

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**GABRIEL LABRES MARTINI**

**MODELO DE DEEP LEARNING PARA ANÁLISE DE NÓDULOS  
PULMONARES EM IMAGENS MÉDICAS**

**CURITIBA**

**2024**

**GABRIEL LABRES MARTINI**

**MODELO DE DEEP LEARNING PARA ANÁLISE DE NÓDULOS  
PULMONARES EM IMAGENS MÉDICAS**

**Deep Learning model for pulmonary nodule analysis in medical images**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Eletrônica do Curso de Bacharelado em Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. José Jair Alves Mendes Junior

**CURITIBA**

**2024**



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**GABRIEL LABRES MARTINI**

**MODELO DE DEEP LEARNING PARA ANÁLISE DE NÓDULOS  
PULMONARES EM IMAGENS MÉDICAS**

Trabalho de Conclusão de Curso de Graduação  
apresentado como requisito para obtenção do  
título de Bacharel em Engenharia Eletrônica  
do Curso de Bacharelado em Engenharia  
Eletrônica da Universidade Tecnológica Federal  
do Paraná.

Data de aprovação: 15/março/2024

---

José Jair Alves Mendes Junior  
Doutor  
Universidade Tecnológica Federal do Paraná

---

Gustavo Benvenuti Borba  
Doutor  
Universidade Tecnológica Federal do Paraná

---

Daniel Rossato de Oliveira  
Mestre  
Universidade Tecnológica Federal do Paraná

**CURITIBA  
2024**

Dedico este trabalho a todos aqueles que enfrentam a batalha contra o câncer, sua coragem e força são uma inspiração constante.

## AGRADECIMENTOS

Agradeço à todo o corpo docente da universidade, pela qualidade do ensino provido e por todo o apoio dado ao longo de minha jornada universitária. Com certeza, não foi um período fácil em minha vida, onde me deparei várias vezes com desafios que pareciam impossíveis, os quais exigiram muito esforço, horas de estudo e desgaste para serem superados. Porém, atribuo muito do meu crescimento pessoal e profissional à minha formação como engenheiro, que me forneceu arcabouço para lidar com os mais variados tipos de situações e adversidades.

Em especial, agradeço ao meu orientador, Prof. Dr. José Jair Alves Mendes Junior, por toda sua dedicação e empenho em exercer seu papel. Agradeço também aos professores Rubens Eduardo Garcia Machado, Hugo Vieira Neto (*In Memoriam*) e Douglas Paulo Bertrand Renaux por serem professores excelentes e me encorajarem à sempre buscar mais conhecimento.

Agradeço ao meu falecido avô Darci Labres (*In Memoriam*), por todos os momentos de carinho e todas as lembranças. Meu avô foi vítima de câncer e foi uma das grandes inspirações para a realização deste trabalho.

Agradeço à toda a minha família e amigos pelo apoio nos momentos difíceis, certamente foram essenciais não só neste trabalho, mas em todo o meu percurso pela universidade.

Registro, também, minha gratidão ao Prof. Dr. André Eugenio Lazzaretti e toda a sua equipe pelo fornecimento de acesso ao *cluster* da UTFPR, onde pude testar meus algoritmos com maior rapidez.

"A inteligência artificial e sua ramificação, o aprendizado de máquina, serão ferramentas fundamentais para promover o bem social, bem como o sucesso nos negócios."(HURD, Mark, 2017)

## RESUMO

O câncer pulmonar é uma das principais causas de morte relacionadas ao câncer em todo o mundo, e o diagnóstico precoce é crucial para melhorar as taxas de sobrevivência dos pacientes. A tomografia computadorizada do tórax é uma ferramenta essencial para a detecção de nódulos pulmonares, que podem ser indicativos de câncer pulmonar. No entanto, a análise manual dessas imagens por radiologistas é uma tarefa demorada e sujeita a variabilidade inter-observador. Nessa perspectiva, este trabalho apresenta um *software* para análise de exames médicos de imagem utilizando aprendizagem de máquina. Essas imagens são provenientes de exames de tomografia computadorizada do tórax. O software desenvolvido realiza o processo de segmentação de possíveis nódulos pulmonares presentes na imagem e os classifica entre nódulos ou não-nódulos, servindo como uma ferramenta de auxílio a diagnóstico. O desenvolvimento foi realizado em linguagem *Python*, utilizando a biblioteca *PyTorch* para implementação da rede neural convolucional utilizando o modelo U-Net. Foi utilizada a base de dados LIDC-IDRI, que contém mais de 800 exames de tomografia de tórax, com anotações feitas por médicos radiologistas sobre os nódulos presentes. Como resultado, o sistema demonstrou capacidade de segmentar e classificar corretamente 85% dos nódulos pulmonares presentes nos exames da base de dados.

**Palavras-chave:** inteligência artificial; deep learning; tomografia.

## ABSTRACT

Lung cancer is one of the leading causes of cancer-related deaths worldwide, and early diagnosis is crucial for improving patient survival rates. Chest computed tomography scans are essential tools for detecting pulmonary nodules, which may be indicative of lung cancer. However, manual analysis of these images by radiologists is time-consuming and subject to inter-observer variability. In this perspective, this study presents software for the analysis of medical imaging exams using machine learning. These images are derived from chest tomography scans. The developed software performs the segmentation process of potential pulmonary nodules present in the image and classifies them as nodules or non-nodules, serving as a diagnostic aid tool. The development was carried out in Python language, using the PyTorch library to implement the convolutional neural network. The LIDC-IDRI database was used, which contains over 800 chest CT scans with annotations made by radiologists about the nodules present. As a result, the system demonstrated the ability to segment and classify 85% of the pulmonary nodules present in exams from the dataset accurately.

**Keywords:** artificial intelligence; deep learning; tomography.

## LISTA DE FIGURAS

|  |    |
|--|----|
| Figura 1 – Anatomia pulmonar . . . . .   | 21 |
| Figura 2 – Exame de tomografia em diferentes planos geométricos . . . . .  | 22 |
| Figura 3 – Imagem de Pulmão Humano com Aparente Edema Pulmonar . . . . .   | 23 |
| Figura 4 – Exemplos de comunicação via DICOM . . . . .   | 25 |
| Figura 5 – Modelo de paciente em DICOM . . . . .   | 26 |
| Figura 6 – Representação de um tensor . . . . .  | 27 |
| Figura 7 – Exame pulmonar com nódulo segmentado . . . . .  | 28 |
| Figura 8 – Exemplo de Modelo de Deep Learning . . . . .  | 29 |
| Figura 9 – Exemplo de Estrutura de CNN . . . . .   | 30 |
| Figura 10 – Convolução em uma Rede Neural Convolutiva . . . . .  | 31 |
| Figura 11 – Pooling em uma CNN . . . . .   | 31 |
| Figura 12 – Arquitetura da U-Net . . . . .   | 33 |
| Figura 13 – Fluxograma da visão geral do trabalho . . . . .  | 35 |
| Figura 14 – Operação de verificação de igualdade entre nódulo e anotação . . . . .   | 42 |
| Figura 15 – Algoritmo de determinação de igualdade entre anotações . . . . .   | 43 |
| Figura 16 – Algoritmo de montagem das tuplas de candidatos . . . . .   | 44 |
| Figura 17 – Fluxograma das funções de dimensionamento . . . . .  | 45 |
| Figura 18 – Diagrama do modelo de segmentação . . . . .  | 46 |
| Figura 19 – Diagrama de cada bloco <i>Backbone</i> . . . . .   | 47 |
| Figura 20 – Diagrama do modelo de classificação . . . . .  | 48 |
| Figura 21 – Esquema elaborado para treinamento e validação . . . . .   | 49 |
| Figura 22 – Distribuição dos resultados obtidos para o conjunto das 5 rodadas de<br>treinamento da segmentação . . . . .   | 57 |
| Figura 23 – Distribuição dos resultados obtidos para o conjunto das 5 rodadas de<br>validação da segmentação . . . . .     | 58 |
| Figura 24 – Distribuição dos resultados obtidos para o conjunto das 5 rodadas de<br>treinamento da classificação . . . . . | 60 |
| Figura 25 – Distribuição dos resultados obtidos para o conjunto das 5 rodadas de<br>validação da classificação . . . . .   | 61 |
| Figura 26 – Curva ROC e AUC do classificador . . . . .   | 62 |

|  |           |
|--|-----------|
| <b>Figura 27 – Exame com nódulo maligno segmentado . . . . .</b> | <b>64</b> |
| <b>Figura 28 – Exame com nódulo benigno segmentado . . . . .</b> | <b>64</b> |
| <b>Figura 29 – Exame com não-nódulo segmentado . . . . .</b>     | <b>65</b> |

## LISTA DE TABELAS

|   |           |
|---|-----------|
| <b>Tabela 1 – Tabela de Descrição dos Campos do Arquivo de Anotações da Lung Image Database Consortium - Image Database Resource Initiative (LIDC-IDRI) . . . . .</b> | <b>37</b> |
| <b>Tabela 2 – Resultados de treinamento do modelo de segmentação . . . . .</b>  | <b>55</b> |
| <b>Tabela 3 – Resultados da validação do modelo de segmentação . . . . .</b>  | <b>56</b> |
| <b>Tabela 4 – Resultados do treinamento do modelo de classificação . . . . .</b>  | <b>59</b> |
| <b>Tabela 5 – Resultados da validação do modelo de classificação . . . . .</b>  | <b>59</b> |
| <b>Tabela 6 – Resultados de validação do sistema . . . . .</b>  | <b>63</b> |

## LISTAGEM DE CÓDIGOS FONTE

|  |    |
|--|----|
| Listagem 1 – Criação de CNN utilizando PyTorch . . . . . | 34 |
|--|----|

## LISTA DE ABREVIATURAS E SIGLAS

### Siglas

|           |   |
|-----------|---|
| AC-GAN    | Auxiliary Classifier Generative Adversarial Network                 |
| AnoGAN    | Anomaly Generative Adversarial Network                              |
| AUC       | Area Under the Curve  |
| CAD       | Computer-Aided Design   |
| CAPES     | Coordenação de Aperfeiçoamento de Pessoal de Nível Superior         |
| CNN       | Convolutional Neural Network  |
| CSV       | Comma Separated Values  |
| DICOM     | Digital Imaging and Communications in Medicine                      |
| DL        | Deep Learning   |
| DPN       | Dual Path Network   |
| DPOC      | Doença Pulmonar Obstrutiva Crônica                                  |
| DTCNN     | Deep Transfer Convolutional Neural Network                          |
| ELM       | Extreme Learning Machine  |
| FAIR      | Facebook's AI Research lab  |
| FPR       | False Positive Rate   |
| GA2M      | Generalized Additive Model with Interactions                        |
| GPU       | Graphics Processing Unit  |
| HU        | Hounsfield Units  |
| LABIC     | Laboratório de Bioinformática e Inteligência Computacional          |
| LIDC-IDRI | Lung Image Database Consortium - Image Database Resource Initiative |
| NNU-Net   | No-New Net  |
| PACS      | Picture Archiving and Communication System                          |

|       |  |
|-------|--|
| ReLU  | Rectified Linear Unit                      |
| ROC   | Receiver Operating Characteristic          |
| SQL   | Structured Query Language                  |
| SSH   | Secure Socket Shell                        |
| TC    | Tomografia Computadorizada                 |
| TPR   | True Positive Rate                         |
| UTFPR | Universidade Tecnológica Federal do Paraná |

## SUMÁRIO

|            |   |           |
|------------|---|-----------|
| <b>1</b>   | <b>INTRODUÇÃO</b>                             | <b>16</b> |
| <b>1.1</b> | <b>Objetivos</b>                              | <b>18</b> |
| 1.1.1      | Objetivo geral                                | 18        |
| 1.1.2      | Objetivos específicos                         | 18        |
| <b>1.2</b> | <b>Estado da Arte</b>                         | <b>18</b> |
| <b>2</b>   | <b>REFERENCIAL TEÓRICO</b>                    | <b>21</b> |
| <b>2.1</b> | <b>Anatomia Pulmonar e Nódulos Pulmonares</b> | <b>21</b> |
| <b>2.2</b> | <b>Tomografia computadorizada</b>             | <b>22</b> |
| <b>2.3</b> | <b>Processamento de Imagens</b>               | <b>24</b> |
| 2.3.1      | DICOM   | 24        |
| 2.3.2      | Pré-processamento                             | 25        |
| 2.3.3      | Segmentação                                   | 26        |
| 2.3.4      | Classificação                                 | 28        |
| <b>2.4</b> | <b>Deep Learning</b>                          | <b>28</b> |
| 2.4.1      | Redes Neurais Convolucionais                  | 29        |
| 2.4.2      | Convolução                                    | 30        |
| 2.4.3      | Pooling                                       | 31        |
| 2.4.4      | Camadas totalmente conectadas                 | 32        |
| 2.4.5      | U-Net   | 32        |
| 2.4.6      | PyTorch                                       | 33        |
| <b>3</b>   | <b>METODOLOGIA</b>                            | <b>35</b> |
| <b>3.1</b> | <b>Visão Geral</b>                            | <b>35</b> |
| <b>3.2</b> | <b>Base de Dados</b>                          | <b>36</b> |
| <b>3.3</b> | <b>Ferramentas Computacionais Utilizadas</b>  | <b>38</b> |
| 3.3.1      | Arquitetura de Rede                           | 38        |
| <b>4</b>   | <b>DESENVOLVIMENTO</b>                        | <b>40</b> |
| <b>4.1</b> | <b>Pré-processamento</b>                      | <b>40</b> |
| 4.1.1      | Carregamento das anotações                    | 40        |
| 4.1.2      | Carregamento dos dados brutos                 | 42        |
| <b>4.2</b> | <b>Segmentação</b>                            | <b>44</b> |

|            |   |           |
|------------|---|-----------|
| 4.2.1      | Arquitetura de rede . . . . .                 | 45        |
| <b>4.3</b> | <b>Classificação . . . . .</b>                | <b>46</b> |
| 4.3.1      | Arquitetura de rede . . . . .                 | 47        |
| <b>4.4</b> | <b>Treinamento e validação . . . . .</b>      | <b>48</b> |
| <b>4.5</b> | <b>Métricas de desempenho . . . . .</b>       | <b>51</b> |
| 4.5.1      | Registro de Métricas . . . . .                | 52        |
| 4.5.2      | Gráfico ROC e AUC . . . . .                   | 53        |
| <b>4.6</b> | <b>Utilização do Cluster . . . . .</b>        | <b>54</b> |
| <b>5</b>   | <b>RESULTADOS E DISCUSSÃO . . . . .</b>       | <b>55</b> |
| 5.1        | Segmentação . . . . .                         | 55        |
| 5.2        | Classificação . . . . .                       | 58        |
| 5.3        | Curva ROC e AUC . . . . .                     | 61        |
| 5.4        | Validação quantitativa . . . . .              | 62        |
| 5.5        | Exemplos de nódulos classificados . . . . .   | 63        |
| 5.6        | Trabalhos com topologias diferentes . . . . . | 64        |
| <b>6</b>   | <b>CONSIDERAÇÕES FINAIS . . . . .</b>         | <b>67</b> |
|            | <b>REFERÊNCIAS . . . . .</b>                  | <b>69</b> |

## 1 INTRODUÇÃO

O câncer é uma das principais causas de morte no mundo, sendo responsável por cerca de 9,6 milhões de mortes em 2018 (OPAS/OMS, 2020). A doença de câncer pulmonar impõe um ônus significativo à saúde pública em todo o mundo e a detecção precoce é um fator crítico para aumentar as chances de tratamento eficaz e, por consequência, reduzir seu impacto devastador. O desenvolvimento de tecnologias voltadas para a identificação precoce do câncer é de suma importância, visto que estas podem auxiliar muito os médicos no diagnóstico da doença.

Afetando milhões de pessoas em todo o mundo, o câncer é uma condição de saúde que causa uma carga significativa de sofrimento e morte (MIDTHUN, 2013). Trata-se de uma doença complexa caracterizada pelo crescimento descontrolado de células anormais no corpo, que podem se espalhar para outras partes e órgãos, tornando-se potencialmente fatal (WHITTING; FFICM *et al.*, 2015). O câncer pode se manifestar em várias formas e afetar diferentes órgãos, tornando-o um desafio formidável para a comunidade médica e de pesquisa. Entre as inúmeras formas de câncer, o câncer de pulmão destaca-se como um dos tipos mais letais, sendo responsável por um grande número de óbitos a cada ano. Sua associação com o tabagismo e a exposição a substâncias nocivas torna-o um problema de saúde pública ainda mais premente. Portanto, a detecção precoce e o tratamento eficaz são cruciais para combater o câncer em todas as suas formas e minimizar seu impacto devastador na sociedade.

Sobre o câncer do pulmão, algumas considerações podem ser feitas para destacar a relevância e a necessidade de pesquisas neste tema. Em primeiro lugar, possui alto impacto na saúde pública, tendo o câncer de pulmão a maior taxa de mortalidade, apresentando o dobro do número de morte anuais (1,76 milhões) do que o câncer colorretal (0,86 milhão), o segundo colocado na escala de mortalidade entre os cânceres (OPAS/OMS, 2020). A detecção precoce de nódulos pulmonares, que frequentemente são indicativos dessa doença, é crucial para melhorar as taxas de sobrevivência e, conseqüentemente, reduzir o impacto devastador do câncer de pulmão na sociedade (PEREIRA; FERRARI, 2018).

Além disso, a interpretação de exames de tomografia de tórax, em busca de nódulos pulmonares, é uma tarefa intrinsecamente complexa e altamente dependente da experiência dos radiologistas. A introdução de sistemas baseados em Deep Learning (DL) pode aprimorar a precisão e a consistência no diagnóstico, reduzindo a variabilidade humana e garantindo resultados mais confiáveis (PAIVA, 2013).

O acesso crescente a grandes volumes de dados de exames médicos rotulados, juntamente com os avanços na capacidade de processamento por meio de Unidades de Processamento Gráfico, do inglês Graphics Processing Unit (GPU), torna viável o treinamento de modelos de DL em larga escala (COZMAN; PLONSKY; NERI, 2021). Essa disponibilidade de dados e recursos computacionais oferece a oportunidade de aprimorar a detecção de nódulos pulmonares e, por conseguinte, melhorar o diagnóstico médico.

Adicionalmente, o uso de DL representa uma inovação tecnológica importante na área da saúde. A automação da detecção de nódulos pulmonares pode acelerar o processo de diagnóstico e liberar os profissionais de saúde para se concentrarem em tarefas mais complexas e clínicas, além de reduzir o tempo necessário para a análise de imagens (WANG; CASALINO; KHULLAR, 2019).

Sendo assim, há o potencial para a redução de custos associados ao diagnóstico precoce de doenças, como o câncer de pulmão. A detecção precoce pode levar a tratamentos menos invasivos e, conseqüentemente, a uma redução nos custos de saúde, aliviando o ônus financeiro das famílias e sistemas de saúde (WANG; CASALINO; KHULLAR, 2019).

Exemplos relevantes para a aplicação de técnicas de DL na detecção de nódulos pulmonares incluem o estudo de Elnakib, Amer e Abou-Chadi (2020), que ilustra o sucesso da aplicação de algoritmos de DL na detecção de nódulos pulmonares em imagens de tomografia computadorizada. Neste trabalho, os autores testaram várias arquiteturas de DL diferentes e propõem uma arquitetura específica que apresenta precisão na detecção de 96,25%. A pesquisa apresentada por Wang (2022) oferece ainda outra perspectiva, demonstrando o uso de técnicas de DL para aprimorar o diagnóstico, abordando as vantagens e desafios desses sistemas, com o objetivo de desenvolver um Design Assistido por Computador, do inglês Computer-Aided Design (CAD), altamente sensível, eficiente e acessível. Estes e muitos outros exemplos acadêmicos demonstram a viabilidade de um sistema baseado em DL para auxílio médico na detecção de nódulos pulmonares.

Portanto, considerando a prevalência do câncer de pulmão, sua alta taxa de mortalidade, a complexidade dos exames de imagem e o potencial impacto na saúde pública e na economia, o trabalho a seguir delineia a concepção e o desenvolvimento de um sistema de segmentação e classificação por DL aplicado a exames de tomografia de tórax, com um enfoque específico na detecção de nódulos pulmonares.

Dada a natureza intrínseca dos exames de tomografia de tórax e a complexidade da interpretação humana, a utilização de técnicas de DL oferece um potencial promissor para melhorar a eficácia e a eficiência do diagnóstico. A capacidade de segmentar regiões de interesse, como os nódulos pulmonares, e classificá-las com precisão é de suma importância para um tratamento adequado e uma gestão otimizada dos pacientes.

A solução proposta neste trabalho emprega simplificações estratégicas para viabilizar a construção de um protótipo funcional de forma que, com pequenas alterações, o projeto possa ser integrado em ambiente produtivo de, por exemplo, uma aplicação de diagnóstico clínico.

## 1.1 Objetivos

### 1.1.1 Objetivo geral

Desenvolver um sistema de segmentação e classificação por DL para a segmentação e detecção de nódulos pulmonares em exames de tomografia de tórax, para fins de auxílio ao diagnóstico feito por médicos radiologistas.

### 1.1.2 Objetivos específicos

- Escolher uma base de dados para ser utilizada como base para as atividades de treinamento;
- Desenvolver um algoritmo capaz de receber os dados (exames de tomografia) da base;
- Implementar um algoritmo baseado em DL capaz de segmentar os dados recebidos;
- Desenvolver um algoritmo baseado em DL para classificar os exames segmentados recebidos e retornar regiões onde provavelmente existem nódulos pulmonares e uma probabilidade destes serem realmente nódulos; e
- Integrar todos os algoritmos em um único sistema e gerar um modelo de aprendizagem reproduzível.

## 1.2 Estado da Arte

Diversas pesquisas envolvendo segmentação de partes do corpo humano e classificação de estruturas anatômicas por meio de DL em exames de imagem foram publicadas nos últimos anos. Foi feito um estudo acerca de técnicas que poderiam ser utilizadas neste trabalho sobre os bancos de dados *Google Scholar* e Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES). Muitas apresentam similaridades interessantes com o domínio de estudo deste trabalho. Entre elas, destacam-se:

- Zeynettin *et al.* (2017), com uma abordagem informativa, delineia a respeito de estudos já feitos acerca de DL aplicado à segmentação de imagens cerebrais obtidas por tomografia e ressonância magnética. O trabalho elucida que existem diversas competições para desenvolvimento de tecnologias para tal área, além de explicar alguns conceitos de pré-processamento de imagens cranianas e as arquiteturas de redes neurais mais usadas.
- Esteva *et al.* (2017) treinou um modelo de DL em um conjunto de dados composto por 129.450 imagens clínicas de 2.032 doenças diferentes. O software foi testado em

comparação com 21 dermatologistas certificados, utilizando imagens clínicas comprovadas por biópsia. Dois casos críticos de classificação binária foram considerados: identificação de carcinomas queratinocíticos versus ceratoses seborreicas benignas, e identificação de melanomas malignos versus benignos. O modelo demonstrou um desempenho impressionante, alcançando resultados comparáveis aos de dermatologistas em ambas as tarefas.

- Beaulieu-Jones *et al.* (2018), que introduziu uma abordagem diferente para gerar dados clínicos sintéticos por meio do uso de redes adversárias generativas auxiliares, do inglês Auxiliary Classifier Generative Adversarial Network (AC-GAN), sob o conceito de privacidade diferencial. Essa técnica permite o compartilhamento de dados clínicos para análises secundárias, enquanto protege a privacidade dos participantes. Os resultados mostraram que os dados sintéticos gerados são similares o suficiente aos dados reais, permitindo que modelos de aprendizado de máquina construídos com os dados sintéticos fornecessem resultados comparáveis aos modelos treinados com dados reais. Além disso, a técnica foi avaliada por clínicos, demonstrando a eficácia da geração de dados sintéticos realistas enquanto preservava a privacidade dos participantes.
- Caruana *et al.* (2015) buscaram por modelos inteligíveis em aplicações de *machine learning* na área da saúde, onde a compreensibilidade e confiabilidade dos modelos são essenciais. Geralmente, modelos inteligíveis são considerados menos precisos, mas os autores demonstram que isso nem sempre é o caso. Eles aplicaram modelos aditivos generalizados com interações em pares, do inglês Generalized Additive Model with Interactions (GA2M) em dois estudos de caso em saúde, obtendo modelos inteligíveis com alta precisão. No primeiro caso, o modelo inteligível revelou padrões surpreendentes nos dados, permitindo a eliminação de complexidades em modelos anteriores. No segundo caso, os mesmos métodos foram escalados para conjuntos de dados extensos, mantendo a inteligibilidade e alcançando precisão comparável aos melhores modelos de DL, que são menos inteligíveis.
- Schlegl *et al.* (2017) apresentou o Anomaly Generative Adversarial Network (AnoGAN), uma abordagem de aprendizado não supervisionado para identificar anomalias em dados de imagens. Os autores propuseram um modelo baseado em redes generativas adversárias convolucionais profundas para aprender variações anatômicas normais. O grande diferencial desta abordagem é a capacidade de lidar com a falta de rótulos (do inglês, *labels*) em dados de imagem. Os autores desenvolveram um esquema de pontuação de anomalias que permite a identificação de imagens anômalas, como aquelas contendo fluído retiniano ou focos hiper-refletivos em imagens de tomografia de coerência óptica da retina.

- Isensee *et al.* (2021), que implementaram o No-New Net (NNU-Net), um algoritmo de segmentação de imagens biomédicas baseado em aprendizado profundo que se configura automaticamente para novas tarefas, incluindo etapas de pré-processamento, arquitetura de rede, treinamento e pós-processamento. O nome do modelo, que em tradução livre seria "Sem Nova Rede", vem do fato do modelo, no lugar de propor um novo modelo diferente, automaticamente configurar a segmentação por si só. O NNU-Net superou grande parte das abordagens existentes em competições internacionais de segmentação biomédica sem intervenção manual. Este método foi disponibilizado publicamente, tornando a segmentação acessível a um amplo público, sem a necessidade de conhecimento especializado ou recursos computacionais avançados.

## 2 REFERENCIAL TEÓRICO

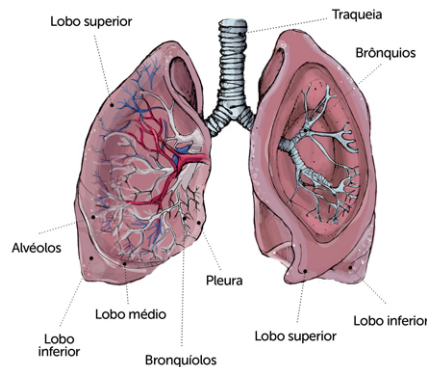
Esta seção descreve os principais tópicos teóricos envolvidos neste trabalho, como anatomia pulmonar, tomografia computadorizada, *Deep Learning* e redes neurais.

### 2.1 Anatomia Pulmonar e Nódulos Pulmonares

Os pulmões, apresentados na Figura 1, são dois órgãos cônicos, localizados no tórax, que são separados pelo mediastino e constituem um complexo sistema de estruturas que asseguram uma respiração eficiente. A traqueia se bifurca nos brônquios, que subsequentemente se ramificam em bronquíolos, responsáveis por conduzir o ar para dentro e para fora dos pulmões. Esta árvore brônquica constitui uma rede intrincada que se estende até os alvéolos, diminutos sacos de ar onde ocorre a troca gasosa entre o oxigênio e o dióxido de carbono (ISAACS; SYKES, 2002).

Ademais, os pulmões são envolvidos por duas camadas finas de tecido chamadas pleuras. A pleura visceral adere à superfície dos pulmões, enquanto a pleura parietal reveste a cavidade torácica. Estas camadas deslizam suavemente uma sobre a outra durante a respiração, garantindo um funcionamento eficaz (ISAACS; SYKES, 2002).

**Figura 1 – Anatomia pulmonar**



**Fonte: Prevenir.pt<sup>1</sup>(2014).**

Quando se trata de nódulos pulmonares, estes representam áreas de crescimento celular anormal nos pulmões. Eles podem ser benignos, como granulomas ou hamartomas, ou malignos, como carcinomas de células não pequenas ou carcinomas de pequenas células. A identificação e caracterização precisa destes nódulos são cruciais para o diagnóstico precoce de doenças pulmonares, especialmente o câncer de pulmão (LOVERDOS *et al.*, 2019).

<sup>1</sup> Disponível em <https://www.prevenir.pt/literacia-em-saude/corpo-humano/os-pulmoes/>

O diagnóstico de nódulos pulmonares envolve uma variedade de métodos, tais como radiografias de tórax, Tomografia Computadorizada (TC), broncoscopia e biópsia. A TC de alta resolução é frequentemente empregada para avaliar o tamanho, forma e textura dos nódulos. O tratamento de nódulos detectados varia de acordo com o tipo e o estágio da lesão. Nódulos benignos podem não necessitar de intervenção, enquanto nódulos malignos geralmente exigem cirurgia, quimioterapia ou radioterapia (LOVERDOS *et al.*, 2019).

Para exemplificar, o adenocarcinoma de pulmão é um tipo comum de câncer de pulmão que se origina nas células glandulares do tecido pulmonar. A doença, em seu primeiro estágio patológico, apresenta uma taxa de sobrevivência em 5 anos de 70% a 85% e, para estágios de metástase avançados, essa taxa de sobrevivência cai para menos de 30%. A detecção precoce do nódulo que origina essa e outras patologias é fundamental, já que o tratamento eficaz é mais provável quando a doença é diagnosticada em estágios iniciais (MYERS; WALLEN, 2023).

Em síntese, a anatomia pulmonar e a identificação de nódulos pulmonares são áreas interligadas de estudo que desempenham um papel crucial no diagnóstico e tratamento de doenças pulmonares. A compreensão profunda da anatomia pulmonar é essencial para avaliar a presença e a natureza dos nódulos, garantindo, assim, a saúde pulmonar e o bem-estar dos pacientes.

## 2.2 Tomografia computadorizada

A tomografia computadorizada é uma técnica de imagenologia médica avançada que permite a visualização detalhada das estruturas internas do corpo em seções transversais. Ao contrário da radiografia convencional, que projeta uma imagem tridimensional em uma superfície bidimensional, os exames de TC criam imagens em camadas que proporcionam uma visão mais precisa e detalhada das estruturas do tórax e de outras regiões do corpo, como ilustrado na Figura 2.

**Figura 2 – Exame de tomografia em diferentes planos geométricos**



**Fonte: Adaptado de Whiting, FFICM *et al.* (2015).**

A tomografia utiliza raios X para obter informações detalhadas sobre a composição e densidade dos tecidos. O *scanner* de TC realiza múltiplas medições de atenuação dos raios X a partir de diferentes ângulos de rotação ao longo de uma seção transversal do corpo (SPRAWLS,

1995). Esses dados são então utilizados para reconstruir uma imagem digital representando a seção transversal, em que cada pixel da imagem representa uma medição da atenuação média através da espessura do segmento. Essa medição é expressa em Unidades Hounsfield, do inglês Hounsfield Units (HU), em que a água, por exemplo, tem uma medição de zero na escala. Tecidos mais densos, como músculo, fígado e osso, têm valores HU positivos, enquanto tecidos menos densos, como pulmão e tecido adiposo, têm valores HU negativos (WHITING; FFICM *et al.*, 2015).

A densidade do tórax tem uma ampla gama de aplicações clínicas. É frequentemente usada para diagnosticar e avaliar condições pulmonares, como pneumonia, fibrose pulmonar, bronquiectasia e Doença Pulmonar Obstrutiva Crônica (DPOC). Além disso, a tomografia é uma ferramenta crucial para a avaliação e detecção de tumores pulmonares, da anatomia vascular e de anormalidades pleurais. Por exemplo, na Figura 3, a partir da imagem de tomografia do pulmão, pode-se deduzir que o paciente sofre de edema pulmonar. Esse diagnóstico pode ser feito pois a patologia é conhecida por induzir grandes manchas na imagem de TC do pulmão, o que indica que houve uma perda de definição nos vasos sanguíneos da região (ROSSER; FFICM *et al.*, 2016).

**Figura 3 – Imagem de Pulmão Humano com Aparente Edema Pulmonar**



**Fonte: Rosser, FFICM *et al.* (2016).**

É importante que os clínicos estejam cientes da radiação potencialmente prejudicial à qual os pacientes são expostos durante exames de tomografia computadorizada. Cada exame de TC envolve uma exposição à radiação ionizante, e os riscos e benefícios devem ser cuidadosamente avaliados, especialmente em pacientes que podem necessitar de múltiplos exames ao longo do tempo (SPRAWLS, 1995).

Dependendo da região do corpo examinada, diversos tipos de tumores podem ser identificados. Na região torácica, por exemplo, a tomografia pode revelar a presença de tumores pulmonares, que podem ser benignos ou malignos. Os tumores malignos, como o carcinoma broncogênico, são frequentemente detectados por meio de alterações na textura pulmonar, presença de massas ou nódulos visíveis na imagem de tomografia. Além disso, a tomografia computadorizada também é crucial na detecção de linfomas torácicos, que podem se manifestar como massas mediastinais ou nódulos nos pulmões (LOVERDOS *et al.*, 2019).

Interpretar exames de tomografia requer um conhecimento detalhado da anatomia humana. A capacidade de interpretar esses exames é essencial para médicos clínicos, anestesiólogos e intensivistas, especialmente quando radiologistas podem não estar disponíveis imediatamente. Uma abordagem sistemática, revisão do histórico do paciente e comparação com exames anteriores são etapas importantes para a interpretação precisa dos dados presentes no exame (SPRAWLS, 1995).

## 2.3 Processamento de Imagens

O processamento de imagens é uma área interdisciplinar que se concentra na manipulação e análise de imagens digitais para melhorar a qualidade, extrair informações significativas e facilitar a interpretação. Em aplicações médicas, como a detecção de nódulos pulmonares, o processamento de imagens desempenha um papel crucial na obtenção de informações diagnósticas precisas a partir das imagens de TC.

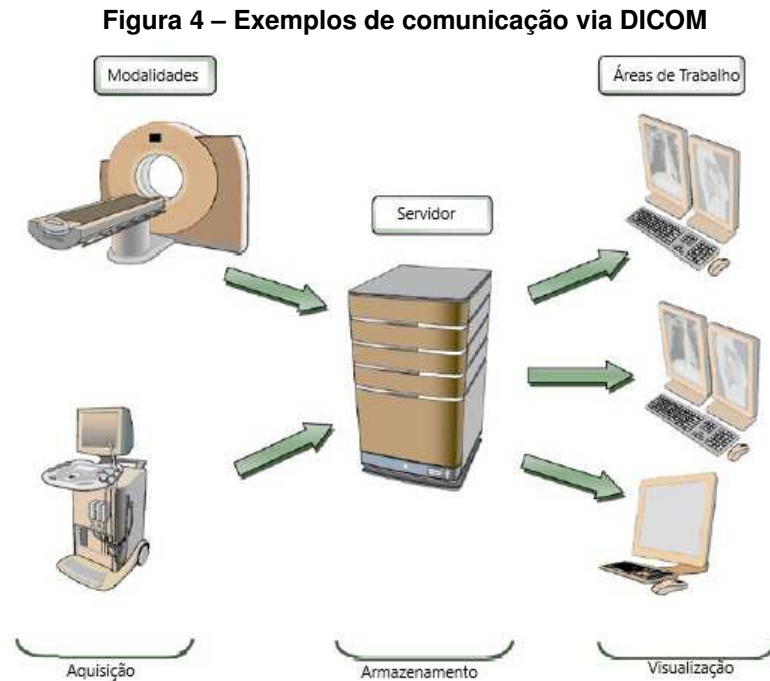
### 2.3.1 DICOM

O Digital Imaging and Communications in Medicine (DICOM) é uma padronização que atua como uma linguagem universal para imagens médicas, garantindo que as informações contidas nas imagens sejam interpretadas de maneira uniforme e precisa, independentemente da modalidade de imagem utilizada, seja tomografia computadorizada (TC), ressonância magnética (RM), radiografia, entre outras (BIDGOOD *et al.*, 1997).

O DICOM não apenas padroniza a representação das imagens como também serve como um meio de comunicação eficaz entre sistemas e profissionais de saúde. Ele permite que as imagens médicas sejam transmitidas e compartilhadas com segurança, mantendo a integridade dos dados. Isso é crucial em cenários onde diferentes instituições de saúde, clínicas e hospitais precisam colaborar e compartilhar informações de pacientes (PIANYKH, 2009).

Na Figura 4, é ilustrada uma rede interconectada de dispositivos médicos em uma configuração normalmente vista em clínicas médicas. Nesse cenário, sistemas como tomógrafos, ressonâncias magnéticas, radiografias e estações de trabalho médicas são capazes de transmitir e receber imagens e informações clínicas de maneira padronizada e harmoniosa. Essa imagem destaca a eficiência e a interoperabilidade que o DICOM proporciona, permitindo que diferentes equipamentos médicos, muitas vezes de fabricantes distintos, compartilhem dados de pacientes com segurança. Isso não apenas agiliza o fluxo de trabalho clínico, mas também garante que os médicos tenham acesso às informações essenciais para tomar decisões de tratamento informadas (BIDGOOD *et al.*, 1997).

Além disso, o DICOM inclui informações críticas relacionadas à aquisição das imagens, detalhando o equipamento de imagem usado e as configurações específicas do exame. Esses



**Fonte: Adaptado de Pianykh (2009).**

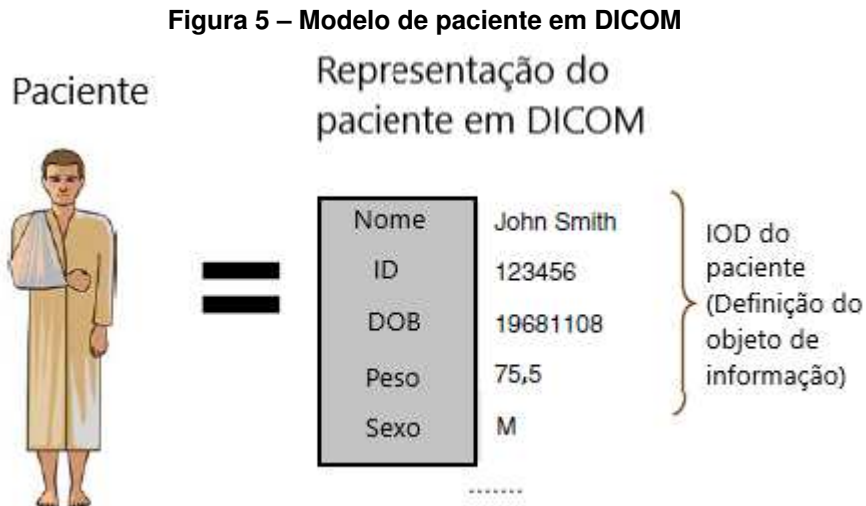
detalhes são essenciais para garantir a interpretação precisa das imagens pelos profissionais de saúde. Em um contexto em que a precisão diagnóstica é crucial, o DICOM desempenha um papel vital (KATHIRAVELU; SHARMA; SHARMA, 2021).

Adicionalmente, o DICOM possibilita a incorporação de informações clínicas diretamente nas imagens. Isso significa que, junto com a imagem, é possível incluir dados como histórico médico do paciente, informações sobre o exame e possíveis diagnósticos. Essa integração de informações clínicas com as imagens DICOM fornece um contexto valioso para os médicos, auxiliando-os na tomada de decisões de diagnóstico informadas e precisas (BIDGOOD *et al.*, 1997).

A Figura 5 traz a representação de um "Objeto Paciente DICOM". Esse objeto não só contém a imagem resultante do exame realizado pelo paciente, mas também encapsula informações clínicas essenciais do indivíduo, como seu histórico médico, dados demográficos, registros de exames e relatórios de diagnóstico.

### 2.3.2 Pré-processamento

O pré-processamento de imagens é fundamental para a análise eficaz. Antes de serem inseridas em programas e sistemas de análise, as imagens precisam ser ajustadas e transformadas conforme os requisitos específicos de entrada desses sistemas. Isso garante uma coleta de dados eficiente e minimiza a ocorrência de erros (SMITH; KAELIN, 2019).



**Fonte: Adaptado de Pianykh (2009).**

Uma das etapas típicas de pré-processamento é o redimensionamento das imagens para um tamanho uniforme. Os modelos de DL geralmente requerem que todas as imagens de entrada tenham as mesmas dimensões (SAPONARA; ELHANASHI, 2022).

Além disso, a normalização é uma prática comum no pré-processamento de imagens. A prática de normalização de imagens envolve a transformação dos valores de pixel para um intervalo específico, geralmente entre 0 e 1 ou -1 e 1. A normalização ajuda a melhorar a estabilidade e a convergência do treinamento do modelo (SMITH; KAELIN, 2019).

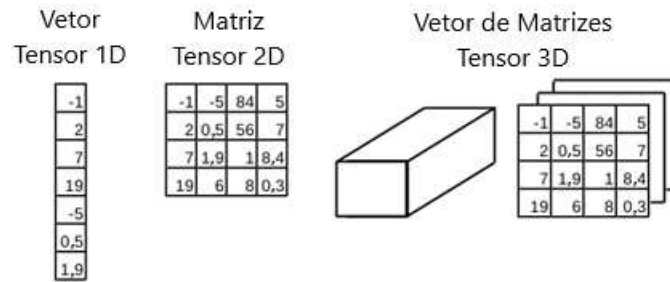
O aumento de dados, ou *data augmentation*, é outra técnica importante. O *PyTorch*, por exemplo, oferece funcionalidades para aplicar diversas transformações nas imagens de treinamento, como rotação, espelhamento e translação. Isso ajuda a diversificar o conjunto de dados, tornando o modelo mais robusto e capaz de lidar com variações nas imagens (PASZKE *et al.*, 2019).

O pré-processamento pode envolver, também, a conversão das imagens para tensores, o formato de dados que as bibliotecas de DL utilizam. Isso permite que as imagens sejam compatíveis com as operações e camadas das redes neurais convolucionais.

A Figura 6 representa um tensor, que pode ser visualizado como uma estrutura multi-dimensional de dados, frequentemente comparada a uma matriz, mas com a capacidade de estender-se para além de duas dimensões. Cada elemento em um tensor armazena informações, como valores numéricos, e sua posição nas diferentes dimensões do tensor determina o contexto desses valores.

### 2.3.3 Segmentação

A segmentação de imagens é uma técnica crucial no campo da análise de imagens, especialmente em tarefas como a detecção de nódulos pulmonares em tomografias computa-

**Figura 6 – Representação de um tensor**

**Fonte: Autoria própria (2023).**

dorizadas. Esta técnica consiste em identificar e destacar áreas de interesse em uma imagem, dividindo-a em regiões distintas com base em características específicas, como textura, cor ou intensidade de pixel (SHEN; WU; SUK, 2017).

No contexto da detecção de nódulos pulmonares, a segmentação desempenha um papel fundamental na identificação precisa dessas estruturas. Ela permite que o modelo se concentre apenas nas regiões suspeitas, reduzindo o ruído de fundo e fornecendo uma representação mais precisa dos nódulos (RONNEBERGER; FISCHER; BROX, 2015). Existem várias abordagens para a segmentação de imagens médicas, incluindo métodos baseados em limiar, técnicas de aprendizado profundo e algoritmos de contorno (STEVENS; ANTIGA; VIEHMANN, 2020).

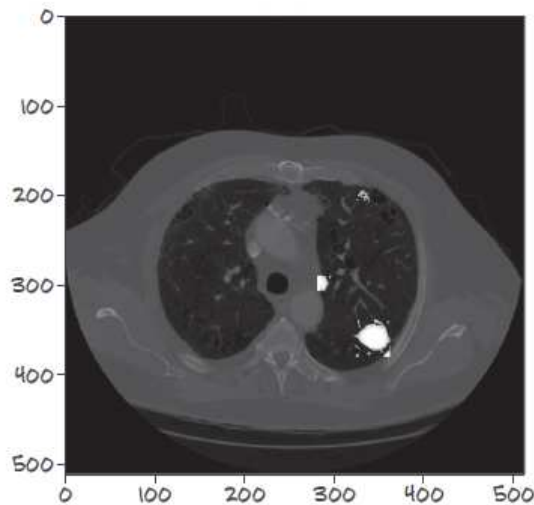
Métodos baseados em limiar envolvem a definição de um valor de limiar para separar as regiões de interesse do fundo. Essa abordagem é simples, mas pode ser sensível a variações na intensidade da imagem (OTSU, 1979).

As técnicas de aprendizado profundo, como as redes neurais convolucionais, revolucionaram a segmentação de imagens médicas. O conceito de Rede Neural Convolucional, do inglês Convolutional Neural Network (CNN) é capaz de aprender automaticamente as características relevantes das imagens, tornando-a eficaz na segmentação precisa de estruturas complexas, como os nódulos pulmonares (GOODFELLOW; BENGIO; COURVILLE, 2016).

A Figura 7 apresenta uma imagem de um pulmão onde um tumor foi segmentado por algoritmos de DL. A segmentação é um processo que envolve a identificação precisa das fronteiras do tumor, separando-o do tecido circundante. Essa representação visual é essencial para a análise médica, pois permite que os médicos e radiologistas avaliem visualmente o tamanho, a forma e a localização do tumor com precisão (PEREIRA; FERRARI, 2018).

Além disso, os algoritmos de contorno são usados para identificar as fronteiras das regiões de interesse. Eles são frequentemente aplicados após a segmentação inicial para refinar as margens das estruturas detectadas (CANNY, 1986).

**Figura 7 – Exame pulmonar com nódulo segmentado**



**Fonte: Stevens, Antiga e Viehmann (2020).**

#### 2.3.4 Classificação

A classificação de imagens médicas é uma das etapas mais importantes no processamento de imagens médicas. Essa técnica envolve atribuir uma ou mais categorias ou rótulos a uma imagem com base em suas características e conteúdo visual. No contexto da detecção de nódulos pulmonares, a classificação é frequentemente usada para determinar se uma região da imagem contém ou não um nódulo maligno.

Uma abordagem comum para a classificação de imagens médicas é o uso de modelos de aprendizado profundo, como as Redes Neurais Convolucionais (CNNs). Essas redes são treinadas em um grande conjunto de dados de imagens rotuladas, onde aprendem a identificar padrões complexos e representações relevantes para as categorias de interesse. À medida que são alimentadas com novas imagens, as CNNs são capazes de classificá-las com base nas características aprendidas durante o treinamento (SHEN; WU; SUK, 2017).

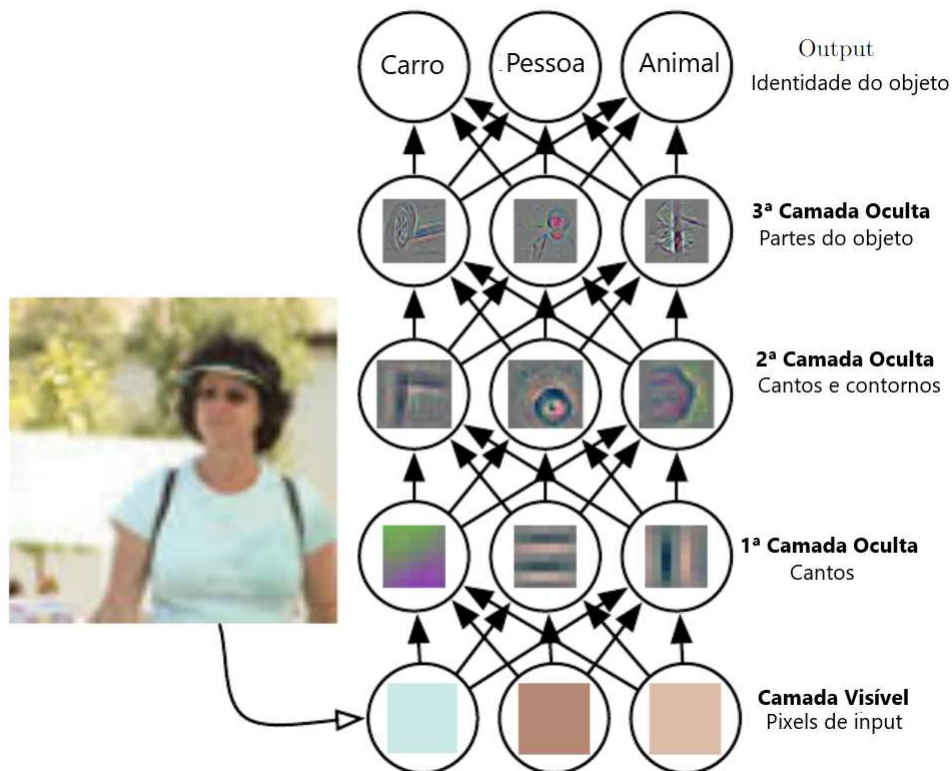
A escolha das características e do conjunto de dados de treinamento desempenha um papel fundamental na classificação precisa de imagens médicas. Os radiologistas e especialistas em imagem desempenham um papel importante na rotulagem de imagens para treinamento, garantindo que as categorias sejam definidas com precisão (SHEN; WU; SUK, 2017).

## 2.4 Deep Learning

*Deep Learning* é uma subárea de aprendizado de máquina que tem revolucionado a forma de tratamento de dados complexos. Esta abordagem, inspirada na estrutura do cérebro humano, tem se destacado em tarefas como reconhecimento de padrões, processamento de linguagem natural e análise de imagens (BEAULIEU-JONES *et al.*, 2018).

A Figura 8 ilustra um exemplo de modelo de DL, destacando suas camadas e conexões. A camada de entrada recebe dados brutos, que são passados por várias camadas ocultas, onde as transformações matemáticas ocorrem. Cada camada aprende representações cada vez mais abstratas dos dados, e a camada de saída produz as previsões ou resultados desejados. O treinamento do modelo envolve a otimização dos pesos das conexões entre neurônios para minimizar a diferença entre as previsões do modelo e os rótulos reais dos dados de treinamento (GOODFELLOW; BENGIO; COURVILLE, 2016).

**Figura 8 – Exemplo de Modelo de Deep Learning**



**Fonte: Adaptado de Goodfellow, Bengio e Courville (2016).**

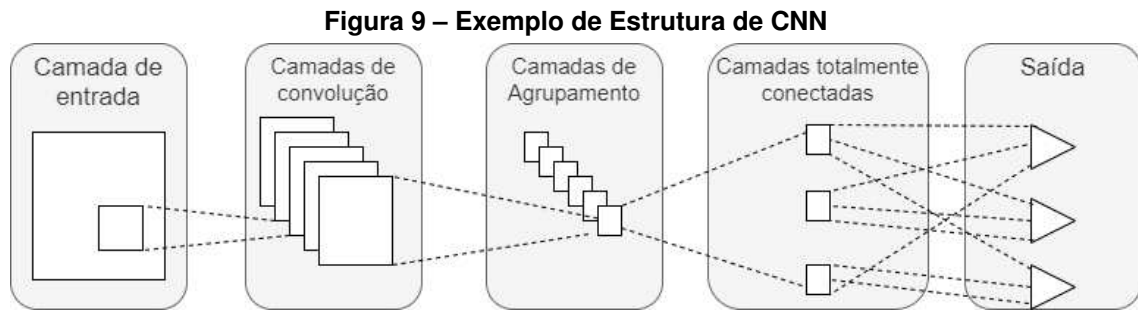
O DL tem uma ampla gama de aplicações, incluindo assistentes virtuais, veículos autônomos, auxílio no diagnóstico médico e muito mais. No contexto da detecção de nódulos pulmonares em tomografias computadorizadas, o DL desempenha um papel crucial, permitindo a criação de modelos sofisticados que podem identificar essas áreas de interesse com alta precisão.

#### 2.4.1 Redes Neurais Convolucionais

As CNNs desempenham um papel fundamental no cenário da inteligência artificial desenvolvida por meio de DL.

Na Figura 9, pode-se observar uma representação esquemática de uma arquitetura típica de uma CNN. Esta arquitetura é composta por várias camadas interconectadas, come-

çando com camadas de convolução e *pooling*, seguidas por camadas totalmente conectadas. As camadas de convolução são responsáveis por extrair características das imagens de entrada, enquanto as camadas de *pooling* reduzem a dimensionalidade das representações. Finalmente, as camadas totalmente conectadas realizam a classificação final com base nas características extraídas. Essa estrutura exemplifica a hierarquia de processamento de informações em uma CNN, onde as informações são refinadas e transformadas em recursos cada vez mais abstratos, permitindo que a rede tome decisões altamente complexas (KIM, 2019).



Fonte: Autoria própria (2023).

#### 2.4.2 Convolução

A convolução é uma operação matemática que combina duas funções para criar uma terceira função que representa como uma delas influencia a outra (GOODFELLOW; BENGIO; COURVILLE, 2016). Ela é definida pela equação 1:

$$(f * g)(s) = \int_{-\infty}^{\infty} f(x)g(s - x)dx \quad (1)$$

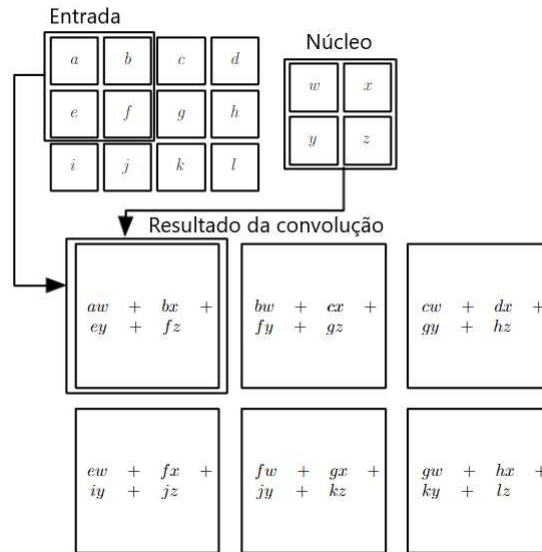
Nesta equação, tem-se:

- $(f * g)(s)$  é o resultado da convolução entre as funções  $f(x)$  e  $g(x)$  em um ponto  $s$ .
- $\int_{-\infty}^{\infty}$  denota uma integral que avalia a sobreposição das duas funções em todo o domínio.

A operação de convolução envolve multiplicar uma das funções ( $f(x)$ ) pela outra ( $g(x)$ ) após esta última ter sido "inversa" e deslocada ao longo do eixo  $x$  por  $s$  unidades. O resultado é então integrado sobre todos os valores possíveis de  $x$ , produzindo o valor da convolução em  $s$  (KIM, 2019).

Agora, considerando o contexto das CNNs, a operação de convolução é aplicada a imagens de entrada usando um filtro (ou *kernel*) deslizante. Neste caso,  $f(x)$  representa a imagem de entrada e  $g(x)$  é o filtro. A convolução é realizada aplicando o filtro à imagem e deslocando-o em várias posições (KIM, 2019).

**Figura 10 – Convolução em uma Rede Neural Convolutacional**



Fonte: Adaptado de Goodfellow, Bengio e Courville (2016).

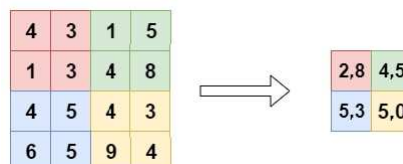
Na Figura 10, pode-se observar a operação de convolução em uma CNN. O filtro desliza pela imagem de entrada, multiplicando e somando os valores correspondentes em cada posição. Isso produz um mapa de características que destaca padrões relevantes, como bordas, texturas e formas (SHEN; WU; SUK, 2017).

#### 2.4.3 Pooling

Além disso, as CNNs geralmente incluem camadas de *pooling*, que são aplicada após a convolução e envolvem a subdivisão do mapa de características em regiões, mantendo apenas os valores máximos ou médios em cada região. Isso reduz a dimensionalidade da representação, tornando-a mais gerenciável (GOODFELLOW; BENGIO; COURVILLE, 2016).

Na Figura 11, exemplifica-se uma operação de *pooling*. Observa-se que são selecionados os valores médios dentro de uma janela 2x2, e este valor passa a ser o único valor na matriz de saída, transformando a janela em uma unidade de representação.

**Figura 11 – Pooling em uma CNN**



Fonte: Autoria própria (2023).

O *Max Pooling* e o *Average Pooling* são as duas técnicas mais comuns para esta operação em redes neurais convolucionais. O *Max Pooling* seleciona o valor máximo dentro de uma janela, como 2x2 ou 3x3, enquanto percorre a imagem. Essa abordagem destaca características

mais proeminentes, preservando informações relevantes e descartando o restante, resultando em uma redução eficaz da resolução. Já o *Average Pooling* calcula a média dos valores dentro da janela, proporcionando uma visão mais suavizada da informação, o que pode ser útil em alguns contextos para reduzir o impacto de características extremas e manter uma representação mais geral da imagem. Ambos os métodos desempenham um papel crucial na redução de dimensionalidade das representações de imagem, contribuindo para a eficiência computacional e focalizando em características essenciais para o processamento subsequente na rede neural (STEVENS; ANTIGA; VIEHMANN, 2020).

#### 2.4.4 Camadas totalmente conectadas

Além das camadas de convolução e de *pooling* outro componente fundamental em CNNs são as camadas totalmente conectadas. Estas camadas desempenham um papel crucial na tarefa de classificação final. Enquanto as camadas de convolução e *pooling* são responsáveis pela extração de características das imagens, as camadas totalmente conectadas estão envolvidas na tomada de decisões finais com base nessas características.

Em uma camada totalmente conectada, cada neurônio está conectado a todos os neurônios da camada anterior. Isso significa que todos os valores das características extraídas anteriormente são considerados em todas as unidades da camada totalmente conectada. Essa conexão total permite que a rede aprenda relações complexas entre as características e faça uma decisão final sobre a presença ou ausência de determinada característica que deseja-se localizar (GOODFELLOW; BENGIO; COURVILLE, 2016).

As camadas totalmente conectadas desempenham um papel vital na capacidade da CNN de aprender a partir dos dados e identificar padrões complexos. A combinação das informações de todas as características extraídas ao longo das camadas convolucionais e de *pooling* permite que a rede auxilie na tomada de decisões de forma cada vez mais precisa sobre a presença de padrões nas imagens (GOODFELLOW; BENGIO; COURVILLE, 2016).

#### 2.4.5 U-Net

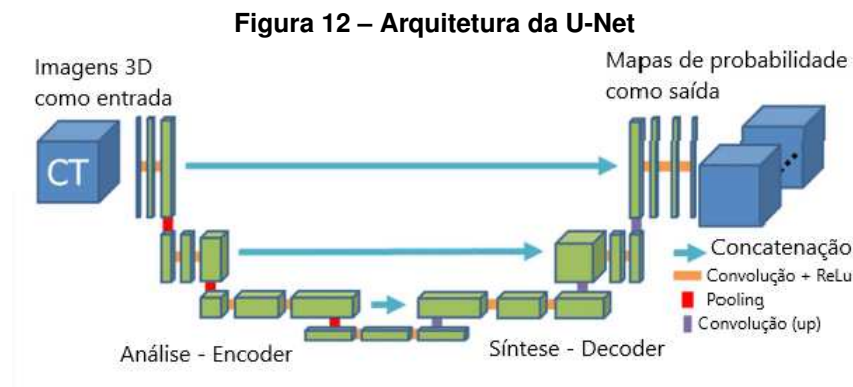
A U-Net é uma arquitetura de rede neural convolucional que tem se destacado como uma ferramenta poderosa para a segmentação de imagens, incluindo a detecção de nódulos pulmonares em exames de tomografia (SHEN; WU; SUK, 2017).

A U-Net foi projetada especificamente para tarefas de segmentação de imagens médicas. Ela possui uma arquitetura única, composta por uma parte descendente (*encoder*) que captura características da imagem em diferentes níveis de abstração e uma parte ascendente (*decoder*) que reconstrói a imagem segmentada com base nessas características (RONNEBERGER; FISCHER; BROX, 2015).

O que torna a U-Net especialmente eficaz é a sua capacidade de preservar detalhes espaciais importantes durante a segmentação. Isso é alcançado através da conexão direta entre camadas no *encoder* e no *decoder*, o que permite que informações detalhadas sejam transmitidas entre eles. Essa abordagem é fundamental para a segmentação precisa de nódulos pulmonares em imagens de tomografia, onde pequenos detalhes podem ser indicativos de doenças (STEVENS; ANTIGA; VIEHMANN, 2020).

A U-Net também é altamente flexível e pode ser adaptada para diferentes tarefas de segmentação. Ela é amplamente usada em pesquisa médica devido à sua eficácia na identificação de estruturas anatômicas e patológicas (WANG, 2022).

O esquema resumido da arquitetura da U-Net pode ser visto na Figura 12, onde fica explícito o funcionamento aqui descrito e, também, o formato em "U" da arquitetura, que serviu como inspiração para o nome do modelo.



#### 2.4.6 PyTorch

*PyTorch* é uma biblioteca de código aberto amplamente adotada para aplicações de DL. Ela foi desenvolvida pelo *Facebook's AI Research lab* (FAIR) e ganhou destaque na comunidade de pesquisa e desenvolvimento de inteligência artificial devido à sua flexibilidade, desempenho e abordagem intuitiva.

Uma das características marcantes do *PyTorch* é seu paradigma de programação dinâmica, que se diferencia de algumas outras bibliotecas de DL que usam gráficos computacionais estáticos. No *PyTorch*, os gráficos são construídos à medida que o código é executado. Isso proporciona uma maior flexibilidade aos desenvolvedores, permitindo que eles definam e modifiquem as redes neurais de maneira mais intuitiva, sem a necessidade de uma estrutura de gráfico pré-definida (PASZKE *et al.*, 2019).

A principal razão pela qual o *PyTorch* é amplamente utilizado na comunidade de DL é sua simplicidade e clareza na construção de redes neurais. Pode-se citar como exemplo a criação de redes neurais totalmente conectadas (ou perceptrons de múltiplas camadas).

Na Listagem 1, pode ser visualizado um exemplo de código que utiliza *PyTorch* para criar uma CNN. Na listagem, pode-se observar que a tarefa de criar uma CNN é completada ao executar *MinhaRedeNeural()* com as dimensões apropriadas para as camadas desejadas para a rede.

#### Listagem 1 – Criação de CNN utilizando PyTorch

```

1 import torch
2 import torch.nn as nn
3
4 # Definindo a arquitetura da rede
5 class MinhaRedeNeural(nn.Module):
6     def __init__(self, entrada_dim, oculta_dim, saida_dim):
7         super(MinhaRedeNeural, self).__init__()
8         self.camada_entrada = nn.Linear(entrada_dim, oculta_dim)
9         self.camada_oculta = nn.Linear(oculta_dim, saida_dim)
10        self.funcao_ativacao = nn.ReLU()
11
12        def forward(self, x):
13            x = self.camada_entrada(x)
14            x = self.funcao_ativacao(x)
15            x = self.camada_oculta(x)
16            return x
17
18 # Criando um objeto CNN
19 entrada_dim, oculta_dim, saida_dim = 10, 20, 5
20 minha_rede = MinhaRedeNeural(entrada_dim, oculta_dim, saida_dim)
21
22 # Exibindo a arquitetura da rede
23 print(minha_rede)

```

Fonte: Autoria própria (2023).

O ecossistema do *PyTorch* também é notável, com uma ampla gama de extensões e ferramentas disponíveis para tarefas específicas, como processamento de imagens, processamento de linguagem natural, visão computacional, entre outras. Entre essas extensões, destaca-se o *torchvision*, que oferece uma variedade de modelos de redes neurais pré-treinados e utilitários para tarefas de visão computacional (PASZKE *et al.*, 2019).

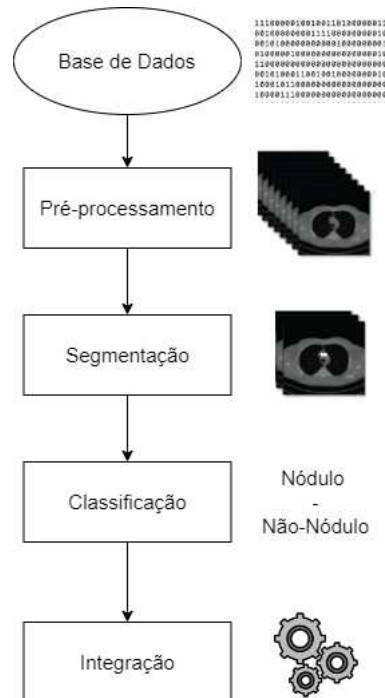
### 3 METODOLOGIA

Neste capítulo são descritos os procedimentos utilizados e as etapas que foram seguidas ao longo do desenvolvimento do projeto, desde a coleta de dados até a classificação dos resultados do algoritmo.

#### 3.1 Visão Geral

Inicialmente foram definidas as etapas que seriam tomadas para o desenvolvimento do sistema de ponta a ponta, como demonstra a Figura 13.

**Figura 13 – Fluxograma da visão geral do trabalho**



**Fonte: Autoria própria (2024).**

Em primeiro lugar, foi escolhida a base de dados LIDC-IDRI. A escolha foi feita em função da reputação da base, visto que é uma das mais abrangentes e clinicamente relevantes fontes de imagens de TC e anotações médicas disponíveis (ARMATO *et al.*, 2011).

A LIDC-IDRI fornece um conjunto de dados rico em variação de casos, permitindo um treinamento mais abrangente e preciso de modelos de detecção de doenças pulmonares. Além disso, o fato da base de dados ser acompanhada de anotações médicas torna-a um recurso valioso para validação e referência clínica, garantindo que o auxílio ao diagnóstico provido no final pelo sistema seja clinicamente relevante.

Em seguida, desenvolveu-se um algoritmo capaz de fazer o pré-processamento dos dados dentro da base escolhida, ou seja, transformar esta base em dados utilizáveis pelas partes seguintes do projeto. Após isso, implementou-se um algoritmo com a função de segmentar os

exames de tomografia em regiões de interesse que, neste caso, são regiões onde provavelmente haveriam nódulos pulmonares. Esta segmentação teve como grande objetivo simplificar e reduzir o *input* da parte de classificação, para que se utilizasse menos recursos computacionais ao treinar o classificador. Em seguida, desenvolveu-se o modelo de classificação, com o objetivo de receber as regiões de interesse segmentadas e classificar cada uma em nódulo ou não-nódulo. Após a parte de desenvolvimento, começou-se a fazer o treinamento dos modelos desenvolvidos com a base de dados escolhida.

### 3.2 Base de Dados

A base de dados LIDC-IDRI é um recurso de extrema importância na pesquisa em radiologia e detecção de câncer de pulmão. Ela oferece uma coleção de imagens de TC de tórax e anotações correspondentes, desempenhando um papel fundamental no desenvolvimento e avaliação de algoritmos de detecção de lesões pulmonares (ARMATO *et al.*, 2011).

A base contém um total de 1018 varreduras torácicas de TC helicoidais coletadas retrospectivamente, com aprovação local apropriada pelos comitês responsáveis, dos sistemas de comunicação e arquivamento de imagens, do inglês Picture Archiving and Communication System (PACS), das sete instituições acadêmicas participantes. Um *software* de anonimização foi aplicado para remover todas as informações de saúde protegidas contidas nos cabeçalhos DICOM das imagens. Nenhuma varredura foi realizada especificamente para o propósito da base de dados, de modo que uma gama heterogênea de modelos de *scanners* e parâmetros técnicos foi intencionalmente representada. A intenção era incluir apenas uma varredura de cada paciente para que as varreduras na base não fossem correlacionadas. Como resultado, a base LIDC-IDRI não é passível de análise de mudanças temporais nos exames, propriedade esta que não é necessariamente interessante para o desenvolvimento deste trabalho (ARMATO *et al.*, 2007).

Cada varredura selecionada para a base deveria ter uma colimação e intervalo de reconstrução de no máximo 3 mm e não foram impostos requisitos quanto à inclinação do *scanner*, exposição, tensão do tubo ou algoritmo de reconstrução. As varreduras foram limitadas a aproximadamente seis nódulos pulmonares com dimensão máxima inferior a 30 mm — consistente com o limite superior aceito do tamanho do nódulo — e maior ou igual a 3 mm — um limite inferior imposto por considerações práticas.

As 1018 varreduras de TC foram adquiridas de 1010 pacientes diferentes. Posteriormente, foi determinado retrospectivamente que duas varreduras distintas de cada um de oito pacientes haviam sido inadvertidamente incluídas entre as 1018 varreduras. No entanto, essas varreduras foram mantidas na base, uma vez que o esforço para a anotação de imagens já havia sido investido; os usuários da base podem identificar esses casos pelo ID de paciente comum nos cabeçalhos de imagem respectivos. Uma variedade de fabricantes e modelos de *scanners* foi representada — 670 varreduras de sete modelos diferentes do GE Medical Systems LightS-

peed, 74 varreduras de quatro modelos diferentes do Philips Brilliance, 205 varreduras de cinco modelos diferentes do Siemens Definition, Emotion e Sensation, e 69 varreduras dos *scanners* Toshiba Aquilion. As energias máximas do tubo usadas para aquisição de varredura foram as seguintes: 120 kV (n=818), 130 kV (n=31), 135 kV (n=69) e 140 kV (n=100) (ARMATO *et al.*, 2007).

O destaque da LIDC-IDRI é a anotação detalhada de lesões pulmonares. Vários radiologistas especializados foram escolhidos para analisar as imagens contidas na base, identificar e rotular cuidadosamente as lesões em cada exame. Cada lesão é caracterizada de acordo com sua natureza, tamanho, localização e outros atributos relevantes. Essas anotações são essenciais para treinar e validar algoritmos de detecção assistida pelo computador de lesões pulmonares, como o projeto desenvolvido neste trabalho (ARMATO *et al.*, 2011).

O arquivo de anotações é composto por informações detalhadas sobre cada possível nódulo na base de dados LIDC-IDRI. Na Tabela 1 estão descritos os campos de dados presentes no arquivo, bem como suas descrições e unidade de representação.

**Tabela 1 – Tabela de Descrição dos Campos do Arquivo de Anotações da LIDC-IDRI**

| Nome da Propriedade | Descrição   | Unidade         |
|---------------------|---|-----------------|
| seriesuid           | Identificador único da série de imagens do paciente           | -               |
| coordX              | Coordenada X do centro do nódulo no espaço da imagem          | Pixels          |
| coordY              | Coordenada Y do centro do nódulo no espaço da imagem          | Pixels          |
| coordZ              | Coordenada Z do centro do nódulo no espaço da imagem          | Pixels          |
| diameter_mm         | Diâmetro estimado do nódulo                                   | Milímetros (mm) |
| mal_bool            | Indicação da natureza do nódulo (maligno ou benigno)          | Booleano        |
| bboxLowX            | Coordenada X do canto inferior esquerdo da caixa delimitadora | Pixels          |
| bboxLowY            | Coordenada Y do canto inferior esquerdo da caixa delimitadora | Pixels          |
| bboxLowZ            | Coordenada Z do canto inferior esquerdo da caixa delimitadora | Pixels          |
| bboxHighX           | Coordenada X do canto superior direito da caixa delimitadora  | Pixels          |
| bboxHighY           | Coordenada Y do canto superior direito da caixa delimitadora  | Pixels          |
| bboxHighZ           | Coordenada Z do canto superior direito da caixa delimitadora  | Pixels          |

**Fonte: Autoria Própria (2023).**

Além das anotações, a base também disponibiliza arquivos no formato .MHD e .RAW, que se referem aos exames de tomografia propriamente ditos. Os arquivos .MHD são arquivos de metadados que contêm informações sobre as imagens médicas, como dimensões, resolução e tipo de dados. Os arquivos .RAW contêm os dados brutos da imagem em formato binário. Juntos, cada par de arquivos .MHD e .RAW representam um exame de TC em sua forma não processada (REEVES *et al.*, 2007).

Essas informações são fundamentais para a pesquisa em radiologia e no desenvolvimento de algoritmos de detecção de nódulos pulmonares. A combinação desses atributos permite que os pesquisadores localizem e caracterizem nódulos pulmonares nas imagens de TC, facilitando a análise e o diagnóstico de condições pulmonares, incluindo o câncer de pulmão.

A LIDC-IDRI desempenha um papel crucial na pesquisa médica, permitindo treinar e validar algoritmos avançados de detecção das mais variadas lesões pulmonares, comparar o desempenho de diferentes algoritmos em cenários controlados e explorar técnicas avançadas de processamento de imagem e análise de características para aprimorar a detecção de câncer de pulmão.

O estudo feito por Causey *et al.* (2017) desenvolveu o modelo *NoduleX* para prever a malignidade de nódulos pulmonares com alta precisão. Utilizando 1065 nódulos com diferentes pontuações de malignidade, o modelo alcançou um *AUC* de 0,974 em uma configuração e 0,993 quando combinado com características radiômicas quantitativas, demonstrando a eficácia do uso de características avançadas na melhoria da precisão do modelo.

O trabalho de Hussein *et al.* (2017) apresenta uma abordagem inovadora para a caracterização de nódulos pulmonares usando técnicas de DL. O método proposto combina uma rede neural convolucional multi-visão CNN com um processo gaussiano para classificar os nódulos pulmonares em benignos ou malignos. A rede CNN é utilizada para extrair características discriminativas dos nódulos a partir de diferentes vistas, enquanto o processo gaussiano é aplicado para modelar a incerteza na classificação e melhorar a robustez do sistema. O modelo apresentou uma acurácia de 92,31%, superando métodos anteriores e mostrando potencial para auxiliar na detecção precoce e diagnóstico de câncer de pulmão. A abordagem multi-visão e o uso do processo gaussiano foram fundamentais para capturar a variabilidade nos dados e lidar com as incertezas inerentes à interpretação de imagens médicas.

### 3.3 Ferramentas Computacionais Utilizadas

O processamento de dados, contando com as etapas de pré-processamento, segmentação e classificação, foi desenvolvido em *Python*, utilizando majoritariamente a biblioteca *PyTorch*. Os algoritmos foram reproduzidos no ambiente de *cluster* do Laboratório de Bioinformática e Inteligência Computacional (LABIC) da Universidade Tecnológica Federal do Paraná (UTFPR), que conta com várias máquinas acessáveis remotamente com poder de processamento gráfico elevado.

#### 3.3.1 Arquitetura de Rede

Para o desenvolvimento dos modelos de classificação e segmentação, foram utilizadas arquiteturas de rede específicas, cada uma adequada às suas respectivas tarefas.

O modelo de classificação foi construído com uma estrutura composta por cauda, *backbone* e cabeça. A cauda é responsável por lidar com a saída da camada de segmentação e consiste em uma camada de normalização em lote. O *backbone* é formado por quatro blocos de convolução, cada um contendo duas convoluções  $3 \times 3$  seguidas por uma ativação Rectified

Linear Unit (ReLU) e uma operação de *max-pooling*. Essa estrutura é responsável por extrair características relevantes da imagem de entrada e reduzir a dimensionalidade dos recursos. Por fim, a cabeça da rede recebe a saída do *backbone*, aplica uma camada linear de camadas totalmente conectadas e finaliza com uma camada de função *Softmax* para obter a classificação final.

Para a segmentação, foi adotada a arquitetura U-Net, conhecida por sua eficácia em segmentação de imagens médicas. O modelo foi adaptado para incluir uma camada de normalização em lote antes da U-Net, visando aprimorar o treinamento e lidar com variações nos lotes de treinamento. A U-Net, que é o núcleo do modelo de segmentação, gera um mapa de probabilidade que indica a probabilidade de cada pixel pertencer a um nódulo pulmonar. Uma camada sigmoide é aplicada à saída da U-Net para restringir os valores de saída ao intervalo  $[0, 1]$ , adequados para representar probabilidades.

A integração das arquiteturas de classificação e segmentação é realizada de maneira que a saída da camada de segmentação (mapa de probabilidade) é diretamente utilizada como entrada para a cauda do modelo de classificação. Essa abordagem permite que o modelo de classificação utilize as informações de segmentação para realizar uma classificação mais precisa dos nódulos pulmonares, limitando o escopo do que é utilizado de fato para a classificação.

## 4 DESENVOLVIMENTO

Neste capítulo, as etapas referentes ao desenvolvimento deste trabalho são descritas, detalhando principalmente como foram feitos os algoritmos de pré-processamento dos dados da base e os modelos de segmentação e classificação. O código do projeto desenvolvido para este trabalho está disponível na plataforma *GitHub*, acessível pelo link (<https://github.com/sSpicey/project-viper>).

### 4.1 Pré-processamento

O objetivo desta etapa é preparar os dados da base para serem utilizados nos modelos de DL a serem desenvolvidos. Esta atividade inclui a conversão dos arquivos de metadados e dados brutos em um formato adequado para análise. Além disso, é importante realizar a transformação das unidades do sistema de coordenadas do paciente para o referencial de localização utilizado pelo *Python*. O pré-processamento também envolve a carga de dados de anotação que fornecem informações sobre as coordenadas dos nódulos e sua malignidade (STEVENS; ANTIGA; VIEHMANN, 2020).

O objetivo final do pré-processamento é criar um conjunto de dados de treinamento que inclua uma amostra 3D do TC, informações sobre a presença de nódulos, o identificador exclusivo da série TC e o índice da amostra no conjunto de dados. Isso permite que um modelo de aprendizado de máquina seja treinado com dados bem estruturados e relevantes para a detecção de nódulos nas imagens.

#### 4.1.1 Carregamento das anotações

O pré-processamento dos dados fornecidos pela base LIDC-IDRI se inicia com o carregamento desses dados. Em vez de carregar cada TC individualmente, a abordagem adotada foi de analisar os arquivos em formato Comma Separated Values (CSV) fornecidos pela base. Esses arquivos contêm informações sobre pontos de interesse em cada exame, incluindo coordenadas, indicações de se o ponto é um nódulo e um identificador único para cada tomografia.

O arquivo CSV *candidates* contém informações sobre estruturas anatômicas que podem se assemelhar a nódulos, sejam eles malignos, benignos ou outros tipos. Essas informações formaram a base para criar uma lista completa de candidatos que será posteriormente dividida em conjuntos de treinamento e validação. Cada linha deste arquivo inclui um *series\_uid* (identificador da série TC), coordenadas (X, Y, Z) e uma coluna de classe que indica o *status* do nódulo. A coluna de classe é uma variável booleana, com o valor "0" indicando que o candidato não é um nódulo real e "1" indicando que é um nódulo, podendo ser maligno ou benigno. Há um total de 1.351 candidatos marcados como nódulos reais.

Já o arquivo CSV *annotations* contém informações sobre alguns dos candidatos que foram identificados como nódulos. Um aspecto relevante nesse arquivo é a informação sobre o diâmetro do nódulo *diameter\_mm*. Essa informação é importante para garantir que os dados de treinamento e validação abranjam uma variedade representativa de tamanhos de nódulos e para estabelecer igualdade entre os dados de ambos os arquivos de anotações.

Em seguida, foi criada uma função para unificar todas as informações usando uma tupla nomeada para representar as informações de cada nódulo. Essas tuplas não são as amostras de treinamento, mas servem como uma interface unificada para os dados humanamente anotados em ambos os arquivos auxiliares, isolando a complexidade do tratamento dos dados da fase de treinamento do modelo.

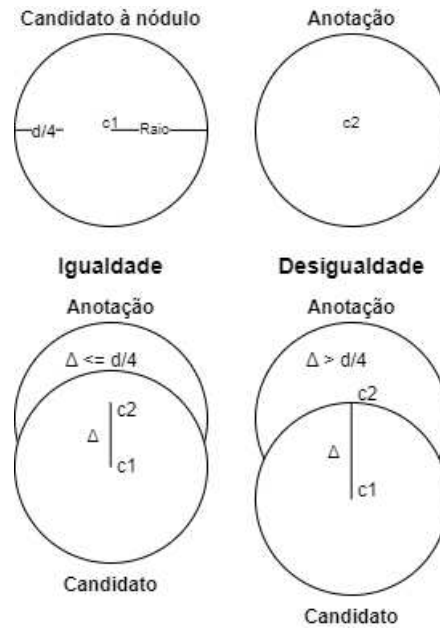
A lista de informações sobre candidatos inclui o status do nódulo, o diâmetro (para obter uma variedade de tamanhos no treinamento), a série (para localizar a tomografia correta) e o centro do candidato (para identificar o candidato na tomografia maior). A função que cria essa lista de tuplas começa com o uso de um decorador de cache em memória e, em seguida, obtém a lista de arquivos presentes no disco. O cache em memória é usado para acelerar futuras chamadas à função, uma vez que ela será utilizada com frequência em capítulos posteriores.

Para garantir que a quantidade de dados seja gerenciável, a função permite limitar os dados de treinamento por meio de um parâmetro booleano específico, identificando quais séries TC estão realmente disponíveis no disco. Isso é útil para verificar se o código funciona como o esperado com uma parcela menor dos dados. Mesmo que os resultados do treinamento com um subconjunto dos dados não sejam ideais, essa funcionalidade auxiliou o teste de funcionalidades como registros e métricas dos modelos ao longo do desenvolvimento.

Uma vez obtidas as informações sobre os candidatos, é importante observar que as informações de localização fornecidas no CSV *candidates* nem sempre coincidem precisamente com as coordenadas do CSV *annotations*. Mesmo que a imprecisão existente seja pequena, foram tomados cuidados especiais ao estabelecer a igualdade entre as estruturas, visto que, se tal atividade não fosse feita, poderiam existir problemas de união entre estruturas anatômicas distintas. Essa tarefa de estabelecimento de igualdade foi executada agrupando as anotações por *series.uid* e comparando as coordenadas dos candidatos com as coordenadas das anotações para determinar se correspondem ao mesmo nódulo. Quando há correspondência, o diâmetro é atribuído; caso contrário, é definido como 0.

Para comparar as estruturas nos dois arquivos de anotação, é utilizado um processo específico. Primeiramente, o diâmetro é dividido por 2 para encontrar o raio correspondente. Em seguida, esse raio é dividido por 2 novamente, o que é equivalente a dividir o diâmetro original por 4. Essa etapa estabelece um limite máximo para a distância entre os centros dos candidatos que estão sendo comparados. A distância entre os dois centros é chamada de delta e, caso seu valor for menor ou igual a um quarto do diâmetro da anotação, os candidatos são considerados equivalentes, indicando que se referem ao mesmo nódulo. Essa operação de igualdade pode ser melhor visualizada na Figura 14.

**Figura 14 – Operação de verificação de igualdade entre nóculo e anotação**



**Fonte: Autoria própria (2024).**

Essa verificação não é uma medida de distância euclidiana real entre os centros, mas uma verificação de limites, conhecida como *bounding-box check*. Ela funciona estabelecendo um limite de proximidade com base no tamanho do nóculo, garantindo que os centros dos candidatos estejam dentro de uma caixa delimitadora virtual. Se ambos os centros estiverem dentro dessa caixa delimitadora, é aferido que eles se referem ao mesmo nóculo. Esse algoritmo de teste da igualdade é ilustrado no fluxograma da Figura 15.

O fluxograma da Figura 16 ilustra como a estruturação das tuplas correspondentes aos candidatos à nóculos pulmonares como um todo é feita.

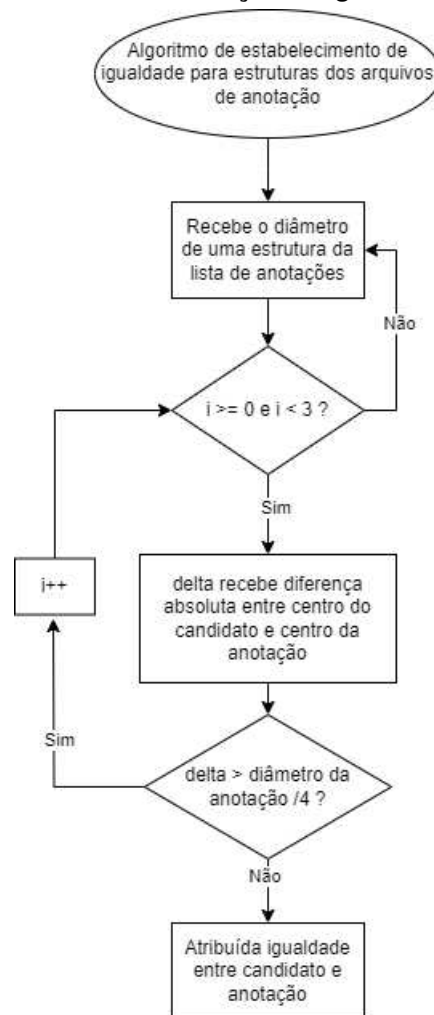
#### 4.1.2 Carregamento dos dados brutos

O processo de carregamento dos dados brutos da base se resume em transformar estes dados em objetos *Python*, para que se possa manipular e utilizar estes dados. Foi escolhido o pacote *Python SimpleITK* para receber e manipular estes dados brutos, de forma a facilitar a manipulação e transformação dos valores (LOWEKAMP *et al.*, 2013).

Como os dados de TC eram representados na base no formato *MetaIO*, o processo pôde ser simplificado, sem a necessidade de mais etapas intermediárias de conversão entre formatos. Com o *SimpleITK*, esses dados foram transformados em matrizes *NumPy*, de forma a serem tratados nos passos seguintes. O *SimpleITK* também permitiu a integração entre os dados, que são separados em dois arquivos, um *.MHD* e outro *.RAW*.

Os dados de TC, após o carregamento, são representados em uma matriz tridimensional, onde as três dimensões estão relacionadas ao espaço, e o canal de intensidade é implícito.

**Figura 15 – Algoritmo de determinação de igualdade entre anotações**



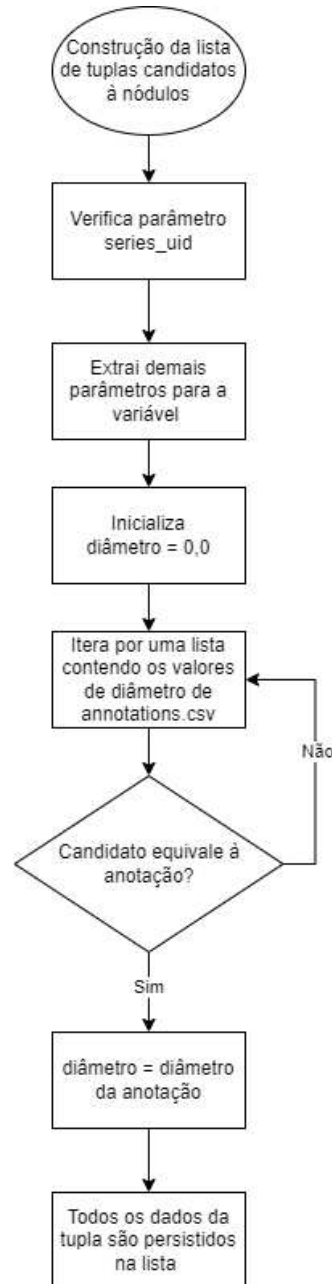
**Fonte: Autoria própria (2024).**

Esse conceito de estrutura é conhecido como *voxel*. Em *PyTorch*, a informação do canal de intensidade é tratada como uma quarta dimensão com tamanho 1.

Os dados sobre as localizações dos candidatos a nódulos que carregou-se inicialmente estavam expressos em milímetros, o que não é adequado para ser usado diretamente como índice de um vetor, do inglês *array*. Fez-se necessário elaborar um algoritmo que convertesse as coordenadas baseadas em milímetros (X, Y, Z) para o sistema de coordenadas baseado em índices, linhas e colunas (I, R, C) usado para extrair fatias de dados de TC.

Para realizar a conversão entre as coordenadas do paciente (XYZ) e as coordenadas de índice, linha e coluna (IRC) do *array*, seguiram-se quatro etapas: inverter as coordenadas de IRC para CRI (para alinhá-las com XYZ), dimensionar os índices pelos tamanhos de *voxel*, multiplicar a matriz de direções e adicionar o deslocamento de origem. O uso de funções da biblioteca *SimpleITK* poderia facilitar essa conversão, mas optou-se por realizar os cálculos manualmente para maior flexibilidade. Foram desenvolvidas duas funções responsáveis por essa conversão, uma para a conversão de coordenadas (X, Y, Z) para (I, R, C) e outra para realizar o processo inverso. Essas funções foram alocadas em um arquivo de funções utilitárias, para que

**Figura 16 – Algoritmo de montagem das tuplas de candidatos**



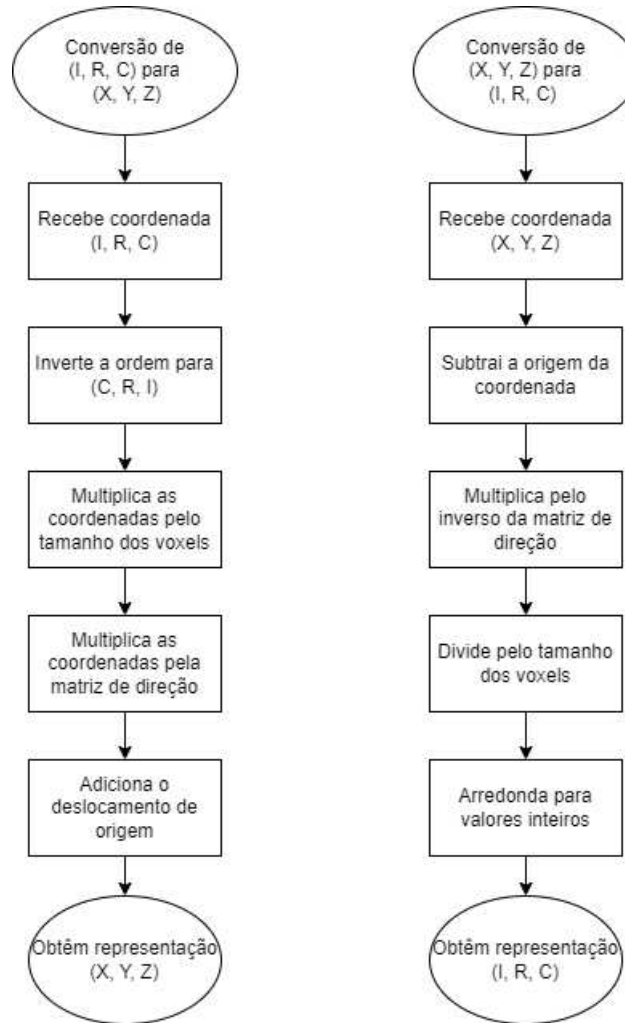
**Fonte: Autoria própria (2024).**

pudessem ser reaproveitadas, caso fosse necessário, ao longo do projeto. O esquema básico de funcionamento dessas funções pode ser visto com mais detalhes em Figura 17.

## 4.2 Segmentação

O principal objetivo da segmentação é identificar e delimitar as regiões de interesse nas imagens de TC, ou seja, áreas que podem conter nódulos. Isso foi feito com o uso de um modelo U-Net, uma arquitetura de rede neural projetada especificamente para tarefas de segmentação de imagem. O modelo de segmentação, após ter sido desenvolvido, foi adaptado

**Figura 17 – Fluxograma das funções de dimensionamento**



**Fonte: Autoria própria (2024).**

ao *loop* de treinamento introduzindo-se como um passo anterior à tarefa de classificação dos nódulos, servindo como um instrutor de regiões de interesse ao modelo de classificação.

#### 4.2.1 Arquitetura de rede

Para a implementação da arquitetura do modelo de DL, recorreu-se à implementação da U-Net de código aberto<sup>1</sup> baseada no trabalho de Ronneberger, Fischer e Brox (2015), feita especialmente para segmentação de imagens médicas em *PyTorch*. A escolha dessa implementação foi feita baseada em sua licença MIT, que permite seu uso livre de quaisquer encargos de *copyright*, sua simplicidade, visto que a implementação é feita em um único arquivo, e a flexibilidade oferecida por diversos parâmetros facilmente ajustáveis.

Foram feitas adaptações ao modelo de rede utilizado. Foi incluída uma camada de normalização em lotes antes da camada U-Net, a fim de aprimorar o treinamento da rede e fornecer

<sup>1</sup> Disponível em <https://github.com/jvanvugt/pytorch-unet>

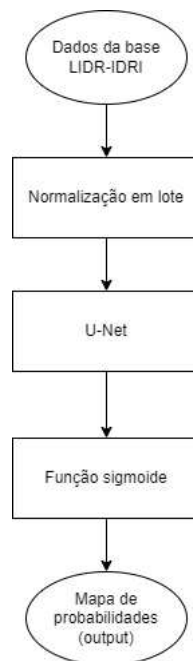
estatísticas de normalização específicas para cada lote. Dessa forma, quando uma amostra da base de dados não possui regiões de interesse (ou possui poucas), essa falta de dados será registrada nas estatísticas do lote e evitará que um lote de treinamento inteiro fique com muitas amostras irrelevantes.

Em seguida, há a U-Net propriamente dita, que pode ser enxergada como o núcleo do modelo de segmentação. A saída da U-Net foi projetada para ser um mapa de probabilidade que reflete a probabilidade de cada pixel pertencer a um nódulo. Cada *pixel* do mapa de saída representa a estimativa da probabilidade de ser uma parte de um nódulo. Para garantir que as saídas estejam no intervalo apropriado para probabilidades, a camada sigmoide é aplicada à saída da U-Net. Essa camada extra restringe os valores de saída no intervalo [0, 1], tornando-os adequados para representar probabilidades.

A saída do modelo como um todo foi, então, projetada para ser um mapa de probabilidades, onde cada pixel possui uma probabilidade, representada no intervalo [0,1], de pertencer à um nódulo pulmonar.

O modelo de segmentação desenvolvido pode ser visto mais claramente na Figura 18.

**Figura 18 – Diagrama do modelo de segmentação**



**Fonte: Autoria própria (2023).**

### 4.3 Classificação

O modelo de classificação foi desenvolvido com o objetivo de receber a entrada das regiões de interesse dos dados brutos da base, iterar por estes e classificá-los em nódulo ou não-nódulo. Para se chegar em um resultado satisfatório, foi construído um projeto de rede

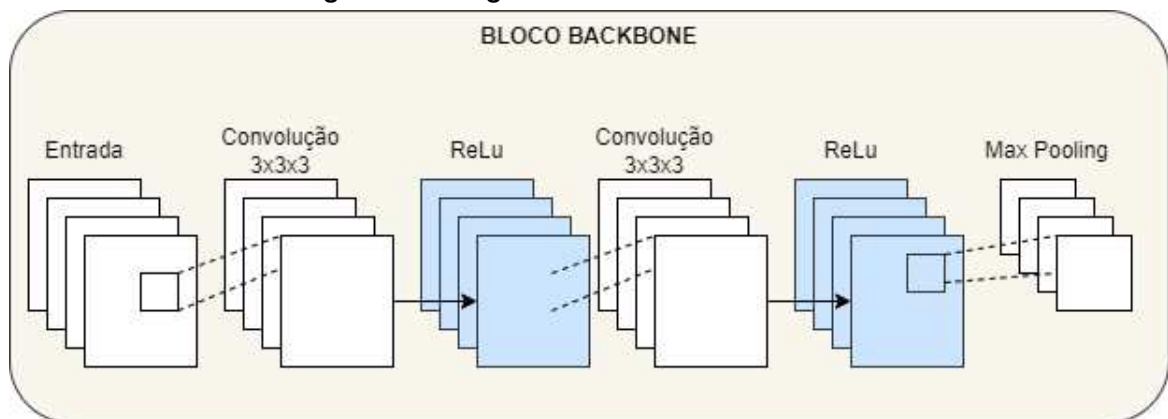
neural com várias partes integradas, a fim de extrair características dos dados brutos e, a partir dessa extração, classificar cada estrutura pré-processada.

#### 4.3.1 Arquitetura de rede

O modelo de classificação foi desenvolvido utilizando uma estrutura composta por cauda, *backbone* (do inglês, espinha dorsal) e cabeça. A cauda é constituída por uma camada de normalização em lote, e é responsável por lidar com o *output* direto da camada de segmentação e passá-lo a diante. Em seguida, o *backbone* da rede é constituído por quatro blocos, que executam determinadas tarefas de convolução. Cada bloco é composto por duas convoluções  $3 \times 3$ , seguidas cada uma, respectivamente, por uma ativação ReLu, com uma operação de *max-pooling* no final do bloco.

Os blocos começam com uma camada de convolução  $3 \times 3$ , que é responsável por extrair características iniciais da imagem de entrada. Essa camada aplica filtros que detectam padrões simples, como bordas e texturas. Após a primeira convolução, uma ativação ReLu é aplicada. A função ReLu introduz não-linearidade na rede, permitindo que a rede aprenda relações complexas nos dados. Em seguida, ocorre uma operação de *max-pooling*, que reduz a dimensionalidade dos recursos, mantendo as características mais importantes mas, ao mesmo tempo, "achatando"o formato de saída. Isso também ajuda a tornar a representação mais eficiente em termos computacionais. A arquitetura usada por cada bloco do *backbone* pode ser vista com mais detalhes na Figura 19.

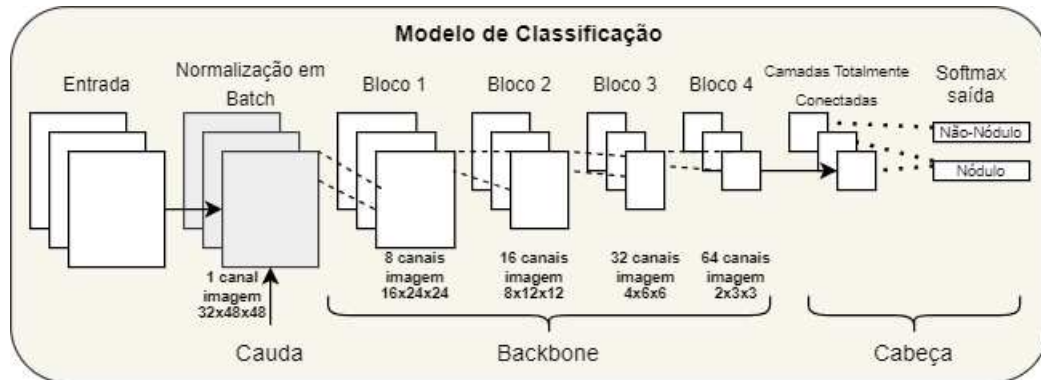
**Figura 19 – Diagrama de cada bloco *Backbone***



**Fonte: Autoria própria (2023).**

Finalmente, a cabeça da rede recebe a saída do *backbone*, aplica uma camada linear de camadas totalmente conectadas e, finalmente, aplica uma camada de função *softmax*. As camadas totalmente conectadas servem para conectar diretamente todos os neurônios da saída do *backbone* à função *softmax* que, por sua vez, transforma a saída em probabilidades e, de acordo com estas, afere uma classificação final. A arquitetura completa de rede para classificação desenvolvida pode ser vista na Figura 20.

**Figura 20 – Diagrama do modelo de classificação**



Fonte: Autoria própria (2023).

#### 4.4 Treinamento e validação

Para a parte de treinamento e validação, fez-se necessário, antes de tudo, estabelecer uma base sólida, para que esta, quando pronta, pudesse ser experimentada exaustivamente e ajustada sob demanda. Foi, então, arquitetado um algoritmo que seria utilizado para a validação e treinamento de ambos os modelos (segmentação e classificação) e tomou-se este como base para aprimoramentos, caso houvesse necessidade.

Os exames da base de dados foram divididos de forma a estabelecer uma repartição de 80% dos dados para a base de treinamento e 20% para a base de validação. Ressalta-se que a repartição foi feita em exames, ou seja, para um exame destinado a integrar a base de, por exemplo, treinamento, todas as imagens deste exame integram a mesma base.

O algoritmo de treinamento pode ser visto com mais detalhes na Figura 21, onde pode-se observar uma representação generalizada para os esquemas de treinamento e validação dos dois modelos desenvolvidos.

Para a elaboração deste esquema, visou-se a simplicidade ao oposto do desenvolvimento de infraestrutura de código exagerada, para que houvesse mais agilidade ao longo da execução do projeto e uma necessidade menor de adequação das demais partes do código à estrutura de testes. Desta forma, para este ciclo de treinamento e validação, foram desenvolvidas funções utilitárias simplificadas de registro de resultados que fossem equivalentes à funções mais complexas.

Um exemplo deste fato é a função de métricas de treinamento que, no lugar de gerar gráficos de desempenho (algo que aumentaria a complexidade de código e também, provavelmente, exigiria um módulo *Python* especializado na plotagem de gráficos a mais), foi desenvolvida uma versão mais simples, onde são somente fornecidos os indicadores numéricos. Tal simplificação não afeta a análise dos resultados e deixou o desenvolvimento do projeto razoavelmente menos complexo, uma troca que sempre é bem-vinda no mundo do desenvolvimento de *software*.

Figura 21 – Esquema elaborado para treinamento e validação



Fonte: Autoria própria (2023).

Primeiramente, a inicialização dos parâmetros de uma rede neural é um passo fundamental para garantir um desempenho adequado do modelo. Os pesos, vieses e outros parâmetros da rede devem possuir propriedades específicas para que a rede opere de maneira eficaz. Se todos os pesos da rede são maiores que 1 e não existem conexões residuais, a multiplicação repetida por esses pesos durante a passagem dos dados pelas camadas da rede resultaria em saídas de camadas excessivamente grandes, o que poderia levar a problemas numéricos como o *overflow*. Por outro lado, pesos menores que 1 fariam com que as saídas das camadas diminuíssem progressivamente até se tornarem insignificantes, um problema conhecido como *"vanishing gradients"* durante a etapa de retropropagação (STEVENS; ANTIGA; VIEHMANN, 2020).

Para evitar esses extremos, diversas técnicas de normalização podem ser aplicadas para manter as saídas das camadas e os gradientes em escalas razoáveis. Uma das abor-

dagens mais simples e eficazes é a inicialização adequada dos pesos da rede. Isso ajuda a garantir que os valores intermediários e os gradientes não se tornem nem muito pequenos nem muito grandes durante o treinamento da rede.

O *PyTorch* não fornece toda a assistência necessária para essa inicialização, portanto foi desenvolvida manualmente neste trabalho. Uma função de inicialização pode ser vista como um modelo padrão a ser utilizado, apesar de os detalhes exatos dessa função não serem cruciais. O importante é compreender o conceito de que uma inicialização adequada é essencial para o bom comportamento dos valores e dos gradientes ao longo do treinamento do modelo.

O treinamento conta com um *loop* que itera sobre as épocas de treinamento. Para cada época, uma função específica é chamada para executar o treinamento no conjunto de dados. Esta função é responsável por treinar o modelo em uma época específica. Ela configura o modelo para o modo de treinamento e inicializa um tensor para armazenar as métricas. Um *loop* interno itera sobre os lotes de dados, onde a otimização e a atualização dos pesos do modelo ocorrem. Dentro do *loop* de lotes, uma outra função é chamada para calcular a perda do lote atual. Esta função não apenas calcula a perda, mas também coleta informações detalhadas sobre cada amostra no lote, como a previsão do modelo, o rótulo verdadeiro e o valor da perda. Após o cálculo da perda, o gradiente é propagado para trás e os pesos do modelo são atualizados pelo otimizador. As métricas detalhadas são coletadas pelo tensor mencionado anteriormente e, em seguida, transferidas para a CPU para análise posterior.

Destaca-se a importância de coletar métricas detalhadas durante o treinamento, pois isso pode fornecer informações valiosas sobre o desempenho do modelo em diferentes classes e ajudar a identificar áreas onde o modelo pode estar tendo dificuldades. Além disso, foi-se implementada uma função responsável por estimar o tempo restante de treinamento, para que fosse possível saber de antemão quanto tempo o treinamento tomaria e, enquanto isso, pudesse ser feito algo a mais no desenvolvimento do projeto.

O processo de validação foi desenvolvido de uma forma muito similar ao treinamento, mas com algumas simplificações importantes. A validação é uma operação de apenas leitura, o que significa que os pesos do modelo não são atualizados e o valor da perda calculado não é utilizado para retropropagação do modelo. Como a validação não envolve a atualização dos pesos da rede, não há necessidade de usar o valor da perda retornado pela função nem de referenciar o otimizador. Há uma função chamada *torch.no\_grad()*, dentro do módulo *PyTorch*, que foi utilizada especificamente na validação, pelo fato da validação não requerer o cálculo de gradientes. Essa função foi utilizada em vista de simplificar a operação, reduzindo o tempo de computação e o uso de memória pela operação. Ainda, para a parte de coleta de métricas, o *loop* de validação, apesar de não utilizar o valor da perda por lote, coleta métricas detalhadas por meio de uma outra função, específica para o procedimento de validação.

A partir do momento em que o código estrutural estava pronto, começou-se a executar os *scripts* responsáveis pelo treinamento e validação. Ao executar tais *scripts*, tomaram-se alguns cuidados adicionais ao longo da execução como, por exemplo, a verificação de que o

poder de processamento disponível no sistema estava sendo quase todo utilizada e de que o treinamento não estava, conseqüentemente, demorando demasiadamente para terminar. Tal eficiência era imprescindível para o trabalho, visto que se os processos fossem muito ineficientes, haveria um atraso para realizar todas as múltiplas etapas que precisavam ser executadas. Ao longo da execução, foram registradas as métricas relevantes para este trabalho e, ao final do processamento, estas métricas foram salvas e listadas na seção de resultados.

#### 4.5 Métricas de desempenho

Para a avaliação dos modelos, foi-se empregado um conjunto abrangente de métricas para avaliar os modelos de forma eficiente. Essas métricas fornecem uma compreensão multifacetada dos pontos fortes e fracos do modelo, o que permite que melhorias adicionais possam ser pensadas exatamente nestes pontos fracos.

As métricas fundamentais utilizadas foram precisão e revocação. A precisão mede a proporção de previsões positivas verdadeiras entre todas as previsões positivas feitas pelo modelo, essencialmente quantificando a precisão do modelo na identificação de instâncias relevantes. A revocação, por outro lado, avalia a capacidade do modelo de capturar todas as instâncias relevantes, calculando a proporção de previsões positivas verdadeiras entre todas as instâncias positivas reais. Essas métricas são particularmente úteis em cenários onde o custo de falsos positivos e falsos negativos varia, permitindo-nos ajustar o desempenho do modelo de acordo com requisitos específicos.

Enquanto a precisão e a revocação fornecem informações valiosas individualmente, elas podem não oferecer uma imagem completa quando consideradas separadamente. A pontuação F1, outra métrica utilizada, harmoniza essas duas métricas calculando sua média harmônica. Essa métrica é especialmente útil ao buscar um equilíbrio entre precisão e revocação, pois penaliza valores extremos e recompensa modelos que mantêm um bom equilíbrio entre essas métricas. A pontuação F1 é uma métrica crucial para avaliar modelos em conjuntos de dados desequilibrados ou quando tanto falsos positivos quanto falsos negativos têm conseqüências significativas.

Os componentes base para a obtenção das métricas anteriormente citadas foram os Positivos Verdadeiros, Falsos Positivos, Negativos Verdadeiros e Falsos Negativos. Positivos verdadeiros (TP) e negativos verdadeiros (TN) representam previsões corretas para instâncias positivas e negativas, respectivamente. Falsos positivos (FP) ocorrem quando o modelo prevê incorretamente uma instância negativa como positiva, enquanto falsos negativos (FN) representam instâncias positivas que o modelo não consegue identificar. Compreender esses componentes é essencial para interpretar a precisão, a revocação e a pontuação F1, pois eles influenciam diretamente essas métricas.

Todas estas métricas, coletivamente, fornecem uma avaliação abrangente do desempenho de modelos de DL. Precisão e revocação oferecem uma vista detalhada sobre a eficiência

da precisão e sensibilidade do modelo, respectivamente, enquanto a pontuação F1 fornece uma medida equilibrada que considera tanto a precisão quanto a revocação. Os componentes subjacentes de positivos verdadeiros, falsos positivos, negativos verdadeiros e falsos negativos são fundamentais no cálculo dessas métricas e na compreensão das capacidades preditivas do modelo. Ao empregar essas métricas, podemos avaliar efetivamente o desempenho do modelo e identificar áreas para melhoria.

#### 4.5.1 Registro de Métricas

Além das métricas mencionadas anteriormente, o registro de métricas desempenha um papel crucial no monitoramento e na análise do desempenho dos modelos ao longo das versões. O registro de métricas permite que rastreie-se a evolução da precisão, da revocação e da pontuação F1 ao longo das épocas de treinamento. Isso oferece uma visão detalhada de como o modelo está aprendendo e se ajustando aos dados, facilitando a identificação de tendências, padrões e possíveis áreas de melhoria.

Implementou-se uma função de registros (*logging*) simples que registra essas métricas em intervalos regulares de épocas durante o treinamento, mostrando-as ao longo do treinamento. Isso nos permite visualizar o desempenho do modelo em tempo real e fazer modificações na arquitetura do modelo conforme necessário para otimizar os resultados. Além disso, o registro de métricas também é essencial para a comparação entre diferentes modelos ou configurações, fornecendo uma base objetiva para a seleção do modelo mais eficaz.

O *logger* criado utiliza o módulo *logging* da linguagem *Python*, que é uma ferramenta padrão para registro de mensagens de registro em aplicações *Python*. Esse módulo oferece uma maneira flexível de emitir registros de diferentes severidades, que vão desde mensagens de depuração (DEBUG) até erros críticos (CRITICAL), passando por informações (INFO) e alertas (WARNING).

No código fornecido, inicialmente configura-se o *logger* para o nível de severidade INFO, o que significa que mensagens de nível INFO e superiores serão capturadas. Essa configuração pode ser sobrescrita em qualquer parte do código, configurando-se o nível desejado de severidade.

O formato das mensagens de *log* é definido pela variável *logfmt\_str*, que inclui informações como a hora do registro, o nível de severidade, o ID do processo, o número da linha de código, o nome da função e a mensagem de *log* propriamente dita. Esse formato ajuda a tornar os registros mais informativos e fáceis de entender, especialmente em ambientes de produção ou ao depurar problemas complexos.

Um *StreamHandler*, que é uma classe do módulo *logging* usada para enviar mensagens de registro para uma saída de fluxo é adicionado ao *logger*, é configurado para emitir mensagens de *log* para a saída padrão (geralmente a tela ou o console). Em outras palavras, ele é um tipo de manipulador que direciona os registros para um fluxo específico, permitindo que os

registros sejam exibidas em um console ou terminal, por exemplo. Neste caso, ele é utilizado para redirecionar todos os registros para o console de visualização.

#### 4.5.2 Gráfico ROC e AUC

O conceito de Característica de Operação do Receptor, do inglês Receiver Operating Characteristic (ROC) e a Área Sob a Curva, do inglês Area Under the Curve (AUC) são fundamentais para a compreensão e avaliação de modelos em DL.

A curva ROC é um gráfico que apresenta o desempenho de um modelo de classificação para todos os limiares de decisão possíveis. Esta curva é plotada com a Taxa de Verdadeiros Positivos, do inglês True Positive Rate (TPR) no eixo Y contra a Taxa de Falsos Positivos, do inglês False Positive Rate (FPR) no eixo X. A TPR é também conhecida como revocação e é calculada como  $TP / (TP + FN)$ , onde TP são os verdadeiros positivos e FN são os falsos negativos. A FPR é calculada como  $FP / (FP + TN)$ , onde FP são os falsos positivos e TN são os verdadeiros negativos.

No contexto da detecção de nódulos, o diâmetro pode ser utilizado como um preditor simplificado de malignidade. Um classificador hipotético poderia ser construído para prever a malignidade com base em um limiar de diâmetro: nódulos maiores que esse limiar seriam classificados como malignos, enquanto os menores seriam classificados como benignos. A escolha desse limiar é crítica, pois influencia diretamente as taxas de verdadeiros e falsos positivos do modelo.

Ao variar o limiar do menor diâmetro possível, onde todos os nódulos seriam classificados como benignos (resultando em uma TPR e FPR de zero), até o maior diâmetro, onde todos seriam classificados como malignos (resultando em uma TPR e FPR de um), obtemos diferentes pares de TPR e FPR que, quando plotados, formam a curva ROC.

A AUC é um único valor escalar que resume a curva ROC, representando o grau de separabilidade que o modelo consegue atingir. Um valor de AUC de 1 indica um modelo perfeito que consegue distinguir completamente entre as classes benigna e maligna, enquanto um valor de 0,5 sugere um desempenho não melhor do que uma classificação aleatória. Portanto, quanto maior a AUC, melhor é o modelo em prever as classes corretamente.

Para calcular a AUC, utiliza-se a integração numérica pelo método da regra trapezoidal. Este método estima a área sob a curva ROC somando a área de trapézios formados entre cada par de pontos consecutivos na curva. Assim, a AUC fornece uma medida agregada da performance do modelo em todos os limiares de classificação, sendo uma métrica robusta para a comparação entre diferentes modelos.

Dois limiares específicos de diâmetro são escolhidos, 5,42 mm e 10,55 mm, como exemplos de pontos que fornecem extremos razoáveis para o intervalo de limiares a serem considerados. Esses valores ajudam a ilustrar como a escolha de um limiar pode influenciar as taxas de verdadeiros e falsos positivos, e consequentemente, o formato da curva ROC e o valor da AUC.

## 4.6 Utilização do Cluster

Para o treinamento dos modelos desenvolvidos, foi utilizado o *cluster* da UTFPR, que dispõe de várias máquinas virtuais com placas de vídeo potentes, que permitiram acelerar o processo de treinamento. O *cluster* foi acessado remotamente por meio de Secure Socket Shell (SSH), que é um protocolo de acesso seguro à máquinas remotas. A máquina remota do *cluster* serviu somente para rodar os *scripts* de treinamento, validação e demais testes de *performance* do modelo, sendo o desenvolvimento feito inteiramente a parte em uma máquina local.

Para que os *scripts* pudessem ser executados remotamente dentro do *cluster* foi hospedado o código fonte do projeto em um repositório *GitHub* e clonado no disco rígido do recurso. O ambiente remoto de acesso do *cluster* utiliza o sistema operacional Ubuntu, o que facilitou a utilização do *Python* e outros recursos necessários do projeto, visto que estes já são embutidos neste sistema. Dentro da pasta contendo o código fonte do projeto, foi criado um ambiente virtual *Python* e foram instaladas todas as dependências do projeto, de acordo com o arquivo de texto *requirements.txt*. Então, foi usado este ambiente virtual para todas as execuções de código para este projeto, para que se garantisse o funcionamento correto do código com todas as suas dependências e se evitasse conflito entre os módulos, algo que poderia ser ocasionado por uma instalação destes pacotes de modo global no *cluster*.

Com o auxílio do recurso fornecido pela UTFPR, os modelos puderam ser treinados e validados de forma mais ágil, visto que parâmetros, como tamanho de *batch* e número de *workers*, puderam ser aumentados sem ocasionar problemas de memória RAM ou de memória da GPU na máquina. Para os *scripts* deste projeto, foi escolhida a combinação de parâmetros de tamanho de *batch* como 128 e de número de *workers* como 8. Essa combinação foi escolhida por tentativa e erro, após execução do código ao longo do projeto e monitoramento do consumo dos recursos computacionais da máquina remota, e foi a combinação que mais se destacou, visto que ofereceu bons resultados em um tempo relativamente curto e não apresentou nenhum problema de *hardware* nas execuções.

## 5 RESULTADOS E DISCUSSÃO

Neste capítulo, serão discutidos os resultados obtidos para os modelos de classificação e de segmentação, de acordo com as métricas de avaliação adotadas. Ainda, será avaliado o comportamento do *software* como um todo, ou seja, com ambos os modelos trabalhando de forma conjunta. Os modelos finais de cada rede foram armazenados no repositório virtual do código fonte do projeto.

Para a avaliação dos resultados dos modelos individualmente, foram usados vários indicadores de desempenho, os quais foram medidos ao longo de intervalos de épocas de treinamento. Os resultados para cada modelo são apresentados em tabelas contendo as médias dos valores obtidos em cinco treinamentos, junto com gráficos *boxplot* das medianas de todos os valores obtidos para cada época nas cinco iterações.

### 5.1 Segmentação

Os resultados do treinamento do modelo de segmentação podem ser vistos na Tabela 2. Pode-se observar que os verdadeiros positivos e a pontuação F1 estão em ascensão, enquanto os falsos positivos e negativos diminuem, o que reflete o comportamento desejado para estas métricas. É importante considerar que, como o treinamento foi realizado em recortes de  $64 \times 64$  *pixels* e a validação em imagens inteiras de  $512 \times 512$  *pixels*, previu-se que haveria taxas drasticamente diferentes para os campos de verdadeiros positivos (VP), falsos negativos (FN) e falsos positivos (FP) nos resultados da validação para este mesmo modelo. Os resultados da validação podem ser vistos na Tabela 3.

**Tabela 2 – Resultados de treinamento do modelo de segmentação**

| Época | Loss   | Precisão | Revocação | Escore F1 | VP    | FN   | FP    |
|-------|--------|----------|-----------|-----------|-------|------|-------|
| E1    | 0,5189 | 0,2287   | 0,9329    | 0,3584    | 93,3% | 6,2% | 97,7% |
| E5    | 0,2517 | 0,5633   | 0,9381    | 0,7057    | 93,8% | 5,8% | 96,3% |
| E10   | 0,2380 | 0,6128   | 0,9489    | 0,7273    | 94,9% | 5,5% | 59,1% |
| E15   | 0,2205 | 0,6168   | 0,9575    | 0,7624    | 95,8% | 4,2% | 55,0% |
| E20   | 0,2110 | 0,6341   | 0,9603    | 0,7731    | 96,0% | 4,0% | 51,6% |

Fonte: Autoria Própria (2024).

É importante ressaltar que a taxa de falsos positivos obtida para a validação foi desconhecida da representação da Tabela 4, pois os valores para esta coluna foram exorbitantes e não agregam muito valor à interpretação dos resultados, visto que a área da fatia de validação é de  $2^{18}$  *pixels*, enquanto o recorte de treinamento é de apenas  $2^{12}$  *pixels*. Essa validação não se tornaria coerente por se tratar de uma área 64 vezes maior.

Ressalta-se que a taxa de verdadeiros positivos não mudou significativamente, pois tudo isso teria sido incluído na amostra de  $64 \times 64$  *pixels* com a qual treinou-se inicialmente. Essa

**Tabela 3 – Resultados da validação do modelo de segmentação**

| Época      | Loss   | Precisão | Revocação | Escore F1 | VP    | FN    |
|------------|--------|----------|-----------|-----------|-------|-------|
| <b>E1</b>  | 0,9335 | 0,0212   | 0,8131    | 0,0415    | 81,3% | 17,9% |
| <b>E5</b>  | 0,9017 | 0,0306   | 0,8372    | 0,0646    | 83,7% | 14,9% |
| <b>E10</b> | 0,9520 | 0,0166   | 0,8428    | 0,0348    | 84,3% | 17,8% |
| <b>E15</b> | 0,8061 | 0,0521   | 0,7841    | 0,1142    | 78,4% | 25,6% |
| <b>E20</b> | 0,8600 | 0,0424   | 0,7676    | 0,0822    | 76,8% | 21,4% |

**Fonte: Aatoria Própria (2024).**

situação também resulta em uma precisão muito baixa e, portanto, em uma pontuação F1 baixa. Esse é um resultado natural da estrutura do treinamento e validação, portanto, não é motivo de alarme.

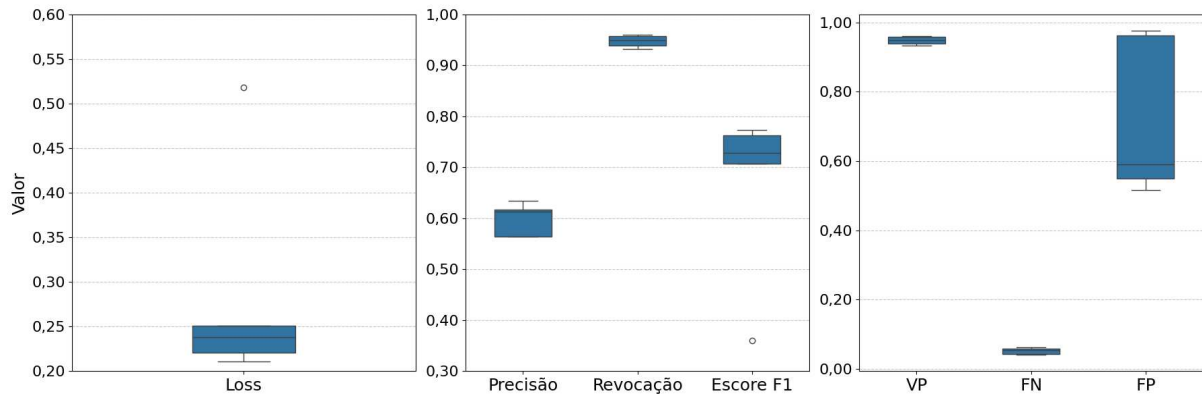
No entanto, o que se torna problemático é a revocação (e, conseqüentemente, a taxa de verdadeiros positivos). A revocação se estabilizou entre as épocas 5 e 10 e depois começou a diminuir. É evidente que começa a ter-se um ajuste excessivo muito rapidamente. A arquitetura U-Net tem uma capacidade significativa de memorizar rapidamente o conjunto de treinamento. Uma vantagem da utilização desta arquitetura de rede neural é que não torna-se necessário investir tempo em excesso para a realização do treinamento e conseqüente validação.

A revocação foi a principal prioridade para a segmentação, pois a precisão foi arquitetada para ser responsabilidade do modelo de classificação. O modelo de classificação também irá ajudar com os indicadores alarmantes de falsos positivos obtidos em ambos os roteiros de obtenção de resultados.

Sobre as medianas dos resultados de todas as cinco iterações de treinamento e validação, também podem ser observadas algumas informações valiosas para a aferição da qualidade do modelo. Analisando o *boxplot* presente na Figura 22, observa-se, primeiramente, uma mediana de *loss* relativamente baixa, indicando que o modelo está convergindo para uma zona na qual há a possibilidade de uma segmentação efetiva. A variabilidade também é baixa, o que sugere consistência durante o treinamento. A precisão tem uma mediana mais alta do que o *loss*, mas com uma variação maior, indicada pelos quartis mais espaçados e *outliers* presentes. Isso pode indicar que, enquanto o modelo geralmente prevê corretamente as classes positivas, existem casos em que ele falha. A revocação tem uma mediana menor e uma variabilidade maior em comparação com a precisão. Isso sugere que o modelo pode estar perdendo algumas instâncias positivas verdadeiras. O escore F1, que é a média harmônica entre precisão e revocação, apresenta uma mediana equilibrada, mas com uma variabilidade significativa. Isso reflete a variação observada nas métricas de precisão e revocação. Ainda, a contagem de verdadeiros positivos é consistente, como indicado pela caixa estreita do *boxplot*. Os Falsos Negativos e Falsos Positivos apresentam comportamentos distintos. Os Falsos Positivos têm uma mediana em torno de 60%, indicando uma taxa relativamente alta, enquanto os Falsos Negativos têm uma mediana em torno de 5%, o que é consideravelmente mais baixo. Embora ambos idealmente

devessem ter medianas baixas, observa-se que os Falsos Positivos apresentam uma variação significativa, o que pode ser um aspecto a ser melhorado em futuras iterações do modelo.

**Figura 22 – Distribuição dos resultados obtidos para o conjunto das 5 rodadas de treinamento da segmentação**



**Fonte: Autoria própria (2024).**

Um gráfico *boxplot* das medianas de validação está presente na Figura 23. A mediana de *loss* na validação é ligeiramente mais alta do que no treinamento, com uma variabilidade um pouco maior. Isso é esperado, pois o modelo pode não generalizar tão bem para dados não vistos durante o treinamento.

A precisão é consistentemente mais alta na validação do que no treinamento, com menos variação, o que é um bom sinal de que o modelo está bem ajustado. Similarmente à precisão, a revocação tem uma mediana mais alta na validação do que no treinamento. No entanto, a variabilidade é significativa, sugerindo que o desempenho do modelo pode variar mais nos dados de validação.

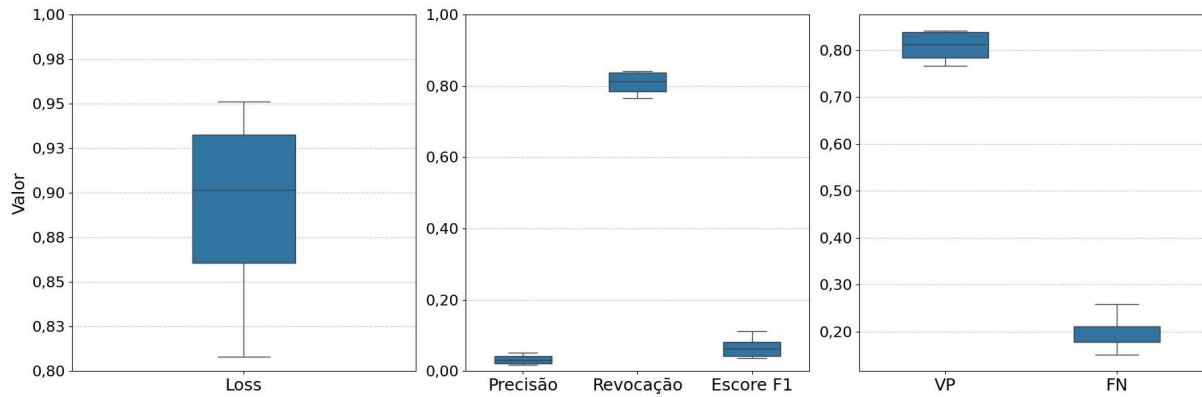
O escore F1 é comparável entre treinamento e validação, mas com uma variação mais ampla na validação, refletindo a variação nas métricas de precisão e revocação.

Os Falsos Positivos não estão presentes no gráfico de validação, pois o resultado distorceria o gráfico e dificultaria a visualização dos outros parâmetros. Destaca-se, mais uma vez, o fato de que os Falsos Positivos tiveram um valor exorbitante por causa da validação atuar em fatias 512 x 512, enquanto o treinamento atua em fatias muito menores, de 64 x 64. Por fim, os Falsos Negativos apresentam uma mediana de zero com uma variação maior do que no treinamento, sugerindo que o modelo tem mais dificuldade em generalizar a detecção de casos positivos nos dados de validação.

O indicador de *loss* e o escore F1 são comparáveis entre treinamento e validação, sugerindo que o modelo está generalizando razoavelmente bem. A precisão e a revocação são melhores na validação do que no treinamento, o que é um pouco atípico, pois geralmente espera-se ver uma queda no desempenho do modelo nos dados de validação. Isso pode indicar que os dados de validação podem não ser tão representativos como se esperava da complexidade ou da distribuição dos dados de treinamento. Ainda, a variabilidade nas métricas de validação é

geralmente maior do que no treinamento, o que é esperado devido à natureza dos dados de validação serem não vistos anteriormente pelo modelo.

**Figura 23 – Distribuição dos resultados obtidos para o conjunto das 5 rodadas de validação da segmentação**



**Fonte: Autoria própria (2024).**

## 5.2 Classificação

O modelo de classificação foi treinado por 75 épocas, quantidade que ofereceu o resultado máximo do modelo, oferecendo menos perdas para um resultado maior, e sem utilizar exaustivamente a máquina de treinamento (para evitar desconexões ou problemas de utilização de recursos). O tempo de treinamento foi de aproximadamente 15 minutos por época, o que resultou em aproximadamente 19 horas totais para que todas as épocas pudessem ser processadas.

Para o treinamento, foi considerada uma base de 499313 amostras, entre estas 498285 negativas e 1028 positivas (nódulos pulmonares). Já a base de validação contou com 51583 amostras, com o objetivo de aferir a reprodutibilidade do modelo em amostras fora da base de treinamento. Dentre as amostras de validação, 51429 eram estruturas anatômicas diversas (não-nódulos) e 154 eram, de fato, nódulos pulmonares.

É importante ressaltar que os resultados para a classificação apresentam os resultados de uma forma um pouco diferente, introduzindo os valores "Positivos" e "Negativos". Esta é uma forma mais focada em trazer objetivamente a dualidade da classificação de maneira simples, sendo que cada valor corresponde à quão bem o classificador conseguiu classificar corretamente cada categoria da base, seja de treinamento ou validação.

Analisando os resultados dos modelos de classificação apresentados na Tabela 4, pode-se observar um desempenho notável na fase de treinamento. Ao longo das 75 épocas, o modelo exibiu uma melhoria consistente em métricas como precisão, revocação e escore F1 para ambas as classes: negativos e positivos. O modelo treinado alcançou um ponto máximo de 99,7% de precisão em amostras positivas e 99,2% em negativas na época 75. Isso ressalta a ca-

**Tabela 4 – Resultados do treinamento do modelo de classificação**

| Época | Loss   | Precisão | Revocação | Escore F1 | Negativos (25000) | Positivos (25000) |
|-------|--------|----------|-----------|-----------|-------------------|-------------------|
| E5    | 0,1539 | 0,9597   | 0,9524    | 0,9557    | 95,9% (23975)     | 95,2% (23804)     |
| E10   | 0,1152 | 0,9713   | 0,9663    | 0,9688    | 96,4% (24102)     | 96,6% (24109)     |
| E15   | 0,0866 | 0,9744   | 0,9726    | 0,9735    | 97,3% (24325)     | 97,2% (24305)     |
| E20   | 0,0670 | 0,9778   | 0,9782    | 0,9780    | 97,6% (24413)     | 97,8% (24439)     |
| E25   | 0,0568 | 0,9804   | 0,9847    | 0,9825    | 98,0% (24521)     | 98,5% (24623)     |
| E30   | 0,0502 | 0,9823   | 0,9868    | 0,9845    | 98,3% (24579)     | 98,7% (24679)     |
| E35   | 0,0421 | 0,9846   | 0,9919    | 0,9882    | 98,6% (24684)     | 99,2% (24803)     |
| E40   | 0,0391 | 0,9870   | 0,9895    | 0,9883    | 98,9% (24743)     | 98,9% (24730)     |
| E45   | 0,0353 | 0,9874   | 0,9914    | 0,9894    | 98,9% (24745)     | 99,1% (24771)     |
| E50   | 0,0302 | 0,9889   | 0,9917    | 0,9903    | 99,0% (24763)     | 99,2% (24789)     |
| E55   | 0,0254 | 0,9917   | 0,9923    | 0,9920    | 99,1% (24782)     | 99,2% (24803)     |
| E60   | 0,0225 | 0,9909   | 0,9926    | 0,9917    | 99,2% (24810)     | 99,2% (24821)     |
| E65   | 0,0208 | 0,9906   | 0,9923    | 0,9914    | 99,2% (24812)     | 99,2% (24830)     |
| E70   | 0,0237 | 0,9915   | 0,9949    | 0,9932    | 99,2% (24813)     | 99,5% (24864)     |
| E75   | 0,0223 | 0,9914   | 0,9959    | 0,9935    | 99,2% (24817)     | 99,6% (24903)     |

Fonte: Autoria própria (2024).

pacidade superior do modelo em identificar corretamente os objetos que não são nódulos em comparação com a classificação precisa dos positivos. A perda do modelo também diminuiu consistentemente ao longo do treinamento, chegando a 0,0216 na época 75, indicando uma boa convergência do modelo.

**Tabela 5 – Resultados da validação do modelo de classificação**

| Época | Loss   | Precisão | Revocação | Escore F1 | Negativos (51429) | Positivos (154) |
|-------|--------|----------|-----------|-----------|-------------------|-----------------|
| E1    | 0,3938 | 0,0188   | 0,8747    | 0,0369    | 85,8% (44127)     | 87,5% (135)     |
| E5    | 0,1635 | 0,0571   | 0,9340    | 0,1077    | 96,1% (49417)     | 93,3% (144)     |
| E10   | 0,1337 | 0,0718   | 0,9664    | 0,1339    | 96,5% (49672)     | 96,6% (149)     |
| E15   | 0,0845 | 0,1087   | 0,9574    | 0,1957    | 98,0% (50413)     | 95,7% (147)     |
| E20   | 0,0521 | 0,1710   | 0,9425    | 0,2897    | 99,0% (50921)     | 94,2% (145)     |
| E25   | 0,0619 | 0,1475   | 0,9509    | 0,2538    | 98,7% (50769)     | 95,1% (146)     |
| E30   | 0,0530 | 0,1725   | 0,9556    | 0,2911    | 99,0% (50921)     | 95,6% (147)     |
| E35   | 0,0530 | 0,1673   | 0,9620    | 0,2837    | 98,9% (50882)     | 96,2% (148)     |
| E40   | 0,0491 | 0,1805   | 0,9689    | 0,3029    | 99,0% (50923)     | 96,9% (149)     |
| E45   | 0,0611 | 0,1450   | 0,9689    | 0,2507    | 98,6% (50731)     | 96,9% (149)     |
| E50   | 0,0390 | 0,2166   | 0,9620    | 0,3527    | 99,2% (51021)     | 96,2% (148)     |
| E55   | 0,0671 | 0,1436   | 0,9729    | 0,2487    | 98,6% (50739)     | 97,3% (150)     |
| E60   | 0,0525 | 0,1710   | 0,9556    | 0,2888    | 99,0% (50923)     | 95,6% (147)     |
| E65   | 0,0337 | 0,2399   | 0,9340    | 0,3835    | 99,4% (51104)     | 93,4% (144)     |
| E70   | 0,0409 | 0,2090   | 0,9556    | 0,3421    | 99,2% (51024)     | 95,6% (147)     |
| E75   | 0,0300 | 0,2678   | 0,9509    | 0,4195    | 99,5% (51185)     | 95,1% (146)     |

Fonte: Autoria própria (2024).

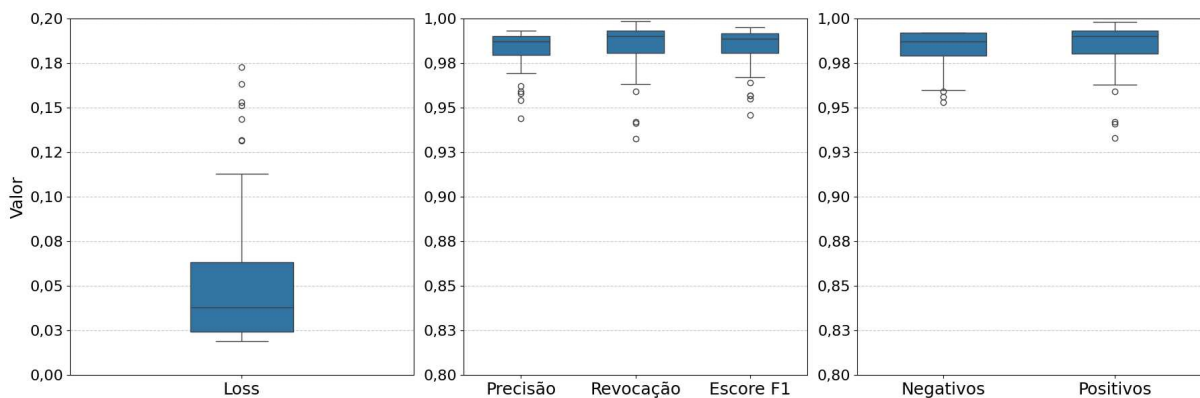
Por outro lado, ao avaliar os resultados de validação apresentados na Tabela 5, percebe-se que, apesar de uma tendência semelhante de melhoria nas métricas ao longo das épocas,

os resultados foram ligeiramente inferiores em comparação com os dados de treinamento. O modelo exibiu uma precisão de 99,2% para amostras positivas e 95,5% para negativas na época 75, o que destaca um desempenho bastante próximo ao observado no treinamento, indicando uma boa capacidade do modelo em generalizar para conjuntos de dados novos, que não foram utilizados no treinamento.

Para o modelo, o resultado obtido na época 75 foi bastante satisfatório, com acurácia alta e perda baixa. Ainda que o resultado tenha sido satisfatório, há espaço para melhorias, em vista de que o modelo apresentou uma capacidade superior de filtrar as amostras negativas em desvantagem de uma melhor classificação das amostras positivas.

Analisando os gráficos *boxplot* das medianas dos treinamentos e validações presentes, respectivamente, nas Figuras 24 e 25, é possível observar algumas tendências e diferenças importantes.

**Figura 24 – Distribuição dos resultados obtidos para o conjunto das 5 rodadas de treinamento da classificação**



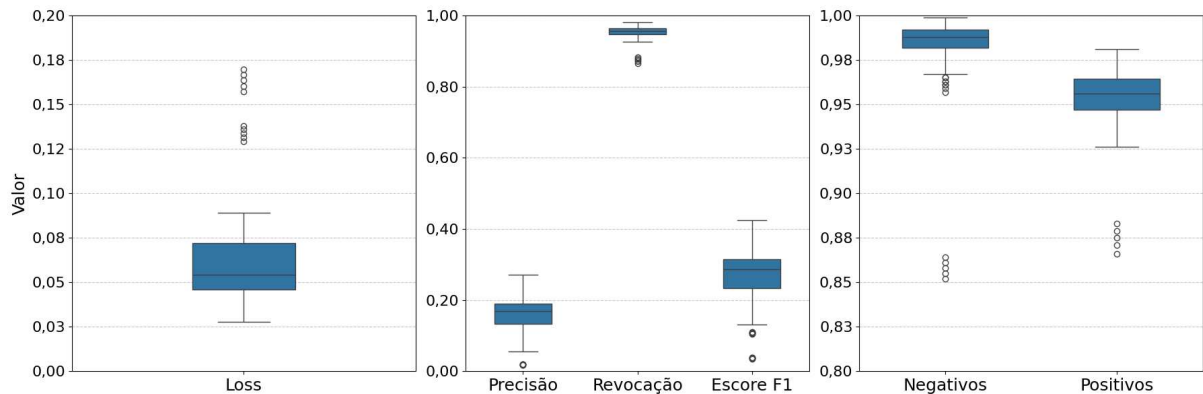
**Fonte: Autoria própria (2024).**

No gráfico de treinamento presente na Figura 24, as métricas de Precisão, Revocação, Score F1, Negativos e Positivos mostram valores altos e consistentes, com medianas muito próximas de 1,0 e uma variação muito pequena, indicada pelas caixas muito estreitas dos *boxplots*. Isso sugere um alto desempenho do modelo nos dados de treinamento. Por outro lado, o *boxplot* de *Loss* destaca-se dos demais com uma mediana próxima de 0,2 e uma variação relativamente maior, o que é esperado, pois durante o treinamento, o objetivo é minimizar essa perda.

No gráfico de validação da Figura 25, há uma situação um pouco diferente. Observa-se uma variação maior em todas as métricas. O *Loss* tem uma mediana superior em comparação com o treinamento, refletindo um desempenho geralmente pior em dados não vistos anteriormente.

A Precisão, Revocação e o Score F1 mostram uma distribuição mais ampla de valores, com medianas significativamente mais baixas do que as do treinamento. Isso pode indicar *overfitting* no modelo, onde o desempenho em dados novos é inferior ao desempenho em dados de

**Figura 25 – Distribuição dos resultados obtidos para o conjunto das 5 rodadas de validação da classificação**



**Fonte: Autoria própria (2024).**

treinamento. Os valores de Negativos e Positivos aparentam uma menor variação, mas ainda apresentam uma discrepância entre treinamento e validação.

Comparando os dois gráficos, a consistência nas métricas de treinamento contrasta com a variabilidade nas métricas de validação. Isso pode sugerir que o modelo está assimilando os dados de treinamento em vez de aprender características generalizáveis. Para melhorar a generalização do modelo, seria útil investigar técnicas de regularização, como, por exemplo, *dropout* ou regularização L1/L2.

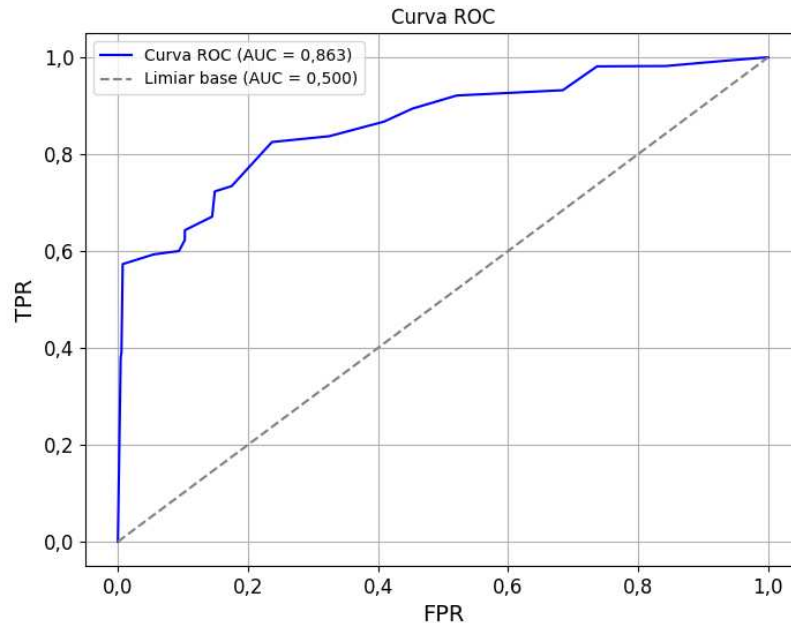
Além disso, o fato de haver *outliers* indicados pelos pontos fora dos bigodes dos *boxplots* em validação, e não em treinamento, reforça a ideia de que o modelo pode não estar tão robusto a novos dados, destacando a importância de considerar esses pontos ao buscar melhorias.

A alta precisão e revocação no treinamento indicam uma habilidade do modelo em classificar corretamente os casos positivos e negativos nesse conjunto. Contudo, a queda dessas métricas em validação sugere dificuldades do modelo em manter essa performance em dados não vistos. O ideal seria que os gráficos de treinamento e validação fossem mais semelhantes, indicando que o modelo é capaz de generalizar bem suas previsões para dados novos.

### 5.3 Curva ROC e AUC

Na Figura 26, está representada a curva ROC gerada a partir do modelo de classificação. A curva é uma representação gráfica que ilustra a capacidade de diagnóstico de um sistema classificador à medida que seu limiar de discriminação varia. O eixo horizontal FPR representa a proporção de falsos positivos, isto é, a fração de negativos verdadeiros que foram incorretamente identificados como positivos pelo modelo. O eixo vertical TPR, também conhecido como revocação ou sensibilidade, representa a proporção de verdadeiros positivos, ou seja, a fração de positivos verdadeiros que foram corretamente identificados como tal.

**Figura 26 – Curva ROC e AUC do classificador**



**Fonte: Autoria própria (2024).**

O eixo horizontal FPR representa a proporção de falsos positivos, isto é, a fração de negativos verdadeiros que foram incorretamente identificados como positivos pelo modelo. O eixo vertical TPR, também conhecido como revocação ou sensibilidade, representa a proporção de verdadeiros positivos, ou seja, a fração de positivos verdadeiros que foram corretamente identificados como tal.

Nesta curva ROC, observa-se que a TPR aumenta rapidamente com um pequeno aumento no FPR no início. Isso indica que o modelo tem uma boa capacidade de identificar corretamente as instâncias positivas sem incorrer em muitos falsos positivos. À medida que o FPR aumenta, o crescimento da TPR torna-se mais gradual. Esta tendência sugere que, após um certo ponto, para aumentar a sensibilidade do modelo TPR, deve-se aceitar um maior número de falsos positivos FPR.

A linha pontilhada representa um classificador aleatório, cuja AUC é 0,500. Este é o pior cenário possível, onde o modelo não tem capacidade de discriminação entre as classes positivas e negativas, equivalente a um lançamento aleatório de moedas. A curva ROC azul está bem acima da linha de base, com um valor de 0,863, o que indica que o modelo tem uma capacidade de discriminação significativamente melhor do que uma escolha aleatória.

#### **5.4 Validação quantitativa**

Realizou-se a validação quantitativa do modelo construído, observando seu desempenho no conjunto de validação completo. Esta parte consistiu na validação do esquema como um

todo, abrangendo o funcionamento do algoritmo de classificação diretamente ligado com a parte de segmentação. Neste procedimento foi utilizado o conjunto de validação inteiro. O conjunto de dados utilizado continha imagens de TC que somavam um total de 154 nódulos passíveis de detecção. Os resultados da varredura do sistema completo podem ser analisados na Tabela 6, onde tem-se a matriz de validação do sistema como um todo.

**Tabela 6 – Resultados de validação do sistema**

|                   | Não detectado | Filtrado (segmentação) | Detectado (classificação) |
|-------------------|---------------|------------------------|---------------------------|
| <b>Não nódulo</b> |               | 164893                 | 2156                      |
| <b>Benigno</b>    | 12            | 3                      | 87                        |
| <b>Maligno</b>    | 1             | 6                      | 45                        |

**Fonte: Autoria Própria (2024).**

Analisando a Tabela 6, vê-se que o total de nódulos identificados pelo *software* foi de 132 (a soma de nódulos benignos e malignos detectados). Essa quantia representa 85% do total de nódulos presentes no conjunto de validação, o qual é um bom resultado. Além disso, o sistema apontou como nódulo 2156 estruturas anatômicas que não eram nódulos, um número grande e indesejado.

Ressalta-se que o fato de haver falsos positivos como estes, apesar de ser indesejado, representa uma situação melhor e menos crítica do que haver vários falsos negativos, onde um nódulo não é apontado como tal mas é. Ainda, observa-se que a maior parte das estruturas anatômicas que não eram nódulos foi filtrada pelo algoritmo de segmentação antes de chegar à parte de classificação, o que reforça o bom funcionamento da respectiva parte.

## 5.5 Exemplos de nódulos classificados

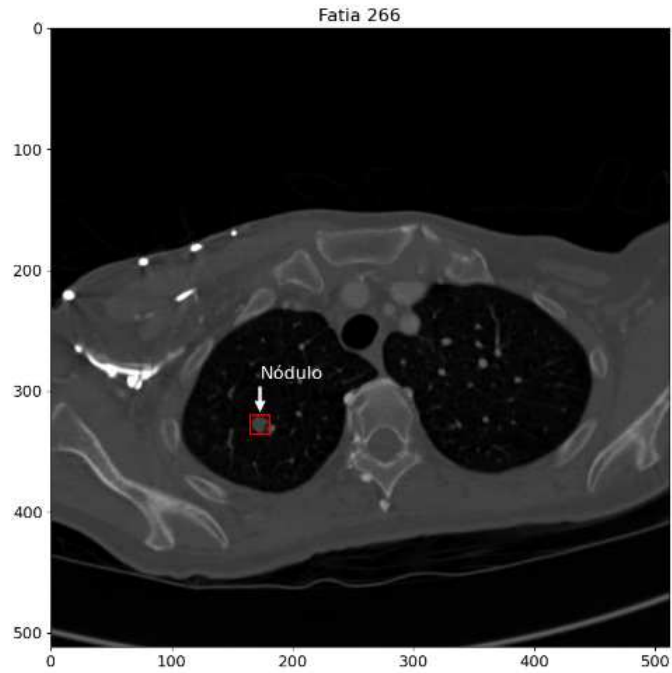
Para a melhor visualização dos resultados, optou-se por desenvolver um *script* que, a partir dos resultados obtidos pelo sistema, gerasse a representação visual do nódulo segmentado no contexto da imagem.

Na Figura 27, consta uma "fatia" de um dos exames de tomografia da base de dados. Neste exame, o sistema segmentou e classificou um nódulo maligno, que encontra-se destacado por um quadrado de borda vermelha na imagem do exame.

Como na base de dados haviam nódulos benignos e malignos, também houveram situações onde foram detectados nódulos benignos em exames, como presente na Figura 28. No caso, esta estrutura é um nódulo pulmonar, mas não apresenta riscos à saúde do paciente. Ainda, na Figura 29, tem-se um exame com uma estrutura que aparenta ser um tumor genérico, classificado como nódulo pelo *software*.

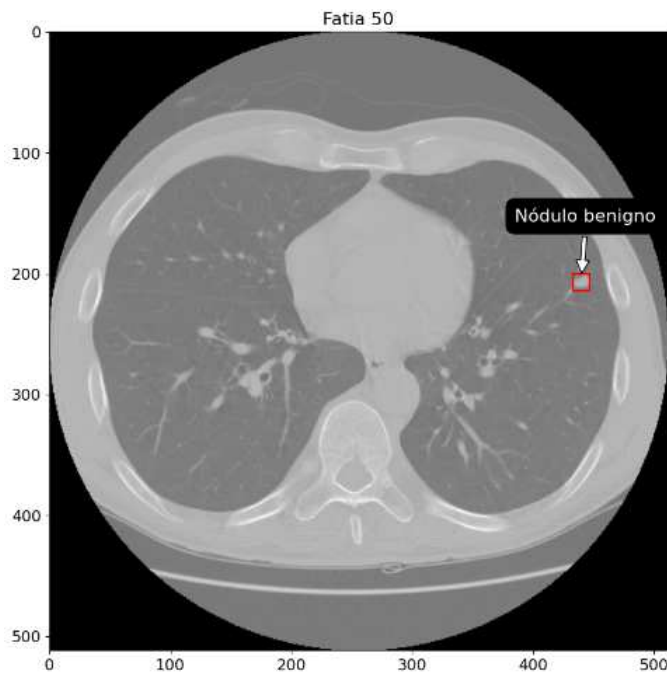
Destaca-se que alguns fatores de imagem estão diferentes entre as 3 figuras, tais como resolução e intensidade, pois os exames foram obtidos de diversas instituições, que contavam com aparelhos diferentes de TC.

**Figura 27 – Exame com nódulo maligno segmentado**



Fonte: Autoria própria (2024).

**Figura 28 – Exame com nódulo benigno segmentado**

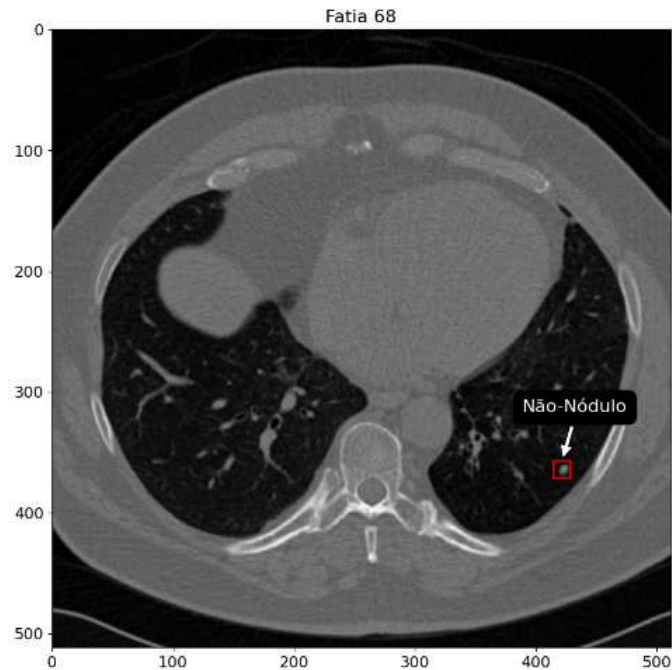


Fonte: Autoria própria (2024).

## 5.6 Trabalhos com topologias diferentes

Para fins de contextualização, é útil analisar o trabalho em comparação com o cenário tecnológico de DL atual, como forma de atestar a relevância do mesmo. Para tal, analisou-se

**Figura 29 – Exame com não-nódulo segmentado**



**Fonte: Aatoria própria (2024).**

três modelos relevantes da literatura que também utilizaram o conjunto de dados LIDC-IDRI. Os modelos comparados incluem uma topologia desenvolvida por Huang *et al.* (2020) que apresentou alta assertividade, o sistema *DeepLung* desenvolvido por Zhu *et al.* (2018) e o sistema de aprendizado profundo 3D para detecção de nódulos pulmonares de Gruetzemacher, Gupta e Paradise (2018). Cada um destes trabalhos tem suas respectivas particularidades e diferem bastante em quesitos como topologias e métricas.

O primeiro estudo, conduzido por Huang *et al.* (2020), apresenta uma abordagem inovadora que combina Rede Neural Convolutiva de Transferência Profunda, do inglês Deep Transfer Convolutional Neural Network (DTCNN) e Máquina de Aprendizado Extremo, do inglês Extreme Learning Machine (ELM) para classificar nódulos pulmonares. Esta metodologia utiliza um DTCNN pré-treinado para extrair características relevantes das imagens de TC, seguido pela aplicação de um classificador ELM para determinar se os nódulos são benignos ou malignos. Os resultados foram notáveis, com o modelo alcançando uma precisão de classificação de 94,57%, sensibilidade de 93,69%, especificidade de 95,15% e uma área sob a curva ROC (AUC) de 94,94% no conjunto de dados LIDC-IDRI. A topologia do modelo capitaliza as vantagens do aprendizado profundo para extração de características, combinadas com a rapidez e eficiência da ELM na classificação, oferecendo uma abordagem sinérgica para o diagnóstico de nódulos pulmonares.

O segundo estudo, realizado por Zhu *et al.* (2018), introduz o *DeepLung*, um sistema composto por dois componentes: um para detecção de nódulos e outro para classificação. O componente de classificação utiliza Redes de Caminho Duplo 3D, do inglês Dual Path Network

(DPN) para distinguir entre nódulos benignos e malignos. O *DeepLung* demonstrou ser altamente eficaz, superando abordagens de ponta e até mesmo o desempenho de médicos experientes em alguns casos. A utilização de DPNs 3D permite que o sistema aprenda eficientemente hierarquias espaciais em nódulos pulmonares, contribuindo para sua alta precisão tanto na detecção quanto na classificação de nódulos.

O terceiro estudo, de Gruetzemacher, Gupta e Paradise (2018), explora o uso de aprendizado profundo 3D para a detecção automática de nódulos pulmonares. O sistema emprega dois modelos de aprendizado profundo 3D: um para a geração de candidatos a nódulos e outro para a redução de falsos positivos. Os resultados mostraram uma taxa de detecção geral de 89,29% com 1,789 falsos positivos por varredura no conjunto de dados LIDC-IDRI. A abordagem inovadora de combinar arquiteturas de rede neural 3D para diferentes etapas do processo de detecção demonstra o potencial do aprendizado profundo para generalizar a dados não vistos e oferecer uma solução abrangente para a detecção de nódulos pulmonares.

Embora os resultados deste trabalho possam ter sido inferiores aos de alguns estudos de ponta, é importante destacar o valor significativo da abordagem adotada. O compromisso com a transparência e a acessibilidade, evidenciado pelo código aberto, legível e bem comentado, é um aspecto notável deste projeto. Essa abordagem não apenas facilita a compreensão e a reprodução do trabalho por outros pesquisadores, mas também contribui para o avanço coletivo do conhecimento na área. Além disso, a clareza e a organização do código podem ser especialmente benéficas para a comunidade científica, permitindo que outros construam sobre a base do trabalho e, potencialmente, melhorem os resultados no futuro.

Os resultados obtidos reforçam a viabilidade do modelo como uma ferramenta promissora para auxiliar no diagnóstico precoce do câncer de pulmão, potencializando a precisão na identificação e classificação de nódulos pulmonares em imagens de TC. A comparação com modelos estabelecidos na literatura evidencia não apenas uma similaridade com o estado da arte do domínio do trabalho, mas também a importância de ter-se código aberto para que a tecnologia possa ser desenvolvida em trabalhos futuros, até mesmo por outros autores interessados pelo assunto.

## 6 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo principal desenvolver um sistema integrado baseado em algoritmos de aprendizado de máquina profundo para a segmentação e classificação de nódulos pulmonares em exames de tomografia computadorizada. Através da realização deste estudo, foi possível atingir os objetivos específicos propostos de maneira satisfatória.

Inicialmente, foi desenvolvido um algoritmo capaz de receber e processar exames da base de dados LIDC-IDRI, transformando os dados binários de uma forma que pudessem ser processados pelas ferramentas utilizadas nas etapas subsequentes. Em seguida, implementou-se um algoritmo de segmentação baseado em técnicas de DL. Este algoritmo foi capaz de identificar com relativa precisão as regiões de interesse nos exames, destacando as áreas potencialmente contendo nódulos pulmonares. A acurácia da segmentação foi validada através de métricas específicas estabelecidas para a avaliação do modelo, demonstrando a eficácia do método empregado.

Posteriormente, foi desenvolvido um algoritmo de classificação, também baseado em DL, que teve como objetivo classificar as regiões segmentadas quanto à presença de nódulos pulmonares. Este algoritmo forneceu probabilidades associadas a cada região, indicando a confiança na presença de nódulos. Os resultados obtidos nesta etapa foram encorajadores, mostrando a capacidade do algoritmo em distinguir entre nódulos benignos e malignos. A validação quantitativa do sistema classificador, uma das métricas mais importantes para a avaliação do modelo, foi de 85% de detecção do total de nódulos presentes na base de dados, sendo 15% dos nódulos não identificados como tal.

Todos os algoritmos foram integrados em um único sistema, proporcionando um modelo de aprendizagem reprodutível e modularizado. A integração dos diferentes componentes permitiu a automatização do processo de análise dos exames de tomografia, desde a recepção dos dados até a classificação dos nódulos, facilitando o uso prático do sistema.

Ressalta-se que os resultados poderiam ser melhorados em alguns quesitos além da validação quantitativa, por exemplo, no fornecimento de falsos positivos. A questão de fornecer falsos positivos é melhor vista do que uma situação onde fossem fornecidos falsos negativos, visto que, caso a solução fosse implantada em ambiente médico, é muito mais desejável que o *software* mostre estruturas que não são nódulos como nódulos e solicite ao médico responsável uma revisão destas estruturas classificadas, do que simplesmente oculte nódulos reais. Esse comportamento do sistema poderia induzir o médico ao erro, se o mesmo confiasse cegamente no programa.

Porém, é relevante destacar que o sistema proposto neste trabalho não tem autonomia nenhuma para elaborar laudos médicos, portanto, seus resultados (caso o trabalho fosse implantado em ambiente de tomografia real) seriam meramente sugestões que deveriam passar por uma análise mais detalhada pelo médico encarregado de elaborar o laudo para determinado exame.

Com relação aos modelos desenvolvidos, várias melhorias ainda podem ser feitas. As estratégias de treinamento dos modelos, por exemplo, poderiam ser exploradas de algumas formas. Além dos dados prontamente disponíveis, contratar um médico consultor para examinar a base de dados em mais detalhe e identificar cada nódulo com um tipo de câncer, e então forçar o modelo a relatar esse tipo, poderia resultar em um treinamento mais eficiente. O custo de terceirizar esse trabalho é proibitivo para um projeto de estudo, mas poderia fazer sentido em contextos comerciais. Outra maneira de usar várias avaliações seria treinar vários modelos em vez de um, cada um treinado com as anotações fornecidas por um radiologista individual. Na inferência, poderia-se então agregar os modelos, por exemplo, fazendo a média de suas probabilidades de saída.

No sentido de malignidade de um nódulo, poderíamos voltar aos dados de anotação fornecidos pelo, por exemplo, *PyLIDC*. O *PyLIDC* é um módulo *Python* onde os dados da base LIDC-IDRI são representados em um banco de dados SQL e podem, então, serem manipulados por meio de comandos SQL. Esse módulo também fornece outras classificações para cada anotação (subtileza, estrutura interna, calcificação, esfericidade, definição de margem, lobulação, espiculação e textura), dados que podem nos fornecer *insights* sobre os nódulos, como a possibilidade de construção de uma escala de malignidade. No entanto, seria necessário um estudo mais sério e profundo de como esses fatores afetam a malignidade de um tumor.

Quanto à segmentação, um trabalho futuro é o desenvolvimento de um teste para analisar se as máscaras fornecidas pelo *PyLIDC* funcionam melhor do que as implementadas neste projeto. Como os dados do *LIDC* têm anotações de vários radiologistas, seria possível agrupar os nódulos em grupos de "alta concordância" e "baixa concordância". Pode ser interessante verificar se isto corresponde a "fácil" e "difícil" de classificar os nódulos, em termos de ver se o classificador proposto acerta quase todos os fáceis e só tem dificuldade com os que eram mais ambíguos para os especialistas humanos. Ou poderia-se abordar o problema sob outro olhar, definindo o quão difícil é detectar os nódulos em termos do desempenho do nosso modelo: "fácil" (classificado corretamente após uma ou duas épocas de treinamento), "médio" (eventualmente classificado corretamente) e "difícil" (classificado persistentemente de forma errada).

Por fim, a realização deste trabalho foi uma oportunidade única para solidificar os conhecimentos adquiridos ao longo do curso de forma integradora, como cálculo, estatística, programação, processamento de imagens, processamento de dados e aprendizado de máquina. Além disso, foi uma ótima oportunidade de se estender o interesse de pesquisa em desenvolvimento de tecnologias específicas para a luta contra o câncer, visto que este trabalho pode ser estendido livremente por qualquer estudante da universidade, onde talvez seja localizado por outros estudantes e desperte o interesse dos mesmos, ou até mesmo por pessoas da comunidade externa à UTFPR, pois o trabalho foi publicado com código aberto e qualquer pessoa que desejar pode utilizá-lo da forma que bem entender.

## REFERÊNCIAS

- ARMATO, S. G. r. *et al.* The lung image database consortium (lidc): an evaluation of radiologist variability in the identification of lung nodules on ct scans. **Academic Radiology**, Elsevier, v. 14, n. 11, p. 1409–1421, 2007.
- ARMATO, S. G. r. *et al.* The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans. **Medical Physics**, 2011.
- BEAULIEU-JONES, C. F. *et al.* Privacy-preserving generative deep neural networks support clinical data sharing. **Nature Biotechnology**, 2018. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/28117445/>. Acesso em: 14 out. 2023.
- BIDGOOD, J. W. D. *et al.* Understanding and using dicom, the data interchange standard for biomedical imaging. **J Am Med. Inform. Assoc.**, 1997.
- CANNY, J. A computational approach to edge detection. **IEEE Trans Pattern Anal Mach Intell.**, v. 8, n. 6, p. 679–698, jun. 1986. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/21869365/>. Acesso em: 10 set. 2023.
- CARUANA, R. *et al.* Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. **Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**, 2015. Disponível em: <https://dl.acm.org/doi/10.1145/2783258.2788613>. Acesso em: 14 out. 2023.
- CAUSEY, J. L. *et al.* Highly accurate model for prediction of lung nodule malignancy with ct scans. **Scientific Reports**, Nature Publishing Group, v. 7, n. 1, 2017.
- COZMAN, F.; PLONSKY, A. G.; NERI, H. **Inteligência artificial: avanços e tendências**. 2021. Disponível em: <https://www.livrosabertos.sibi.usp.br/portaldelivrosUSP/catalog/download/650/579/2181?inline=1>. Acesso em: 12 out. 2023.
- ELNAKIB, A.; AMER, H. M.; ABOU-CHADI, F. E. Early lung cancer detection using deep learning optimization. 05 2020. Disponível em: <https://www.learntechlib.org/p/217904>. Acesso em: 12 out. 2023.
- ESTEVA, A. *et al.* Dermatologist-level classification of skin cancer with deep neural networks. **Nature**, 2017. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/28117445/>. Acesso em: 14 out. 2023.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Acesso em: 10 set. 2023.
- GRUETZEMACHER, R.; GUPTA, A.; PARADICE, D. 3d deep learning for detecting pulmonary nodules in ct scans. **Journal of the American Medical Informatics Association**, v. 25, p. 1301–1310, 2018.
- HUANG, X. *et al.* Deep transfer convolutional neural network and extreme learning machine for lung nodule diagnosis on ct images. **Knowl. Based Syst.**, v. 204, p. 106230, 2020.
- HUSSEIN, S. *et al.* Tumornet: Lung nodule characterization using multi-view convolutional neural network with gaussian process. p. 1007–1010, 2017.

- ISAACS, R. S.; SYKES, J. M. Anatomy and physiology of the upper airway. **Anesthesiology Clinics of North America**, v. 20, n. 4, p. 733–745, 2002. ISSN 0889-8537. Disponível em: [https://doi.org/10.1016/S0889-8537\(02\)00017-2](https://doi.org/10.1016/S0889-8537(02)00017-2). Acesso em: 06 set. 2023.
- ISENSEE, F. *et al.* nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. **Nat Methods**, v. 18, p. 203–211, 2021. Disponível em: <https://www.nature.com/articles/s41592-020-01008-z#citeas>. Acesso em: 14 out. 2023.
- KATHIRAVELU, P.; SHARMA, A.; SHARMA, P. Understanding scanner utilization with real-time dicom metadata extraction. **IEEE Access**, v. 9, p. 10621–10633, 2021.
- KIM, H. The definition of convolution in deep learning by using matrix. **Journal of Engineering and Applied Sciences**, v. 14, n. 7, p. 2272–2275, 2019.
- LOVERDOS, K. *et al.* Lung nodules: A comprehensive review on current approach and management. **Ann Thorac Med**, v. 15, n. 6, p. 226–238, out. 2019. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6784443/>. Acesso em: 06 set. 2023.
- LOWEKAMP, B. C. *et al.* The design of simpleitk. **Front. Neuroinform.**, 2013.
- MIDTHUN, D. E. Midthun, david e. **Revista da AcBO-ISSN 2316-7262**, v. 3, n. 1, 2013. Disponível em: <http://www.rvacbo.com.br/ojs/index.php/ojs/article/viewFile/147/97>. Acesso em: 30 set. 2023.
- MYERS, D. J.; WALLEN, J. M. **Lung Adenocarcinoma**. [updated 2023 jun 12]. Treasure Island (FL), 2023. Disponível em: <https://www.ncbi.nlm.nih.gov/books/NBK519578/>. Acesso em: 06 set. 2023.
- OPAS/OMS. **Câncer**. 2020. Disponível em: <https://www.paho.org/pt/topicos/cancer>. Acesso em: 06 set. 2023.
- OTSU, N. A threshold selection method from gray-level histograms. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 9, n. 1, p. 62–66, 1979. Disponível em: 10.1109/TSMC.1979.4310076. Acesso em: 10 set. 2023.
- PAIVA, M. Do diagnóstico e da interpretação radiográfica. **Revista da AcBO-ISSN 2316-7262**, v. 3, n. 1, 2013. Disponível em: <http://www.rvacbo.com.br/ojs/index.php/ojs/article/viewFile/147/97>. Acesso em: 30 set. 2023.
- PASZKE, A. *et al.* Pytorch: An imperative style, high-performance deep learning library. **CoRR**, abs/1912.01703, 2019. Disponível em: [https://proceedings.neurips.cc/paper\\_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf). Acesso em: 10 set. 2023.
- PEREIRA, F. R.; FERRARI, L. d. O. Proposta de uma solução computacional para detecção de nódulos pulmonares. **Anais do XVIII Simpósio Brasileiro de Computação Aplicada à Saúde**, v. 3, n. 1, 2018. Disponível em: <https://sol.sbc.org.br/index.php/sbcas/article/view/3672>. Acesso em: 12 out. 2023.
- PIANYKH, O. S. **Digital Imaging and Communications in Medicine (DICOM): A Practical Introduction and Survival Guide**. [S.l.]: Springer, 2009. <http://www.deeplearningbook.org>. ISBN 978-3642108495. Acesso em: 10 set. 2023.
- REEVES, A. P. *et al.* The lung image database consortium (lidc): a comparison of different size metrics for pulmonary nodule measurements. **Academic Radiology**, Elsevier, v. 14, p. 1475–1485, 12 2007.

- RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. **CoRR**, abs/1505.04597, ago. 2015. Disponível em: [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28). Acesso em: 10 set. 2023.
- ROSSER, J.; FFICM, F. *et al.* Computed tomography of the chest—ii: clinical applications. **BJA Education**, v. 16, n. 1, p. 15–20, jan. 2016. Disponível em: <https://doi.org/10.1093/bjaceaccp/mku063>. Acesso em: 04 set. 2023.
- SAPONARA, S.; ELHANASHI, A. **Impact of Image Resizing on Deep Learning Detectors for Training Time and Model Performance**. Cham: Springer International Publishing, 2022. 10–17 p. ISBN 978-3-030-95498-7.
- SCHLEGL, T. *et al.* Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. **International Conference on Information Processing in Medical Imaging**, 2017. Disponível em: [https://doi.org/10.1007/978-3-319-59050-9\\_12](https://doi.org/10.1007/978-3-319-59050-9_12). Acesso em: 14 out. 2023.
- SHEN, D.; WU, G.; SUK, H. Deep learning in medical image analysis. **Annu Rev Biomed Eng.**, v. 19, p. 221–248, 2017. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/28301734/>. Acesso em: 10 set. 2023.
- SMITH, B. H.; KAELIN, W. Developing image analysis pipelines of whole-slide images: Pre- and post-processing. **Journal of Clinical and Translational Science**, Cambridge University Press, v. 5, n. 2, p. 81–90, 2019.
- SPRAWLS, P. J. **Physical Principles of Medical Imaging**. [S.l.]: Medical Physics Pub Corp, 1995.
- STEVENS, E.; ANTIGA, L.; VIEHMANN, T. **Deep learning with PyTorch**. [S.l.]: Manning Publications, 2020.
- WANG, F.; CASALINO, L. P.; KHULLAR, D. Deep Learning in Medicine—Promise, Progress, and Challenges. **JAMA Internal Medicine**, v. 179, n. 3, p. 293–294, mar. 2019. ISSN 2168-6106. Disponível em: <https://doi.org/10.1001/jamainternmed.2018.7117>. Acesso em: 30 set. 2023.
- WANG, L. **Deep Learning Techniques to Diagnose Lung Cancer**. 2022. Disponível em: <https://www.mdpi.com/2072-6694/14/22/5569>. Acesso em: 12 out. 2023.
- WHITING, P.; FFICM, F. *et al.* Computed tomography of the chest: I. basic principles. **BJA Education**, v. 15, n. 6, p. 299–304, dez. 2015. Disponível em: <https://doi.org/10.1093/bjaceaccp/mku063>. Acesso em: 04 set. 2023.
- ZEYNETTIN, A. *et al.* Deep learning for brain mri segmentation: State of the art and future directions. **J Digit Imaging**, 2017. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/28577131/>. Acesso em: 14 out. 2023.
- ZHU, W. *et al.* Deeplung: Deep 3d dual path nets for automated pulmonary nodule detection and classification. **2018 IEEE Winter Conference on Applications of Computer Vision (WACV)**, p. 673–681, 2018.