Trabalho de Conclusão de Curso

Detecção automática de nódulos pulmonares utilizando Transformers

Nilson Sales de Carvalho
nsc@ic.ufal.br

Orientador:
Prof. Dr. Marcelo Costa Oliveira

Maceió, Julho de 2022

Nilson Sales de Carvalho

Detecção automática de nódulos pulmonares utilizando Transformers

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Engenharia de Computação do Instituto de Computação da Universidade Federal de Alagoas.

Orientador:

Prof. Dr. Marcelo Costa Oliveira

Maceió, Julho de 2022

Catalogação na Fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 - 1767

C331d Carvalho, Nilson Sales de.
Detecção automática de nódulos pulmonares utilizando Transformers / Nilson Sales de Carvalho. – 2022.
45 f. : il.

Orientador: Marcelo Costa Oliveira.
Monografia (Trabalho de conclusão de curso em Engenharia de Computação) - Universidade Federal de Alagoas, Instituto de Computação. Maceió, 2022.

Bibliografia: f. 38-43.
Apêndices: f. 44-45.

1. Detecção de objetos. 2. Transformers - Aprendizagem profunda. 3. Processamento de imagem assistida por computador. 4. Nódulos pulmonares. I. Título.

CDU: 004.81:159.953.5

Agradecimentos

À minha família, em especial à minha mãe, Cirlene, e ao meu irmão, Dielson, que sempre foram a base para que eu pudesse ter uma boa educação e trilhar sempre o melhor caminho;

A todos os meus amigos, em especial ao Bruno, Crispim e João Paulo, que me acompanharam durante boa parte da minha formação, que sempre me incentivaram a nunca desistir e que além de amigos são inspirações profissionais;

A todos os professores que me ajudaram ao longo do percurso da faculdade, que me deram a base do conhecimento acadêmico e profissional e que acreditaram no meu potencial;

Ao professor Leonardo Viana, que foi meu orientador durante boa parte do curso;

Ao professor Marcelo, por todo o apoio que me deu durante a pesquisa e no desenvolvimento deste TCC;

Aos professores Baldoino e Erick, pelo interesse e disponibilidade em participar da banca examinadora;

A todos os funcionários do IC.

"What we do in life... echoes in eternity."

– Gladiator

Resumo

O câncer de pulmão (CP) é o segundo tipo de câncer mais prevalente no mundo e o mais letal, sendo responsável por uma a cada cinco mortes por câncer no mundo. As chances de sobrevida dos pacientes detectados com este tipo de câncer aumentam consideravelmente quando o diagnóstico é realizado de maneira precoce, com a taxa de sobrevida em 5 anos chegando a até 70%. O diagnóstico do CP é realizado por radiologistas através de imagens de Tomografia Computadorizada (TC), porém tal diagnóstico é uma tarefa complexa e sujeita a erros, uma vez que os nódulos podem apresentar tamanho muito pequeno ou estar localizado próximos a outras estruturas anatômicas. Os profissionais da área sofrem ainda de fadiga e pressa no diagnóstico, já que a quantidade de exames a serem analisados é geralmente muito alta. Através das ferramentas de auxílio computadorizado (CAD), esse processo de diagnóstico pode ser automatizado, reduzindo tempo e esforço dos especialista, bem como melhorar a confiança no diagnóstico. Atualmente as técnicas de deep learning (DL), em especial com o uso de CNNs, são o estado-da arte para a detecção automática de nódulos pulmonares, porém com a introdução da arquitetura transformer às tarefas de visão computacionais, abre-se mais uma área a ser explorada. Nesse contexto, o objetivo deste trabalho foi apresentar um sistema de detecção automática de nódulos pulmonares em imagens de TC utilizando a arquitetura transformer. Avaliamos ainda a hipótese de que a arquitetura transformer seja tão eficiente quanto os modelos de CNN na detecção de nódulos pulmonares.

Palavras-chave: Detecção de objetos, Transformers, Aprendizagem profunda, Processamento de imagens assistida por computador, Nódulos pulmonares

Abstract

Lung cancer is the second most prevalent type of cancer in the world and the most deadly, accounting for one in five cancer deaths worldwide. The chances of survival of a patient detected with lung cancer increase considerably when the diagnosis is made early, with the 5-year survival rate reaching up to 70%. The lung cancer diagnosis is performed by radiologists through Computed Tomography (CT) images, but this is a complex task and subject to errors, since the nodules may be very small or located close to other anatomical structures. Professionals in the area also suffer from fatigue and a rush to diagnose, given that the number of exams to be analyzed is usually very high. Computer aided diagnosis (CAD) systems can automate this diagnosis process, reducing the time and effort from the specialists, as well as improving diagnostic confidence. Currently, deep learning (DL) techniques, especially with the use of CNNs, are the state-of-the-art for the automatic detection of pulmonary nodules, but with the introduction of the transformer architecture to computer vision tasks, a new area can be explored. In this context, the objective of this work is to present a system for the automatic detection of pulmonary nodules in CT images using the transformer architecture. We also evaluated the hypothesis that the transformer architecture is as efficient as the CNN models in the detection of pulmonary nodules.

Keywords: Object detection, Transformers, Deep learning, Computer-assisted image processing, Pulmonary nodules

Conteúdo

Lista de Figuras

1

Introdução

1.1 Motivação

Estatísticas globais de câncer indicam que o câncer de pulmão (CP) é o segundo tipo de câncer mais prevalente no mundo, com 2,2 milhões de novos casos, e o mais letal, com 1,8 milhões de mortes registradas no ano de 2020, o que representa aproximadamente um em cada 10 (11,4%) diagnósticos de câncer e uma a cada 5 (18%) mortes por câncer no mundo. Em homens, o CP é a principal causa de morbidade e mortalidade por câncer, enquanto em mulheres, é o terceiro em incidência, atrás de câncer de mama e de cólon, e o segundo em mortalidade, atrás apenas do câncer de mama (OMS, 2020). Dentre as principais causas do CP estão o tabagismo, poluição atmosférica, exposição a substâncias químicas, e influência genética, e cerca de metade dos pacientes diagnosticados com câncer de pulmão vêm a óbito ainda no primeiro ano da descoberta da doença, pelo fato do diagnóstico muitas vezes ser realizado em estágios avançados da doença, comprometendo outras regiões do organismo (Lima et al., 2019).

O diagnóstico precoce do câncer de pulmão é imprescindível para aumentar as chances de sobrevida do paciência, com taxa de sobrevida de 5 anos chegando a 70% quando diagnosticado no estágio I (Blandin Knight et al., 2017). No Brasil, no entanto, estima-se que 70% dos casos de câncer de pulmão sejam diagnosticados já em estado avançado (estágio II) ou metastático (estado IV), e que apenas 9% sejam diagnosticados no estágio I, número inferior ao aferido em alguns países desenvolvidos (Araujo et al., 2018). O atraso nos diagnósticos foi agravado mais recentemente por conta da pandemia de COVID-19, que limitou o acesso a consultas e atendimentos para dar vazão aos casos de COVID (Araujo-Filho et al., 2020). É imprescindível, portanto, que programas que visem o diagnóstico precoce de nódulos pulmonares sejam implementados para aumentar as chances de sobrevivência da população acometida pelo câncer de pulmão.

A Tomografia Computadorizada (TC) é a principal ferramenta utilizada pelos radiologistas para detectar nódulos pulmonares, pois fornecem imagens 3D em alta resolução e contraste,

refletindo as diferenças de intensidade, textura e forma dos tumores. Porém, o diagnóstico de câncer de pulmão através do uso de imagens de TC, no entanto, apresenta diversos desafios e limitações para os profissionais da área. Os nódulos podem ter tamanho muito pequeno, estar localizados em estruturas anatômicas complexas da região pulmonar (e.g. vasos e pleura) e ainda apresentar contraste similar ao tecido pulmonar. É comum ainda que profissionais da área de radiologia enfrentem outras dificuldades, como fadiga e pressa no diagnóstico, condições externas adversas, como ruído nas imagens e baixa luminosidade do ambiente de trabalho, e aumento contínuo do volume de imagens, o que pode resultar em falhas no diagnóstico do câncer (Lima et al., 2019; Patz et al., 2014; Degnan et al., 2019).

A falha ao não detectar um nódulo potencialmente cancerígeno pode ter sérias consequências tanto para o paciente quanto para o radiologista, e é uma das causas mais comuns de ações por imperícia contra os especialistas (Emani et al., 2019). Estudos mostram ainda que em apenas 68% das vezes os nódulos pulmonares são devidamente diagnosticados pelo radiologista, subindo para 82% das vezes quando examinados por dois radiologistas (Nasrullah et al., 2019).

Com o objetivo de melhorar o diagnóstico por imagem médica, os sistemas para o auxílio à detecção e diagnóstico (CAD) - do inglês, *Computer-Aided Diagnosis* - são importantes ferramentas que podem fornecer suporte à decisão do radiologista, funcionando como uma segunda opinião (Halder et al., 2020). As ferramentas CAD podem assim automatizar o processo do diagnóstico, reduzindo o tempo e esforço necessários para a análise, bem como melhorar a confiabilidade e capacidade de repetição da tarefa (Ferreira et al., 2018; Choi and Choi, 2013). Os sistemas CAD envolvem tanto a tarefa de localização de lesões nas imagens médicas (CADe) quanto a classificação dessas lesões em malignas ou benignas (CADx), o que traz diversos desafios para a área (Firmino et al., 2016).

Atualmente, as técnicas de Deep Learning (DL) são o estado-da-arte em aplicações CAD para a detecção de nódulos pulmonares (Halder et al., 2020), e diversos trabalhos (McBee et al., 2018; Adams et al., 2021; Halder et al., 2020) evidenciram o potencial do uso de DL na detecção de nódulos pulmonares, além de ratificar o custo-benefício da utilização dos sistemas CAD em centros de saúde. Dentre as arquiteturas de DL, as Redes Neurais Convolucionais (CNNs) têm se desenvolvido como sendo a principal arquitetura na área de informática médica por conta dos resultados excepcionais obtidos em visão computacional (Ravì et al., 2016). Diversos trabalhos (Ardila et al., 2019; Cui et al., 2020; Li and Fan, 2020) utilizando modelos de CNN para a detecção de nódulos demonstraram ótimos resultados. Contudo, apesar dos avanços dos modelos de CNN, o número de falsos-positivos na detecção de nódulos pulmonares continua sendo um empecilho para a implantação dos sistemas em clínicas médicas (Liang et al., 2021; Shaukat et al., 2019). Segundo Liu et al. (2022), a maioria dos modelos de CNN possuem um campo receptivo limitado devido ao pequeno *kernel* convolucional (3x3 ou 5x5) geralmente usado em prol da eficiência computacional. O campo receptivo reduzido faz com que os modelos de CNN estejam limitados a modelar relações locais, não sendo capazes de representar grandes distâncias entre os pixels das imagens (Li et al., 2022). Isso resulta em

um problema, pois a classificação dos NP é baseada em características locais e globais. Por exemplo, os atributos locais descrevem detalhes relacionados a densidade e a textura dos NP, enquanto as características globais descrevem a forma e o tamanho dos NP.

O *Transformer* (Vaswani et al., 2017) é uma arquitetura *encoder-decoder* que revolucionou a área de Processamento de Linguagem Natural (PLN) ao apresentar uma estrutura mais simples e sem a necessidade da utilização de convoluções. Seu principal componente é o mecanismo de atenção (*attention*), que utiliza dependências globais entre entradas e saídas. O *transformer* se tornou o estado-da-arte em tarefas de tradução, e desde então passou a ser utilizada em diversas áreas (Jumper et al., 2021; Radford et al., 2021; Zandie and Mahoor, 2021). Nesse contexto o *Vision Transformer* (ViT) (Dosovitskiy et al., 2020) surgiu como uma adaptação do *Transformer* para tarefas de visão computacional, funcionando em combinação com arquiteturas de CNNs tradicionais ou as substituindo por completo. Diversos modelos inspirados no ViT original surgiram posteriormente e obtiveram resultados excelentes em tarefas de detecção de objetos (Carion et al., 2020), segmentação (Wang et al., 2021), geração de imagens sintéticas (Jiang et al., 2021) entre outras tarefas.

Por conta das dependências de longo-alcance entre os *patches* da imagem em combinação com o mecanismo de atenção, os modelos ViT conseguem cobrir todo o espaço de entrada (Zhu et al., 2021). Essa característica faz com que os modelos ViT possam focar no contexto global da imagem, criando dependências de longo alcance, o que os difere das CNNs, que focam nas características locais. Isso oferece uma maior robustez a oclusões e perturbações e faz com que os modelos ViT sejam menos enviesados com relação a texturas locais (Naseer et al., 2021). Diversos trabalhos, no entanto, avaliaram a combinação dos modelos ViT com CNNs, visando capturar melhor a distribuição dos dados nas imagens e obter um aumento de performance (Wu et al., 2021; Shamshad et al., 2022; Zhu et al., 2020). Enquanto a CNN foca sua atenção em texturas locais, usando campos receptivos menores, o ViT direciona sua atenção para relacionamentos globais entre os pixels da imagem, através dos seus campos receptivos de longo-alcance.

Os modelos ViT têm atraído bastante interesse da comunidade médica para ser adaptado para aplicações de imagens médicas por conta dos excelentes resultado obtidos em tarefas de visão computacional, e se tornou um assunto recorrente em conferências e jornais da área (Shamshad et al., 2022). Na área de detecção de nódulos com a utilização de ViT, Zhu et al. (2022) propôs uma arquitetura fim-a-fim que utiliza uma rede residual em formato de U em combinação com o mecanismo de atenção, e obteve 95% de sensibilidade na detecção de nódulos pulmonares com um número bastante inferior de parâmetro em comparação a modelos de CNN, ao mesmo tempo em que conseguiu reduzir o número de falsos positivos. Niu and Wang (2022) propôs um modelo ViT 3D baseado em região para identificar nódulos pulmonares em um conjunto de regiões candidatas. O modelo proposto conseguiu resultados superiores (3%) na detecção de nódulos em comparação a modelos CNN 3D estado-da-arte.

1.2 Objetivos Gerais

Considerando os problemas e as soluções descritos anteriormente, este trabalho tem como objetivo principal desenvolver um sistema de detecção automática de nódulos pulmonares utilizando a arquitetura Transformer. O software desenvolvido poderá ser utilizado para auxiliar profissionais da área no diagnóstico clínico do câncer de pulmão. Como objetivo secundário, avaliamos a hipótese que a arquitetura Transformer é tão eficiente quanto os modelos de CNN na detecção de nódulos pulmonares.

1.3 Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Realizar o pré-processamento das tomografias computadorizadas da região torácica;
- Realizar a segmentação da região interna do pulmão;
- Geração da base de imagens em 2D para uso no modelo de detecção de objetos;
- Adaptação do Transformer para o uso com a base de dados de nódulos pulmonares;
- Calibração dos hiper-parâmetros do modelo para otimização dos resultados;
- Treinamento do modelo e geração dos pesos para inferência;
- Avaliação dos resultados do modelo;

1.4 Estrutura do Trabalho

A estrutura deste trabalho está da seguinte forma:

- Capítulo 2 - Fundamentação Teórica: Este capítulo introduz conceitos fundamentais sobre imagens médicas, processamento de imagens e detecção de objetos, importantes para o entendimento do artigo;
- Capítulo 3 - Materiais e Métodos: Este capítulo apresenta os materiais e as técnicas utilizadas para gerar a base de imagens, bem como configurar e treinar o modelo para a detecção de nódulos pulmonares;
- Capítulo 4 - Resultados e Discussão: Este capítulo apresenta os resultados obtidos e a discussão em torno deles e perante aos resultados obtidos nos trabalhos relacionados;
- Capítulo 5 - Conclusão: Por fim, este capítulo finaliza este trabalho apresentando as conclusões e os trabalhos futuros.

2

Fundamentação Teórica

Este capítulo apresenta conceitos e ferramentas utilizados no desenvolvimento deste projeto. O uso de imagens médicas requer tanto um entendimento de conceitos relacionados à medicina quanto conceitos relacionados ao processamento de imagens.

2.1 Tomografias Computadorizadas

A Tomografia Computadorizada (TC) é formada pela projeção de diversas imagens de raio-X, representada como uma matriz 3D de único canal (nível de cinza). O objetivo da TC é obter uma representação em 3D da estrutura interna de um objeto, ou órgãos, no caso de tomografias médicas. O resultado são imagens em escala de cinza de partes do corpo ou de órgãos selecionados, que são então empilhadas de forma paralela e uniformemente espaçadas, para obter-se uma representação 3D da seção do corpo (figura 2.1) (Lima et al., 2019).

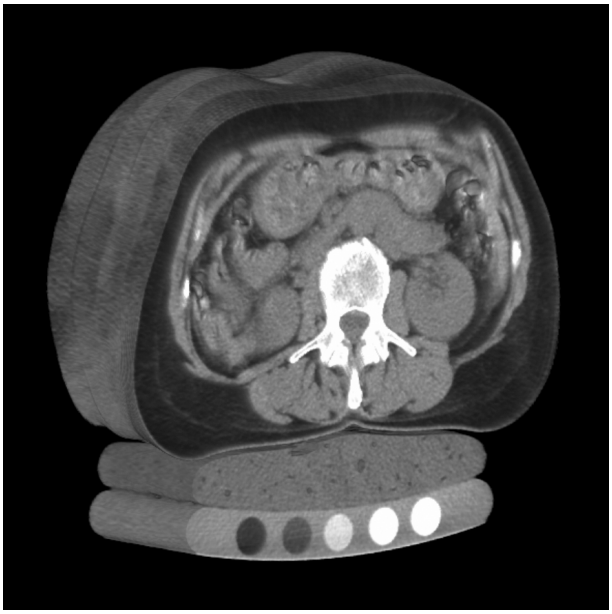


Figura 2.1: Tomografia computadorizada de um torso humano, mostrando pele, órgãos, coluna vertebral e a cama de suporte. (MindwaysCTSoftware, 2012)

Ao empilhar as imagens da TC com espaçamento uniforme entre elas, cada *pixel* da imagem passa a ter a capacidade de representar um volume, e não mais uma área, recebendo o nome de *voxel* (do inglês *volumetric pixel*). Um exemplo visual de um *voxel* pode ser visualizado na figura 2.2.

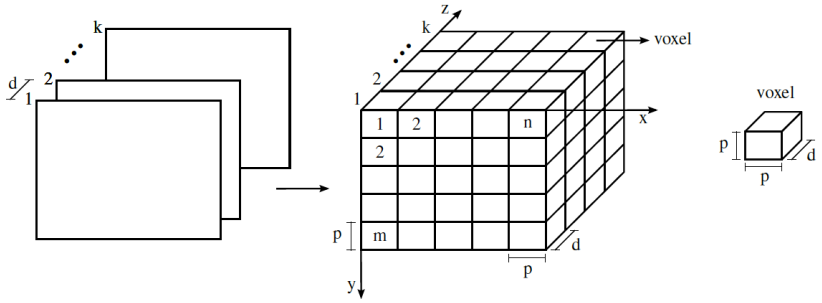


Figura 2.2: Representação visual de um voxel, com m linhas, n colunas e k camadas. Cada elemento do voxel possui dimensões $p \times p$, com profundidade d. (FALCÃO, 1993)

2.1.1 Resolução Espacial e Nível de Intensidade

A resolução espacial de uma imagem pode ser definida como a habilidade em se distinguir pequenos objetos em alto contraste e é limitada pelo tamanho mínimo do pixel (Furquim and Costa, 2009). Quanto maior o número de pixels numa matriz, melhor é a sua resolução espacial, o que permite uma melhor diferenciação espacial entre as estruturas. Embora na radiologia o termo leve em consideração o espaçamento do objeto escaneado (em $pixels/mm^2$), neste

trabalho utilizamos o termo resolução espacial como sendo o tamanho da matriz obtida na TC, ou em outras palavras, o número de pixels utilizado na construção da imagem.

O nível de intensidade, ou resolução de contraste, é a habilidade de um sistema em distinguir dois objetos com diferentes intensidades de sinal. É afetada pela quantização e limitada pela profundidade de bit. Por exemplo, uma resolução de contraste de 8 bits significa que a imagem possui 256 níveis de intensidade naquele canal.

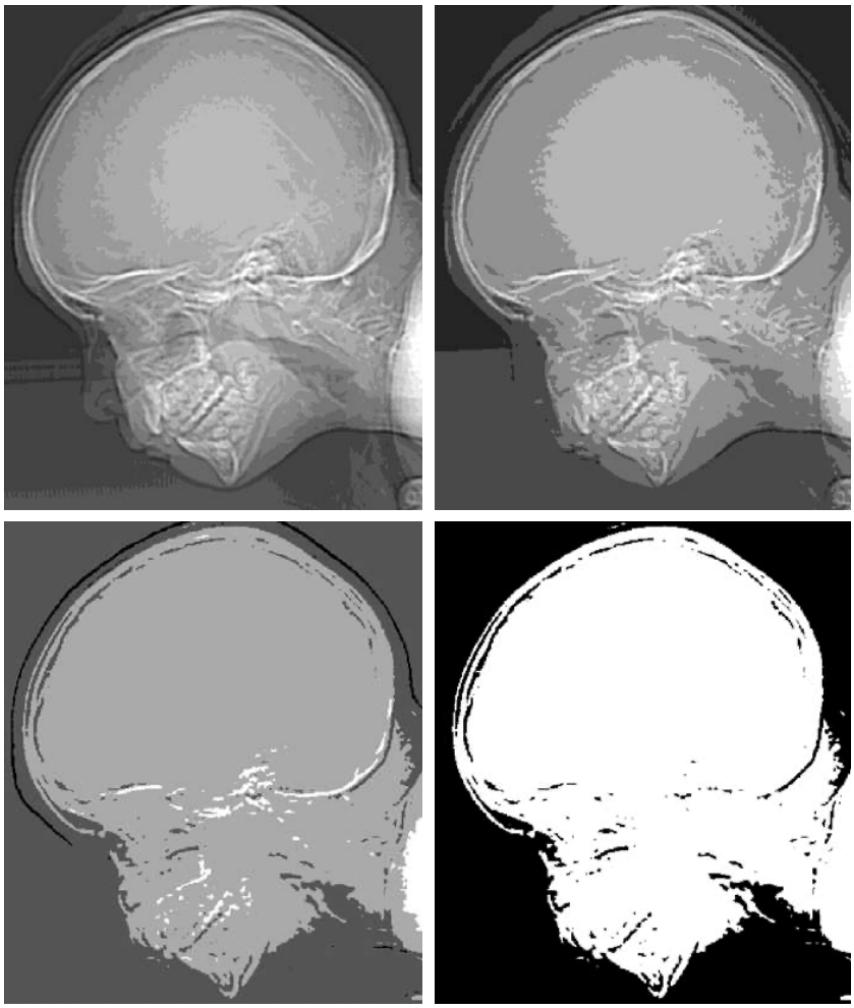


Figura 2.3: Imagem mostrada com nível de intensidade em 16, 8, 4 e 2. (Gonzalez, 2009)

2.1.2 DICOM e Sistemas de Coordenadas

O DICOM (*Digital Imaging and Communications in Medicine*) (Bidgood Jr et al., 1997), é um formato não proprietário bastante utilizado em imagens médicas. As imagens em DICOM contém diversas informações sobre o protocolo do exame e informações técnicas do *scanner*, como vetor de origem, espaçamento e vetor de direção. Os exames neste formato utilizam um sistema de coordenadas anatômicas (Onken et al., 2010), onde o eixo positivo X é definido como sendo à esquerda do paciente, positivo Y como sendo em direção às costas e positivo Z como sendo o eixo que vai dos pés à cabeça, como mostrado na figura 2.4. O sistema de coorde-

nadas anatômicas possui seu ponto de origem posicionado de maneira arbitrária na imagem, e seus valores são em milímetros. Os valores do ponto de origem e o vetor de espaçamento estão disponíveis em metadados do arquivo DICOM. Para utilizar esses valores no sistema matricial, é necessário que cada corte da TC esteja associado a um índice, onde o ponto de origem está localizado no canto superior esquerdo da imagem.

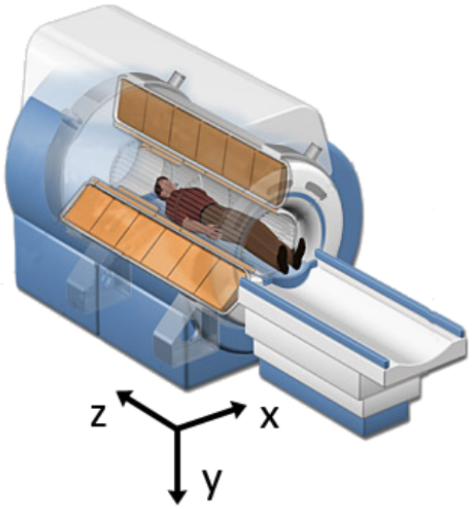


Figura 2.4: Representação do sistema de coordenadas anatômicas. (3DSlicer, 2010)

2.1.3 Conversão entre Sistemas de Coordenadas

A transformação de um sistema de coordenadas matriciais $\begin{pmatrix} i & j & k \end{pmatrix}$ para um sistema de coordenadas anatômicas $\vec{x}$ consiste de uma transformação linear A seguida de uma translação $\vec{t}$.

$$\vec{x} = A \begin{pmatrix} i & j & k \end{pmatrix}' + \vec{t}$$

A matriz de transformação A é uma matriz 3×3, que contém informações sobre direções espaciais e escala de eixos, enquanto $\vec{t}$ é um vetor 3×1 que contém informação sobre a posição de origem.

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} i \\ j \\ k \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

Para representar a translação e a rotação juntos com uma única multiplicação de matrizes, é necessário criar uma matriz quadrada afim, que tem uma dimensionalidade a mais que o nosso espaço. A técnica requer que a matriz A seja acrescida de uma linha extra de zeros, e da coluna do vetor de translação acrescida de 1 na última linha. Todos os demais vetores são escritos em

coordenadas homogêneas.

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & t_1 \\ A_{21} & A_{22} & A_{23} & t_2 \\ A_{31} & A_{32} & A_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ k \\ 1 \end{bmatrix}$$

2.1.4 Escala Hounsfield

A Tomografia Computadorizada utiliza valores na escala Hounsfield (Hounsfield, 1980), um tipo de unidade adimensional universal. As unidades Hounsfield são obtidas a partir de uma transformação linear dos coeficientes de atenuação medidos. Esta transformação é baseada em densidades arbitrariamente atribuídas ao ar e água pura. Nessa escala, o ar tem valor de -1000 HU, o tecido pulmonar de -500, enquanto a água fica em 0 HU. Outros valores da escala Hounsfield podem ser visualizados na figura 2.5.

Geralmente, as TCs usam imagens com tamanho de 12 bits, para armazenar valores entre -1024 e 3071. A visualização desses valores é determinada pela aplicação, através dos valores de janela (*window width*) e nível (*level*), discutidas no próximo subtópico.

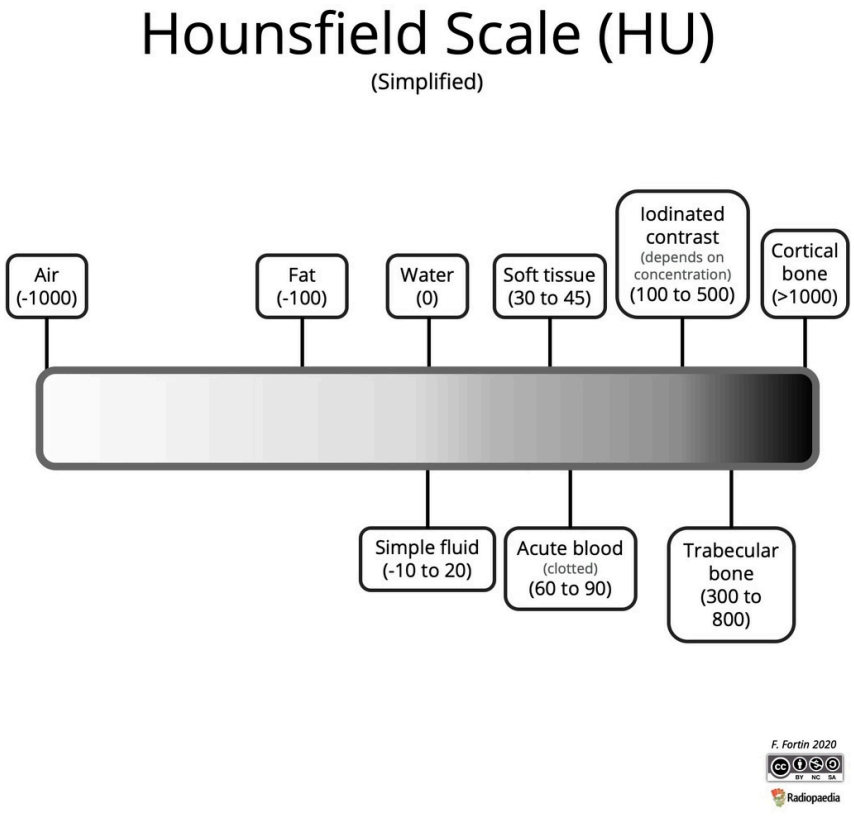


Figura 2.5: Escala de Hounsfield com valores simplificados para referência (Greenway, 2021).

Janelamento

Como explicado por Gaillard et al. (2011), o janelamento (*windowing*) é o processo no qual o componente de escala de cinza da imagem TC de uma imagem é manipulado através dos números da TC. Ao fazer isso, a aparência da imagem é modificada para evidenciar estruturas particulares de interesse. O brilho da imagem é ajustado através do nível da janela, enquanto o contraste é ajustado através da largura da janela.

A largura da janela, como o nome diz, é a medida do intervalo dos números de TC que a imagem possui. Uma largura de janela larga (2000 HU), mostrará um intervalo maior de números de TC. Isso significa que a transição entre as estruturas claras e escuras ocorrerá em uma área de transição maior para uma largura de janela estreita (<1000 HU).

O nível da janela, ou centro da janela, é o ponto médio do intervalo dos números de TC exibidos. Quando o nível da janela é diminuído, a imagem de TC torna-se mais clara. Já quando aumentada, a imagem torna-se mais escura.

Alguns valores típicos (Murphy, 2021) usados para largura de janela e nível para diferentes partes do corpo humano estão listados na tabela 2.1.4.

Tabela de Valores Típicos (HU)		
Órgão	Largura de janela (W)	Nível (L)
Cérebro	80	40
Ossos temporais	2800-4000	600-700
Tecidos moles da cabeça	350–400	20–60
Pulmões	1500	-600
Tecidos moles do abdômen	400	50
Fígado	150	30
Ossos da coluna vertebral	1800	400

2.2 Processamento Morfológico

A palavra morfologia significa o estudo das formas ou estruturas. No contexto de processamento de imagens a morfologia matemática é usada como uma ferramenta para extrair componentes da imagem que são úteis na representação e descrição da geometria das regiões. Operações morfológicas podem ser aplicadas a qualquer tipo de imagem, mas o uso principal é para o processamento binário de imagens, e os operadores principais da morfologia são chamados de *dilatação* e *erosão*.

Como explicado por Solomon and Breckon (2011), uma imagem binária é uma imagem na qual cada pixel assume apenas um valor discreto (1 ou 0). O valor lógico 1 geralmente é descrito como valor 'alto' ou 'verdadeiro', enquanto o valor 0 é descrito como 'baixo' ou 'falso'. Geralmente os pixels com valor 0 são referenciados como *background*, e os pixels com valor 1, como *foreground*. Já um *objeto* consiste de um grupo de pixels conectados. Pelo fato

de imagens binárias naturalmente não possuírem textura (escala de cinza ou colorida), suas únicas propriedades de interesse são formas, tamanhos e localização dos objetos na imagem. Dessa forma, o objetivo de qualquer transformação morfológica é determinar quais pixels de *foreground* vão se tornar *background*, e quais pixels de *background* vão se tornar *foreground*.

2.2.1 Elementos Estruturantes

Elemento estruturante (SE) é a entidade que determina exatamente quais pixels da imagem no entorno de um dado *foreground/background* devem ser considerados na tomada de decisão para mudar ou não o seu valor. Os SE são pequenas matrizes de pixel retangulares com valores 0 ou 1 que deslizam sobre a imagem de maneira similar a *kernels* convolucionais. Elementos estruturais possuem ainda o chamado *pixel central*, que é um elemento importante nas operações morfológicas.

No processamento digital de imagens, é necessário que os elementos estruturantes sejam matrizes retangulares. Para isso, é acrescentado o menor número possível de elementos de *background* (figura 2.6) necessário para formar a matriz retangular. O formato interno do elemento estruturante, no entanto, é arbitrário. Sendo escolhido para se adequar à aplicação ou objetivo específico que temos em mente.

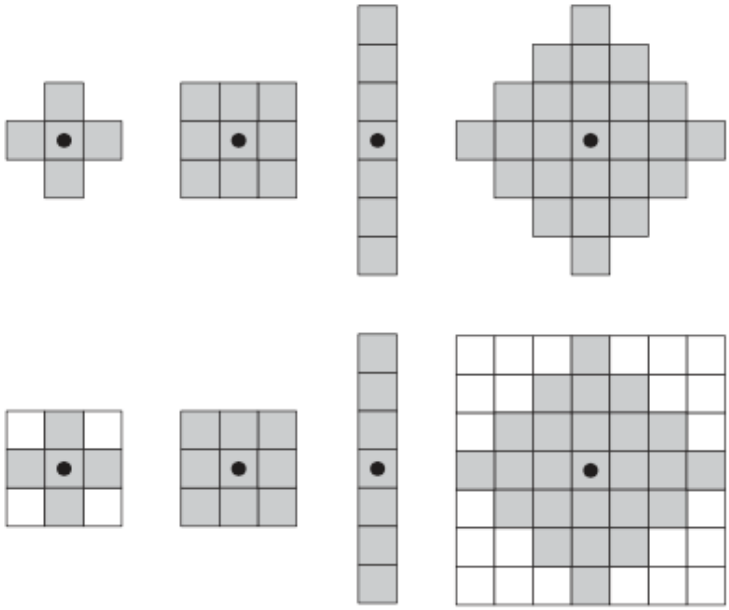


Figura 2.6: Exemplos de elementos estruturantes e sua forma convertida em retângulos. (Gonzalez, 2009)

2.2.2 Dilatação e Erosão

Dois dos principais operadores morfológicos são a *dilatação* e a *erosão*. As mecânicas de ambos operadores funcionam de forma similar.

- **Dilatação**:

  Para realizar a dilatação numa imagem binário, pixel central do elemento estruturante é colocado sobre cada pixel do *background* (valor 0). Se algum dos pixels vizinhos for um elemento de *foreground* (valor 1), o pixel central se torna *foreground*. A dilatação de uma imagem **A** por um elemento estruturante **B** é denotada por $A \ominus B$.

- **Erosão**:

  Para realizar a erosão numa imagem binária, o pixel central do elemento estruturante é colocado sobre cada pixel do *foreground* (valor 1). Se algum dos pixels vizinhos for um elemento de *background* (valor 0), o pixel central se torna *background*. A erosão de uma imagem **A** por um elemento estruturante **B** é denotada por $A \oplus B$.

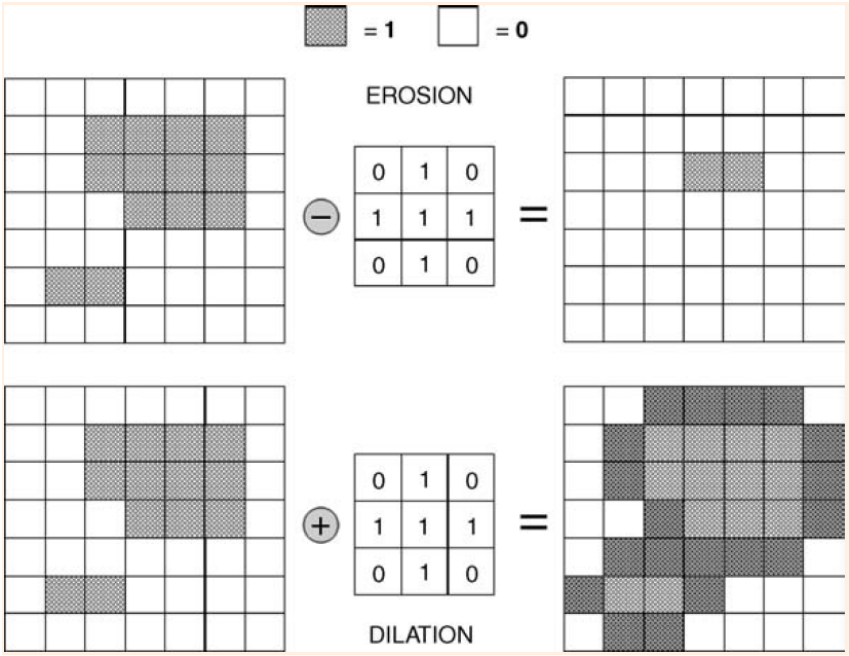


Figura 2.7: Exemplo visual das operações de erosão e dilatação. (Solomon and Breckon, 2011)

2.3 Vision Transformers (ViT)

O *Transformers* é uma arquitetura de aprendizagem profunda que ganhou notoriedade inicialmente em tarefas de Processamento de Linguagem Natural (NLP) (Vaswani et al., 2017), superando modelos do estado-da-arte. Os Transformers utilizam o mecanismo de atenção, que

mede os relacionamentos entre pares de tokens de entrada - entre palavras, no caso de texto. Tal cálculo da relação de cada palavra em relação a todas as outras tem um custo computacional quadrático.

No caso de imagens, como a unidade básica é o pixel, fazer a mesma operação a nível de pixels teria um custo ainda mais elevado, já que uma imagem contém por si só uma dimensão quadrática. Por exemplo: Em uma imagem de tamanho 512x512, o custo da operação de atenção a nível de pixel seria $((512)^2)^2$, totalizando mais de 68 bilhões de operações - um número impraticável mesmo com os computadores dos dias de hoje. Para resolver este problema, Dosovitskiy et al. (2020) propôs o modelo *Vision Transformers* (ViT), para classificação de imagens, que utiliza um mecanismo de atenção global, porém em *patches* da imagem. Como ilustrado na figura 2.8, o transformer recebe como entrada um vetor unidimensional com os tokens, contendo *patch embeddings* e *embeddings* posicionais (seção 2.3.1).

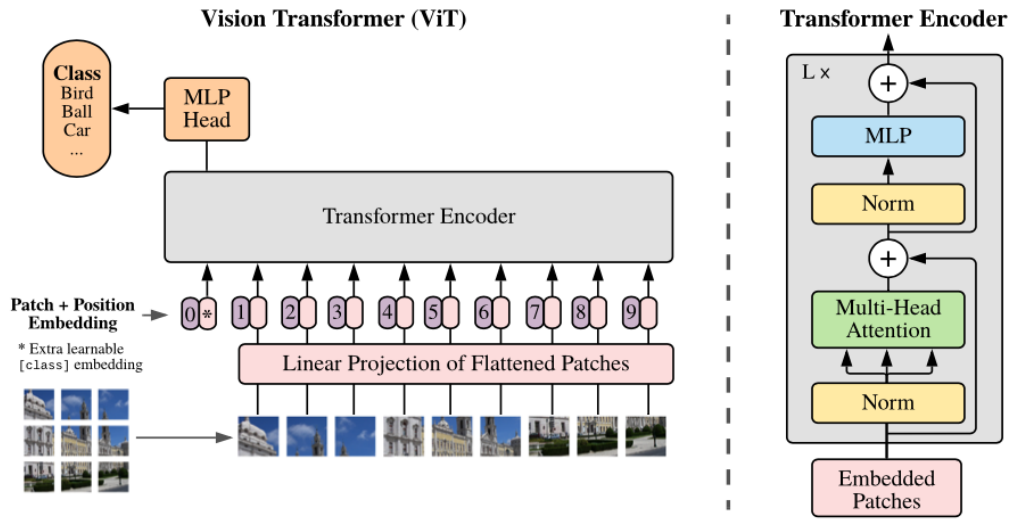


Figura 2.8: Visão global da estrutura do modelo ViT: Imagem é dividida em *patches* e passada como entrada junto com uma lista de suas posições (Dosovitskiy et al., 2020).

2.3.1 Patch Embeddings e Embeddings Posicionais

Para uma uma imagem de $(x) \in \mathbb{R}^{H \times W \times C}$ entrada e tamanho de patch p, são criados N patches de imagem, denotados $(x)_p \in \mathbb{R}^{N \times (P^2.C)}$, onde (H,W) é a resolução da imagem original, C é o número de canais, (P,P) é a resolução de cada patch e $N = HW/P^2$ é o número resultante de patches. Por fim, para gerar os *patch embeddings* é realizado um mapeamento dos $P^2.C$ *patches* da imagem para um vetor com D dimensões com projeções lineares treináveis de baixa dimensionalidade.

Já os *embeddings* posicionais contém informações relativas à posição de cada patch da imagem num vetor unidimencional. Numa arquitetura híbrida, a sequência de entrada pode ser

formada por mapas de características obtidos pela CNN. Nesse modelo, os patches são referentes ao mapa de características, e não a partes da imagem original.

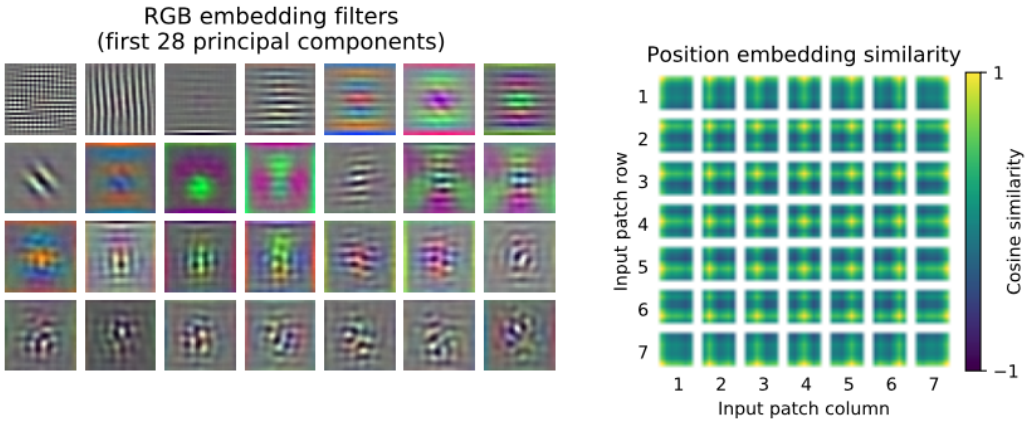


Figura 2.9: Filtros aprendidos aprendidos pela rede (similar aos filtros de uma CNN) que são usado para gerar os *patch embeddings*, (à esquerda) e os *embeddings* posicionais aprendidos pelo transformer (à direita), mostrando que *patches* próximos possuem mais similaridade (Dosovitskiy et al., 2020).

A camada de *encoder* do transformer é formada por camadas alternadas de dois blocos, *multihead self-attention* (também chamadas simplesmente de *heads*) e MLP, que são simplesmente camadas de transformações lineares. Não existe uma camada de *decoder*. A última camada do transformer é chamada *MLP head*, que também é uma simples camada de transformação linear para realizar a classificação.

O modelo ViT é pré-treinado com datasets com enorme quantidade de imagens, e então refinado para modelos mais específicos. A única modificação necessária é excluir a camada final de predição (*MLP head*) e adicionar uma nova camada *feed-foward* com o número desejado de classes.

O mecanismo de atenção é uma importante ferramenta para que o modelo foque em partes específicas da entrada (das imagens, no caso do ViT) para fazer sua predição. O mecanismo de *self-attention* (auto-atenção) usa este princípio para descobrir regiões de similaridade através dos pesos de atenção (*attention weights*). Este mecanismo permite que o ViT integre informações sobre toda a imagem, mesmo utilizando um número pequeno de camadas, através dos pesos de atenção.

2.4 DEtection TRansformers (DETR)

Seguindo o exemplo do ViT, pesquisadores do Facebook Research foram mais além e propuseram em 2020 o modelo DETR, o primeiro modelo de detecção de objetos utilizando transformers (Carion et al., 2020). A metodologia proposta considera a detecção de objetos como

um problema de predição direta de conjuntos. O DETR é capaz de predizer todos os objetos paralelamente, e utiliza uma função de perda que realiza a correspondência bipartida (*bipartite matching*) entre os objetos preditos e os objetos de *ground-truth*, baseado no método Húngaro (Kuhn, 1955).

O modelo DETR infere um número fixo N de predições por imagem a cada vez que passa pelo bloco de *decoder*, onde N é definido como sendo um número significantemente maior do que o número típico de objetos na imagem. A função de perda produz a correspondência bipartida ótima entre as predições e os objetos de *ground-truth*, o que significa que apenas uma caixa correspondente ao objeto é considerada, com perda mínima.

2.4.1 Arquitetura

O DETR possui uma arquitetura simples, que contém três elementos principais: Um *backbone* CNN, um bloco transformer encoder-decoder, e uma rede *feed-forward* (FFN), que pode ser visualizado na figura 2.10. O *backbone* CNN é usado para extrair representações reduzidas das características das imagens. Em sequência encoding posicionais são adicionados para produzir um token que é passado como entrada para o transformer. O modelo conta ainda com blocos encoder-decoder, que utilizam *embeddings* posicionais (*object queries*) para observar partes específicas da imagem e predizer os objetos (com boxes e classe) ou "no object". Por fim, uma rede *feed-forward* (FFN) é responsável pela predição final do modelo. Abaixo cada um desses elementos é explicado em mais detalhes.

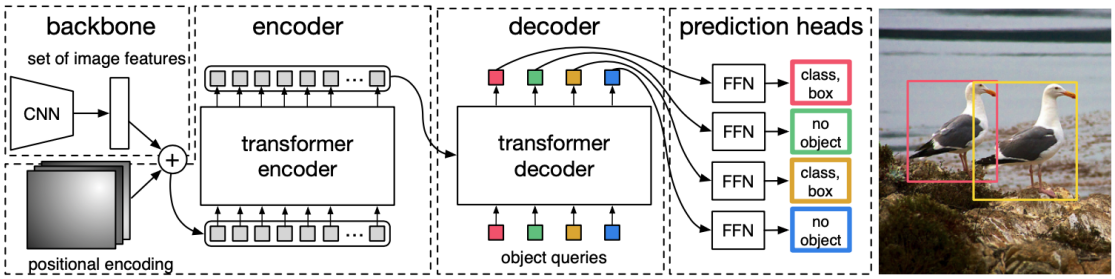


Figura 2.10: Arquitetura geral do DETR (Carion et al., 2020).

- **Backbone**:

  O DETR utiliza uma CNN convencional (ResNet) para transformar a imagem original $x_{img} \in \mathbb{R}^{3\times H_0\times W_0}$, com 3 canais RGB, em um mapa de ativação de baixa resolução $f \in \mathbb{R}^{C\times H\times W}$, com $C = 2048$ e $H, W = \frac{H_0}{32}, \frac{W_0}{32}$.

- **Transformer Encoder**:

  Uma convolução 1×1 reduz as dimensões do mapa f de C para um número menor de dimensões d, criando um mapa de características $z_0 \in \mathbb{R}^{d\times H\times W}$. Em seguida o mapa é re-

duzido (*flatten*) a apenas uma dimensão, e *embeddings* posicionais, aprendidos de forma semelhante ao que ocorre no modelo ViT (seção 2.3), são adicionados aos tokens que são passados como entrada para o transformer. O encoder possui uma série de camadas, compostas por blocos *multi-head self-attention* e FFNs, com *embeddings* posicionais sendo adicionados a cada uma das camadas de atenção. O número de blocos de encoder utilizado é importante para atingir a *atenção global* na imagem. Através dos blocos de *self-attention*, o modelo calcula a similaridade dos pontos em relação aos seus vizinhos. O resultado são pesos de atenção maiores para pontos com similaridade, por exemplo, pertencentes a um mesmo objeto, como é possível ver na figura 2.11.

- **Transformer Decoder**:

  O decoder também funciona similar ao que ocorre no transformer original, e utiliza *multi-headed self-attention* e FFNs para predizer N objetos. A diferença para os modelos anteriores é que os N objetos são preditos de maneira paralela. No bloco de decoder, N embeddings posicionais são aprendidos pelo modelo (chamados *"object queries"*) e passados como entrada. Esses embeddings são adicionados em cada uma das camadas de atenção do decoder, similar ao que ocorre no encoder. Esses embeddings são transformados em saídas e decodificados para as coordenadas das caixas de marcação e classe do objeto através de uma FFN, resultando em N predições.

- **Feed Foward Networks**:

  A camada final é composta por uma rede *perceptron* de 3 camadas, com uma ReLU como função de ativação e dimensão d. A FFN prediz a coordenada central, altura e largura da caixa de marcação. Como são realizadas N predições, com um número maior do que o número de objetos, uma classe adicional $(\emptyset)$ é usada para preencher os espaços em excesso, similar ao objeto *background* utilizado em outros modelos.

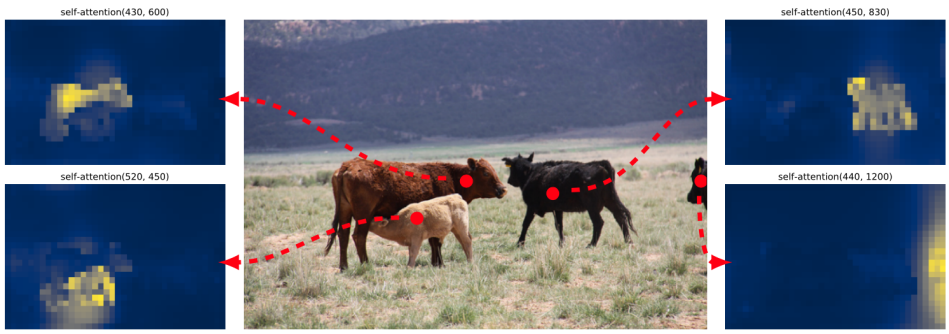


Figura 2.11: Visualização dos mapas de atenção da última camada do encoder, onde já é possível separar instâncias dos objetos (Carion et al., 2020).

2.5 Métricas de Avaliação de Desempenho

Em detecção de objetos, diversas métricas são utilizadas para avaliar o desempenho dos modelos. Em geral, são adotados padrões utilizados em competições, como COCO e PASCAL VOC. Nesta são descritos alguns conceitos fundamentais e métricas utilizadas na detecção de objetos, sumarizado por Padilla et al. (2020).

2.5.1 Conceitos Fundamentais

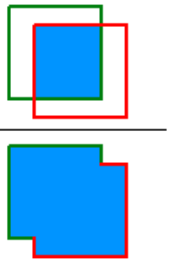
- **Nível de confiança (Confidence score)**: Cada previsão do modelo é associada a uma pontuação que indica a confiança com a qual a previsão foi feita. Na detecção de objetos, mais especificamente, indica a confiança de que a caixa de marcação (*anchor box*) contém um objeto.

- **IoU (Interseção sobre união)**: Representa o nível de sobreposição entre a caixa predita e o valor verdadeiro esperado (*ground truth*). Quando melhor o encaixe entre as caixa, mais próximo o valor do IoU estará de 1, e quanto mais desajustado, mais próximo de 0. O cálculo é feito dividindo a área de interseção pela área de união entre uma caixa de marcação predita (B_p) e uma caixa de *ground-truth* (B_{gt}).

$$IoU = \frac{area(B_p \cup B_gt)}{area(B_p \cap B_{gt})}$$

Uma representação visual do IoU é mostrado na figura abaixo.

$$IOU = \frac{\text{area of overlap}}{\text{area of union}} =$$



Tipicamente uma classificação pode ser um verdadeiro positivo (TP), um falso positivo (FP), um falso negativo (FN) ou um verdadeiro negativo (TN). Em detecção de objetos, essas categorias são definidas da seguinte forma:

- **Verdadeiro positivo (TP)**: Uma detecção correta. Detecção com $IoU \geq threshold$.

- **Falso positivo (FP)**: Uma deteção errada. Deteção com $IoU < threshold$.

- **Falso negativo (FN)** *Ground truth* não detectado.

- **Verdadeiro negativo (TN)** Não se aplica à detecção de objetos.

2.5.2 Métricas de Detecção de Objetos

Algumas das métricas mais populares utilizadas na detecção de objetos estão descritas abaixo.

- **Precisão**: Precisão é a capacidade de um modelo para evitar rotular amostras negativas como positivas. O cálculo da precisão é feito dividindo o número de verdadeiros positivos (TP) pela soma entre verdadeiros positivos (TP) e falsos positivos (FP).

$$precision = \frac{TP}{TP+FP} = \frac{TP}{all\ detections}$$

- **Recall**: Recall é a capacidade de um modelo para detectar todas as amostras positivas. O cálculo é feito dividindo o número de verdadeiros positivos (TP) pela soma entre verdadeiros positivos (TP) e falsos negativos (FN).

$$recall = \frac{TP}{TP+FN} = \frac{TP}{all\ ground\text{-}truths}$$

- **F1-score** Também conhecido como *F-score* ou *f-measure*, o F1-score pode ser definido como sendo a média ponderada entre a precisão e o recall, com valores variando em 0 e 1. O cálculo do F1-score é:

$$F1 = 2 \times \frac{precision * recall}{precision + recall}$$

- **Curva de precisão-recall**:

  A curva de precisão-recall plota a relação entre precisão e recall à medida que o *threshold* é alterado. Para cada valor de threshold, existe um par de precisão e recall, com o recall estando no eixo x e a precisão no eixo y. Dessa forma é possível desenhar a curva de precisão-recall, que indica a associação entre as duas métricas.

- **Precisão média (AP)** A precisão média (AP) é um valor numérico que ajuda a comparar diferentes modelos de detecção de objetos. A definição geral de precisão média (*average precision*) é como sendo a área abaixo da curva precisão-recall. Para lidar com o efeito zig-zag da curva, primeiro é realizado a interpolação da precisão utilizando diferentes valores de recall. O método mais popular é utilizar 11 pontos para a interpolação: $[0, 0.1, 0.2, ..., 1]$.

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, ..., 1\}} p_{interp}(r)$$

  O método de avaliação COCO usa um método de interpolação de 101 pontos para cálculo de AP junto com a média de mais de dez limites de IoU. AP@[.5:.95] corresponde a

uma média de AP para IoU de 0.5 a 0.95 com intervalos de 0.05. O COCO utiliza ainda as métricas AP@0.50 e AP@0.75, que são os mAPs para valores de IoU de 0.5 e 0.75, respectivamente.

3

Materiais e Métodos

Esta seção visa descrever em detalhes os materiais e métodos utilizados no desenvolvimento deste projeto. A figura 3.1 apresenta uma visão geral da metodologia proposta. A base de imagens utilizada no treinamento do modelo foi gerada a partir do LUNA16 (seção 3.1), que usa imagens 3D de TCs de tórax. A seção 3.2 apresenta o pré-processamento realizado para obter a base de imagens para o treinamento do modelo. Em seguida, a seção 3.3 mostra detalhes da configuração e treinamento do modelo utilizado.

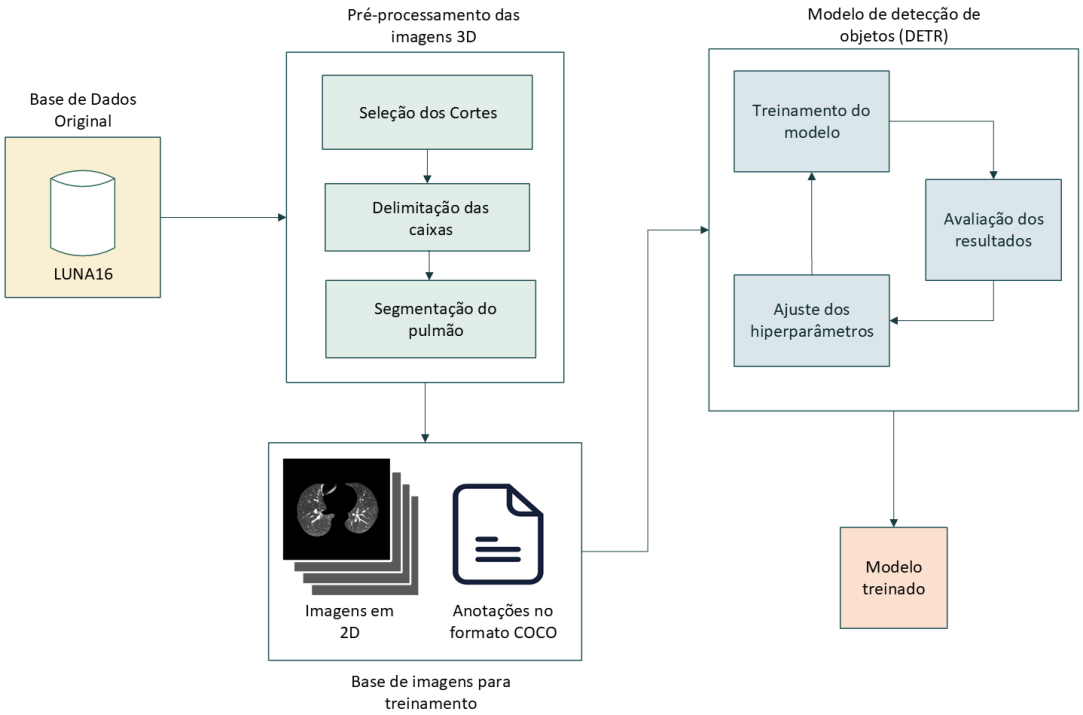


Figura 3.1: Esquemático do Projeto.

3.1 Base de Dados

A base de dados escolhida para esse trabalho foi a LUNA16 (*LUng Nodule Analysis 2016*) (Setio et al., 2017), uma base pública de tomografias computadorizadas (TCs) da região torácica de pacientes identificados com nódulos pulmonares desenvolvida em 2016 para uma competição com foco no desenvolvimento de algoritmos para detecção de nódulos pulmonares, sendo uma extensão de outra base pública, a LIDC-IDRI (*Lung Medical Imaging Database Consortium*) (Armato III et al., 2011), mas com algumas melhorias pontuais direcionadas à detecção de nódulos.

O LIDC-IDRI é uma base composta por 1.018 exames da região torácica de indivíduos identificados com nódulos pulmonares. A anotação dos nódulos foi realizada por 4 radiologistas experientes, os quais fizeram marcações manuais dos nódulos. Os autores do LUNA16 utilizaram a base de imagens do LIDC-IDRI e realizaram algumas filtragens com o intuito de uniformizar as TCs. Foram descartados todos os exames do LIDC-IDRI cuja espessura do corte fosse superior a 2.5mm, com espessura inconsistente ou ainda com cortes faltantes. Após esse processo, a base do LUNA16 obteve um total de 888 exames, com mais de 159 mil cortes. Na base do LUNA16 foram considerados ainda apenas nódulos $\geq$ 3mm de diâmetro. Na base do LIDC-IDR os nódulos foram anotados por quatro radiologistas experientes, sendo cada nódulo anotado por 1, 2, 3 ou 4 radiologistas. Os autores do LUNA16 decidiram considerar apenas nódulos anotados pela maioria, pelo menos 3 dos 4 radiologistas. Isso resultou num total de 1.186 nódulos, dos 2.290 nódulos iniciais.

As imagens do LUNA16 são disponibilizados no formato DICOM. Cada exame consiste num número variável de imagens (cortes), entre 100 e 400, no eixo axial, em escala de cinza, com resolução espacial de 512 x 512 *pixels* e resolução de contraste de 16 bits. As anotações a respeito dos nódulos estão disponíveis num arquivo *CSV*, contendo as coordenadas referentes ao centro do nódulo (X, Y e Z), raio do nódulo (em milímetros) e ID único do exame (*series UID*).

A base do LUNA16 está dividida em 10 partes compactadas, com tamanho total de 120 GB. Cada TC é identificada por um ID único e possui dois arquivos associados a ela, um arquivo MHD e outro arquivo RAW, sendo o primeiro responsável por armazenar os metadados do exame, e o segundo por armazenar os dados brutos referente à matriz 3D do exame.

Por estar em coordenadas anatômicas, foi necessário realizar uma conversão para nível de *voxel* para poder localizar os nódulos na imagem. O algoritmo descritivo utilizado para realizar a conversão, seguindo o que foi discutido no capítulo de fundamentação teórica, foi:

- O *offset* da origem é subtraído do valor das coordenadas;
- É realizada uma multiplicação de matrizes com a matriz de direção;
- É feito o redimensionamento do valor das coordenadas com o tamanho do *voxel*;

- Arredondamento das coordenadas para o inteiro mais próximo, para evitar números decimais;
- Por fim, a ordem das coordenadas é invertida, passando de (C,R,I) para (I,R,C).

3.1.1 Seleção das Imagens

Devido ao custo computacional em processar imagens 3D e pelo fato dos modelos com Transformer que utilizam detecção de objetos em objetos 3D ainda serem escassos na literatura, optamos por utilizar imagens 2D no projeto. Para a construção da nossa base de imagens 2D, considerando que o número de imagens com nódulos em relação ao total de imagens no exame é extremamente baixo, foi necessário selecionar quais os cortes serão utilizados. Para isso, foi necessário localizar quais cortes possuem informações sobre os nódulos.

Como todos os cortes de uma mesma TC compartilham as mesmas propriedades (origem, espaçamento, matriz de direção, diretórios, etc), foi criada uma classe para armazenar as informações referentes ao mesmo exame e agilizar o processamento das imagens. Cada objeto, referente a um exame, possui os seguinte *atributos*:

- ID único do exame;
- Caminho para os arquivos do exame no disco;
- Tamanho do voxel;
- Lista de todos os nódulos do exame, contendo a localização e o raio, já em pixels;

A classe conta ainda com um *método* para a obtenção do *array* com os dados do exame, para que os dados sejam obtidos dinamicamente e que não sobrecarreguem a memória.

Uma abordagem simples para salvar as imagens 2D com os nódulos seria utilizar o corte referente ao centro do nódulo, a qual temos a informação a partir da coordenada Z, e utilizando o raio para calcular a altura e largura da caixa. Isso, porém, resultaria em perda de informações, uma vez que o nódulo geralmente é visível nos cortes anteriores e posteriores ao corte central. Essa perda de cortes poderia inclusive atrapalhar o treinamento do modelo, pois regiões com informações referentes a nódulos que atingissem corte de outros nódulos estariam sem marcação.

Com o objetivo de maximizar a quantidade de imagens com informações sobre os nódulos a serem salvas e resolver o problema descrito acima, foi desenvolvido um algoritmo para aumentar a quantidade de cortes referente a cada nódulo considerando o raio do nódulo e a espessura dos cortes.

A partir da lista com a localização do centro dos nódulos de um dado exame, o algoritmo calcula a quantidade de cortes até a metade do raio de cada nódulo e os adiciona a uma lista. A restrição de considerar os cortes apenas até a metade do raio se dá pelo fato de que o raio da

área efetiva do nódulo se reduz bastante após esse ponto, além dos nódulos possuírem formas muito variadas. Desta forma, a restrição até a metade garante que parte do nódulo ainda seja visível, inclusive para nódulos pequenos.

Após gerar os cortes de interesse a partir do algoritmo descrito acima, a base passou de 1186 para um total de 4098 imagens com um ou mais nódulos, um aumento de quase 4 vezes.

3.2 Pré-processamento das Imagens

O principal objetivo do pré-processamento das imagens de TC é eliminar ruídos, artefatos e outras informações irrelevantes dos exames para tentar focar apenas nos órgãos que são objeto do estudo, de forma a tentar obter uma melhor qualidade da imagem final e aumentar a capacidade dos algoritmos de *deep learning* (DL) em detectar informações relevantes. Todo o pré-processamento deste trabalho foi realizado utilizando a linguagem de programação Python.

Para tratar o brilho e contraste da imagem e evidenciar a região do pulmão, a primeira estratégia foi utilizar os valores usados por Alakwaa et al. (2017) para largura de janela e nível, com 680 e -660, respectivamente. No entanto, foi observados que apesar de tais valores evidenciarem os nódulos pulmonares, algumas texturas acabaram se perdendo. Por esse motivo, os valores de janela e nível foram alterados para 1200 e -400, respectivamente, ficando próximos dos valores utilizados por Lima et al. (2019). Um exemplo de corte com a caixa de *ground truth* do nódulo pode ser visualizado na figura 3.2.

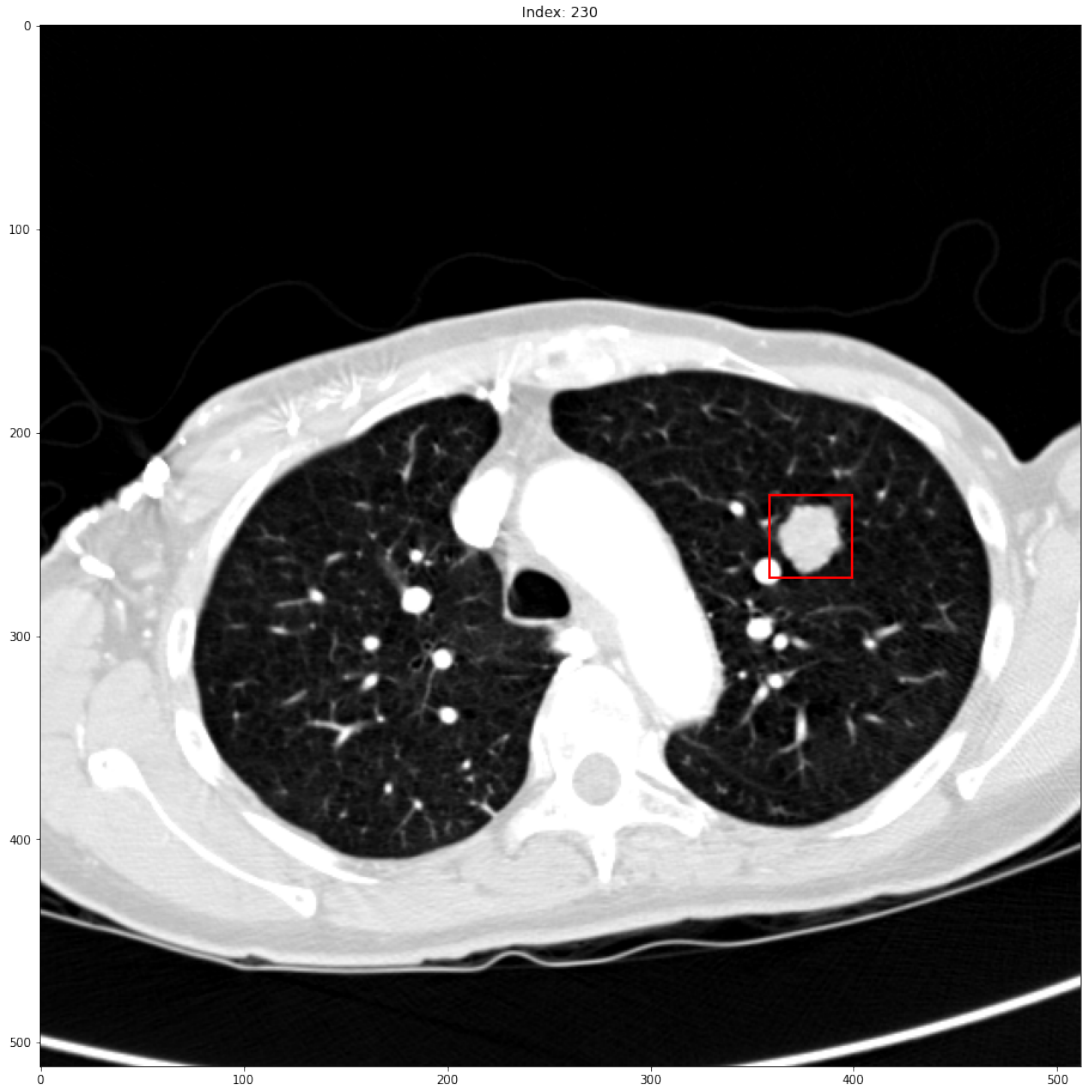


Figura 3.2: Região torácica com nódulo pulmonar em evidência. Fonte: elaborado pelo autor.

3.2.1 Segmentação da Região do Pulmão

O objetivo da segmentação neste trabalho foi isolar a região interna do pulmão, visando descartar fatores de confusão, localizados fora do pulmão, de forma a permitir que durante o treinamento a rede foque na região de interesse onde os nódulos estão localizados.

Este trabalho utilizou técnicas de segmentação com operações morfológicas, com o auxílio das bibliotecas *Skimage* (Van der Walt et al., 2014) e *Scipy* (Virtanen et al., 2020). As operações aplicadas para a obtenção da região de interesse (ROI) estão listadas abaixo, junto com a visualização gráfica de cada uma das operações numa imagem de CT (figura 3.3).

1. Binarização da imagem: Pixels abaixo de -400 HU são convertidos para zero; e acima disso são convertidos para 1;

2. Limpeza da borda: Remove os objetos conectadas à borda da imagem;

3. Rotulação de regiões conectadas na imagem;
4. Manutenção apenas das duas maiores regiões da imagem;
5. Operação de erosão com disco de raio 2;
6. Operação de fechamento com raio 10;
7. Utilização do operador cruzado de Roberts e preenchimento de pequenos buracos dentro da região do pulmão;
8. Superimposição da máscara binária na imagem de entrada.

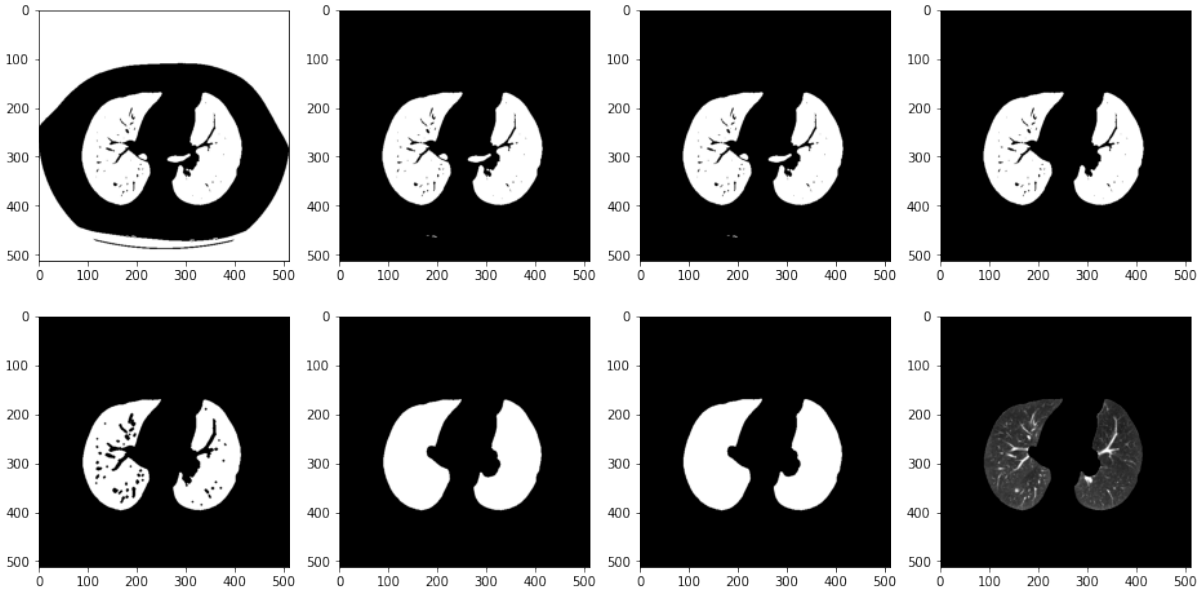


Figura 3.3: Etapas da segmentação da região do pulmão. Fonte: elaborado pelo autor.

A base do LUNA16 oferece as segmentações da região interna do pulmão, realizadas também de forma automatizada por um algoritmo. No entanto, diante da necessidade de que teríamos que realizar a segmentação de novas imagens para realizar a detecção de novas entradas no sistema, optamos por implementar a etapa de segmentação.

Apesar da segmentação realizada pelo nosso algoritmo funcionar bem na maioria das imagens, foram detectadas algumas imagens onde a segmentação acabou perdendo informações importantes, como nódulos anexados à região externa do pulmão. Por essa razão, decidimos adicionar mais uma etapa ao nosso *pipeline* e gerar duas bases de imagens em nosso testes, com e sem a segmentação da região intrapulmonar. O objetivo dessa etapa extra é realizar um comparativo entre os resultados obtidos por ambas as bases e decidir se a realização da segmentação realmente promove um ganho na detecção de nódulos.

3.3 Configuração do Modelo

O modelo escolhido para ser utilizado nesse projeto para para a detecção de nódulos pulmonares foi o DETR (Carion et al., 2020) (seção 2.4). Este modelo utiliza a arquitetura Transformer junto com uma CNN, e simplifica o pipeline de detecção ou não requerer camadas personalizadas, de forma que pode ser reproduzido facilmente em qualquer estrutura que contenha CNN padrão e classes Transformer.

A escolha deste modelo se deu pelo fato do DETR ter sido o primeiro modelo a integrar um bloco *Transformer* no pipeline de detecção de objetos, inspirando diversos outros trabalhos (Zhu et al., 2020; Liu et al., 2021), de disponibilizar os códigos da implementação de forma aberta num repositório online e por possuir extensa discussão em fóruns, o que facilita o entendimento do modelo e sua adaptação a diferentes cenários.

O modelo DETR utiliza o padrão *Common Objects in Context* (COCO) (Lin et al., 2014) para a base de imagens. No entanto, por ser um padrão relativamente complexo, cujas anotações são salvas num único arquivo JSON, e pelo fato da nossa base ter poucas informações a serem salvas, optamos por salvar as anotações utilizando o padrão do YOLOv5 (Ultralytics), que utiliza as coordenadas centrais (X,Y), junto com a altura e largura da caixa de marcação. As anotações são salvas num arquivo de texto padrão.

Antes de salvar as imagens, foi feita a divisão da base em treino e teste, sendo 80% para treino e 20% para teste. Para evitar que imagens de um mesmo exame se separassem e acabassem fazendo parte tanto da base de treino quanto da base de testes, o que resultaria em *data leakage*, a divisão treino/teste foi realizado a nível de TC. No total a base de treino contém 3255 imagens, enquanto a de testes contém 832. As imagens foram salvas na resolução original (512 x 512) com 8 bits. Após geradas as bases de treino e teste, ambas com e sem segmentação, foi utilizado um *script* disponibilizado publicamente num repositório do GitHub[1] para realizar a conversão do padrão YOLOv5 para COCO. A conversão gerou os arquivos JSON desejados.

3.3.1 Treinamento do Modelo

O treinamento do modelo foi realizada em um servidor remoto, com o sistema operacional Linux (Ubuntu 20.04 LTS) e placa gráfica NVIDIA Tesla T4, com 16GB de memória. O modelo foi configurado num ambiente virtual, onde foi instaladas todas as dependências, como especificado no repositório do projeto.

O procedimento inicial para o treinamento consistiu em criar um *fork* do repositório DETR no GitHub[2] para realizar as modificações necessárias e adaptar o modelo para a detecção de nódulos pulmonares. O número de camadas foi mantido como no modelo original. Os valores dos parâmetros relativos ao módulo do Transformer são mostrados na tabela 3.3.1.

[1] https://github.com/Taeyoung96/Yolo-to-COCO-format-converter/
[2] https://github.com/nilsonsales/detr_luna

Variável	Descrição do parâmetro	Valor
enc_layers	Número de camadas *encoder*	6
dec_layers	Número de camadas *decoder*	6
hidden_dim	Tamanho dos *embeddings*	256
num_classes	Número de classes	1
num_queries	Número de slots para detecção	10
dropout	Dropout aplicado no transformer	10^{-1}
nheads	Número de *attention heads* usadas para atenção	8

A primeira modificação consistiu em mudar o número de *object queries*, que é a quantidade de objetos preditos por imagem. O modelo original é configurado para detectar 100 objetos por imagem, porém como o número de nódulos numa mesma imagem é geralmente muito menor, esse número foi reduzido de maneira empírica para 10 objetos. Outros hiperparâmetros do modelo foram reajustados a medida que o treinamento foi realizado.

Uma das transformações realizadas pelo modelo antes do treinamento para uma melhor generalização consiste em redimensionar as imagens de maneira aleatória, seguindo uma lista de tamanhos. Inicialmente o tamanho máximo para *upscaling* foi definido como 512 x 512, para evitar adicionar informações não existentes à imagem. Por conta do tamanho reduzido, foi possível utilizar um *batch size* de 16. O modelo foi então treinado por 200 épocas, porém, após ser constatado resultados insatisfatórios especificamente para nódulos pequenos, optamos por aumentar a resolução máxima para 800 x 800. O *batch size* teve de ser reduzido para 8 por conta da limitação de memória. O valor do *learning drop* também foi aumentado de 200 para 1000 por conta da quantidade relativamente pequena de imagens na nossa base. Após estas modificações, o modelo demonstrou uma leve melhora na detecção de nódulos menores.

É importante ressaltar que o DETR possui uma limitação quanto a detectar objetos pequenos. Uma alternativa sugerida pelos autores é a troca do *backbone* padrão, ResNet50, por um *backbone* modificado, chamado DETR-DC5, que é uma ResNet50 com uma modificação no último bloco convolucional, onde o bloco de *stride* é substituído por uma camada de dilatação, aumentando a resolução da rede. No entanto isso resulta também num aumento considerável de tempo de treinamento e memória utilizada. Este formato ficou inviável tanto pelo tempo necessário para treinamento quando pela quantidade de memória exigida. Mesmo utilizando *batch size* com tamanho 2, a memória exigida fica próxima aos 16 GB presentes na máquina, de forma que o treinamento poderia ser interrompido a qualquer momento por falta de memória. Por esse motivo, foi mantido o uso da ResNet-50.

Seguindo com o treinamento e utilizando os hiperparâmetros identificados, pudemos treinar o modelo para a detecção utilizando as duas bases: com a máscara de segmentação e sem a máscara. Cada modelo foi treinado por 400 épocas, com cada época levando em média 10 minutos para ser finalizada, totalizando aproximadamente 66 horas de treinamento para cada dataset.

4

Resultados e Discussões

Neste capítulo serão apresentados os resultados obtidos através da metodologia apresentada no capítulo anterior para a detecção de nódulos pulmonares em imagens de TC.

4.1 Segmentação

Um exemplo comparativo entre a segmentação da região interna do pulmão obtida pelo nosso algoritmo e a disponibilizada pela base do LUNA16 é mostrado na figura 4.1. Nosso algoritmo conseguiu resultados semelhantes às segmentações presentes no LUNA na maioria das imagens. Porém, como mencionado anteriormente, alguns nódulos acabaram sendo perdidos durante a segmentação em imagens com baixo contraste ou quando os nódulos estão anexados à região externa do pulmão. Por este motivo os dois datasets, com e sem a segmentação, foram utilizados na avaliação dos resultados.

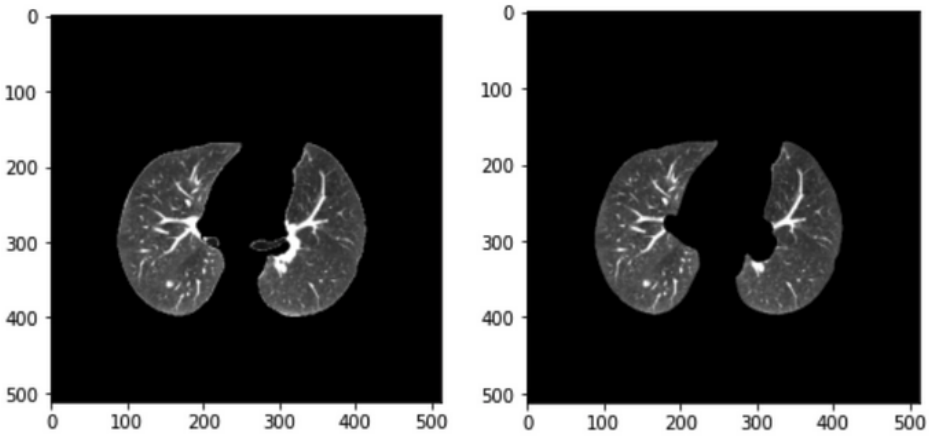


Figura 4.1: Segmentação do LUNA16 (à esquerda) e a segmentação obtida pelo nosso algoritmo (à direita).

4.2 Resultados do Treinamento

A figura 4.2 apresenta os resultados para erro de classe, *mAP0.5* e *mAP0.5:0.95* obtidas após o treinamento durante 400 épocas para ambos os datasets. Já na figura 4.3 temos as curvas de precisão-*recall*.

O modelo treinado com as imagens de TC sem a máscara de segmentação intrapulmonar (curva laranja) apresentou resultados levemente superiores ao modelo treinado com o dataset com a máscara de segmentação da região intrapulmonar (curva azul). Os valores de erro de classe mostrados na imagem não são exatos pelo fato de usarmos menos predições por imagem (10) do que o modelo original (100). Porém as curvas são úteis para observarmos que o erro de classe está diminuindo ao longo do treinamento. A segunda métrica mostrada é a precisão média (mAP, ou mAP@50), e se refere à precisão média utilizando IoU de 0.5.

O modelo sem a segmentação obteve mAP@50=86.6, recall=0.899 e f1=0.882, enquanto o modelo com segmentação conseguiu mAP@50=80.7, recall=0.849 e f1=0.828. Observamos ainda que a curva de precisão média mAP0.5:0.95, que inclui caixas mais ajustadas, ainda apresentam uma leve tendência de crescimento até as últimas épocas registradas, o que indica que com mais algumas épocas de treinamento, os *bounding boxes* poderiam ficar ainda mais justos.

4.3 Inferências e Discussões

Para realizar as inferências nas imagens, o DETR nos permite carregar o modelo genérico a partir do repositório do TorchHub[1] e atualizá-lo com os pesos obtidos no treinamento. Cada inferência retorna dez *queries* com dez predições, porém mantemos apenas as predições com confiança acima de 0.8. Alguns exemplos de inferências para ambos os datasets são mostrados nas figuras 4.4 e 4.5. As caixas de marcação azuis representam a inferência do modelo, enquanto as caixas vermelhas mostram os valores reais (*ground truth*). Os valores em amarelo são os valores de saída do modelo após aplicada uma função de ativação (*softmax*), e representam a confiança do modelo de que o objeto marcado é um nódulo pulmonar.

As detecções obtidas pelo modelo apresentaram bons resultados em localização e tamanho dos *bounding boxes*, se aproximando bastante dos valores marcados pelos radiologistas. O modelo carregado ocupa 1,9 GB de memória na GPU, e cada inferência leva 0,6 segundos para ser realizada (1,4 segundos para realizar a inferência e plotar o resultado). Dessa forma, entendemos que o modelo poderia ser utilizado em um computador padrão com uma GPU simples.

O fato do modelo sem a máscara de segmentação ter apresentado resultados melhores em comparação com o outro modelo pode ser explicado pelo fato de alguns nódulos serem perdidos

[1] https://pytorch.org/docs/stable/hub.html

durante o processo de segmentação, o que interfere na detecção desses nódulos pelo modelo. Imaginamos que uma outra estratégia de segmentação mais precisa possa corrigir este problema e igualar ou até superar o modelo sem a máscara de segmentação.

Modelos de detecção de objetos utilizando a arquitetura ViT são relativamente novos, e apresentam alguns desafios para a comunidade. Na área de imagens médicas, esse desafio é especialmente ampliado pelo fato de que imagens médicas, principalmente com anotações, são ainda mais escassas do que em outras áreas. Apesar disso, os modelos de ViT apresentam importantes diferenças em sua arquitetura com relação aos modelos CNN, como discutido anteriormente, e ganharam uma atenção extra da comunidade científica. Neste trabalho pudemos ver que um modelo de detecção de nódulos pulmonares em imagens de TC utilizando uma arquitetura de *Transformers* conseguiu alcançar resultados satisfatórios de mAP, score e f1, e os *bounding boxes* observados nas inferências apresentaram encaixe satisfatório com relação aos valores esperados.

Os modelos de *Transformers*, incluindo o DETR, requerem um tempo de treinamento consideravelmente elevado para alcançarem os melhores resultados, o que dificulta testes com mais opções de hiperparâmetros. Isso pode ser melhor explorado em trabalhos futuros. Os modelos *Transformers* também têm como particularidade o fato de precisarem de um número elevado de dados para atingir os melhores resultados. Por esse motivo, a base de dados deste trabalho também poderia ser estendida, para incluir outras bases de imagens. Também, como dito anteriormente, o modelo DETR tem como ponto fraco a detecção de objetos pequenos, como é o caso de grande parte dos nódulos. Outros trabalhos (Zhu et al., 2020) mais recentes se propõe a resolver este problema, e podem ser utilizados para uma futura versão do *framework*.

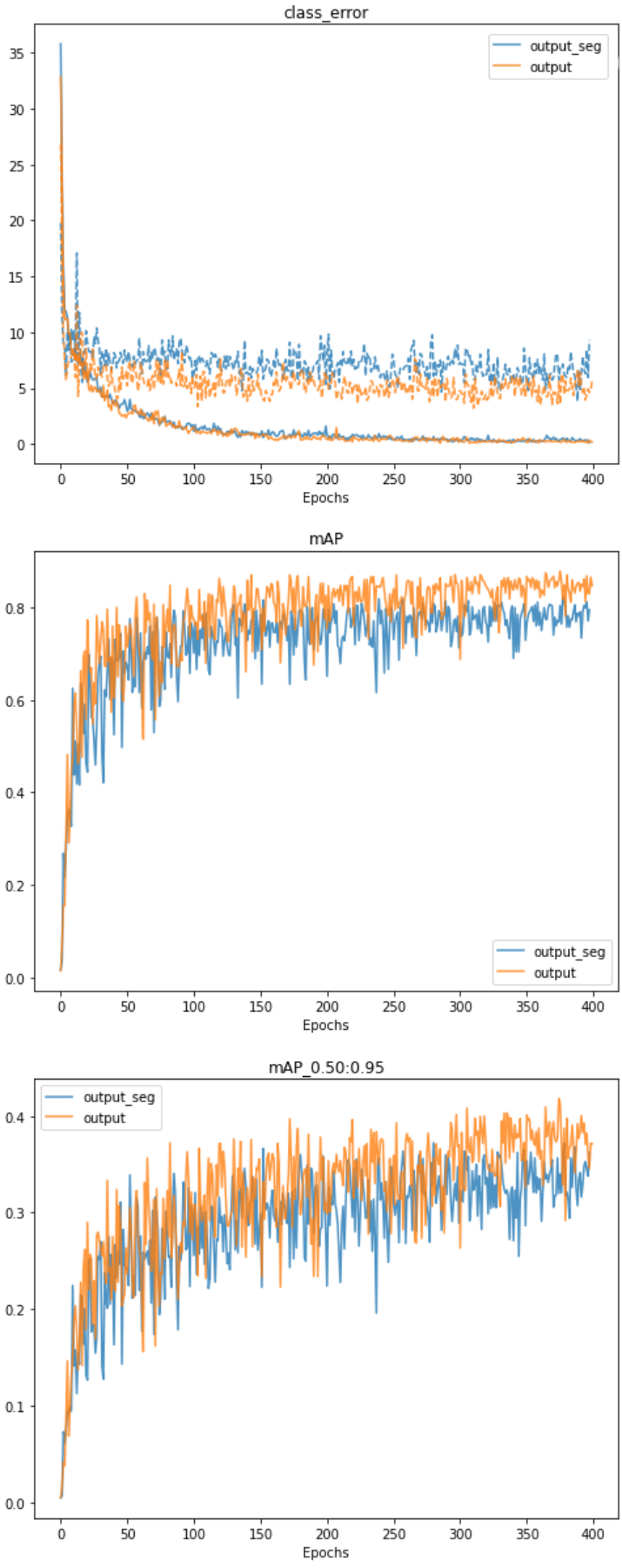


Figura 4.2: Gráficos dos logs do treinamento do modelo.

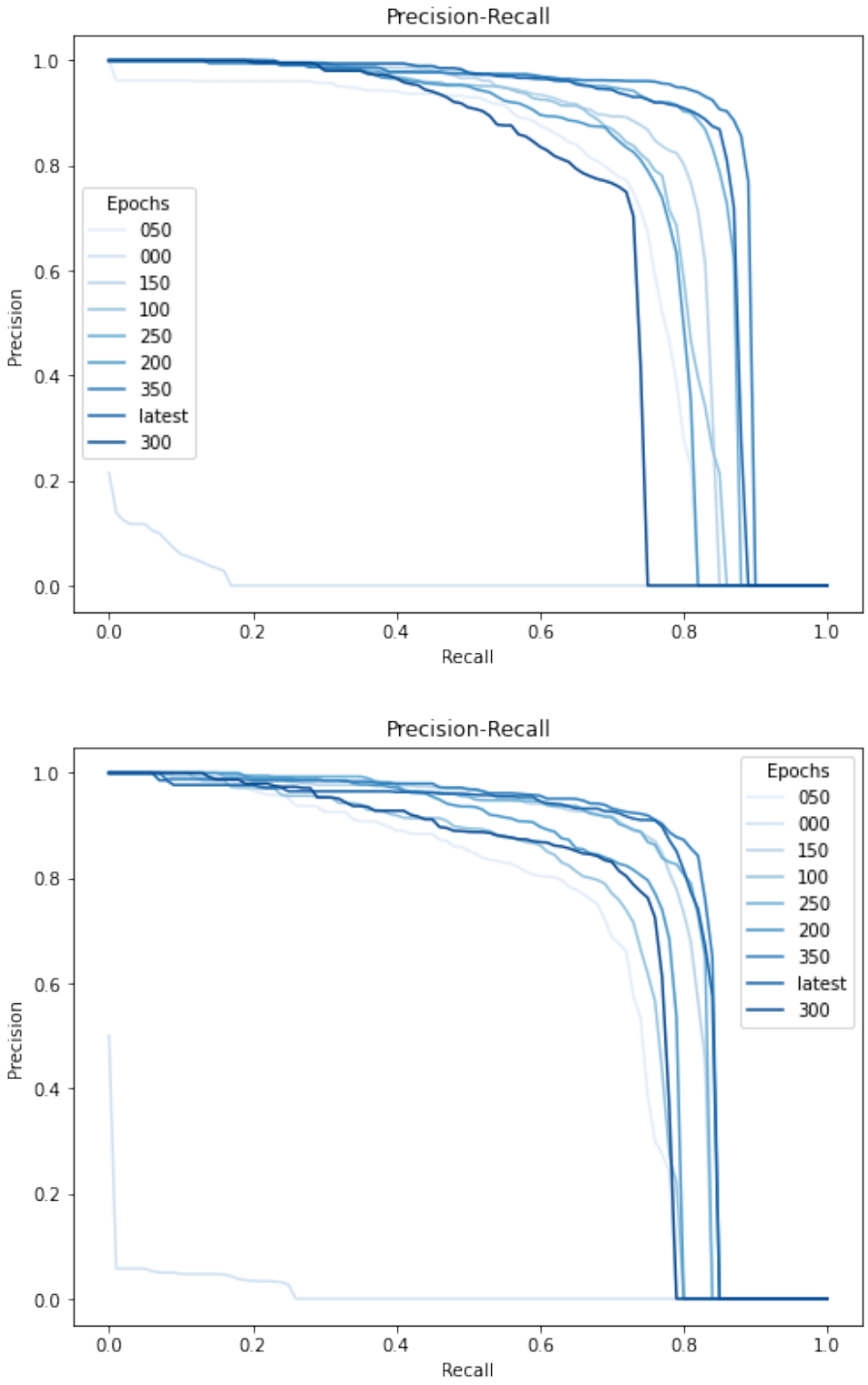


Figura 4.3: Curvas de precisão-recall ao longo do treinamento: Modelo treinado com imagens sem segmentação em cima e com segmentação embaixo.

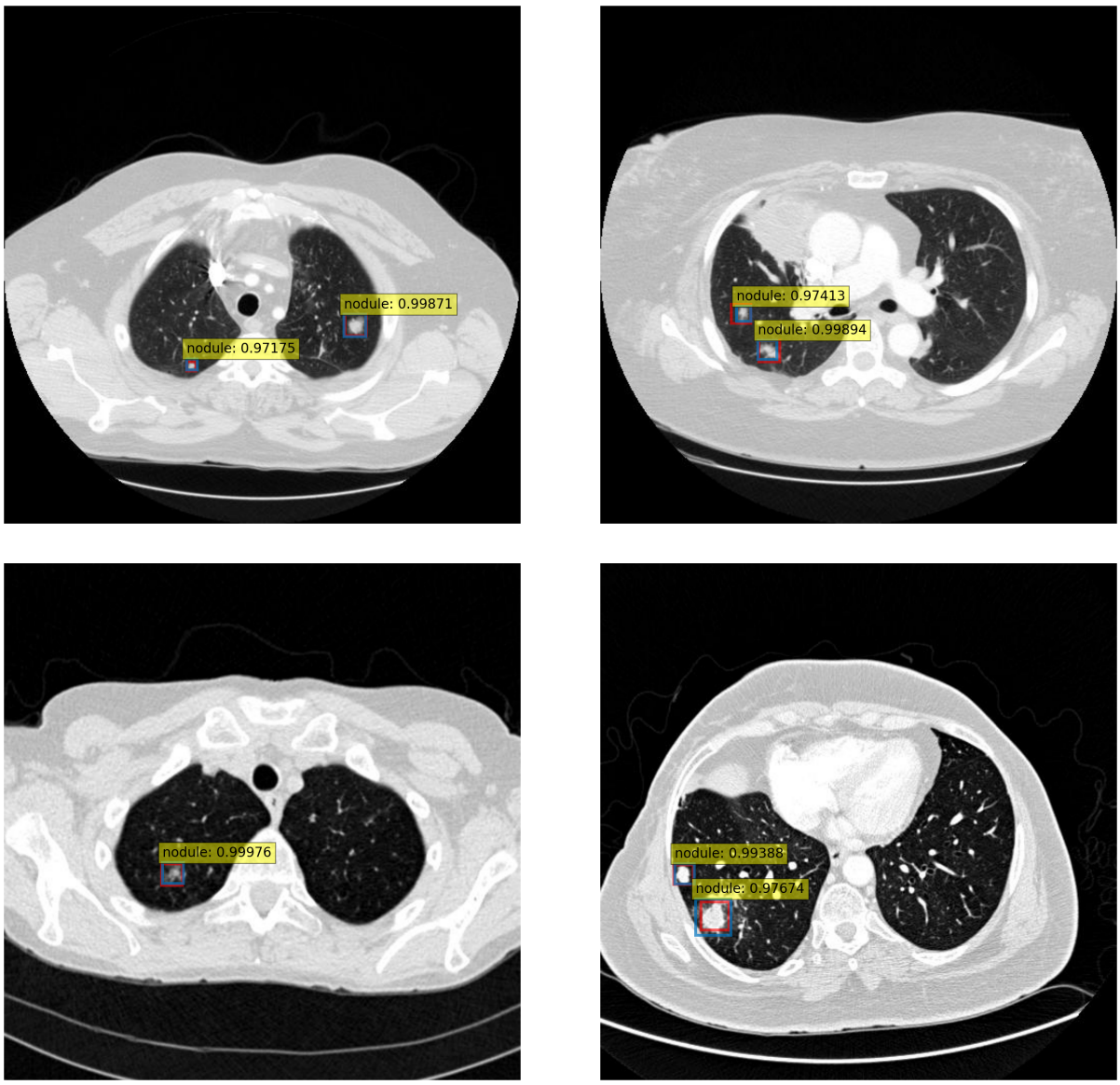


Figura 4.4: Inferências do modelo (em azul) junto com os valores de *ground truth* (em vermelho) em imagens sem segmentação.

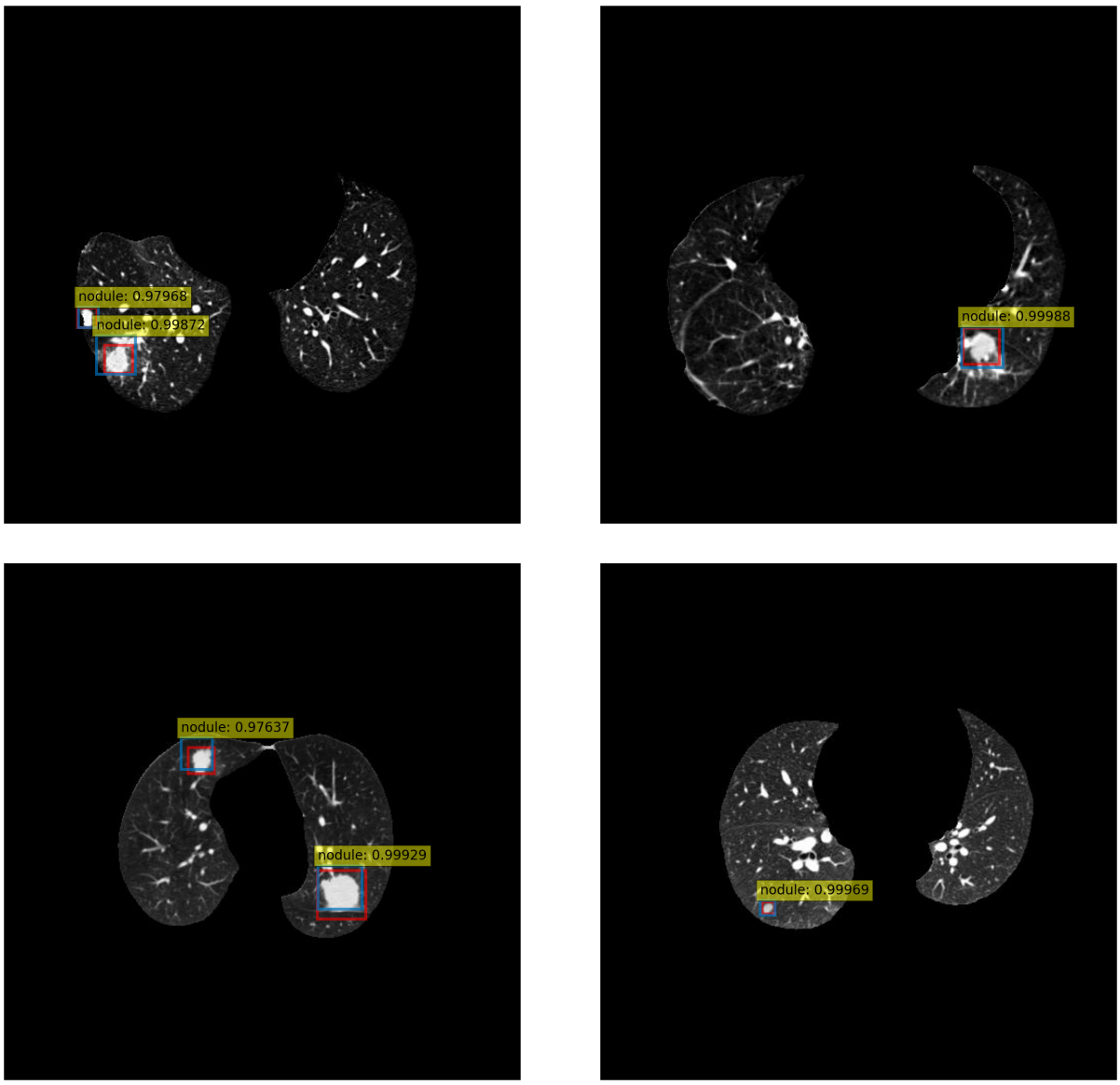


Figura 4.5: Inferências do modelo (em azul) junto com os valores de *ground truth* (em vermelho) em imagens com máscara de segmentação.

5

Conclusão

Este trabalho apresenta um *software* para a detecção automática de nódulos pulmonares utilizando a arquitetura *Transformer*. Foram usados exames torácicos de uma base pública e geradas imagens 2D, de forma a adequá-las ao modelo de detecção de imagens escolhido. Nosso modelo conseguiu valores de mAP de 86.6 e f1 de 0.882, resultados considerados satisfatórios, que confirmam boa precisão e *recall* (sensibilidade). As caixas de detecção dos nódulos apresentaram ainda bom ajuste em relação aos valores de *ground-truth*. Considerando os resultados alcançados neste trabalho, podemos afirmar que um modelo com a arquitetura Transformer é capaz de alcançar resultados similares a um modelo de CNN na detecção de nódulos pulmonares.

As inferências realizadas neste trabalho foram feitas utilizando um *Jupyter notebook*, mas o modelo pode facilmente ser portado para ser incorporado ao *pipeline* de software mais robusto ou mesmo ser adicionado a uma aplicação móvel, o que abre diversas possibilidades para o seu uso.

É importante frisar que durante o desenvolvimento deste Trabalho de Conclusão de Curso, uma enorme gama de assuntos e tecnologias foram explorados, em especial nas áreas de processamento de imagens, projeto de software, aprendizagem de máquina, visão computacional e imagens médicas. O conhecimento poderá ser utilizado pela comunidade para o desenvolvimento de novas ferramentas e a continuação dos estudos na área.

5.1 Trabalhos Futuros

Seguem algumas sugestões para trabalhos futuros:

- Melhorar o algoritmo de segmentação da região interna do pulmão utilizando algoritmos de inteligência artificial para um melhor aprimoramento da região de interesse (ROI) utilizada.

- Os modelos *Transformer* necessitam de um grande número de imagens para conseguir melhores resultados, então é importante adicionar imagens de outras bases de TCs com o objetivo de aumentar o número de amostras no treino e atingir melhores resultados.
- Testar outros hiperparâmetros no modelo e treiná-lo por mais épocas para tentar otimizar os resultados.
- Verificar a possibilidade de utilizar a ResNet modificada com camada de dilatação para melhorar resultados para objetos pequenos.

Referências bibliográficas

3DSlicer. Patient coordinate system, 2010. URL https://www.slicer.org/wiki/File:Coordinate_sytems.png.

Scott J Adams, Robert DE Henderson, Xin Yi, and Paul Babyn. Artificial intelligence solutions for analysis of x-ray images. *Canadian Association of Radiologists Journal*, 72(1): 60–72, 2021.

Wafaa Alakwaa, Mohammad Nassef, and Amr Badr. Lung cancer detection and classification with 3d convolutional neural network (3d-cnn). *Lung Cancer*, 8(8):409, 2017.

Luiz Henrique Araujo, Clarissa Baldotto, Gilberto de Castro Jr, Artur Katz, Carlos Gil Ferreira, Clarissa Mathias, Eldsamira Mascarenhas, Gilberto de Lima Lopes, Heloisa Carvalho, Jaques Tabacof, et al. Lung cancer in brazil. *Jornal Brasileiro de Pneumologia*, 44:55–64, 2018.

Jose de Arimateia Batista Araujo-Filho, Paulo Garcia Normando, Marcelo Dantas Tavares de Melo, André Nathan Costa, and Ricardo Mingarini Terra. Câncer de pulmão na era da covid-19: o que devemos esperar? *Jornal Brasileiro de Pneumologia*, 46, 2020.

Diego Ardila, Atilla P Kiraly, Sujeeth Bharadwaj, Bokyung Choi, Joshua J Reicher, Lily Peng, Daniel Tse, Mozziyar Etemadi, Wenxing Ye, Greg Corrado, et al. End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nature medicine*, 25(6):954–961, 2019.

Samuel G Armato III, Geoffrey McLennan, Luc Bidaut, Michael F McNitt-Gray, Charles R Meyer, Anthony P Reeves, Binsheng Zhao, Denise R Aberle, Claudia I Henschke, Eric A Hoffman, 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*, 38(2):915–931, 2011.

W Dean Bidgood Jr, Steven C Horii, Fred W Prior, and Donald E Van Syckle. Understanding and using dicom, the data interchange standard for biomedical imaging. *Journal of the American Medical Informatics Association*, 4(3):199–212, 1997.

Sean Blandin Knight, Phil A Crosbie, Haval Balata, Jakub Chudziak, Tracy Hussell, and Caroline Dive. Progress and prospects of early detection in lung cancer. *Open biology*, 7(9): 170070, 2017.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

Wook-Jin Choi and Tae-Sun Choi. Automated pulmonary nodule detection system in computed tomography images: A hierarchical block classification approach. *Entropy*, 15(2): 507–523, 2013.

Sijia Cui, Shuai Ming, Yi Lin, Fanghong Chen, Qiang Shen, Hui Li, Gen Chen, Xiangyang Gong, and Haochu Wang. Development and clinical application of deep learning model for lung nodules screening on ct images. *Scientific reports*, 10(1):1–10, 2020.

Andrew J Degnan, Emily H Ghobadi, Peter Hardy, Elizabeth Krupinski, Elena P Scali, Lindsay Stratchko, Adam Ulano, Eric Walker, Ashish P Wasnik, and William F Auffermann. Perceptual and interpretive error in diagnostic radiology—causes and potential solutions. *Academic radiology*, 26(6):833–845, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Srinivas Emani, Thomas D Sequist, Ronilda Lacson, Ramin Khorasani, Kunal Jajoo, Laura Holtz, and Sonali Desai. Ambulatory safety nets to reduce missed and delayed diagnoses of cancer. *The Joint Commission Journal on Quality and Patient Safety*, 45(8):552–557, 2019.

Alexandre X FALCÃO. Visualização de volumes aplicada à área médica. *São Paulo*, 1993.

José Raniery Ferreira, Marcelo Costa Oliveira, and Paulo Mazzoncini de Azevedo-Marques. Characterization of pulmonary nodules based on features of margin sharpness and texture. *Journal of digital imaging*, 31(4):451–463, 2018.

Macedo Firmino, Giovani Angelo, Higor Morais, Marcel R Dantas, and Ricardo Valentim. Computer-aided detection (cade) and diagnosis (cadx) system for lung cancer with likelihood of malignancy. *Biomedical engineering online*, 15(1):1–17, 2016.

Tania AC Furquim and Paulo R Costa. Quality assurance in diagnostic radiology. *Revista Brasileira de Fisica Medica (Online)*, 3(1):91–99, 2009.

F Gaillard et al. Radiopaedia: building an online radiology resource. European Congress of Radiology-RANZCR ASM 2011, 2011.

Rafael C Gonzalez. *Digital image processing*. Pearson education india, 2009.

Bell D Greenway, K. Hounsfield unit. reference article, 2021. URL https://doi.org/10.53347/rID-38181.

Amitava Halder, Debangshu Dey, and Anup K Sadhu. Lung nodule detection from feature engineering to deep learning in thoracic ct images: a comprehensive review. *Journal of Digital Imaging*, 33(3):655–677, 2020.

Godfrey N Hounsfield. Computed medical imaging. *Science*, 210(4465):22–28, 1980.

Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two pure transformers can make one strong gan, and that can scale up. *Advances in Neural Information Processing Systems*, 34, 2021.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873): 583–589, 2021.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Shunfeng Li, Chunxue Wu, and Naixue Xiong. Hybrid architecture based on cnn and transformer for strip steel surface defect classification. *Electronics*, 11(8):1200, 2022.

Yuemeng Li and Yang Fan. Deepseed: 3d squeeze-and-excitation encoder-decoder convolutional neural networks for pulmonary nodule detection. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 1866–1869. IEEE, 2020.

Jinglun Liang, Guoliang Ye, Jianwen Guo, Qifan Huang, and Shaohui Zhang. Reducing false-positives in lung nodules detection using balanced datasets. *Frontiers in Public Health*, page 517, 2021.

Lucas Lins de Lima et al. Modelo computacional para classificação de nódulos pulmonares utilizando redes neurais convolucionais. 2019.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

Dongxu Liu, Fenghui Liu, Yun Tie, Lin Qi, and Feng Wang. Res-trans networks for lung nodule classification. *International Journal of Computer Assisted Radiology and Surgery*, pages 1–10, 2022.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

Morgan P McBee, Omer A Awan, Andrew T Colucci, Comeron W Ghobadi, Nadja Kadom, Akash P Kansagra, Srini Tridandapani, and William F Auffermann. Deep learning in radiology. *Academic radiology*, 25(11):1472–1480, 2018.

MindwaysCTSoftware. Image of 3d volumetric qct scan, 2012. URL https://commons.wikimedia.org/wiki/File:Image_of_3D_volumetric_QCT_scan.jpg. Creative Commons license.

Baba Y Murphy, A. Windowing (ct). reference article, radiopaedia.org, 2021. URL https://doi.org/10.53347/rID-52108.

Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34, 2021.

Nasrullah Nasrullah, Jun Sang, Mohammad S Alam, Muhammad Mateen, Bin Cai, and Haibo Hu. Automated lung nodule detection and classification using deep learning combined with multiple strategies. *Sensors*, 19(17):3722, 2019.

Chuang Niu and Ge Wang. Unsupervised contrastive learning based transformer for lung nodule detection. *arXiv preprint arXiv:2205.00122*, 2022.

OMS. Global Health Estimates 2020: Deaths by Cause, Age, Sex, by Country and by Region, 2000-2019. who.int/data/gho/data/themes/mortality-and-global-health-estimates/ghe-leading-causes-of-death, 2020. Accessed: 2022-05-16.

Michael Onken, Marco Eichelberg, Jörg Riesmeier, and Peter Jensch. Digital imaging and communications in medicine. In *Biomedical Image Processing*, pages 427–454. Springer, 2010.

R. Padilla, S. L. Netto, and E. A. B. da Silva. A survey on performance metrics for object-detection algorithms. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 237–242, 2020.

Edward F Patz, Paul Pinsky, Constantine Gatsonis, JoRean D Sicks, Barnett S Kramer, Martin C Tammemägi, Caroline Chiles, William C Black, Denise R Aberle, et al. Overdiagnosis in low-dose computed tomography screening for lung cancer. *JAMA internal medicine*, 174(2):269–274, 2014.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

Daniele Ravì, Charence Wong, Fani Deligianni, Melissa Berthelot, Javier Andreu-Perez, Benny Lo, and Guang-Zhong Yang. Deep learning for health informatics. *IEEE journal of biomedical and health informatics*, 21(1):4–21, 2016.

Arnaud Arindra Adiyoso Setio, Alberto Traverso, Thomas De Bel, Moira SN Berens, Cas Van Den Bogaard, Piergiorgio Cerello, Hao Chen, Qi Dou, Maria Evelina Fantacci, Bram Geurts, et al. Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the luna16 challenge. *Medical image analysis*, 42:1–13, 2017.

Fahad Shamshad, Salman Khan, Syed Waqas Zamir, Muhammad Haris Khan, Munawar Hayat, Fahad Shahbaz Khan, and Huazhu Fu. Transformers in medical imaging: A survey. *arXiv preprint arXiv:2201.09873*, 2022.

Furqan Shaukat, Gulistan Raja, and Alejandro F Frangi. Computer-aided detection of lung nodules: a review. *Journal of Medical Imaging*, 6(2):020901, 2019.

Chris Solomon and Toby Breckon. *Fundamentals of Digital Image Processing: A practical approach with examples in Matlab*. John Wiley & Sons, 2011.

Ultralytics. Yolov5. https://github.com/ultralytics/yolov5/. Accessed: 2022-06-13.

Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17 (3):261–272, 2020.

Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5463–5474, 2021.

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22–31, 2021.

Rohola Zandie and Mohammad H Mahoor. Topical language generation using transformers. *arXiv preprint arXiv:2103.06434*, 2021.

Xiaoning Zhu, Yannan Jia, Sun Jian, Lize Gu, and Zhang Pu. Vitt: Vision transformer tracker. *Sensors*, 21(16):5608, 2021.

Xiaoyu Zhu, Xiaohua Wang, Yueting Shi, Shiwei Ren, and Weijiang Wang. Channel-wise attention mechanism in the 3d convolutional network for lung nodule detection. *Electronics*, 11(10):1600, 2022.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

6

Apêndice

6.1 Algoritmos

Listing 6.1: Algoritmo para aumentar o número de cortes

```
# Generate a list of slices to plot based on the nodule center and diameter
# Larger nodules will have more slices included

slices_dict = {}
for nodule in annotations_list:
    idx = nodule.center_irc[0]  # Get the nodule index (Z axis)

    # Add the center slice
    if idx in slices_dict:
      slices_dict[idx].append((nodule.center_irc[1],  # X position
                               nodule.center_irc[2],  # Y position
                               nodule.radius_px))
    else:
      slices_dict.setdefault(idx, []).append((nodule.center_irc[1],
                                              nodule.center_irc[2],
                                              nodule.radius_px))

    # Calculate how far the nodule goes in the Z axis
    axial_spacing = vx_size_xyz[0]/vx_size_xyz[2]
    depth = nodule.radius_px * axial_spacing

    # Keep only the slices up to half the radius
    half_depth = int(depth/2)

    for i in range(1,half_depth+1):
      # Add previous slice
```

```
    if idx-i in slices_dict:
      slices_dict[idx-i].append((nodule.center_irc[1],
                                 nodule.center_irc[2],
                                 nodule.radius_px))
    else:
      slices_dict.setdefault(idx-i, []).append((nodule.center_irc[1],
                                                nodule.center_irc[2],
                                                nodule.radius_px))

    # Add subsequent slice
    if idx+i in slices_dict:
      slices_dict[idx+i].append((nodule.center_irc[1],
                                 nodule.center_irc[2],
                                 nodule.radius_px))
    else:
      slices_dict.setdefault(idx+i, []).append((nodule.center_irc[1],
                                                nodule.center_irc[2],
                                                nodule.radius_px))
```