

Ciência da Computação
Escola Politécnica PUC-PR

Daniela dos Santos Lima

**DEEP LEARNING PARA A SEGMENTAÇÃO DE ÓRGÃOS EM IMAGENS
MÉDICAS UTILIZANDO SELEÇÃO DINÂMICA E AUMENTO DE DADOS**

CURITIBA

02/2023

Ciência da Computação
Escola Politécnica PUC-PR

Daniela dos Santos Lima

**DEEP LEARNING PARA A SEGMENTAÇÃO DE ÓRGÃOS EM IMAGENS
MÉDICAS UTILIZANDO SELEÇÃO DINÂMICA E AUMENTO DE DADOS**

Relatório Parcial apresentado à Pontifícia
Universidade Católica do Paraná, Programa
Institucional de Bolsas de Iniciação Científica.

Orientador: Prof. Dr. André Hochuli

CURITIBA

02/2023

SUMÁRIO

1	INTRODUÇÃO	4
2	OBJETIVO(S)	6
3	FUNDAMENTAÇÃO TEÓRICA	7
3.1	INTELIGÊNCIA ARTIFICIAL	7
3.2	APRENDIZADO DE MÁQUINA.....	7
3.3	OVERFITTING E UNDERFITTING	8
3.4	REDES NEURAIS ARTIFICIAIS	8
3.4.1	<i>Neurônio Artificial</i>	<i>9</i>
3.4.2	<i>Soma Ponderada (combinação das entradas - O "Net"):</i>	<i>10</i>
3.4.3	<i>Função de Ativação</i>	<i>10</i>
3.5	REDE NEURAL MULTICAMADAS	12
3.6	DEEP LEARNING.....	12
3.7	REDES NEURAIS CONVOLUCIONAIS	14
3.7.1	<i>Camada de convolução</i>	<i>14</i>
3.7.2	<i>Camada de pooling.....</i>	<i>16</i>
3.7.3	<i>Camada totalmente conectada.....</i>	<i>16</i>
3.8	SEGMENTAÇÃO SEMÂNTICA	17
4	MATERIAIS E MÉTODOS	19
4.1	DETALHES DE IMPLEMENTAÇÃO	19
4.2	BASE DE DADOS	20
4.3	PRÉ-PROCESSAMENTO	20
4.3.1	<i>Conversão de formato</i>	<i>20</i>
4.4	AUMENTO DE DADOS.....	21
4.5	DADOS DE TREINAMENTO, VALIDAÇÃO E TESTE	22
4.6	ARQUITETURA	23
4.6.1	<i>U-Net.....</i>	<i>23</i>
5	RESULTADOS PARCIAIS	24
5.1	TREINAMENTO E VALIDAÇÃO	24
5.2	ANÁLISES VISUAIS	26
6	CONSIDERAÇÕES FINAIS.....	27
6.1	ETAPAS FUTURAS.....	28

1 INTRODUÇÃO

O tratamento mais comum para o câncer é a radioterapia, porém, por se tratar do uso de radiação, os órgãos adjacentes ao órgão alvo ficam expostos, correndo o risco de terem seus tecidos saudáveis afetados. Para isso, são feitas, por um especialista, segmentações manuais em imagens do tumor no órgão alvo. Uma tarefa trabalhosa e que consome muito tempo, além de demandar de profissionais altamente qualificados. Por essas razões, uma abordagem automática pode ser essencial para melhorar e simplificar a segmentação dos órgãos de risco.

A computação tem se revelado uma ferramenta fundamental para o avanço da medicina, possibilitando desde a análise de grandes volumes de dados até o desenvolvimento de terapias e dispositivos médicos inovadores. Com o progresso da tecnologia, os médicos agora são capazes de acessar e processar quantidades massivas de informações, auxiliando-os na identificação de padrões e anomalias em enfermidades antes indecifráveis. Adicionalmente, o emprego de técnicas de inteligência artificial, como as redes neurais, tem viabilizado a realização de prognósticos precisos acerca da evolução clínica e, isso pode ser visto, por exemplo, em estudos para detectar lesões derivadas do Coronavírus (SILVA, 2021).

Neste projeto, nosso foco constitui-se em criar algoritmos de Deep Learning (aprendizado profundo) e selecionar, de forma dinâmica, aqueles que apresentam os melhores resultados na segmentação dos órgãos presentes nas imagens médicas. Essas imagens, por sua vez, apresentam órgãos torácicos, que consistem no coração, na traqueia, no esôfago e na aorta. Essas partes possuem diferentes tamanhos e formas. Para segmentar esses órgãos de forma tridimensional, utilizamos uma base de dados chamada SegTHOR, Segmentation of THoracic Organs at Risk (LAMBERT et al., 2020), que contém imagens que usamos para ensinar e testar nosso algoritmo. No entanto, mesmo com o avanço tecnológico nos permitindo lidar com grandes volumes de dados para análises, ainda existe como obstáculo a insuficiência de dados para garantir uma aplicação confiável.

Portanto, com o propósito de estudar este óbice, inspirando-se em técnicas do estado da arte em Machine Learning (aprendizado de máquina) e Deep Learning,

tornamos a segmentação dos órgãos automatizada, bem como tratamos do aumento da base de dados e sua variabilidade, gerando imagens artificiais e rotuladas. Isto é, treinamos um modelo e segmentamos esses órgãos através de inteligência artificial e aprendizado de máquina, com o objetivo de tornar esse processo mais eficiente.

Este trabalho está organizado em seis seções. A Seção II apresenta nosso objetivo, expondo o problema e a proposta de solução. Na Seção III, é descrita a fundamentação teórica utilizada no desenvolvimento deste projeto. A Seção IV aborda os recursos utilizados e especifica os métodos seguidos, enquanto a Seção V apresenta os experimentos realizados e seus respectivos resultados. Finalmente, a Seção VI contém a conclusão e as etapas futuras.

2 OBJETIVO(S)

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 tumores, segmentando-as, e aferindo a porcentagem de determinado(s) órgão(os) afetado(s). 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 que atenda os padrões do estado da arte;
- Propor e desenvolver melhorias no modelo de base;
- Realizar validações e testes no modelo proposto;
- Comparar resultados, mostrar limitações e propor possíveis melhorias.

3 FUNDAMENTAÇÃO TEÓRICA

O propósito deste capítulo é apresentar algumas das áreas da Inteligência Artificial que são empregadas neste projeto. Começando com uma breve introdução ao Aprendizado de Máquinas, apresentando algumas técnicas e termos relevantes. Em seguida, serão expostos os fundamentos teóricos da Aprendizagem Profunda e das Redes Neurais Convolucionais, abordando seus principais métodos, características e benefícios.

3.1 Inteligência artificial

Inteligência artificial pode ser definida como "o ramo da ciência da computação que se ocupa do estudo e desenvolvimento de algoritmos e sistemas capazes de realizar tarefas que, até então, somente poderiam ser executadas por seres humanos, tais como reconhecimento de fala, processamento de imagens, tomada de decisões, entre outras" (RUSSELL; NORVIG, 2010).

3.2 Aprendizado de Máquina

A definição de Aprendizado de Máquina consiste em "um campo da ciência da computação que visa desenvolver algoritmos e técnicas para permitir que os computadores aprendam com dados, sem necessidade de programação explícita" (ALPAYDIN, 2010).

Com base em informações prévias, uma máquina pode simular determinado comportamento, buscando explorar dados úteis e maximizar uma medida de desempenho, visando o treinamento de um modelo eficiente e auxiliando na análise de informações futuras (JORDAN; MITCHELL, 2015).

3.3 Overfitting e Underfitting

O *overfitting* e *underfitting* são dois problemas que surgem frequentemente no campo de aprendizado de máquina, e que têm um impacto significativo na capacidade dos modelos de generalizar para novos dados. O *overfitting* acontece quando o modelo é excessivamente complexo e se ajusta demasiado aos dados de treinamento, perdendo assim a capacidade de generalizar para dados que não tenha visto anteriormente. O *underfitting*, por outro lado, ocorre quando o modelo é muito simples e não consegue capturar a complexidade dos dados de treinamento, e, portanto, não consegue ajustar-se aos dados de treinamento ou generalizar para novos dados. Para evitar esses problemas, é importante ajustar os parâmetros do modelo ou modificá-lo para torná-lo mais adequado para o conjunto de dados, e dividir os dados em conjuntos de treinamento e teste para avaliar o desempenho do modelo (BISHOP, 2006).

Dessa forma, é possível avaliar se o modelo está sofrendo de *overfitting* ou *underfitting*, e fazer as correções necessárias para melhorar seu desempenho. Além disso, é importante escolher cuidadosamente as variáveis independentes ou camadas ocultas em redes neurais para garantir que o modelo possa lidar com a complexidade dos dados de treinamento e ainda assim generalizar para novos dados. Uma vez que esses problemas são resolvidos, o modelo pode ser mais eficaz na previsão de dados não vistos anteriormente e na identificação de padrões úteis para o usuário final (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

3.4 Redes Neurais Artificiais

Uma Rede Neural Convolutacional (Convolutional Neural Network - CNN) é um tipo de rede neural profunda especialmente projetado para o processamento de dados com estrutura em forma de matriz, como imagens, sinais de áudio e vídeos. Essa arquitetura é composta por camadas convolucionais, de pooling e de classificação, e

foi desenvolvida para extrair características dos dados de entrada por meio de filtros convolucionais e reduzir sua dimensionalidade por meio de camadas de pooling.

As redes neurais convolucionais foram amplamente adotadas em aplicações de visão computacional, incluindo reconhecimento de imagem, detecção de objetos, segmentação semântica e classificação de vídeo (GOODFELLOW; BENGIO; COURVILLE, 2016).

3.4.1 Neurônio Artificial

Um neurônio artificial é uma unidade básica de processamento em uma rede neural artificial. Ele é inspirado no neurônio biológico encontrado no cérebro humano. Assim como um neurônio biológico recebe sinais de entrada de outros neurônios, um neurônio artificial, Figura 3.1, recebe sinais de entrada de outras unidades de processamento, e processa esses sinais para produzir uma saída.

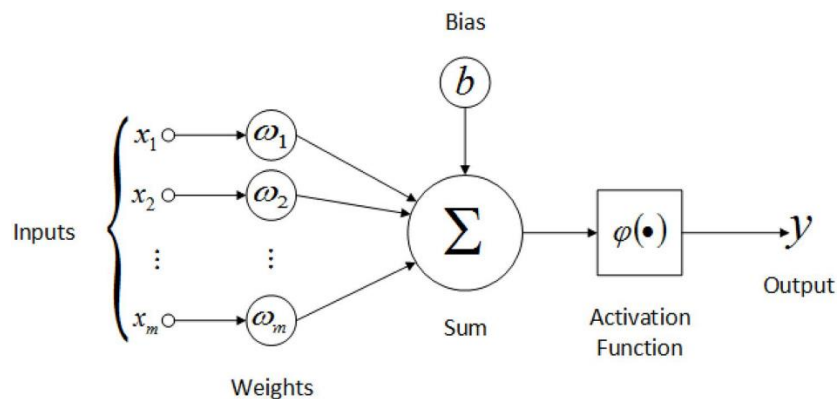


Figura 3.1: Neurônio Artificial (OLIVEIRA et al., 2017)

O funcionamento de um neurônio artificial é dividido em três partes principais: a soma ponderada, a função de ativação e a saída.

O processo de ajuste dos pesos sinápticos de um neurônio artificial é conhecido como treinamento. Durante o treinamento, a rede neural é exposta a um conjunto de exemplos de entrada e saída conhecidos, e os pesos sinápticos são ajustados de

forma a minimizar a diferença entre a saída produzida pelo neurônio e a saída desejada.

3.4.2 Soma Ponderada (combinação das entradas - O "Net"):

Na soma ponderada, Figura 3.2, cada entrada é multiplicada por um peso sináptico correspondente e os resultados são somados, dando origem ao chamado "net" de um neurônio.

$$net_i(t) = \sum_{j=1}^n w_{ij} x_j(t)$$

Figura 3.2: Somatório das Entradas - O "Net" (ZAMBIASI, 2017)

w_{ij} é um número real que representa a conexão sináptica da entrada **j**-ésimo neurônio. A conexão sináptica é conhecida como excitatória se **w_{ij}** > 0 ou inibitória caso **w_{ij}** < 0.

A função de ativação é então aplicada à soma ponderada para produzir a saída do neurônio.

3.4.3 Função de Ativação

Existem várias funções de ativação diferentes que podem ser usadas em um neurônio artificial, incluindo a função sigmoide e a função ReLU.

A escolha da função de ativação depende do problema em questão e das características dos dados.

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.

A função sigmoide, Figura 3.3, é muito utilizada em problemas de classificação, onde tem como saída valores entre 0 e 1, e resulta na probabilidade de os dados de entrada estarem contidos na classe analisada.

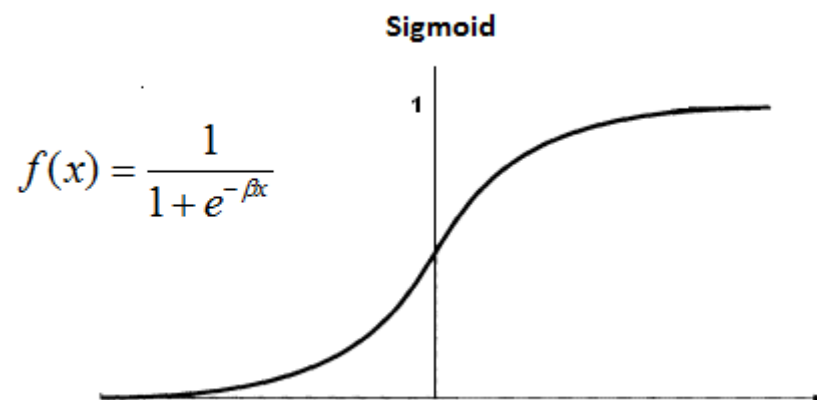


Figura 3.3: Sigmoide

A função ReLU, Figura 3.4, é uma função de ativação mais eficiente, pois não faz uso de expoentes, por este motivo é comumente utilizada em camadas ocultas.

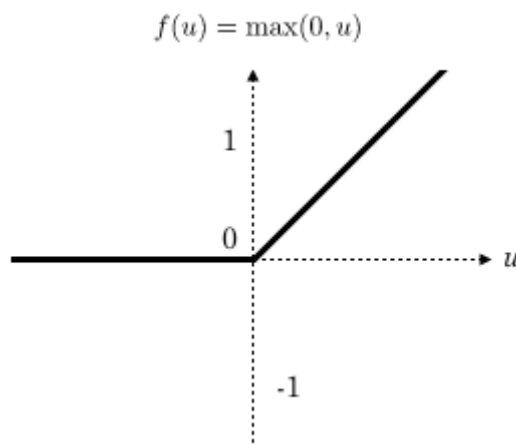


Figura 3.4: ReLU

3.5 Rede Neural Multicamadas

Redes Neurais Artificiais são modelos computacionais inspirados no sistema nervoso central humano. As redes neurais multicamadas (ou MLP, do inglês Multi-Layer Perceptron), Figura 3.5, são um tipo de rede neural que consiste em uma ou mais camadas de neurônios artificiais. Cada camada recebe entradas, realiza uma transformação linear das entradas por meio de um conjunto de pesos sinápticos e aplica uma função de ativação não linear aos resultados para gerar saídas. A saída da camada anterior é a entrada da camada seguinte, e assim por diante, até que a saída final da rede seja produzida. A rede MLP é treinada para aprender a relação entre as entradas e as saídas por meio do ajuste iterativo dos pesos sinápticos com base em um conjunto de dados de treinamento (BISHOP, 1995).

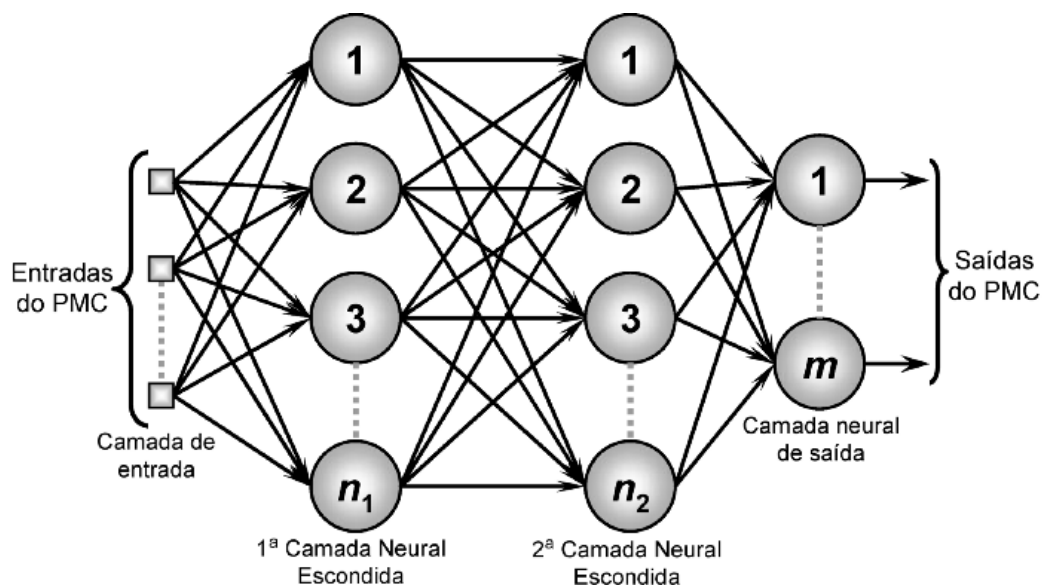


Figura 3.5: Multi-Layer Perceptron (OLIVEIRA et al., 2016)

3.6 Deep Learning

O *Deep Learning* (aprendizado profundo) é uma subárea do campo de inteligência artificial que envolve a construção e treinamento de modelos de redes neurais profundas para realizar tarefas de aprendizado automático. Esses modelos

são chamados de "profundos" porque possuem muitas camadas intermediárias entre a entrada e a saída da rede, permitindo que as informações sejam processadas de maneira hierárquica, como apresenta a Figura 3.6. O objetivo final do aprendizado profundo é criar modelos de inteligência artificial capazes de realizar tarefas complexas, como reconhecimento de fala, visão computacional, processamento de linguagem natural, entre outras (GOODFELLOW; BENGIO, 2015).

Uma das principais razões pelas quais o aprendizado profundo é tão poderoso é a capacidade das redes neurais profundas de aprender representações de alta qualidade das entradas que recebem. As camadas intermediárias da rede aprendem a reconhecer características abstratas nas entradas, e as camadas superiores da rede combinam essas características para realizar a tarefa de saída. Isso permite que os modelos de aprendizado profundo capturem nuances sutis nos dados de entrada e realizem tarefas complexas com precisão (GOODFELLOW; BENGIO, 2015).

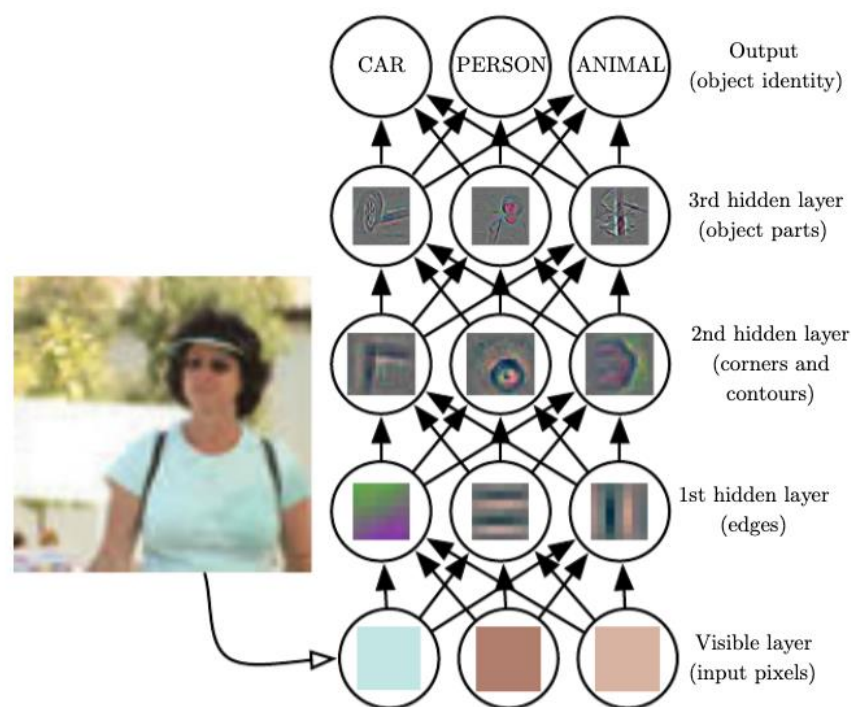


Figura 3.6: Modelo DL extraindo conceitos simples e abstratos.
(GOODFELLOW; BENGIO, 2015)

3.7 Redes Neurais Convolucionais

Uma rede neural convolucional (CNN) é um tipo de rede neural projetada especificamente para processar dados que possuem uma estrutura espacial, como imagens. As CNNs utilizam camadas convolucionais, que aplicam filtros ou kernels a pequenas regiões dos dados de entrada, buscando identificar padrões e características específicas, Figura 3.7. As camadas de pooling são comumente usadas para reduzir a dimensionalidade dos dados, preservando apenas as características mais importantes. As camadas totalmente conectadas são usadas para combinar as características extraídas pelas camadas convolucionais e produzir uma saída final (GOODFELLOW; BENGIO, 2015).

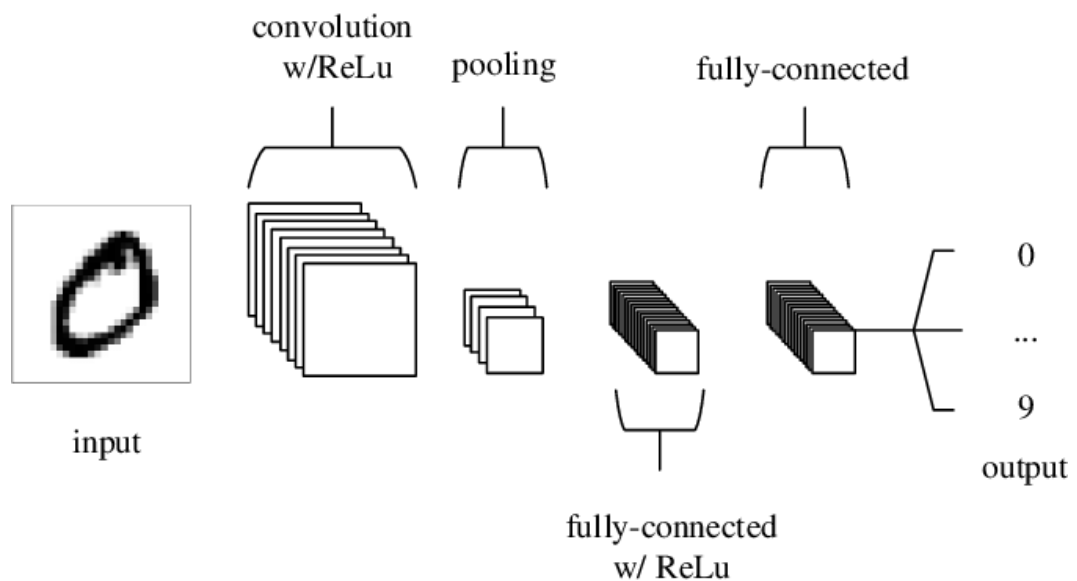


Figura 3.7: Arquitetura de uma Rede Neural Convolucional (GOODFELLOW; BENGIO, 2015)

3.7.1 Camada de convolução

Uma camada convolucional é uma das camadas fundamentais em uma rede neural convolucional (CNN). Esta camada é responsável por extrair características

importantes das imagens de entrada por meio da aplicação de operações de convolução em filtros de tamanho definido.

Segundo (GOODFELLOW et al., 2016), a camada convolucional é composta por vários filtros, cada um dos quais consiste em um pequeno conjunto de pesos que é aplicado a uma pequena região da imagem de entrada, produzindo um mapa de características para cada filtro, Figura 3.8. A convolução é aplicada repetidamente em diferentes regiões da imagem de entrada para produzir um conjunto de mapas de características que representam diferentes aspectos da imagem. Esses mapas de características são então passados para a próxima camada da rede neural para processamento adicional.

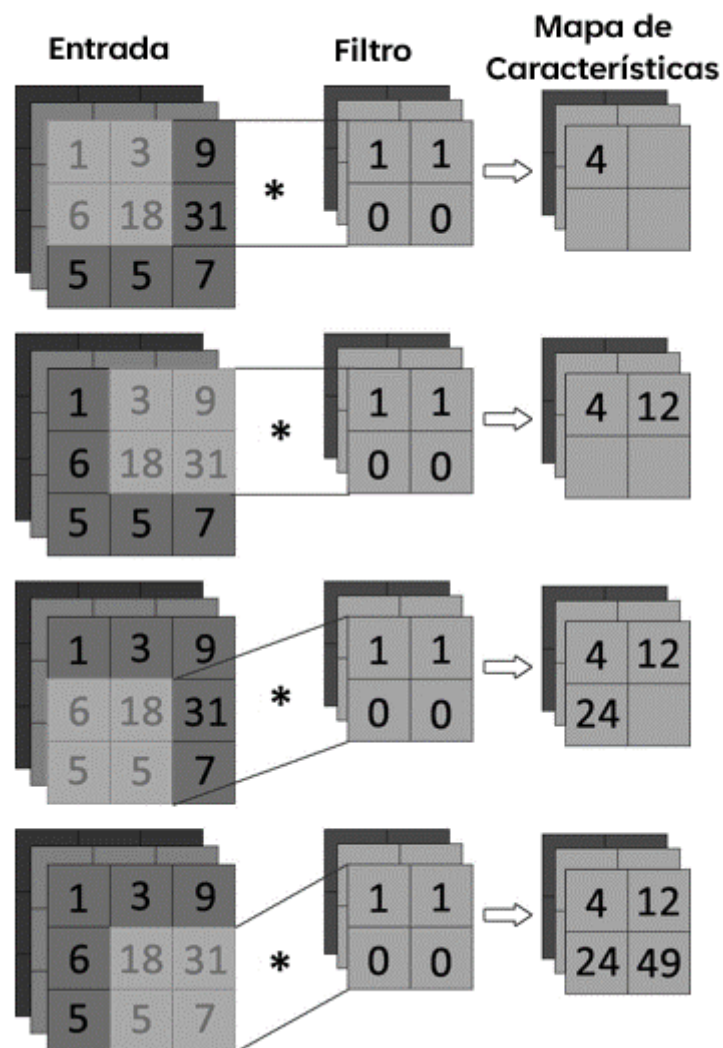


Figura 3.8: Representação visual da camada de convolução

3.7.2 Camada de pooling

De acordo com (GOODFELLOW et al., 2016), uma camada de pooling é usada principalmente para reduzir a dimensionalidade do conjunto de dados, simplificando o cálculo em camadas subsequentes e reduzindo o risco de *overfitting*. Na operação de pooling, a camada recebe um conjunto de características e aplica uma função matemática para reduzir o tamanho dos mapas de características de entrada. Isso é geralmente feito selecionando o valor máximo (max-pooling) ou a média (average-pooling) em uma região de entrada, Figura 3.9.

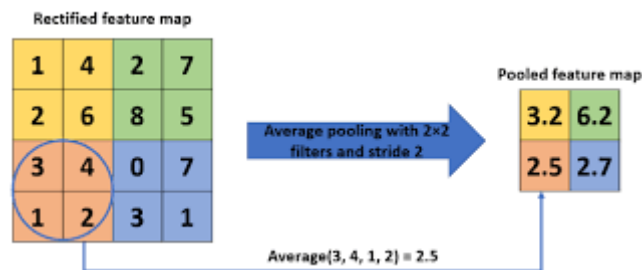


Figura 3.9: Representação visual da camada de pooling (GHOLAMALINEZHAD; KHOSRAVI, 2020)

3.7.3 Camada totalmente conectada

Uma camada totalmente conectada é uma camada em uma rede neural que conecta todos os neurônios de uma camada à camada seguinte, ou seja, cada neurônio na camada totalmente conectada é conectado a todos os neurônios na camada subsequente. Essa camada é frequentemente usada para a classificação final em uma rede neural, onde cada neurônio representa uma classe possível.

Segundo (GOODFELLOW et al., 2016), uma camada totalmente conectada é responsável por mapear as características extraídas pelas camadas convolucionais ou outras camadas intermediárias para as classes de saída, Figura 3.10. Cada

neurônio na camada totalmente conectada recebe como entrada a ativação de todos os neurônios na camada anterior, e produz uma saída que é passada para a próxima camada da rede neural.

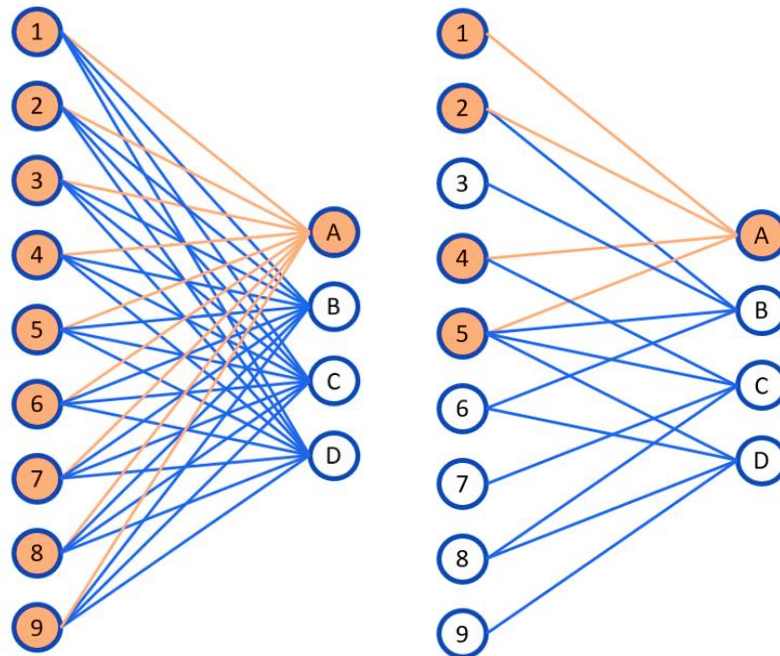


Figura 3.10: Representação visual da camada totalmente conectada (UNZUETA, 2021)

3.8 Segmentação Semântica

A segmentação semântica é um processo que envolve a atribuição de um rótulo a cada pixel de uma imagem com base em suas características, como cor, textura e forma, Figura 3.11. Essa tarefa é realizada por meio do uso de redes neurais convolucionais, que são capazes de aprender a representação de características das imagens de entrada e associá-las aos rótulos corretos (LONG et al., 2015).

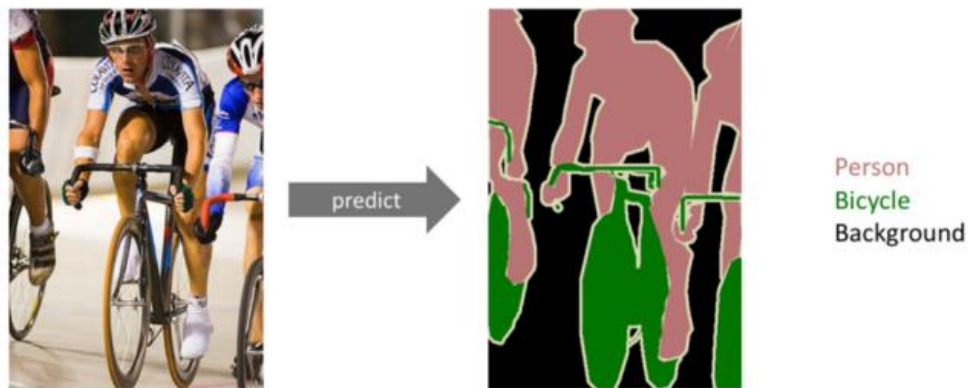


Figura 3.11: Segmentação Semântica (LAMBA, 2019)

4 MATERIAIS E MÉTODOS

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 Convolucional para segmentação dos órgãos anteriormente citados a partir de tomografias torácicas. Para isto o experimento foi dividido em duas principais etapas: pré-processamento, aumento dos dados e concepção da arquitetura do modelo.

4.1 Detalhes de implementação

A rede neural convolucional e todo o pré-processamento dos dados foram implementados em Python, utilizando bibliotecas como TensorFlow, Keras, NumPy e OpenCV, no um sistema operacional Microsoft Windows 11.

As execuções foram feitas em duas plataformas, sendo uma parte no Google Colaboratory e outra parte em uma máquina com CPU Intel(R) Core(TM) i5- 12500H @ 2.50 GHz, GPU NVIDIA GeForce RTX 3050 Ti 4 GB VRAM e 16 GB 4800 MHz DDR5 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 a biblioteca TensorFlow-GPU.

Todo código fonte necessário para reproduzir os experimentos pode ser encontrado em um repositório aberto na plataforma GitHub disponível em: <<https://github.com/mrslima/pibic>>.

4.2 Base de Dados

As amostras utilizadas foram retiradas do dataset SegTHor (LAMBERT et al., 2020), dedicado à segmentação de órgãos de risco no tórax.

Um total de 14.680 imagens foram utilizadas, sendo subdivididas igualmente entre 2 categorias: tomografias torácicas e máscaras com recorte de cada órgão.

4.3 Pré-processamento

O pré-processamento de dados é uma etapa essencial 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.

4.3.1 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 Tagged Image File Format (.tiff).

As imagens foram transformadas 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 4.1.

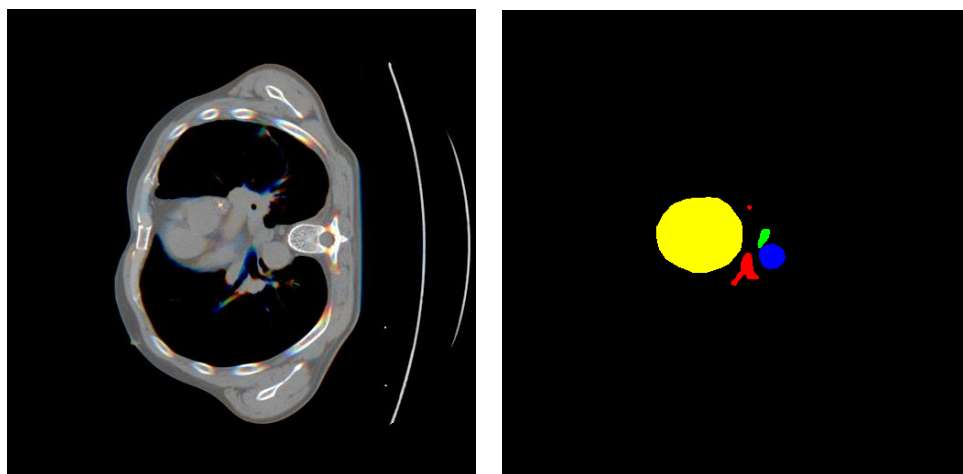


Figura 4.1: Dados de entrada: Tomografia e Máscara dos órgãos.

4.4 Aumento de dados

Data Augmentation, ou aumento de dados, é uma técnica utilizada para gerar novos exemplos de dados de treinamento com o objetivo de melhorar a generalização do modelo e reduzir o *overfitting*.

Uma rede neural convolucional possui a capacidade de classificar e segmentar dados de forma consistente, mesmo quando eles estão em orientações diferentes, devido à sua invariância em relação a escala, translação e outras transformações. No entanto, devido à grande quantidade de parâmetros que uma CNN utiliza, é necessário apresentar uma quantidade proporcionalmente grande de exemplos para obter uma boa generalização e desempenho do modelo. Como muitas vezes há uma baixa quantidade de amostras de entrada disponíveis em acervos públicos, é necessário recorrer ao aumento de dados para obter resultados satisfatórios.

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

- Chance aleatória de virar a imagem verticalmente;
- Chance aleatória de virar a imagem horizontalmente;
- Chance aleatória de redimensionar o zoom da imagem, ampliando ou reduzindo por um fator de 30%;

- Chance aleatória de rotacionar em 40° para esquerda ou direita.

O resultado dessas transformações pode ser observado na Figura 4.2, onde para cada dado de entrada, foi gerado seu respectivo par transformado, dobrando a quantidade de exemplares de treinamento.

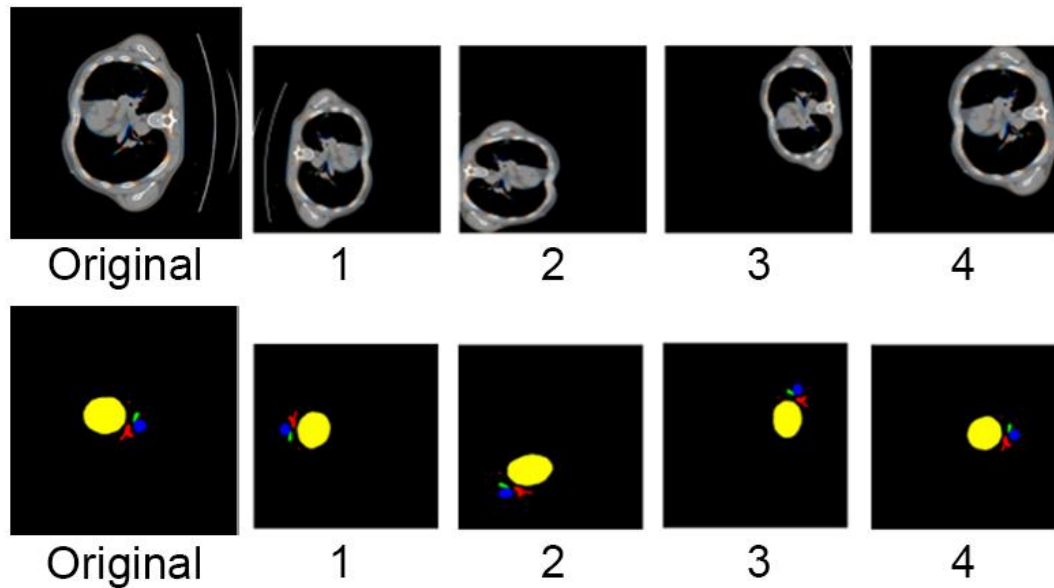


Figura 4.2: Resultado do aumento de dados aplicado em uma imagem.

4.5 Dados de Treinamento, Validação e Teste

Com o aumento de dados, o número de imagens foi duplicado, resultando em 29.360 amostras. Logo em seguida, foram feitas separações aleatórias dos conjuntos em: 80% das imagens para treinamento, 10% para validação e 10% para teste.

4.6 Arquitetura

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

4.6.1 U-Net

Para ter sucesso no treinamento de redes profundas, é necessário ter dezenas de milhares de exemplos de treinamento. A Figura 4.3 apresenta a U-Net, uma rede e estratégia de treinamento que se destaca pelo uso intensivo de aumento de dados para tornar o melhor uso das amostras disponíveis. A U-Net é uma arquitetura derivada de uma Rede Neural Totalmente Convolucional, mas que foi modificada e expandida para exigir poucas imagens de treinamento e, ao mesmo tempo, produzir segmentações mais precisas. (RONNEBERGER; FISCHER; BROX, 2015).

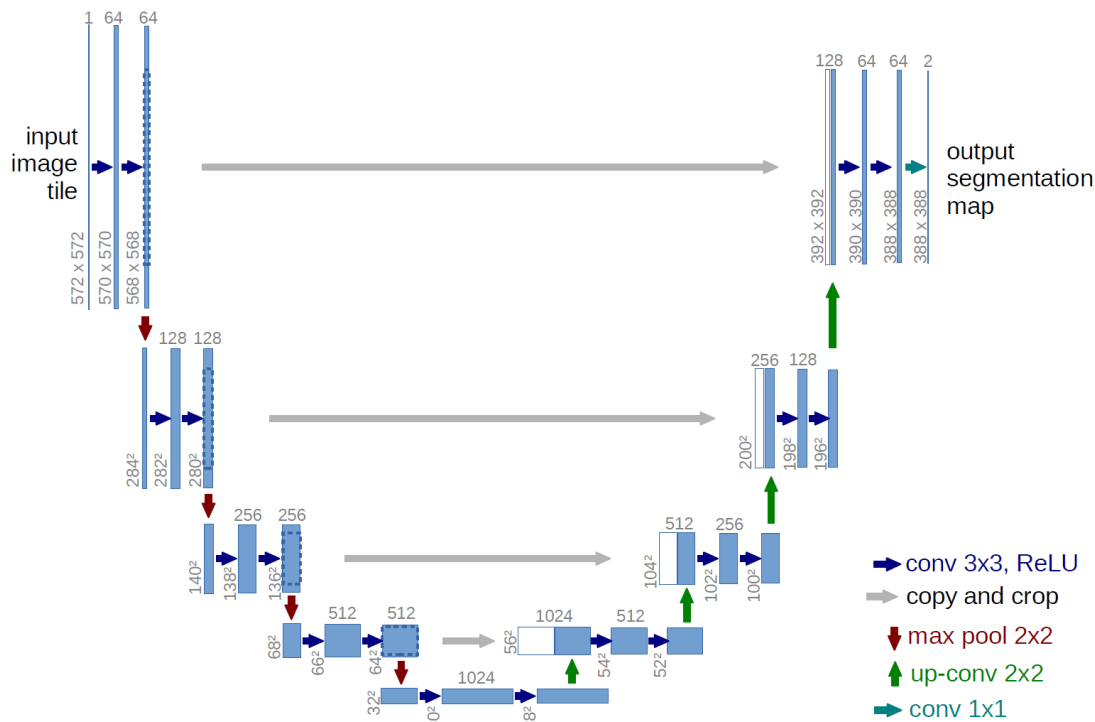


Figura 4.3: Arquitetura U-Net (RONNEBERGER; FISCHER; BROX, 2015)

5 RESULTADOS PARCIAIS

5.1 Treinamento e validação

Para realizar o treinamento da topologia, as etapas de pré-processamento, aumento de dados e separação do conjunto de dados foram executadas, conforme descrito no Capítulo 3. O treinamento foi realizado em 50 épocas e com um tamanho de lote de 16. Foram utilizados callbacks de redução da taxa de aprendizagem em platôs e parada precoce para evitar *overfitting* do modelo durante o treinamento.

Foi gerado um gráfico para cada época, demonstrando os resultados de acurácia, Figura 5.1, e erro, Figura 5.2, das segmentações pulmonares e das lesões para os conjuntos de treinamento e validação. Esses gráficos permitem uma análise qualitativa da influência da taxa de aprendizagem e do desempenho das topologias durante o treinamento.

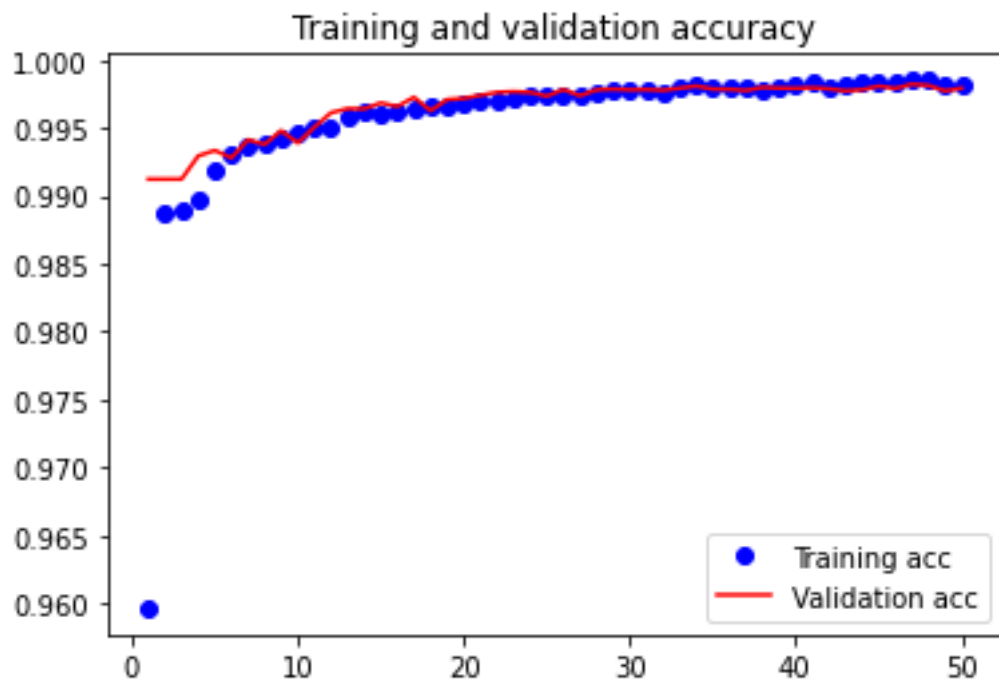


Figura 5.1: Gráfico de acurácia da rede U-Net

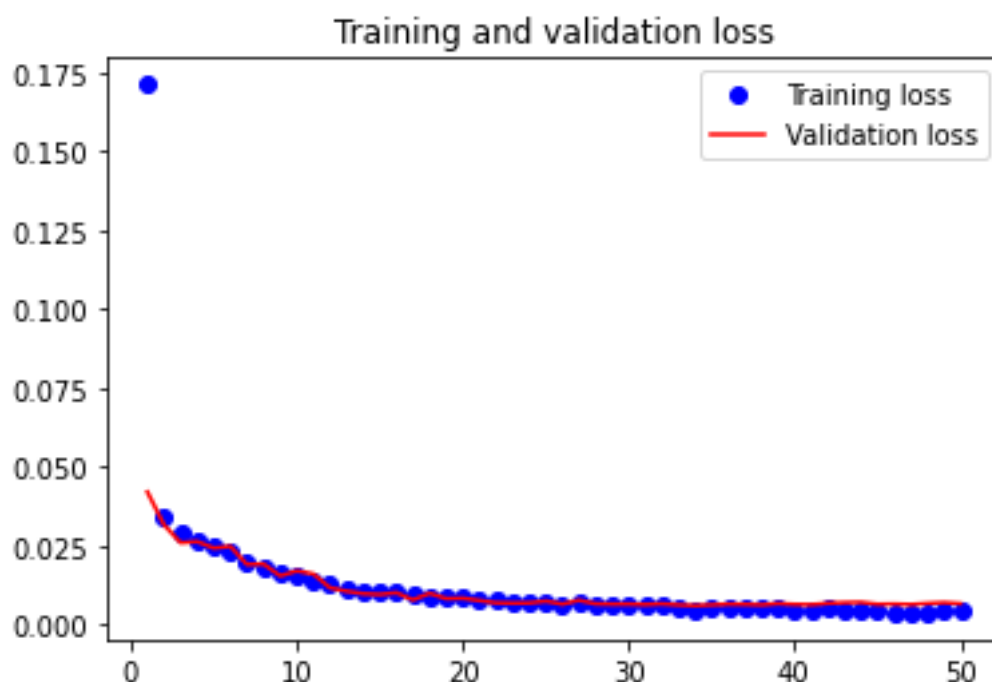


Figura 5.2: Gráfico de custo da rede U-Net

	Modelo
Acurácia	99,804%
Custo	00,056%
IoU Score (Jaccard Index)	99,495%
Dice Coefficient (F1 Score)	99,746%

Tabela 5.1: Resultados do conjunto de teste da U-Net.

Conforme os resultados da Tabela 5.1, essa estrutura apresentou um desempenho excelente. No entanto, durante a análise, identificamos que há uma falha na forma como as métricas são calculadas, o que nos gera questionamentos. Foi observado que o modelo captura uma classe indesejada, o background, o que pode levar a uma resposta tendenciosa, uma vez que essa classe não deve ser considerada. Por isso, estamos trabalhando para corrigir esse problema. Infelizmente, devido a essa falha, não podemos verificar se ocorre excesso (overfitting) ou falta (underfitting) de ajuste do modelo.

5.2 Análises Visuais

Com o objetivo de validar a precisão dos resultados de desempenho, foram criados exemplos de saída e comparados com as máscaras de treinamento correspondentes. A Figura 5.5 e a Figura 5.6 apresentam uma comparação entre as máscaras de treinamento e as geradas pelo modelo, permitindo observar a semelhança entre elas.

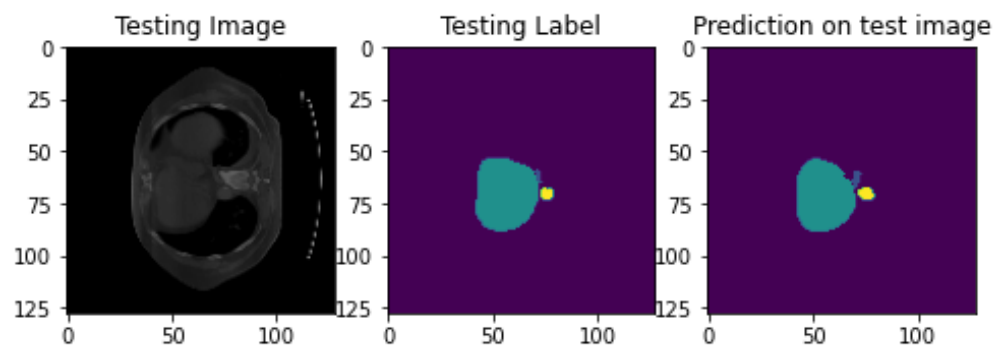


Figura 5.5: Saída prevista pelo modelo: Segmentação dos órgãos aparentes

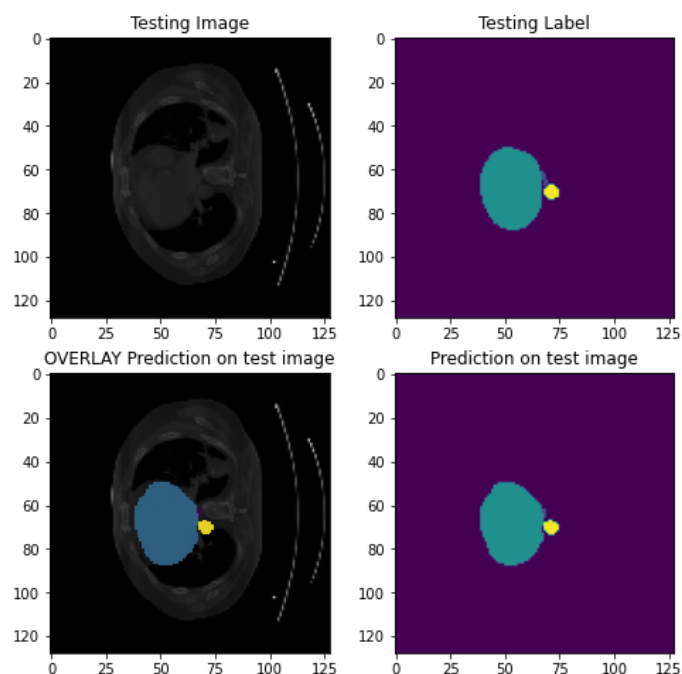


Figura 5.6: Sobreposição dos órgãos previstos com a tomografia original

6 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo principal desenvolver um modelo de segmentação de órgãos torácicos em imagens médicas por meio de aprendizado profundo. Após a realização dos experimentos e análise dos resultados obtidos, verificamos que a abordagem de deep learning aplicada apresentou excelentes resultados na segmentação dos órgãos de interesse, proporcionando alta acurácia e sensibilidade na identificação dos tecidos-alvo.

Os resultados obtidos confirmam que o modelo desenvolvido pode ser uma ferramenta promissora para o auxílio ao diagnóstico e tratamento de doenças oncológicas, oferecendo uma solução automatizada e mais eficiente para a segmentação de órgãos em imagens médicas.

Apesar dos excelentes resultados, também identificamos algumas limitações em relação à segmentação do background e o risco de viés na avaliação das métricas. Sendo assim, sugerimos como trabalhos futuros a correção desses problemas, bem como a ampliação da base de dados e a realização de novos experimentos para validar e aprimorar ainda mais o modelo desenvolvido.

Em suma, a aplicação de deep learning na segmentação de órgãos torácicos em imagens médicas mostrou-se uma abordagem promissora, com potencial para contribuir significativamente para o avanço da medicina e oferecer benefícios práticos na identificação e tratamento de doenças oncológicas.

Além disso, como uma estudante que iniciou este projeto no segundo período do curso de ciência da computação e sem qualquer experiência com inteligência artificial, pude ter um desenvolvimento expressivo nesta área, graças à esse programa e o tempo para a fundamentação teórica deste.

6.1 Etapas Futuras

As etapas realizadas até agora foram relacionadas principalmente no aprendizado teórico e prático de aplicações de inteligência artificial para manipulação de imagens. Após a entrega deste relatório parcial, o objetivo é aplicar as teorias pesquisadas de forma prática com a seleção dinâmica dos modelos e avaliar possíveis resultados.

O algoritmo de seleção dinâmica, Figura 6.1, é responsável por identificar a melhor arquitetura de rede neural para segmentar uma classe específica em uma imagem aleatória. Para isso, ele passará por alguns modelos diferentes e, com base nos melhores resultados de métricas, o algoritmo determinará qual arquitetura apresenta o melhor desempenho para segmentar uma classe específica contida na imagem de entrada. Em seguida, o algoritmo irá gerar uma imagem de saída que combina as classes segmentadas com o modelo mais adequado para cada uma delas, garantindo resultados mais confiáveis.

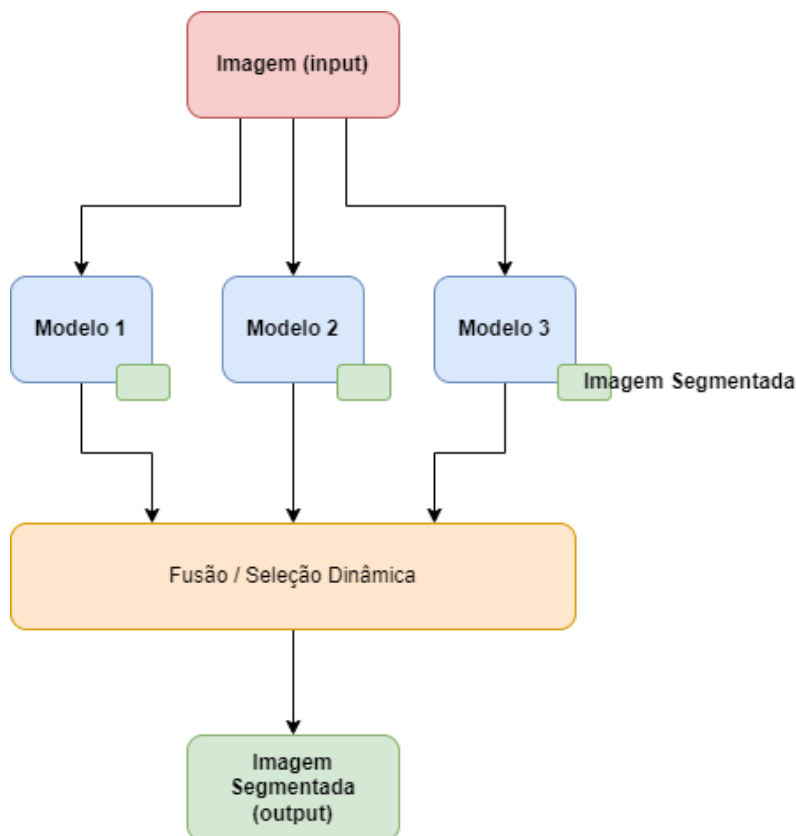


Figura 6.1: Seleção Dinâmica

Portanto, para que todos os resultados sejam atingidos, elaboramos o cronograma abaixo:

Tarefas	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out
Seleção dinâmica dos modelos	X	X	X					
Experimentações Finais			X	X				
Produção do Relatório Final SEMIC				X	X			
								X

Tabela 6.1: Cronograma

Inicialmente, abordaremos os problemas descritos na seção cinco e, em seguida, procederemos com a construção e treinamento dos algoritmos de seleção dinâmica dos modelos. Posteriormente, realizaremos as experimentações finais nos modelos para produzir o relatório final. Por fim, apresentaremos o projeto no SEMIC.

REFERÊNCIAS BIBLIOGRÁFICAS

LAMBERT, Z., PETITJEAN, C., DUBRAY, B., & KUAN, S. SegTHOR: segmentation of thoracic organs at risk in CT images. 2020.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: convolutional networks for biomedical image segmentation. 2015.

RUSSELL, Stuart J.; NORVIG, Peter. Artificial Intelligence: A Modern Approach. Prentice Hall, 2010.

ALPAYDIN, Ethem. Introduction to Machine Learning. MIT Press, 2010.

JORDAN, M. I.; MITCHELL, T. M. Machine learning: Trends, perspectives and prospects. 2015.

BISHOP, C. M. Pattern recognition and machine learning (Vol. 4). springer. 2006.

HASTIE, T., TIBSHIRANI, R., & Friedman, J. The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media, 2009.

GOODFELLOW, I., BENGIO, Y., & COURVILLE, A. Deep Learning. MIT Press, 2016.

OLIVEIRA, R. M. et al. A system based on artificial neural networks for automatic classification of hydro-generator stator windings partial discharges. 2017.

ZAMBIASI, Saulo P. O Neurônio Artificial. GSIGMA UFSC, 2022. Disponível em: <https://www.gsigma.ufsc.br/~popov/aulas/rna/neuronio_artificial/index.html>. Acesso em: 25 fev, 2023.

BISHOP, C. M. Neural networks for pattern recognition. Oxford University press, 1995.

OLIVEIRA, R. A. et al. Modelo de previsão de demanda energética utilizando redes neurais artificiais: um estudo de caso para a região Nordeste do Brasil. Revista Brasileira de Métodos Quantitativos. 2016.

GHOLAMALINEZHAD, Hossein; KHOSRAVI, Hossein. Pooling Methods in Deep Neural Networks, a Review. 2020.

UNZUETA, D. Convolutional Layers vs Fully Connected Layers. Towards Data Science, 2021. Disponível em: <<https://towardsdatascience.com/convolutional-layers-vs-fully-connected-layers-364f05ab460b>>. Acesso em: 25 fev, 2023.

LAMBA, H. Understanding Semantic Segmentation with UNET. Towards Data Science, 2019. Disponível em: <<https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47>>. Acesso em: 25 fev, 2023.

LONG, J., SHELHAMER, E., & DARRELL, T. Fully convolutional networks for semantic segmentation. 2015.

SILVA, Thiago Araújo da. Uma rede UNet modificada para segmentação e identificação de lesões pulmonares em tomografias. 2021. 48 f. Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação) – Curso de Engenharia de Computação – Universidade de Brasília, Brasília, 2021.