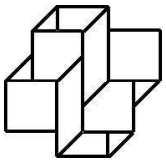


Laboratório Nacional de Computação Científica

Programa de Pós-Graduação em Modelagem Computacional

Detecção Automática do Coração em Tomografia Computadorizada Utilizando Técnicas de Aprendizagem Profunda

Jefferson da Silva Fernandes de Azevedo

Petrópolis, RJ - Brasil

Dezembro de 2022

Jefferson da Silva Fernandes de Azevedo

Detecção Automática do Coração em Tomografia Computadorizada Utilizando Técnicas de Aprendizagem Profunda

Dissertação submetida ao corpo docente do Laboratório Nacional de Computação Científica como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências em Modelagem Computacional.

Laboratório Nacional de Computação Científica

Programa de Pós-Graduação em Modelagem Computacional

Orientador(es): Pablo Javier Blanco e Carlos Alberto Bulant

Petrópolis, RJ - Brasil

Dezembro de 2022

Ficha catalográfica elaborada por Patrícia Vieira Silva - CRB7 5822

A994d Azevedo, Jefferson da Silva Fernandes de.

Detecção automática do coração em tomografia computadorizada utilizando técnicas de aprendizagem profunda / Jefferson da Silva Fernandes de Azevedo. - Petrópolis, RJ: Laboratório Nacional de Computação Científica, 2022.
89 f.: il.; 30 cm.
Referências: f. 77-80

Dissertação (Mestrado em Modelagem Computacional) - Laboratório Nacional de Computação Científica, 2022.
Orientadores: Pablo Javier Blanco e Carlos Alberto Bulant.

1. Coração. 2. Faster R-CNN 3D. 3. Detecção de objetos. 4. Tomografia computadorizada. 5. Redes Neurais Artificiais. I. Blanco, Pablo Javier. II. Bulant, Carlos Alberto. III. LNCC/MCTI. IV. Título.

CDD – 611.12

Laboratório Nacional de Computação Científica

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES

JEFFERSON DA SILVA FERNANDES DE AZEVEDO

DETECÇÃO AUTOMÁTICA DO CORAÇÃO EM TOMOGRAFIA COMPUTADORIZADA UTILIZANDO TÉCNICAS DE APRENDIZAGEM PROFUNDA

Dissertação submetida ao corpo docente do Laboratório Nacional de Computação Científica como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências em Modelagem Computacional.

Aprovada por:

Prof. Pablo Javier Blanco, D.Sc.
(Presidente)

Prof. Antonio Tadeu Azevedo Gomes, D.Sc.

Prof. Marco Antonio Gutierrez, D.Sc.

Documento assinado eletronicamente por **Pablo Javier Blanco**, **Coordenador de Métodos Matemáticos e Computacionais**, em 07/12/2022, às 13:21 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Antônio Tadeu Azevedo Gomes**, **Tecnologista em Ciência e Tecnologia**, em 08/12/2022, às 09:32 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **MARCO ANTONIO GUTIERREZ (E)**, **Usuário Externo**, em 08/12/2022, às 16:03 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site

https://sei.mcti.gov.br/verifica.html, informando o código verificador **10605449** e o código CRC **6BF4C449**.

Referência: Processo nº 01209.000098/2020-83 SEI nº 10605449

Agradecimentos

Este trabalho é resultado de esforço, paciência, companheirismo, consideração, profissionalismo e dedicação que vão além daqueles que poderiam ter sido praticados apenas por mim. Eu, como autor, reconheço e agradeço a todas as pessoas e instituições que contribuíram para que eu possa ser candidato a Mestre em Modelagem Computacional. Sendo assim, das instituições, agradeço ao Governo Federal pelo incentivo à formação de profissionais com o mais alto nível acadêmico, agradeço ao LNCC pela infraestrutura e ao CNPq pelas ajudas de custo dadas em forma de bolsa. Quanto às pessoas, agradeço à equipe HeMoLab por compartilhar seu conhecimento comigo e agregar alto valor técnico em minha formação acadêmica, dentre eles: Pablo, Carlos, Paulo, Alonso, Matheus e Daniel; agradeço aos colegas de turma do LNCC que nunca se opuseram a ajudar-me quando precisei: Andressa e Rafael; agradeço ao meu chefe, o Capitão-de-Mar-e-Guerra Walter, por flexibilizar a minha rotina de trabalho para que eu pudesse fazer as provas e participar das reuniões; e, finalmente, a minha noiva, Mariana, que em todos os meus momentos de dificuldade, exaustão e estresse, deu-me o suporte emocional necessário para que eu não desistisse e fez de mim tudo o que sou hoje.

Resumo

Algoritmos de inteligência artificial vêm sendo um forte tema de estudo na área da medicina assistida por computação científica. O objetivo de tal técnica é reproduzir tarefas que, até então, eram realizáveis apenas por seres humanos. A principal vantagem destes algoritmos é eliminar tediosas horas de trabalho executadas exclusivamente por humanos. Além de, em algumas abordagens, mitigar o viés que ocorre na marcação entre especialistas em um mesmo banco de dados. Mais especificamente, extrair somente a região que contenha o coração em uma Tomografia Computadorizada facilita diversas técnicas que fazem diagnóstico de doenças cardíacas. Para tal, pode-se utilizar um algoritmo de Redes Neurais Convolucionais dedicado à detecção de objetos, que, por sua vez, tem-se manifestado em diversos artigos científicos atuais em imagens bidimensionais, o Faster R-CNN. Este algoritmo teve uma adaptação para 3 dimensões no desafio *Multi-Modality Whole Heart Segmentation* de 2017. Assim sendo, nesta dissertação foram incrementadas melhorias e análises neste algoritmo apresentado. Mais precisamente, nas etapas responsáveis pela detecção da região do espaço 3D na qual se encontra incluído o coração. Após as devidas modificações e análise dos hiperparâmetros, obtivemos uma precisão de 74% na métrica *Intersection Over Union* no teste deste modelo, assim como há, em média, cortes na região do coração em torno de 6% de seu volume. Portanto, concluímos que o algoritmo implementado e estudado é capaz de detectar a região espacial da imagem tomográfica contendo o coração com precisão aceitável, tendo em vista que o intuito é reduzir o tamanho da imagem, destacando somente o essencial, a fim de facilitar o processamento de técnicas de diagnóstico.

Palavras-chave: *Faster* R-CNN 3D, Detecção de Objetos, Coração, Tomografia Computadorizada e Redes Neurais Artificiais.

Abstract

Artificial intelligence algorithms have been a strong topic of study in the field of computer-aided medicine. The goal of such a technique is to reproduce tasks that, until now, were only achievable by human beings. The main advantage of these algorithms is to eliminate tedious hours of work performed exclusively by humans. In addition, in some approaches, it mitigates the bias that occurs in the marking between experts in the same database. More specifically, extracting only the region containing the heart in a CT scan facilitates many techniques that diagnose heart diseases. To do this, a Convolutional Neural Network algorithm dedicated to object detection can be used, which, in turn, has been manifested in several current scientific papers on two-dimensional images, the Faster R-CNN. This algorithm had an adaptation for 3 dimensions in the 2017 Multi-Modality Whole Heart Segmentation challenge. Therefore, in this dissertation, improvements and analysis were incremented in this presented algorithm. More precisely, in the steps responsible for detecting the 3D space region in which the heart is included. After the appropriate modifications and analysis of the hyperparameters, an accuracy of 74% was observed in the metric Intersection Over Union in the test of this model, as well as there are, on average, cuts in the region of the heart around 6% of its volume. Therefore, we conclude that the algorithm implemented and studied is able to detect the spatial region of the CT image containing the heart with acceptable accuracy, considering that the intention is to reduce the size of the image, highlighting only the essential, in order to facilitate the processing of diagnostic techniques.

Keywords: *Faster R-CNN 3D, Object Detection, Heart, Computerized Tomography* e *Artificial Neural Networks.*

Lista de figuras

Lista de tabelas

Lista de abreviaturas e siglas

2D	2 Dimensões
3D	3 Dimensões
BN	*Batch Normalization*
DTL	*Detection Target Layer*
EQM	Erro Quadrático Mínimo
FFR	*Fractional Flow Reserve*
FPN	*Feature Pyramid Network*
FR	Fator de Redução
HeMoLab	*Hemodynamics Modeling Laboratory*
ID	Identificação
IoU	*Intersection Over Union*
MMWHS	*Multi-Modality Whole Heart Segmentation*
P3D ResNet	*Pseudo 3 Dimensions Residual Network*
PReLU	*Parameterized ReLU*
R-CNN	*Regions with Convolutional Neural Networks Features*
RC	Rede de Classificação
ReLU	*Rectified Linear Unit*
ResNet	*Residual Network*
RNA	Rede Neural Artificial
RNC	Rede Neural Convolucional
RNP	Rede Neural Profunda
RoI	*Region of Interest*
RPN	*Region Proposal Network*

SVM	*Support Vector Machinne*
TC	Tomografia Computadorizada
YOLO	*You Only Look Once*

Lista de símbolos

$\mapsto$	Mapeia para
$\in$	Pertence
∂	Derivada parcial
$\cap$	Interseção
$\cup$	União
$\mathbb{R}$	Conjunto dos números reais

Sumário

1 Introdução

Tendo em vista a importância da medicina para a vida humana, seja para alívio de sintomas ou cura de doenças, é importante ressaltar a contribuição que outras áreas proporcionam a ela. Mais especificamente, a matemática e a computação introduzem ferramentas poderosas que auxiliam na obtenção e processamento de informações que fornecem mais subsídios para o médico decidir qual caminho deve seguir para tratar seu paciente.

Dentre estas ferramentas, destaca-se a Tomografia Computadorizada (TC), que obtém imagens das estruturas anatômicas internas do tórax, por exemplo. Tais imagens permitem que a visualização dessas estruturas seja feita de maneira não invasiva, eliminando a agressão mecânica ao corpo do paciente, que é um fator crítico. Porém, tais imagens não dão suporte apenas à visualização, elas também possibilitam aplicar técnicas avançadas para predições de diagnósticos (BAKATOR; RADOSAV, 2018), que, para isso, utilizam modelos matemáticos processados em computadores. Tais modelos subsidiam o médico compilando informações que utilizam, entre outras técnicas: detecção de objetos de interesse (S. et al., 2018), segmentação de estruturas anatômicas (MA et al., 2020) e classificação (RAHIMZADEH; ATTAR; SAKHAEI, 2021). Ressalta-se que todas estas técnicas são meios cujo fim é diagnosticar um paciente.

Vale salientar que a possibilidade de predizer diagnósticos, assim como a implementação de modelos matemáticos para tal fim, é consequência do avanço de tecnologias na área da computação. Estas tecnologias englobam as técnicas de aprendizagem de máquina, um subcampo da engenharia e da ciência da computação que evoluiu do estudo de reconhecimento de padrões e da teoria do aprendizado computacional em inteligência artificial. Mais precisamente, Redes Neurais Convolucionais (RNC) têm sido fortemente utilizadas como método de processamento de imagens para modelos de aprendizagem de máquina. (ARAúJO, 2020)

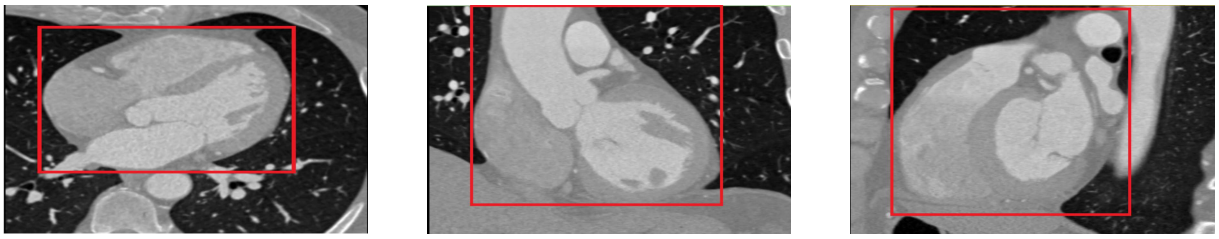
O processamento de imagens médicas mediante RNCs é usado comumente para resolver problemas de classificação, segmentação ou detecção de objetos. Nos últimos dois casos, o modelo de aprendizagem de máquina realiza um pré-processamento para uma análise posterior, que resultará na predição diagnóstica efetiva. Estas análises cobrem um amplo espectro, desde detecção, passando por medições geométricas, até simulação de escoamento sanguíneo (BEZERRA et al., 2019) e mecânica de tecidos (TALOU et al., 2018).

Os métodos supervisionados de aprendizagem de máquina requerem grandes volumes de dados corretamente etiquetados para poder treinar modelos que consigam

generalizar o resultado em novas imagens nunca vistas durante o treinamento. Embora a obtenção destes dados seja o maior requerimento, nem sempre é possível aplicar os métodos de aprendizagem diretamente sobre os dados. Geralmente o motivo para tal é o alto tempo de processamento, decorrente do poder computacional da máquina ser inferior ao exigido ou do número de operações necessárias ser demasiadamente grande, tornando inviável a utilização em uma situação real, assim como a quantidade de memória necessária pode exceder a capacidade de um computador tradicional.

1.1 Objetivo

Desenvolver um modelo de RNC para detecção da caixa delimitadora do coração, incluindo a aorta ascendente, em uma imagem de 3 dimensões (3D) de TC cardíaca com contraste, observe a Figura 1.



(a) Plano Axial. (b) Plano Coronal. (c) Plano Sagital.

Figura 1 – Imagens de tomografia com a caixa delimitadora do coração.

A representação do destaque do coração na TC é ilustrada em 3D na Figura 2. O coração é representado pela caixa vermelha contida dentro da caixa maior.

1.2 Justificativa

Tendo em vista o universo de possibilidades de avaliar clinicamente o coração com base em imagens de TC, ressalta-se que um dos cenários de análise é realizado por meio da *fractional flow reserve* (FFR) para avaliar a gravidade funcional da estenose coronariana. Um método de simulação por computador baseado em TC é uma abordagem não invasiva plausível para calcular o FFR (KWON et al., 2014). Porém, tendo em vista que para tal é necessário obter a geometria das artérias coronárias, extrair somente o coração na TC oferece grande benefício para a obtenção dessas, pois o processamento seria reduzido para esta única região. Contudo, considerando que o processo de obtenção da geometria da artéria é computacionalmente custoso, a fim de mitigar este fato, busca-se otimizar a quantidade de operações necessárias para compilar a informação. No caso de TC, tendo em vista que é uma imagem de 3 dimensões (3D), é exigido um espaço de armazenamento considerável para essa, em torno de 160MB, assim como a quantidade de parâmetros necessários para modelar e diagnosticar uma imagem com essas dimensões é enorme. Esse

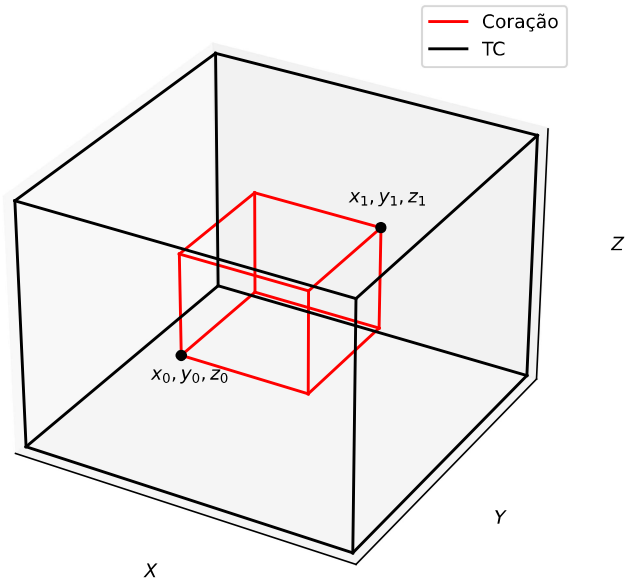


Figura 2 – Representação da caixa delimitadora em 3D.

tipo de limitação evidencia a necessidade dos processos intermediários, como a detecção de objetos, pois, caso o intuito seja avaliar apenas um órgão específico, não há necessidade de possuir os outros órgãos da região. Dessa forma, a detecção de objetos reduz a exigência de espaço de memória, otimiza o processamento dos parâmetros e, com isso, permite explorar mais técnicas para análise de sistemas anatômicos, além de tornar o modelo, ao todo, mais preciso.

Vale salientar que a detecção pode ser obtida por meio de técnicas manuais, entretanto o trabalho manual é tedioso para um ser humano realizar e é vulnerável ao viés de cada detector. Sendo assim, a elaboração de uma técnica automática é importante, pois otimiza o trabalho repetitivo realizado pelo especialista.

2 Técnicas de Aprendizagem de Máquina em Processamento de Imagens

Segundo (MITCHELL, 1997) "aprendizagem de máquina é o estudo de algoritmos de computador que permitem que programas de computador evoluam através da experiência". Ressalta-se que um fator importante desta técnica é a capacidade de reproduzir tarefas executadas por um ser humano que, em alguns casos, não necessariamente são matematicamente regidas por leis da física, como identificação de objetos em uma imagem. Dentro do subconjunto de aprendizagem de máquina há a aprendizagem profunda, em que há aplicação das Redes Neurais Artificiais (RNAs), que é uma técnica matemática inspirada na interconexão entre os neurônios do cérebro humano capaz de aprender, por meio de treinamento, a executar determinada tarefa.

2.1 Conceitos Básicos de Redes Neurais Artificiais

Na Figura 3 é representado o modelo de um neurônio artificial de (MCCULLOCH; PITTS, 1943). Matematicamente, o modelo se inspira no conceito biológico do neurônio humano: dado um sinal de entrada, produz um sinal de saída. O método de processamento da informação no neurônio é dado a partir da combinação da entrada $\mathbf{x}$, equivalente às sinapses, com seus respectivos pesos $\mathbf{w}$ e *bias* b associados. O resultado desta combinação, chamado de y, dado por

$$y = w_i x_i + b, \mathbf{x} \text{ e } \mathbf{w} \in \mathbb{R}^D, b \in \mathbb{R} \tag{2.1}$$

é processado por uma função de ativação que retorna a saída do neurônio. Vale salientar que a função de ativação, em redes neurais, é a relação matemática que define a transformação do valor obtido pela soma das entradas, pós processadas pelos pesos e *bias* (y), para o valor de saída do neurônio.

Tendo em vista o funcionamento individual de um neurônio, introduz-se o conceito de camadas, que são neurônios que recebem a mesma informação simultaneamente. Já o termo redes neurais diz respeito à interconexão entre neurônios. Inicialmente, a entrada do sistema passa pela primeira camada da RNA e, em seguida, a saída de cada neurônio alimenta a camada seguinte, e assim por diante, até a última camada da RNA, que é a saída desta. Todas as camadas entre a primeira e a última são chamadas de camadas escondidas e este processo, da informação seguir de uma camada à próxima, é chamado de propagação para frente ou *feedforward* em inglês. Em tempo, vale salientar que uma RNA

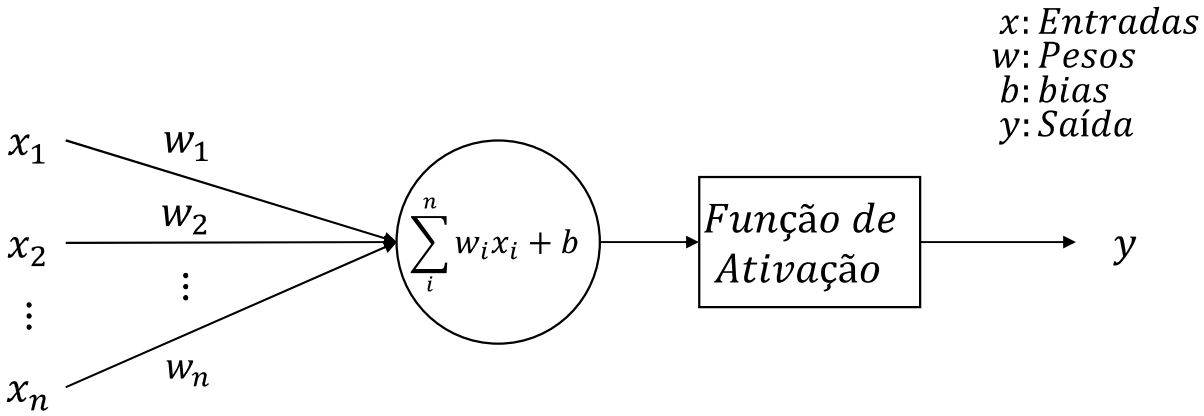


Figura 3 – Modelo matemático de um neurônio

formada por M neurônios é caracterizada pelo conjunto $\mathbf{P}$ de parâmetros, dados por $\mathbf{w}$ e b. Diferentes valores de $\mathbf{P}$ acarretam em potencialmente diferentes saídas para os mesmos valores de entrada. Logo, a propagação para frente pode ver vista como $y = RNA(\mathbf{x}, \mathbf{P})$. A Figura 4 ilustra uma RNA genérica.

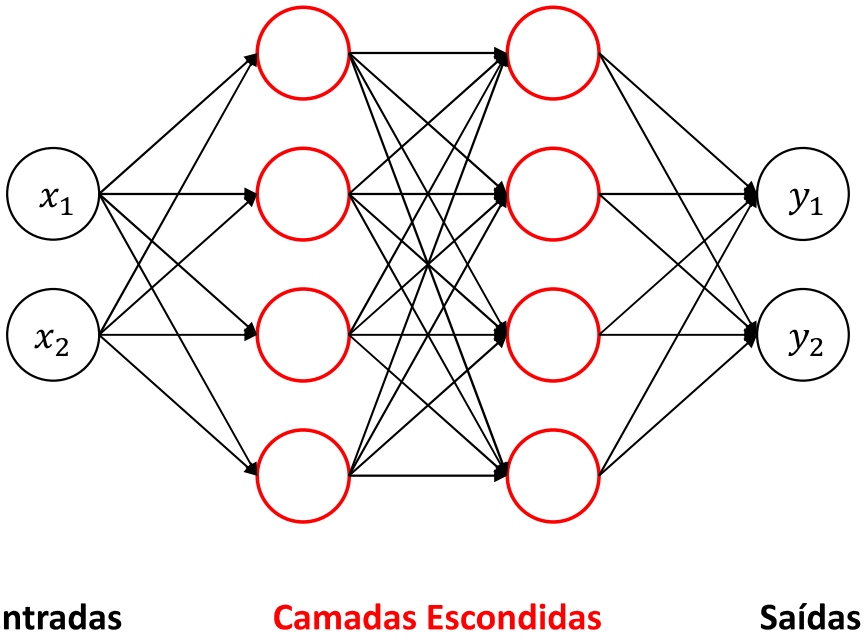


Figura 4 – Camadas em uma RNA com propagação para frente.

Ressalta-se que os tipos de função de ativação abordados neste trabalho são: *Rectified Linear Unit* (ReLU), *Parametric Rectified Linear Unit* (PReLU) e Softmax. A função de ativação ReLU pode ser definida como $F(x) = max(0, x)$, ou seja, dado o valor do neurônio de entrada, ReLU zera a entrada caso esta seja negativa e, caso contrário, não interfere no valor. Na Figura 5 pode ser observado o gráfico $F(x)$ da função ReLU (AGARAP, 2018).

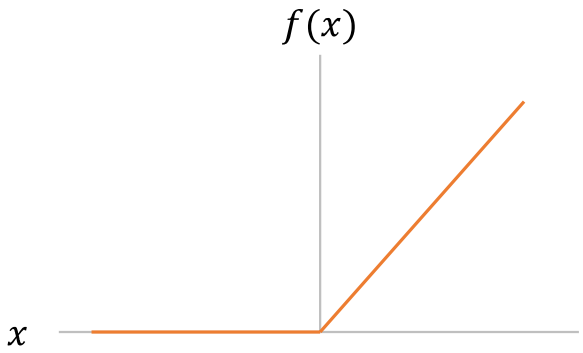


Figura 5 – Gráfico da Função ReLU

Já a função de ativação PReLU pode ser definida como

$$F(x) = \begin{cases} ax, & \text{se } x \leq 0 \\ x, & \text{caso contrário.} \end{cases} \tag{2.2}$$

similar à ReLU, porém quando a entrada possui valor negativo a saída é linear ao invés de nula. O gráfico da função PReLU pode ser observado na figura abaixo.

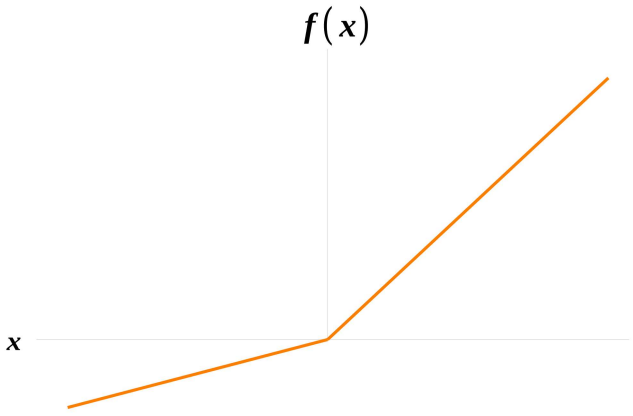


Figura 6 – Gráfico da Função PReLU

A função de ativação Softmax, geralmente, é aplicada na última camada do modelo, em etapas de classificação, pois normaliza a saída de uma rede em uma distribuição de probabilidades. Sendo assim, a saída do Softmax indica, separadamente, qual a probabilidade de cada classe ser referente à imagem avaliada.

Matematicamente, a função Softmax recebe um vetor $\mathbf{Z}$ com k números reais e os normaliza em distribuições normais consistindo em k probabilidades proporcionais ao exponencial dos números de entrada. Ressalta-se que a soma desses números normalizados,

obrigatoriamente, será 1, assim como todos os valores pertencem ao intervalo $(0, 1)$. Portando, os valores podem ser interpretados como probabilidades.

Dados $i \in (1, K)$ e $\mathbf{Z} = (z_1, \cdots, z_K) \in \mathbb{R}^K$, a função de ativação Softmax $\sigma : \mathbb{R}^K \mapsto (0, 1)^K$ é definida pela relação matemática:

$$\sigma(\mathbf{Z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{2.3}$$

Uma vez que a RNA está definida, faz-se necessário treiná-la. Baseado em um conjunto de dados $\mathbf{T}$, chamado de treinamento, tal que, dado $\mathbf{x}_i$ e $y_i \in \mathbf{T}$, $\mathbf{x}_i \mapsto y_i$, o treinamento consiste em encontrar o conjunto de parâmetros $\mathbf{P}^*$ ótimo de tal forma que estes treinados façam com que $\mathbf{x}_k \mapsto s_k$ e $|y_k - s_k|$ seja o mínimo de uma função de perda escolhida, como, por exemplo, o erro quadrático mínimo (EQM), Equação 2.4, em que s_k é a saída do modelo e n é a quantidade de pares (x_i, y_i).

$$EQM = \frac{1}{n}\sum_{i=1}^{n}(y_i - RNA(\mathbf{x}_i, \mathbf{P}))^2 \tag{2.4}$$

Quanto ao método de obtenção deste conjunto de parâmetros $\mathbf{P}^*$ ótimo, utiliza-se uma solução iterativa que parte de uma matriz de pesos $\mathbf{P}_0$ inicial e reduz a função de perda, como a Equação 2.4. Cada iteração é abordada como épocas (e) e o valor ideal dos pesos se dá no mínimo global da função de perda E, que pode ser alcançado utilizando o Gradiente Descendente ∇E, definido por

$$\nabla E = \frac{\partial E(\mathbf{P})}{\partial \mathbf{P}} \tag{2.5}$$

Como o vetor gradiente aponta para o crescimento da função (Figura 7), o negativo deste aponta para onde a função decresce, ou seja, seus mínimos. A atualização dos pesos, por época, ocorre conforme a Equação 2.6.

$$\mathbf{P}^{e+1} = \mathbf{P}^e - \eta \nabla E^e \tag{2.6}$$

É utilizado uma constante, definida como taxa de aprendizagem η, que serve para controlar a atuação do gradiente no conjunto de parâmetros treináveis da rede. Uma taxa de aprendizagem alta faz com que os pesos oscilem em torno do mínimo e nunca o atinja, já uma taxa pequena faz com que a convergência dos pesos seja lenta. Sendo assim, a taxa de aprendizagem é um fator importante a ser decidido na hora de estruturar um modelo.

Outro fator importante que envolve a atualização dos parâmetros é o lote de treinamento. Este consiste na quantidade de dados levados em consideração antes de atualizar os parâmetros do modelo. Existem 3 tipos de abordagem para tal:

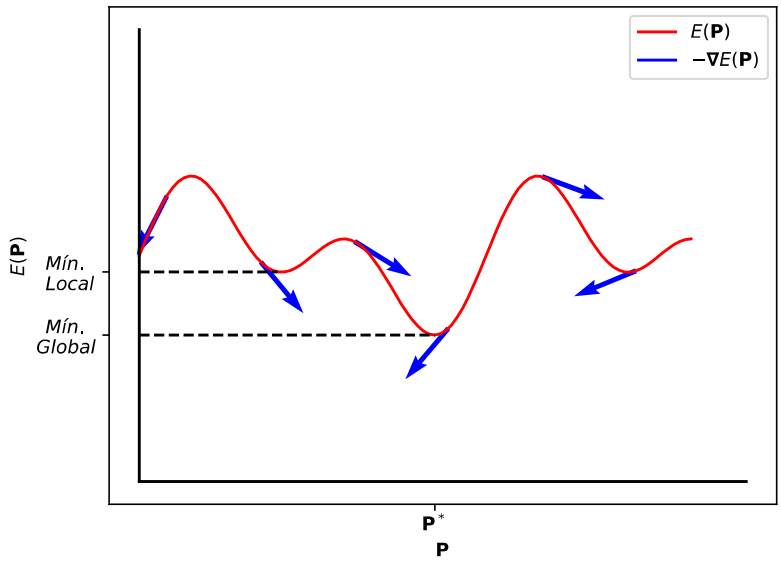


Figura 7 – Representação do gradiente de uma função.

1. Descida do Gradiente em Lote;
2. Descida do Gradiente Estocástica; e
3. Descida do Gradiente em Mini Lote.

Primeiramente, suponha-se um conjunto de dados de treinamento com N instâncias. A descida do gradiente em lote consiste em determinar o gradiente do erro em cada uma destas instâncias e, após analisar todas, faz-se a média destes gradientes e atualiza os parâmetros do modelo com base nesta média. O treinamento em lote possui duas vantagens: eficiência computacional, pois só atualiza os parâmetros depois de analisar todo o conjunto de dados, e convergência estável, tendo em vista que tende ao conjunto de parâmetros ótimo. Porém, também há desvantagens, que são: treinamento mais lento, pois os parâmetros são atualizados somente 1 vez após avaliar todo o conjunto de dados, e é mais susceptível a cair em mínimos locais.

Em segundo lugar, a descida do gradiente estocástica consiste em atualizar os parâmetros em cada instância avaliada, selecionada aleatoriamente. Vantagens desta abordagem: percepção de melhoria imediata, pois não precisa aguardar avaliar todas as instâncias para tal, e aprendizado mais rápido. As desvantagens desta abordagem são: gradientes mais ruidosos, exige maior esforço computacional e, como é atualizado de acordo com uma imagem por vez, não atinge o mínimo global da função de perda.

Por último, a descida do gradiente em mini lote une conceitos das outras 2 técnicas anteriores. Esta abordagem divide todas as q instâncias em β mini lotes, ao qual $1 \leq \beta \leq q$,

e atualiza os parâmetros do modelo de acordo com a média do gradiente obtida neste mini lote. Como vantagens, esta abordagem é computacionalmente eficiente, permite uma convergência estável e possui aprendizado rápido. Porém, a desvantagem para este tipo de atualização é que, agora, há um novo hiperparâmetro para ser ajustado, a quantidade β de instâncias.

Vale salientar que as RNAs são algoritmos considerados "caixa preta", ou seja, as operações realizadas internamente no seu processo decisório não possuem valor explicativo para o ser humano. Adicionalmente, conforme aumenta a complexidade do problema a ser abordado, como é o caso de processamento de imagens, consequentemente, aumenta a profundidade, número de camadas, da RNA, originando as Redes Neurais Profundas (RNPs), que, por sua vez, exigem maior poder de processamento para serem treinadas e processadas. (RAUBER, 2005)

Contudo, em virtude da capacidade de aprendizado dos algoritmos de inteligência artificial, há novos estudos voltados para o que é chamado de Inteligência Artificial Explicável (GUNNING et al., 2019). Tal objeto de estudo foca em tornar os processos de aprendizagem de máquina inteligíveis para seres humanos, fornecendo explicações sobre a natureza do objeto de estudo e não apenas predizendo resultados. Dessa forma, por motivos de melhor entendimento e, consequentemente, confiabilidade, a tendência para o futuro é que os algoritmos de inteligência artificial não sejam "caixa preta".

2.2 Redes Neurais Artificiais Aplicadas em Imagens

A aplicabilidade de um determinado modelo ao problema desejado depende da sua arquitetura, que é a disposição das camadas de acordo com o seu tipo, número de filtros convolucionais, função de ativação, entre outros. Os mais comuns, mas não limitados a estes, tipos de camadas em processamento de imagens são: convolucional, pool máximo, normalização em lote e totalmente conectada. Adicionalmente, cada uma destas camadas possui parâmetros de arquitetura, tais como: função de ativação, tamanho do *kernel* e quantidade de filtros. Estes, por sua vez, também são chamados de hiperparâmetros, e sua variação pode ser explorada a fim de verificar a possibilidade de otimização da rede. Vale ressaltar que os hiperparâmetros são definidos por quem constrói o modelo e não são treináveis.

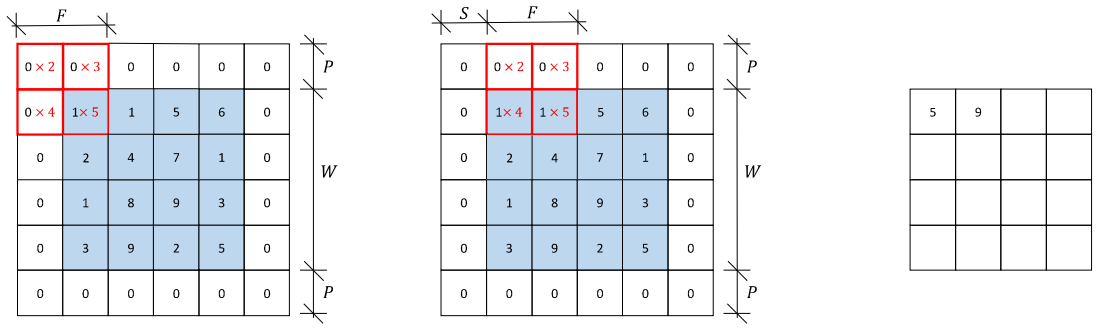
2.2.1 Convolução

Quanto à aplicação das RNPs em processamento de imagens, assume papel de destaque a camada de convolução, cuja saída está relacionada a uma janela, chamada de *kernel*, que percorre a imagem por etapas. Sendo assim, quanto a imagens, a convolução é uma técnica eficiente, pois leva em consideração uma vizinhança de pixeis e não oferece

aumento significativo de parâmetros treináveis, pois, mesmo que a imagem de entrada possua grande resolução, a quantidade de parâmetros depende exclusivamente do tamanho do *kernel*.

O *kernel* da convolução calcula a saída da seguinte maneira: dado o tamanho do *kernel* $m \times n$, portanto, ocorre o produto escalar entre o *kernel* e a região aplicada na imagem, ou Mapa de Recurso, de entrada. Como o *kernel* percorre a imagem com passo S, em cada iteração este se desloca S pixeis à direita, nas colunas, e, ao chegar no final da linha, S pixeis para baixo, nas linhas. Para convoluções 3D o processo é o mesmo, acrescido do deslocamento na terceira dimensão.

Vale ressaltar que a dimensão da saída da convolução depende do tamanho do *kernel*, do passo S e se há utilização de *padding*, preenchimento com zeros ao redor da imagem, que, por sua vez, permite o cálculo da convolução nos pixels do contorno. Na Figura 8 é ilustrado o processo de convolução. (DUMOULIN; VISIN, 2016)



(a) Primeiro passo da convolução

(b) Segundo passo da convolução

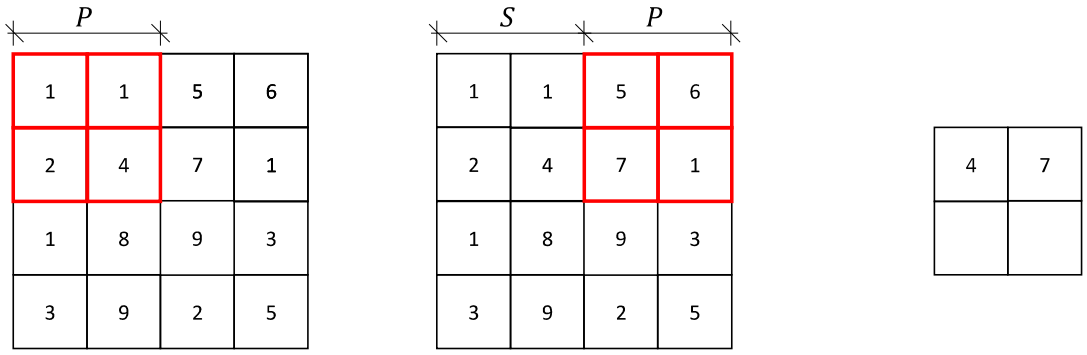
(c) Resultado parcial da convolução

Figura 8 – Representação do processo de convolução.

Nota: A imagem de entrada está representada com o fundo azul, o *padding* com fundo branco e o filtro em vermelho. Em que F, P, W e S são: tamanho do filtro, *padding*, largura da imagem e passo, respectivamente.

2.2.2 Camadas de Redução de Dimensão

A fim de destacar as características principais e, consequentemente, reduzir a dimensão da imagem são utilizadas camadas que atuam como filtros. Como exemplo há a camada pool máximo possui um funcionamento análogo à convolução, porém a operação retorna para cada janela o valor máximo da entrada. Este processo pode ser visualizado na Figura 9. Vale ressaltar que existem outros métodos cujo objetivo é o mesmo, tais como: pool mínimo e pool médio. Porém, o pool máximo é o mais comum na literatura. (GHOLAMALINEZHAD; KHOSRAVI, 2020)



(a) Primeiro passo do pool máximo. (b) Segundo passo do pool máximo. (c) Resultado parcial do pool máximo.

Figura 9 – Representação do processo de pool máximo.

Nota: O filtro está destacado em vermelho, sendo S e P o passo e o tamanho, respectivamente.

2.2.3 Normalização em Lote

O intuito desta camada é tornar o processo de treinamento da RNP mais rápido e estável. Sendo assim, consiste na normalização do vetor de entrada $\mathbf{x}$ em cada uma das camadas ocultas. Para isso, é considerado o primeiro e o segundo momento estatístico do lote de treinamento $\beta = \{x_1, \cdots, x_m\}$. Esta normalização é aplicada logo antes, ou após, da função de ativação.

Na prática, a camada normalização em lote transforma o sinal de entrada utilizando a média μ e a variância σ^2 dos valores do vetor de ativação $x^{(i)}$ ao longo do lote, as quais estão definidas como segue.

$$\mu = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i \tag{2.7}$$

$$\sigma^2 = \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - \mu)^2 \tag{2.8}$$

Em seguida, normaliza o vetor de ativação formado pela componente x_i utilizando a Equação 2.9.

$$x_i^{(norm)} = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{2.9}$$

Portanto, a saída de cada neurônio segue um padrão de distribuição normal ao longo do lote. Adicionalmente, ϵ é uma constante utilizada para estabilidade numérica. Por fim, a saída da camada é determinada por x^*, dado por:

$$x_i^* = \gamma \cdot x_i^{(norm)} + \beta \tag{2.10}$$

Esta etapa permite que o modelo escolha a distribuição ótima para cada camada escondida somente ajustando γ e β. Sendo assim, em cada iteração, a rede calcula μ e σ correspondentes a cada lote e, logo, treina γ e β por meio de gradiente descendente. (IOFFE; SZEGEDY, 2015)

2.2.4 Camada Totalmente Conectada

Esta camada consiste em gerar m neurônios de saída com base em n neurônios de entrada. Para tal, realiza-se uma multiplicação matriz-vetor $\mathbf{S}_{1\times m} = \mathbf{E}_{1\times n} \times \mathbf{W}_{n\times m} + \mathbf{B}_{1\times m}$, em que $\mathbf{E}$ são os neurônios de entrada e $\mathbf{S}$ os de saída, $\mathbf{W}$ e $\mathbf{B}$ são os parâmetros treináveis da camada. Ressalta-se que este processo consiste apenas em relacionar os neurônios de entrada com os de saída. Porém, esta informação $\mathbf{S}$ ainda será processada por uma função de ativação. O termo totalmente conectada parte da representação da ligação entre os neurônios conforme a Figura 10.

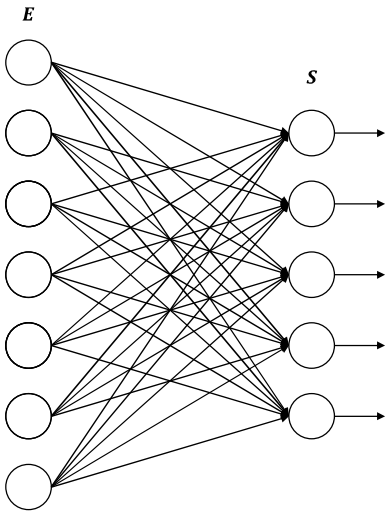


Figura 10 – Representação da camada totalmente conectada.

Como a função de ativação é utilizada para transformar os valores de entrada na saída, há casos em que é necessário explorar funções não-lineares para que as RNAs possam aprender a relação entre os neurônios de entrada e saída, que podem estar relacionados de maneira não linear.

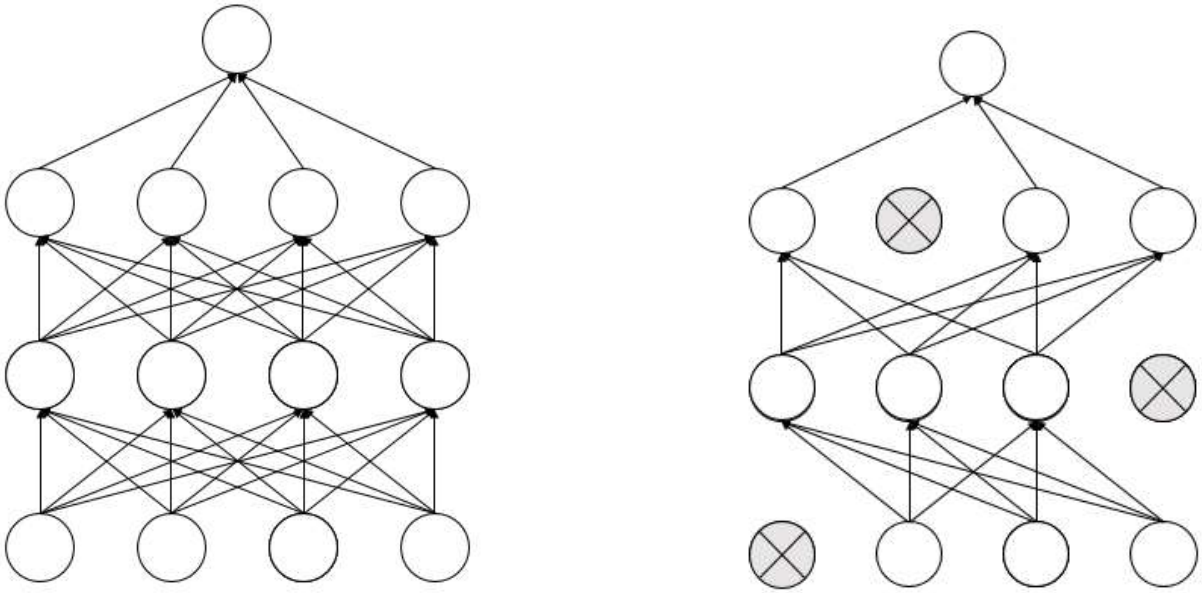
2.2.5 *Dropout*

Um modelo é considerado treinado quando o erro da função de perda ou a precisão dos resultados atinge um valor satisfatório. Porém, este valor não necessariamente se reproduz no conjunto de teste ou validação. Esse efeito é chamado de sobre-ajuste. Essa discrepância de resultados pode ser consequência de diversos fatores: má distribuição do conjunto de dados para treinamento e teste, ruídos nos dados de treinamento ou

treinamento excessivo, este último faz com que os dados "decorem" os resultados ao invés de generalizar.

Outro fator importante é que, a fim de dar maior credibilidade à predição do modelo, há a possibilidade de treinar o mesmo modelo com diferentes dados, ou diferentes inicializações, e, ao final, o resultado da rede é a média, ou moda, dos resultados obtidos em cada modelo. Porém, ressalta-se que há alto custo computacional em treinar diversos modelos. O fator mais crítico é a falta de confiança no modelo treinado, por haver a possibilidade de ter ocorrido o sobre-ajuste.

Entretanto, (HINTON et al., 2012) introduzem uma técnica, *Dropout*, que atua nestes dois problemas, pois mitiga os efeitos de sobre-ajuste e, ao mesmo tempo, permite explorar o efeito de diferentes redes no mesmo treinamento. O termo *Dropout* é referente a jogar fora unidades de uma RNA. Esta técnica anula os efeitos, temporariamente, de uma determinada quantidade de neurônios das camadas intermediárias que compõem o modelo, observe a Figura 11. Isso faz com que haja menor interdependência entre os neurônios da mesma camada e a atualização dos pesos destes seja mais significativa. A escolha de cada neurônio para ser ignorado é de maneira aleatória. Vale salientar que o *Dropout* ocorre somente durante o treinamento, no teste todos os neurônios são usados. Certamente, é notável que os pesos treinados, então, teriam um viés, pois foram treinados em circunstâncias especiais. Para compensar isso, durante o teste os pesos treinados são multiplicados por um fator p, que condiz com o percentual de neurônios excluídos pelo *Dropout*. (SRIVASTAVA et al., 2014)



(a) RNA padrão (b) Após aplicar *Dropout*

Figura 11 – Representação do *Dropout*

2.3 Revisão da Literatura

Tendo em vista o atual cenário da exploração de RNAs aplicadas a imagens médicas, inicialmente, faz-se necessário conhecer o estado da arte deste ramo científico, a fim de garantir que o método abordado está na vanguarda do campo de estudo, assim como é necessário conhecer as dificuldades encontradas pelos pesquisadores da área. Porém, apesar do problema deste trabalho ser em 3D, vale salientar que a maior parte do conteúdo sobre detecção de objetos é aplicado a imagens de bidimensional (2D). Dessa forma, inicialmente é necessário conhecer os conceitos sobre a técnica para, então, avaliar e implementar as adaptações feitas para atender aos problemas em 3D.

2.3.1 Modelos Dedicados à Detecção de Objetos

A detecção de objetos consiste em desenhar uma caixa delimitadora ao redor do objeto de interesse. Na prática, isso pode ser obtido por meio das duas coordenadas extremas da caixa delimitadora. Particularmente, para imagens tridimensionais, um modelo geral detecta n objetos 3D e retorna n vetores $\in \mathbb{N}^6$ que contenham as coordenadas $(x_1, y_1, z_1, x_2, y_2, z_2)$, referente aos pixeis extremos dos n objetos que se deseja identificar na imagem. Adicionalmente, ressalta-se que na detecção de objetos, diferentemente de modelos de segmentação e classificação, não é conhecida de antemão a quantidade de saídas, devido ao número variável de objetos que uma imagem pode conter. Por este motivo os modelos podem retornar n vetores de saída.

Diante deste cenário, destacam-se 4 modelos disponíveis na literatura que fazem a detecção de objetos em imagens 2D: R-CNN, *Fast* R-CNN, *Faster* R-CNN e YOLO, os quais serão abordados a seguir.

2.3.1.1 Modelo R-CNN

(GIRSHICK et al., 2014) Este modelo possui 3 módulos. O primeiro módulo gera propostas de região, que são o conjunto de melhores candidatos para conter o objeto. O segundo é uma RNC que extrai o Mapa de Recursos de cada região indicada. O terceiro módulo é um conjunto de *Support Vector Machine* (SVM).

Quanto ao primeiro módulo, este utiliza o algoritmo de Busca Seletiva (UIJLINGS et al., 2013) para indicar as propostas de regiões e o modelo se limita a propor 2000 destas. Sendo assim, ao invés de avaliar um número enorme de regiões, o modelo fica limitado a avaliar somente 2000, que já produz o resultado desejado e limita o custo computacional de gerar inúmeras propostas.

Já no segundo módulo, as imagens são cortadas em regiões de formato quadrado e alimentam uma RNC. A RNC atua como um extrator de recursos e a saída desta etapa, que é uma camada totalmente conectada, consiste nos recursos extraídos da imagem e,

posteriormente, alimentarão o SVM que classificará se o objeto está contido naquela região. Adicionalmente, o terceiro módulo, além de predizer se o objeto procurado está presente nesta região, indica, ainda, 4 valores de ajuste para serem aplicados a esta região indicada para melhorar a região de delimitação do objeto. A arquitetura deste modelo pode ser visualizada na Figura 12.

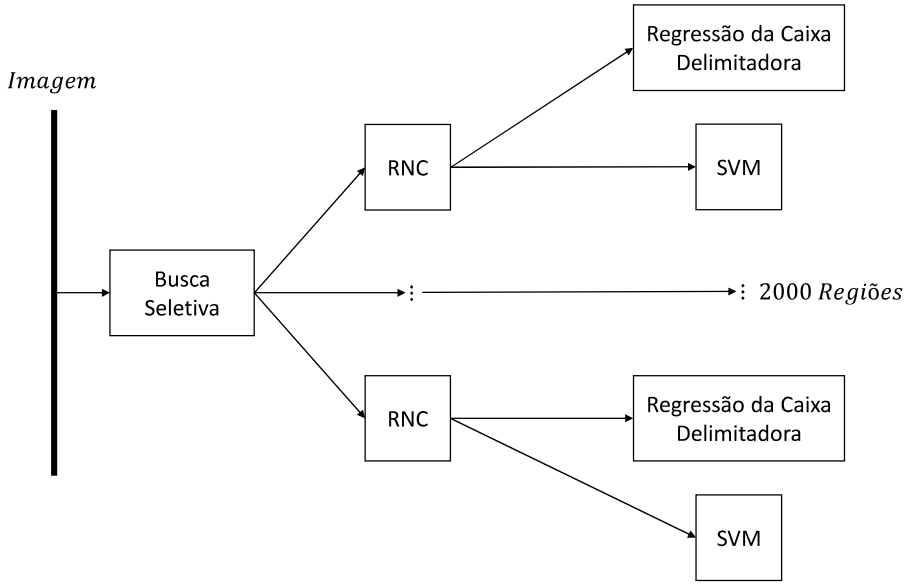


Figura 12 – Arquitetura do R-CNN.

As desvantagens deste modelo são:

1. Por mais que somente 2000 propostas de região sejam indicadas, estas ainda levam uma grande quantidade de tempo, pois todas precisam ser classificadas pelo terceiro módulo, mesmo que não contenham o objeto de interesse;

2. Não pode ser implementado em tempo real, pois o tempo de predição relatado pelo autor é significativamente elevado; e

3. Como o algoritmo de Busca Seletiva é uma etapa fixa, ou seja, não há processo de aprendizado deste, torna-se um gargalo por propor regiões ruins e não ter potencial de adaptar-se ao cenário aplicado.

2.3.1.2 Modelo *Fast* R-CNN

(GIRSHICK, 2015) Desenvolvido pelos mesmos autores do R-CNN, este modelo é uma versão aprimorada do primeiro, cujo objetivo da mudança é fazer a detecção de maneira mais veloz. O método de funcionamento é semelhante ao R-CNN, porém ao invés de alimentar a RNC com as propostas de região, alimenta-a com a imagem de entrada para gerar o Mapa de Recursos. Este Mapa de Recursos é utilizado para extrair as propostas

de região e, logo após, o quadrado da respectiva região é cortado. Em seguida, utiliza-se a camada RoI *Pooling* (RoI - *Region of Interest*) que formata as regiões cortadas para terem o mesmo tamanho para, em seguida, serem passadas à camada totalmente conectada. A partir do vetor de recursos RoI, utiliza-se uma camada Softmax para predizer a classe da proposta de região e, a partir dele, também, utiliza-se uma camada para fazer a regressão dos ajustes da caixa delimitadora. A arquitetura do modelo está representada na Figura 13.

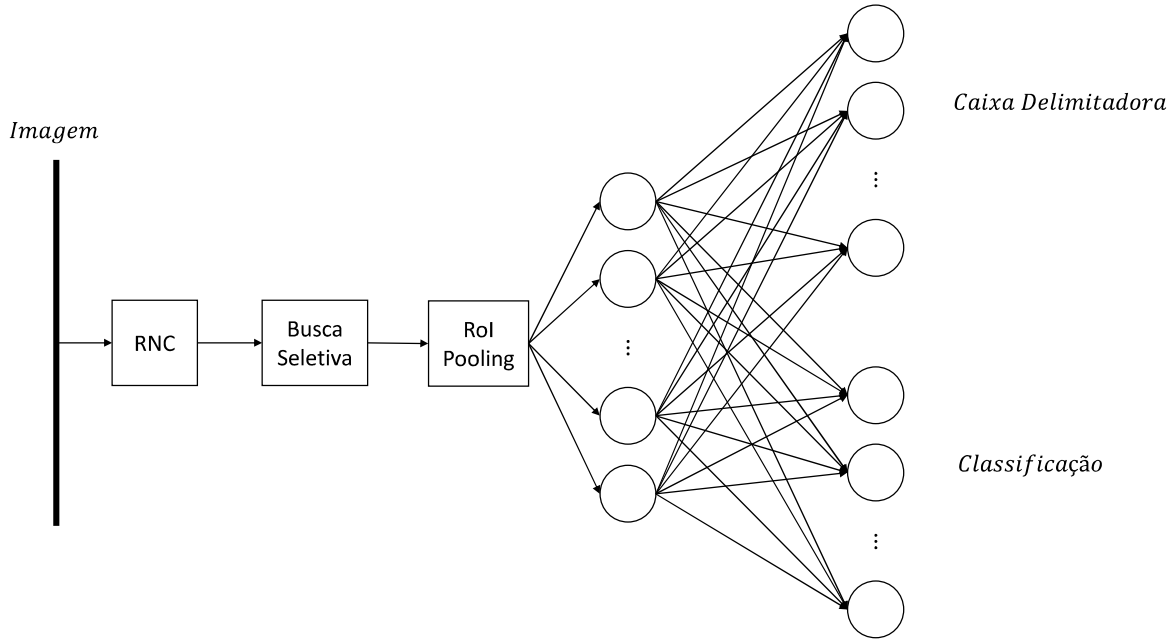


Figura 13 – Arquitetura do *Fast* RCNN.

O motivo pelo qual o *Fast* R-CNN é mais rápido que o R-CNN é, justamente, a alimentação da RNC ser a imagem de entrada e não as propostas de região indicadas pelo algoritmo de Busca Seletiva. Portanto, não há necessidade da RNC avaliar 2000 imagens toda vez, avalia apenas uma vez, gerando o Mapa de Recursos, e neste é aplicado o algoritmo de Busca Seletiva.

2.3.1.3 Modelo *Faster* R-CNN

(REN et al., 2015) Apesar da melhoria significativa do *Fast* R-CNN em relação ao R-CNN, esse ainda apresenta um gargalo comum ao antecessor, o uso do algoritmo de Busca Seletiva que, além de não evoluir ao longo do processo de treinamento, é custoso em relação a tempo. Portanto, os autores decidiram remover a Busca Seletiva e inserir uma sub arquitetura convolucional, que é a *Feature Pyramid Network* (FPN) seguida da *Region Proposal Network* (RPN). Ressalta-se que esta sub arquitetura não possui as duas desvantagens da Busca Seletiva citadas anteriormente e culminou em uma melhoria

significativa no tempo de inferência, cerca de 250× em relação ao R-CNN e 10× em relação ao *Fast* R-CNN.

Vale salientar que o *Faster* R-CNN permitiu que a detecção de objetos ocorra em tempo real, como em vídeos, pois o tempo gasto para processar cada imagem é cerca de 0.2 segundos. A arquitetura deste modelo pode ser visualizada na Figura 14.

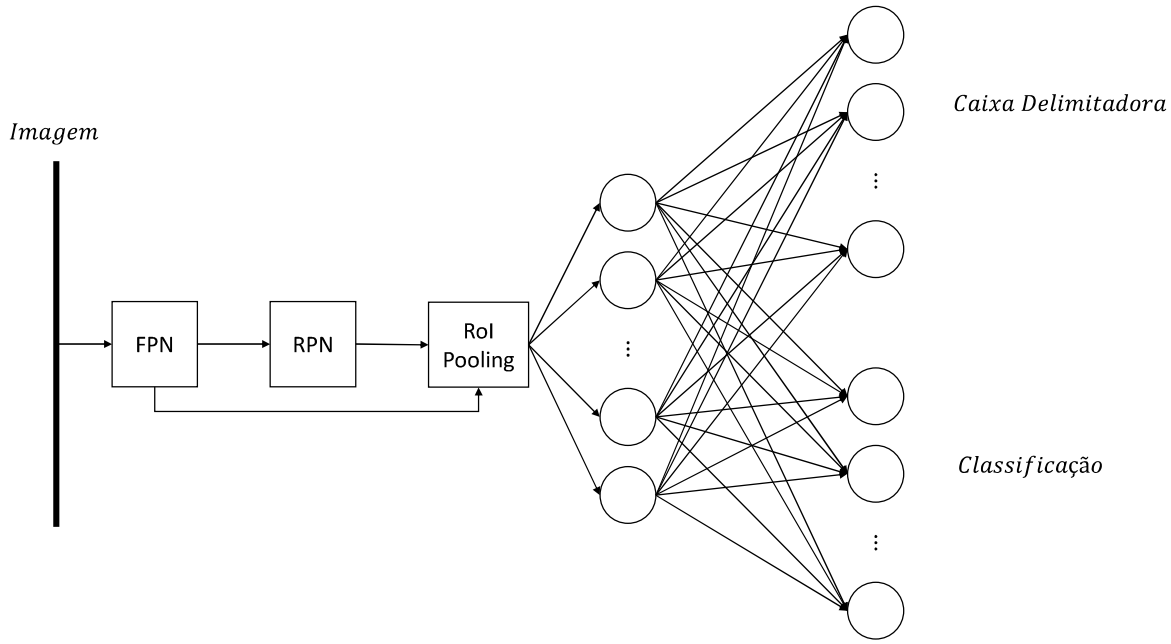


Figura 14 – Arquitetura do *Faster* R-CNN.

2.3.1.4 YOLO

Todos os modelos de detecção de objetos anteriores primeiramente localizam regiões de interesse na imagem e, em seguida, processa-as a fim de classificar se de fato há um objeto em potencial e ajusta as coordenadas de delimitação. Este processo é chamado de detecção em dois estágios. Já o YOLO (*You Only Look Once*) faz a detecção em um único estágio e avalia a imagem inteira em uma vez, sem precisar avaliar propostas de região.

Quanto ao funcionamento, o modelo aborda a detecção como um problema de regressão. YOLO utiliza uma imagem e a divide em uma grade $S \times S$, dentro de cada célula da grade prediz B caixas delimitadoras, que contém os valores: x, y, w, h e um grau de confiança para esta caixa conter um objeto, e C probabilidades de classe, observe a Figura 15. Dessa forma, a saída do modelo é um tensor no formato $S \times S \times (B \cdot 5 + C)$. A probabilidade de classe está relacionada a qual classe melhor se enquadra naquela célula da grade. Para o teste, são multiplicados a probabilidade de classe e o grau de confiança da caixa delimitadora, o qual indica a confiança específica de classe. As caixas delimitadoras

com probabilidade acima de um valor limite são selecionadas e usadas para localizar o objeto dentro da imagem.

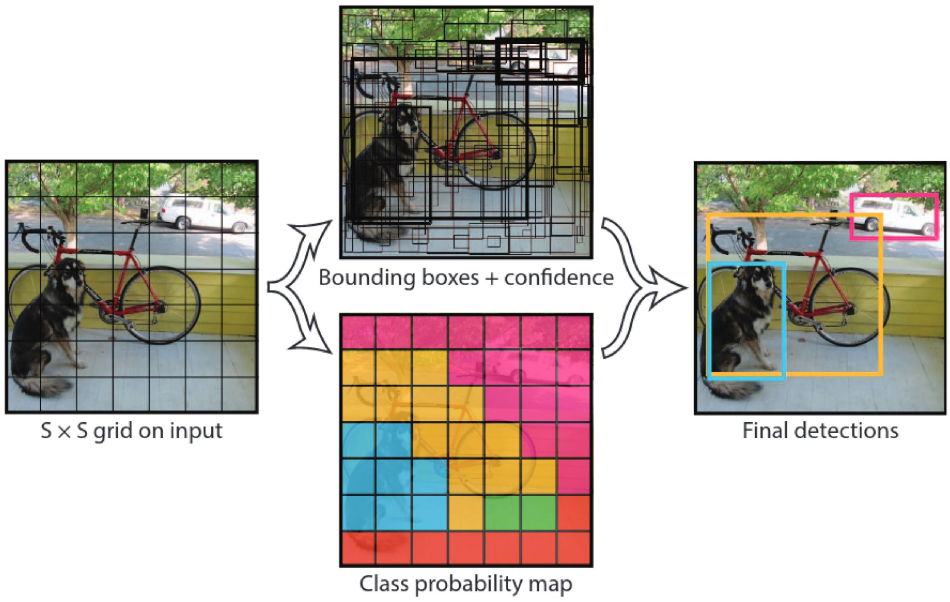


Figura 15 – Modo de funcionamento do YOLO.

Fonte: Redmon et al. (2016, p. 2)

A respeito da arquitetura, as camadas convolucionais iniciais da rede processam a imagem para extrair recursos e as camadas totalmente conectadas predizem as probabilidades das classes e suas respectivas coordenadas. Este modelo possui 24 camadas convolucionais seguidas por 2 camadas totalmente conectadas. A arquitetura do modelo pode ser visualizada na Figura 16.

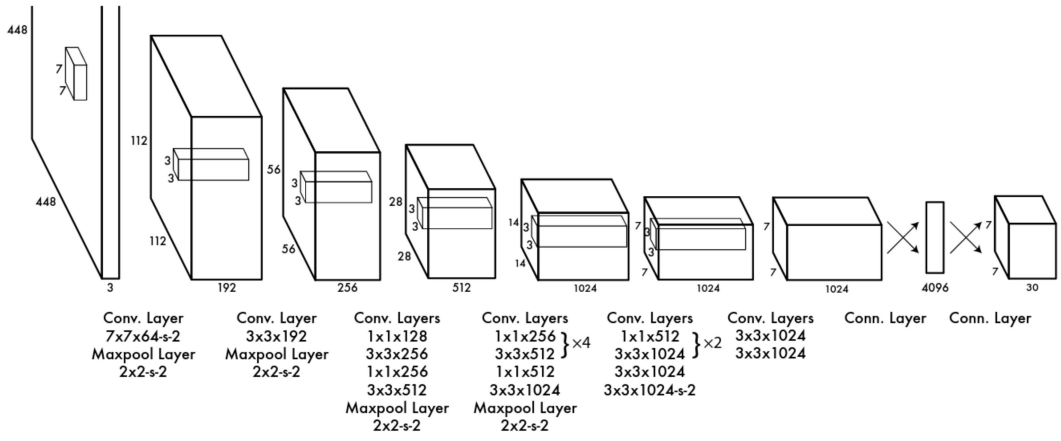


Figura 16 – Arquitetura do YOLO.

Fonte: Redmon et al. (2016, p. 3)

Porém, as desvantagens deste modelo estão relacionadas às restrições espaciais da caixa delimitadora, pois cada célula da grade prediz apenas duas caixas e pode conter somente uma classe. Estes fatores prejudicam a detecção de múltiplos objetos próximos, como bandos de pássaros. Vale ressaltar que este modelo apresenta dificuldade para generalizar objetos em proporções ou configurações diferentes das contidas no treinamento. Outro fator é que usa recursos relativamente grosseiros para prever caixas delimitadoras, uma vez que a arquitetura possui várias camadas de redução da resolução da imagem de entrada. Por último, a função de perda utilizada trata os pequenos erros de caixas delimitadores grandes iguais aos pequenos erros de caixas pequenas. Um pequeno erro em uma caixa grande é benigno, mas um pequeno erro em uma caixa pequena tem um efeito muito maior na métrica de precisão, o *Intersection Over Union* (IoU), definida por: (REDMON et al., 2016)

$$IoU = \frac{A \cap B}{A \cup B} \tag{2.11}$$

Em que A e B são caixas delimitadoras.

2.3.2 Trabalhos Dedicados à Detecção do Coração

Tendo em vista que já foi situado o avanço da arquitetura R-CNN e seu funcionamento, faz-se necessário procurar modelos que façam a detecção em imagens 3D, mais especificamente, modelos que destacam a presença do coração em imagens de TC.

O modelo apresentado em (HUMPIRE-MAMANI et al., 2018) é composto por 3 RNAs, cada uma avalia a presença do órgão em um dos eixos da TC, axial, coronal e sagital. Com a informação da presença do órgão na respectiva fatia do eixo, é possível definir a extremidade de cada órgão e, com isso, a caixa delimitadora. O modelo é alimentado por sequências de S fatias da TC por vez, ou seja, $512 \times 256 \times S$ nos eixos sagital e coronal e $256 \times 256 \times S$ no eixo axial. Ao processar esta entrada, cada RNA do modelo retorna um vetor de tamanho N, contendo a informação de quais N órgãos estão contidos naquela fatia indicada. A arquitetura do modelo é baseada em oito camadas convolucionais, quatro camadas pool máximo e uma camada totalmente conectada, com 600 neurônios, para predizer a presença dos N órgãos, observe a Figura 17. Quanto à precisão, concluiu-se que para $S = 2$ o modelo atinge IoU médio de $83\% \pm 0.23$ em todos os órgãos envolvidos, que eram: pulmões, rins, fígado, baço, vesícula biliar, sacro, bexiga e cabeças femorais.

Tendo em vista o alto desempenho do Faster R-CNN em imagens 2D, (XU et al., 2019) adaptou este modelo para ser aplicado em toda a imagem da TC, sem necessidade de aplicá-lo, separadamente, por eixo ou por fatia como ocorre em (HUMPIRE-MAMANI et al., 2018) e (ZHOU et al., 2012), respectivamente. O modelo adaptado é capaz de detectar o coração, dentre outros órgãos, com precisão de 70%. Porém, os autores argumentam

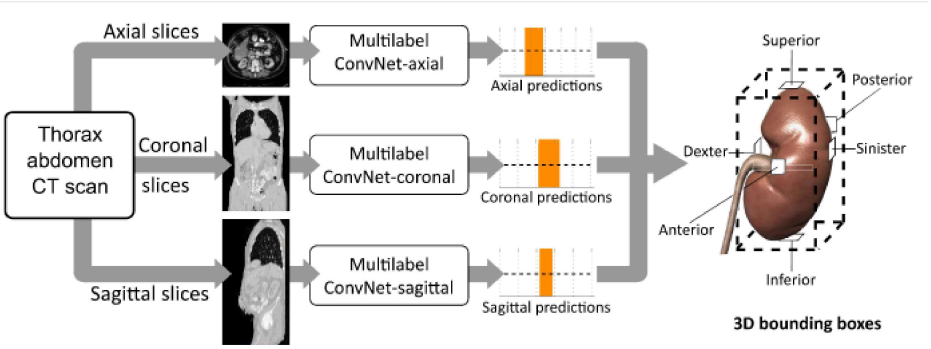


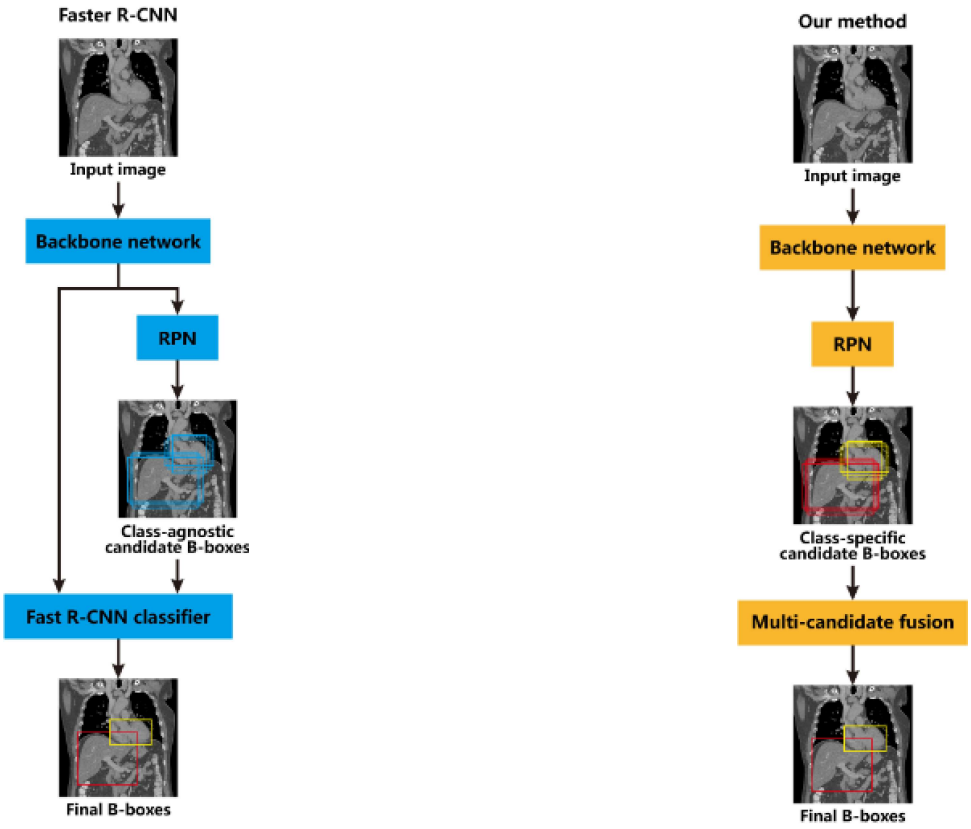
Figura 17 – Arquitetura do modelo de (HUMPIRE-MAMANI et al., 2018).

Fonte: Humpire-Mamani et al. (2018, p. 3)

que o Faster R-CNN é um detector de objetos gerais e que não necessariamente possui arquitetura ideal para ser aplicado em estruturas anatômicas, até porque foi desenvolvido para detectar uma quantidade arbitrária de instâncias para cada classe e nas imagens de TC não há um número arbitrário de instâncias do órgão. Sendo assim, simplificaram a estrutura do Faster R-CNN e desenvolveram o próprio modelo, observe a Figura 18. A diferença do modelo indicado para o Faster R-CNN é que, ao invés da RPN indicar propostas de região e estas serem filtradas pela camada subsequente de classificação, a própria classificação da RPN, cujo objetivo original era de somente acusar se a RoI é um objeto ou não, já faz a etapa de classificação, indicando qual órgão foi encontrado e a caixa delimitadora deste. A precisão deste modelo em relação ao coração é de 80.52%.

Já os autores (XU; WU; FENG, 2018) desenvolveram um modelo que faz tanto a detecção do coração como a segmentação do mesmo em seguida. O método de detecção também utiliza uma Faster R-CNN adaptada para 3 dimensões e uma 3D U-NET para, em seguida, segmentar a região indicada, observe a Figura 19. Quanto às restrições apontadas por (XU et al., 2019), estes autores fizeram alterações no Faster R-CNN de tal forma que sua configuração atue adequadamente em imagens de TC. Diferentemente de (XU et al., 2019), este modelo detecta somente o coração e, por isso, as generalidades quanto à quantidade de órgãos procurados e as diferentes classes não se aplicam. Sendo assim, o modelo é treinado para detectar somente o coração e, com isso, passa para a etapa de segmentação somente o mesmo. A precisão do modelo na segmentação dos dados de teste é de 85.9% e a segmentação ocorre nas estruturas anatômicas: cavidade sanguínea do ventrículo esquerdo, o miocárdio do ventrículo esquerdo, a cavidade sanguínea do ventrículo direito, a cavidade sanguínea do átrio esquerdo, a cavidade sanguínea do átrio direito, a aorta ascendente e a artéria pulmonar. O tempo de inferência para detecção e segmentação é de 15 segundos.

Uma abordagem diferente é feita em (SOANS; SHACKLEFORD, 2018), pois utilizam uma RNC apenas para indicar potenciais regiões de interesse onde há probabilidade



(a) Arquitetura original do Faster R-CNN.

(b) Arquitetura elaborada por (XU et al., 2019).

Figura 18 – Representação das modificações aplicadas por (XU et al., 2019) no modelo Faster R-CNN.

Fonte: Xu et al. (2019, p. 3).

de conter o coração, a estrutura pulmonar e, em seguida, aplicam uma RNC 3D para predizer se nesta respectiva região apontada pela RNC de fato há alguma das 3 regiões de interesse. A arquitetura desta segunda etapa do modelo é composta por 3 convoluções 3D seguida por unidades de max pooling, *Dropout* de 25% e *Flattening*, uma técnica que converte os mapas de recursos $N \times N$ para o formato de uma única coluna com N^2 elementos, observe a Figura 20. A probabilidade de conter cada classe é produzida na saída do segundo modelo através de uma camada de classificação Bayesiana. A precisão encontrada na primeira RNC para o coração é de 90.76% e sobe para 95.15% depois de aplicar a segunda RNC.

Em (CHHABRA; GAGAN; KUMAR, 2022) o trabalho visa a segmentação do ventrículo esquerdo (VE) em Imagens de Ressonância Magnética e, como meio para obter essa, faz-se a detecção do VE. Quanto à detecção, a imagem original é percorrida utilizando uma segunda imagem do VE como referência e se calcula um grau de semelhança entre essas para detectar o VE utilizando a técnica de correspondência de modelo normalizado multiescala. A precisão obtida, quanto à parte de detecção, é de 89.13%.

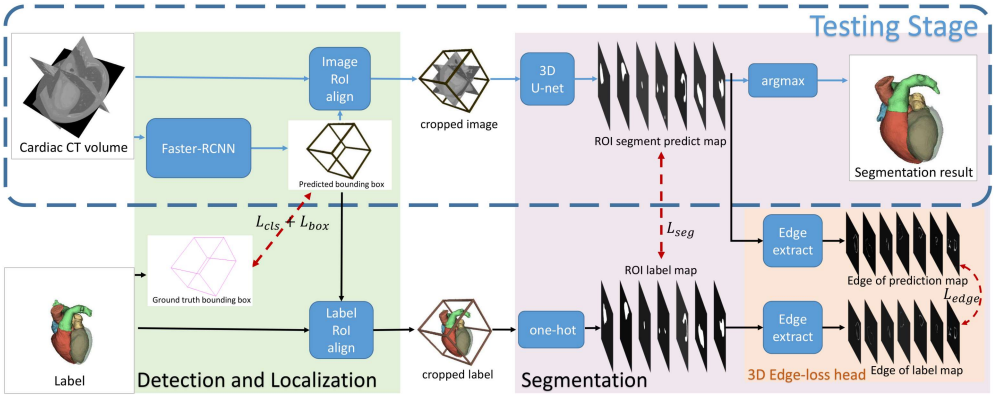


Figura 19 – Arquitetura do modelo de (XU; WU; FENG, 2018).

Fonte: Xu, Wu e Feng (2018, p. 2).

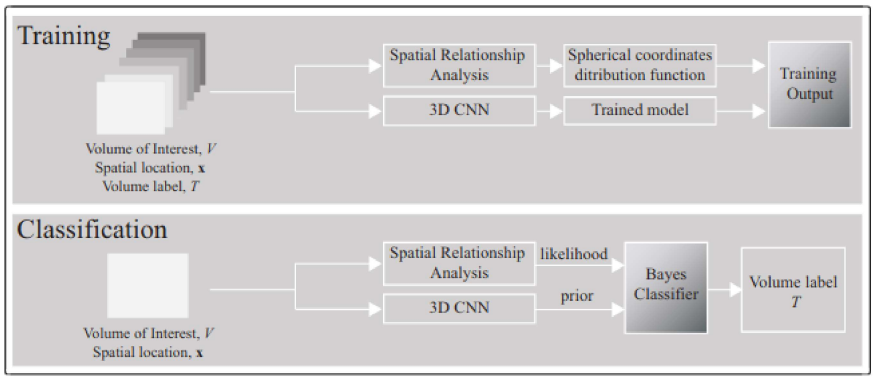


Figura 20 – Arquitetura do modelo de (SOANS; SHACKLEFORD, 2018).

Fonte: Soans e Shackleford (2018, p. 2).

3 Modelo de Aprendizagem de Máquina para Detecção do Coração em Imagem de TC

Tendo em vista as técnicas apresentadas no capítulo anterior, o modelo selecionado para abordar o problema é aquele proposto por (XU; WU; FENG, 2018), devido à adaptação do algoritmo de detecção de objetos Faster R-CNN de 2 dimensões (2D) para 3 dimensões (3D) e o forte alinhamento do objetivo daquele artigo com o deste trabalho. Diferente das outras técnicas (XU; WU; FENG, 2018) prediz somente a região do coração, ou seja, é otimizado para não avaliar as outras estruturas anatômicas contidas na Tomografia Computadorizada (TC) e retorna uma caixa delimitadora com a maior probabilidade de conter o coração. Entretanto, vale salientar que (XU; WU; FENG, 2018) possui, também, uma etapa de segmentação das 7 estruturas anatômicas do coração, que são:

i) Cavidade sanguínea do ventrículo esquerdo;

ii) Miocárdio do ventrículo esquerdo;

iii) Cavidade sanguínea do ventrículo direito;

iv) Cavidade sanguínea do átrio direito;

v) Aorta ascendente; e

vi) Artéria pulmonar.

Porém, inicialmente, esta etapa de segmentação não possui alinhamento com o objetivo deste trabalho, todavia, como possui atuação na função de perda do modelo, não se sabe se essa agrega valor à otimização dos pesos da etapa de classificação e regressão da caixa delimitadora, tendo em vista que uma melhor predição desta implica em melhoria da segmentação. Sendo assim, como uma melhor segmentação reduz a função de perda, a etapa de segmentação também faz parte da análise de desempenho do modelo de detecção e está inclusa no capítulo de resultados.

3.1 Tomografia Computadorizada

TC é um procedimento não invasivo de diagnóstico por imagem que utiliza raio X para criar imagens detalhadas dos tecidos do corpo humano. O procedimento é realizado através da emissão rotacionada de raios X ao redor do corpo, que, a depender de cada tecido, atenua o feixe de raios X que são absorvidos por detectores de radiação e enviam os

dados para um sistema computacional. Foi inventada pelo engenheiro eletrônico britânico Sir Godfrey Newbold Hounsfield, em 1972. Originando o nome Escala Hounsfield, que é uma escala utilizada em TC para distinção dos tons de cinza ao avaliar cada estrutura anatômica. (CARVALHO, 2007)

As imagens de TC podem ser aplicadas à região do crânio, em que imagens cranio-encefálicas são indicadas para o diagnóstico de traumatismos e hemorragias intracranianas, face, cuja aplicação é útil tanto no planejamento de cirurgias quanto na avaliação de alguns tipos de cistos e tumores, tórax, excelentes para detecção de alterações agudas ou crônicas do parênquima pulmonar, coração, permitindo a visualização das artérias coronárias e do músculo cardíaco, viabilizando a substituição de alguns processos invasivos, e abdominal e pélvica, para avaliação do sistema urinário. (MOURÃO, 2018)

Para fins de orientação, como a TC é uma imagem com 3D, cada eixo da imagem se refere a um dos planos anatômicos: sagital, coronal e axial (transversal). Observe na Figura 21 a definição destes planos.

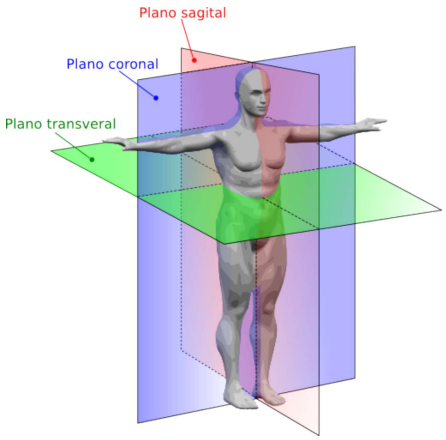


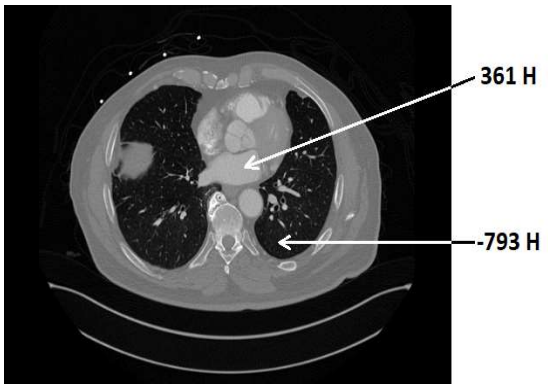
Figura 21 – Planos anatômicos.

Disponível em: Wikipedia; acesso em abril de 2022.

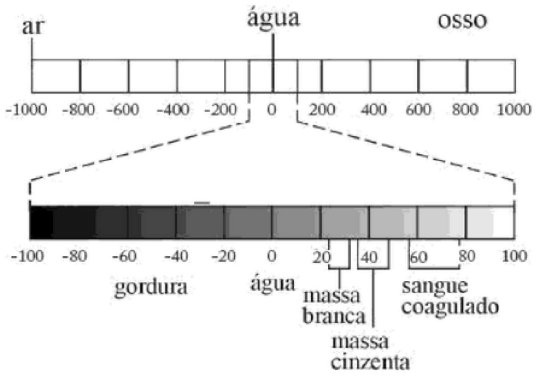
Quanto às imagens em si, a intensidade de cada voxel, menor unidade de volume para imagens 3D, varia de -1000 a 1000 na escala Hounsfield. Valor este que está relacionado ao coeficiente de atenuação de cada material presente no organismo, observe a Figura 22. A obtenção das imagens de TC é descrita em maiores detalhes no Apêndice B.

3.1.1 Dados Utilizados

Primeiramente, todos os pacientes envolvidos no presente estudo deram consentimento para a utilização das imagens médicas. Os protocolos dos estudos realizados na aquisição das imagens foram aprovados pelos comitês de ética dos centros de saúde correspondentes, e estão de acordo com a declaração de Helsinki.



(a) Representação dos valores da escala Hounsfield na TC.



(b) Escala Hounsfield aplicada em tecidos moles. Mourão (2018, Fig. 4.7).

Figura 22 – Representação dos valores da escala Hounsfield na TC

Quanto à distribuição dos dados, são utilizados 3 grupos ao longo do processo: treinamento, validação e teste. O grupo de treinamento contém os dados que servirão de base para o aprendizado do modelo. O grupo de validação é composto de dados que não são vistos durante o treinamento na hora de otimizar os pesos. Esse grupo é importante para verificar o grau de generalização do modelo, pois a obtenção de uma alta precisão durante o treinamento não necessariamente quer dizer que o modelo esteja aprendendo a predizer a informação, e sim "decorando" os resultados baseado na entrada. Este fenômeno é chamado de sobreajuste e um método de conferir, durante o treinamento, se os dados não estão sofrendo esse é a utilização do grupo de validação. Durante o treinamento é feito, a cada n épocas, a predição do modelo no grupo de validação e o resultado pode ser interpretado como uma transparência do grau de generalização do aprendizado. Obviamente, espera-se que a precisão e a função de perda na validação sejam, qualitativamente, ligeiramente inferiores ao treinamento. Por último, parte das imagens são dedicadas para o teste final do modelo. A utilização deste grupo implica no fato de aplicar o modelo, versão final, em dados nunca visto antes para saber o quão eficiente foi o treinamento. Vale salientar que um bom desempenho na validação não implica em bom desempenho no teste, pois o que pode ocorrer é que a alteração dos hiper-parâmetros esteja apenas otimizando a inferência no grupo de validação e não generalizando o aprendizado, além de expressar uma informação mais confiante como resultado final.

A respeito dos dados em si, (XU; WU; FENG, 2018) utilizaram o banco de dados do desafio de 2017 Multi-Modality Whole Heart Segmentation (MMWHS) e, para este trabalho, foram utilizados dados não públicos provenientes das seguintes instituições.

- Instituto do Coração do Hospital das Clínicas da FMUSP;
- Hospital Israelita Albert Einstein Morumbi;
- (CARSON et al., 2019); e

- Empresa FLOUIT.

Os dados provenientes das instituições acima serão referenciados como dados do *Hemodynamics Modeling Laboratory* (HeMoLab). Ressalta-se que a marcação das caixas delimitadoras foi feita no programa 3D Slicer. A distribuição das imagens de cada um desses grupos está representada na Tabela 1.

Tabela 1 – Distribuição das imagens.

Dados	**Treinamento**	**Validação**	**teste**
MMWHS	40	20	0
HeMoLab	94	36	36

3.1.1.1 Distribuição das Imagens

A distribuição das imagens nos grupos de treinamento, teste e validação deve ser feita de tal forma que não contenha viés. Isso ocorre se uma determinada característica se manifesta somente em um grupo e não nos outros. Por exemplo, se o grupo de treinamento contém somente imagens obtidas pelo tomógrafo A e o teste, ou validação, contém imagens do tomógrafo B, não vistas no treinamento, isso implica em um viés da informação e pode resultar em um modelo que não generalize bem para imagens obtidas com o tomógrafo B. Para evitar este tipo de viés nos dados do HeMoLab, a seleção dos grupos foi feita levando em consideração as informações, referentes a cada imagem, contidas na Tabela 2. Já para as imagens utilizadas no (XU; WU; FENG, 2018), no total os autores possuíam 60 imagens. Destas, 20 imagens foram manualmente marcadas por um especialista e as outras 40 foram marcadas por um algoritmo de segmentação. Destas que foram segmentadas, o MMWHS informou que possuem um *Intersection Over Union* (IoU) de 88% com a marcação fiel das estruturas segmentadas. Sendo assim, eles utilizaram as imagens rotuladas para validação e as outras para treinamento. Além disso, ressalta-se que o banco de dados do MMWHS não possui a informação detalhada de cada imagem, assim como ocorre com as do HeMoLab. Portanto, este tipo de distribuição não ocorre para aquele grupo.

Dentre as variáveis que representam as informações de cada imagem, há dois tipos: qualitativas e quantitativas. As variáveis qualitativas representam informações cujo conteúdo não é de valor numérico. Ou seja, não representam intensidade e sim característica. Já as variáveis quantitativas indicam uma intensidade capaz de mensurar algo e é representada por um valor numérico (BUSSAB; MORETTIN, 2010). O intuito é dividir as imagens em três grupos, treinamento, validação e teste, que possuam similaridade entre si em relação às informações apresentadas na Tabela 2.

Para tal fim, implementou-se uma metodologia de partição de dados que consiste nos seguintes passos: Primeiramente, são selecionadas por uma função aleatória as imagens que comporão cada grupo e a informação considerada para tal é a Base de Dados. A fim

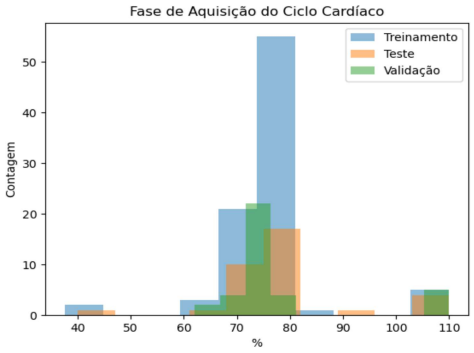
Tabela 2 – Informações utilizadas para segregar as imagens.

Informações Qualitativas	**Descrição**	**Quantidade de Imagens na categoria**
Base de Dados	FFR	39
	Heritability	82
	CFR	8
	Flouit	25
	benchmark-study	10
	medstar-ct-leaman	2
Modelo do Scanner	Aquilion ONE	94
	Discovery CT750 HD	9
	LightSpeed VCT	2
	Philips Brillance 64	19
	Philips iCT 256	3
	Siemens Somatom Definition AS+	17
	Siemens Somatom Definition Flash	22
Informações Quantitativas	**Média**	**Intervalo**
Espaçamento do Pixel (mm) Eixo X	0.42	0.29 - 0.86
Espaçamento do Pixel (mm) Eixo Y	0.42	0.29 - 0.86
Espaçamento do Pixel (mm) Eixo Z	0.34	0.24 - 0.7
Fase de Aquisição do Ciclo Cardíaco (%)	73	37.5 - 90
	14 imagens sem rótulo	
Ano do Estudo	2013	2006 - 2019
Volume relativo do coração e aorta ascendente (%)	31	4.6 - 71.5

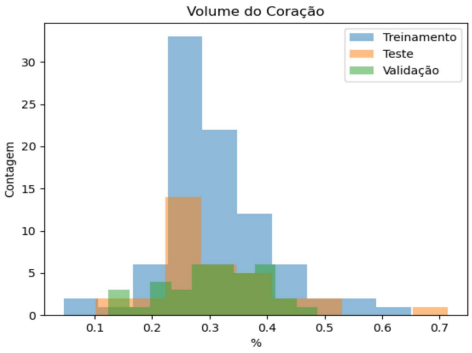
de validar tal segregação são utilizados dois testes estatísticos: o teste Chi-Quadrado para as variáveis qualitativas e o teste U de Mann-Whitney para as quantitativas (Apêndice A). Ressalta-se que ambos são testes não-paramétricos, ou livres de distribuição, pois não fazem suposição prévia a respeito da forma das distribuições das populações comparadas. Cada teste indica se para aquela variável analisada os grupos possuem similaridade ou não. Caso não possua, a seleção aleatória é refeita utilizando outra semente até que este critério seja atendido para todas as variáveis. Para tal, assume-se que a hipótese nula H_0 é que os dados entre os grupos são similares, já a hipótese alternativa assume que estes são diferentes. O critério de similaridade para os testes leva em consideração um nível de significância de 5%.

Sendo assim, para obter os dados de teste, utiliza-se o algoritmo para extrair 20% do total de imagens, o restante permanece para treinamento. Destes dados para treinamento, aplica o algoritmo novamente para se obter os dados de validação. Porém, agora, na proporção 25%, pois 25% dos dados do treinamento equivalem a 20% do total de dados. Os valores p obtidos para os grupos de teste e validação em relação ao grupo de treinamento estão contidos na Tabela 3.

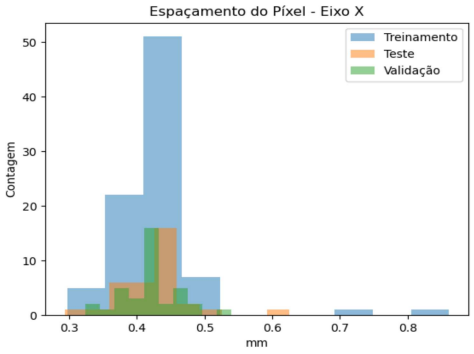
A Figura 23 ilustra a dispersão dos dados entre os grupos de treinamento, validação e teste. Vale ressaltar que referente às imagens as quais não há informação sobre a Fase de Aquisição do Ciclo Cardíaco foi atribuído um valor de 110% a essas, que está fora do intervalo de 0 a 100%, para realizar a distribuição.



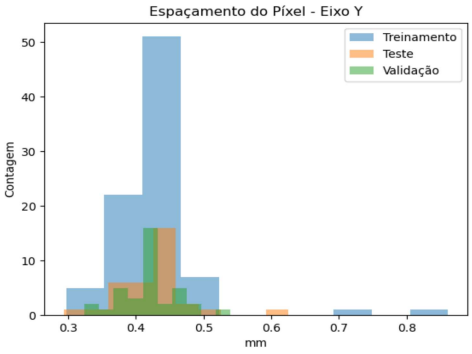
(a)



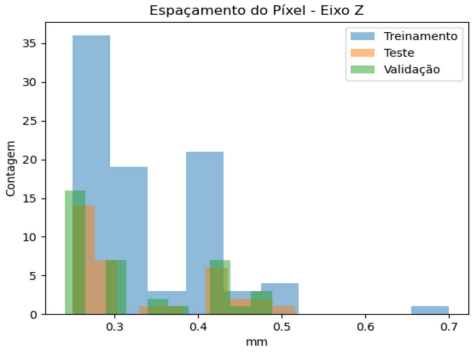
(b)



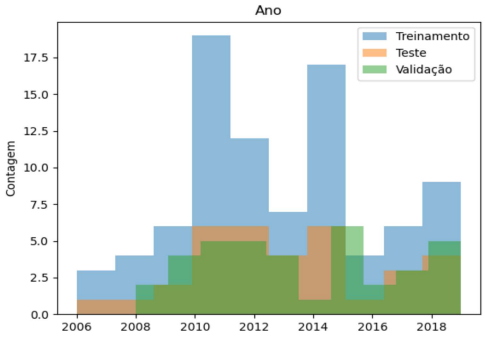
(c)



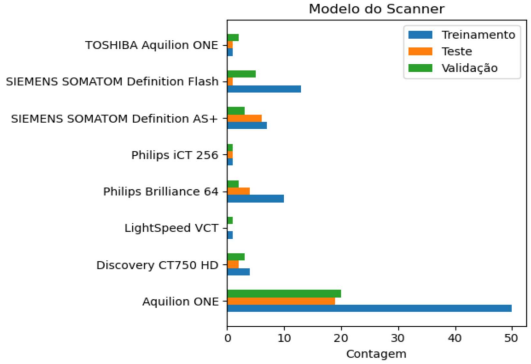
(d)



(e)



(f)



(g)

Figura 23 – Histogramas da distribuição dos dados entre os grupos de treinamento, teste e validação.

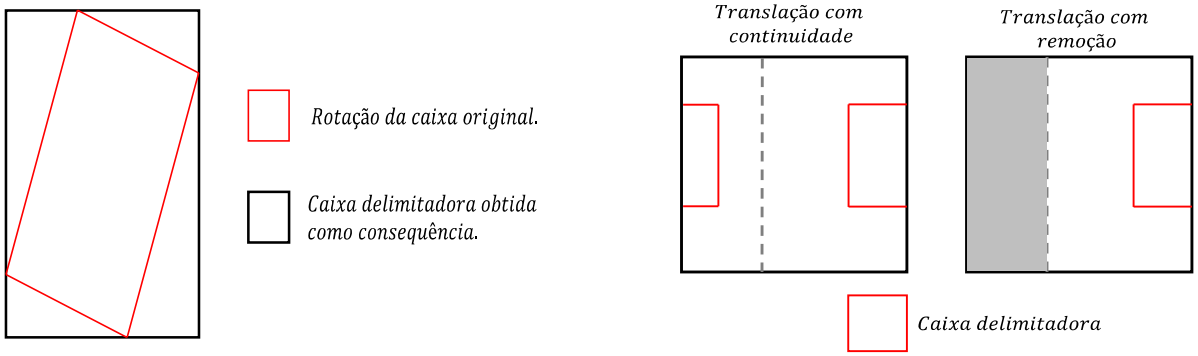
Tabela 3 – Valores p obtidos entre os grupos.

Informação	**Teste**	**Validação**
Modelo do *Scanner*	0.573	0.626
Fase de Aquisição do Ciclo Cardíaco	0.732	0.116
Espaçamento do Pixel no eixo X	0.891	0.595
Espaçamento do Pixel no eixo Y	0.891	0.595
Espaçamento do Pixel no eixo Z	0.759	0.798
Ano do Estudo	0.870	0.331
Volume do Coração	0.902	0.668

3.1.1.2 Aumento dos dados

Tendo em vista que a quantidade de dados para treinamento tem grande influência na qualidade do mesmo, porém nem sempre há uma enorme quantidade de dados disponível para tal, principalmente quando se trata de imagens médicas, são utilizadas técnicas para gerar mais dados de treinamento baseado nos dados atuais. No caso de imagens, as técnicas englobam, entre outras: rotação, translação, cortes, zoom e filtro gaussiano. (SHORTEN; KHOSHGOFTAAR, 2019)

As técnicas de aumento de dados abordadas neste trabalho são: filtro gaussiano, zoom e cortes laterais. O motivo de ter escolhido estas técnicas ao invés de outras é que estas não afetam radicalmente a caixa delimitadora e, portanto, não há necessidade de processar manualmente a saída desejada. Por exemplo, levando em conta que a predição deste modelo não prevê angulação, a rotação ocasionaria um aumento na caixa delimitadora final, que poderia prejudicar a precisão da rede, observe a Figura 24a. Outra técnica que pode prejudicar a qualidade da caixa delimitadora é a translação, que desloca as colunas de pixels ao longo de um eixo e, como consequência, copia a parte que foi deslocada para fora da limitação da imagem ao início desta ou, simplesmente, corta o trecho que saiu da imagem, observe a Figura 24b.



(a) Consequência da Rotação. (b) Consequência da Translação.

Figura 24 – Representação da rotação e translação como aumento de dados.

Acerca das técnicas de aumento de dados utilizadas, vale ressaltar que, dependendo da técnica, é necessário processar a imagem da TC e a caixa delimitadora. O aumento de

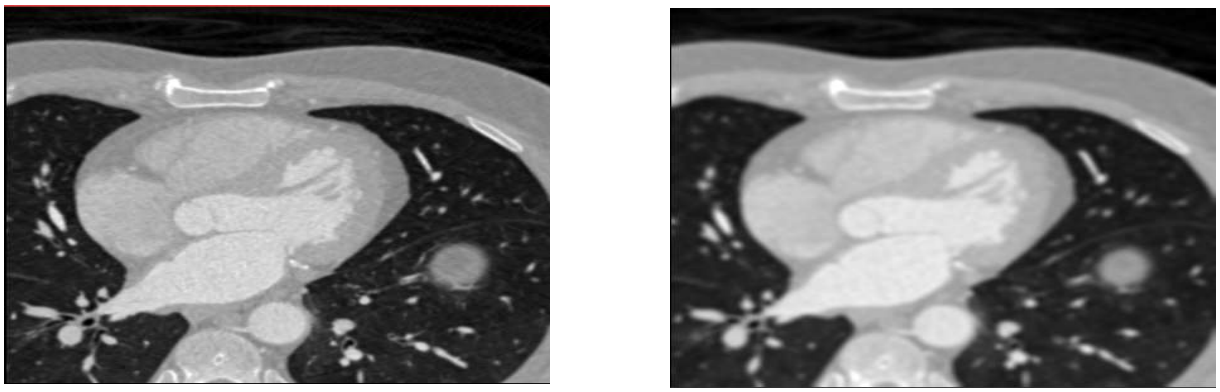
dados utilizado passa por 3 processos: filtro gaussiano na TC, cortes laterais e zoom.

Quanto ao filtro gaussiano, este realiza um desfoque na imagem utilizando a função Gaussiana em 2D, dada por:

$$g(x, y) = ce^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.1}$$

Em que x e y são as coordenadas em pixeis, c é uma constante definida de tal forma que a área sob a curva seja 1 e σ é o desvio padrão da distribuição Gaussiana. Sabendo que o tensor da TC é de 3D, $X \times Y \times Z$, o filtro Gaussiano é aplicado em cada fatia $X \times Y$ ao longo do eixo Z. Adicionalmente, vale ressaltar que esta técnica não afeta a caixa delimitadora, sendo assim, é aplicada somente na imagem de entrada.

A fim de obter maior grau de variedade das imagens no aumento de dados, o valor de σ, em cada chamada para uma nova imagem, é dado, aleatoriamente, entre 0, sem filtro, e 3, valor escolhido. Observe a Figura 25. (SHAPIRO, 1992)



(a) Fatia original.

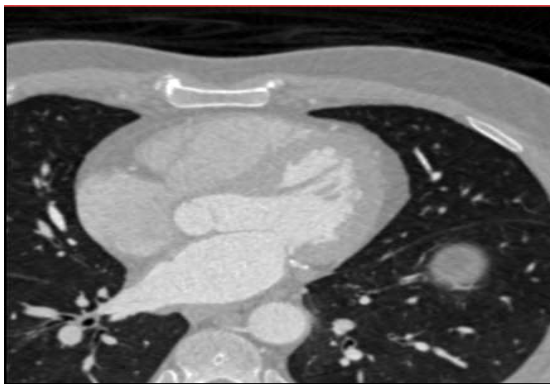
(b) Fatia com filtro Gaussiano com $\sigma = 3$.

Figura 25 – Exemplo de filtro Gaussiano.

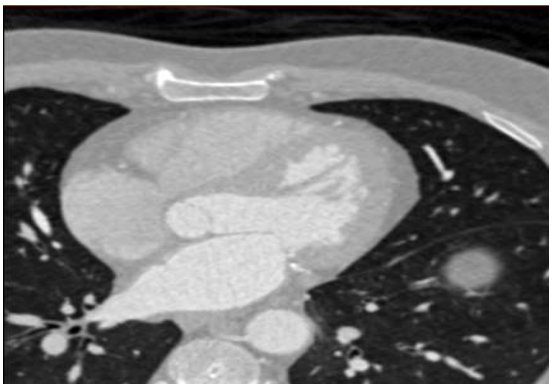
Em seguida, a imagem resultante do filtro passa por uma etapa de cortes laterais. A técnica consiste em remover até n linhas ou colunas de cada lado da imagem. Como resultado, obter-se-ia uma imagem com dimensões $X' \times Y'$, diferentes da original. Ressalta-se que este aumento de dados, neste trabalho, não interfere no eixo Z. Para corrigir isso, é feito um processo de interpolação, que mantém o resultado do corte com mesma dimensão da imagem original. Observe a Figura 26.

Por último, o resultado das etapas anteriores passa por um processo de zoom. Tendo em vista que um zoom inferior a 100% resultaria em valores nulos ao redor da imagem e na extremidade dos eixos X e Y, utiliza-se apenas o zoom de 100% a 130%, observe a Figura 27. O valor máximo é limitado em 130%, pois acima disso já é possível ter grandes cortes no coração.

Agora que é de conhecimento as 3 técnicas utilizadas para aumento de dados, ressalta-se que, para garantir que haja a mesma relação entre o aumento proporcionado

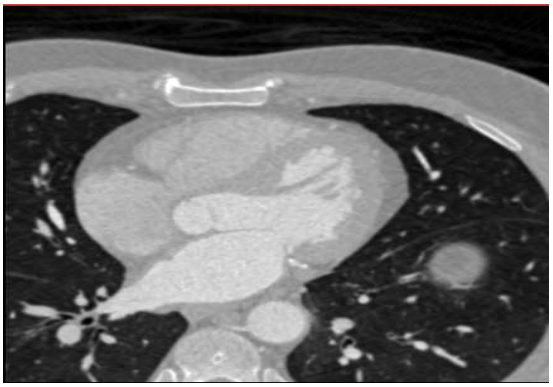


(a) Fatia original.

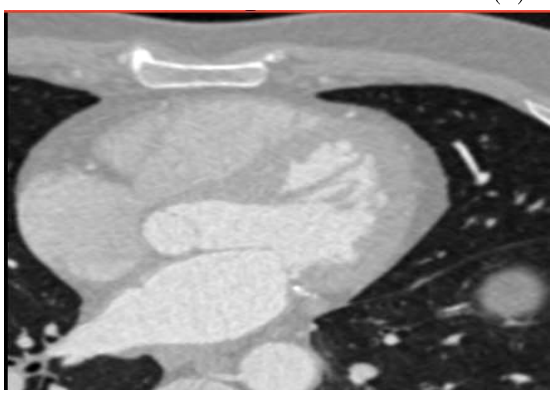


(b) Fatia com cortes laterais de 16 pixeis.

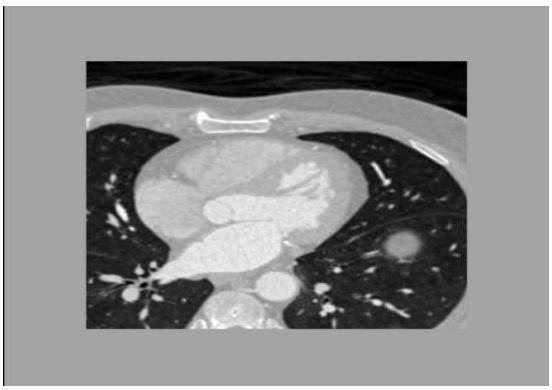
Figura 26 – Exemplo de cortes laterais.



(a) Fatia original.



(b) Fatia com zoom de 130%.



(c) Fatia com zoom de 70%.

Figura 27 – Exemplo de zoom.

à imagem e a sua respectiva consequência à caixa delimitadora, mais precisamente a respeito das etapas de cortes laterais e zoom, estas duas também são aplicadas com mesma intensidade à caixa delimitadora. Para isso, cria-se um tensor de zeros com mesma dimensão da imagem de entrada e, dentro da região apontada pela caixa delimitadora, atribui o valor 1. Após isso, aplica a este tensor o mesmo tipo de aumento de dados sofrido pela imagem original, com exceção do filtro Gaussiano, e, a partir deste resultado, extrai os pontos extremos que haja valor diferente de zero. Estes novos valores serão as coordenadas da caixa delimitadora da imagem proveniente do aumento de dados.

Ressalta-se que o processo de aumento de dados ocorre apenas no treinamento. A fim de otimizar espaço de armazenamento, estas imagens são geradas durante a execução do ajuste do modelo. Por mais que gerar as imagens em toda época seja mais custoso em termos de processamento, é considerado inexequível, para fins de armazenamento, mantê-las armazenadas no disco. Por exemplo, tendo em vista que cada imagem ocupa em torno de 200 Megabytes de espaço em disco, ao explorar todas as opções dos parâmetros utilizados no aumento de dados, exigir-se-ia espaço de armazenamento da ordem de Terabytes de armazenamento de disco a mais.

Após a etapa de aumento de dados, a imagem original sofre um pré processamento relacionado ao seu tamanho. Tendo em vista que, normalmente, as imagens de TC possuem resolução $512 \times 512 \times Z$, em que Z é um valor que varia entre imagens, 512 pixeis nos eixos X e Y pode demandar uma quantidade maior nos tensores resultantes das camadas intermediárias do modelo, ocupando maior espaço de memória. A fim de mitigar tal efeito, realiza-se uma reformulação na imagem de entrada antes de alimentar o modelo. Dessa forma, a imagem passa a ter um tamanho menor, $320 \times 320 \times 192$, escolhido. As vantagens de utilizar tal abordagem são a diminuição do espaço de memória necessário para processar a informação e torna conhecida a quantidade de informações provenientes do eixo Z.

Quanto ao método de implementação, utiliza-se a interpolação. Esta consiste em calcular um valor intermediário entre dois pontos conhecidos. O método de interpolação se dá por funções polinomiais, porém, aqui são utilizadas somente funções polinomiais de primeiro grau. Sendo assim, caso seja desejável reformular o tamanho de um determinado eixo de N para M, interpola-se os N pontos originais, divide este espaço em M partes e utiliza o valor indicado pela interpolação em cada uma destas partes, observe a Figura 28. Ao aplicar esta técnica nos 3 eixos, interpolação trilinear, o resultado final consiste na imagem original modificada para o tamanho desejado.

3.2 Arquitetura do Modelo

Quanto à arquitetura do modelo, este é dividido em 3 partes: *Feature Pyramid Network* (FPN), *Region Proposal Network* (RPN) e Classificação, observe a Figura 29. Cada uma destas partes são sub arquiteturas do modelo ao todo e responsáveis por etapas específicas do processo de detecção. Resumidamente, a FPN é o primeiro processamento da imagem de entrada e se responsabiliza por extrair mapas de recursos, que são os resultados dos filtros aplicados na imagem original. Em seguida, a RPN avalia os mapas de recursos e identifica propostas de região, que são partes da imagem onde há possibilidade de conter o coração. Por último ocorre a etapa de classificação. Esta é alimentada com as propostas de região da RPN e os mapas de recursos extraídos pela FPN. Com base nestas duas informações, a rede de classificação informa, para cada indicação da RPN, um grau de

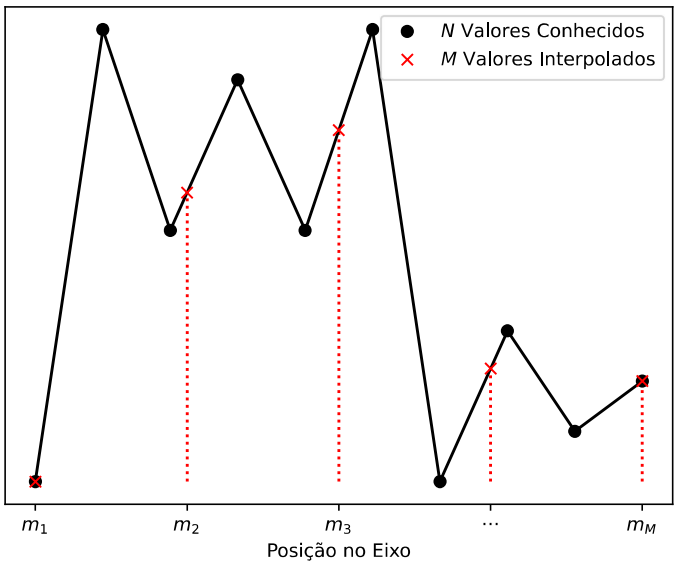


Figura 28 – Representação da interpolação linear.

confiança para aquela região ser o coração e um ajuste necessário para melhor enquadrá-lo. Por fim, o modelo retorna a região, devidamente ajustada, que possui o maior grau de confiança e esta é a saída do modelo.

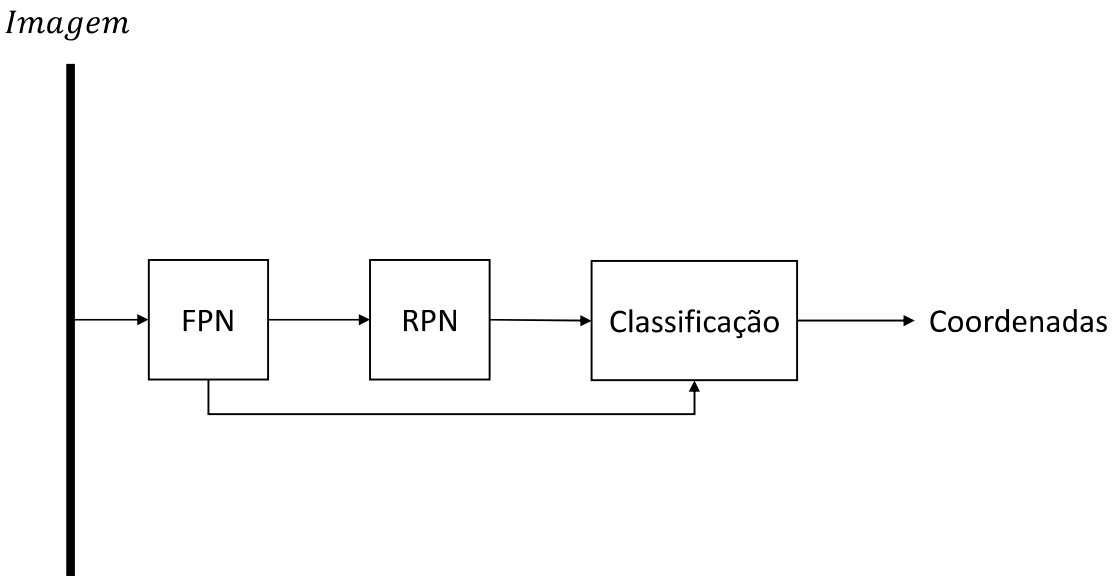


Figura 29 – Arquitetura geral do modelo.

3.2.1 *Feature Pyramid Network*

A FPN é o primeiro processamento da imagem e este consiste em extrair os mapas de recursos dela. Estes mapas de recursos podem ser interpretados como a informação bruta

contida na imagem e o intuito é lapidá-los nas etapas subsequentes. Porém, inicialmente, é necessário que esta informação possua alto grau de relevância dentro do contexto e das dimensões abordadas. Para isso, primeiramente, ressalta-se que este modelo 3D foi baseado no 2D de (REN et al., 2015). Sendo assim, (XU; WU; FENG, 2018) aplicou diversas mudanças para adaptá-lo a imagens de 3D. A primeira delas foi substituir a Coluna Vertebral, termo que define a parte do modelo responsável por extrair recursos, *Residual Network* (ResNet) por uma que processe imagens de 3D.

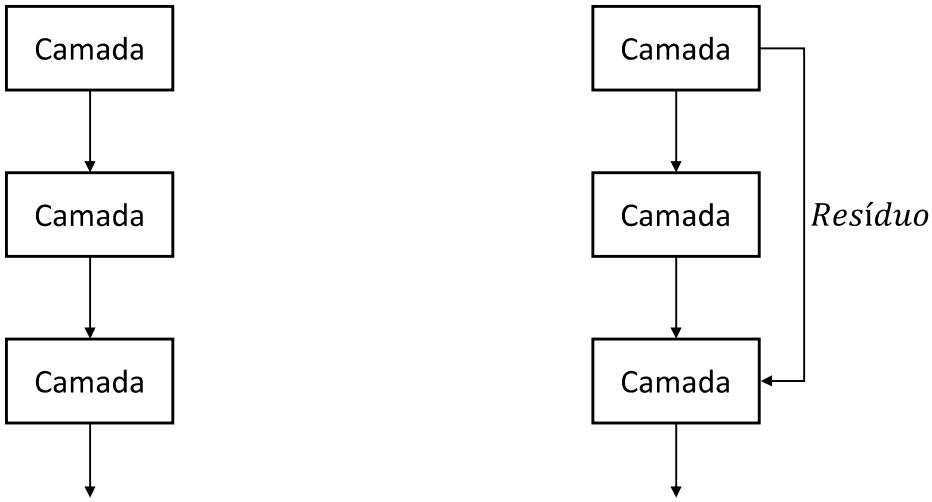
A fim de entender a utilização da ResNet, primeiramente é necessário entender o que é a retropropagação. Esta é utilizada para calcular o gradiente dos parâmetros das camadas a fim de otimizá-los de tal forma que a função de perda seja reduzida (NIELSEN, 2015). Porém, ao aplicar esta técnica em Redes Neurais Profundas ocorre um efeito de desaparecimento do gradiente das camadas mais profundas. Isso ocorre, pois ao calcular

$$\nabla E = \frac{\partial E}{\partial W} \tag{3.2}$$

por meio da regra da cadeia de derivadas parciais, como o ∇E se torna o produto de gradientes de todas as funções de ativação, ao utilizar muitas camadas a tendência é que o gradiente vá a zero. Particularmente, isso ocorre com as derivadas associadas aos parâmetros das camadas próximas da entrada. A consequência disso é a paralisação da atualização dos pesos relacionados às primeiras camadas. Dessa forma, os parâmetros das camadas mais altas não mudam de maneira tão significativa e isso se agrava cada vez mais conforme a rede se torna mais profunda. (HOCHREITER, 1998)

Tendo em vista tal efeito, o intuito da ResNet é, justamente, mitigá-lo. Para isso, a arquitetura do modelo é modificada de tal forma que a informação, ao invés de prosseguir somente à camada seguinte, propaga-se, também, para camadas mais profundas, preservando a informação não processada. Observe a Figura 30. Os autores da ResNet (HE et al., 2016) argumentam que a adição de novas camadas não deveria degradar o desempenho da rede e um modelo mais profundo não deveria produzir resultados inferiores aos modelos mais rasos. Por exemplo, ao introduzir camadas cujo mapeamento equivale à identidade, ou seja, não muda o resultado, isso não deveria acarretar em pior desempenho do que um modelo sem essas camadas no final. Nesse sentido, os autores supõem que um mapeamento de camadas residual é melhor, em termos de perda de informação e desaparecimento do gradiente, do que um mapeamento direto.

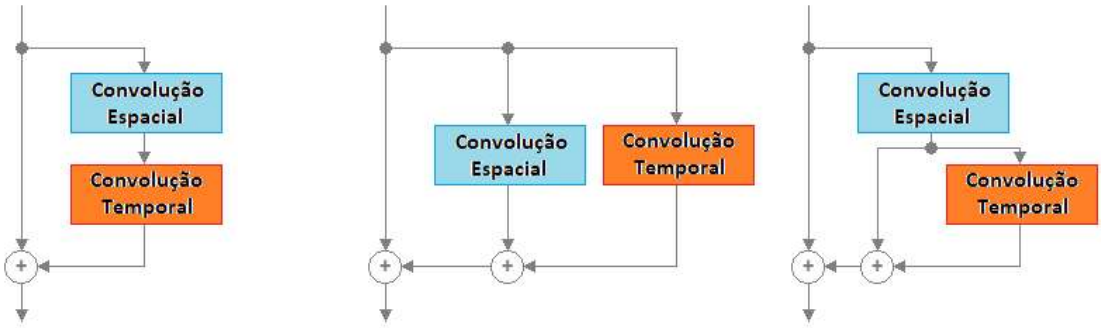
Uma vez que se conhece o efeito da ResNet na arquitetura de 2D, a Coluna Vertebral do *Faster* R-CNN foi modificada por (XU; WU; FENG, 2018) da ResNet para a *Pseudo 3D Residual Network* (P3D ResNet), que apresentou benefícios em analises de vídeos (QIU; YAO; MEI, 2017). Ressalta-se que, tendo em vista o alto custo de processamento e consumo de memória ao aplicar modelos em 3D, a P3D ResNet aplica a arquitetura da rede residual



(a) Mapeamento direto. (b) Mapeamento da ResNet.

Figura 30 – Comparação do fluxo da informação na ResNet e no mapeamento direto.

conforme indicado pelo modelo 2D, porém, ao invés de utilizar convolução $3 \times 3 \times 3$, por exemplo, troca-a por convolução no espaço $1 \times 3 \times 3$ combinada com convolução temporal $3 \times 1 \times 1$, o método de combinação pode ser dado por 3 blocos de construção diferentes, observe a Figura 31. Apesar deste trabalho não envolver variações temporais ao longo do eixo X e sim espaciais, por se tratar de planos anatômicos, a P3D-ResNet introduz ganho de performance e reduz a quantidade de parâmetros treináveis da Coluna Vertebral do modelo.



(a) P3D-A. (b) P3D-B. (c) P3D-C.

Figura 31 – Blocos de construção do P3D

Agora que se conhece a Coluna Vertebral do modelo, é possível apresentar a arquitetura geral desta etapa, que é a FPN. Primeiramente, vale salientar que o intuito de utilizar a extração de recursos por meio de pirâmide é a possibilidade de obter recursos referentes a objetos em diferentes escalas na imagem. Sendo assim, esta arquitetura possui fluxo de informação de cima para baixo com conexões laterais e, com isso, obtém-se mapas de recursos semânticos de alto nível em todas as escalas. A FPN apresenta melhorias significativas ao ser utilizada como um extrator de recursos genérico e flexível. Na Figura 32

é possível observar o fluxo da informação de uma FPN. (LIN et al., 2017)

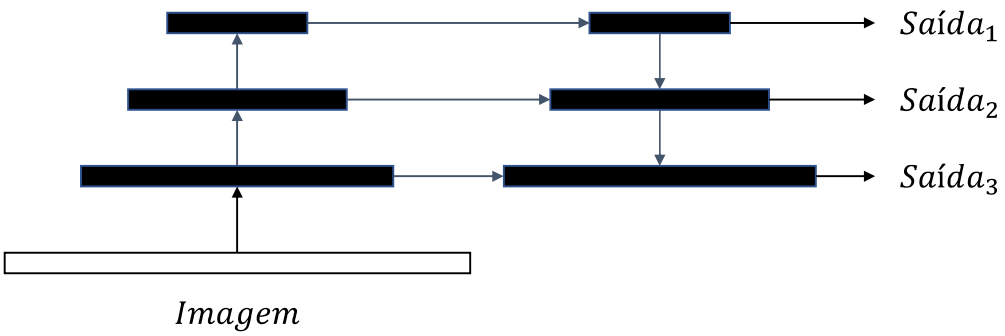


Figura 32 – Fluxo de informação de uma FPN.

Quanto à implementação, utiliza-se a P3D ResNet como o fluxo de subida da FPN. As etapas de subida envolvem, cada uma, uma sequência de camadas, as quais contém: Normalização em Lotes, convolução 3D, convolução espacial e convolução temporal. A subida envolve 3 partes, chamadas de C1, C2 e C3. Na Figura 33 é possível observar a estrutura da FPN, assim como a dimensão do tensor de saída em cada uma das etapas.

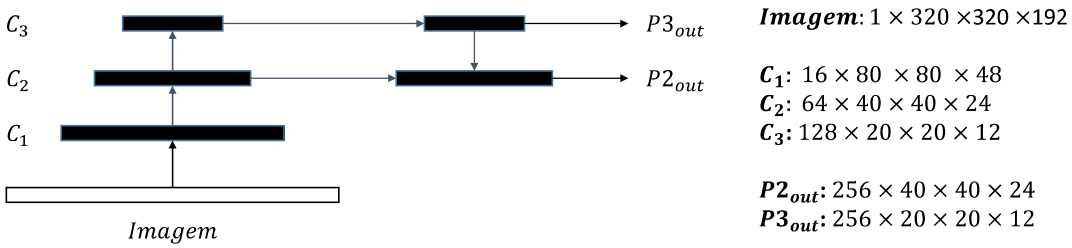
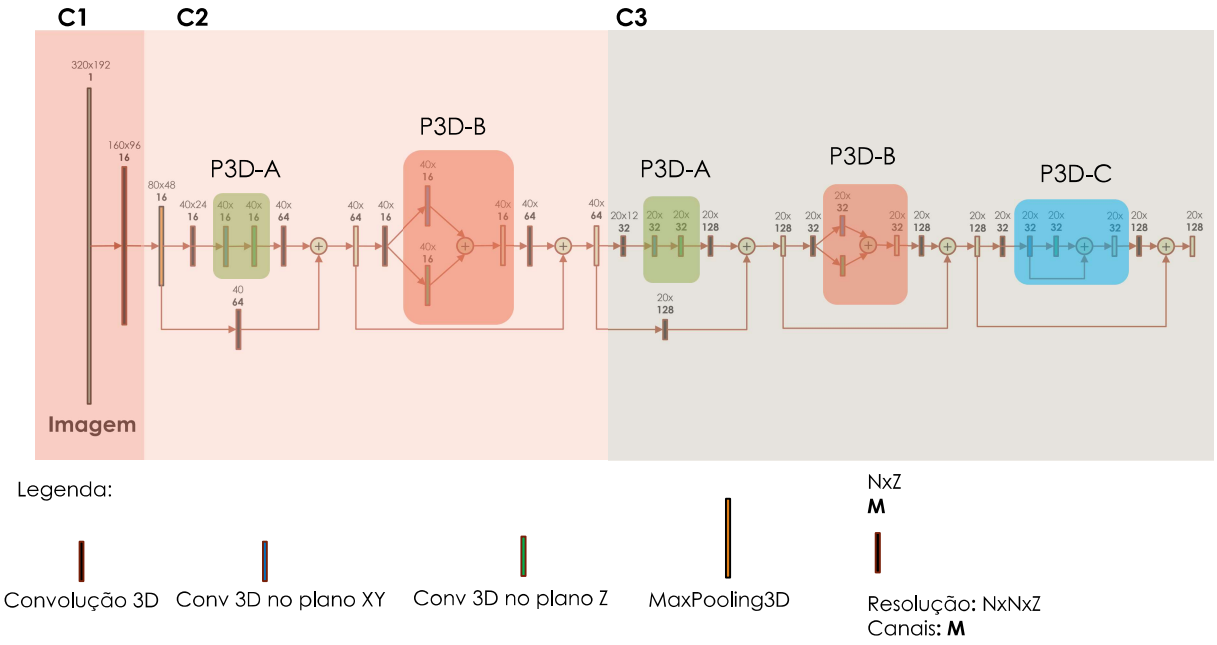


Figura 33 – Representação da FPN implementada.

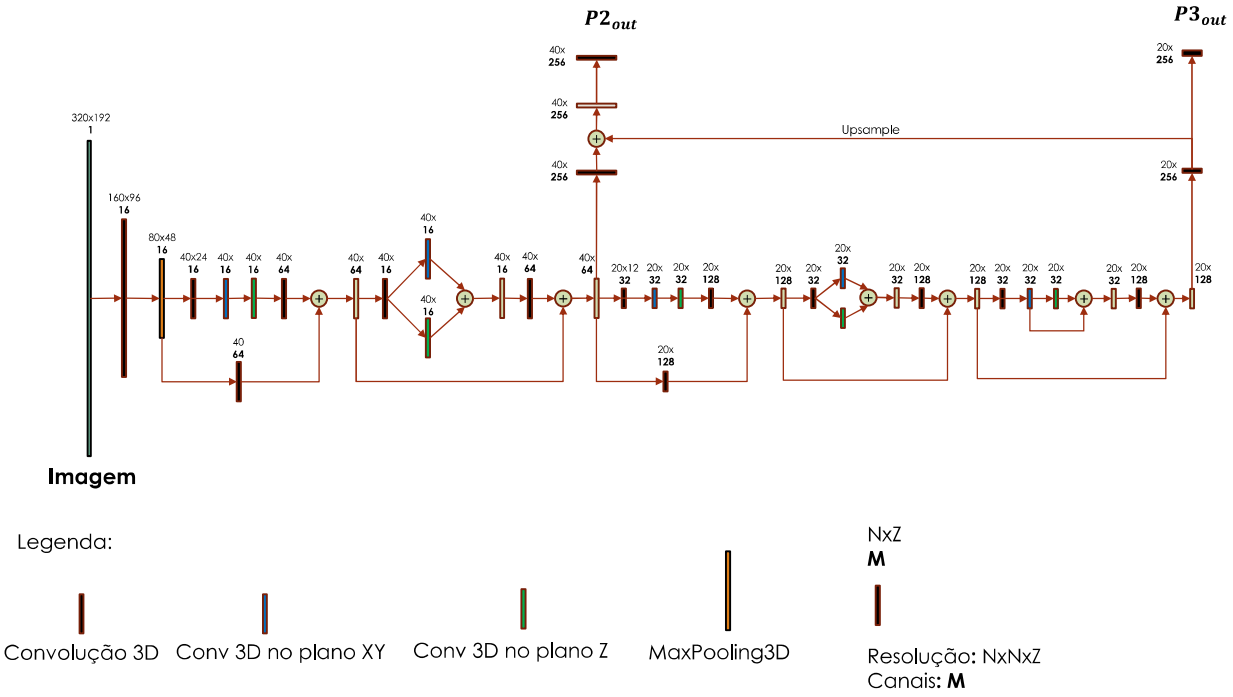
Na Figura 34a é possível observar as camadas internas na subida da FPN, discriminando as etapas da P3D ResNet, e na Figura 34b é possível observar as camadas da descida que originam a saída da FPN. Ao final deste processo, as informações $P2_{out}$ e $P3_{out}$ são os mapas de recursos que serão processados pela RPN e pela etapa de classificação. Vale salientar que o motivo pelo qual estas saídas são susceptíveis a detectar objetos em diferentes escalas é devido a sua dimensão final. Observe na Figura 33 que a resolução de saída da $P2_{out}$ é $256 \times 40 \times 40 \times 24$ e a da $P3_{out}$ é $256 \times 20 \times 20 \times 12$.

3.2.2 *Region Proposal Network*

A próxima etapa na sequência de processamento é a RPN. Esta é responsável por indicar propostas de região, que são partes da imagem cuja probabilidade de conter um objeto é maior. Ressalta-se que nesta etapa não há preocupação em saber se os objetos indicados são o coração ou não, e sim se são bons candidatos para tal.



(a) Camadas na subida da FPN.



(b) Representação completa da FPN.

Figura 34 – Camadas internas da FPN.

A fim de entender o funcionamento da RPN, primeiramente é necessário apresentar as âncoras, que são caixas delimitadoras referentes a regiões igualmente espaçadas ao longo da imagem, cujo propósito é ser uma referência espacial para a predição. Observe a Figura 35.

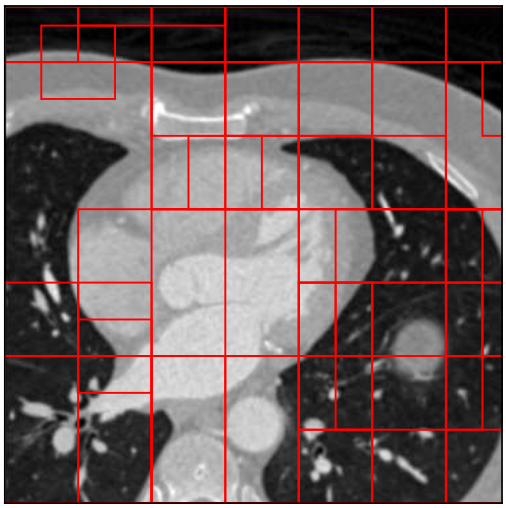
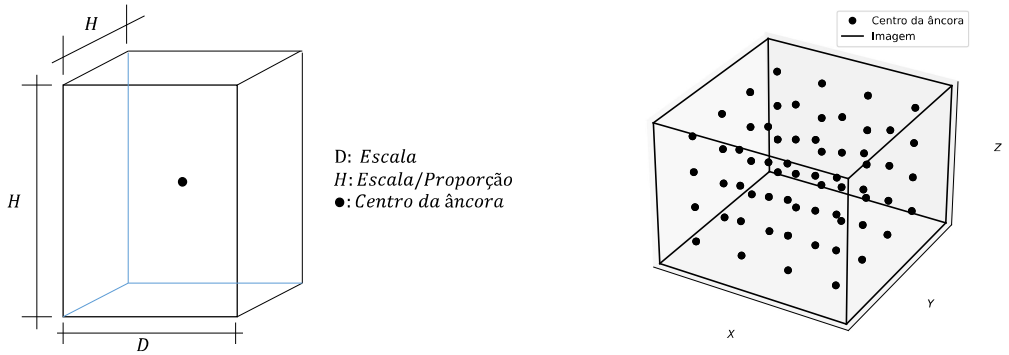


Figura 35 – Exemplos de âncoras na imagem.

A criação das âncoras é baseada em três fatores: escala, proporção e centro. A escala de uma âncora equivale ao comprimento desta ao longo de um eixo escolhido, geralmente se utiliza a largura, a proporção é referente às relações $largura : altura$ e $largura : profundidade$ e o centro se refere à posição a partir da qual o corpo da âncora será construído, observe a Figura 36a. Vale ressaltar que o centro é a referência utilizada para criar o espaçamento entre âncoras, as quais são distanciadas por um valor d de centro a centro ao longo de cada um dos eixos, observe a Figura 36b



(a) Determinação de uma âncora com base no centro, escala e proporção.

(b) Posicionamento dos centros das âncoras ao longo de uma imagem 3D.

Figura 36 – Processo de criação e distribuição de âncoras.

Cada âncora é associada ao mapa de recursos $P2_{out}$ e $P3_{out}$ e, como consequência da sua dimensão de saída, o mapa de recursos $P2_{out}$ é mais sensível a objetos menores

em escala enquanto $P3_{out}$ é mais sensível a objetos maiores. Para respeitar tal critério, são criados dois grupos de âncoras: o primeiro é referente às possibilidades de detecção de $P2_{out}$ e o segundo referente às de $P3_{out}$. Para cada um dos mapas de recursos, seu respectivo grupo de âncoras é criado conforme o seguinte:

1. Os centros das âncoras são gerados com distância d entre si de tal forma que todo o mapa de recurso com dimensão $X' \times Y' \times Z'$ seja compreendido;

2. A largura, altura e profundidade das âncoras, W, H e D, são baseadas nas escalas e proporções de acordo com: $D = escala$, $H = escala/proporção$ e $W = H = escala/proporção$; e

3. As âncoras são multiplicadas por um fator P_{mr}, Passo do Mapa de Recursos, de tal forma que seus valores sejam convertidos para a dimensão da imagem original, cujo valor é $X \times Y \times Z$. Vale salientar que $P_{mr} = X/X' = Y/Y' = Z/Z'$.

Tendo em vista a presença das âncoras ao longo da imagem, e com dimensões condizentes com a saída da FPN, a RPN é responsável em processar os mapas de recursos da etapa anterior e indicar qual deve ser o ajuste feito às âncoras que melhor se ajustaram ao objeto. O critério para definir qual âncora está mais próxima do objeto é dado pelo IoU, cujo valor deve ser, no mínimo, 50%, sendo este um valor padrão definido. A arquitetura da RPN pode ser visualizada na Figura 37. Observe que este modelo apresenta duas saídas RPN_{class} e RPN_{caixa}, cuja dimensão de saída está relacionada com a quantidade de âncoras criadas. Outro fator importante a salientar é a camada *Logits* antes da normalização $Softmax$. Esta camada está discriminada, pois será a saída utilizada na função de perda a fim de minimizar o erro da RPN_{class}.

A respeito da interpretação, a $RPN^{(i)}_{caixa}$ indica ajustes a serem aplicados na $\hat{A}ncora^{(i)}$, a dimensão de saída é $N^o\hat{a}ncoras \times 6$. A conversão para a caixa final $RoI^{(i)}_{RPN} = [x'_0, y'_0, z'_0, x'_1, y'_1, z'_1]$ leva em consideração $RPN^{(i)}_{caixa} = [dz, dy, dx, log(dd), log(dh), log(dw)]$ e a $\hat{A}ncora^{(i)} = [x_0, y_0, z_0, x_1, y_1, z_1]$. Abaixo está a descrição da conversão para as coordenadas referentes ao eixo Z, os eixos X e Y exigem procedimento análogo. Para tal, primeiramente é necessário definir o $centro'$ da caixa final, que será dado por $\Delta Z = z_2 - z_1$ conforme:

$$centro' = z_1 + \frac{1}{2} \cdot \Delta Z + d_z \cdot \Delta Z \tag{3.3}$$

Em posse do centro da caixa, é necessário calcular a profundidade desta. Caso a análise fosse em X ou Y seria a largura ou altura, respectivamente. A profundidade é dada por:

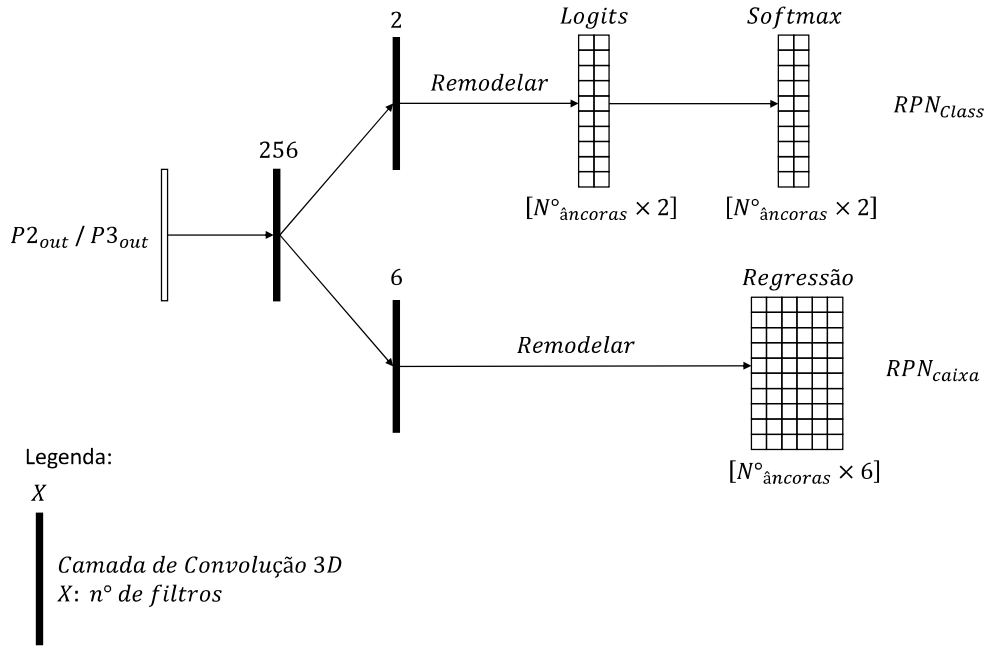


Figura 37 – Representação da arquitetura da RPN.

$$profundidade = \Delta Z \cdot e^{log(dd)} \tag{3.4}$$

Conhecendo o centro da caixa e a profundidade desta, torna-se fácil determinar a delimitação z'_0 e z'_1 conforme:

$$z'_0 = centro' - \frac{1}{2} \cdot profundidade \tag{3.5}$$

$$z'_1 = centro' + \frac{1}{2} \cdot profundidade \tag{3.6}$$

Já $RPN^{(i)}_{class}$ indica se a $\hat{A}ncora^{(i)}$ com os devidos ajustes aponta para um objeto ou não, motivo pelo qual a resolução final é $N^o\hat{a}ncoras \times 2$. Esta saída é processada por uma função de ativação $Softmax$, em que a posição 1 indica a probabilidade de não ser objeto e a 2 de ser um.

Outro fator a ser considerado é a possibilidade de diferentes caixas contidas em RoI_{RPN} apontarem para a mesma região. A fim de evitar tal repetição da mesma informação, utiliza-se a Supressão Não Máxima. O critério de seleção das regiões ocorre de forma decrescente baseado em RPN_{class} e no valor do IoU entre elas. Sendo assim, as caixas delimitadoras indicadas pela Supressão Não Máxima não possuem IoU $\geq L$ entre si, sendo L um valor escolhido de 0.7.

3.2.3 Rede de Classificação

Após indicadas as regiões provenientes da RPN, ainda é necessário que haja um refinamento destas para que o objeto seja adequadamente encapsulado. A necessidade de tal etapa se deve ao fato da RPN ser treinada para indicar, genericamente, objetos, sem preocupação de qual classe este se enquadra. Dessa forma, a rede de classificação consiste em verificar se, de fato, a região indicada pela RPN consiste no coração, assim como ela fornece outro ajuste para a caixa delimitadora a fim de melhor enquadrá-lo. Quanto a esta indicação de qual classe se trata, vale salientar que é necessário que haja um grau de confiança dizendo se a região indicada pela RPN de fato se trata do coração ou do fundo. Resumidamente, a RPN pode ser interpretada como um detector grosseiro e a rede de classificação a utiliza para fazer uma detecção minuciosa no objeto.

Vale salientar que existem duas abordagens distintas para a Rede de Classificação. Uma delas atua no processo de treinamento, em que há necessidade de realizar um pré processamento na saída da RPN a fim de nortear o que se espera da rede de classificação e a outra é referente à inferência, cujo intuito é testar a performance do modelo e, portanto, não há este pré processamento.

Em primeiro lugar, a respeito da etapa de treinamento, o fluxo da informação parte da RPN, que seleciona um número máximo de L regiões com base no RPN_{Class}, aplica RPN_{caixa} e Supressão Não Máxima nas âncoras, normaliza-as com base na dimensão da imagem e o resultado, RPN_{RoIs}, alimenta a *Detection Target Layer* (DTL), que é uma etapa responsável em filtrar os RoIs que de fato são de interesse da rede de classificação, indica os *deltas*, ajustes da caixa delimitadora, esperados que a rede de classificação retorne e os RoI_{Class}, também desejados como saída da rede de classificação. O primeiro valor retornado serve para alimentar a rede de classificação e seguir com o fluxo de treinamento, pois não é interessante que esta etapa do modelo seja treinada com regiões ruins indicadas pela RPN. Ressalta-se que essa remoção de regiões ruins atua principalmente no início do treinamento, quando a RPN ainda não é capaz de indicar bons candidatos. Os demais valores retornados pela DTL servem para ser aplicados posteriormente na função de perda. Vale salientar que este modelo não é treinado de ponta a ponta justamente devido ao fato da rede de classificação não poder atuar até que a RPN possua bons candidatos para prosseguir. O fluxograma desta etapa pode ser visualizado na Figura 38.

Quanto à DTL, esta, inicialmente, calcula o IoU das RPN_{RoIs} em relação à caixa verdadeira. Em seguida, verifica quais das RPN_{RoIs} tiveram um IoU ≥ 0.5. Caso haja, salva as n^+ RPN_{RoIs} que tiveram o maior valor de IoU e as define como $RoIs^+$. Para estas, calcula o *delta* necessário para que preencham corretamente o coração. Além disso, ressalta-se que $n^+ = n^o_{RoIs} \cdot R_{\pm}$, em que n^o_{RoIs} é a quantidade de regiões que se deseja passar para a classificação e $R_{\pm}$ é uma relação de regiões positivas e negativas. É importante que haja essa relação, pois as regiões positivas são responsáveis em treinar a rede de

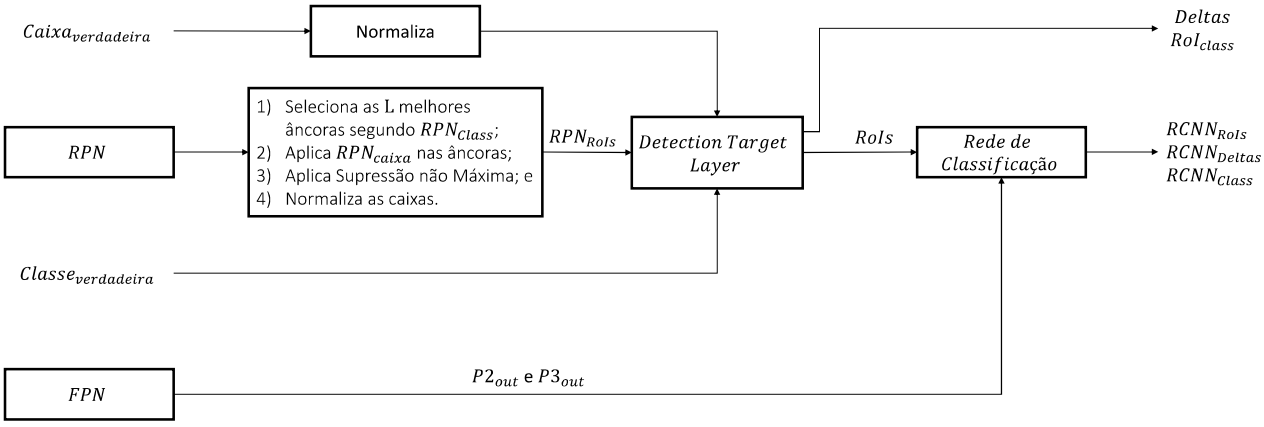


Figura 38 – Fluxograma da etapa de treinamento da rede de classificação.

classificação para aquilo que realmente é o coração e as regiões negativas servem para treiná-la a identificar o que é fundo. Por definição, $n^o_{RoIs} = 15$ e $R_{\pm} = 1/3$.

Em seguida, ainda na DTL, define-se as $RoIs^-$, que são as RPN_{RoIs} cujo IoU em relação à caixa delimitadora verdadeira seja menor que 0.5. Ou seja, não representam o coração. Caso haja $RoIs^+$ e $RoIs^-$, a quantidade de $RoIs^-$ é ajustada a fim de respeitar a relação $n^- = n^+ \cdot (R_{\pm}^{-1} - 1)$. Além disso, o valor de RoI_{class} para essas regiões é zero, indicando fundo, assim como o valor de RoI_{deltas} é um vetor de zeros, com formato $\mathbb{R}^6$, indicando que não há ajuste para essas regiões, tendo em vista que não são o coração.

É importante ressaltar que a etapa de definição das regiões negativas somente ocorre se houver regiões positivas, pois não é adequado treinar a rede de classificação com amostras cujos valores dos resultados desejados são todos 0, classificação e ajuste. Ao contrário do que ocorre caso haja somente regiões positivas, situação a qual é possível treinar o ajuste da caixa delimitadora. O fluxograma da DTL pode ser visualizado na Figura 39.

Vale ressaltar que há outras maneiras de implementar a DTL. Originalmente, (XU; WU; FENG, 2018) seleciona as n^+ e n^- regiões aleatoriamente. Ou seja, as $RoIs^+$ podem ser quaisquer das RPN_{RoIs} que obtiveram IoU ≥ 0.5 e não necessariamente as melhores entre essas. O mesmo vale para $RoIs^-$. Um fator crítico para estas abordagens é que $n^o_{RoIs} \leq RPN_{RoIs}$, ou seja, a DTL faz uma pré-seleção nos candidatos que serão repassados à Rede de Classificação e isso introduz um viés ao modelo. Tal viés ocorre, pois este pré-processamento é impossível de ser executado na inferência, em que é assumido que não se conhece a caixa verdadeira.

A fim de avaliar o efeito deste viés, foi desenvolvido nesta dissertação um terceiro tipo da DTL. Desta vez, esta etapa não seleciona um grupo específico de candidatos para passar adiante, passa todos. O único efeito desta camada, neste caso, é determinar os objetivos esperados como predição da Rede de Classificação, pois são necessários para

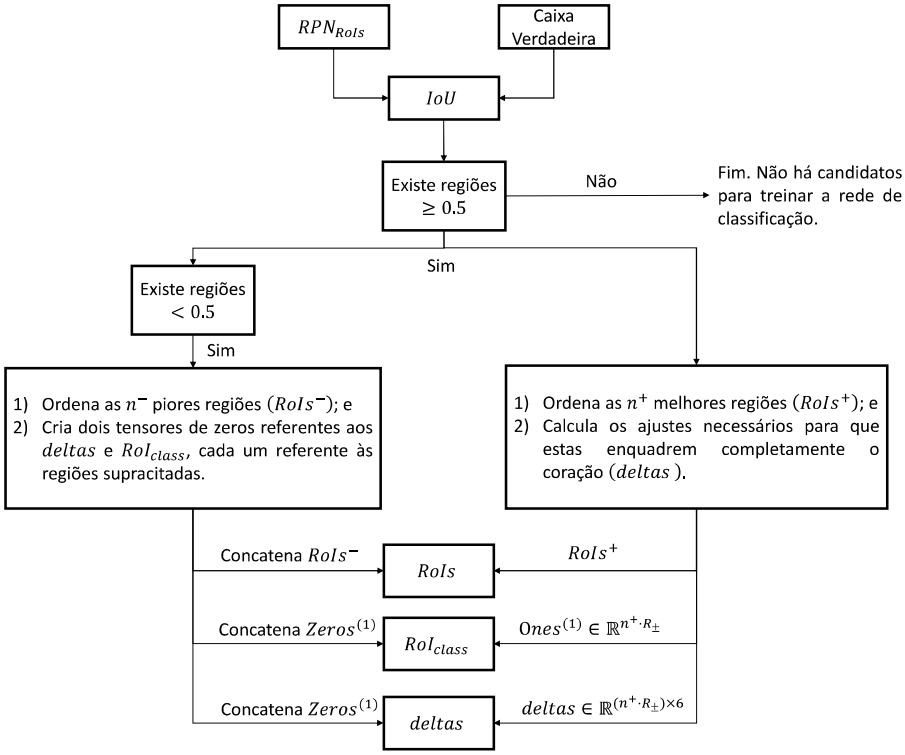


Figura 39 – Fluxograma da DTL.

Nota: (1) Vetor que contenha apenas 0 ou 1.

a função de perda. Como critério de bons candidatos, são definidas as 1/3 melhores RPN_{RoIs}, com base no IoU, como $classe = 1$, coração, e o restante como $classe = 0$. Para estas RPN_{RoIs} com a classe indicando que é coração, determina-se o *delta* esperado para minimizar o erro da caixa delimitadora. Para as outras, *delta* é um vetor de zeros.

Dessa forma, após a DTL, as regiões filtradas por ela alimentam a rede de classificação junto com as saídas da FPN para, no final, definir um ajuste fino a elas e indicar se de fato são o coração. A respeito do fluxo deste processamento, primeiramente passam por uma etapa chamada de *Pyramid RoI Align*, que serve para destacar, no mapa de recursos, somente a região apontada pelo RoI e esta sofre uma reformulação de tamanho para o formato $7 \times 7 \times 7$, observe a Figura 40. Vale ressaltar que a reformulação ocorre por meio da interpolação trilinear da região apontada pelo RoI no mapa de recursos.

Em seguida, os $RoIs$ alinhados são processados por duas convoluções e este resultado em seguida é processado por camadas totalmente conectadas a fim de obter a classificação final, $RCNN_{Class}$, e os respectivos ajustes do RoI, $RCNN_{Deltas}$. A arquitetura da Rede de Classificação pode ser visualizada na Figura 41. Vale ressaltar que a quantidade de $RoIs$ é desconhecida de antemão, diferentemente ao que ocorre com a RPN, cuja quantidade de âncoras é conhecida. Outro fator a ser levado em consideração é que a etapa "Remodelar", subsequente às convoluções, não modifica o tamanho da saída daquelas camadas, apenas a transforma para um formato matricial para ser processada pelas Camadas Totalmente

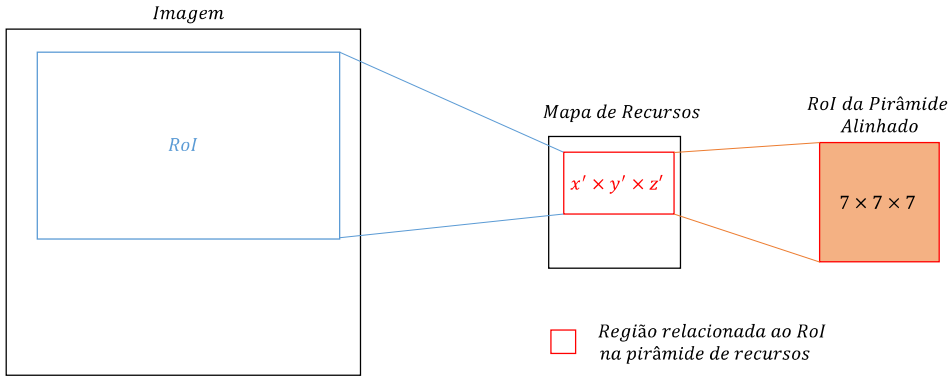


Figura 40 – Processo de reformulação do *RoI* no mapa de recursos.

Conectadas. Além disso, assim como a RPN, a camada *Logits* é a utilizada para calcular a função de perda referente a esta etapa de classificação, não a *Softmax*.

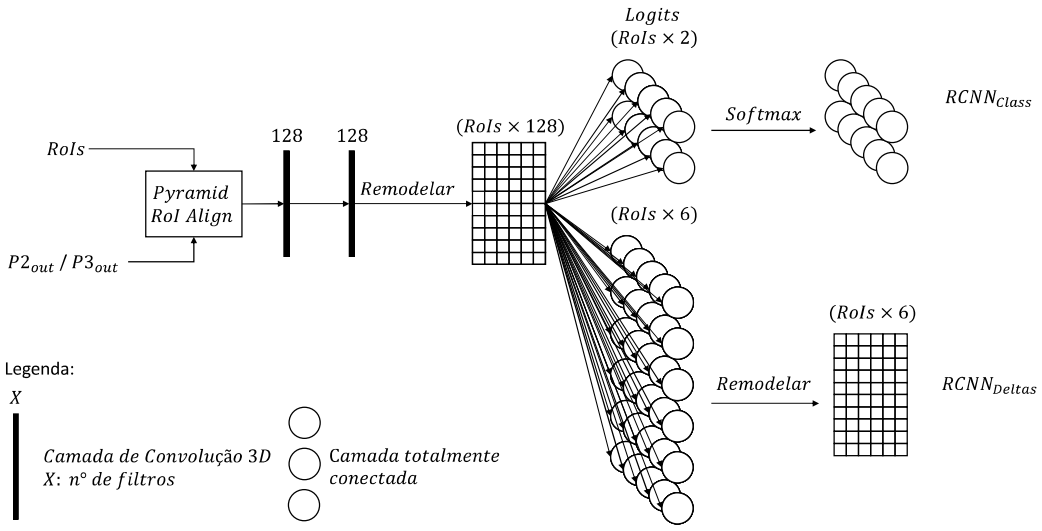


Figura 41 – Arquitetura da Rede de Classificação.

Quanto à etapa de inferência, como não se deseja obter os valores de *deltas* e RoI_{class}, não há necessidade de processar os dados pela DTL. Sendo assim, a informação proveniente da RPN segue diretamente à rede de classificação. Observe o fluxograma na Figura 42.

Todavia, observa-se que nesta etapa há a presença da Camada de Detecção. Esta, por sua vez, é responsável em selecionar a melhor região candidata a ser o coração e indica a caixa delimitadora final. Para tal, esta etapa aplica os $RCNN_{Deltas}$ nos $RoIs$, mesmo modo de aplicação que a RPN_{Deltas} para as Âncoras, e, tendo em vista que são valores normalizados, multiplica-os pela largura, altura e profundidade da imagem pós processada, assim como limita seus valores para os valores mínimos e máximos de cada uma destas dimensões. O resultado desta sequência de operações são as caixas delimitadoras

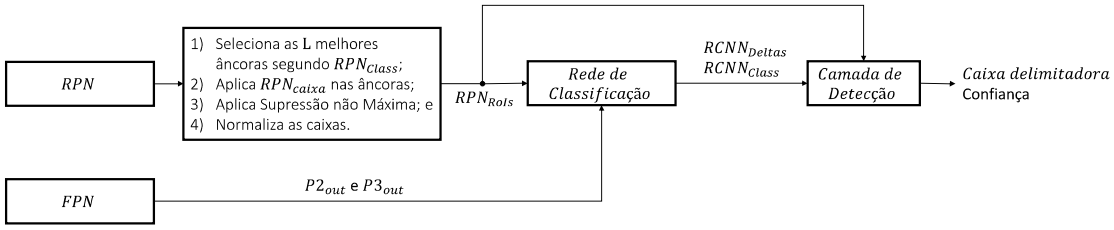


Figura 42 – Fluxograma da etapa de inferência da rede de classificação.

do coração. Em seguida, seleciona a caixa delimitadora cuja confiança de ser o coração é maior entre todas. Por fim, esta camada retorna a caixa delimitadora do coração e o grau de confiança para tal. O fluxograma da camada de detecção pode ser visualizado na Figura 43.

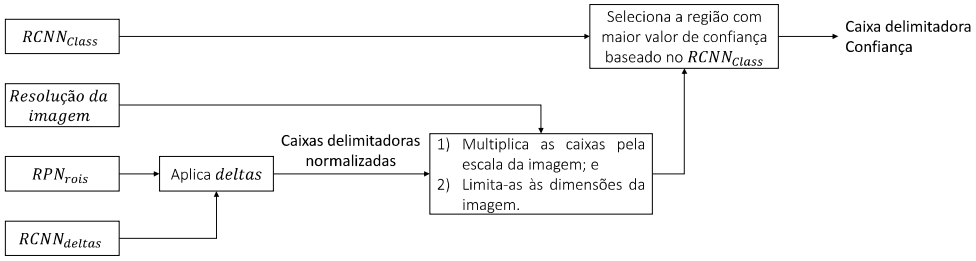


Figura 43 – Fluxograma da camada de detecção.

Porém, quanto à etapa de inferência, um último passo de pós processamento é necessário. Tendo em vista que, inicialmente, a imagem foi reduzida de $512 \times 512 \times Z$ para $320 \times 320 \times 192$ é necessário converter os valores extraídos como caixa delimitadora para contemplarem a resolução original da imagem. Para isso, aplica-se:

$$x_0 = x_0 \cdot \frac{512}{320} \text{ e } x_1 = x_1 \cdot \frac{512}{320} \tag{3.7}$$

$$y_0 = y_0 \cdot \frac{512}{320} \text{ e } y_1 = y_1 \cdot \frac{512}{320} \tag{3.8}$$

$$z_0 = z_0 \cdot \frac{Z}{192} \text{ e } z_1 = z_1 \cdot \frac{Z}{192} \tag{3.9}$$

3.3 Métricas de Avaliação

Tendo em vista que já se conhece a arquitetura do modelo, é necessário apresentar o método de ajuste deste, assim como as métricas que avaliam a precisão. As métricas referentes ao ajuste dizem respeito à abordagem para minimizar a função de perda, que faz com que o modelo se aproxime do resultado desejado e indicam o caminho para tal.

Quanto às métricas de precisão, estas indicam o grau de proximidade do modelo com aquilo que se deseja.

3.3.1 Medidas de Precisão

Inicialmente, vale salientar que, apesar de já haver uma métrica relacionada à proximidade do modelo em relação aos resultados esperados, que é a função de perda, esta não necessariamente está alinhada com aquilo que de fato se espera como resultado desse. Sendo assim, são utilizadas duas métricas para avaliar o quão próximo a caixa delimitadora predita p está em relação à marcação y do especialista: IoU e Falso Negativo.

Primeiramente, o IoU indica o quão próximo o espaço volumétrico da predição está em relação à marcação. Além disso, tendo em vista que seu cálculo leva em consideração a área somada dos dois, esta métrica tende a possuir valores ruins quando o tamanho de um é significativamente maior em relação ao outro. Ou seja, indica imprecisão, mesmo que a predição tenha englobado todo o coração, porém o fez de forma exagerada. É válido ressaltar que o IoU está limitado entre 0 e 1, sendo que quanto mais próximo de 1, melhor é. A métrica pode ser visualizada na Figura 44 e é dada pela expressão Equação 3.10, sendo que p e y são vistos como o conjunto de pixeis da imagem compreendidos pela caixas delimitadoras preditas e verdadeiras:

$$IoU = \frac{|p \cap y|}{|p \cup y|} \tag{3.10}$$

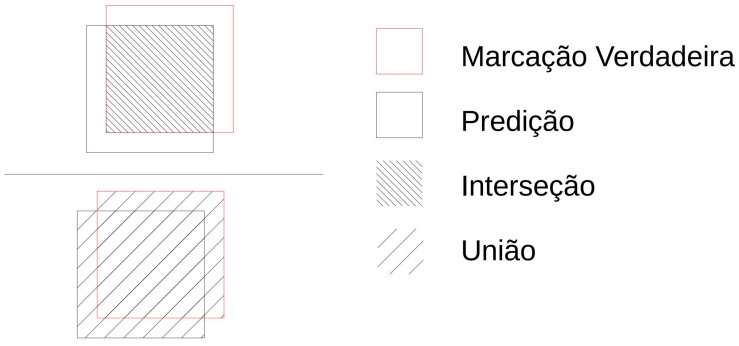


Figura 44 – Representação do IoU.

Entretanto, apesar do IoU ser uma boa métrica para indicar a proximidade entre as duas regiões, nos casos em que a predição corta parte do coração, o IoU ainda pode ser um valor alto, porém ignora o fato da predição ter descartado informações que podem ser cruciais para as técnicas subsequentes a este estudo, as quais foram introduzidas na justificativa, seção 1.2. Dessa forma, outra medida é utilizada para tal que é o Falso Negativo. Esta, por sua vez, informa o percentual do coração que foi cortado da predição, observe a representação da Figura 45. Sendo assim, seu valor também é limitado entre 0 e

1, sendo que, quanto mais próximo de zero, melhor. O cálculo do Falso Negativo é dado por:

$$FalsoNegativo = \frac{|y| - |p \cap y|}{|y|} \tag{3.11}$$

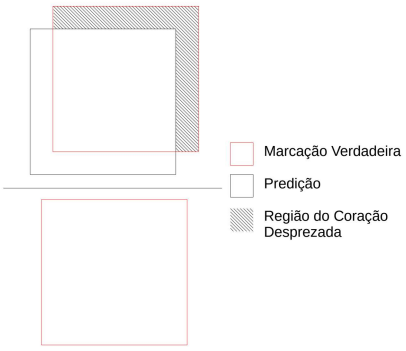


Figura 45 – Representação da região da imagem atribuída como Falso negativo.

É válido salientar que, caso a predição seja dada de maneira exagerada e englobe grande parte da imagem, ou seja, com alto índice de falso positivo, área que não é coração marcada como tal, o Falso Negativo não tem sensibilidade e indicaria valor perfeito, zero. Dessa forma, tanto o IoU quanto o Falso negativo se complementam e precisam ser avaliados em conjunto.

3.3.2 Função de Perda

Quanto ao ajuste, este é feito com base na função de perda, que é dividida em 5 partes: 2 dedicadas em ajustar a RPN, 2 dedicadas à Rede de Classificação e, a última, responsável pela otimização do Falso Negativo. A respeito de cada uma, as 2 da RPN visam minimizar a RPN_{logits} e a RPN_{Deltas}, as referentes à Rede de Classificação, minimizam a $RCNN_{logits}$ e $RCNN_{Deltas}$, e o Falso Negativo é uma das métricas de precisão, cujo intuito é avaliar qual o percentual do coração que foi classificado erroneamente como não coração. Quanto a este, utilizá-lo na função de perda garante que o modelo atribui uma penalidade aos casos em que a caixa predita corta uma parte da região do coração.

A respeito da RPN, RPN_{logits} é treinado utilizando $RPN_{match} \in \mathbb{R}^{\hat{A}ncoras \times 2}$, que é definido como as Âncoras, sem ajuste, que mais se aproximaram do coração. Sendo assim, caso uma $\hat{A}ncora^{(i)}$ tenha IoU ≥ 0.7, então $RPN^{(i)}_{match} = [0, 1]$, senão $RPN^{(i)}_{match} = [1, 0]$. Caso não haja Âncoras, cujo critério de proximidade satisfaça a condição IoU ≥ 0.7, $RPN^{(i)}_{match} = [0, 1]$ é atribuído àquela com maior grau de IoU. Com isso, a RPN_{logits} é treinada de tal forma que o objetivo é indicar qual é a melhor Âncora para ser ajustada ao coração.

Quanto à função utilizada para minimizar tal diferença, utiliza-se a Entropia Cruzada. Esta, por sua vez, é indicada quando as saídas de uma rede representam hipóteses independentes e cada neurônio indica a probabilidade de uma hipótese ser verdadeira, ao passo que, para tal, a outra deve ser falsa. A função de perda da entropia cruzada (L_{ec}) é dada por

$$L_{ec} = -\sum_{i} y_i \cdot log(p_i) \tag{3.12}$$

Em que y_i é o valor real daquela instância e p_i o valor predito pelo modelo.

Sobre a RPN_{Deltas}, utiliza-se a função de perda *Smooth L1* (L_{Smooth}), desenvolvida por (GIRSHICK, 2015). Segundo os autores, esta é uma função de perda L1, $L_1 = \sum |y-p|$, menos sensitiva a valores atípicos quando comparada à função L2, $L_2 = \sum (y-p)^2$, utilizada em (GIRSHICK et al., 2014). Ressalta-se que quando os valores utilizados como objetivo não possuem limite superior, treinar o modelo com a função L2 exige um ajuste cuidadoso das taxas de aprendizado a fim de evitar gradientes explosivos, vulnerabilidade a qual a *Smooth L1* é menos susceptível. Esta função é definida por

$$L_{Smooth}(y^{(i)}, p^{(i)}) = \sum_{j \in \{x,y,z,dw,dh,dd\}} SmoothL1(y_j^{(i)} - p_j^{(i)}) \tag{3.13}$$

Em que $y^{(i)}$ são os ajustes necessários para fazer com que a $\hat{A}ncora^{(i)}$ englobe corretamente o coração, $p^{(i)} = RPN_{Deltas}^{(i)}$ e

$$SmoothL1(x) = \begin{cases} 0.5 \cdot x^2/\beta, & \text{se } |x| < \beta \\ x - 0.5 \cdot \beta, & \text{caso contrário.} \end{cases} \tag{3.14}$$

A respeito das perdas relacionadas à Rede de Classificação, $RCNN_{logits}$ também utiliza a função de perda Entropia Cruzada, Equação 3.12, cujos valores passados como argumento são: y_i são os RoI_{Class}, extraídos da DTL, e p_i é a saída $RCNN_{Logits}$. Ainda, $RCNN_{Deltas}$ utiliza a função de perda *Smooth L1*, Equação 3.13, em que $y^{(i)}$, neste caso, é o ajuste necessário para fazer com que $RoI^{(i)}$, indicado pela RPN, englobe o coração e $p^{(i)}$ é $deltas^{(i)}$, também extraído na DTL. Entretanto, para este último, o processo de cálculo da função de perda ocorre apenas para os $RoIs$ relacionados ao coração. Os que apontam para o fundo retornam diretamente zero.

Quanto à perda associada ao Falso Negativo, tendo em vista que o intuito desta é fazer com que a predição final seja penalizada caso exclua uma região da caixa delimitadora que realmente seja o coração, realiza-se uma predição do modelo, determina a caixa delimitadora final por meio do processo de inferência da Rede de Classificação e, em posse do resultado desta, calcula o Falso Negativo conforme Equação 3.11

Por fim, são somadas todas as 5 parcelas citadas acima, das quais, para *Smooth L1* $\beta = 1$, valor escolhido, e, juntas, definem a função de perda do modelo. Um fator importante a ressaltar é que cada uma possui um peso associado, que também é um hiperparâmetro, os quais podem ser visualizados na Tabela 4.

Tabela 4 – Pesos associados às parcelas da função de perda.

Parcela	**Peso**
RPN_{Class}	100
RPN_{Caixa}	50
$RCNN_{Class}$	500
$RCNN_{Caixa}$	500
$FalsoNegativo$	300

4 Resultados Obtidos

Para avaliar de maneira crítica o desempenho do modelo, foram realizados treinamentos com diferentes hiper-parâmetros e configurações de arquitetura. Primeiramente, avaliamos o modelo com as imagens originais do *Multi-Modality Whole Heart Segmentation* (MMWHS), pois exigiu mínimas alterações do código, o que permite observar a performance original do mesmo. Tais resultados podem ser visualizados nas tabelas abaixo. Ressalta-se que conforme foram ocorrendo modificações no código, sejam de arquitetura ou hiperparâmetros, estas foram registradas por meio de sua Identificação (ID). Sendo assim, a ID indica o versionamento do código conforme houve avanço no estudo.

Todos os testes e execuções foram implementados com as seguintes ferramentas:

- Python versão 3.10.5;
- PyTorch versão 1.11.0+cu113;
- Numpy versão 1.23.1; e
- GPU Tesla V100-SXM2 32Gb, acessada pelo Supercomputador Santos Dumont no LNCC.

Tabela 5 – Configurações dos modelos treinados com as imagens do MMWHS.

ID	Mask[2]	DTL[5]	Aumento de Dados	Tamanho do Lote	FN na Função de Perda	+5% na caixa[4]	Épocas[3]
2	Não	Aleatório	Rotação	1	Não	Sim	670
4[6]	Sim	Aleatório	Rotação	1	Não	Sim	515
14	Não	Ordenado	(1)	1	Sim	Não	835
21	Não	Ordenado	(1)	2	Sim	Não	425
35	Não	Sem intervir	Rotação	1	Não	Sim	655

Notas: (1) - Filtro Gaussiano, Corte Lateral e Zoom; (2) - Utilizou rede de segmentação em série com a detecção; (3) - Todos os modelos foram treinados com tempo máximo de 90 horas; (4) - Os lados das caixas delimitadoras verdadeiras foram aumentados em 5% conforme o artigo original; (5) - *Detection Target Layer* (subseção 3.2.3); e (6) - Configuração original do artigo (XU; WU; FENG, 2018).

Na Tabela 6 pode ser visualizada a precisão do modelo de acordo com o modo de execução, "Treinamento" ou "Teste". O valor referente à *Region Proposal Network* (RPN) indica o *Intersection Over Union* (IoU) da melhor *Region of Interest* indicada por esta Rede. Já os valores referentes à Rede de Classificação (RC) indicam o IoU da caixa delimitadora final do modelo. Em ambas as etapas, pode-se observar o Falso Negativo (FN)

da caixa delimitadora. Adicionalmente, ressalta-se que os valores nas colunas são a média de todas as imagens que compuseram aquele respectivo grupo, "Treino" e "Validação" (Val.).

Tabela 6 – IoU e FN dos modelos treinados com as imagens do MMWHS.

ID	Treinamento						Teste			
	RPN		RC		FN		IoU		FN	
	Treino	Val.	Treino	Val.	Treino	Val.	Treino	Val.	Treino	Val.
2	0.85	0.89	0.92	0.88	NA	NA	0.78	0.77	0.00	**0.03**
4	0.88	0.83	0.91	0.86	NA	NA	0.79	0.75	0.00	0.04
14	0.88	0.83	0.89	0.83	0.04	0.05	0.77	0.72	0.08	0.11
21	0.82	0.78	0.89	0.88	0.07	0.07	0.80	**0.78**	0.11	0.13
35	0.85	0.78	0.77	0.74	NA	NA	0.50	0.55	0.33	0.19

Nota: NA - Não Aplicado.

Uma vez que é conhecido o modo de convergência do modelo, assim como a precisão final obtida com as entradas originais, é possível realizar maiores modificações. Tais modificações englobam a adaptação do modo de abertura das imagens, devido ao formato das imagens do *Hemodynamics Modeling Laboratory* (HeMoLab) ser diferente do formato das imagens utilizadas no MMWHS. Vale ressaltar que tal modificação, embora simples, influencia outras partes do código original de (XU; WU; FENG, 2018). Portanto, exigiu um esforço considerável até estar em pleno funcionamento. Ressalta-se os seguintes pontos de dificuldade encontrados de acordo com as modificações elaboradas:

1. Os eixos X, Y e Z podem ser invertidos de acordo com a biblioteca de leitura das imagens;

2. Caso haja essa inversão, deve-se avaliar se há consequências para a alimentação da imagem na rede, assim como a desnormalização da caixa delimitadora no final do processo;

3. Como o código original extraía a caixa delimitadora de um arquivo que possuía a segmentação de algumas estruturas cardíacas, fez-se necessário modificar essa obtenção para a caixa delimitadora ser obtida por meio de um arquivo $json$;

4. Originalmente, o modelo possuía uma rede de segmentação. Como não é desejável esta parte, é necessário eliminá-la, mas de maneira cautelosa a fim de não excluir uma etapa direcionada à segmentação que também atue em outras partes do modelo; e

5. Vale ressaltar que o cálculo do FN foi uma implementação proveniente deste trabalho. Sendo assim, deve-se verificar se o cálculo está sendo executado conforme planejado e se não interfere em outras etapas do modelo.

Tabela 7 – Configurações dos modelos treinados com as imagens do HeMoLab.

ID	DTL	Aumento de Dados	Tamanho do Lote	Função de Ativação	Treinar BN[4]	Épocas[3]
19	Ordenado	(1)	1	*ReLU*	Não	290
20	Ordenado	(1)	2	*ReLU*	Não	215
22	Ordenado	(1)	4	*ReLU*	Não	405
23	Ordenado	(1)	2	*ReLU*	Não	390
24	Ordenado	(1)	1	*ReLU*	Não	150
25	Sem Intervir	(1)	1	*ReLU*	Não	150
26	Sem Intervir	(1)	1	*ReLU*	Não	150
27	Sem Intervir	(1)	1	*ReLU*	Sim	350
28A	Sem Intervir	(1)	1	*PReLU*	Sim	100
28B	Sem Intervir	(1)	1	*PReLU*	Sim	100
32	Sem Intervir	(1)	6	*ReLU*	Sim	100
33	Sem Intervir	*NA*	1	*ReLU*	Sim	100
29	Aleatório	(1)	1	*ReLU*	Não	350
30	Aleatório	*NA*	1	*ReLU*	Não	350
34	Aleatório	*NA*	1	*PReLU*	Não	350
31	Aleatório	(2)	1	*PReLU*	Não	350
36	Aleatório	*NA*	1	*PReLU*	Não	350

Notas: (1) - Filtro Gaussiano, Corte Lateral e Zoom; (2) - Corte Lateral e Zoom; *NA* - Não Aplicado; (3) - Todos os modelos foram treinados com tempo máximo de 90 horas; e (*BN*) - *Batch Normalization*.

Observações acerca dos treinamentos contidos na Tabela 7:

ID 23: Foi trocada a quantidade de RPN_{RoIs} após a Supressão não Máxima da RPN 64 para 500 e a quantidade de $RoIs$ que sai da DTL passa 15 para 500. O intuito era minimizar o efeito do viés aplicado pela DTL ao repassar todas as imagens da RPN para a classificação;

ID 24: Foi modificado o retorno da DTL para retornar o mesmo número de $RoIs$ que RPN_{RoIs}, pois foi observado que o modelo 23 não garantia que todas as RPN_{RoIs} seguiam para a Rede de Classificação. Agora, essa função classifica até 1/3 das melhores RPN_{RoIs} como positivas, porém somente as que possuem $IoU \geq 0.5$, ou a 1ª melhor, caso não atendam ao critério, e o restante como negativas. Além disso, a quantidade de épocas foi limitada a 150, pois a partir daí o gráfico da precisão se estabilizou;

ID 25: Neste treinamento foi implementada a DTL que nao interfere no fluxo das RPN_{RoIS}, apenas cria os objetivos da rede de classificação. Agora, a DTL define as 1/3 melhores RPN_{RoIs} como coração e calcula seus respectivos deltas;

ID 26: Consiste na continuação do treinamento 23 por mais 150 épocas;

ID 28A: A função de ativação *ReLU* foi substituída pela *PReLU*. Obervação: Devido a uma manutenção no Supercomputador Santos Dumont, o treinamento foi interrompido, e, quando retornou, utilizou um modelo que continha a função *ReLU* ao invés da *PReLU*. Sendo assim, este modelo começou o treino utilizando *PReLU* e finalizou com a *ReLU*;

ID 28B: O treinamento 28 foi refeito, porém sem a interferência da modificação da função de ativação; e

ID 36: Este modelo foi treinado utilizando um acréscimo de 5% na caixa delimitadora verdadeira.

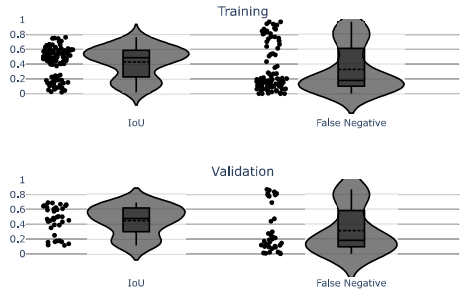
Tabela 8 – IoU e FN dos modelos treinados com as imagens do HeMoLab.

ID	Treinamento						Teste			
	RPN		RC		FN		IoU		FN	
	Treino	Val.	Treino	Val.	Treino	Val.	Treino	Val.	Treino	Val.
19	0.81	0.79	0.90	0.85	0.06	0.09	0.71	0.67	0.08	0.13
20	0.79	0.79	0.89	0.85	0.07	0.07	0.69	0.68	0.06	0.06
22	0.82	0.78	0.89	0.87	0.06	0.07	0.70	0.68	0.16	0.18
23	0.83	0.77	0.88	0.85	0.08	0.08	0.77	0.73	0.07	0.08
24	0.75	0.76	0.85	0.84	0.09	0.07	0.74	0.71	0.06	0.09
25	0.78	0.69	0.77	0.65	0.12	0.09	0.69	0.68	0.09	0.11
26	0.80	0.73	0.78	0.68	0.11	0.11	0.69	0.68	0.09	0.11
27	0.85	0.73	0.82	0.68	0.10	0.09	0.60	0.60	0.15	0.16
28A	-	-	0.91	0.81	0.04	0.11	0.93	**0.81**	0.04	0.12
28B	0.83	0.73	0.82	0.69	0.09	0.10	0.59	0.58	0.16	0.16
32	0.82	0.73	0.83	0.69	0.09	0.08	0.63	0.62	0.10	0.15
33	0.92	0.75	0.87	0.69	0.08	0.20	0.77	0.70	0.13	0.19
29	-	-	0.89	0.78	0.07	0.06	0.82	0.78	0.06	0.06
30	-	-	0.93	0.80	0.04	0.12	0.93	0.79	0.04	0.12
34	-	-	0.92	0.81	0.05	0.09	0.90	**0.81**	0.04	0.10
31	-	-	0.88	0.75	0.07	0.09	0.80	0.76	0.06	0.08
36	-	-	0.95	0.81	0.03	0.11	0.78	0.73	0.00	**0.04**

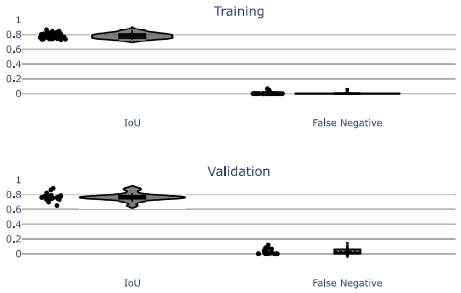
Os treinamentos da Tabela 8 cujas colunas referentes à RPN estão vazias se deve a uma modificação feita no código. Tal modificação é devida, pois, originalmente, a validação do modelo foi implementada da mesma forma que o treinamento. Ou seja, a imagem passava pela DTL, que faz a pré-seleção das RPN_{RoIs}. Como na validação não é desejável essa pré-seleção, pois o intuito é saber o desempenho do modelo caso fosse aplicado fora do conjunto de treinamento, essa inconsistência foi observada no treinamento ID 28 e, a partir dele, foi removida. Portanto, como a DTL é quem acusa qual foi a melhor RPN_{RoI} em relação ao IoU com a Caixa Verdadeira, e ela foi removida da validação, não é mais possível mapear tal desempenho.

O fato da DTL ter participado da validação explica a diminuição drástica da precisão do modelo no grupo de validação entre o modo de execução “treinamento” e “inferência”.

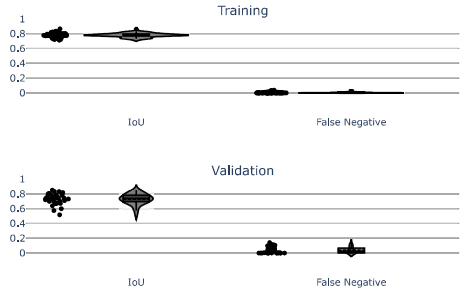
Na Figura 46 é possível observar a distribuição dos dados obtida no modelo original sem a máscara, ID 2, e no melhor modelo modificado, ID 36. Os respectivos modelos foram testados tanto nas imagens do MMWHS quanto nas imagens do HeMoLab. Nessas distribuições é possível observar que o modelo treinado por (XU; WU; FENG, 2018) não generalizou a detecção do coração em um banco de dados externo ao MMWHS. Além disso, também é possível observar no modelo ID 36 que, por mais que este tenha sido treinado com mais imagens e sofrido mais modificações no algoritmo, o ganho de precisão dele em um grupo nunca visto antes não é significativo, indicando que não houve generalização em nenhum dos casos. Todavia, é possível observar que a precisão dos dados do modelo ID 36 no grupo nunca visto, MMWHS, é maior do que a precisão do modelo ID 2 no grupo nunca visto por ele, HeMoLab. Tal fato indica que há necessidade de aumentar o banco de dados de treinamento.



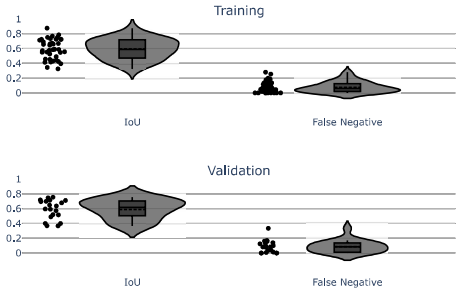
(a) Modelo ID 2 - Imagens HeMoLab.



(b) Modelo ID 2 - Imagens MMWHS



(c) Modelo ID 36 - Imagens HeMoLab.

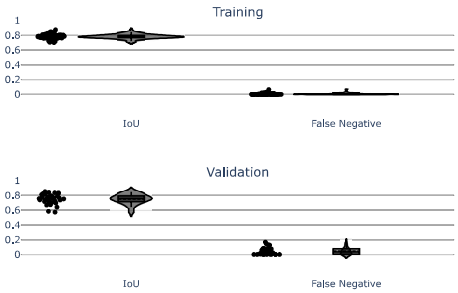


(d) Modelo ID 36 - Imagens MMWHS.

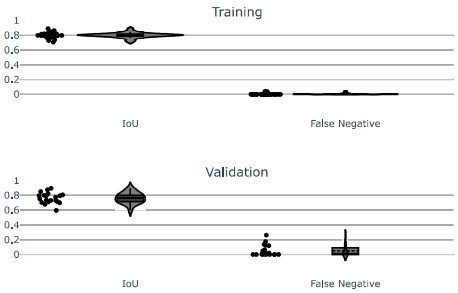
Figura 46 – Distribuição do IoU e do FN dos modelos original e modificado.

Adicionalmente, foi feito um treino, ID 37, com as mesmas configurações do treino ID 36, porém com o grupo de treinamento com as imagens do HeMoLab acrescido com

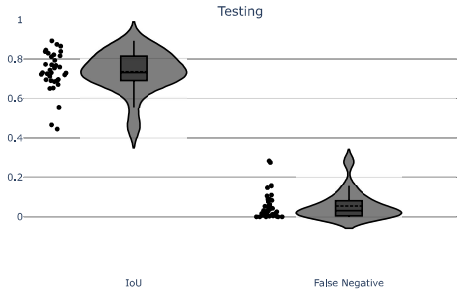
as imagens de treino do MMWHS. Os resultados podem ser visualizados na Tabela 9. A distribuição dos dados referentes ao treinamento 37 pode ser visualizada na Figura 47.



(a) Dados HeMoLab.



(b) Dados MMWHS



(c) Dados HeMoLab (Teste)

Figura 47 – Distribuição do IoU e do FN do modelo 37.

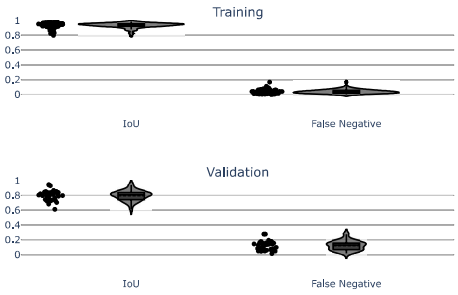
Por fim, tendo em vista a melhoria de generalização do modelo 37, foi feito um treinamento, ID 38, que consistia em, a partir dos pesos obtidos no modelo ID 37, treinar a Rede sem o acréscimo de 5% na caixa delimitadora. O resultado deste modelo pode ser visualizado na Tabela 9.

Tabela 9 – IoU e FN dos modelos ID 37 e 38.

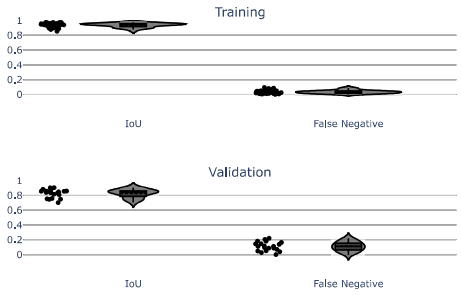
ID	Inferência (HeMoLab)						Inferência (MMWHS)			
	IoU			FN			IoU		FN	
	Treino	Val.	Teste	Treino	Val.	Teste	Treino	Val.	Treino	Val.
37	0.78	0.75	0.74	0.01	0.04	0.06	0.80	0.76	0.00	0.05
38	0.94	0.80	0.80	0.04	0.12	0.12	0.94	0.83	0.03	0.11

Para este treinamento, a distribuição dos resultados nos grupos pode ser visualizada na Figura 48.

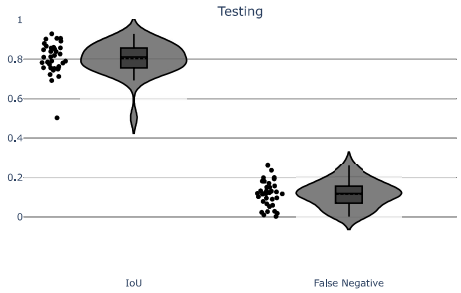
Tendo em vista que o melhor modelo foi o ID 37 pois é o que menos despreza regiões do coração baseado no FN, foram selecionadas algumas imagens do grupo de teste para servirem de amostra para visualizar a caixa delimitadora predita por este modelo. Além disso,



(a) Dados HeMoLab.



(b) Dados MMWHS.



(c) Dados HeMoLab (Teste).

Figura 48 – Distribuição do IoU e do FN do modelo 38.

vale salientar que, quanto ao Fator de Redução (FR), $FR = volume(Caixa)/volume(TC)$, médio das imagens do grupo de teste, a relação volumétrica entre a caixa delimitadora predita e a TC completa é de 39%. Vale ressaltar que as caixas delimitadoras originais possuem um FR de 31%, conforme indicado na Tabela 2. Ao todo, foram utilizados 6 critérios de seleção de amostras:

1. A melhor imagem baseada no IoU;
2. Uma imagem cujo IoU possui valor mais próximo da média;
3. A pior imagem baseada no IoU;
4. A melhor imagem baseada no FN;
5. Uma imagem cujo FN possui valor mais próximo da média;
6. A pior imagem baseada no FN;

O desempenho do treinamento ID 37 pode ser visualizado na Figura 49. As curvas indicam a Função de Perda total e, separadamente, a contribuição de cada parcela desta função. Ou seja, as parcelas da RPN e Classificação, *class* e *deltas*. Além dessas é possível

observar o desempenho do IoU, do FN e do desvio padrão do IoU, como métricas de precisão. A exibição do desvio padrão do IoU tem o intuito de observar se há melhor agrupamento dos dados ao longo do treinamento. Tal análise é necessária, pois, tendo em vista que o IoU, na validação, estabiliza-se após 80 épocas, aproximadamente, é desejável saber se a distribuição dos dados melhora, tornando-os mais próximos. Com base no gráfico do desvio padrão do IoU, é possível observar que há melhor agrupamento do grupo de treinamento, mas isso não ocorre no grupo de teste, o que indica sobre-ajuste do modelo.

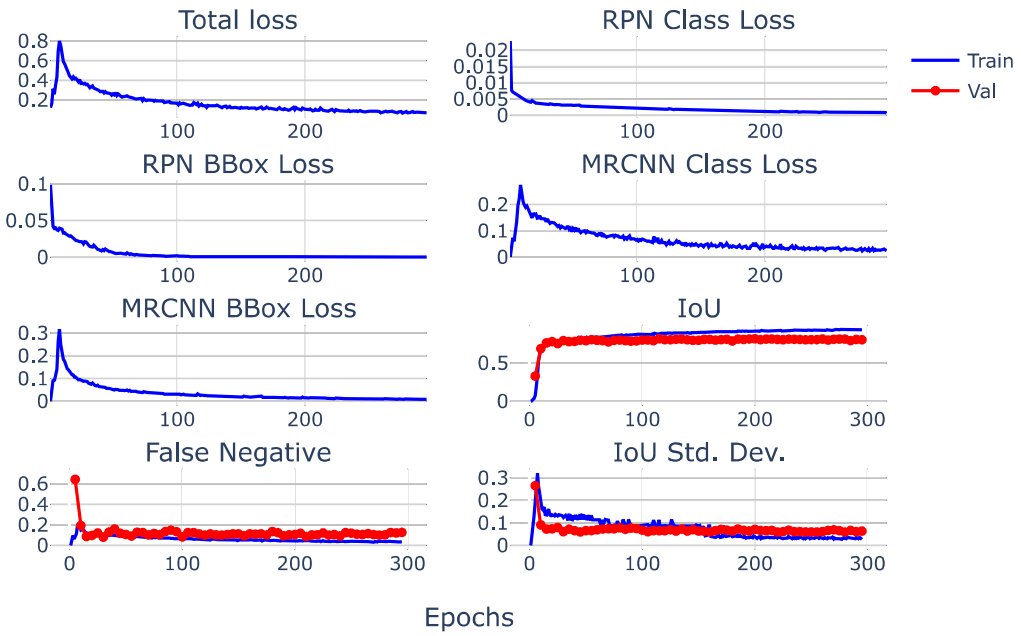


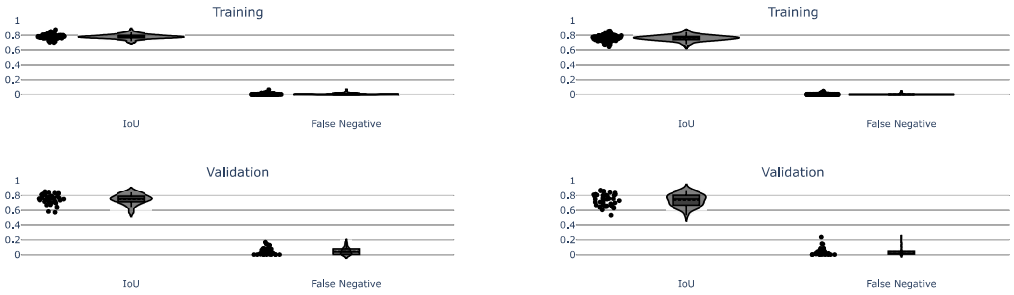
Figura 49 – Gráfico de desempenho do treinamento ID 37.

Nota: MRCNN é o nome, originalmente atribuído pelo autor do modelo, que se refere à rede de classificação.

A fim de avaliar a contribuição específica do FN, foi realizado um treinamento, ID 39, com as mesmas configurações do ID 37 com exceção da presença do FN na Função de Perda. Os gráficos comparativos podem ser visualizados na Figura 50.

Com base na Tabela 10 é possível observar que a introdução do FN na função de perda melhora o valor do IoU, tanto na média quanto na distribuição. Todavia, apresenta diminuição no FN em geral. Porém, é válido ressaltar que os ganhos no IoU são significativamente maiores do que as perdas no FN, estas que sequer chegam em 1%. Dessa forma, é possível concluir que é vantajoso agregar o FN na Função de Perda.

Outra análise feita, ainda na configuração do modelo ID 37, é o aumento da precisão deste em função da taxa de aprendizagem. Na Figura 51 é possível verificar o gráfico de



(a) ID 37 - Possui FN. (b) ID 39 - Não possui FN.

Figura 50 – Comparação do ganho de precisão ao utilizar o FN na Função de Perda.

Tabela 10 – Comparação da distribuição dos dados de validação de acordo com o uso do FN.

ID	**IoU**	**FN**	**Desvio Padrão IoU**	**Desvio Padrão FN**
37	0.750	0.045	0.065	0.050
39	0.737	0.038	0.078	0.045
Ganho(1)	+0.013	-0.007	+0.013	-0.005

Nota: (1) - Equivale ao ganho de informação ao utilizar o FN na função de perda. Os valores positivos indicam as medidas que melhoraram, e sua intensidade, e os negativos indicam o oposto.

desempenho do modelo de acordo com o IoU, FN e a Função de Perda, respectivamente. Ressalta-se que os pontos destacados nesta análise são os resultados do grupo de validação após cem épocas de treinamento. Também é possível observar que há regiões sem pontos, estas lacunas se devem a erros durante a execução devido à saída ser infinita. Tal fato decorre da explosão do gradiente, pois a taxa de aprendizagem estava muito alta. Nestes resultados é possível concluir que o ponto ideal da Taxa de Aprendizagem é 0.001, valor indicado por (XU; WU; FENG, 2018) e utilizado no ID 37.

Por fim, as modificações de código do modelo ID 37 em comparação com o modelo de (XU; WU; FENG, 2018) são:

1. Exclusão de todas as etapas de segmentação;
2. A saída do modelo passa a ser somente a caixa delimitadora do coração;
3. Inclusão do FN como métrica de precisão e contribuição na Função de Perda;
4. Alteração da biblioteca de leitura das imagens de “nibabel” para “Simple ITK”, pois esta última é capaz de abrir imagens com outras extensões de arquivo;
5. Modo de execução da validação passa a ser “inferência” e não “treinamento”; e

Figura 51 – Desempenho do modelo em função da Taxa de Aprendizagem

6. Suspensão das técnicas de aumento de dados no pré-processamento.

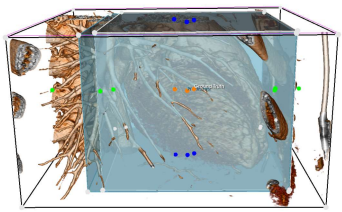
Em suma, os hiperparâmetros estudados podem ser visualizados na Tabela 11. Vale ressaltar que aqueles que demonstraram ganho de precisão no grupo de validação estão destacados em negrito.

Sendo assim, na Figura 52 visualizações do tipo "*volume rendering*" 'com a configuração "CT-Coronary-Arteries-3" do 3D-Slicer junto com as caixas delimitadoras selecionadas conforme os critérios acima podem ser observadas. Ressalta-se que, nas imagens, a caixa delimitadora predita é a que possui arestas e lados azuis, a caixa verdadeira é a que possui arestas brancas e as arestas pretas indicam a limitação da imagem. Além disso, na legenda de cada imagem os parênteses indicam o IoU e o FN da Caixa ao qual a figura se refere, respectivamente.

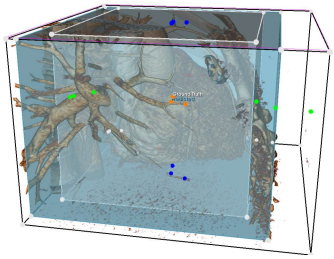
Tabela 11 – Descrição dos hiperparâmetros.

Hiperparâmetro	**Variação**
DTL	**Aleatório**, Ordenado e sem intervir
Aumento de dados	(1), (2) e **NA**
Função de Ativação	**PReLU** e ReLU
Tamanho do Lote	**1** a 6
Treinar BN	Sim ou **Não**
Segmentação	Sim ou **Não**
Aumento da Caixa em 5%	**Sim** ou Não
FN na Função de Perda	**Sim** ou Não
Taxa de Aprendizagem	10^{-7} a 0.8 (**0.001**)

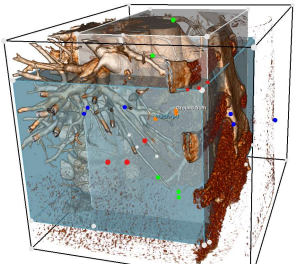
Nota: DTL - *Detection Target Layer*; (1) - Filtro Gaussiano, Corte Lateral e Zoom; (2) - Corte Lateral e Zoom; NA - Não Aplicado; e BN - Normalização em Lote.



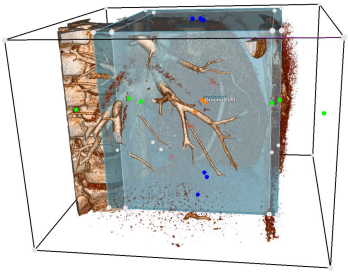
(a) Melhor IoU. (0.89, 0.02)



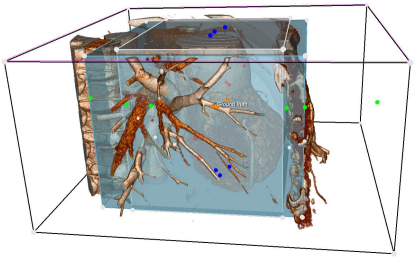
(b) Melhor FN. (0.65, 0.00)



(c) Piores IoU e FN. (0.45, 0.28)



(d) IoU médio. (0.73, 0.01)



(e) FN Médio. (0.67, 0.05)

Figura 52 – Distribuição do IoU e do FN do modelo ID 37.

5 Conclusão

A fim de indicar uma caixa delimitadora que contenha o coração e a aorta ascendente em imagens de Tomografia Computadorizada (TC), foi utilizado como base o modelo apresentado no *Multi Modality Whole Heart Segmentation* (MMWHS) por (XU; WU; FENG, 2018). Após modificações e correções do código original, observou-se que o modelo é capaz de detectar o coração, porém uma série de dificuldades foram aparecendo ao longo da implementação. Tais problemas decorreram, principalmente, pela implementação original do modo de abertura das imagens. Além disso, outro fator importante observado foi um viés introduzido pela *Detection Target Layer* (DTL), uma camada intermediária utilizada para selecionar bons candidatos para treinar a Rede de Classificação. Tal viés faz com que os resultados no treinamento sejam superiores aos observados na inferência e tal fator não foi possível de ser mitigado sem decréscimo de desempenho no modelo. Além disso, por último, observou-se que o código original de (XU; WU; FENG, 2018) não bloqueia o Aumento de Dados na fase de validação e esta também utilizava o viés da DTL.

Entretanto, os fatores citados acima devem ser interpretados como pontos melhorados e não invalidam o modelo para ser abordado em situação real. Adicionalmente, foram implementadas modificações na arquitetura de tal forma a eliminar a segmentação original do modelo a fim de focá-lo somente na detecção e, além disso, introduzimos uma nova métrica de avaliação, o Falso Negativo. Esta métrica indica o percentual do coração que é cortado pela detecção da rede e, inclusive, foi implementado como métrica a ser minimizada na Função de Perda.

Por fim, ressalta-se que, originalmente, (XU; WU; FENG, 2018) treinou o modelo utilizando 40 imagens provenientes de um algoritmo de segmentação e testaram com 20 manualmente segmentadas. Já neste trabalho, possuímos 166 imagens das quais foram estatisticamente separadas em 94 para treinamento, 36 para validação e 36 para teste. O critério para segregação foi feito com base nas informações da aquisição, métricas espaciais e aparelho de obtenção de cada imagem. O modelo original de (XU; WU; FENG, 2018) obteve precisão de 77% na métrica *Intersection Over Union* (IoU) e 3% no Falso Negativo nas imagens utilizadas para validação no MMWHS. O mesmo modelo aplicado no nosso grupo de teste, pontuou 38% no IoU e 45% no Falso Negativo. Já o nosso melhor modelo modificado, número 36, treinado com nossos dados, pontuou 59% no IoU e 9% no Falso Negativo nos dados MMWHS e 73% no IoU e 4% no Falso Negativo no nosso grupo de teste. Por fim, um modelo treinado com ambos os grupos de treinamento pontuou 76% e 5% no IoU e Falso Negativo nos dados do MMWHS, respectivamente, e 74% e 6% no IoU e Falso Negativo nos dados de teste do HeMoLab, além de reduzir a TC, em média, para 39% do volume original.

Portanto, tais resultados demonstram o potencial de generalização da detecção do modelo. Todavia, para melhorar a precisão é necessário um grupo maior para treinamento. Tal fato pôde ser observado pois a alta precisão de um modelo treinado em um grupo de dados não se replica quando este é testado em outro grupo. Sendo assim, faz-se necessário ampliar a quantidade de dados para realizar o processo inteiro, incluindo treino, validação e teste.

Referências

AGARAP, A. F. Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375, 2018. Citado na página 19.

ARAúJO, V. S. e Luan Silva e Adam Santos e L. Análise comparativa de redes neurais convolucionais no reconhecimento de cenas. Anais do Computer on the Beach, v. 11, n. 1, p. 419–426, 2020. ISSN 2358-0852. Disponível em: <https://siaiap32.univali.br/seer/index.php/acotb/article/view/16801>. Citado na página 15.

BAKATOR, M.; RADOSAV, D. Deep learning and medical diagnosis: A review of literature. Multimodal Technologies and Interaction, Multidisciplinary Digital Publishing Institute, v. 2, n. 3, p. 47, 2018. Citado na página 15.

BEZERRA, C. G. et al. Coronary fractional flow reserve derived from intravascular ultrasound imaging: validation of a new computational method of fusion between anatomy and physiology. Catheterization and Cardiovascular Interventions, Wiley Online Library, v. 93, n. 2, p. 266–274, 2019. Citado na página 15.

BUSSAB, W. d. O.; MORETTIN, P. A. Estatística básica. In: Estatística básica. [S.l.: s.n.], 2010. p. xvi–540. Citado 4 vezes nas páginas 40, 82, 83 e 84.

CARSON, J. M. et al. Non-invasive coronary ct angiography-derived fractional flow reserve: A benchmark study comparing the diagnostic performance of four different computational methodologies. International journal for numerical methods in biomedical engineering, Wiley Online Library, v. 35, n. 10, p. e3235, 2019. Citado na página 39.

CARVALHO, A. C. P. História da tomografia computadorizada. Revista Imagem, v. 29, n. 2, p. 61–66, 2007. Citado na página 38.

CHHABRA, G.; GAGAN, J.; KUMAR, J. Automatic segmentation of left ventricle in cardiac magnetic resonance images. arXiv preprint arXiv:2201.12805, 2022. Citado na página 35.

DUMOULIN, V.; VISIN, F. A guide to convolution arithmetic for deep learning. 2016. Cite arxiv:1603.07285. Disponível em: <http://arxiv.org/abs/1603.07285>. Citado na página 24.

GHOLAMALINEZHAD, H.; KHOSRAVI, H. Pooling methods in deep neural networks, a review. arXiv preprint arXiv:2009.07485, 2020. Citado na página 24.

GIRSHICK, R. Fast r-cnn. In: 2015 IEEE International Conference on Computer Vision (ICCV). [S.l.: s.n.], 2015. p. 1440–1448. Citado 2 vezes nas páginas 29 e 62.

GIRSHICK, R. et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [s.n.], 2014. v. 00, p. 580–587. ISSN 1063-6919. Disponível em: <https://ieeexplore.ieee.org/abstract/document/6909475/>. Citado 2 vezes nas páginas 28 e 62.

GUNNING, D. et al. Xai—explainable artificial intelligence. Science robotics, American Association for the Advancement of Science, v. 4, n. 37, p. eaay7120, 2019. Citado na página 23.

HE, K. et al. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. [S.l.: s.n.], 2016. p. 770–778. Citado na página 48.

HINTON, G. E. et al. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580, 2012. Citado na página 27.

HOCHREITER, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, World Scientific, v. 6, n. 02, p. 107–116, 1998. Citado na página 48.

HUMPIRE-MAMANI, G. E. et al. Efficient organ localization using multi-label convolutional neural networks in thorax-abdomen ct scans. Physics in Medicine & Biology, IOP Publishing, v. 63, n. 8, p. 085003, 2018. Citado 3 vezes nas páginas 7, 33 e 34.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: BACH, F.; BLEI, D. (Ed.). Proceedings of the 32nd International Conference on Machine Learning. Lille, France: PMLR, 2015. (Proceedings of Machine Learning Research, v. 37), p. 448–456. Disponível em: <https://proceedings.mlr.press/v37/ioffe15.html>. Citado na página 26.

KWON, S.-S. et al. A novel patient-specific model to compute coronary fractional flow reserve. Progress in biophysics and molecular biology, Elsevier, v. 116, n. 1, p. 48–55, 2014. Citado na página 16.

LIN, T.-Y. et al. Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. [S.l.: s.n.], 2017. p. 2117–2125. Citado na página 50.

MA, J. et al. Towards Efficient COVID-19 CT Annotation: A Benchmark for Lung and Infection Segmentation. arXiv, 2020. Disponível em: <https://arxiv.org/abs/2004.12537v1>. Citado na página 15.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, Springer, v. 5, n. 4, p. 115–133, 1943. Citado na página 18.

MITCHELL, T. M. Machine learning, International Edition. McGraw-Hill, 1997. (McGraw-Hill Series in Computer Science). ISBN 978-0-07-042807-2. Disponível em: <https://www.worldcat.org/oclc/61321007>. Citado na página 18.

MOURÃO, A. P. Tomografia computadorizada: tecnologias e aplicações. [S.l.]: Difusão Editora, 2018. Citado 4 vezes nas páginas 38, 39, 88 e 89.

NIELSEN, M. A. Neural networks and deep learning. [S.l.]: Determination press San Francisco, CA, USA, 2015. v. 25. Citado na página 48.

QIU, Z.; YAO, T.; MEI, T. Learning spatio-temporal representation with pseudo-3d residual networks. In: proceedings of the IEEE International Conference on Computer Vision. [S.l.: s.n.], 2017. p. 5533–5541. Citado na página 48.

RAHIMZADEH, M.; ATTAR, A.; SAKHAEI, S. M. A fully automated deep learning-based network for detecting covid-19 from a new and large lung ct scan dataset. Biomedical Signal Processing and Control, v. 68, p. 102588, 2021. ISSN 1746-8094. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1746809421001853>. Citado na página 15.

RAUBER, T. W. Redes neurais artificiais. Universidade Federal do Espírito Santo, v. 29, 2005. Citado na página 23.

REDMON, J. et al. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. [S.l.: s.n.], 2016. p. 779–788. Citado 2 vezes nas páginas 32 e 33.

REN, S. et al. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, v. 28, 2015. Citado 2 vezes nas páginas 30 e 48.

S., S. R. et al. Using YOLO based deep learning network for real time detection and localization of lung nodules from low dose CT scans. In: PETRICK, N.; MORI, K. (Ed.). Medical Imaging 2018: Computer-Aided Diagnosis. SPIE, 2018. v. 10575, p. 347 – 355. Disponível em: <https://doi.org/10.1117/12.2293699>. Citado na página 15.

SHAPIRO, L. Computer vision and image processing. [S.l.]: Academic Press, 1992. Citado na página 44.

SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. Journal of big data, Springer, v. 6, n. 1, p. 1–48, 2019. Citado na página 43.

SOANS, R. E.; SHACKLEFORD, J. A. Organ localization and identification in thoracic ct volumes using 3d cnns leveraging spatial anatomic relations. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. Medical Imaging 2018: Image Processing. [S.l.], 2018. v. 10574, p. 105741X. Citado 3 vezes nas páginas 7, 34 e 36.

SRIVASTAVA, N. et al. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014. Citado na página 27.

TALOU, G. D. M. et al. Mechanical characterization of the vessel wall by data assimilation of intravascular ultrasound studies. Frontiers in Physiology, Frontiers Media SA, v. 9, p. 292, 2018. Citado na página 15.

UIJLINGS, J. R. et al. Selective search for object recognition. International journal of computer vision, Springer, v. 104, n. 2, p. 154–171, 2013. Citado na página 28.

XU, X. et al. Efficient multiple organ localization in ct image using 3d region proposal network. IEEE transactions on medical imaging, IEEE, v. 38, n. 8, p. 1885–1898, 2019. Citado 4 vezes nas páginas 7, 33, 34 e 35.

XU, Z.; WU, Z.; FENG, J. Cfun: Combining faster r-cnn and u-net network for efficient whole heart segmentation. arXiv preprint arXiv:1812.04914, 2018. Citado 13 vezes nas páginas 7, 34, 36, 37, 39, 40, 48, 56, 64, 65, 68, 72 e 75.

ZHOU, X. et al. Automatic localization of solid organs on 3d CT images by a collaborative majority voting decision based on ensemble learning. Comput. Medical Imaging Graph., v. 36, n. 4, p. 304–313, 2012. Disponível em: <https://doi.org/10.1016/j.compmedimag.2011.12.004>. Citado na página 33.

Apêndices

APÊNDICE A – Testes Estatísticos

A.1 Teste Qui-Quadrado

Este teste avalia as frequências de ocorrência de C características dos grupos A e B. O valor de χ^2 é uma medida de afastamento entre os dados. Se a hipótese H_0 for verdadeira, o valor de χ^2 deverá ser próximo de zero, caso seja grande indica que há independência entre as variáveis (BUSSAB; MORETTIN, 2010). Dentre as informações das imagens na Tabela 2, apenas duas são do tipo qualitativo: Banco de Dados e modelo do *scanner*. Porém, como Banco de Dados já é considerado para fazer a seleção inicial, apenas no modelo do *scanner* que é avaliado o grau de similaridade. A Tabela 12 exemplifica a distribuição das C características do modelo de *scanner*, que, neste caso, são os tipos deste, assim como as f_{gc} informações que indicam a quantidade de imagens com a informação c no grupo g.

Tabela 12 – Tabela de frequências.

Grupo	**Aquilion ONE**	**Discovery CT750 HD**	$\cdots$	**Siemens Somatom Definition Flash**
A	f_{11}	f_{12}	$\cdots$	f_{17}
B	f_{21}	f_{22}	$\cdots$	f_{27}

Em posse das informações das frequências de ocorrência, o método de cálculo do χ^2 se dá por (A.1).

$$\chi^2 = \sum_c \sum_g \frac{(f_{gc} - f^*_{gc})^2}{f^*_{gc}} \tag{A.1}$$

Em que f^*_{gc} é a frequência esperada da característica c no grupo g se H_0 for verdadeira. Vale ressaltar que f^*_{gc} é dada por (A.2).

$$f^*_{gc} = \frac{S_g \cdot S_c}{Q} \tag{A.2}$$

Em que S_g é a quantidade de imagens do grupo g, S_c é a quantidade de imagens com a característica c e Q é a quantidade total de imagens. Expandindo (A.2) obtemos (A.3).

$$f^*_{gc} = \frac{\sum_{i=1}^{G} f_{ic} \cdot \sum_{j=1}^{C} f_{gj}}{\sum_{i=1}^{G} \sum_{j=1}^{C} f_{ij}} \tag{A.3}$$

Em que G é a quantidade de grupos e C é a quantidade de características. Ressalta-se que a distribuição Qui-Quadrado possui v graus de liberdade, calculado por $v = (G-1)\cdot(C-1)$, e sua densidade se dá por (A.4).

$$f(y,v) = \begin{cases} \frac{y^{v/2-1}}{\Gamma(v/2)\cdot 2^{v/2}} \cdot e^{-y/2} & , \quad y > 0 \\ 0 & , \quad y < 0 \end{cases} \tag{A.4}$$

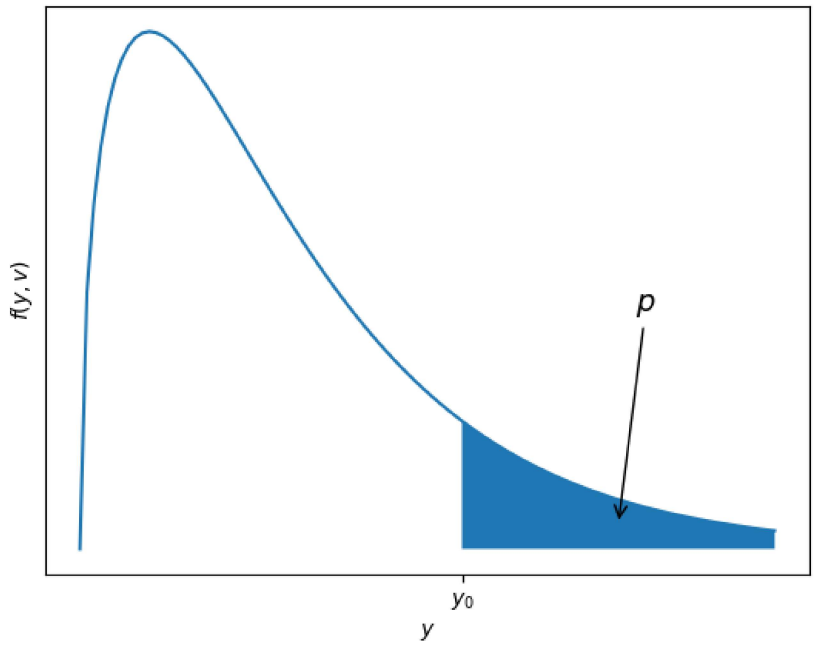
Sendo a função $\Gamma(x)$ uma interpolação do fatorial de um número $x > 0$, dada por (A.5) (BUSSAB; MORETTIN, 2010).

$$\Gamma(x) = \int_0^\infty e^{-t} \cdot t^{x-1} dt \tag{A.5}$$

A distribuição Qui-Quadrado pode ser utilizada para determinar a probabilidade de se obter um número. A interpretação de $P(Y \geq y_0) = p$ é que, dados v e y_0, p é a probabilidade de se obter um valor de $y \geq y_0$, observe a Figura 53. A função $P(Y \geq y_0)$ para a distribuição qui-quadrado é dada por (A.6).

$$P(Y \geq y_0) = \int_{y_0}^\infty f(y,v) dy \tag{A.6}$$

Figura 53 – Gráfico da função Qui-Quadrado para $v = 3$.



Sendo assim, em posse do valor de χ^2, obtido em (A.1), faz-se $y_0 = \chi^2$ em (A.6) e se obtêm o valor de p, cuja interpretação é: p é a probabilidade de se obter aquele

valor de χ^2, ou mais distante, se os dados não forem homogêneos. Assume-se um nível de significância $\alpha = 0.05$, cuja interpretação é: dado $p < \alpha$, rejeito H_0, pois não confio na probabilidade p de ter obtido aquela distribuição assumindo que H_0 seja verdadeira. (BUSSAB; MORETTIN, 2010)

A.2 Teste U de Mann-Whitney

Este teste é aplicado às informações quantitativas, cujas amostras devem ser independentes, ou seja, o valor de uma amostra não pode estar relacionado com a outra. O objetivo do teste é saber se uma população tende a ter valores maiores do que a outra, ou se possuem a mesma mediana ou média. Este teste é baseado nos postos dos valores e não na sua intensidade. O posto é dado pela ordenação das amostras, da menor para a maior, independentemente do grupo associado.

Para realizar o teste, primeiramente é necessário escolher a variável V para análise, que possui valor v_{gi}, sendo g referente ao grupo e i ao índice para a informação. Para cada uma das $n + m$ informações dos grupos A e B, n sendo o número de amostras em A e m em B, determina-se o posto r_k para cada valor. Observe a Tabela 13.

Tabela 13 – Tabela de postos.

Grupo	V	**Posto**
A	v_{11}	r_1
A	v_{12}	r_2
A	v_{13}	r_3
⋮	⋮	⋮
A	v_{1n}	r_n
B	v_{21}	r_{n+1}
B	v_{22}	r_{n+2}
B	v_{23}	r_{n+3}
⋮	⋮	⋮
B	v_{nm}	r_{n+m}

Vale salientar que caso haja repetição entre os valores v_{gi}, os postos r_k de todos os k itens repetidos serão dados pela média dos postos na ordenação comum r_k^*, calculados por (A.7).

$$r_k = \frac{1}{k} \cdot \sum_k r_k^* \tag{A.7}$$

A etapa seguinte consiste em calcular o posto total R_g de cada grupo, para isso, faz-se (A.8):

$$R_g = \sum_{i=1}^{n+m} r_i, \text{ se } r_i \in g \tag{A.8}$$

O próximo passo é calcular as estatísticas de Mann-Whitney para cada grupo, dadas por (A.9) e (A.10):

$$U_1 = n_1 \cdot n_2 + \frac{n_1(n_1+1)}{2} - R_1 \tag{A.9}$$

$$U_2 = n_1 \cdot n_2 + \frac{n_2(n_2+1)}{2} - R_2 \tag{A.10}$$

Em que n_1 e n_2 é a quantidade de amostras no grupo 1 e 2, respectivamente. Com base nestas informações, é escolhido o menor valor entre U_1 e U_2 e avalia o seu afastamento em uma distribuição normal, conforme (A.13), levando em consideração a média (A.11) e o desvio padrão (A.12).

$$\mu = \frac{n_1 \cdot n_2}{2} \tag{A.11}$$

$$\sigma = \sqrt{\frac{n_1 \cdot n_2(n_1+n_2+1)}{12}} \tag{A.12}$$

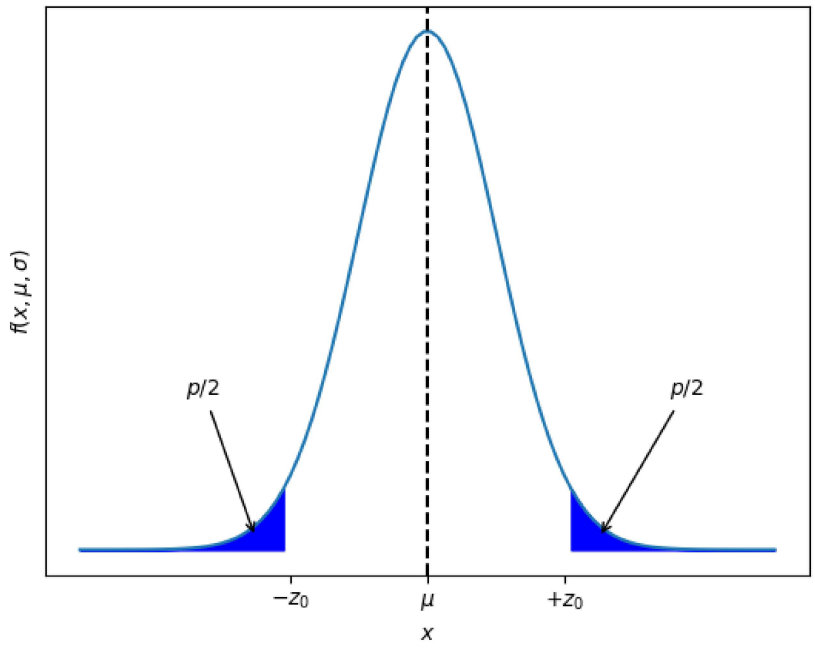
$$z_0 = \frac{U_{min} - \mu}{\sigma} \tag{A.13}$$

Como a hipótese nula diz que $\mu(A) = \mu(B)$ e não se sabe, a priori, a relação entre estas, o teste aplicado à distribuição deve ser o bicaudal. É levado em consideração um nível de significância $\alpha = 0.05$, cuja interpretação é: $P(Z \geq |z_0|) = p$, definida por A.14.

$$P(Z \geq |z_0|) = 2\int_{|z_0|}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} dx \tag{A.14}$$

Observe que há um fator 2 multiplicando a integral, que se deve ao fato do teste ser bicaudal, observe a Figura 54. Dessa forma, p é a probabilidade de obter aquela distribuição, ou uma mais distante, considerando que H_0 seja verdadeira. Caso $p \leq \alpha$, rejeita-se H_0 devido à baixa probabilidade de ocorrência.

Figura 54 – Representação da probabilidade para teste bicaudal.



APÊNDICE B – Método de Obtenção de Imagens de TC

O método de obtenção da imagem de TC está relacionado com o grau de absorção do raio X pelos diferentes tecidos do corpo humano. Este grau de absorção está relacionado com a característica da estrutura e a energia do fóton, definida pelo coeficiente de atenuação linear μ. Outro fator a ser considerado é a distância x que o feixe percorre a matéria. Assim, para determinado número de fótons de entrada (N_0), o número de fótons transmitidos (N_t) é dado por:

$$N_t = N_0 \cdot e^{-\mu x} \tag{B.1}$$

O parâmetro utilizado para observar a interação do feixe com o material é a intensidade (I), pois esta relaciona a taxa de fótons do feixe e a energia destes fótons. Para um único material, a intensidade do feixe transmitido (I_t), dependente da intensidade do feixe incidente (I_0), é definida por:

$$I_t = I_0 \cdot e^{-\mu x} \tag{B.2}$$

Em TC, a obtenção da imagem depende da medição do feixe de raios X. O feixe gerado é devidamente colimado e direcionado para a região do corte desejado. Parte dos fótons emitidos são absorvidos pelas estruturas atravessadas por este feixe proporcionalmente ao coeficiente de atenuação linear médio. A intensidade do feixe transmitido é definida por:

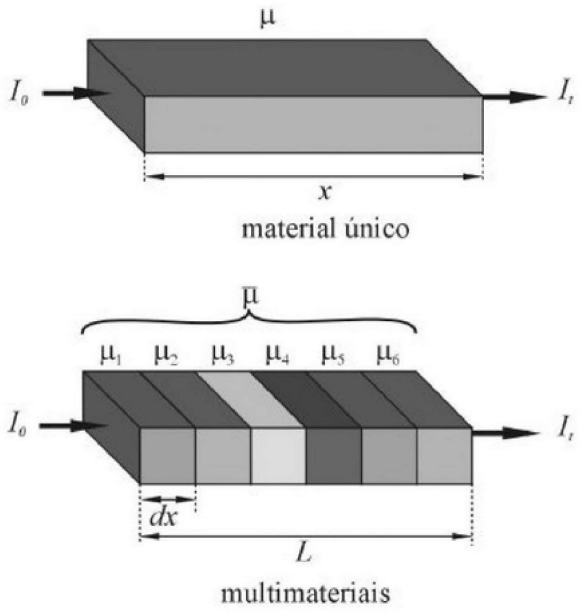
$$I_t = I_0 \cdot e^{-\int_0^L \mu(x)\,dx} \tag{B.3}$$

Em que L é o comprimento do caminho percorrido pelo feixe de raios X e $\mu(x)$ é o coeficiente de atenuação linear que varia com o tecido ao longo do percurso L. A Figura 55 ilustra as atenuações do feixe que podem ser calculadas utilizando as Equações B.2 e B.3. A integral do coeficiente de atenuação é dada por:

$$\int_0^L \mu(x)\,dx = -\frac{1}{L} \cdot ln\left(\frac{I_t}{I_0}\right) \tag{B.4}$$

O feixe transmitido é definido pela intensidade inicial do mesmo menos a parcela que foi absorvida pelo material. Esta atenuação depende do tecido, espessura, composição,

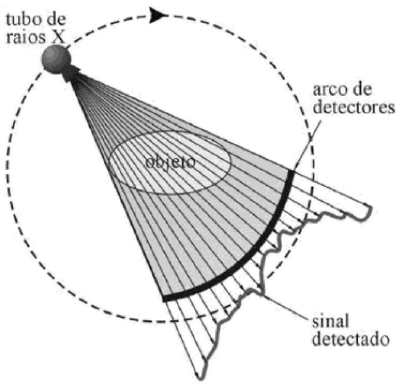
Figura 55 – Interação do feixe de raios X com material único e com multi materiais.



Fonte: Mourão (2018, Fig. 4.1).

etc. A intensidade inicial (I_0) é conhecida pela calibração do sistema e as atenuações são provenientes das medições da intensidade do feixe transmitido (I_t). Para a elaboração da imagem são necessárias muitas medições (I_t), por diferentes caminhos que levam aos detectores de radiação. A Figura 56 representa a aquisição de um sinal em determinado instante de rotação em torno do paciente. Quanto mais vezes o sinal é capturado ao longo dos 360º de rotação do cubo de raios X, melhor é a qualidade da imagem gerada.

Figura 56 – Geração do sinal com atenuação do feixe promovida pelo objeto.



Fonte: Mourão (2018, Fig. 4.2).

As informações coletadas pelos detectores são processadas por um programa que

calcula a atenuação do feixe promovida por cada uma das fileiras de voxels, estes sendo elementos de volume em imagens 3D. Como a intensidade do feixe que sai do tubo é conhecida, assim como a atenuação, que foi obtida por meio do detector, calcula-se a parcela do feixe atenuada pelo tecido que foi atravessado. A tonalidade de cinza, correspondente à atenuação gerada pelos voxels atravessados, depende dos coeficientes de atenuação linear dos tecidos envolvidos, observe a Figura 22.

Esses valores de atenuação μ_x são normalizados em uma escala que, para organismos humanos, varia de -1000 a 1000. Esta escala de atenuação é conhecida como escala Hounsfield H_x e a normalização dos valores está definida neste intervalo devido aos tecidos envolvidos no processo. Ressalta-se que não há valor máximo para esta escala, porém, para os tecidos humanos, pode ser limitado a 1000, que é o maior valor obtido pelo osso. A normalização é calculada conforme: (MOURÃO, 2018)

$$H_x = 1000 \cdot \frac{\mu_x - \mu_{(H_2O)}}{\mu_{(H_2O)}} \tag{B.5}$$