1	Gráfico de acurácia da rede UNet clássica	27
4.2	Gráfico de custo da rede UNet clássica	28
4.3	Gráfico de acurácia da rede UNet modificada	29
4.4	Gráfico de custo da rede UNet modificada	30
4.5	Saídas previstas pelo modelo: Pulmão e Lesões	31

4.6 Sobreposição das lesões previstas com a tomografia original 32

Lista de Tabelas

4.1 Resultados do conjunto de teste da UNet clássica.	28
4.2 Resultados do conjunto de teste da UNet modificada.	30
5.1 Comparação do modelo com outros trabalhos citados.	33

Lista de Abreviaturas e Siglas

AI *Artificial Intelligence.*

ANN *Artificial Neural Network.*

CL *Convolutional Layer.*

CNN *Convolutional Neural Network.*

DL *Deep Learning.*

FC *Fully Connected Layers.*

FCNN *Fully Convolutional Neural Network.*

ML *Machine Learning.*

MLP *Multilayer Perceptron.*

Capítulo 1

Introdução

Em 11 de março de 2020, a Organização Mundial da Saúde declarou pandêmico o surto do novo vírus COVID-19, sendo esta considerada a primeira pandemia causada por um coronavírus (ORGANIZATION, 2020a). Em ordem para frear o contágio, é necessário que os países detectem, testem, tratem, isolem, rastreiem e mobilizem sua população, desta maneira, aqueles com poucos casos podem evitar que esses casos se tornem *clusters* e esses *clusters* se tornem transmissões comunitárias (ORGANIZATION, 2020b).

Estudos realizados com pacientes que adquiriram a infecção demonstraram que os sintomas iniciais comuns são febre, tosse, mialgia e fadiga, sendo que grande índice de pacientes desenvolveram pneumonia com tomografia alterada. As principais complicações apresentadas foram Síndrome do desconforto respiratório, lesões cardíacas, lesões pulmonares e possíveis infecções secundárias (ZHI, 2020).

A experiência clínica revela que a detecção por RT-PCR de RNA viral em muco nasofaríngeo tem uma baixa taxa de assertividade nos estágios iniciais da doença, ao passo em que um número significativo de tomografias anormais do tórax são obtidas de pacientes com esta doença nos mesmos estágios (HUANG et al., 2020).

Características como opacidade em vidro fosco, distribuições periféricas, múltiplos focos de infecção e consolidações podem ser observadas nas Figuras 1.1, 1.2 e 1.3, provenientes de diferentes tipos de doenças respiratórias, como COVID-19, Pneumonia Bacteriana e H1N1.

Dessa maneira a identificação e classificação de lesões pulmonares em tomografias continua sendo um dos principais meios diagnósticos para doenças respiratórias do trato inferior.

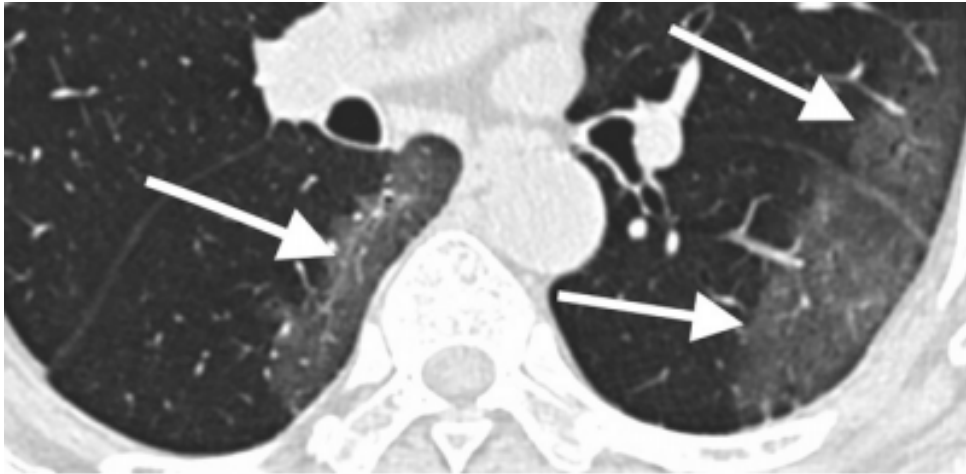


Figura 1.1: Tomografia COVID-19 (CHUNG et al., 2020)



Figura 1.2: Tomografia Pneumonia Bacteriana (REITTNER et al., 2003)

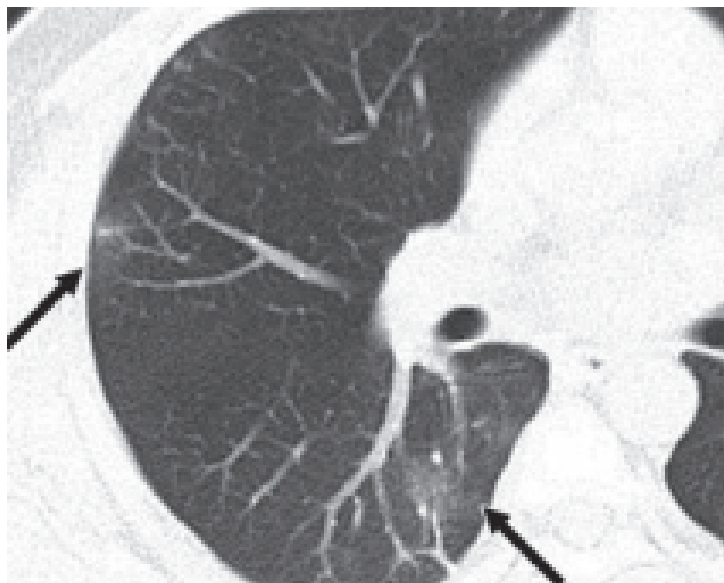


Figura 1.3: Tomografia H1N1 (YIN et al., 2020)

Devido aos pontos mencionados e características demonstradas, é possível automatizar o processo de segmentação dos focos de infecção e lesão pulmonar, utilizando-se técnicas de aprendizado de máquina, que facilitam e agilizam o diagnóstico de doenças respiratórias do trato inferior com base nas imagens de tomografia do tórax.

1.1 Objetivos

O principal objetivo desse trabalho é desenvolver uma solução que auxilie e facilite o diagnóstico de um profissional da saúde ao visualizar uma tomografia computadorizada de um tórax com possíveis lesões pulmonares, segmentando-as, e aferindo a porcentagem do pulmão afetada. Para se alcançar o resultado esperado, deve-se atingir as seguintes metas:

- Obter dados que representem a problemática apresentada;
- Revisar bibliografias para definir o modelo e arquitetura a serem usados como base;
- Propor e desenvolver melhorias no modelo de base;
- Realizar validações e testes no modelo proposto;
- Comparar resultados, mostrar limitações e propor possíveis melhorias.

1.2 Estrutura do trabalho

O Capítulo 1 apresenta uma contextualização da motivação e objetivos do trabalho, e por fim apresenta a estrutura do trabalho.

O Capítulo 2 abrange toda fundamentação teórica necessária para entender da problemática e dos métodos utilizados para a execução do trabalho, tendo uma abordagem *top-down*. Apresenta conteúdos mais abrangentes como Inteligencia Artificial e vai especificando ao longo do capítulo, até alcançar os tópicos tratados neste trabalho, como Aprendizado Profundo e Redes Neurais Totalmente Convolucionais.

O Capítulo 3 descreve os materiais, métodos e técnicas utilizadas para que seja possível reproduzir este experimento, baseados em aspectos teóricos e práticos.

O Capítulo 4 apresenta os resultados, discussões e análises obtidas, e explora as possíveis limitações e melhorias a serem feitas.

Por fim, o Capítulo 5 traz conclusões do trabalho e considerações a cerca de possíveis trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este capítulo tem como objetivo apresentar algumas áreas da Inteligência Artificial utilizadas neste projeto. Iniciando com uma breve introdução sobre Aprendizado de Máquinas e mostrando algumas técnicas e termos utilizados. Em seguida, será apresentado os fundamentos teóricos de Aprendizagem Profunda e Redes Neurais Convolucionais, com os principais métodos, características e benefícios de sua utilização.

2.1 Inteligência artificial

A Inteligência Artificial, ou *Artificial Intelligence* (AI), pode ser definida sobre diversos pontos de vista, sendo estes organizados em quatro principais categorias: sistemas que pensam como humanos, sistemas que agem como humanos, sistemas que pensam racionalmente e sistemas que agem racionalmente (RUSSELL; NORVIG, 1994). Dessa maneira, pode-se entender inteligência artificial como o ramo da computação onde se propõe a criação de sistemas e dispositivos inteligentes, capazes de simular um comportamento humano, sendo ele um pensamento, um raciocínio ou uma atitude.

2.2 Aprendizado de Máquina

Aprendizado pode ser entendido como um comportamento em que uma pessoa melhora a si mesmo através de interações com o mundo exterior. Aprendizado de Máquina, ou *Machine Learning* (ML), é uma subárea da inteligência artificial, que busca simular esse comportamento em computadores executando uma determinada tarefa.

A máquina simula esse comportamento explorando informações úteis a partir da experiência histórica pré coletada, sob o critério de maximizar uma medida de desempenho, para treinar um bom modelo e ajudar a analisar possíveis dados futuros (JORDAN; MITCHELL, 2015).

2.2.1 *Overfitting e Underfitting*

O que separa Aprendizado de Máquina de um problema de otimização é a necessidade de técnicas capazes de treinar modelos que generalizem exemplos nunca antes processados (RUSSELL; NORVIG, 1994).

Para determinar o desempenho de generalização de uma rede neural, é necessário dois conjuntos de dados, um de treinamento e um de teste. O objetivo é diminuir ambos os erros de treinamento e de teste, quando esses dois não caminham juntos, ocorrem dois fenômenos definidos como *Underfitting* e *Overfitting*.

O *Underfitting* ocorre quando o erro de treinamento não reduz, ou seja, o modelo não é capaz de determinar uma relação nos dados de treinamento, por outro lado, o *Overfitting* ocorre quando o erro de treinamento diminui, porém o de teste continua alto, ou seja, a rede não é capaz de generalizar para exemplos não vistos.

2.3 Redes Neurais Artificiais

Redes Neurais Artificiais, ou *Artificial Neural Network* (ANN), são modelos computacionais de aprendizado de máquina inspirados em redes neurais biológicas, projetados para simular a maneira como o cérebro humano processa as informações. As ANNs obtêm seu conhecimento detectando padrões e relacionamentos nos dados e aprendem através da experiência.

Assim como uma rede neural biológica, a unidade básica da ANN é o neurônio. Cada neurônio pode ser representado como na Figura 2.1

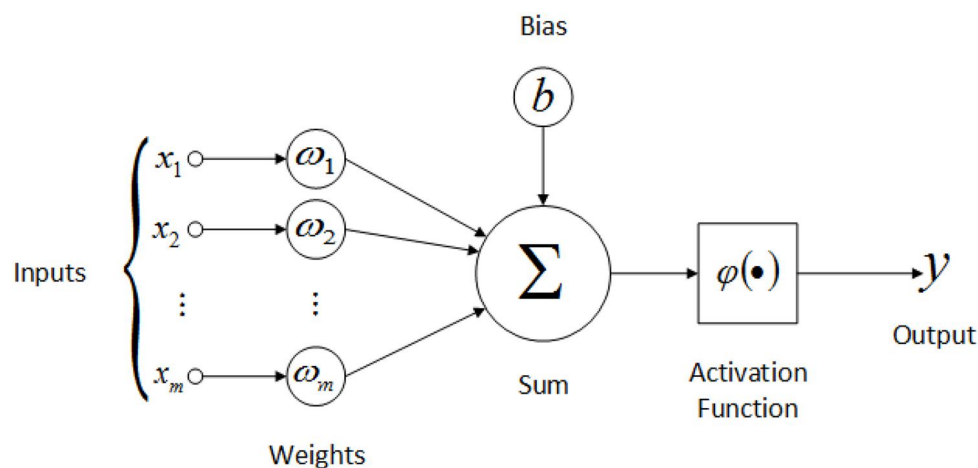


Figura 2.1: Neurônio Artificial (OLIVEIRA et al., 2017)

sendo a função de cada elemento:

- *Input* (x_m): Entrada do neurônio, podendo ser dados externos de treinamento ou as saídas de outros neurônios.
- *Weights* (ω_m): Peso sináptico, um número real que representa a conexão sináptica do neurônio, determina a influência da entrada para o resultado final, sendo excitatória se $\omega_m > 0$ ou inibitória caso $\omega_m < 0$.
- *Bias* (b): Tendência, permite movimentar a função de ativação adicionando uma constante à entrada, aumentando o grau de liberdade e permitindo uma melhor adaptação da rede.
- *Sum* (Σ): Soma, é o núcleo do neurônio, onde cada entrada é multiplicada pelo seu devido peso para que em seguida passe pela função de ativação.
- *Activation Function* ($\varphi(\bullet)$): Função de ativação, é o potencial necessário para ativação do neurônio, com a aplicação de uma não-linearidade que determina como ele deverá ser ativado.

Matematicamente, pode-se descrever o neurônio artificial por uma função com as variáveis de entrada x_m e saída y , como descrito pela equação (2.1):

$$y = \varphi\left(\sum_{i=1}^m x_i w_{ki} + b\right) \quad (2.1)$$

2.3.1 Função de Ativação

As funções de ativação são funções não-lineares conectadas ao final da estrutura de um neurônio artificial, definem a saída com base nos dados de entrada e o limiar de ativação.

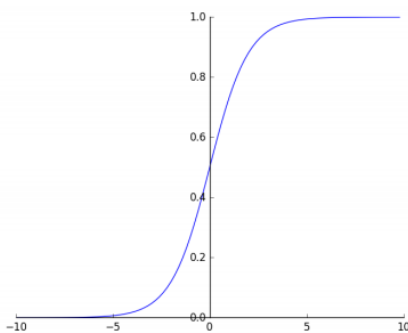


Figura 2.2: Sigmóide

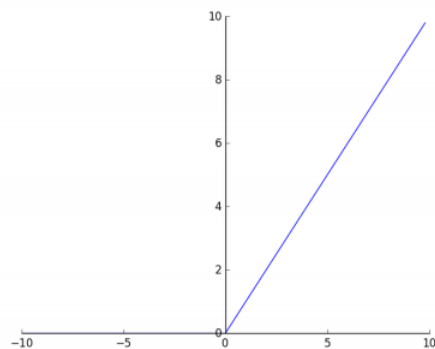


Figura 2.3: ReLU

A função sigmóide, Figura 2.2, é muito utilizada em problemas de classificação, onde tem como saída valores entre 0 e 1, e resulta na probabilidade dos dados de entrada estarem contidos na classe analisada.

A função ReLU, Figura 2.3, é uma função de ativação mais eficiente, pois não faz uso de expoentes, por este motivo é comumente utilizada em camadas ocultas.

2.3.2 Rede Neural Multicamadas

Uma rede com apenas um neurônio, chamada de Perceptron, é capaz de classificar padrões de forma extremamente limitada, pois agrupa apenas dados linearmente separáveis. Para escalar a rede e resolver problemas de maior magnitude, utiliza-se uma arquitetura mais robusta chamada de Rede Neural Multicamadas, ou *Multilayer Perceptron* (MLP).

Uma MLP, Figura 2.4, consiste em um sistema de neurônios simplesmente conectados, distribuídos em camadas, podendo estas serem de entrada, ocultas ou de saída. Uma MLP com mais de uma camada oculta é denominada de Rede Neural Profunda, ou *Deep Artificial Neural Network*.

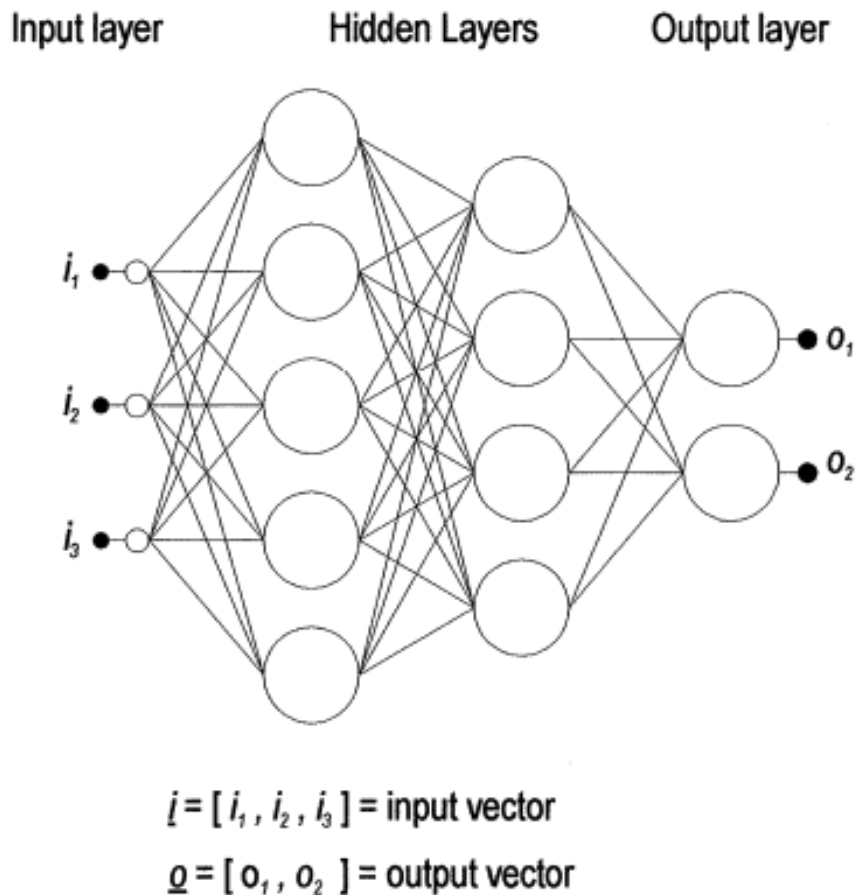


Figura 2.4: MLP com duas camadas ocultas (GARDNER; DORLING, 1998)

MLPs aprendem de maneira supervisionada, sendo assim, é necessário um conjunto de dados de treinamento, que consiste em uma série de vetores de entrada e saída as-

sociados. Durante o treinamento, a saída de uma interação, dado um vetor de entrada, não necessariamente será igual a saída definida pelos dados de treinamento, portanto é necessário calcular-se o erro, definido como a diferença entre a saída desejada e a obtida.

O treinamento usa da magnitude desse erro para determinar em que grau os pesos e as tendências da rede devem ser reajustadas, de maneira a minimizá-lo. Uma vez treinada com um conjunto de dados representativos adequados, a MLP pode generalizar para novos dados de entradas não previamente vistos (GARDNER; DORLING, 1998).

2.4 Aprendizado Profundo

Aprendizado Profundo, ou *Deep Learning* (DL), nasceu da necessidade de se extrair informações abstratas e de alto nível de um conjunto de dados brutos, visto que MLPs tradicionais têm dificuldade em generalizar tarefas de AI, que necessitam de uma maior quantidade de informações não observáveis para serem solucionadas, como reconhecimento de fala e objetos (GOODFELLOW; BENGIO, 2015).

DL tem a capacidade de resolver esse problema de representação, pois consegue expressar dados em termos de outras formas mais simples, permitindo que o computador crie conceitos complexos a partir de abstrações mais simples, como demonstrado na Figura 2.5. Este modelo representa a imagem de uma pessoa através da combinação de dados mais simples, como cantos e contornos, definidos em termos de arestas.

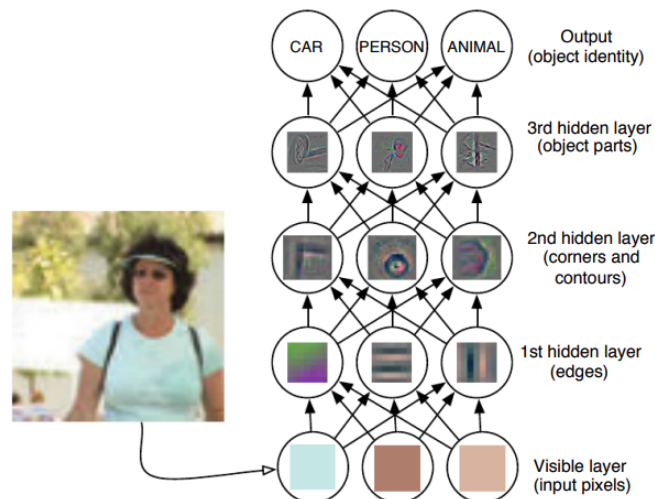


Figura 2.5: Modelo DL extraindo conceitos simples e abstratos (GOODFELLOW; BENGIO, 2015)

DL é um modelo de MLP com mais de duas camadas ocultas e técnicas de treinamento mais eficientes. Visto que o aumento do número de camadas causa um crescimento

considerável no número de parâmetros que devem ser ajustados, dois fatores são fundamentais para sua viabilidade: alta capacidade de processamento e um extenso banco de dados. Portanto, esta técnica de aprendizado só se tornou viável recentemente com o avanço dos *hardwares* que permitem uma maior captação e processamento dos dados (GOODFELLOW; BENGIO, 2015).

2.5 Redes Neurais Convolucionais

Rede Neural Convolutiva, ou *Convolutional Neural Network* (CNN), é um tipo especializado de rede neural profunda para processamento de dados que contém uma topologia em *grid*, onde, ao invés de simplesmente aplicar uma multiplicação matricial em suas camadas ocultas, aplica-se uma operação de convolução.

As camadas de convolução trazem características muito importantes para a relevância dessa arquitetura, como invariância da escala, translação e outras transformações. Essas conseguem reconhecer padrões de forma mais robusta, permitindo a aplicação de técnicas de aumento de dados, ou *Data Augmentation*, e utilizar uma maior variedade e quantidade de dados de treinamento (GOODFELLOW; BENGIO, 2015).

Uma CNN, Figura 2.6, contém diversas camadas que transformam uma entrada com filtros de convolução de pequena extensão. Sua arquitetura é composta de 3 principais camadas: convolução, *pooling* e camadas totalmente conectadas.

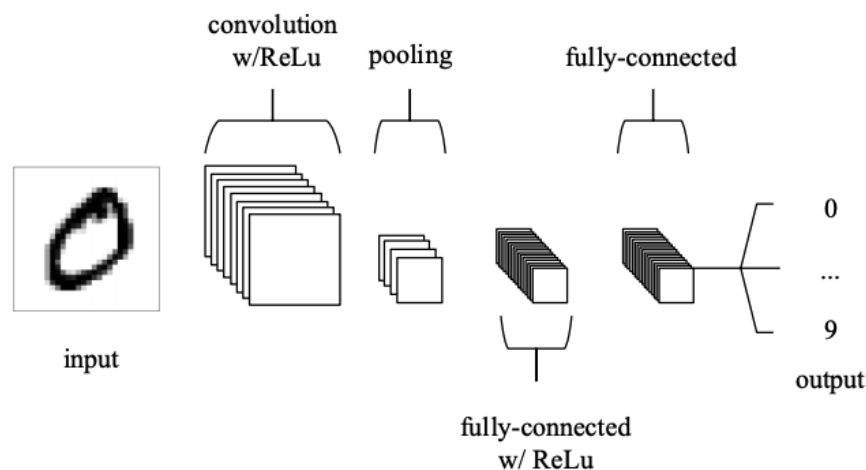


Figura 2.6: Arquitetura de uma Rede Neural Convolutiva (GOODFELLOW; BENGIO, 2015)

2.5.1 Camada de convolução

Camada de convolução, ou *Convolutional Layer* (CL), é um conjunto de filtros não lineares (*kernels*) de baixa dimensionalidade, que percorrem sequencialmente toda profundidade dos dados de entrada e produzem assim matrizes chamadas de mapas de características, ou *feature maps*. A combinação desses *feature maps* gerados pelos diferentes filtros representam a saída da CL (O'SHEA; NASH, 2015).

Durante o processo de treinamento, nas camadas mais rasas, esses filtros são ajustados para que sejam ativados na presença de características e padrões visuais mais simples, como bordas ou aglomerado de *pixels* de mesmo valor, enquanto nas camadas mais profundas, contém informações de padrões mais complexos, como diferenciar partes de objetos.

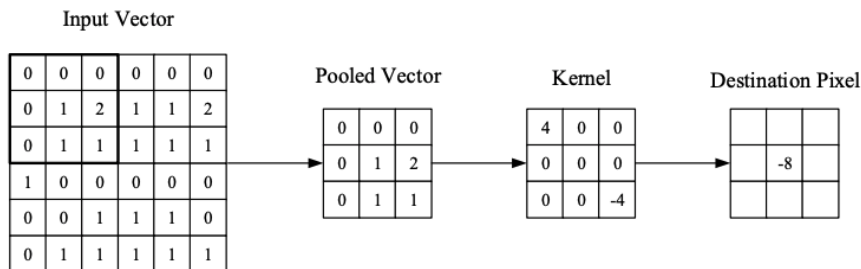


Figura 2.7: Representação visual da camada de convolução (O'SHEA; NASH, 2015)

Podemos representar a camada de convolução matematicamente como uma operação de convolução discreta em um *array* multidimensional, neste caso, imagens bidimensionais, assim como descrito na Equação (2.2) (GOODFELLOW; BENGIO, 2015).

$$s[i, j] = (I * K)[i, j] = \sum_m \sum_n I[i - m, j - n] K[m, n] \quad (2.2)$$

Passo (*Stride*)

Conforme explicado, os filtros de convolução percorrem sequencialmente toda matriz, isso ocorre em passos, passando de *pixel* para *pixel* em uma imagem. Quando o passo é igual a 1, a altura e largura da camada de saída será igual a de entrada; quando é maior que 1, reduz a quantidade de sobreposições e assim produz uma saída de menores dimensões espaciais (O'SHEA; NASH, 2015).

Preenchimento (*Padding*)

Durante o deslizamento dos filtros, é necessário lidar com as bordas e para isso é utilizada a técnica *same padding* ou *zero padding*, em que as fronteiras da imagem são

preenchidas com zeros, de modo a controlar a altura e largura da camada de saída (Figura 2.8)

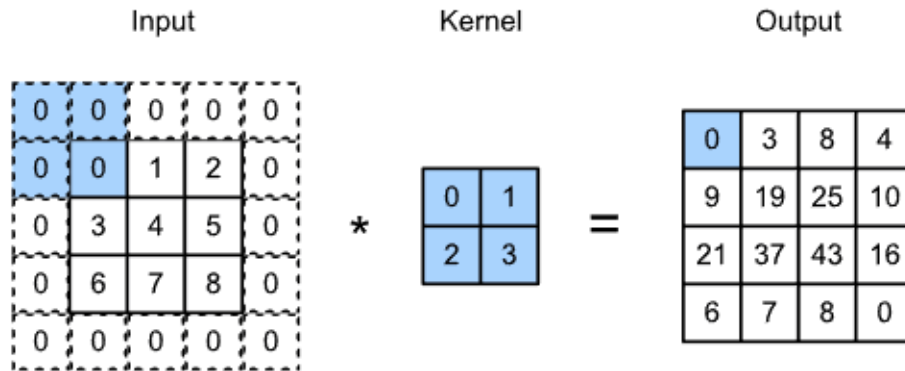


Figura 2.8: Preenchimento para um filtro 2×2 (ZHANG; LIPTON, 2019)

O tamanho deste preenchimento é determinado pela Equação (2.3), onde K é o tamanho do filtro.

$$P = \frac{K - 1}{2} \quad (2.3)$$

É importante notar que ao usar esta técnica, a dimensão espacial da saída é alterada, seguindo a fórmula da Equação (2.4), onde V é o volume da entrada, K é o tamanho do filtro, P é o preenchimento e S é o passo do filtro (O'SHEA; NASH, 2015).

$$O = \frac{(V - K) + 2P}{S + 1} \quad (2.4)$$

2.5.2 Camada de *pooling*

A camada de *pooling* tem como objetivo reduzir gradualmente a dimensionalidade da representação, e assim, reduzir o número de parâmetros, juntamente com a complexidade computacional do modelo, evitando *overfitting* (O'SHEA; NASH, 2015). O método mais comum, *Max Pooling*, consiste em reduzir a dimensão das camadas utilizando o valor máximo de cada região baseado em um filtro de tamanho $N \times N$. Desta forma, o método elimina valores desprezíveis, criando uma invariância a pequenas mudanças e distorções locais (GOODFELLOW; BENGIO, 2015).

2.5.3 Camada Totalmente Conectada

Camada Totalmente Conectada, ou *Fully Connected Layers* (FC), são camadas MLP tradicionais, assim como demonstrado na Figura 2.4, onde todos os neurônios estão di-

retamente conectados as camadas adjacentes. Nesta camada, as características extraídas nas camadas convolucionais são usadas no processo de classificação.

2.6 Segmentação Semântica

Redes convolucionais são tipicamente utilizadas em tarefas de classificação, em que a saída para uma imagem é um rótulo de classe única. No entanto, em muitas tarefas visuais, especialmente no processamento de imagens biomédicas, a saída desejada deve incluir a localização, ou seja, um rótulo de classe deve ser atribuído a cada *pixel*.

O objetivo da segmentação semântica é rotular cada *pixel* de uma imagem com uma classe que corresponda com o que está sendo representado. Como são feitas previsões para cada *pixel* da imagem, essa tarefa é comumente chamada de previsão densa. Uma questão importante a ser observada é que o sistema não separa categorias da mesma classe, ou seja, se existirem dois objetos diferentes da mesma classe em sua imagem de entrada, o mapa de segmentação não os distingue inerentemente como objetos separados.

Para atingir a segmentação, é possível substituir a camada totalmente conectada por convoluções, gerando uma rede neural totalmente convolucional, ou *Fully Convolutional Neural Network* (FCNN). Desta maneira o modelo pode obter imagens de entrada maiores do que foi treinado e produzir um mapa de probabilidade, ao invés de uma saída para um único *pixel*. Devido as camadas de *pooling*, é possível obter uma saída com uma resolução muito menor do que a de entrada (LONG; SHELHAMER; DARRELL, 2015).

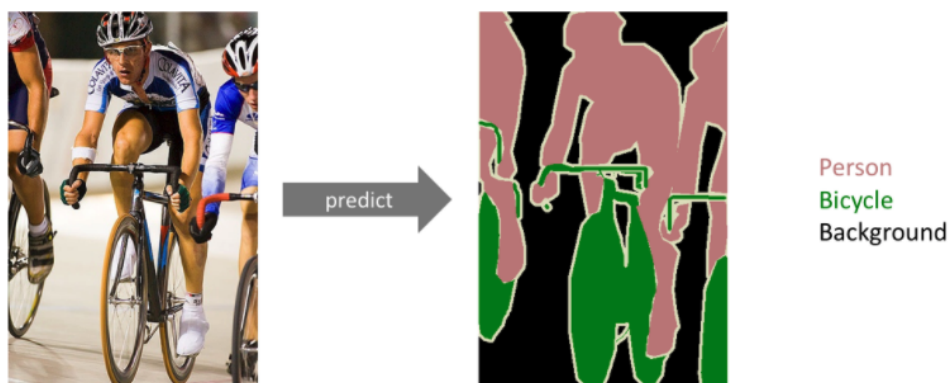


Figura 2.9: Segmentação Semântica (LAMBA, 2019)

Capítulo 3

Metodologia

Neste Capítulo são descritos os materiais e métodos utilizados para a solução do problema apresentado. A metodologia consiste no treinamento e validação de uma Rede Neural Totalmente Convolutiva para segmentação de lesões pulmonares a partir de tomografias torácicas. Para isto o experimento foi dividido em três principais etapas: pré-processamento, aumento dos dados e concepção da arquitetura do modelo.

3.1 Detalhes de implementação

A rede neural convolutiva e todo o pré-processamento dos dados foram implementados em Python, utilizando bibliotecas como TensorFlow, Keras, NumPy e OpenCV, no um sistema operacional Debian GNU/Linux.

Todas as execuções foram feitas em uma máquina com CPU Intel(R) Core(TM) i5-7400 @ 3.00 GHz, GPU NVIDIA GeForce GTX 1060 6 GB Vram e 16 GB 2400 MHz de memória RAM. A fim de diminuir o tempo computacional de execução e otimizar as tarefas, fez-se o uso de processamento paralelo através da GPU, utilizando as respectivas bibliotecas, TensorFlow-GPU, CUDA e cuDNN.

Todo código fonte necessário para reproduzir os experimentos pode ser encontrado em um repositório aberto na plataforma GitHub em (ARAUJO, 2021).

3.2 Banco de dados

As amostras utilizadas foram retiradas de diversas fontes de dados abertos disponibilizados em uma ação mundial para ajudar pesquisadores a combater a nova pandemia.

Um total de 13347 imagens foram utilizadas, sendo subdivididas igualmente entre 3 categorias: tomografias torácicas, máscaras com recorte pulmonar e máscaras das lesões.

De cada conjunto de 4449 imagens, 3520 foram coletados através dos sites (JUN et al., 2020) e (GLICK, 2020), enquanto 929 foram retirados de (MEDSEG, 2020).

3.3 Pré-processamento

O pré-processamento de dados é uma etapa integral do aprendizado de máquina, pois a qualidade dos dados e informações úteis que podem ser derivadas deles afetam diretamente a capacidade de aprendizado do modelo. Dessa maneira, é de suma importância trazê-los a um estado que o algoritmo possa facilmente analisá-los.

3.3.1 Conversão de formato

A primeira etapa do pré-processamento consiste em converter as imagens de entrada para um formato que se adéque aos tensores de entrada da rede convolucional. Inicialmente, as tomografias se encontram comprimidas no formato de arquivo *Nifti* (.nii), com um cabeçalho contendo informações sobre a orientação espacial e dimensional da imagem, evitando possíveis ambiguidades no estudo das tomografias (LAROBIINA; MICHELE; MURINO, 2014).

As imagens foram transformados para o formato *numpy array* do tipo *float64* e redimensionados para obtenção de imagens de 512 de largura por 512 de altura, apresentando apenas um canal de cor, contendo valores de intensidade de 0 a 255. Após a transformação, é possível visualizar as imagens, mostradas na Figura 3.1.



Figura 3.1: Dados de entrada: Tomografia, Máscara Pulmonar e Máscara das lesões

3.3.2 Normalização

A normalização dos dados de entrada é um método muito comum e importante para evitar problemas de estabilidade numérica, uma vez que redes neurais com valores de en-

trada muito discrepantes, não conseguem obter bons resultados de generalização (IOFFE; SZEGEDY, 2015).

A fim de evitar esse problema, foram normalizados os dados de entrada utilizando-se o método *Min-Max*, descrito matematicamente pela Equação (3.1):

$$x_{normalized} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (3.1)$$

onde x representa o *pixel*, $\min(x)$ representa o valor do menor *pixel* e $\max(x)$ representa o valor do maior *pixel*. Desta maneira, todos os valores foram transformados para um intervalo de $[0,1]$.

3.3.3 Equalização de histograma

Como a base de dados utilizada trata-se de tomografias computadorizadas da região torácica, grande parte do espectro do histograma se encontra nos menores valores de intensidade. A fim de se obter contrastes superiores, bem como melhor distribuição de intensidade e identificação de lesões pulmonares, aplicou-se uma equalização de histograma.

Foi utilizada técnica de Equalização de Histograma Adaptativo Limitado por Contraste, ou *Contrast Limited Adaptive Histogram Equalization (CLAHE)*, que consiste em dividir a imagem em pequenos blocos $N \times N$, e equalizá-los individualmente. Para evitar realçar possíveis ruídos, foi definido um limiar máximo, que caso fosse ultrapassado, o *pixel* seria recortado e distribuído uniformemente pelo local (PISANO et al., 1998).

Para obter os resultados mostrados na Figura 3.3, utilizou-se blocos 8×8 com um limiar *clipLimit* de valor 3.

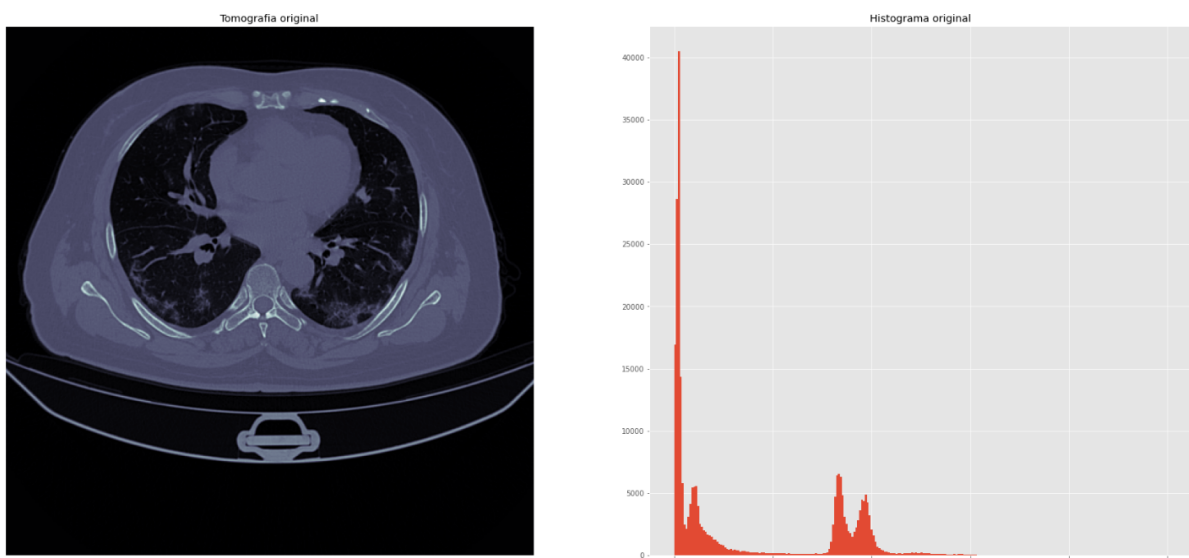


Figura 3.2: Tomografia original e seu respectivo histograma

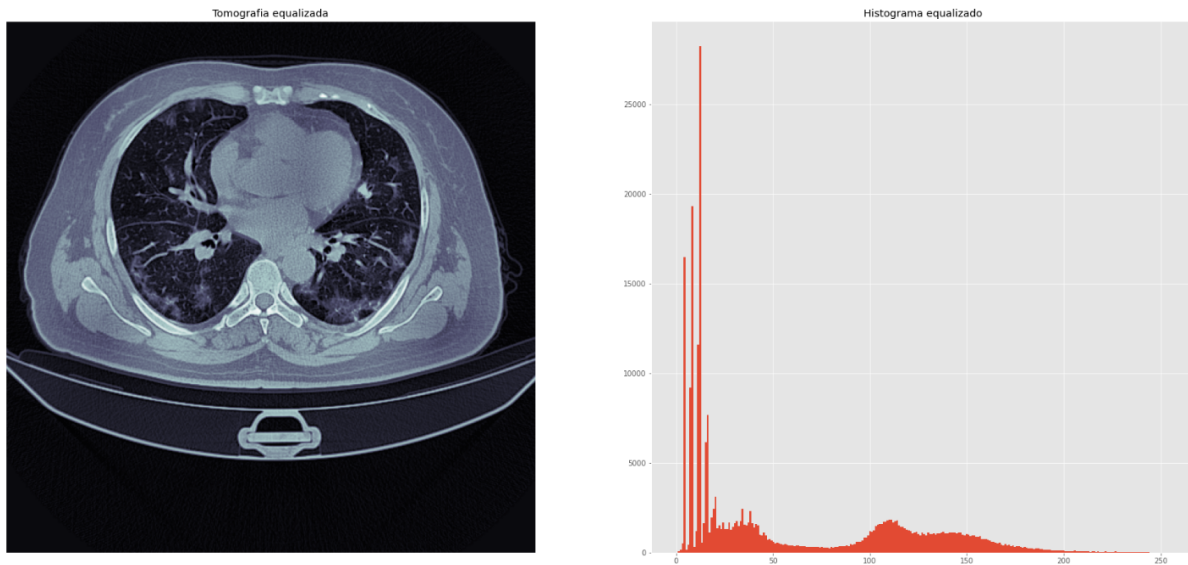


Figura 3.3: Tomografia equalizada e seu respectivo histograma

3.3.4 Recorte pulmonar

Com o objetivo de remover ruídos e informações desnecessárias para o treinamento, foi realizado um recorte nas tomografias originais para que a imagem englobasse apenas a região pulmonar, removendo qualquer tecido externo que possa afetar a generalização da rede.

Para atingir esse objetivo, uma série de operações morfológicas foram aplicadas sobre a máscara pulmonar para encontrar o menor retângulo possível que incluísse toda sua área, sendo elas:

1. Aplicar um filtro Gaussiano de tamanho 5×5 , para remover possíveis ruídos existentes e suavizar as bordas da imagem;
2. Binarizar a imagem com um limiar de valor 50, assim *pixels* menores que 50 são definidos como 0, e os maiores como 1. Como contornos podem ser explicados simplesmente como uma curva que une todos os pontos contínuos de mesma intensidade, esta etapa ajuda no cálculo do mesmo;
3. Calcular os contornos presentes na imagem resultante;
4. Após a obtenção dos contornos do pulmão, para destacar a região de interesse, calcular e desenhar um retângulo aproximado ao redor da imagem, que limite e inclua toda área pulmonar. Caso a tomografia não contenha nenhuma área pulmonar, é removida do conjunto de dados utilizados;

5. Por fim, recortar os dados de entrada com base no retângulo encontrado no passo anterior.

Os resultados de todos os passos descritos anteriormente podem ser observados em ordem de execução nas Figuras 3.4 e 3.5. Após todas as operações morfológicas aplicadas e o recorte dos dados de entrada, esses foram redimensionados para um tamanho de 256×256 de altura e largura.

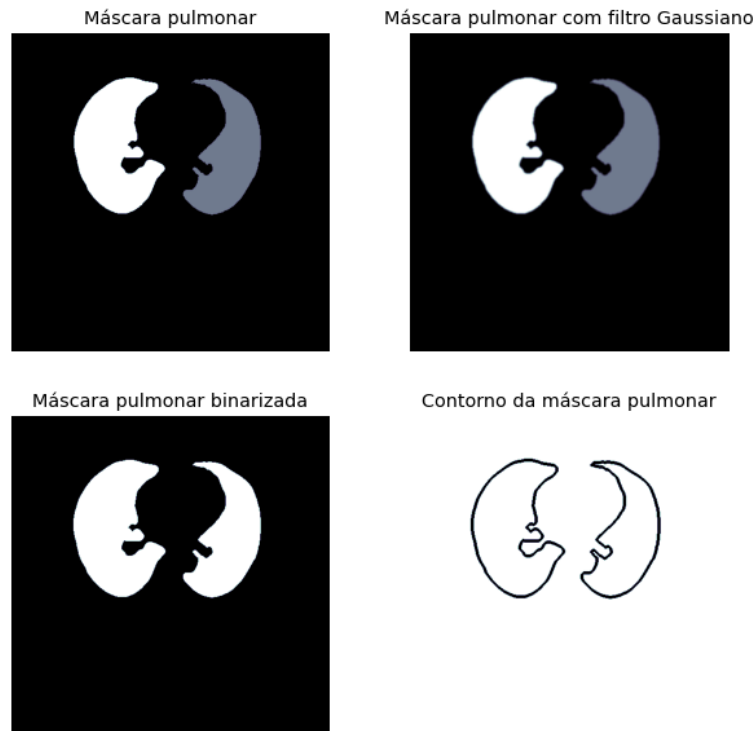


Figura 3.4: Processo de cálculo do contorno pulmonar

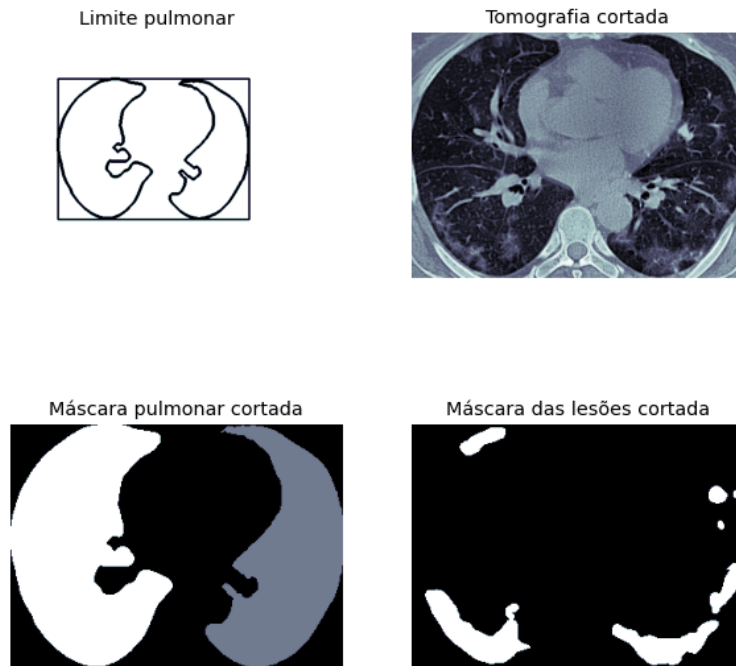


Figura 3.5: Limite pulmonar calculado e imagens cortadas

3.4 Aumento de dados

Aumento de dados, ou *Data Augmentation* é uma técnica utilizada para gerar novos exemplares de dados de treinamento a fim de aumentar a generalidade do modelo e reduzir *overfitting*.

Uma rede neural convolucional pode classificar e segmentar dados de maneira robusta, mesmo se colocados em orientações diferentes, devido a sua invariância de escala, translação e outras transformações.

Visto a alta quantidade de parâmetros utilizados por uma FCNN, é necessário mostrar uma quantidade proporcional de exemplos para se obter uma boa generalização e um bom desempenho. Porém, a baixa quantidade de amostras de entrada disponíveis em acervos públicos, devido ao caráter confidencial das informações dos pacientes na medicina, exige o aumento dos dados para obtenção de resultados satisfatórios.

Neste trabalho, a fim de aumentar a quantidade de dados, foram aplicadas as transformações que se seguem:

- 50% de chance de virar a imagem verticalmente;
- 50% de chance de virar a imagem horizontalmente;

- Chance aleatória de redimensionar o *zoom* da imagem, ampliando ou reduzindo por um fator de 20%;
- Chance aleatória de transladar a imagem no eixo x ou y em 20% de sua largura/altura (no canto inferior direito);
- Chance aleatória de rotacionar em 15° para esquerda ou direita.
- Chance aleatória de cortar as bordas da imagem em 15° para esquerda ou direita.

O resultado dessas transformações podem ser observados na Figura 3.6, onde para cada dado de entrada, foi gerado seu respectivo par transformado, dobrando a quantidade de exemplares de treinamento.

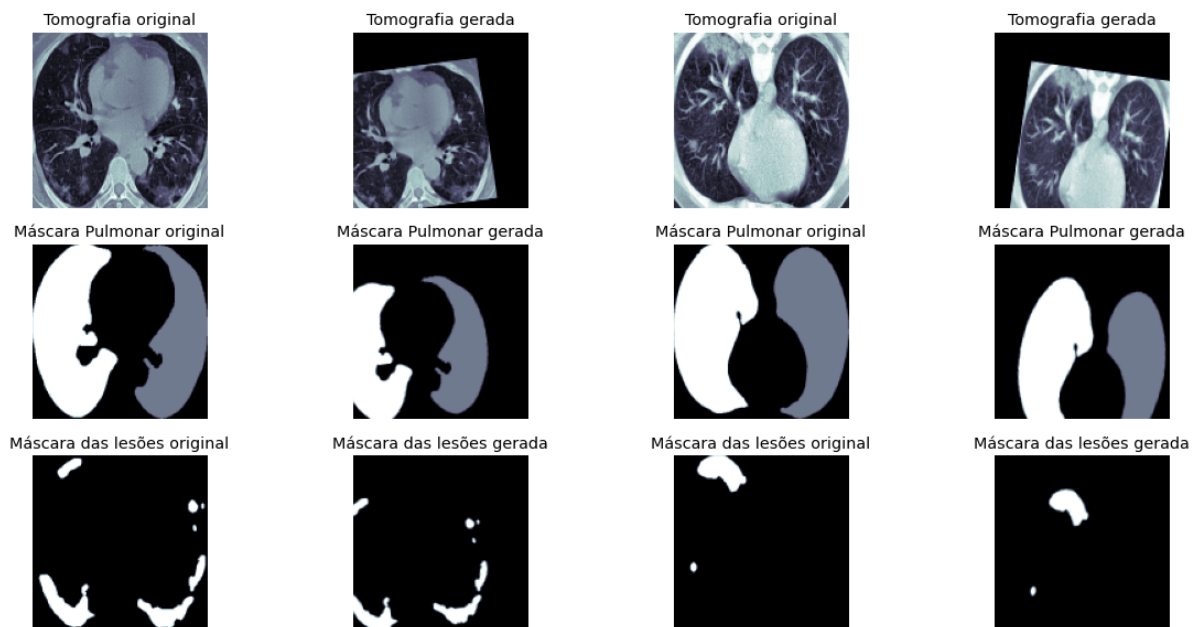


Figura 3.6: Resultado do aumento de dados aplicado em duas imagens

3.5 Dados de treinamento, validação e teste

A etapa de pré-processamento resultou em um total de 3750 imagens de treinamento, visto que houve a exclusão de alguns dados por não conterem informações relevantes ao estudo, como tomografias sem nenhuma área pulmonar. Dentre essas 3750 tomografias, aproximadamente 60% contém lesões e 40% não apresentam nenhum acometimento relevante ao estudo.

Com o aumento de dados, o número de imagens foi duplicado, resultando em 7500 amostras. Logo em seguida, foram feitas separações aleatórias dos conjuntos em: 80% das imagens para treinamento, 10% para validação e 10% para teste.

3.6 Arquitetura

A rede neural convolucional escolhida como base para a realização dos experimentos foi a UNet, devido a sua ampla utilização em tarefas de segmentação contendo baixa quantidades de dados de treinamento.

3.6.1 UNet

O treinamento bem-sucedido de redes profundas requer milhares de exemplos de treinamento. A UNet, Figura 3.7, apresenta uma rede e estratégia de treinamento que se baseia no forte uso de aumento de dados para usar as amostras disponíveis de forma mais eficiente.

Montada a partir de uma FCNN, a arquitetura foi modificada e estendida de modo a necessitar de pouquíssimas imagens de treinamento e produzindo segmentações mais precisas. O objetivo principal é complementar uma rede de contração usual (*downsampling*) por camadas sucessivas, onde os operadores de *pooling* são substituídos por operadores de *upsampling*, e conseqüentemente, há o aumento da resolução da saída.

Uma modificação importante na arquitetura, é que na parte de *upsampling* existe um grande número de *feature channels*, que permitem que a rede propague informações de contexto para camadas de resolução mais alta. Como consequência, o caminho de expansão é mais ou menos simétrico ao caminho de contração e produz uma arquitetura em forma de U.

A rede não possui nenhuma camada totalmente conectada e usa apenas a parte válida de cada convolução, ou seja, o mapa de segmentação contém apenas os *pixels* para os quais o contexto completo está disponível na imagem de entrada. Essa estratégia permite a segmentação contínua de imagens arbitrariamente grandes por meio da sobreposição de blocos (RONNEBERGER; FISCHER; BROX, 2015).

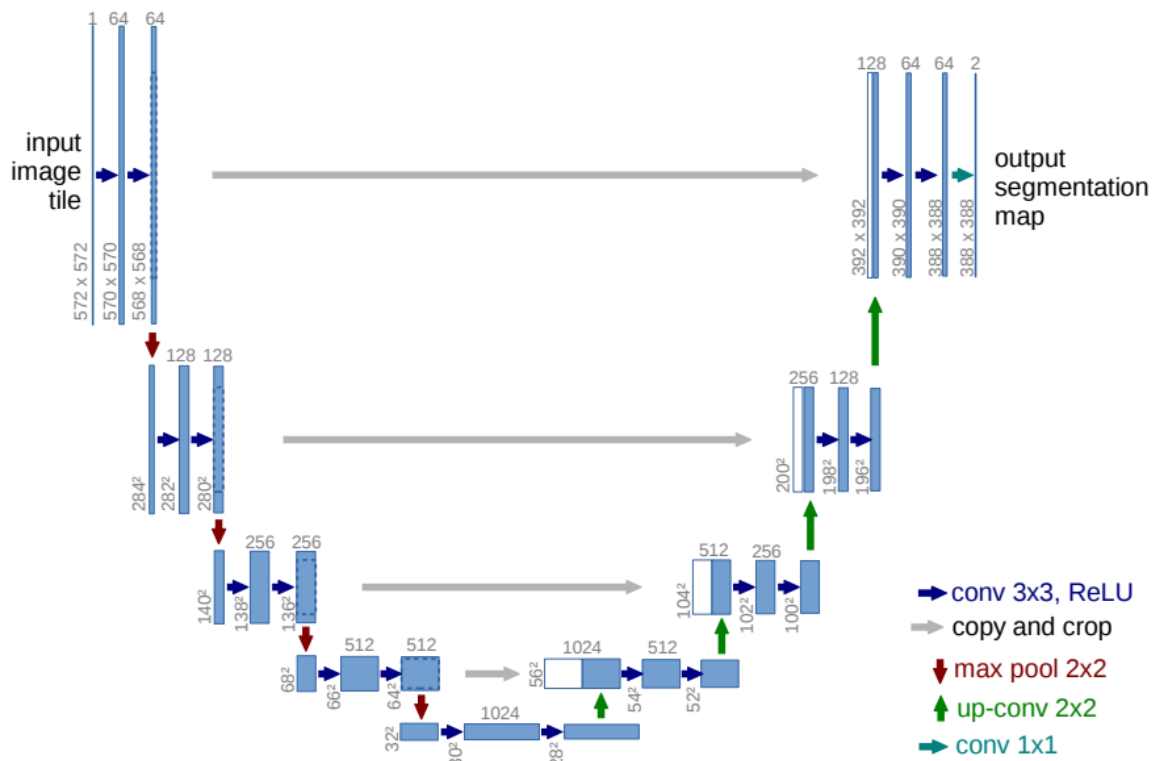


Figura 3.7: Arquitetura UNet clássica (RONNEBERGER; FISCHER; BROX, 2015)

3.6.2 UNet Modificada

De acordo com (KUCHANA et al., 2021), a utilização de um esqueleto UNet, em conjunto com mudanças em hiper-parâmetros, número de filtros e mudanças nas camadas arquiteturais, resultam em uma melhor acurácia, quando comparadas com a arquitetura tradicional.

(KUCHANA et al., 2021) propõe uma arquitetura baseada na substituição dos blocos básicos de convolução e de concatenação por *spatial pyramid pooling* e *attention gates*, podendo essa ser visualizada na Figura 3.8. Apesar dos bons resultados apresentados, o modelo continua relativamente complexo, com um alto número de parâmetros treináveis, necessitando de uma alta capacidade computacional e tempo de treinamento.

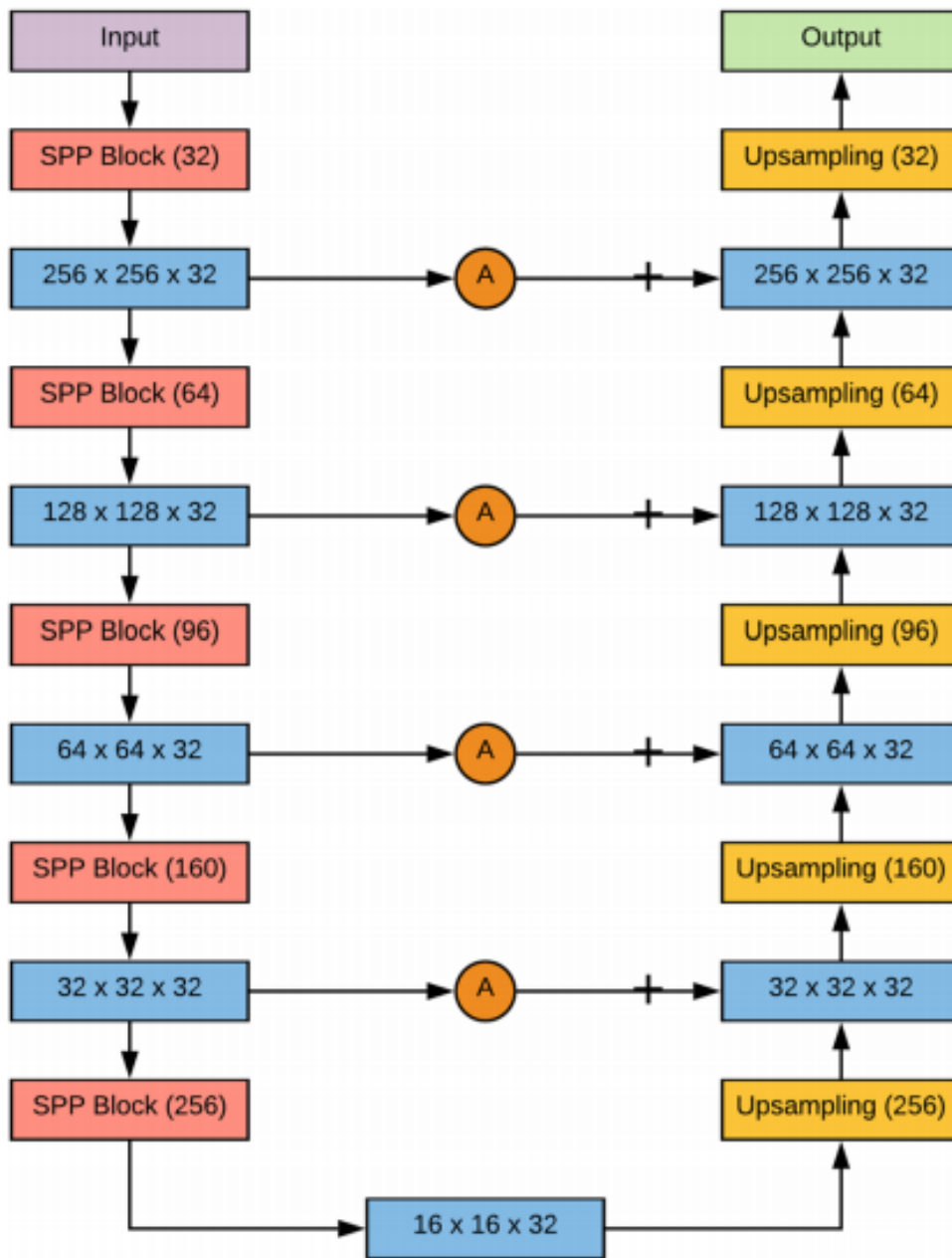


Figura 3.8: Arquitetura UNet proposta em (KUCHANA et al., 2021)

Desta maneira, foram feitas modificações de maneira a tornar o modelo mais simples, menos complexos e de mais fácil treinamento. Estas modificações podem ser observadas nas Figuras 3.9 e 3.10, onde são utilizados dois modelos em série para completar as tarefas de segmentação da área pulmonar e das lesões.

Após todas as modificações realizadas na rede, a quantidade de parâmetros treináveis diminuiu de 34139778 para 5919682 em relação a rede UNet tradicional.

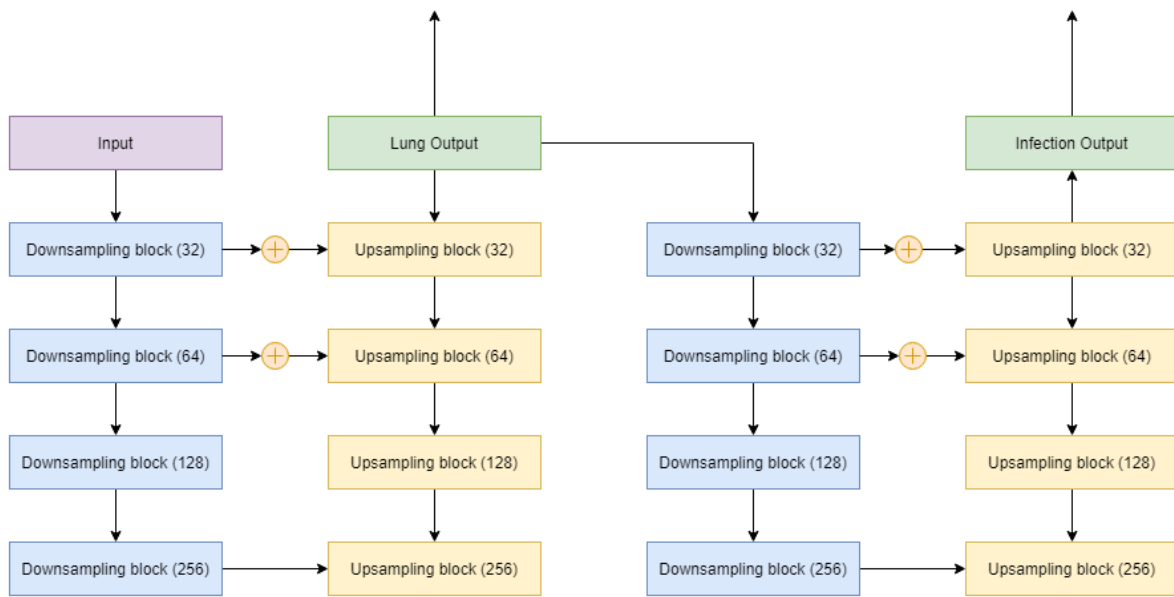


Figura 3.9: Arquitetura UNet Modificada

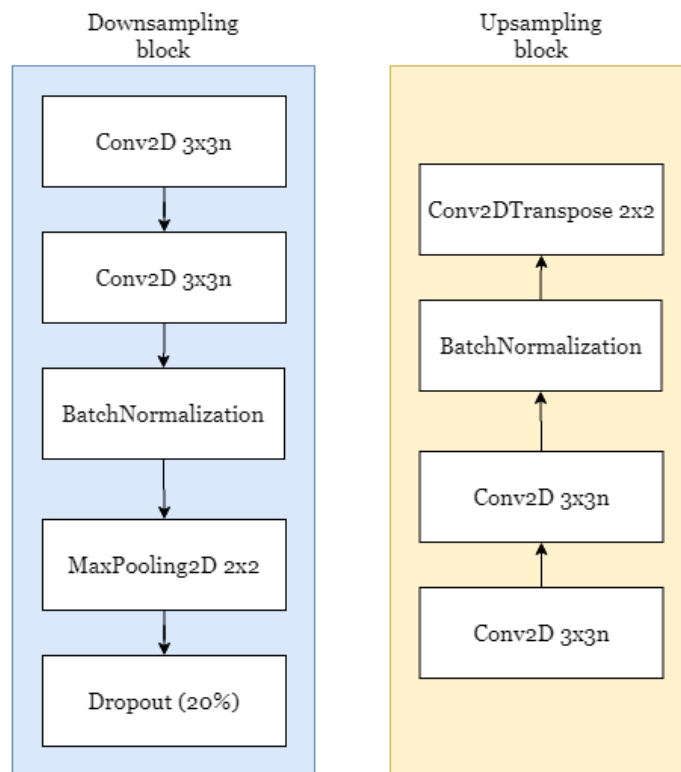


Figura 3.10: Blocos detalhados

Primeiramente foi diminuída a quantidade de camadas de convolução e de concatenações realizadas, e adicionada camadas de normalização e de *dropout* como técnicas de

regularização.

A camada de normalização, chamada de *Batch Normalization*, aumenta a velocidade de treinamento e reduz a mudança de covariável interna, garantindo que a distribuição de cada camada seja em torno da mesma média e do mesmo desvio padrão. A camada de *Dropout* reduz a aprendizagem interdependente entre os neurônios e incentiva a aprendizagem de recursos mais robustos que são úteis em conjunto com ligações aleatórias, evitando assim *overfitting* do modelo.

As camadas de convolução seguiram o modelo padrão de filtros com valores exponenciais de 2, tendo os valores respectivos de 32, 64, 128, 256 e ao longo dos blocos de *Downsampling* e *Upsampling*, a função de ativação utilizada foi a *ReLU*, enquanto as camadas de saída utilizaram a função sigmoide.

No Capítulo 5 serão apresentados os resultados e desempenhos de cada uma das topologias mencionadas.

Capítulo 4

Resultados

Neste capítulo são apresentadas de forma gráfica os resultados obtidos do treinamento e validação das redes, incluindo valores de performance. É também apresentado resultados visuais de saída do modelo proposto, mostrando como na prática pode auxiliar um profissional de saúde.

4.1 Treinamento e validação

Para o treinamento de ambas as topologias (UNet Clássica e UNet Modificada), foram executados os passos de pré-processamento, aumento de dados e separação do conjunto de dados, descritos no Capítulo 3. Todos os treinamentos foram realizados utilizando 100 *epochs* e *batch size* de tamanho 16. *Callbacks* de redução da taxa de aprendizado em platôs e parada precoce foram aplicados para evitar *overfitting* do modelo ao longo das épocas de treinamento.

Foram plotados gráficos que demonstram os resultados para cada época da acurácia e do erro das segmentações pulmonares e das lesões, para ambos os conjuntos de treinamento e validação. Nestas figuras, é possível verificar de modo qualitativo a influência da taxa de aprendizagem e o desempenho das topologias ao longo de seu treinamento.

4.1.1 UNet Clássica

O primeiro experimento foi realizado utilizando a arquitetura UNet clássica descrita na Figura 3.7. As Figuras 4.1, 4.2 e Tabela 4.1 mostram o desempenho do modelo para os dados de treinamento e teste respectivamente.

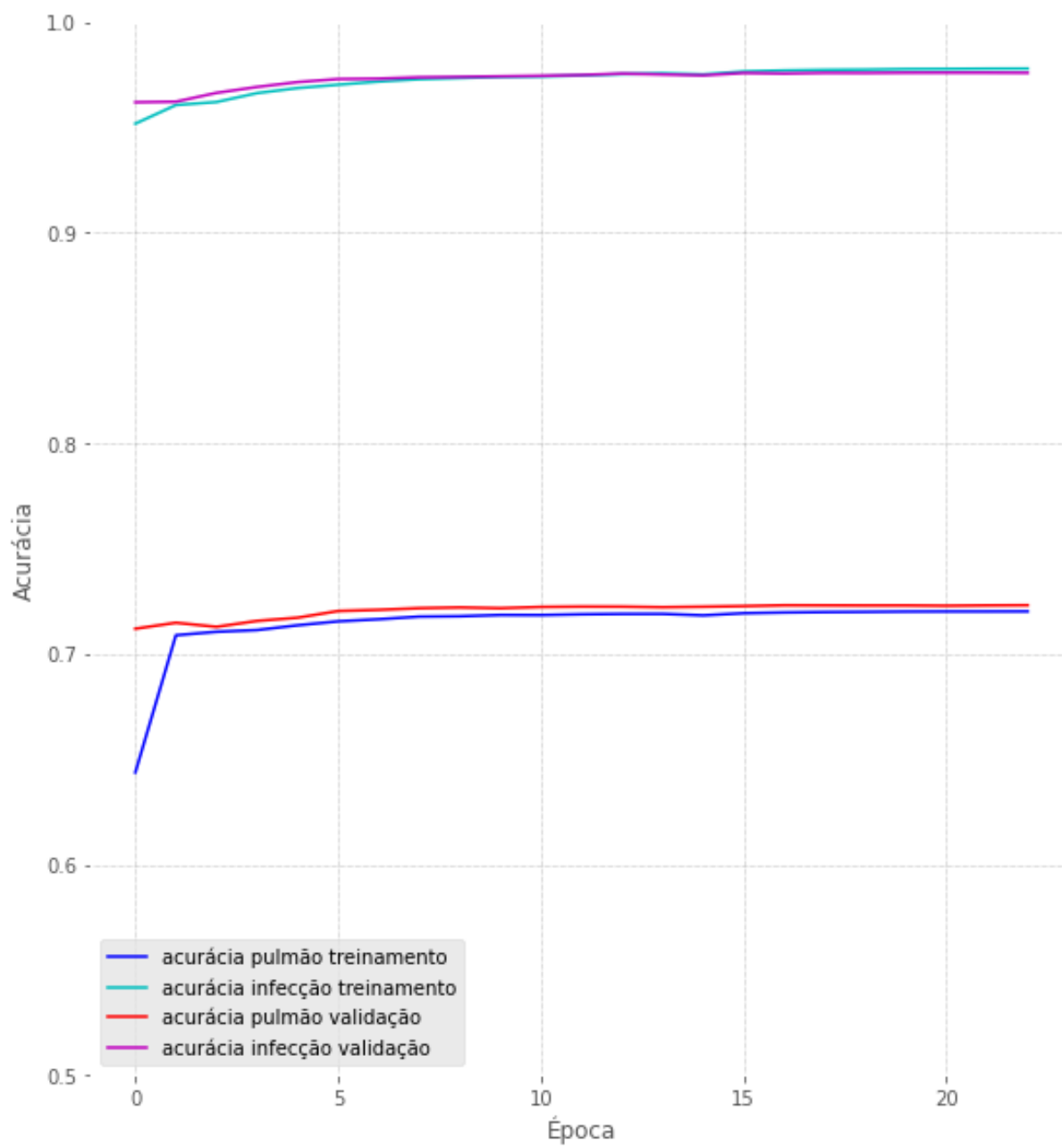


Figura 4.1: Gráfico de acurácia da rede UNet clássica

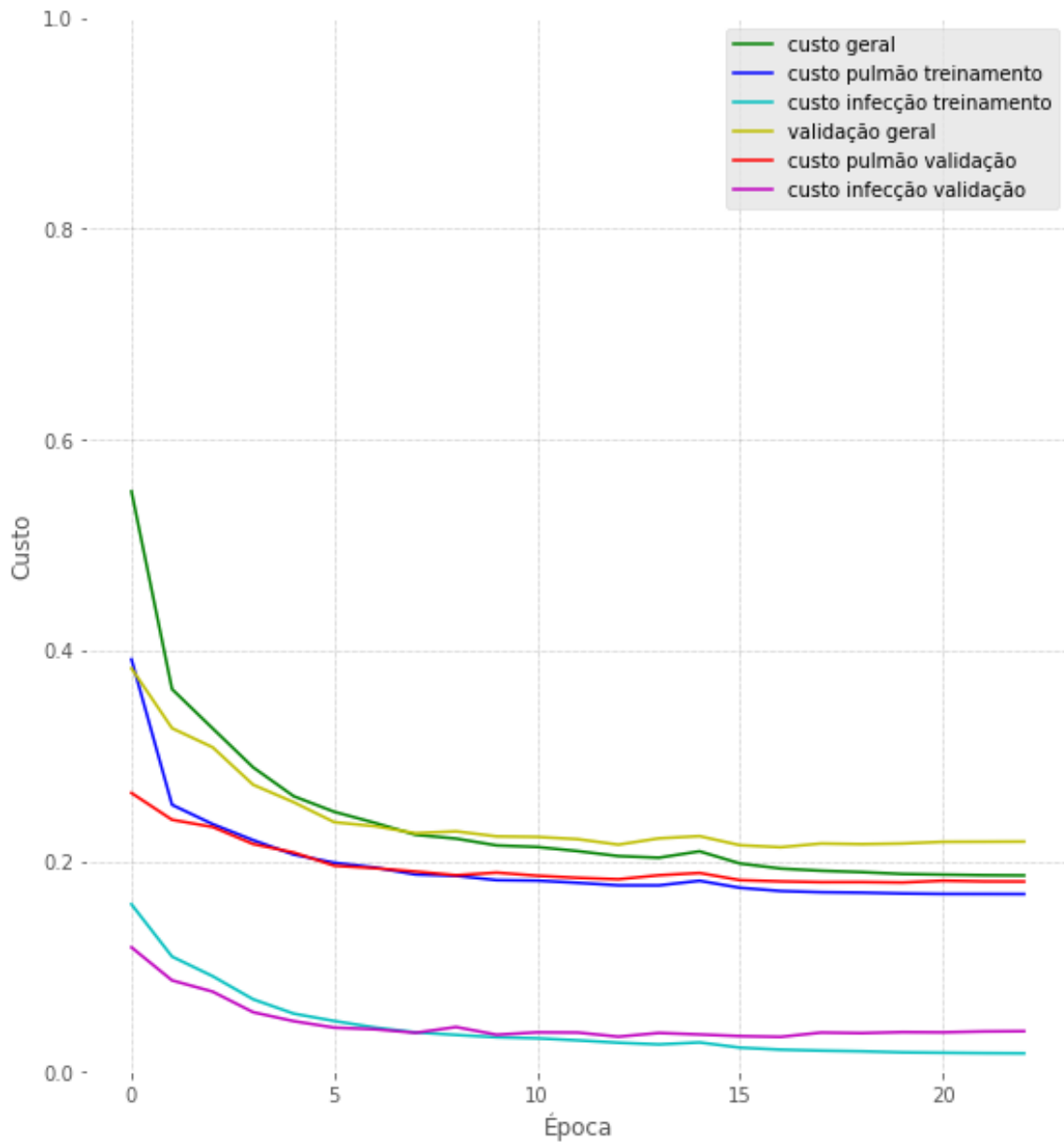


Figura 4.2: Gráfico de custo da rede UNet clássica

Tabela 4.1: Resultados do conjunto de teste da UNet clássica.

	Pulmão	Lesões
Acurácia (%)	70.62	95.19
Custo (%)	20.14	4.29

Assim como esperado, esta arquitetura tem uma boa performance para problemas de segmentação semântica onde se têm uma baixa quantidade de dados de treinamento.

É possível observar que não ocorreu *overfitting* ou *underfitting*, resultando em uma boa generalização e conseqüentemente um bom resultado com as amostras de teste.

4.1.2 UNet Modificada

O segundo experimento foi realizado utilizando a arquitetura UNet Modificada descrita na Figura 3.9. As Figuras 4.3, 4.4 e Tabela 4.2 mostram o desempenho do modelo para os dados de treinamento e teste respectivamente.

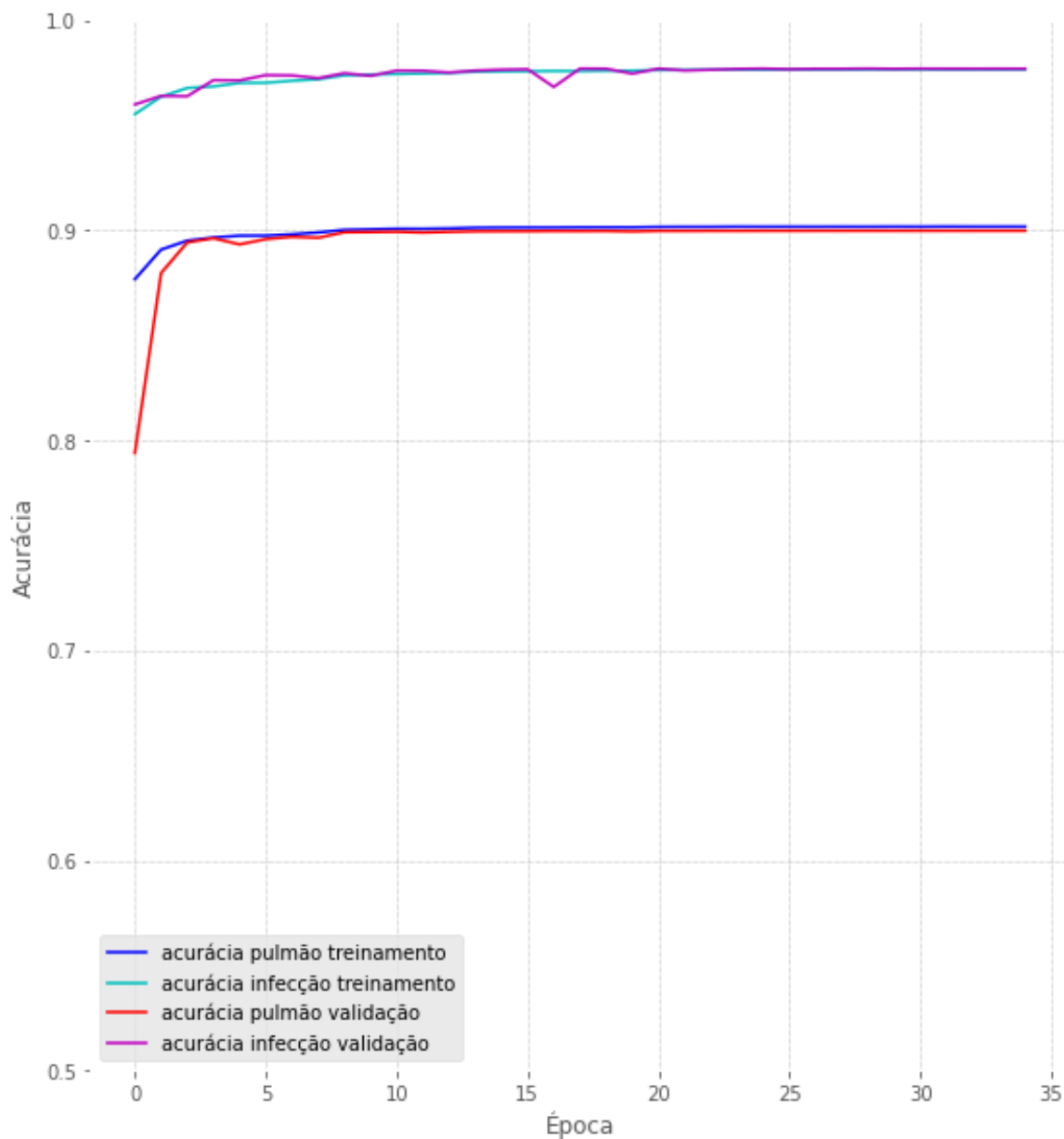


Figura 4.3: Gráfico de acurácia da rede UNet modificada

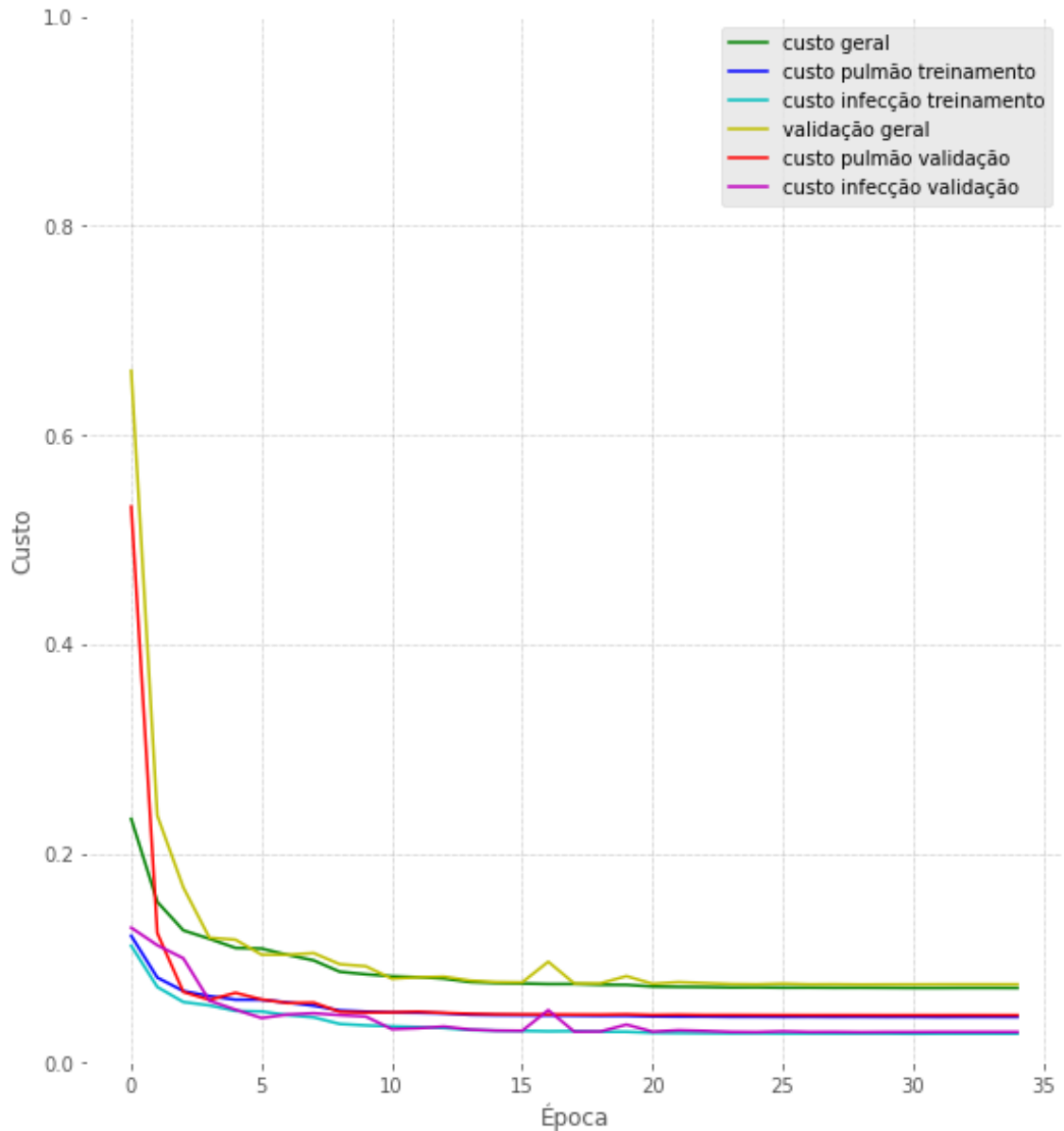


Figura 4.4: Gráfico de custo da rede UNet modificada

Tabela 4.2: Resultados do conjunto de teste da UNet modificada.

	Pulmão	Lesões
Acurácia (%)	71.16	98.01
Custo (%)	18.37	3.31

Ocorreu uma leve melhoria na performance quando comparada com a arquitetura clássica, percebe-se que o modelo se adequou melhor ao treinamento, seguindo um padrão logarítmico ao longo das épocas. Assim como no experimento 1, não verificou-se *overfitting* ou *underfitting* no modelo.

4.2 Resultados visuais

A fim de conferir a corretude dos valores de performance, foram gerados exemplares de saída e comparados com suas respectivas máscaras de treinamento. É possível verificar a proximidade das máscaras de treinamento com as previstas pelo modelo na Figura 4.5.

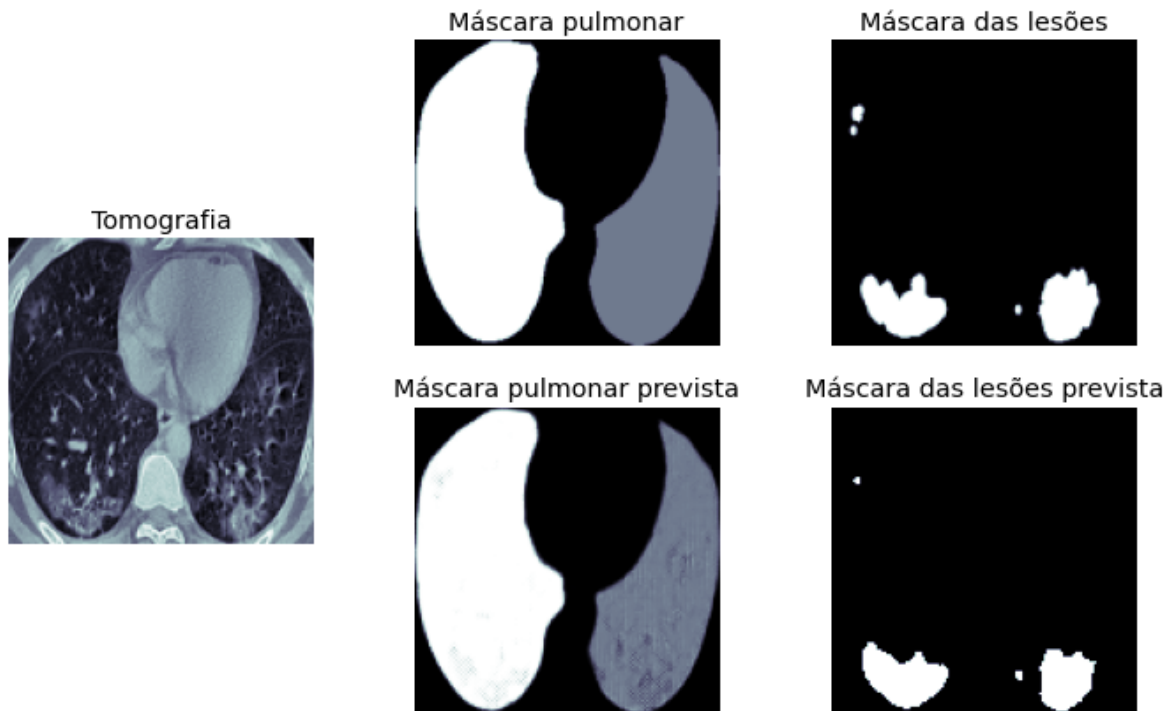


Figura 4.5: Saídas previstas pelo modelo: Pulmão e Infecção

Um exemplo prático de utilização do modelo pode ser visto na Figura 4.6, em que é calculado a porcentagem do pulmão afetado, e as máscaras das lesões geradas pelo modelo são utilizadas para realçar os possíveis focos de infecção, auxiliando e direcionando o diagnóstico dos profissionais de saúde.

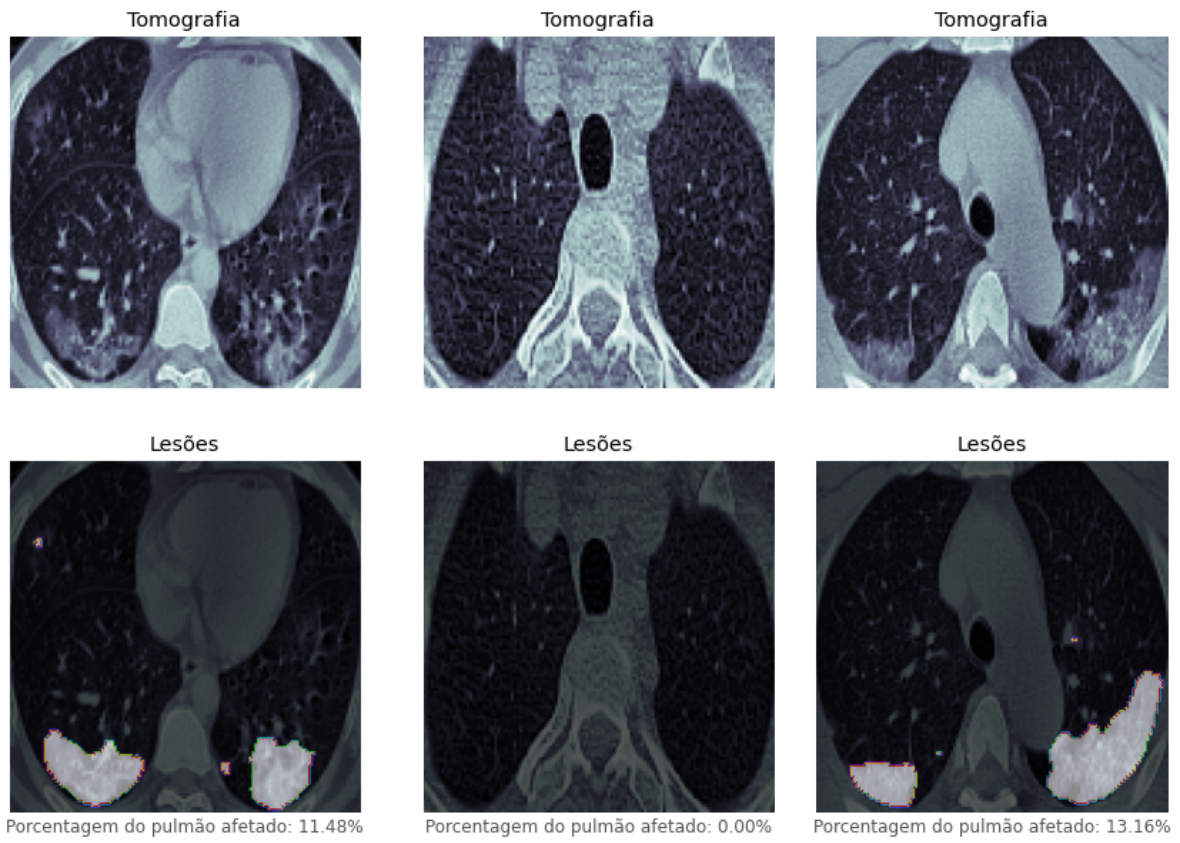


Figura 4.6: Sobreposição das lesões previstas com a tomografia original

Capítulo 5

Conclusão

Este trabalho realizou uma análise e comparação de uma rede UNet modificada para segmentação de lesões pulmonares, visto que o trabalho (KUCHANA et al., 2021) concluiu que a utilização de um esqueleto UNet, em conjunto com mudanças em hiper-parâmetros, número de filtros e mudanças nas camadas arquiteturais, resultam em um bom desempenho.

É possível observar na tabela Tabela 5.1, que quando comparada com a arquitetura proposta em (KUCHANA et al., 2021), a diferença de performance é praticamente insignificante, visto que os valores podem variar em pequena escala de acordo com diversas variáveis, como aleatoriedade da divisão dos dados, arquitetura computacional e a própria natureza estocástica dos modelos.

Tabela 5.1: Comparação do modelo com outros trabalhos citados.

Arquitetura	Acurácia (%)
UNet clássica (RONNEBERGER; FISCHER; BROX, 2015)	95.19
UNet (KUCHANA et al., 2021)	97.31
UNet modificada	98.01

Apesar da performance geral não ser melhorada, o modelo proposto é mais simples, de menor complexidade (menor número de parâmetros) e mais rápida execução. Somente foi possível alcançar uma performance similar devido ao maior pré-processamento das imagens, reduzindo a quantidade de ruídos, informações desnecessárias e, conseqüentemente, melhorando a qualidade dos dados de treinamento.

5.1 Trabalhos Futuros

Este projeto teve como objetivo desenvolver uma solução que auxilie o diagnóstico de um profissional da saúde ao visualizar tomografias computadorizadas com possíveis lesões pulmonares.

Apesar de muitas técnicas arquiteturais para melhorar o modelo não terem sido abordadas, é possível aumentar a eficácia do projeto proposto para o auxílio do profissional de saúde ampliando a gama de segmentações ou até mesmo classificando as lesões recortadas. Possíveis trabalhos futuros incluem:

- Aumento e melhor balanceamento das amostras de treinamento, evitando viés proveniente da possível multiplicidade de imagens iguais encontradas ao juntar diversas fontes de dados abertas e do excesso de somente um tipo de doença;
- Generalização dos dados de entrada, para abranger e segmentar mais tipos de lesões provenientes de outras doenças;
- Disponibilização do modelo como um serviço na nuvem, possibilitando a utilização em larga escala.

Referências Bibliográficas

- ARAUJO, T. *Source code GitHub repository*. 2021. Available at: <<https://github.com/Thiago-AS/covid-ct-scan-tcc>>. 14
- CHUNG, M. et al. Ct imaging features of 2019 novel coronavirus (2019-ncov). 2020. 2
- GARDNER, M.; DORLING, S. Artificial neural networks (the multilayer perceptron) — a review of applications in the atmospheric sciences. 1998. 8, 9
- GLICK, Y. *COVID-19 CT Lung and Infection Segmentation Dataset*. 2020. Available at: <<https://radiopaedia.org/playlists/25887>>. 15
- GOODFELLOW, I.; BENGIO, Y. Deep learning. 2015. 9, 10, 11, 12
- HUANG, C. et al. Clinical features of patients infected with 2019 novel coronavirus in wuhan, china. 2020. 1
- IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015. 16
- JORDAN, M. I.; MITCHELL, T. M. Machine learning: Trends, perspectives, and prospects. 2015. 5
- JUN, M. et al. *COVID-19 CT Lung and Infection Segmentation Dataset*. 2020. Available at: <<https://zenodo.org/record/3757476>>. 15
- KUCHANA, M. et al. Ai aiding in diagnosing, tracking recovery of covid-19 using deep learning on chest ct scans. 2021. ix, 22, 23, 33
- LAMBA, H. *Understanding Semantic Segmentation with UNET*. 2019. Available at: <<https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47>>. 13
- LAROBINA; MICHELE; MURINO, L. Medical image file formats. 2014. 15
- LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. 2015. 13
- MEDSEG. *COVID-19 CT segmentation dataset*. 2020. Available at: <<http://medicalsegmentation.com/covid19/>>. 15
- OLIVEIRA et al. A system based on artificial neural networks for automatic classification of hydro-generator stator windings partial discharges. 2017. 6

ORGANIZATION, W. H. Coronavirus disease 2019 (covid-19): situation report, 51. 2020. 1

ORGANIZATION, W. H. *Director-General's opening remarks at the media briefing on COVID-19*. 2020. 1

O'SHEA, K.; NASH, R. An introduction to convolutional neural networks. 2015. 11, 12

PISANO et al. Contrast limited adaptive histogram equalization image processing to improve the detection of simulated spiculations in dense mammograms. 1998. 16

REITTNER, P. et al. Pneumonia: high-resolution ct findings in 114 patients. 2003. 2

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. 2015. 21, 22, 33

RUSSELL, S. J.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. [S.l.: s.n.], 1994. 5, 6

YIN, Z. et al. A comparison of clinical and chest ct findings in patients with influenza a (h1n1) virus infection and coronavirus disease (covid-19). 2020. 3

ZHANG, A.; LIPTON, Z. *Dive into Deep Learning*. 2019. 12

ZHI, Z. L. Z. The epidemiological characteristics of an outbreak of 2019 novel coronavirus diseases (covid-19) in china. 2020. 1