



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de São José do Rio Preto

Aldimir José Bruzadin

Segmentação Interativa de Imagens de Tomografia
Computadorizada Pulmonar com COVID-19 via
Aprendizado Profundo

São José do Rio Preto
2021

Aldimir José Bruzadin

Segmentação Interativa de Imagens de Tomografia
Computadorizada Pulmonar com COVID-19 via
Aprendizado Profundo

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Matemática, junto ao Programa de Pós-Graduação em Matemática, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de São José do Rio Preto.

Orientador: Prof. Dr. Wallace Correa de Oliveira Casaca

Co-orientador: Prof. Dr. Maurílio Boaventura

Financiadora: CAPES

São José do Rio Preto
2021

B914s

Bruzadin, Aldimir José

Segmentação interativa de imagens de tomografia computadorizada pulmonar com COVID-19 via aprendizado profundo / Aldimir José Bruzadin.

-- São José do Rio Preto, 2021

72 p.

Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto

Orientadora: Wallace Correa de Oliveira Casaca

Coorientadora: Maurílio Boaventura

1. Segmentação de Imagens. 2. Inteligência Computacional. 3. Aprendizado Profundo. 4. COVID-19. 5. Imagens Médicas. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

Aldimir José Bruzadin

Segmentação Interativa de Imagens de Tomografia
Computadorizada Pulmonar com COVID-19 via
Aprendizado Profundo

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Matemática, junto ao Programa de Pós-Graduação em Matemática, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de São José do Rio Preto.

Financiadora: CAPES

Comissão Examinadora

Prof. Dr. Wallace Correa de Oliveira Casaca
UNESP – Câmpus Experimental de Rosana
Orientador

Prof. Dr. Harlen Costa Batagelo
UFABC – Universidade Federal do ABC

Dr. Lucas Rodrigues Borges
FMRP USP – Faculdade de Medicina de Ribeirão Preto

São José do Rio Preto
2 de junho de 2021

*Dedico este trabalho a Deus e todos que,
apesar das adversidades, estiveram sempre me apoiando.*

AGRADECIMENTOS

Primeiramente agradeço a Deus, por ter aberto os caminhos até aqui. Sem Ele nada seria possível e, com Ele, nada é impossível.

Agradeço a toda minha família. Em especial aos meus pais, Aldair e Sionara, por me apoiarem e acreditar nos meus sonhos, de que eles poderiam ser realizados. Sem eles não poderia ter chego tão longe.

À minha namorada Beatriz, pelo imenso apoio e paciência em todos os momentos dessa caminhada. Por sempre acreditar em mim, quando nem mesmo eu acreditava, me apoiando e incentivando a seguir meus sonhos.

Aos grandes amigos que fiz durante a graduação, e que vou levar para a vida. Em especial Thainá e Thaís, pelas horas de risadas e descontração.

Aos meus amigos Adriano, Carlos, Maurício, Murillo, Neto e Pedro, que o programa PET me proporcionou, pelo apoio ao longo dos anos de programa e principalmente nos momentos difíceis do curso.

Ao meu orientador Prof. Dr. Wallace Correa de Oliveira Casaca, e co-orientador Prof. Dr. Maurílio Boaventura, profissionais exemplares que me inspiro muito. Obrigado pela oportunidade de aprendizado e por acreditarem na minha capacidade, serei eternamente grato por todo conhecimento adquirido.

À banca examinadora, composta pelo Prof. Dr. Harlen Costa Batagelo e Dr. Lucas Rodrigues Borges, pela disponibilidade, sugestões e contribuições.

Aos Prof. Dr. Rogério Luiz Iope, Prof. Dr. José Remo Ferreira Brega, e Marcos Venicius pelo compartilhamento de recursos computacionais para o desenvolvimento da dissertação.

À toda a equipe técnica e acadêmica do Centro de Ciências Matemáticas Aplicadas à Indústria (CeMEAI) e à FAPESP, que disponibilizaram recursos computacionais para o desenvolvimento desta pesquisa.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

*Noventa por cento do sucesso
se baseia simplesmente em insistir.*
Woody Allen (1983, p. 1)

RESUMO

Neste trabalho, apresentamos um novo método de segmentação interativa de imagens baseado em aprendizado profundo, com foco em imagens médicas do pulmão de pacientes acometidos pela Covid-19. Inicialmente, aplicamos uma pré-segmentação nas imagens, utilizando-se para tal o modelo de Mistura Gaussiana visando prever os pesos das bordas de cada imagem para posterior uso no algoritmo denominado *Random Walker*, que foi integrado como parte de uma arquitetura inteligente de Redes Neurais Convolucionais. Além disso, no processo de aprendizado, empregamos o método do Gradiente Descendente em conjunto com duas técnicas de simplificação, a saber: Gradiente Esperso e Poda do Gradiente, reduzindo assim a quantidade de cálculos a serem realizados durante a etapa de backpropagation no pipeline de aprendizado proposto. Já para a obtenção das segmentações após o aprendizado dos pesos das bordas, e tendo em mente que representações 3D, de Tomografia Computadorizada, podem chegar a reunir centenas de imagens 2D (*slices*), o método emprega apenas um mapa de *pixels* rotulados pelo usuário como entrada a fim de criar automaticamente novos mapas de rotulação para os demais *slices* 2D do órgão do paciente. Resultados experimentais e comparações com outras técnicas da literatura atestam a alta performance e acurácia da metodologia de segmentação de imagens proposta.

Palavras-chave: Segmentação de Imagens. Inteligência Computacional. Aprendizado Profundo. COVID-19. Imagens Médicas.

ABSTRACT

In this dissertation, we present a new interactive image segmentation method based on deep learning for pulmonary medical images, from patients affected by Covid-19. First, we apply a pre-segmentation step on the input images by using the so-called Gaussian Mixture Model in order to predict the edge weights of the images, which are used by the Random Walker algorithm coupled with an integrated Convolutional Neural Network-based architecture. During the learning process, we take the Descending Gradient method in conjunction with two simplification techniques: Sparse Gradient and Gradient Pruning, to reduce the amount of calculations in the back-propagation stage of the proposed learning pipeline. In order to obtain the segmentations after learning the edge weights, and keeping in mind that 3D representations from Computed Tomography can usually reach hundreds of slices, we employ only a user-labeled map of pixels as input data to automatically create new seed maps to all other slices of the patient's organ. Experimental results and comparisons against existing techniques attest to the high performance and accuracy of the proposed image segmentation methodology.

Keywords: Image Segmentation. Computational Intelligence. Deep Learning. COVID-19. Medical Images.

Lista de Figuras

1.1	Segmentação binária de uma imagem, em que a raposa é a região de interesse a ser particionada em meio ao fundo da fotografia. Fonte: [2].	23
1.2	Segmentação do tipo <i>multi-clusters</i> , isto é, com varias partições. Fonte: [3].	23
1.3	Segmentação de duas fatias adjacentes em uma imagem CT, do pulmão. Fonte: Autoria própria.	24
1.4	Ilustração das <i>seeds</i> (marcações feitas pelo usuário, em vermelho e verde) em uma imagem do tipo CT. Fonte: Autoria própria.	25
1.5	(a) Segmentação obtida com o método proposto por [4], e (b) por um agente humano (<i>ground truth</i>) dos tumores de mama. Fonte: [4].	25
1.6	Segmentação obtida com o método <i>Watershed</i> . Fonte: [5].	26
1.7	Segmentação de baixa qualidade utilizando o método <i>Learned Watershed</i> por conta de “fraco contorno” (<i>weak edges</i>). Fonte: [6].	27
1.8	Fatias de uma CT dos pulmões de um homem adulto saudável. Fonte: < https://www.nytimes.com/2021/03/09/health/lung-cancer-smoking-screenings-black-women-younger-adults.htm >.	28
1.9	Exemplos de lesões representadas por vidro fosco, consolidação e derrame pleural nas fatias de um volume CT. Fonte: Autoria própria.	28
2.1	Representação de um grafo sobre uma malha quadriculada em que o caminhante aleatório se locomove conforme a concepção do método <i>Random Walker</i> . Fonte: Autoria própria.	32
2.2	Propagação das sementes após a atribuição de rótulos de <i>Objeto</i> (F) e <i>Fundo</i> (B) aos pontos ainda não rotulados. Fonte: Autoria própria.	33
2.3	Representação da vizinhança 4-conectada de um pixel P_5 . Fonte: [3].	33
2.4	Ilustração de duas bacias vizinhas, seu divisor, e seus pontos de acumulo. Fonte: [2].	35
2.5	(a) Imagem de entrada com as sementes, (b) superfície topográfica da imagem de entrada, de (c) a (e) os processos de “inundação” do algoritmo <i>Watershed</i> até a segmentação final (f). Fonte: Adaptado de < http://www.cmm.mines-paristech.fr/~beucher/wtshed.html >.	35
2.6	Resultados do processamento de imagens via GMM através do Algoritmo EM para diferentes parâmetros k . Da esquerda para a direita: imagens de entrada, e os resultados obtidos tomando-se $k = 2$, $k = 4$ e $k = 7$. Fonte: [2].	37
2.7	Modelo <i>Perceptron</i> . Fonte: [7].	39
2.8	Rede Neural Simples e Rede Neural de Aprendizado Profundo. Fonte: < http://deeplearningbook.com.br >.	39

2.9	Parte da curva de uma função de custo. Os vetores representam os gradientes para quatro diferentes valores de peso w_1 , w_2 e w_3 , e w_{opt} é um valor de mínimo da função (ponto ótimo). Fonte: [7].	41
2.10	Exemplo simples da operação convolução 2-D. Fonte: [8]	42
2.11	Arquitetura do algoritmo COB. Fonte: [9].	43
2.12	Arquitetura de aprendizado profundo de COB. As conexões mostram os diferentes estágios que são usados para gerar os contornos multiescala. Fonte: [9].	43
2.13	(a) Imagem de entrada, (b) ground truth, (c) segmentação do Watershed padrão e (d) segmentação do <i>Learned Watershed</i> . Fonte: [6].	44
2.14	Arquitetura do algoritmo Learned Watershed. Fonte: [6].	45
2.15	Arquitetura da rede neural do algoritmo <i>Learned Watershed</i> . Fonte: [6].	45
3.1	Visão geral do pipeline do algoritmo proposto. Acima: A imagem de entrada passa por uma pré-segmentação, que é então mapeada por uma CNN para um grafo de borda ponderada não direcionado. Uma matriz de probabilidade é calculada para cada região usando o segmentador <i>Random Walker</i> . Na etapa de teste, uma segmentação é calculada a partir das probabilidades. Abaixo: A partir dos mapas de probabilidade, uma amostra esparsa do gradiente é calculada e, então, fornecida a rede CNN. Fonte: Autoria própria.	50
3.2	Erosão da região azul escuro para a região azul claro, e dilatação da região azul escuro para a região azul claro. Fonte: Autoria própria.	50
3.3	(a) Sementes de objetos antes da restrição e (b) sementes de objeto após a restrição. Fonte: Autoria própria.	51
3.4	Visão geral do processo de automatização das sementes. Fonte: Autoria própria.	52
4.1	Exemplos de imagens de três dos volumes do conjunto de dados Medseg. Fonte: Autoria própria.	54
4.2	Caso de falsas sementes de objeto como consequência da escolha de um alto valor para β . (a) Imagem de entrada com as sementes tomadas com $\beta = 0.3$, (b) <i>ground truth</i> e (c) segmentação obtida pelo método proposto. Fonte: Autoria própria.	57
4.3	Caso em que poucas sementes implicam em uma segmentação de baixa qualidade, devido à adoção de um baixo valor para β . (a) Imagem de entrada com as sementes tomadas com $\beta = 0.05$, (b) <i>ground truth</i> e (c) segmentação obtida pelo método proposto. Fonte: Autoria própria.	57
4.4	(a) Imagem de entrada com as sementes tomadas com os parâmetros $r_{D1} = 20$, $r_{D2} = 18$ e $r_E = 8$, (b) <i>ground truth</i> e (c) segmentação pelo método proposto. Fonte: Autoria própria.	57
4.5	(a) Imagem de entrada com as sementes tomadas com os parâmetros $r_{D1} = 14$, $r_{D2} = 12$ e $r_E = 8$, (b) <i>ground truth</i> e (c) segmentação pelo método proposto. Fonte: Autoria própria.	58
4.6	(a) Imagens de entrada com as sementes tomadas com $\beta = 0.05$, (b) $r_{D1} = 14$, $r_{D2} = 12$ e $r_E = 8$, (c) <i>ground truth</i> e (d) segmentação pelo método proposto. Fonte: Autoria própria.	59

4.7	(a) Pesos de bordas vertical, (b) pesos de borda horizontal com pré-segmentação, (c) pesos de bordas vertical, e (d) horizontal sem pré-segmentação. Fonte: Autoria própria.	60
4.8	Relação entre o tempo de execução da técnica <i>Backpropagation</i> e as diferentes quantidades de amostras tomadas no gradiente. Fonte: Autoria própria.	61
4.9	Relação entre a convergência da função de custo no decorrer das iterações em relação a diferentes quantidades de amostras tomadas. Fonte: Autoria própria.	61
4.10	Comparação da segmentação de pulmão com lesões aparentes. (a) Imagens de entrada, (b) <i>ground truth</i> , (c) segmentações pelo método Random Walker, (d) segmentações pelo método <i>Learned Watershed</i> e (e) segmentação pelo método proposto. Fonte: Autoria própria.	63
4.11	Comparação entre o método LWS e o método proposto. (a) Imagens de entrada, (b) <i>ground truth</i> , (c) segmentações pelo método <i>Learned Watershed</i> e (d) segmentações pelo método proposto. Fonte: Autoria própria.	64
4.12	Comparação entre o método LWS, RW e o método proposto em cada imagem do volume CT de teste #1, através das métricas Precision, Recall, F-score, VOI, RI e ARI. Fonte: Autoria própria.	65
4.13	Comparação entre o método LWS, RW e o método proposto em cada imagem do volume CT de teste #2, através das métricas Precision, Recall, F-score, VOI, RI e ARI. Fonte: Autoria própria.	66
4.14	Comparação entre o método LWS, RW e o método proposto em cada imagem do volume CT de teste #3, através das métricas Precision, Recall, F-score, VOI, RI e ARI. Fonte: Autoria própria.	67

Lista de Tabelas

4.1	Comparação quantitativa entre os método RW, LWS e o método proposto (MP). Fonte: Autoria Própria.	62
-----	-----------------------------------------------------------------------------------------------------------	----

Sumário

1	INTRODUÇÃO	23
1.1	Contextualização do Tema	23
1.2	Objetivos e Propostas da Pesquisa	27
1.3	Contribuições	29
1.4	Estrutura da Dissertação	29
2	FUNDAMENTAÇÃO TEÓRICA	31
2.1	Segmentação de Imagem	31
2.1.1	Algoritmo Random Walker	31
2.1.2	Algoritmo Watershed	34
2.1.3	Modelo de Mistura Gaussiana	36
2.1.4	Expectativa Maximização	36
2.2	Aprendizado de Máquina	37
2.2.1	Aprendizado Profundo	38
2.2.2	Treinamento	39
2.2.3	Rede Neural Convolucional	40
2.2.4	Bordas Orientadas Convolucionais	42
2.2.5	Algoritmo <i>Watershed</i> via Aprendizagem Profunda	43
3	METODOLOGIA PROPOSTA	47
3.1	Descrição do Esquema de Aprendizagem	47
3.2	Automatização das Sementes	49
4	RESULTADOS E DISCUSSÕES	53
4.1	Base de Dados e Métricas de Avaliação	53
4.1.1	Base de Dados	53
4.1.2	Métricas de Avaliação	53
4.1.3	Resultados e Comparações	56
5	CONCLUSÃO	69
	REFERÊNCIAS	71

1 INTRODUÇÃO

1.1 Contextualização do Tema

O processo de decompor uma imagem em partes distintas é designado pelo termo *Segmentação* [10]. O processo de segmentar uma imagem se resume em criar agrupamentos de pixels através de verificação de semelhanças entre os objetos da imagem, ou ainda, a partir de algum critério pré-determinado entre os pixels, tais como cor, intensidade ou textura. Esses agrupamentos recebem o nome de partições, ou ainda *clusters*, podendo esses ser binários (ver Figura 1.1), ou ainda *multi-clusters* (ver Figura 1.2).



Figura 1.1: Segmentação binária de uma imagem, em que a raposa é a região de interesse a ser particionada em meio ao fundo da fotografia. Fonte: [2].

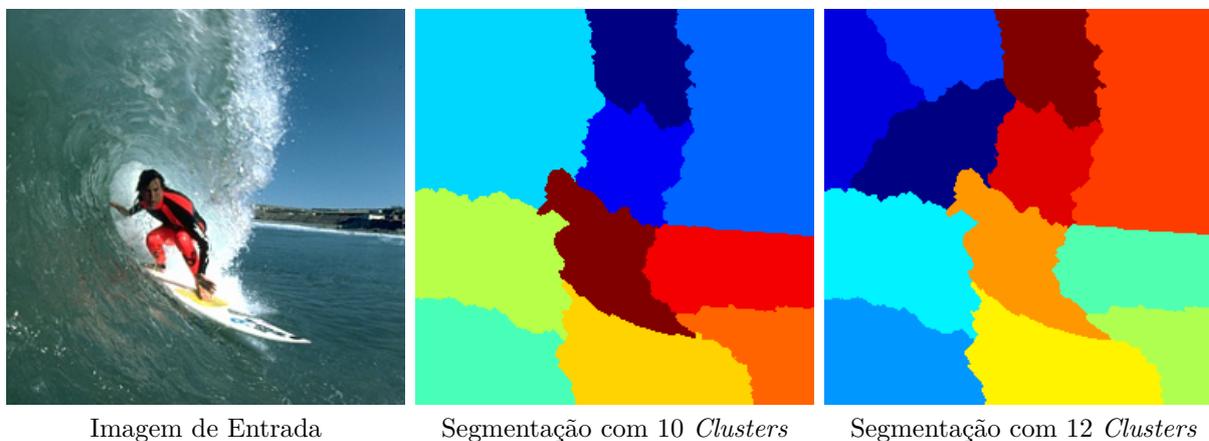


Figura 1.2: Segmentação do tipo *multi-clusters*, isto é, com varias partições. Fonte: [3].

A tarefa de segmentação é um componente vital em diferentes aplicações da área de Processamento Digital de Imagens (PDI) e Visão Computacional como, por exemplo, na colorização de fotografias [11] e no reconhecimento de padrões em Sensoriamento Remoto [12, 13]. Além disso, sua utilização na medicina é um importante pré-requisito para a detecção e tratamento de doenças, como patologias oculares [14] e pulmonares [15]. Ela também é aplicada na análise de imagens volumétricas de Tomografia Computadorizada, em inglês *Computed Tomography* (CT), que é uma etapa chave na intervenção assistida por computador, cirurgia guiada por imagens, etc (ver Figura 1.3).

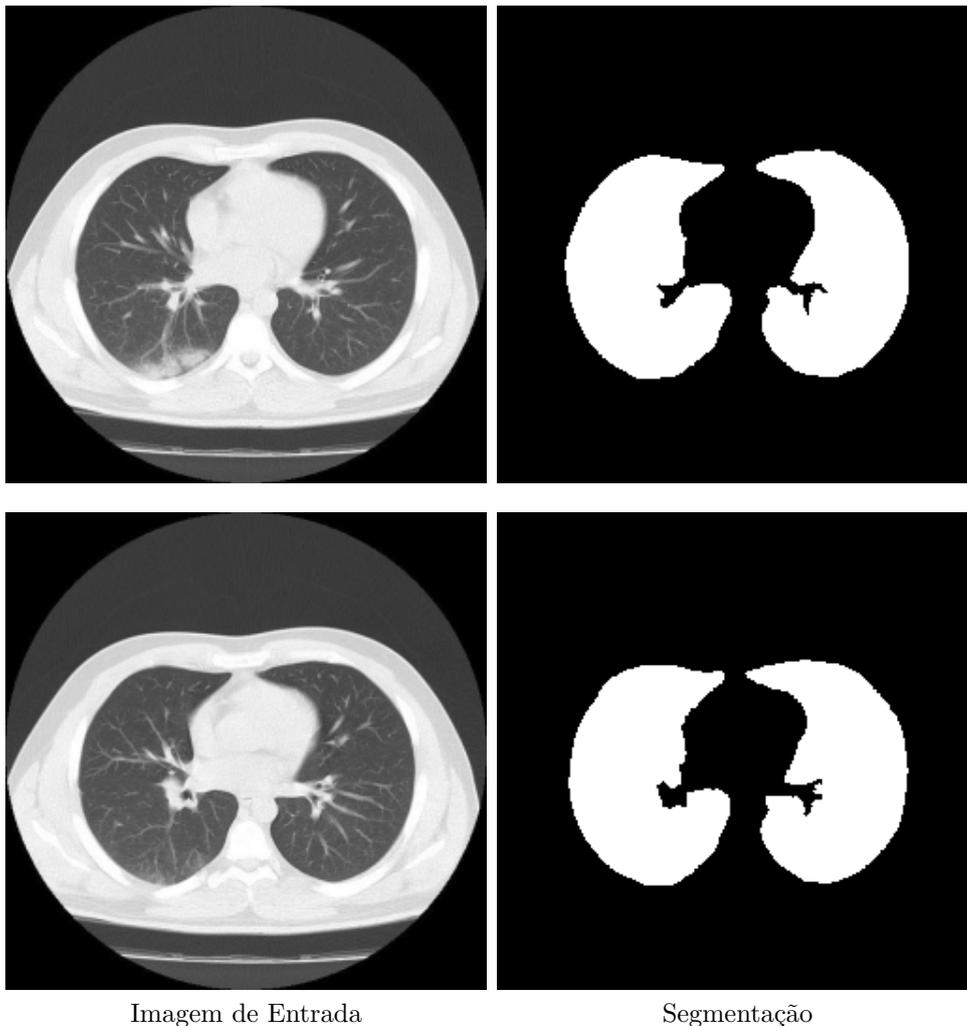


Figura 1.3: Segmentação de duas fatias adjacentes em uma imagem CT, do pulmão. Fonte: Autoria própria.

Visto os conceitos elementares acima, a referida pesquisa de mestrado procurou endereçar o tema *Segmentação de Imagens Digitais*, mais precisamente, no escopo de imagens do tipo CT. Assim, dada a grande abrangência e quantidade de obras a respeito do tema como um todo, este texto procurou focar em abordagens do tipo *segmentação interativa/semada de imagens*, isto é, um ramo de pesquisa em segmentação de imagens em que o usuário participa ativamente do processo de particionamento das mesmas ao fornecer as sementes (*seeds*), que guiam o segmentador na tarefa de clusterização das imagens (ver Figura 1.4 para uma ilustração).

Dentre os diversos métodos de segmentação semeada, duas abordagens foram explo-

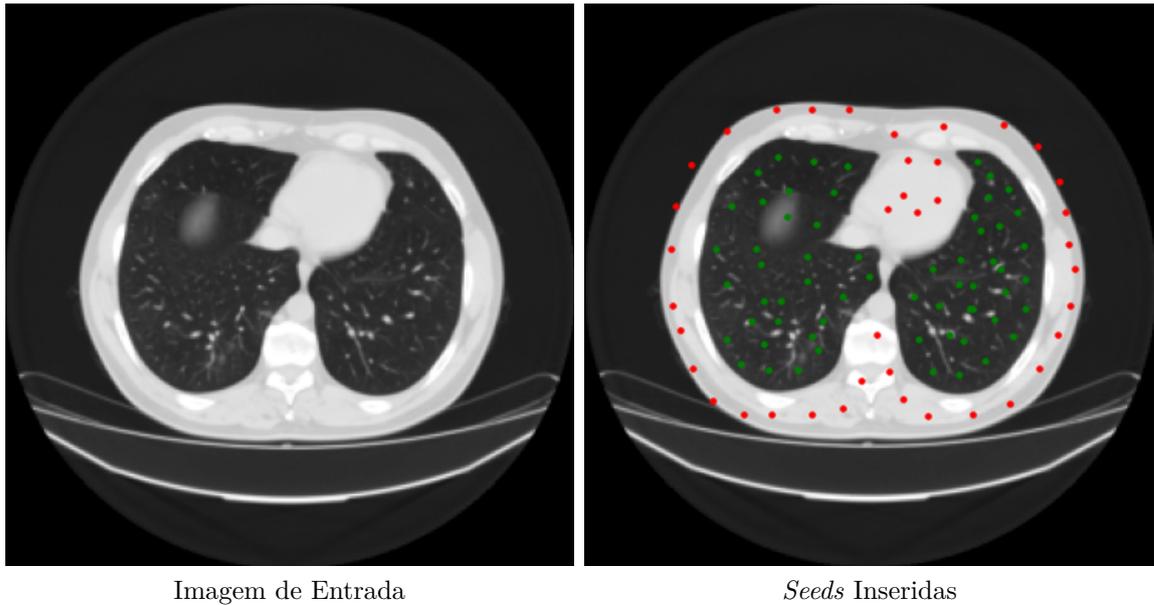


Figura 1.4: Ilustração das *seeds* (marcações feitas pelo usuário, em vermelho e verde) em uma imagem do tipo CT. Fonte: Autoria própria.

radas em nossa pesquisa: os modelos *Random Walker* (RW) [16] e *Watershed* (WS) [5].

O algoritmo RW, implementado por L. Grady, possui vasta aplicabilidade e é de fácil prototipação, cuja solução do problema de segmentação é dada em termos da resolução de um sistema linear esparso. O algoritmo foi readaptado por Zheng et al. [4] para segmentar regiões de tumor em mamografia, obtendo uma taxa de acurácia de 93,4% com testes feitos em 15 imagens contendo lesão (vide Figura 1.5). Embora a alta acurácia alcançada, o método não produz resultados acurados em imagens sem contornos bem definidos.

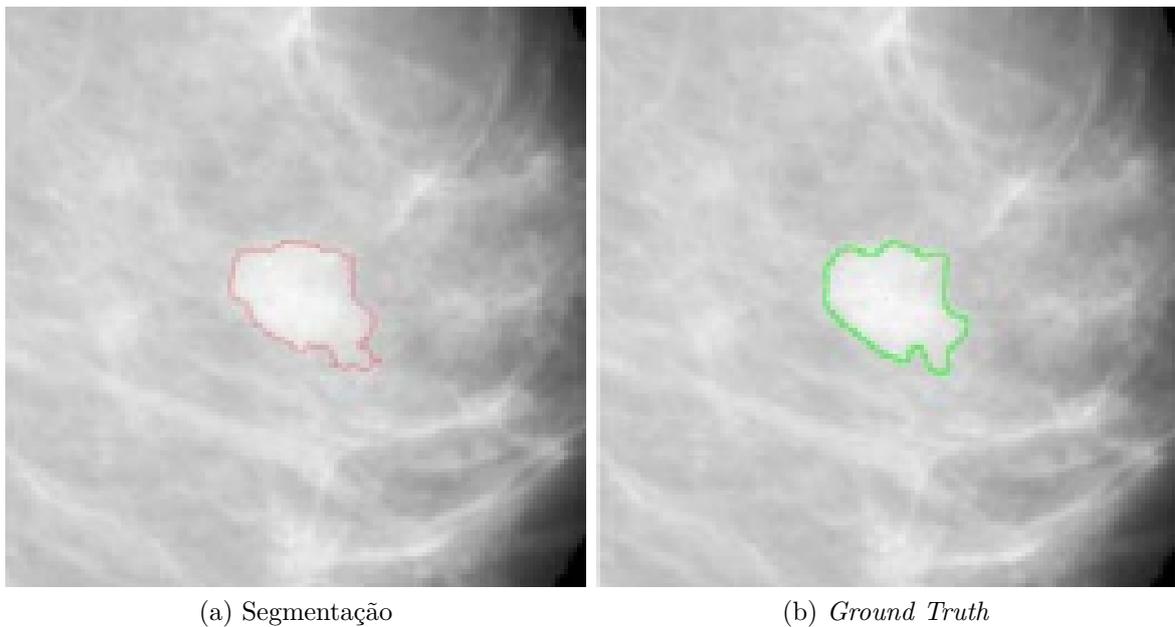


Figura 1.5: (a) Segmentação obtida com o método proposto por [4], e (b) por um agente humano (*ground truth*) dos tumores de mama. Fonte: [4].

Com relação ao algoritmo *Watershed* (WS), este representa os objetos contidos na

imagem como “bacias de captação”, realizando a segmentação dessas “bacias” e de suas linhas de nível (pontos igualmente prováveis de assumirem mais de um mínimo) [17]. O algoritmo WS é muito popular na literatura de Visão Computacional: possui ampla aplicabilidade e é especialmente útil para a extração de objetos sobrepostos. Por outro lado, o método não é plenamente efetivo no ajuste de objetos onde o gradiente é localmente irregular [18] (vide Figura 1.6). Além disso, chegar a um balanceamento entre a segmentação multiparticionada e alta acurácia é outro ponto de difícil obtenção na prática [19], pois as “bacias hidrográficas” podem sofrer com quedas e a degradação da solução nos “planaltos” da função de peso [20].

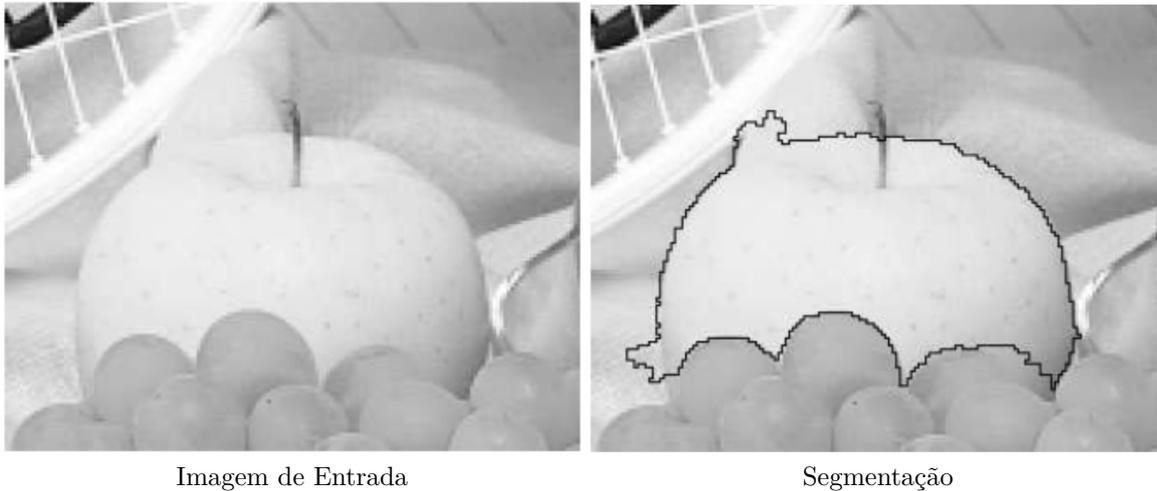


Figura 1.6: Segmentação obtida com o método *Watershed*. Fonte: [5].

Embora existam muitos métodos de segmentação propostos na literatura, nenhum deles é “universalmente efetivo” para todas as aplicações [21]. Por exemplo, há uma ampla gama de segmentadores de imagens de pulmão [22, 23, 24, 25, 26, 27, 28], os quais produzem resultados acurados no cálculo do volume pulmonar e na iniciação de sistemas de detecção auxiliados por computador, porém, estes falham quando há uma condição patológica, ou quando se tem a existência de anomalias, sejam elas de moderadas a marcadas ou até mesmo contendo padrões complexos de atenuação.

A fim de mitigar os desafios da segmentação apresentados acima, muitos trabalhos propuseram métodos baseados no chamado Aprendizado Profundo [29, 6, 30]. Aprendizado Profundo – em inglês, *Deep Learning* – é um subtema da Inteligência Artificial que é baseado na ideia de que sistemas podem aprender a partir dos dados, identificar padrões, e tomar decisões com o mínimo de intervenção humana.

Nessa linha de estudo, Can et al. [31] propuseram utilizar o algoritmo *Random Walker* integrado a uma *Rede Totalmente Convolutiva*, em inglês *Fully Convolutional Network* (FCN), para alcançar segmentações semi-automatizadas.

Outra abordagem bastante efetiva é usar métodos clássicos não supervisionados como estimativa inicial para um processo de aprendizagem profundo. Em Rajchl et al. [32], uma segmentação inicial a partir do modelo *Grabcut* [3] é usada para este fim, e o desempenho de segmentação é então melhorado a partir de de uma *Rede Neural Convolutiva* (CNN) mais densa chamada *Campo Aleatório Condicional Totalmente Conectado*, em inglês *Fully Connected Conditional Random Field* (CRF), para pós-processamento. Da mesma forma, Zhang et al. [33] produzem resultados de segmentação baseados no algoritmo *K-means*, que é usado para treinar uma rede de segmentação profunda em regiões pulmonares

císticas.

Já Steffen et al. [6] propuseram um método para aprender os pesos das bordas das imagens na tarefa de segmentação do método *Watershed* de uma forma estruturada, isto é, tornando o algoritmo adaptativo a fim de endereçar imagens com partições de difícil segmentação, como por exemplo, contornos mais estreitos. Entretanto, o método classifica erroneamente os alvos em casos onde as imagens apresentam contornos fracos, isto é, de menor espessura (vide Figura 1.7 para uma ilustração).

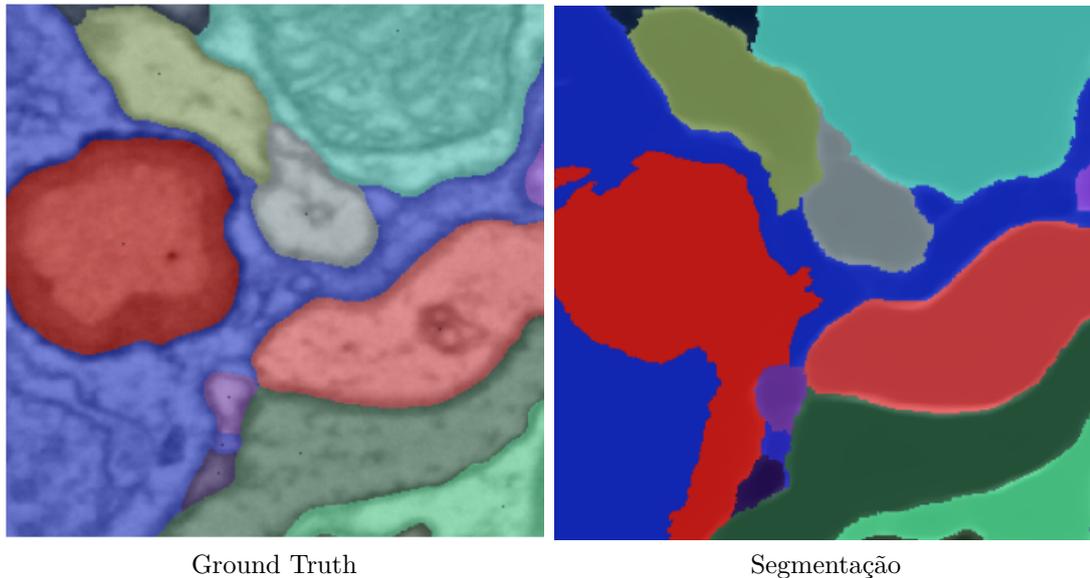


Figura 1.7: Segmentação de baixa qualidade utilizando o método *Learned Watershed* por conta de “fraco contorno” (*weak edges*). Fonte: [6].

Ao segmentar uma imagem do pulmão a partir de representações volumétricas do tipo CT, é necessário considerar grandes variações no formato do pulmão no decorrer das fatias – conjunto de imagens 2D denominadas de *slices* – e lesões existentes no órgão alvo, por exemplo fibrose pulmonar, derrame pleural, tumor e muitos outros, alterando não só a anatomia do órgão como também as intensidades de tonalidades da imagem no local das lesões, dificultando assim que a segmentação seja precisa.

1.2 Objetivos e Propostas da Pesquisa

Nesta dissertação, delineamos como objetivo primário a realização da segmentação binária das fatias que compõem imagens médicas de volumes do tipo CT, do pulmão, sendo ele saudável ou não (vide Figura 1.8). Para este fim, foi empregado o conjunto de dados fornecido pela *MedSeg*[34], a qual disponibilizou volumes CT do pulmão de pacientes acometidos pela Covid-19.

Desta forma, a pesquisa visou superar os obstáculos levantados e discutidos ao longo deste capítulo sobre a problemática de segmentação de fatias de imagens do tipo CT. Para tal, foi proposta uma nova metodologia de segmentação de imagens, que tomou como base algorítmica o aprendizado profundo. Mais precisamente, uma Rede Neural Convolutiva, em inglês *Convolutional Neural Network* (CNN), procura aprender os pesos de bordas das imagens para posterior uso no algoritmo Random Walker, com a finalidade de maximizar

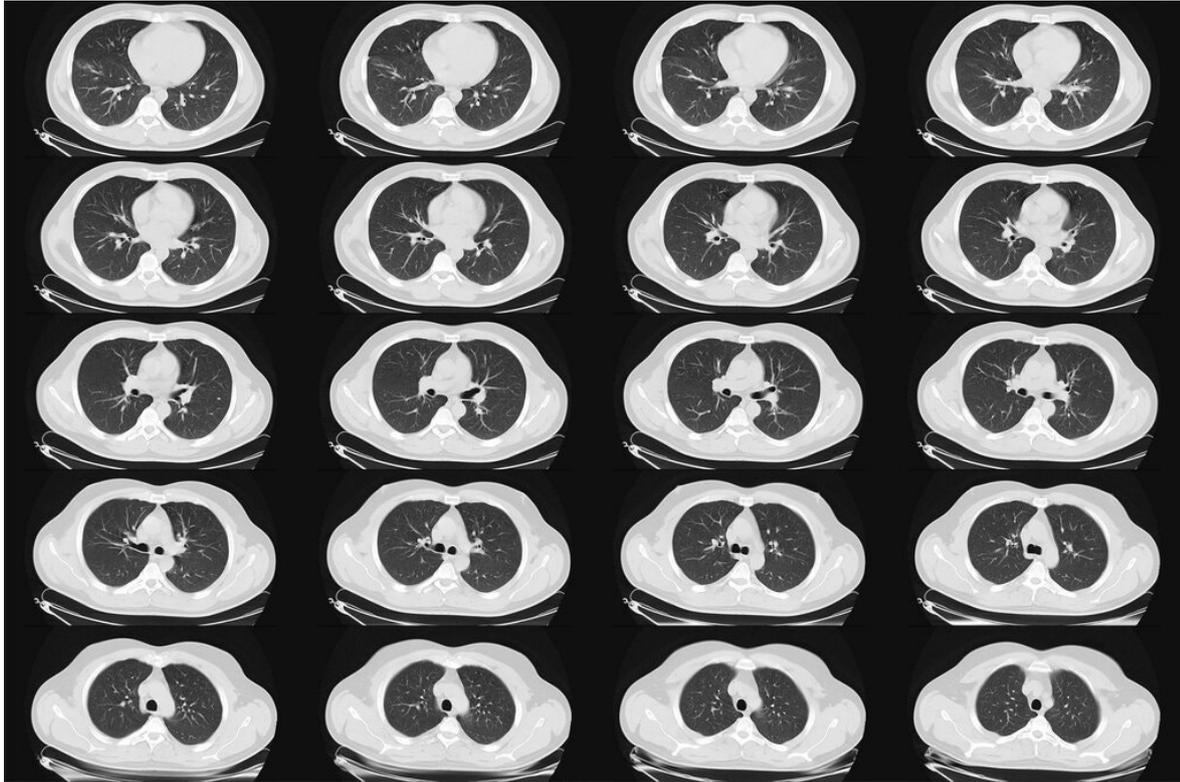


Figura 1.8: Fatias de uma CT dos pulmões de um homem adulto saudável. Fonte: <<https://www.nytimes.com/2021/03/09/health/lung-cancer-smoking-screenings-black-women-younger-adults.htm>>.

a segmentação independente da imagem do pulmão estar ou não apresentando sintomas da doença (vide Figura 1.9 para uma ilustração).

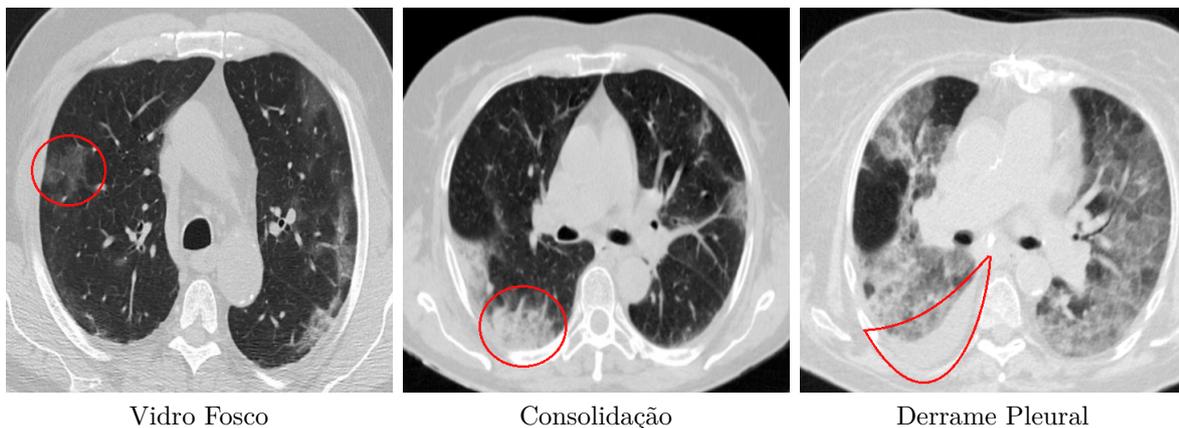


Figura 1.9: Exemplos de lesões representadas por vidro fosco, consolidação e derrame pleural nas fatias de um volume CT. Fonte: Autoria própria.

A fim de filtrar ruídos e desconsiderar tecidos que não sejam os do pulmão e, assim, melhorar a precisão com que a CNN irá aprender os pesos das bordas, uma pré-segmentação é aplicada nas fatias do volume CT a partir do modelo denominado Mistura Gaussiana.

Além disso, posto que os métodos clássicos da literatura procuram treinar seus segmentadores a partir de uma grande quantidade de iterações, quanto maior o tempo de execução entre um ciclo de treinamento e outro, maior será o tempo de espera para a

obtenção do algoritmo aprendido. Desta forma, foi implementada uma técnica otimizada de esparsamento do gradiente a fim de aumentar a velocidade de cada iteração sem perder a qualidade do aprendizado, e assim diminuir o tempo de processamento ao longo das iterações de treinamento, ou seja, foi almejado balancear os aspectos da precisão da segmentação e performance computacional do pipeline de aprendizado proposto, sendo este um objetivo secundário do trabalho.

E por fim, visto que volumes CT tendem a ter uma alta quantidade de fatias, podendo chegar às centenas delas, semear fatia por fatia seria uma tarefa bastante trabalhosa e demorada. Portanto, a fim de automatizar o processo, a metodologia proposta utiliza uma estratégia específica de sementeção quase-automática, em que as sementes de apenas uma fatia são fornecidas pelo usuário, e o conjunto de sementes de todas as demais fatias são então induzidas automaticamente.

1.3 Contribuições

As principais contribuições deste trabalho são:

1. Desenvolvimento de um algoritmo de aprendizado profundo de segmentação semeada quase-automático, onde o usuário fornece o conjunto de sementes de apenas uma fatia do volume CT, e a abordagem induz a criação de mapas de sementes nas demais fatias de forma automática.
2. Utilização de um módulo de pré-tratamento das imagens, a fim de filtrar a região alvo de ruídos indesejáveis e tecidos que não sejam do pulmão, melhorando assim o aprendizado dos pesos das bordas das imagens.
3. Desenvolvimento de uma metodologia de aprendizado dos pesos das bordas, os quais são empregados em um algoritmo que consiste em uma rede neural profunda integrada ao segmentador *Random Walker*.
4. Uma estratégia de otimização de esparsamento do gradiente, aumentando assim a velocidade de treinamento sem perder a qualidade do aprendizado.
5. O algoritmo foi treinado e avaliado no conjunto *MedSeg Covid Dataset* [34], onde foi observado que o mesmo apresentou resultados altamente competitivos com trabalhos recentes de segmentação de pulmão com lesões.

1.4 Estrutura da Dissertação

Esta dissertação de mestrado está organizada em 5 capítulos, como descritos abaixo:

Capítulo 2 [Fundamentação Teórica]: apresenta os conceitos para o entendimento de segmentação de imagens e do aprendizado de máquina, os quais são implementados/empregados no método proposto.

Capítulo 3 [Metodologia proposta]: descreve o método de aprendizado profundo proposto para a segmentação de imagens, demonstrando o processo de aprendizado e a estratégia de validação do algoritmo.

Capítulo 4 [Resultados e Discussões]: demonstra os resultados das técnicas aplicadas no método proposto, assim como sua eficácia, comparando-o qualitativamente e quantitativamente com outros métodos.

Capítulo 5 [Conclusão]: resume o conteúdo da dissertação e destaca suas principais contribuições e conclusões.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Segmentação de Imagem

Uma etapa importante no processamento de imagens digitais para analisar o que está presente em uma imagem é dividir seu conteúdo entre pixels de diferentes grupos. Tal processo, de decompor uma imagem em partes menores, designa-se por *Segmentação* [10].

A segmentação de uma imagem digital pode ser realizada de duas maneiras distintas:

1. De forma automática, através de métodos que interpretam a imagem e criam agrupamentos de acordo com algum tipo de semelhança entre os pixels, os quais são chamados de métodos de segmentação automática;
2. Com a participação do usuário, que também ocorre de acordo com algum tipo de semelhança entre os pixels, entretanto a segmentação ocorre a partir de pixels demarcados pelo usuário (sementes), que facilitam o processo de segmentação e a obtenção do resultado esperado, sendo esse grupo de métodos chamado de segmentação interativa.

Conforme discutido no capítulo anterior, focaremos no estudo e exploração dos métodos da última categoria.

2.1.1 Algoritmo Random Walker

Métodos de segmentação interativa (baseados em “sementes”) apresentam um desempenho satisfatório na tarefa de particionamento de imagens com pouca participação do usuário. O algoritmo *Random Walker* (RW) é um método de segmentação do tipo semeado proposto por L. Grady [16], que teve como principal propósito a segmentação de imagens na área médica, mais precisamente, no estudo de ressonâncias. Para o caso binário, o modelo toma como parâmetro chave a ideia de que, dado um conjunto de sementes fornecidas e rotuladas como *fundo* e *objeto* pelo usuário, qual deve ser a probabilidade de um caminhante, partindo de um pixel não semeado e de forma aleatória, chegar a uma semente rotulada como *objeto* ou como *fundo*. Desta maneira, tal pixel (não semeado) receberá o rótulo em que o caminhante tiver maior probabilidade de chegar até que todos os pixels estejam rotulados. Esta abordagem também pode ser generalizada para uma quantidade $k \in \mathbb{N}$ de rótulos.

Em termos matemáticos, considere o conjunto V de nós do grafo sobre uma malha quadriculada tal como apresentado na Figura 2.1. O algoritmo RW consiste na ideia de que, dado um conjunto de sementes fornecidas e rotuladas pelo usuário como *fundo* e *objeto*, em inglês *Background* (B) e *Foreground* (F), respectivamente, qual é a probabilidade

de um caminhante partindo de um pixel Y , não rotulado e de forma aleatória, chegar a uma semente rotulada por B ? E por F ?

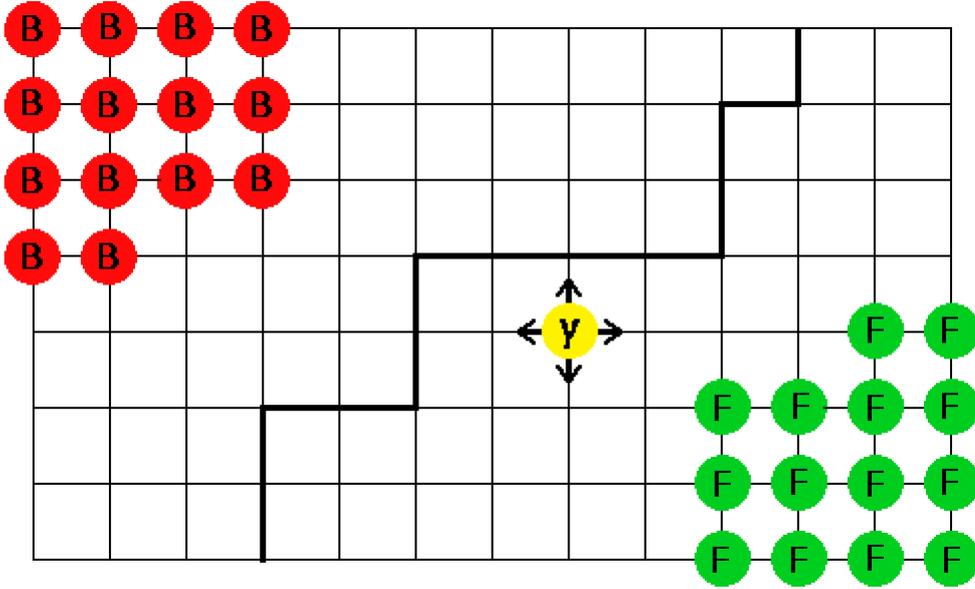


Figura 2.1: Representação de um grafo sobre uma malha quadriculada em que o caminhante aleatório se locomove conforme a concepção do método *Random Walker*. Fonte: Autoria própria.

Em arestas que possuem ao menos um nó de fronteira, o caminhante tem sua probabilidade diminuída drasticamente de atravessá-las, dificultando assim a passagem através dessa “barreira”. Logo, na Figura 2.1, onde tal fronteira está destacada, os pixels localizados do lado direito têm maior chance de receberem o rótulo F , uma vez que a fronteira dificulta a passagem do caminhante, de atravessar para o lado esquerdo até os pixels de rótulo B , e analogamente para os pixels do lado esquerdo da fronteira, de receberem os rótulos B . Portanto, tal resultado leva à segmentação ilustrada na Figura 2.2.

Em termos da formulação do problema, dada uma imagem I , definimos um grafo $G = (V, E, W_E)$, em que V é o conjunto de vértices, de tamanho N , onde cada $i \in V$ corresponde ao pixel P_i da imagem I , E é o conjunto de arestas, que ligam os pixels no grafo-base, e neste caso, considera-se uma vizinhança de 4 pixels, e W_E o conjunto de pesos das arestas, com cada $w_{ij} \in W$ sendo o peso da aresta que liga os nós i e j .

Para cada nó $i \in V$ do grafo, o conjunto de sua vizinhança é definido por $N(i) = \{j : (i, j) \in E\}$, ou seja, a lista de vértices j que tenham arestas adjacentes ao nó i , como exemplificado na Figura 2.3. O algoritmo RW consiste em minimizar a seguinte função de energia em $x = (x_1, x_2, \dots, x_N)^T$, em que cada componente do vetor representa um pixel P_i (ou ainda um nó i do grafo) na imagem:

$$E_{RW}(x) = \frac{1}{2} \sum_{(i,j) \in E} w_{ij} (x_i - x_j)^2, \quad (2.1)$$

Por outro lado, considerando a matriz Laplaciana definida por:

$$L_{i,j} = \begin{cases} -w_{i,j} & \text{se } j \in N(i) \\ \sum_{k \in N(i)} w_{i,k} & \text{se } i = j \\ 0 & \text{caso contrario} \end{cases}, \quad (2.2)$$

Portanto, podemos decompor a equação (2.3) por:

$$\begin{aligned}
 E_{RW}(x) &= \frac{1}{2} \begin{bmatrix} x_S^T & x_U^T \end{bmatrix} \left[\begin{array}{c|c} L_S & R \\ \hline R^T & L_U \end{array} \right] \begin{bmatrix} x_S \\ x_U \end{bmatrix} \\
 &= \frac{1}{2} \begin{bmatrix} x_S^T L_S + x_U^T R^T + x_S^T R + x_U^T L_U \end{bmatrix} \begin{bmatrix} x_S \\ x_U \end{bmatrix} \\
 &= \frac{1}{2} (x_S^T L_S x_S + 2x_U^T R^T x_S + x_U^T L_U x_U), \tag{2.5}
 \end{aligned}$$

em que x_S e x_U são chamados de mapas de probabilidade dos vértices semeados e não semeados, respectivamente. Entretanto, os valores dos componentes do vetor x_S já são conhecidos, uma vez que são os nós já semeados cujos rótulos são fornecidos pelo usuário e, portanto, sem perda de generalidade, podemos definir $E_{RW}(x) = E_{RW}(x_U)$. Logo, diferenciando E_{RW} em relação a x_U :

$$\frac{\partial E_{RW}}{\partial x_U}(x_U) = \frac{1}{2} (2R^T x_S + 2L_U x_U) = 0. \tag{2.6}$$

Assim, o problema de encontrar os mínimos da função E_{RW} equivale a resolver o seguinte sistema linear com $|U|$ incógnitas em x_U :

$$L_U x_U = -R^T x_S. \tag{2.7}$$

Com a solução da equação (2.7), a segmentação pode ser obtida aplicando um algoritmo de agrupamento convencional, como o *K-means*, o algoritmo de *Otsu* ou outros esquemas de limiarização aos valores de x_U . Por exemplo, a rotulação final da imagem pode ser obtida a partir de uma seleção do tipo *winner-take-all* do rótulo com valor máximo para cada vértice:

$$v \longmapsto \arg \max_{a \in \{x_F, x_B\}} x_{v,a}, \forall v \in U, \tag{2.8}$$

em que x_F e x_B são os valores base referentes aos rótulos marcados como objeto e fundo na imagem, respectivamente.

2.1.2 Algoritmo Watershed

Bacias hidrográficas são regiões separadas por linhas chamadas de “divisores de água” (*Watershed*), que constituem todas as fontes de uma bacia hidrográfica. Neste caso, água da chuva, de rios ou afluentes e até mesmo vazões de origem subterrânea, tendem a escoar naturalmente do ponto mais alto de uma dada região até o ponto mais baixo, onde acontece o acúmulo na bacia. O que difere duas bacias vizinhas é para onde a água está escoando, ou seja, os limites da região em que toda a água escoa para o mesmo lugar [2]. A Figura 2.4 ilustra um modelo simples de bacias hidrográficas.

O algoritmo *Watershed* parte do princípio de que qualquer imagem pode ser vista como uma superfície topográfica, com diversas bacias. E então diversas “inundações” são criadas, a partir das sementes, e se espalham pela imagem. À medida que a água sobe, dependendo dos picos próximos, a água de diferentes vales começará a se fundir. Para evitar isso, constrói-se barreiras nos locais onde a água iria se fundir. Ao final,

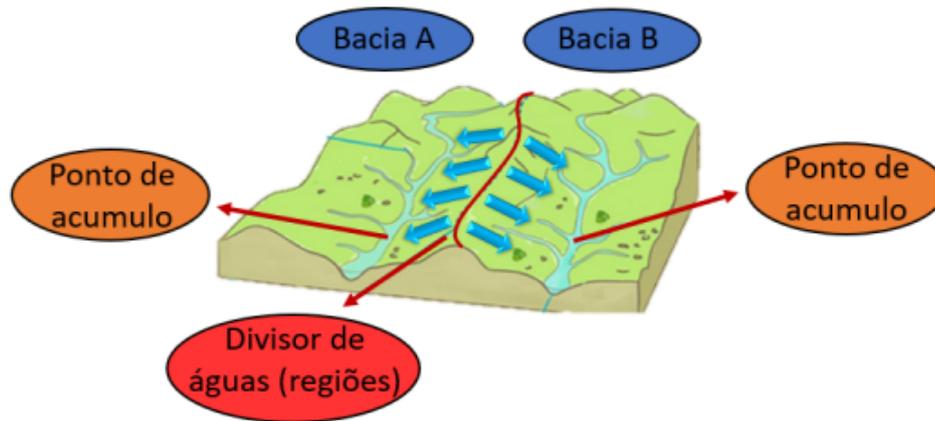


Figura 2.4: Ilustração de duas bacias vizinhas, seu divisor, e seus pontos de acumulo. Fonte: [2].

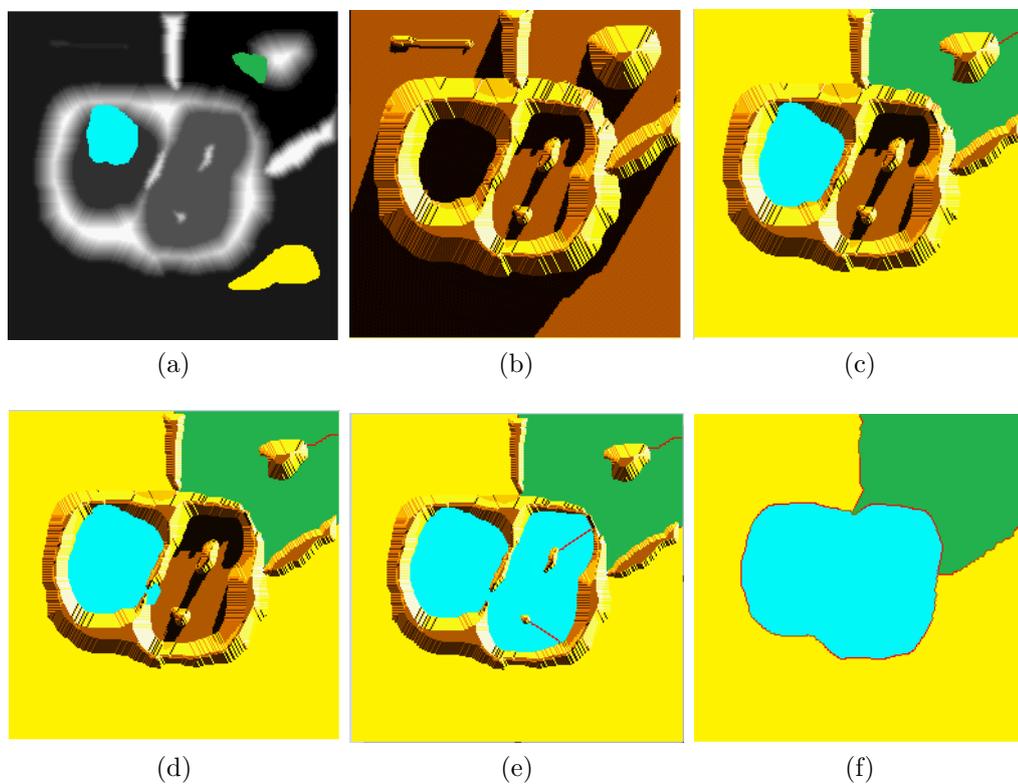


Figura 2.5: (a) Imagem de entrada com as sementes, (b) superfície topográfica da imagem de entrada, de (c) a (e) os processos de “inundação” do algoritmo Watershed até a segmentação final (f). Fonte: Adaptado de <<http://www.cmm.mines-paristech.fr/~beucher/wtshed.html>>.

esse processo gera a divisão da imagem em diferentes agrupamentos de pixels, ou seja, a segmentação da imagem, enquanto que as barreiras erguidas são as fronteiras de cada região. A Figura 2.5 ilustra o processo de “inundação” iniciado em 3 pontos diferentes definidos pelas sementes.

O método descrito acima foi inicialmente proposto na área de segmentação de imagens por Cousty et al. [5], já os cálculos dos divisores de águas, ou seja, dos limites das regiões da imagem, pode ser realizado, por exemplo, com a ajuda de um dentre dois algoritmos

clássicos: o Algoritmo de *Kruskal* [35] ou o Algoritmo de *Prim* [36].

2.1.3 Modelo de Mistura Gaussiana

O modelo de mistura gaussiana, mais conhecido por *Gaussian Mixture Model* (GMM), é um método de pré-segmentação automático que se baseia na combinação linear de filtros gaussianos e emprega o algoritmo *Expectation Maximization* (EM).

Conforme apresentado no trabalho de Fu e Wang [37], o GMM é definido como uma combinação do tipo M-linear da função de densidade gaussiana, ou seja,

$$p(x) = \sum_{i=1}^M \pi_i \mathcal{N}_i(x|\mu_i, C_i) \quad (2.9)$$

onde $\mathcal{N}_i(x|\mu_i, C_i)$ denota a distribuição normal relacionada ao vetor de média μ_i e a matriz de covariância C_i . Já π_i é o coeficiente de mistura para o i -ésimo grupo, que pode ser também denotado como uma probabilidade, a priori, e deve satisfazer a igualdade

$$\sum_{i=1}^M \pi_i = 1, \text{ com } 0 \leq \pi_i \leq 1. \quad (2.10)$$

Porém, ao aplicar tal método na prática, os valores de μ_i , C_i e π_i não são conhecidos. Desta forma, precisamos estimá-los de alguma maneira. Uma abordagem eficiente para encontrar tais parâmetros é aplicar o algoritmo EM, a qual é apresentada a seguir.

2.1.4 Expectativa Maximização

Introduzido por Dempster et al. [38], o algoritmo EM é muito eficiente no papel de estimar parâmetros desconhecidos, o que é comum em diversos tipos de problemas de agrupamento.

O algoritmo EM é dividido em duas etapas. Na chamada etapa E, *Expectation*, é definido um conjunto de dados completo que baseia-se nos dados observados para calcular o valor esperado da função de máxima verossimilhança, enquanto que na etapa M, *Maximization*, usa-se os dados obtidos da função de verossimilhança para definir os novos valores para os parâmetros desconhecidos anteriormente. As duas etapas se intercalam, desta forma, até a convergência do método iterativo.

Etapa de Expectativa

Inicialmente, os valores de π_i , μ_i e C_i , com $i = 1, \dots, M$, são estimados por outro algoritmo como, por exemplo, o clássico algoritmo *K-Means*. Então, a etapa E de um modelo de Mistura Gaussiana com M componentes Gaussianas e com N dados x_k pode ser executada da seguinte forma:

$$\mathcal{Q}(\theta|\theta^{(t)}) = \sum_{l=1}^M \sum_{i=1}^N \gamma(Z_{il}) \log \pi_l + \sum_{l=1}^M \sum_{i=1}^N \gamma(Z_{il}) \log \mathcal{N}(x_i|\theta_l), \quad (2.11)$$

em que $\theta_l = (\mu_l, C_l)$, $\mathcal{N}(x_i|\theta_l)$ representa a probabilidade da distribuição para a l -ésima componente Gaussiana de x_i , e $\gamma(Z_{il})$ é a probabilidade posterior da l -ésima componente Gaussiana, que segundo Bishop [39], é representada por:

$$\gamma(Z_{il}) = \frac{\pi_l \mathcal{N}(x_i|\theta_l)}{\sum_{j=1}^M \pi_j \mathcal{N}(x_i|\theta_j)}. \quad (2.12)$$

Etapa de Maximização

Após os dados serem adquiridos na etapa E, eles são então usados na reestimativa dos parâmetros na etapa M, que são adquiridos pelos seguintes passos:

$$\pi_l^{(t+1)} = \frac{1}{N} \sum_{i=1}^N \gamma(Z_{il}), \quad (2.13)$$

$$\mu_l^{(t+1)} = \frac{1}{N\pi_l^{(t+1)}} \sum_{i=1}^N x_i \gamma(Z_{il}), \quad (2.14)$$

$$C_l^{(t+1)} = \frac{1}{N\pi_l^{(t+1)}} \sum_{i=1}^N (x_i - \mu_l^{(t+1)})^2 \gamma. \quad (2.15)$$

E por fim, as etapas E e M se repetem até que o critério de parada definido seja satisfeito. Exemplos visuais para diferentes valores do parâmetro k , de número de *clusters*, são evidenciados na Figura 2.6.

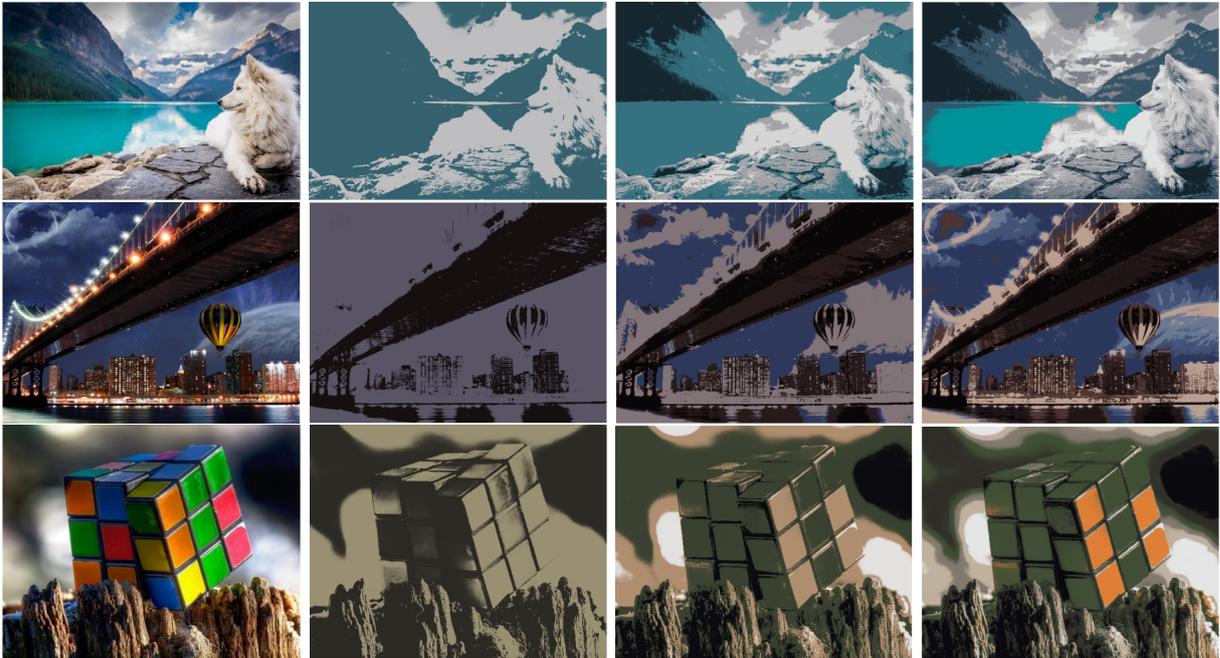


Figura 2.6: Resultados do processamento de imagens via GMM através do Algoritmo EM para diferentes parâmetros k . Da esquerda para a direita: imagens de entrada, e os resultados obtidos tomando-se $k = 2$, $k = 4$ e $k = 7$. Fonte: [2].

2.2 Aprendizado de Máquina

O Aprendizado de Máquina, em inglês *Machine Learning*, é uma linha de pesquisa de análise de dados que automatiza a construção de modelos analíticos. É um ramo da Inteligência Artificial, baseado na ideia de que sistemas podem aprender com os dados, identificar padrões, e tomar decisões com o mínimo de intervenção humana. O Aprendizado de Máquina é separado em duas categorias, sendo elas o Aprendizado de Máquina Supervisionado, e o Aprendizado de Máquina Não Supervisionado.

No Aprendizado de Máquina Supervisionado, dado o conjunto de dados, tem-se os dados e as classes de cada item. Assim, após o treinamento, dado um item do conjunto, o modelo deve ser capaz de colocá-lo em alguma das classes disponíveis. O Aprendizado de Máquina Supervisionado é separado entre as tarefas de classificação e regressão.

- **Classificação:** Ocorre quando o problema analisado tem a resposta em classes bem definidas e discretas. Exemplo: dado tamanho e idade do tumor, classificar em câncer ou não-câncer.
- **Regressão:** Ocorre quando o problema analisado não tem classes bem definidas, e a resposta é sempre um número real que pode estar em um intervalo muito grande, ou seja, um problema definido em um escopo contínuo. Exemplo: prever o preço de uma casa, dadas as características como metros quadrados, número de cômodos, região onde o imóvel está localizado, entre outras características.

Já no Aprendizado de Máquina Não Supervisionado, a máquina é treinada com dados sem informações de saída, e o objetivo é agrupar os elementos baseados em características similares ou em características que os tornam únicos. Esses grupos são chamados de *clusters*. Com o aprendizado não supervisionado, o objetivo não é procurar por uma resposta única, aproximada, específica ou certa. Ao invés disso, é relacionar cada elemento do grupo pela similaridade das características ou do comportamento entre os membros, e pelas diferenças com os outros grupos.

2.2.1 Aprendizado Profundo

O Aprendizado Profundo, em inglês *Deep Learning*, é um conjunto de técnicas relacionada tanto ao Aprendizado de Máquina quanto à Visão Computacional, que apresenta resultados bastante satisfatórios, na maioria dos casos. Pode ser entendido como uma classe de algoritmos que explora diversas camadas de informações a partir de estruturas não-lineares. Basicamente, o aprendizado profundo trata do aprendizado de múltiplos níveis de representação e abstrações que ajudam a construir um conhecimento acerca dos dados como imagens, sons, e texto.

Rosenblatt [40] propôs um neurônio artificial, com capacidade de encontrar superfícies de separação para características linearmente separáveis. Esse neurônio é denominado de *Perceptron*. Ele pode se conectar com outras unidades para configurar redes mais complexas, e assim, possibilitando o aprendizado de padrões não-lineares. A Figura 2.7 ilustra um modelo do tipo *Perceptron*, onde as entradas ($X_1, X_2, X_3, \dots, X_N$) são as saídas de outros neurônios, ou os dados de entrada quantizados. É então efetuado um produto escalar com os pesos ($W_1, W_2, W_3, \dots, W_N$) e um parâmetro de viés θ é somado, resultando em um valor Y' . Por fim, Y' é passado para a uma função de ativação $S(Y')$ de modo a gerar a saída Y do *Perceptron*.

Na Figura 2.8, são apresentadas duas arquiteturas de redes neurais, onde as camadas de entrada representam os dados em forma numérica, as camadas de saída as respostas processadas, e as camadas ocultas originam a sequência de decisões entre as camadas de entrada e de saída. Com uma camada *Perceptron*, oculta, tem-se apenas uma rede neural, de forma que o modelo aprenderá apenas relações lineares. Já com duas ou mais camadas ocultas, tem-se uma rede neural profunda, e então o modelo poderá aprender relações mais complexas, isto é, não-lineares entre as camadas de entrada e saída.

A qualidade de um modelo baseado em redes neurais depende da sua capacidade de aprender as características dos dados apresentados na etapa de treinamento. Nessa etapa,

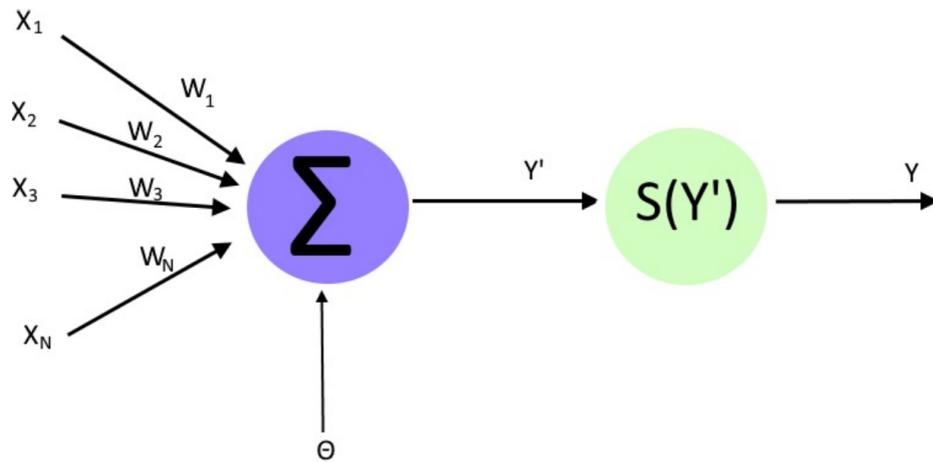


Figura 2.7: Modelo *Perceptron*. Fonte: [7].

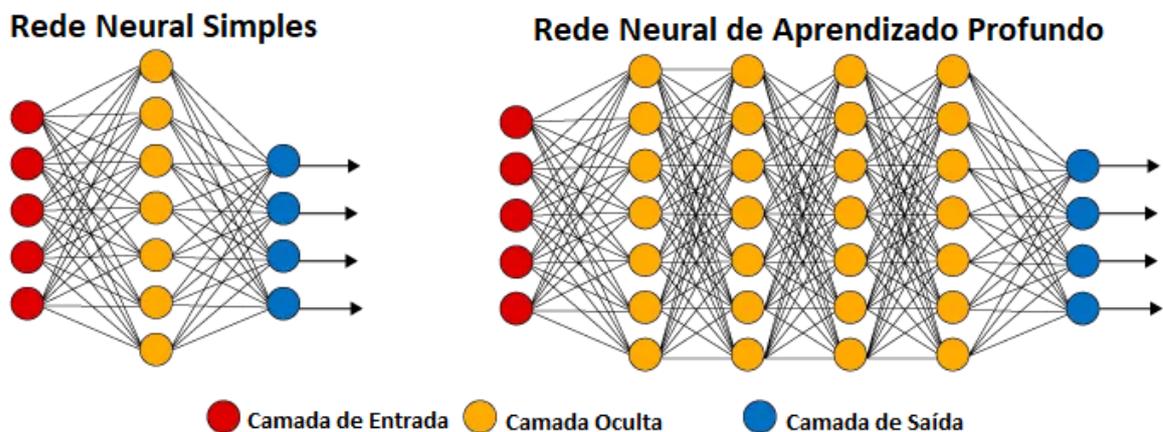


Figura 2.8: Rede Neural Simples e Rede Neural de Aprendizagem Profundo. Fonte: <<http://deeplearningbook.com.br>>.

uma rede neural percorre todo o conjunto de dados e quantifica seu aprendizado na forma de pesos, sendo esses valores chamados de parâmetros. Um método bastante usual de treinamento é o *Backpropagation*, o qual estima os parâmetros da rede propagando o erro para as camadas anteriores.

2.2.2 Treinamento

Ao treinar uma rede neural, seus parâmetros são inicialmente tomados aleatoriamente. Entretanto, utiliza-se de uma função de custo para avaliar o quão distante os resultados produzidos pela rede estão dos resultados alvos e, em seguida, estimar novos parâmetros por meio do algoritmo *Backpropagation*.

Função de Custo

Para o ajuste dos parâmetros da rede neural, uma função de custo compara a sua saída com os resultados pretendidos (*ground truth*), calculando assim um erro. A escolha da função de custo é importante para a resolução do problema desejado. De fato, vários trabalhos na literatura propuseram funções para aplicações específicas como, por exemplo, a Perda Tripla [41], utilizada em reconhecimento facial. Por outro lado, muitas funções

de custo podem ser utilizadas em tarefas mais genéricas, como é o caso do chamado *Erro de Entropia Cruzada*:

$$\text{CE}(x^*, x) = -\frac{1}{N} \sum_{i=1}^N x_i^* \log(x_i), \quad (2.16)$$

em que N representa a quantidade de amostras no conjunto de treinamento, e x_i e x_i^* representam, respectivamente, os valores preditos pela rede e aquele observado (real). Um alto resultado indica que o modelo apresenta valores muito distantes do correto, enquanto que pequenos apresenta valores próximos.

Backpropagation

Em um rede neural profunda, os parâmetros de cada neurônio do tipo *Perceptron* são estimados continuamente a fim de que a rede apresente resultados mais precisos a partir da minimização da função de custo. Este é um processo iterativo, de tal modo que a cada etapa, novos dados são fornecidos para a entrada da rede, e a função de erro é calculada e os parâmetros ajustados. Neste caso, o método do Gradiente Descendente é utilizado na correção dos parâmetros: o erro é propagado de volta para cada neurônio do modelo. Tal abordagem é chamada de *Backpropagation* [42].

O método do Gradiente Descendente, como o próprio nome diz, utiliza o gradiente (variação) da função de custo a fim de minimizá-la. No cálculo vetorial, o gradiente de uma função é um vetor que mostra a direção e o sentido de maior incremento da função, considerando um ponto no espaço. Logo, o sentido contrário do vetor gradiente guia as estimativas dos parâmetros da rede neural até um ponto de mínimo.

A Figura 2.9 ilustra a curva de uma função de custo com apenas uma dimensão e quatro configurações de parâmetros. Observe que os valores de w_1 , w_2 e w_3 não são ótimos, portanto, por meio do gradiente descendente (representado pelas setas), é possível encontrar o ponto ótimo w_{opt} , que é o menor valor da função. Esta abordagem também pode ser generalizada para uma função de custo com duas ou mais variáveis.

Cada parâmetro de uma rede neural de múltiplas camadas é uma variável “otimizável” por meio da técnica *Backpropagation*, que se relaciona com a função de custo E por meio da regra da cadeia:

$$\frac{\partial E}{\partial w_x} = \frac{\partial E}{\partial w_g} \frac{\partial w_g}{\partial w_x} \quad (2.17)$$

em que w_g é um parâmetro da função de custo E , que depende do parâmetro w_x .

A regra da cadeia permite que a derivada parcial da função de custo em relação a cada um dos parâmetro da rede seja encontrada de uma forma computacionalmente eficiente. A derivada parcial permite encontrar o vetor gradiente, que por sua vez mostrará qual o incremento e sentido que os parâmetros terão que tomar para minimizar a função de custo.

A derivada parcial da função de custo E , por meio da regra da cadeia, pode ser calculada em relação a qualquer parâmetro w_x da rede neural. Assim, todos os parâmetros são ajustados para que o erro calculado pela função de custo seja minimizado.

2.2.3 Rede Neural Convolutacional

Redes Neurais Convolucionais, em inglês *Convolutional Neural Networks* (CNN), constitui um tipo particular de rede neural especializada em processar dados que tem como

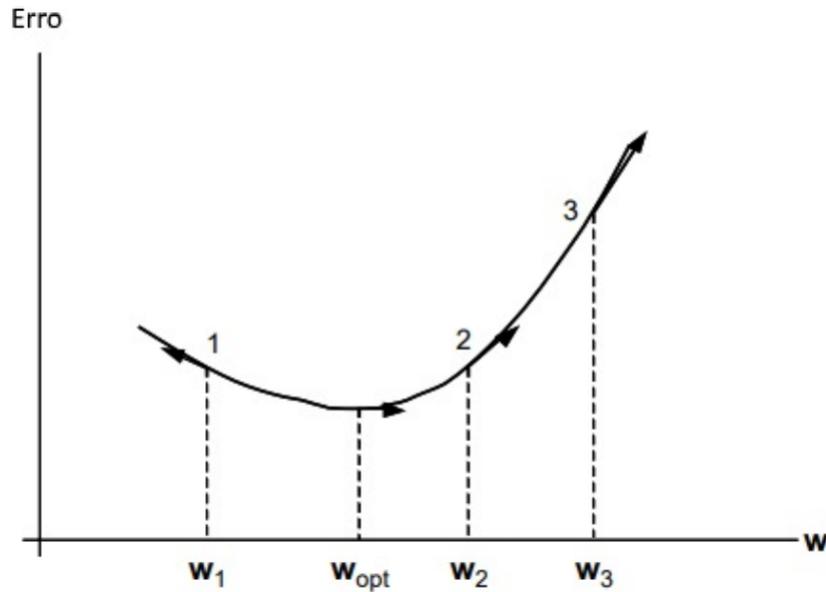


Figura 2.9: Parte da curva de uma função de custo. Os vetores representam os gradientes para quatro diferentes valores de peso w_1 , w_2 e w_3 , e w_{opt} é um valor de mínimo da função (ponto ótimo). Fonte: [7].

característica uma topologia no formato de grades (*grids*), ou seja, dados representados em 1-D, 2-D, \dots , n-D. Segundo Goodfellow et al. [8], a CNN é uma rede que usa convolução no lugar da matriz comum de multiplicação, em pelo menos uma de suas camadas.

Convolução é um operador linear que, a partir de duas funções dadas, resulta em uma terceira que mede a soma do produto dessas funções ao longo da região subentendida pela superposição delas em função do deslocamento existente entre elas. Em termos mais matemáticos:

$$s(t) = \int x(a)w(t-a)da. \quad (2.18)$$

A operação de convolução geralmente é representada com um asterisco:

$$s(t) = (x * w)(t), \quad (2.19)$$

em que o elemento x é chamado de entrada, e w é chamado de núcleo, ou ainda, *kernel*. Em intervalos discretos, esta é representada por meio do seguinte somatório:

$$(x * w)(t) = \sum_{a=-\infty}^{+\infty} x(a)w(t-a). \quad (2.20)$$

Entretanto, no aprendizado de máquina aplicado à imagens, os intervalos são discretos e têm dimensão maior que 1-D. Para uma imagem I de duas dimensões como entrada, e também um *kernel* K , também bi-dimensional, a fórmula fica sendo apresentada da seguinte forma:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i-m, j-n), \quad (2.21)$$

e pela convolução ser comutativa, é equivalente escrever

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i-m, j-n)K(m, n). \quad (2.22)$$

A Figura 2.10 ilustra uma operação de convolução entre uma imagem de dimensão 3×4 com um kernel de dimensão 2×2 .

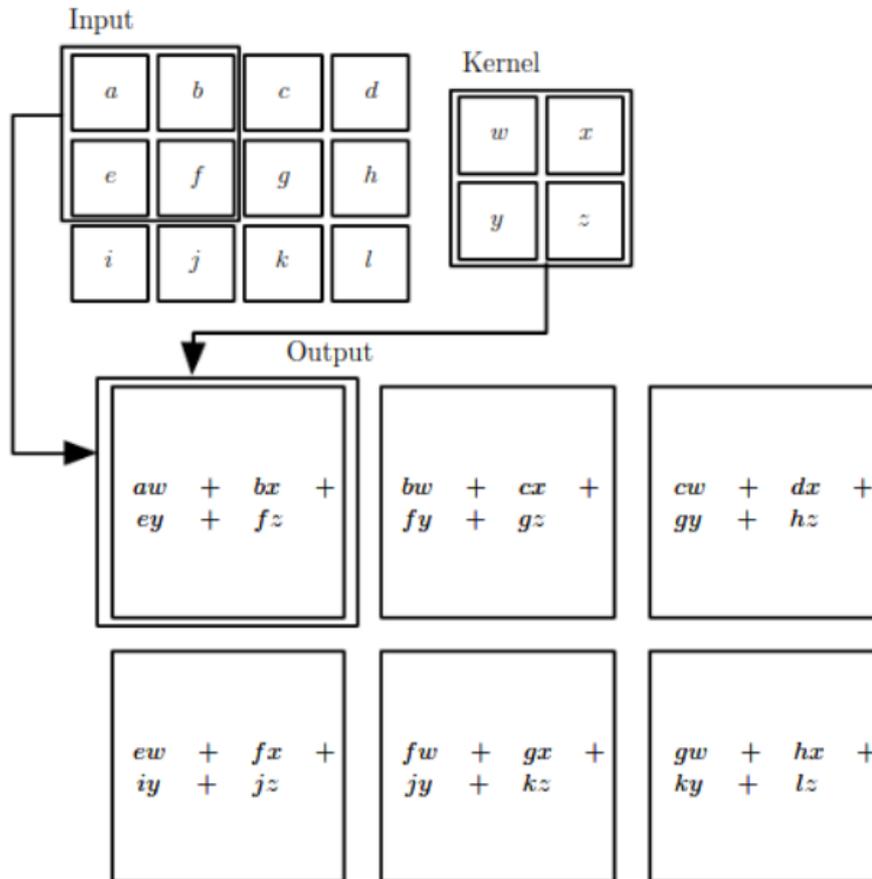


Figura 2.10: Exemplo simples da operação convolução 2-D. Fonte: [8]

2.2.4 Bordas Orientadas Convolucionais

Segundo Maninis et al. [9], a técnica denominada Bordas Orientadas Convolucionais, do inglês *Convolutional Oriented Boundaries* (COB), produz contornos orientados multi-escala e hierarquias de região a partir da classificação genérica da imagem. A arquitetura CNN genérica permite o aprendizado de ponta a ponta dos contornos orientados multi-escala. A estrutura do algoritmo utilizado pelos autores é apresentada na Figura 2.11.

Aproveitando as arquiteturas clássicas da CNN para avaliar tanto os contornos mais grossos quanto os detalhes (mais finos) da imagem, bem como as orientações dos contornos. A técnica COB combina esses canais de saída de forma não-linear em uma única segmentação hierárquica. A Figura 2.12 mostra como é usada as informações fornecidas pelas camadas intermediárias de uma CNN para detectar contornos e suas orientações em múltiplas escalas.

Mais especificamente, a CNN base usualmente utilizada na técnica COB é a *ResNet* de 50 camadas [43], onde as camadas totalmente conectadas são retiradas. Portanto, a rede consiste, principalmente, de camadas convolucionais acopladas a ativações da função do tipo ReLU, divididas em 5 estágios. Cada estágio é tratado como uma escala diferente, uma vez que contém mapas de características de tamanhos semelhantes. No final de um estágio, há uma camada de *pooling* máxima, que reduz as dimensões dos mapas de recursos

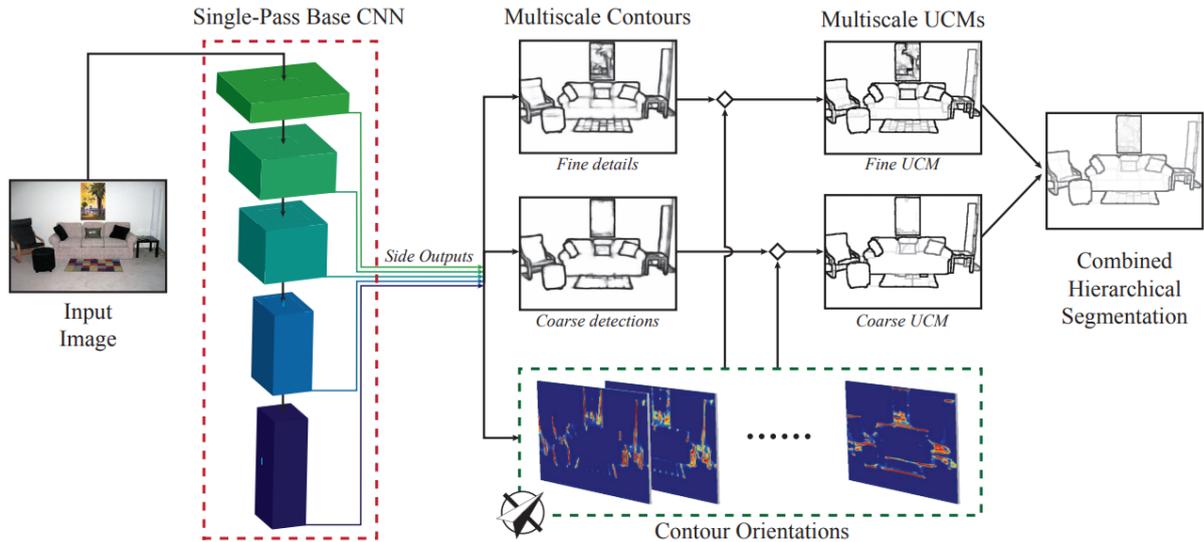


Figura 2.11: Arquitetura do algoritmo COB. Fonte: [9].

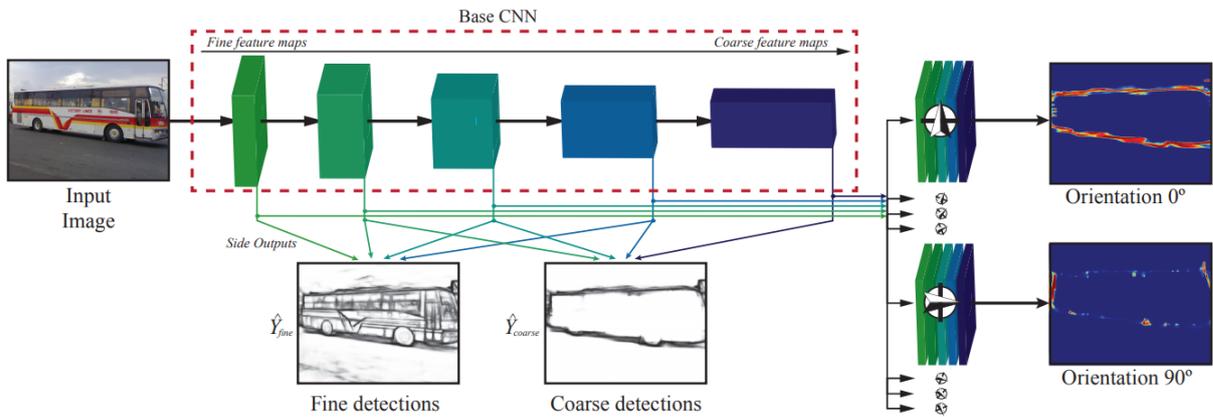


Figura 2.12: Arquitetura de aprendizado profundo de COB. As conexões mostram os diferentes estágios que são usados para gerar os contornos multiescala. Fonte: [9].

produzidos à metade. Cada saída lateral produz um mapa de contorno intermediário em diferentes resoluções. Na CNN base de 5 camadas ilustrada na Figura 2.12, é combinado linearmente as ativações laterais das 4 camadas mais finas e 4 camadas mais grossas para uma saída de escala fina e grossa. A escala fina contém contornos com maior incidência de detalhes, enquanto a escala grossa leva a detecções menos ruidosas.

Para obter os contornos orientados, é utilizada uma extensão da CNN base, descrita anteriormente. Foi conectado K sub-redes diferentes à rede base, cada uma associada a um compartimento de orientação, tendo acesso a mapas de características que são gerados a partir das camadas convolucionais intermediárias em M escalas diferentes. Cada sub-rede tem a tarefa de classificar os pixels dos contornos que correspondem a uma orientação específica. Cada sub-rede consiste em M camadas convolucionais, cada uma delas anexada em diferentes camadas da rede base. Assim, é necessárias $M * K$ camadas adicionais.

2.2.5 Algoritmo *Watershed* via Aprendizagem Profunda

Proposto por Steffen et al. [6], *Learned Watershed (LWS)* é um método de aprendizado baseado no algoritmo clássico *Watershed*, tendo como objetivo resolver com sucesso

segmentações em imagens complexas, como partes de regiões estreitas e limites de baixo contraste, cujos algoritmos anteriores falham.

Os autores propuseram treinar a função da altitude junto com as decisões de atribuição das regiões subsequentes fazendo com que o erro de segmentação final seja diretamente minimizado. A Figura 2.13 mostra um exemplo onde o Watershed padrão segmenta erroneamente, por conta de uma lacuna de fronteira, e o LWS utiliza formas anteriores aprendidas para fechar as lacunas de fronteiras e retornar uma segmentação mais precisa.

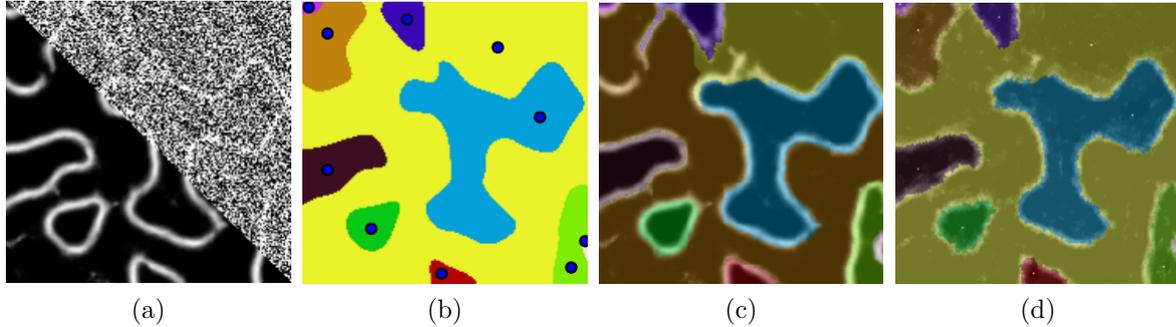


Figura 2.13: (a) Imagem de entrada, (b) ground truth, (c) segmentação do Watershed padrão e (d) segmentação do *Learned Watershed*. Fonte: [6].

O algoritmo *Learned Watershed* mantém a estrutura básica do algoritmo *Watershed* intacta. Cada iteração atribui o melhor candidato atualmente para sua região, e atualiza a fila. A distância topográfica é induzida por uma função de altitude estimada com uma CNN. As altitudes são calculadas sob demanda, permitindo seu condicionamento em decisões anteriores, ou seja, segmentações parciais. A CNN, portanto, tem a oportunidade de aprender, a priori, os formatos de região prováveis nos dados presentes a partir de segmentações de *ground truth*, usando aprendizagem estruturada.

A Figura 2.14 mostra uma visão geral do algoritmo LWS. Em cada iteração, a aresta mínima é encontrada usando uma fila de prioridade (a), e o rótulo da região é propagado (b), o que atualiza a projeção \mathcal{P} . Para todas as arestas não atribuídas que não estão na fila de prioridade e precisa ser avaliadas pelo algoritmo de *Prim* na próxima iteração, a altitude $f_{dyn}(e)$ é avaliada usando a rede de predição de borda dinâmica (c).

Já a arquitetura da rede neural utilizada no LWS baseia-se, principalmente, no trabalho de Yu e Koltun [44], que introduziram convoluções dilatadas para alcançar segmentações densas e agregar sistematicamente informações contextuais em múltiplas escalas sem agrupar operações.

A rede é dividida em dois ramos convolucionais (vide Figura 2.15): O ramo superior f_{static} compreende modelos que, uma vez treinados, dependem apenas da imagem de entrada I , e o inferior f_{dyn} incorpora adicionalmente informações que mudam dinamicamente sobre o estado atual do algoritmo de *Prim*. Uma vez que a entrada do ramo superior não muda durante a previsão, suas ativações de rede podem ser pré-computadas para todas as bordas.

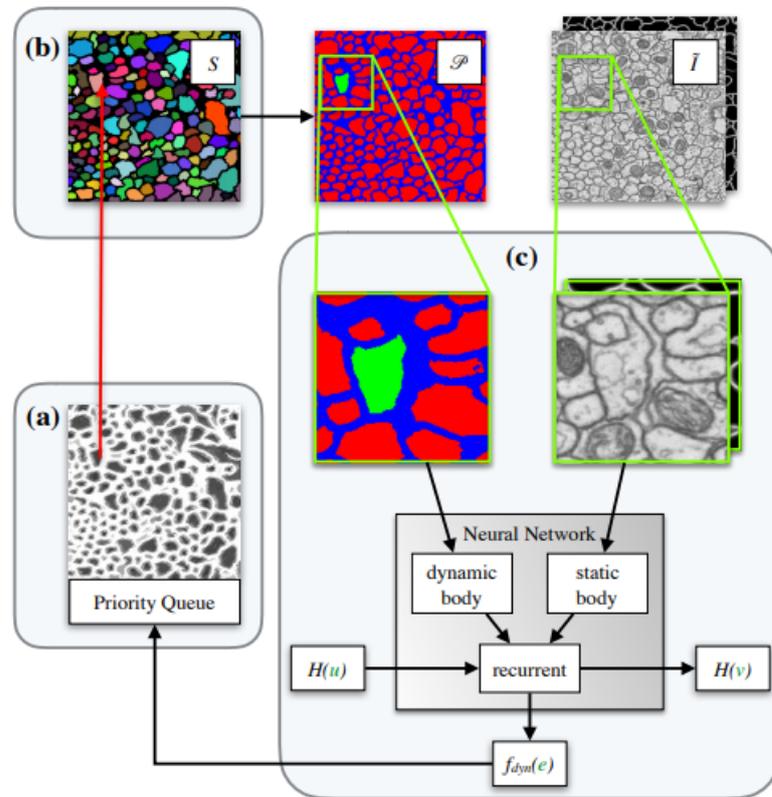


Figura 2.14: Arquitetura do algoritmo Learned Watershed. Fonte: [6].

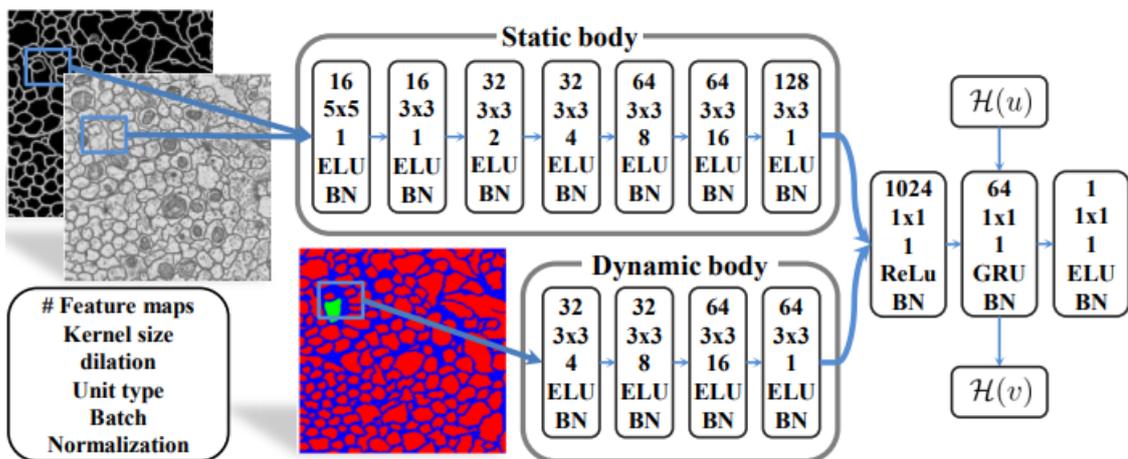


Figura 2.15: Arquitetura da rede neural do algoritmo *Learned Watershed*. Fonte: [6].

3 METODOLOGIA PROPOSTA

Neste capítulo, apresentamos um método de aprendizado profundo para segmentação interativa com foco em imagens médicas, de tomografia computadorizada do pulmão. A abordagem faz uso de uma Rede Neural Convolutiva, que aprende os pesos das bordas da imagem e os integra ao algoritmo *Random Walker*, visando maximizar a precisão da segmentação subsequente no processo de treinamento.

Inspirado em Dong et al. [45] e Pratondo et al. [46], foi utilizado o modelo de Mistura Gaussiana para uma pré-segmentação das imagens de entrada, visto que o GMM filtra ruídos e tecidos que não sejam os do pulmão, obtendo assim um aprendizado mais robusto dos pesos de bordas por parte da CNN.

Também foi descrito as etapas de aprendizado dos pesos de bordas, juntamente com as técnicas de redução dos cálculos da etapa de *Backpropagation* no pipeline de aprendizado. E por fim, foi proposta uma técnica de automatização das sementes.

Para o desenvolvimento do método, utilizou-se como protótipo a CNN desenvolvida por Maninis et al. [9], COB, embora o método proposto seja independente da arquitetura da CNN adotada.

3.1 Descrição do Esquema de Aprendizagem

Vimos na Seção 2.1.1 que a segmentação oriunda do método RW é dada através da solução da equação (2.7), isto é:

$$L_U x_U = -R^T x_S.$$

Para encontrarmos os pesos mais apropriados para as arestas, empregamos o método do Gradiente Descendente de modo a aprender estimadores de pesos de maneira estruturada, isto é, no sentido do que foi proposto em [47].

Ao resolver a equação (2.7) para uma imagem de entrada, iremos então treinar a CNN, que equivale ao problema de minimizar uma função de custo l em relação ao parâmetro Θ da rede neural:

$$\arg \min_{\Theta} l(x_U^*, x_U) = \arg \min_{\Theta} l(x_U^*, -L_U(I; \Theta)^{-1} R(I; \Theta)^T x_S), \quad (3.1)$$

com base na solução ótima x_U^* . Para solucionar este problema de otimização, aplicamos a regra da cadeia a fim de facilitar o uso do método do Gradiente Descendente, isto é:

$$\frac{\partial l(x_U^*, x_U)}{\partial \Theta} = \frac{\partial l(x_U^*, x_U)}{\partial x_U} \frac{\partial x_U}{\partial w} \frac{\partial w}{\partial \Theta}, \quad (3.2)$$

em que $w = (w_1, w_2, \dots, w_{|W_E|})$, com $w_i \in W_E, \forall i \in \{1, 2, \dots, |W_E|\}$.

Observe que a derivada parcial da função de perda em relação ao candidato à solução e a derivada parcial do peso em relação ao parâmetro da rede são simples de serem calculados, porém, o tensor remanescente não, uma vez que sua dimensão é extremamente grande para imagens digitais usuais: 2 rótulos de sementes em milhões de vértices $|U|$ e milhões de arestas $|E|$, ou seja, $\partial x_U/\partial w \in \mathbb{R}^{|U| \times 2 \times |E|}$.

Por outro lado, derivando a equação (2.7) em relação a w tem-se que

$$\frac{\partial L_U x_U}{\partial w} = -\frac{\partial R^T x_S}{\partial w}. \quad (3.3)$$

e aplicando a regra do produto

$$\frac{\partial L_U}{\partial w} x_U + L_U \frac{\partial x_U}{\partial w} = -\left(\frac{\partial R^T}{\partial w} x_S + R^T \frac{\partial x_S}{\partial w} \right). \quad (3.4)$$

Entretanto, note que x_S é constante, e não depende dos pesos da arestas, uma vez que é dado inicialmente pelo usuário.

Logo $\partial x_S/\partial w = 0$, e com isso, chegamos no seguinte tensor de sistema linear, cuja a solução é $\partial x_U/\partial w$:

$$L_U \frac{\partial x_U}{\partial w} = -\left(\frac{\partial L_U}{\partial w} x_U + \frac{\partial R^T}{\partial w} x_S \right). \quad (3.5)$$

Assim, tem-se $2|E|$ equações lineares com a matriz L_U tendo solução $\partial x_U/\partial w$.

O lado direito da equação (3.5) é fácil de calcular, pois $\partial L_U/\partial w$ e $\partial R^T/\partial w$ são esparsos e constantes em relação a w , porém, ainda é caro computacionalmente, já que para calcular esse grande tensor de 3 dimensões é requerido, em cada etapa da descida do gradiente, resolver o sistema linear (3.5).

Tendo em mente que $\partial x_U/\partial w$ é utilizado apenas na multiplicação

$$\frac{\partial l(x_U^*, x_U)}{\partial w} = \frac{\partial l(x_U^*, x_U)}{\partial x_U} \frac{\partial x_U}{\partial w}, \quad (3.6)$$

podemos fazer duas simplificações conforme descritas a seguir.

Gradiente Esparso

Ao invés de solucionar todo o gradiente de três dimensões $\partial l(x_U^*, x_U)/\partial w$, tomamos uma quantidade $k \ll |E|$ de arestas aleatórias, resolvemos os sistemas lineares correspondentes, e então definimos todos os restantes como zero.

Tal abordagem pode ser vista como um algoritmo de descida do gradiente estocástico. Para otimizar o aprendizado do método, tomaremos as k arestas de forma aleatória respeitando duas regras:

1. Não estejam na 4-vizinhança das sementes, ou seja, para todo peso w_i tomado como anteriormente, tem-se $i \notin N(j)$, $\forall j \in S$, pois tais pontos terão sempre alta probabilidade de pertencerem a região da semente que faz vizinhança, sendo o ideal é que as sementes não estejam na borda da sua região;
2. Estejam no interior das regiões delimitadas pelas sementes de fundo, ou seja, mais próximos a região dos pulmões, desta forma focando no aprendizado dos pesos de maior interesse.

Poda do Gradiente

Komodakis et al. [48], propuseram uma *Poda dos Rótulos*, onde ao visitar os nós de um Campo Aleatório de Markov, reduzem dinamicamente o número de rótulos possíveis para cada nó, descartando rótulos que são improvável de ser atribuído a esse nó.

Ao realizar a multiplicação na equação (3.6), o tensor $\partial x_U / \partial w$ é modulado com as dimensões de $\partial l(x_U^*, x_U) / \partial x_U$. Assim, inspirado em Komodakis et al., a cada etapa de *Backpropagation*, será computado apenas o rótulo de maior contribuição no gradiente, ou seja, para cada aresta $(i, j) \in E$, é computado as contribuições dos rótulos que satisfazem a equação a seguir:

$$\arg \max_{a \in \{x_F, x_B\}} \left| \left(\frac{\partial l(x_U^*, x_U)}{\partial x_U} \right)_{i,a} \right|. \quad (3.7)$$

Essas duas abordagens combinadas reduzem a dimensão do gradiente $\partial l(x_U^*, x_U) / \partial w$, de $|U| \times 2 \times |E|$, para $|U| \times 1 \times k$ e, assim, ao invés de resolvermos $2|E|$ sistemas lineares esparsos de tamanho $|U|$, solucionamos k sistemas lineares esparsos de tamanho $|U|$.

Em nossa metodologia, foi escolhida a função de Perda de Entropia Cruzada (CE) – equação (2.16)) – definida por:

$$\text{CE}(x^*, x) = -\frac{1}{|V|} \sum_{i \in V} \sum_{j \in \mathcal{L}} x_{i,j}^* \log(x_{i,j}),$$

em que x é a nossa matriz de atribuição calculada pelo método, e x^* é a solução ótima que procuramos.

Visando selecionar as sementes no processo de treinamento do método, a região do pulmão será tomada como objeto, e o restante como fundo. Um usuário deve então fornecer sementes pra cada região do tipo “objeto” e contorná-la, de “forma esparsa”, com sementes rotuladas no fundo. As sementes utilizadas na fase de teste do modelo foram tomadas aleatoriamente seguindo as regras acima, e os cuidados para estarem distantes o suficiente das bordas dos objetos.

A Figura 3.1 mostra um panorama geral do pipeline de aprendizado proposto. Em nossa implementação, usamos um grafo 4-conectado, portanto, precisamos de apenas dois pesos de bordas por pixel, um horizontal e um vertical. Para a CNN, foi utilizado a rede COB com $K = 8$ e $M = 5$ (vide Seção 2.2.4).

3.2 Automatização das Sementes

Na etapa de validação, adotamos uma estratégia de sementeção semi-automática para que o método proposto possa ser utilizado com sucesso na segmentação de imagens do tipo CT carregando grande quantidade de fatias. Assim, inspirado no trabalho de Dong et al. [45], as sementes são dadas de forma quase automática. Para esta abordagem, foi considerado o fato de que as fatias adjacentes tem alta correlação entre si. Assim, definida a segmentação de uma fatia, ela pode servir por base para estimar as sementes de fundo e objeto para a fatia subsequente.

Inicialmente, dado o conjunto CT de entrada, foi tomada como fatia inicial aquela em que o objeto tenha a maior região em relação às outras fatias. Desta forma, uma vez que as sementes são marcadas manualmente para a fatia mencionada, a mesma é então segmentada com o método aprendido. Com a segmentação da fatia inicial i , as regiões

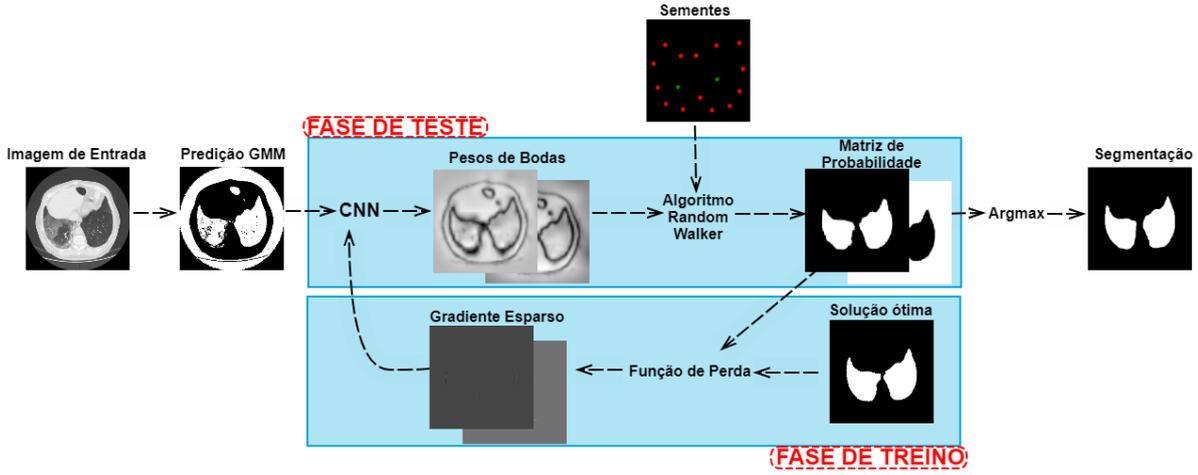


Figura 3.1: Visão geral do pipeline do algoritmo proposto. **Acima:** A imagem de entrada passa por uma pré-segmentação, que é então mapeada por uma CNN para um grafo de borda ponderada não direcionado. Uma matriz de probabilidade é calculada para cada região usando o segmentador *Random Walker*. Na etapa de teste, uma segmentação é calculada a partir das probabilidades. **Abaixo:** A partir dos mapas de probabilidade, uma amostra esparsa do gradiente é calculada e, então, fornecida a rede CNN. Fonte: Autoria própria.

para as sementes candidatas de fundo e objeto da fatia $i + 1$ são dadas, respectivamente, pelas seguintes equações:

$$\begin{aligned} FS_{i+1} &= (SEG_i \oplus FD_1) - (SEG_i \oplus FD_2) \\ OS_{i+1} &= (SEG_i \ominus OE), \end{aligned} \quad (3.8)$$

em que SEG_i é a região segmentada como objeto pelo método treinado referente à fatia i , FD_1 e FD_2 são os elementos estruturantes da dilatação, e OE o elemento estruturante da erosão [49].

Os elementos estruturantes foram selecionados empiricamente como sendo discos com raio de $FD_1 = 14$ pixels, $FD_2 = 12$ pixels e $OE = 10$ pixels.

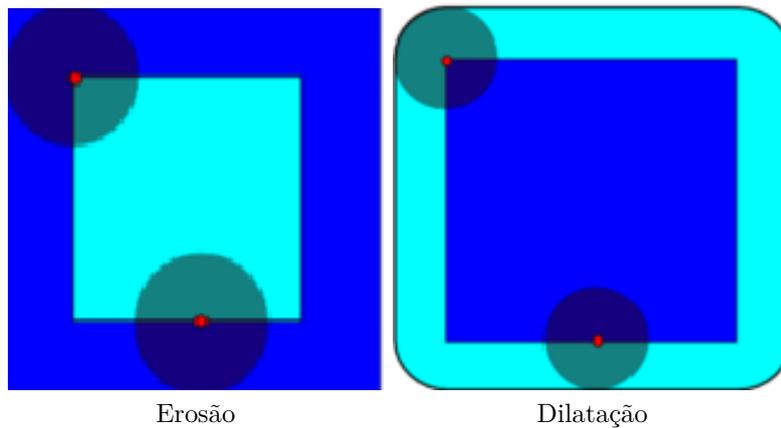


Figura 3.2: Erosão da região azul escuro para a região azul claro, e dilatação da região azul escuro para a região azul claro. Fonte: Autoria própria.

A região FS_{i+1} é tomada como as sementes de fundo. Entretanto, o mesmo não pode ser feito com a região OS_{i+1} para as sementes do tipo objeto, uma vez o método proposto

não prediz com total precisão a região do pulmão e, mesmo que pequena, existe uma diferença entre as fatias, fazendo com que a região OS_{i+1} contenha pixels que não fazem parte do tecido do pulmão e muito próximos a borda, como mostra a Figura 3.3. Porém, pode-se restringir como sementes de fundo os pixels de OS_{i+1} que tenham intensidade no intervalo $[a, b]$, onde:

$$\begin{aligned} a &= \mu - \beta\sigma \\ b &= \mu + \beta\sigma, \end{aligned} \quad (3.9)$$

com μ e σ sendo a média e a variância, respectivamente, do método GMM, e β uma variável de ajuste dada empiricamente no intervalo $[0.05, 0.3]$, correspondendo baixo contraste e alto contraste do conjunto de dados.

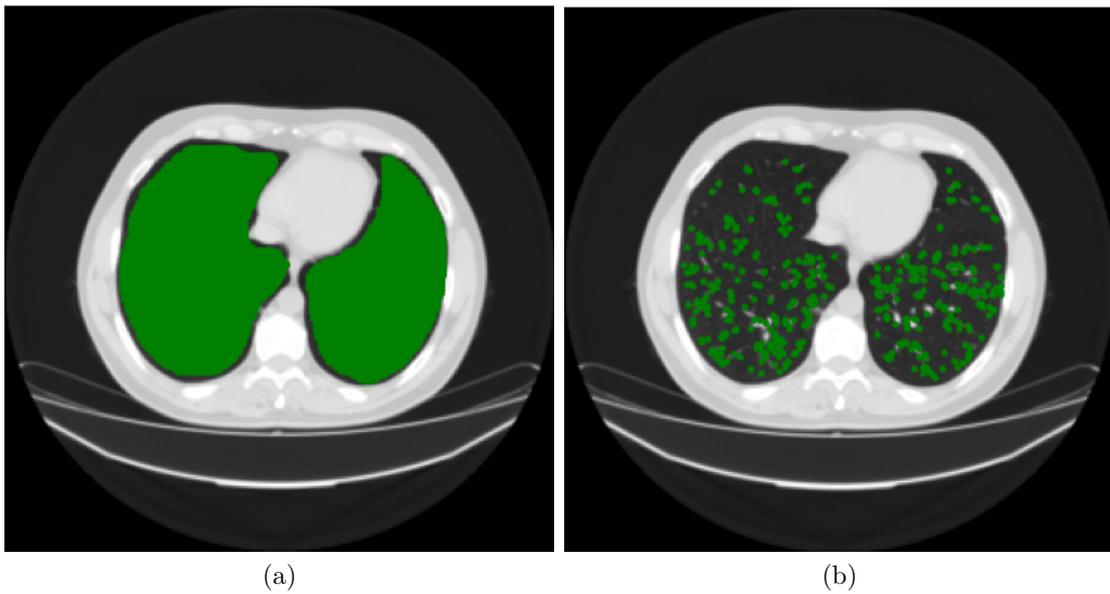


Figura 3.3: (a) Sementes de objetos antes da restrição e (b) sementes de objeto após a restrição. Fonte: Autoria própria.

De posse das sementes de objeto e fundo, aplica-se então o algoritmo treinado para calcular a segmentação da fatia $i + 1$, e assim sucessivamente, até a última fatia do volume CT de entrada.

Para as fatias anteriores, a fatia inicial i segue o processo análogo, porém, com índice decrescente. A Figura 3.4 ilustra o modelo de automatização das sementes.

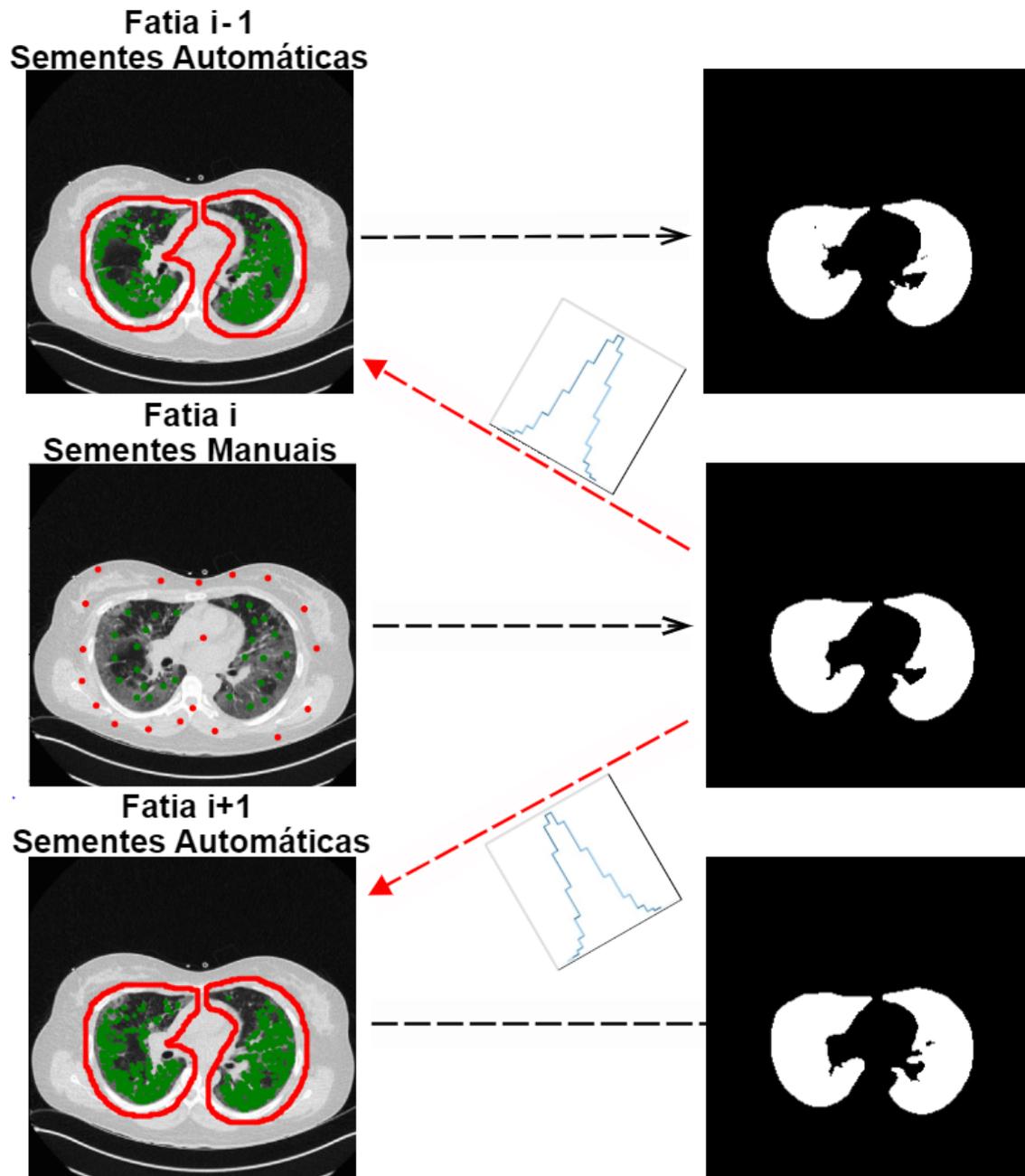


Figura 3.4: Visão geral do processo de automatização das sementes. Fonte: Autoria própria.

4 RESULTADOS E DISCUSSÕES

Este capítulo é destinado à apresentação dos resultados experimentais obtidos e comparações com outros métodos de segmentação da literatura. Assim, para melhor entendimento dos resultados obtidos nessa pesquisa de mestrado, abrimos nossa discussão a partir da descrição da base de dados utilizada nos testes, bem como as métricas de qualidade aplicadas para avaliar o desempenho das segmentações.

Seguindo o mesmos ensaios experimentais de outros trabalhos da literatura [2, 3], os resultados são divididos em dois tipos de análises distintas:

- **Resultados Quantitativos:** são gerados em formatos numéricos, através das métricas, e serão discutidos a partir da adoção de gráficos e tabelas.
- **Resultados Qualitativos:** competem aos resultados visuais, ou seja, as imagens segmentadas serão exibidas e comparadas em termos visuais, tomando-se como parâmetro na análise as imagens de *ground truth*.

4.1 Base de Dados e Métricas de Avaliação

4.1.1 Base de Dados

O conjunto de dados utilizado neste trabalho foi disponibilizado pelo grupo de pesquisa MedSeg [34]. Ele consiste de imagens volumétricas, de tomografia computadorizada do pulmão de diferentes pacientes infectados pelo novo coronavírus SARS-CoV-2, juntamente com suas respectivas representações do tipo *ground truth*. Mais precisamente, a base fornece oito conjuntos de imagens do tipo CT, sendo que as fatias de cada volume possuem dimensões de 610×610 pixels.

Visando acelerar as comparações experimentais com outras técnicas da literatura, em especial, o LWS, que acabou levando tempo demasiado para treinar com as imagens de dimensões originais da base, as fatias dos volumes CT foram redimensionados para 256×256 pixels, através da relação de área interna [50].

Um dos volumes foi utilizado para o treinamento do método, enquanto todos os demais foram empregados para a etapa de testes. A Figura 4.1 apresenta algumas das imagens do conjunto de dados utilizado.

4.1.2 Métricas de Avaliação

A fim de quantificar a qualidade final das segmentações, foram utilizadas seis métricas de avaliação:

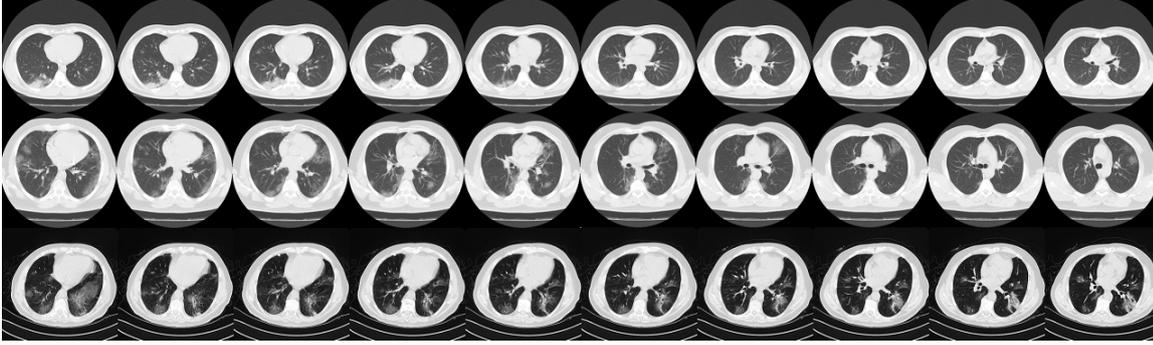


Figura 4.1: Exemplos de imagens de três dos volumes do conjunto de dados Medseg. Fonte: Autoria própria.

- *Variation of Information (VOI).*
- *Rand Index (RI).*
- *Adjusted Rand Index (ARI).*
- *Precision.*
- *Recall.*
- *F-score.*

Tais métricas comparam o resultado da segmentação obtida por um dado modelo com as respectivas segmentações de referência, ou seja, os *ground truth* (GT), retornando valores escalares que representam uma avaliação da qualidade final da segmentação. Dependendo da métrica, seus valores indicam melhor qualidade das segmentações caso sejam próximos a 0 ou, ainda, um valor que seja próximo a 1.

A fim de entendermos melhor cada métrica, será apresentada uma breve descrição de cada métrica quando aplicada ao par segmentação obtida e e representação GT.

Variation of Information

A tradução de *Variation of Information* é *Variação da Informação*, sendo exatamente essa “quantidade” que a métrica VOI afere. Ela procura medir a distância com relação à entropia do resultado da segmentação e do GT, denotada aqui por S e GT respectivamente, ou seja, avalia a quantidade de informação desnecessária entre as duas imagens segmentadas. Logo, o que esperamos obter é o menor valor possível para essa métrica em particular.

Também é possível demonstrar que a VOI é uma métrica fundamentada em conceitos clássicos da Álgebra Linear [51, 52].

O valor escalar da VOI é dado pela equação a seguir:

$$VOI = H(S) + H(GT) - 2M(S, GT), \quad (4.1)$$

em que $H(S)$ e $H(GT)$ são os valores de entropia relacionados à segmentação e ao *ground truth*, respectivamente, e $M(S, GT)$ representa as informações que S e GT compartilham, chamada de *Mutual Information* [53].

Rand Index

A métrica RI mede a proximidade entre a segmentação obtida e o seu respectivo GT. Tal comparação é feita levando-se em consideração o número de pares de pixels associados em S e GT , que possuem o mesmo rótulo. A métrica (RI) era utilizada, a princípio, apenas para avaliar resultados de técnicas de *clustering*, sendo modificada por Unnikrishnan et al. [54], e Yang et al. [55] para aplicações que viessem a contemplar outras aplicações na área de segmentação de dados e imagens.

Se denotarmos um par de pixels por $S_i \in S$ e $G_i \in GT$, ambos de mesmo índice com relação à imagem base, o valor do Rand Index [56] pode ser obtido pela seguinte equação:

$$RI = \frac{\text{Pares com mesmo rótulo}}{\text{Total de pares}}. \quad (4.2)$$

É fácil ver que $0 \leq RI \leq 1$, e também que $100 * RI$ é a porcentagem do total de pixels da imagem que foram segmentados conforme esperado. Logo, o que esperamos obter é o maior valor possível.

Adjusted Rand Index

A métrica ARI é a versão “ajustada por acaso” da métrica *Rand Index*. A métrica RI calcula uma medida de similaridade entre dois *clusters*, considerando todos os pares de amostras e pares de contagem que são atribuídos nas mesmas ou diferentes partições, isto é, nos *clusters* previstos em S e verdadeiros em GT .

A pontuação de RI clássica é então “ajustada por acaso” na pontuação ARI usando o seguinte esquema:

$$ARI = \frac{RI - RI_{esperado}}{\max(RI) - RI_{esperado}}. \quad (4.3)$$

Uma vez que RI está entre 0 e 1, o valor $RI_{esperado}$ (embora não seja um valor constante) deve ser maior ou igual a 0. Por outro lado, $ARI \leq 1$. Portanto, há uma faixa mais ampla de valores que ARI pode assumir, aumentando assim a sensibilidade do índice.

Assim, a métrica ARI é assegurada a ter um valor próximo de 0 para uma rotulagem aleatória independentemente do número de *clusters* e de amostras, e é exatamente 1 quando os clusters são idênticos, ou seja, quanto maior o valor de ARI , melhor.

Precision e Recall

Na segmentação de imagens, *Precision* é a fração das segmentações relevantes entre as segmentações recuperadas, enquanto o *Recall* é a fração das segmentações relevantes que foram recuperadas. Isso significa que ambas são, portanto, baseadas na relevância. *Precision* e *Recall* são dadas, respectivamente, pelas seguintes fórmulas matemáticas:

$$PRECISION = \frac{vp}{vp + fp} \quad (4.4)$$

$$RECALL = \frac{vp}{vp + fn}, \quad (4.5)$$

em que vp (verdadeiro positivo) é a quantidade de pixels de objeto segmentados corretamente, fp (falso positivo) a quantidade de pixel de fundo segmentado como de objeto e fn (falso negativo) a quantidade de pixels de objeto segmentado como de fundo.

Uma pontuação de *Precision* de 1.0 significa que cada pixel rotulado como objeto realmente é um pixel de objeto, mas não diz nada sobre o número de pixels de objeto que não foram rotulados corretamente, enquanto um *Recall* de 1.0 significa que cada pixel de objeto foi rotulado corretamente, mas não diz nada sobre quantos pixels de fundo foram incorretamente também rotulados como sendo pixel de objeto. Logo, é possível calibrar a segmentação e melhorar a *Precision* em detrimento da métrica *Recall*, ou vice-versa. Portanto, o que queremos obter é o maior valor possível em ambas as medidas.

F-score

Normalmente, as pontuações de *Precision* e *Recall* são ponderadas a partir de uma métrica-sumário. Assim, a métrica de qualidade *F-score*, também conhecida por *F1-score*, é um índice que combina as contribuições das métricas *Precision* e o *Recall*. Este é definido como a média harmônica das métricas *Precision* e *Recall*, isto é:

$$F\text{-SCORE} = 2 * \frac{PRECISION * RECALL}{PRECISION + RECALL}. \quad (4.6)$$

O maior valor possível de *F-score* é 1, indicando *PRECISION* e *RECALL* perfeitos, e o menor valor possível é 0, se *PRECISION* ou *RECALL* for zero. Portanto, quanto maior o valor de *F-score*, maior será a qualidade da segmentação.

4.1.3 Resultados e Comparações

Ajuste de Parâmetros na Criação dos Mapas de Sementes

Uma das contribuições do algoritmo proposto é a distribuição das sementes de forma semi-automática que, ao segmentar um volume completo CT, é requerido apenas o mapeamento de um único *slice*, obtendo assim os demais mapas de sementes para todas as outras fatias do volume de maneira inteiramente automática.

Conforme explorado na Seção 3.2, tal estratégia de distribuição de sementes utiliza-se das operações morfológicas de *Erosão* e *Dilatação* que, por sua vez, dependem, respectivamente, dos raios dos elementos estruturantes das dilatações (r_{D1} e r_{D2}), e do raio do elemento estruturante da erosão (r_E). Outro importante parâmetro a ser analisado é a variável de ajuste β da equação (3.9), conforme ilustrado na Figura 3.3. Assim, inspirado em [45], β foi adotado dentro do intervalo [0.05, 0.3], cujas extremidades refletem o baixo/alto contraste da imagem, embora $\beta = 0.3$ possa, em alguns casos, induzir à criação de falsas sementes do tipo objeto, conforme ilustrado Figura 4.2. Por outro lado, valores próximos de 0.05 podem diminuir a quantidade de sementes, podendo fazer com que a segmentação não avance para a região do pulmão, tal como ilustrado na Figura 4.3.

Já os parâmetros r_{D1} , r_{D2} foram escolhidos de forma que as sementes de fundo não ficassem muito distantes da região do pulmão, servindo também como uma heurística de delimitação para o objeto a ser segmentado. O parâmetro r_E foi escolhido de modo que a região das sementes de fundo não fossem limitadas, principalmente em regiões pequenas. A Figura 4.4 ilustra um caso onde os pixels de fundo foram segmentados como objeto, enquanto que na Figura 4.5 foi limitado a região para sementes de objeto, e desta forma, apenas o lado direito recebe sementes para ser segmentada.

Portanto, a partir de uma grande bateria de testes realizados e a fim de mitigar os problemas apresentados nos parágrafos anteriores, maximizando assim a qualidade da segmentação, tomamos $r_{D1} = 14$, $r_{D2} = 12$ e $r_E = 8$.

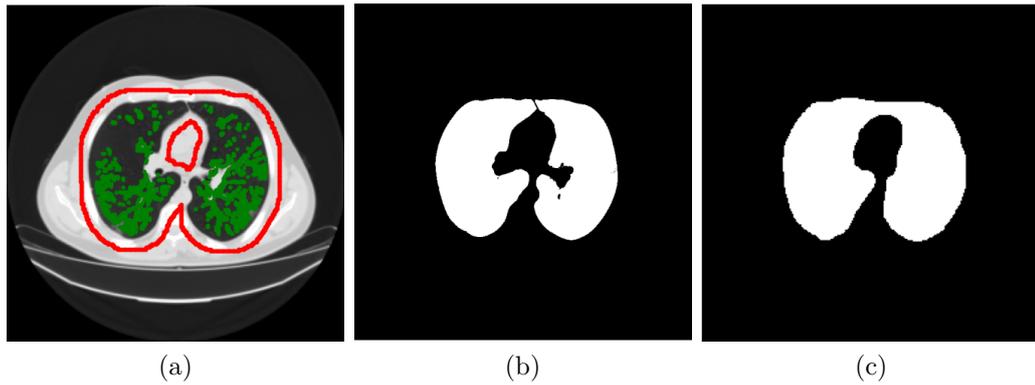


Figura 4.2: Caso de falsas sementes de objeto como consequência da escolha de um alto valor para β . (a) Imagem de entrada com as sementes tomadas com $\beta = 0.3$, (b) *ground truth* e (c) segmentação obtida pelo método proposto. Fonte: Autoria própria.

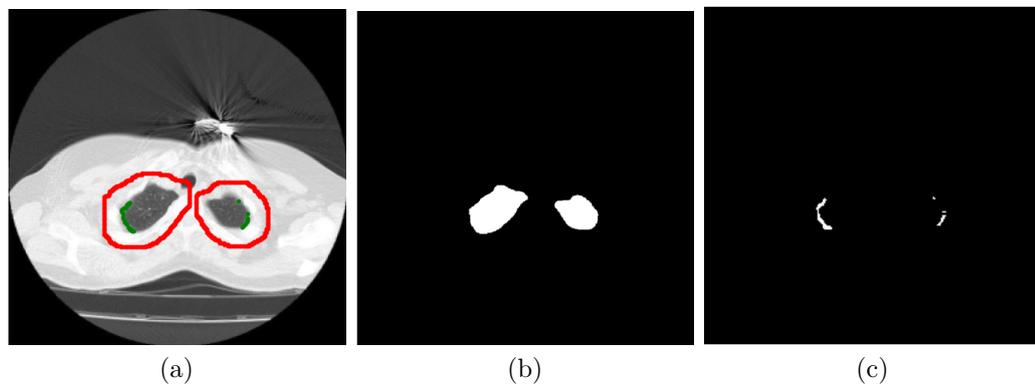


Figura 4.3: Caso em que poucas sementes implicam em uma segmentação de baixa qualidade, devido à adoção de um baixo valor para β . (a) Imagem de entrada com as sementes tomadas com $\beta = 0.05$, (b) *ground truth* e (c) segmentação obtida pelo método proposto. Fonte: Autoria própria.

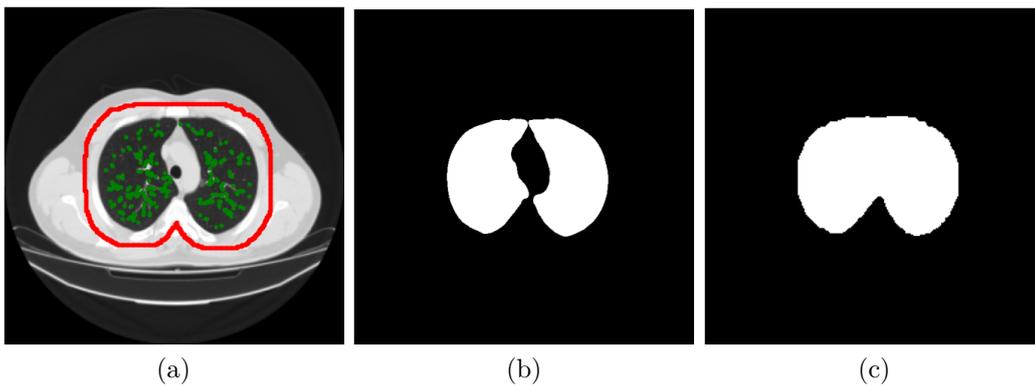


Figura 4.4: (a) Imagem de entrada com as sementes tomadas com os parâmetros $r_{D1} = 20$, $r_{D2} = 18$ e $r_E = 8$, (b) *ground truth* e (c) segmentação pelo método proposto. Fonte: Autoria própria.

A Figura 4.6 ilustra as segmentações obtidas das mesmas imagens complexas exploradas nas Figuras 4.2, 4.3, 4.4 e 4.5, mas com $r_{D1} = 14$, $r_{D2} = 12$, $r_E = 8$ e $\beta = 0.1$. A partir dos novos resultados, foi possível observar uma melhora significativa na qualidade

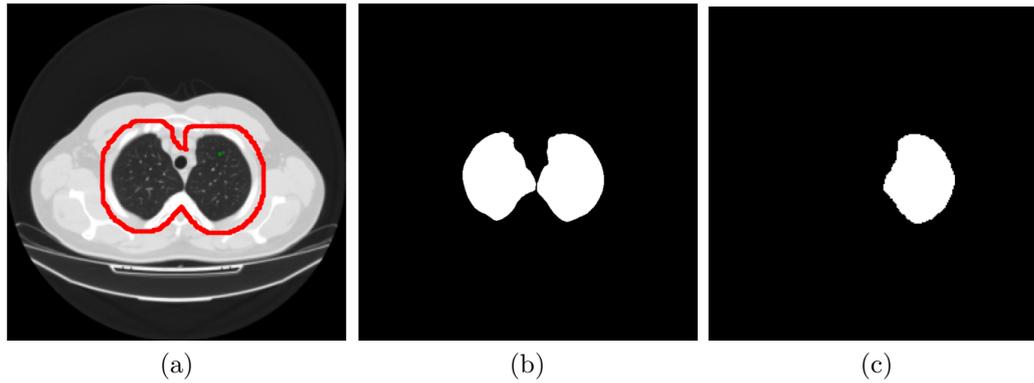


Figura 4.5: (a) Imagem de entrada com as sementes tomadas com os parâmetros $r_{D1} = 14$, $r_{D2} = 12$ e $r_E = 8$, (b) *ground truth* e (c) segmentação pelo método proposto. Fonte: Autoria própria.

das segmentações.

Pre-segmentação e Realce de Bordas via Algoritmo GMM

Ao fazer a pré-segmentação utilizando o algoritmo GMM, passa-se pelo detector de bordas uma imagem onde foi filtrado os ruídos e tecidos que não sejam os do pulmão. Desta forma, a CNN retorna os pesos de bordas de forma mais precisa. Na Figura 4.7, podemos visualizar que, com a pré-segmentação (primeira linha), a CNN retorna pesos de borda vertical e horizontal mais robustos do que sem a pré-segmentação (segunda linha), onde este último apresenta ruídos no entorno externo da região do pulmão.

Aspectos Computacionais

Conforme discutido na Seção 3.1, a fim de otimizar o tempo de treinamento do método, aplicamos as técnicas de poda do gradiente e gradiente esparso, sem que houvesse perda na qualidade do aprendizado dos pesos de bordas. A Figura 4.8 ilustra o crescimento exponencial do tempo de execução da técnica *Backpropagation* em relação ao aumento da quantidade de amostras tomadas no gradiente.

Os resultados foram obtidos em uma imagem de entrada com dimensões 256×256 , utilizando uma máquina de CPU (i7-8565U, 1.80GHz\1.99GHz).

Já na Figura 4.9 é ilustrado o gráfico da convergência da função de custo no decorrer das iterações em relação a diferentes quantidades de amostras tomadas.

Pode-se observar que o aumento das amostras está intimamente ligado a convergência da função de custo, entretanto, a partir de 1024 amostras, a melhora na convergência é mínima, enquanto que o aumento do tempo de execução dos passos da *Backpropagation* é exponencial. Portanto, foi escolhido $n = 1024$ levando em conta o melhor resultado entre aceleração ($\simeq 35$ vezes) e precisão, onde uma etapa de *Backpropagation* com gradiente completo leva em média 12 minutos.

Comparações

Esta seção é destinada aos testes comparativos de qualidade e de eficiência envolvendo a metodologia de segmentação proposta e demais técnicas da literatura. Desta forma, iremos apresentar os resultados quantitativos e qualitativos do método proposto,

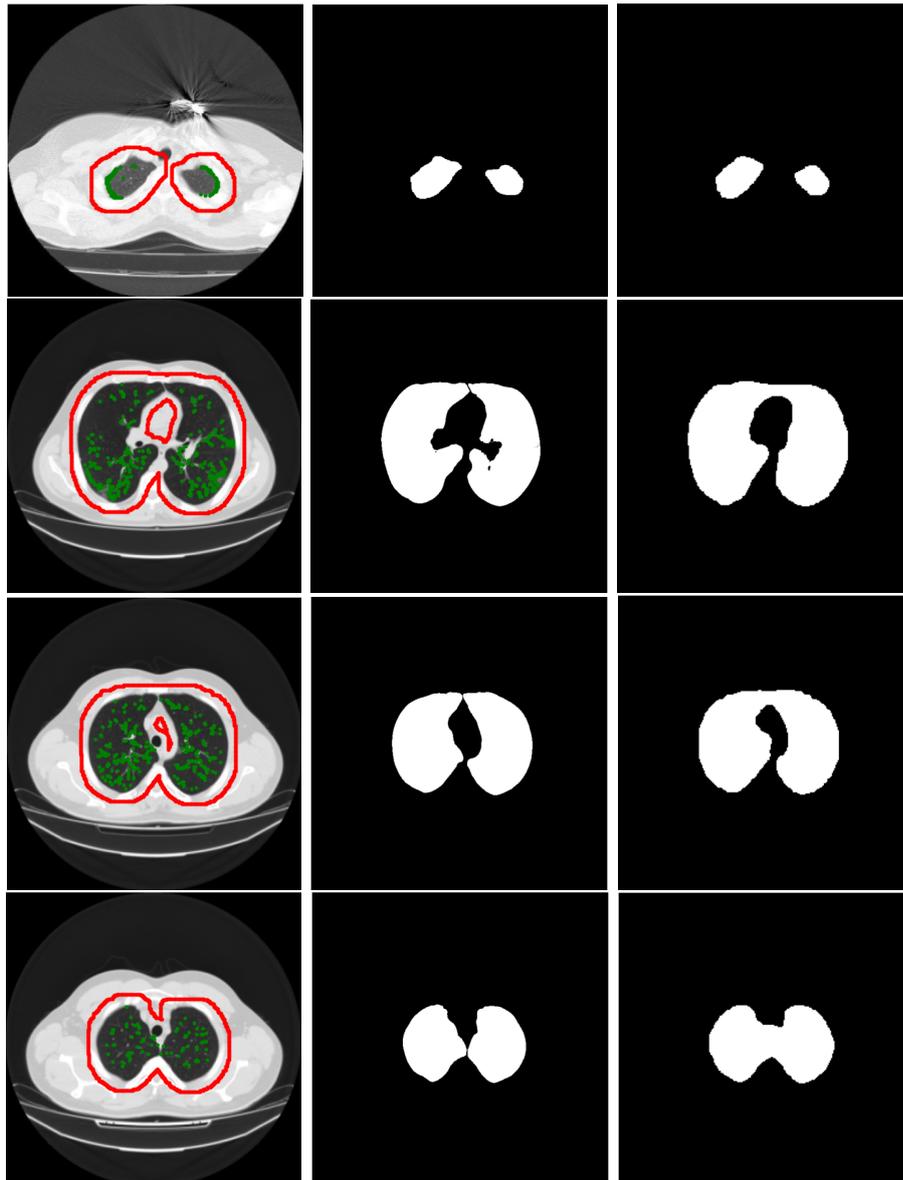


Figura 4.6: (a) Imagens de entrada com as sementes tomadas com $\beta = 0.05$, (b) $r_{D1} = 14$, $r_{D2} = 12$ e $r_E = 8$, (c) ground truth e (d) segmentação pelo método proposto. Fonte: Autoria própria.

e compará-los com os seguintes métodos da literatura: *Random Walker* [16], e *Learned Watershed* [6]. Os métodos RW e LWS tiveram seus parâmetros ajustados de forma a otimizar a segmentação das imagens, conforme sugerido pelos próprios autores em seus respectivos trabalhos.

Com relação à metodologia proposta, o pré-segmentador GMM foi treinado para três *clusters*, em um volume CT de 30 imagens com dimensões 256×256 do conjunto de dados *Medseg* (vide Seção 4.1.1). O detector de bordas COB [9] integrado ao segmentador *Random Walker* foi treinado no mesmo conjunto citado anteriormente, assim como o método LWS, que também foi treinado nesse mesmo conjunto, a partir de um total de 100 épocas. As sementes definidas para o métodos proposto e RW foram fornecidas conforme descrito na Seção 3.2. Já para o LWS, foram dadas como descrito em [6] de forma a otimizar sua segmentação.

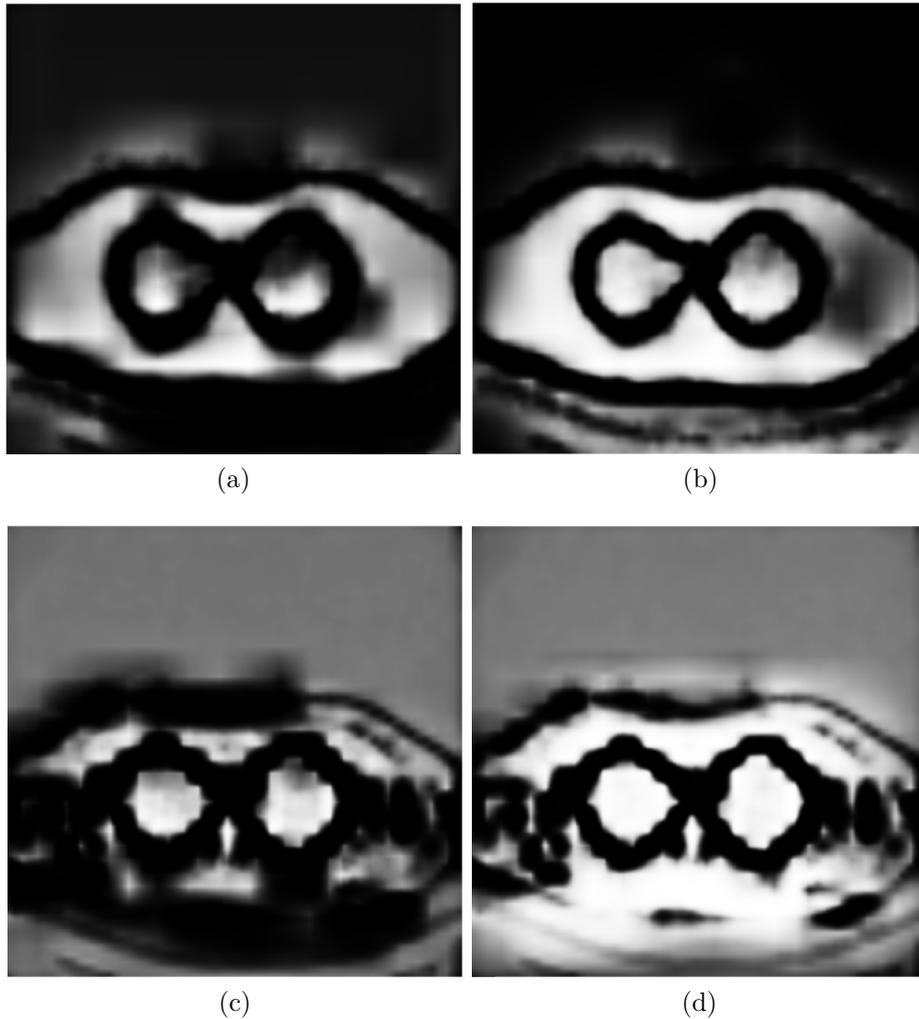


Figura 4.7: (a) Pesos de bordas vertical, (b) pesos de borda horizontal com pré-segmentação, (c) pesos de bordas vertical, e (d) horizontal sem pré-segmentação. Fonte: Autoria própria.

Na Figura 4.10, comparamos as segmentações da metodologia proposta com as segmentações do RW e do LWS em pulmões com lesões aparentes. O intuito desse experimento foi demonstrar que o algoritmo proposto proporciona segmentações de alta qualidade apesar do pulmão apresentar pouca ou muitas lesões.

Na primeira e segunda linha, pode-se observar poucas lesões, sendo representadas por tons de cinza mais próximos ao do tecido do pulmão e localizados no interior do mesmo. Os três algoritmos apresentaram segmentações muito próximas do ideal. Na terceira e quarta linha, os três métodos ainda apresentaram segmentações próximas do desejado, apesar das lesões aparecerem em maior quantidade e com intensidades de cinza mais distantes daquelas do tecido do pulmão, porém, ainda localizadas predominantemente no interior do pulmão. Na quinta, sexta e sétima linhas, as lesões que estão localizadas na parede do pulmão apresentaram intensidades de cinza distantes aos do tecido pulmonar e próximas aos da região externa. Nesses casos, as segmentações dos algoritmos RW e LWS não se propagam até as regiões citadas, falhando na captura correta de tal região. Por fim, note que o aprendizado do método proposto supera tal dificuldade, identificando corretamente as bordas da região do pulmão e segmentando corretamente onde os outros dois métodos concorrentes falharam.

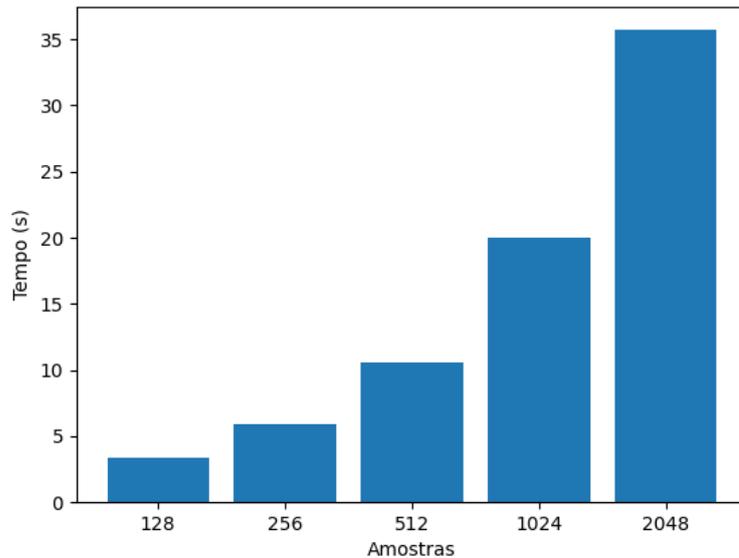


Figura 4.8: Relação entre o tempo de execução da técnica *Backpropagation* e as diferentes quantidades de amostras tomadas no gradiente. Fonte: Autoria própria.

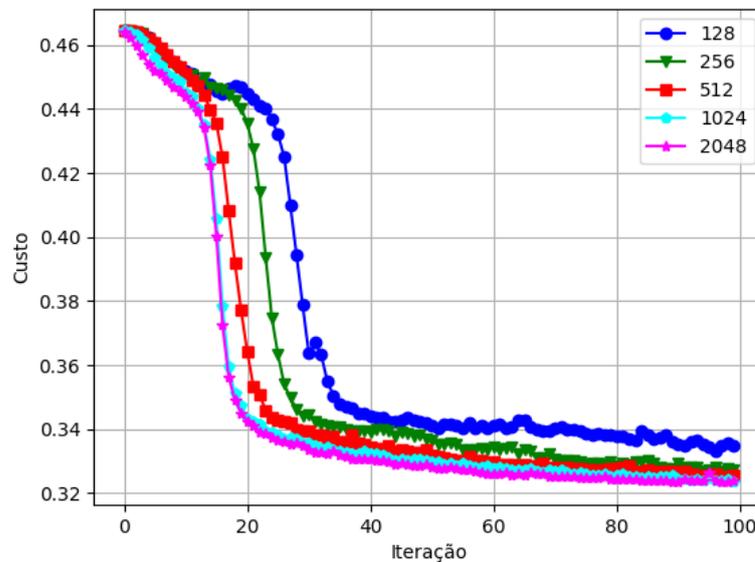


Figura 4.9: Relação entre a convergência da função de custo no decorrer das iterações em relação a diferentes quantidades de amostras tomadas. Fonte: Autoria própria.

A Figura 4.11 ilustra alguns casos em que o LWS “vaza” por conta de um “fraco contorno”. Nos referidos casos, o LWS segmenta toda uma “bacia”, propagando uma falsa segmentação. Entretanto, o detector de bordas do método proposto aprende os pesos de bordas, fechando assim lacunas de fronteira, e ainda mais, as sementes de fundo (vide Seção 3.2) também passaram a funcionar como um “delimitador de região”, impedindo que a falsa segmentação se espalhe demasiadamente.

Por fim, apresentamos os resultados quantitativos com relação ao método proposto, comparando-o com os algoritmos RW e LWS para, assim, endereçar os aspectos numéricos quanto à precisão e acurácia na segmentação de volumes CT de pulmão com lesões. Iremos utilizar as métricas descritas na Seção 4.1.2: *Precision*, *Recall*, *F-score*, *Variation of Information* (VOI), *Rand Index* (RI) e *Adjusted Rand Index* (ARI). Para uma melhor compreensão das métricas, aquelas que apresentam o melhor resultado no sentido de

“quanto maior, melhor” estão acompanhadas do símbolo (\uparrow) nas tabelas, enquanto que aquelas que estabelecem “quanto menor, melhor” estão seguidas do símbolo (\downarrow), enquanto que a melhor avaliação de cada uma das métricas estará representada em negrito.

A Tabela 4.1 compara os métodos *Random Walker tradicional* (RW), *Leared Watershed* (LWS), e o *Método Proposto* (MP) através das médias e variância do valor dado pelas métricas citadas acima em sete volumes CT do conjunto de dados *MedSeg*. E por fim, as Figuras 4.12, 4.13 e 4.14 ilustram gráficos comparando quantitativamente os métodos RW, LWS, e a abordagem proposta, através das métricas descritas na Seção 4.1.2 em cada uma das fatias de 3 volumes CT distintos.

	MP	RW	LWS
(\uparrow) PRECISION	0.9512 \pm 0.0772	0.9540 \pm 0.1169	0.8254 \pm 0.2625
(\uparrow) RECALL	0.9420 \pm 0.1294	0.8085 \pm 0.1579	0.9445 \pm 0.0951
(\uparrow) F-SCORE	0.9419 \pm 0.1156	0.8681 \pm 0.1356	0.8398 \pm 0.2459
(\downarrow) VOI	0.1448 \pm 0.0621	0.2532 \pm 0.1140	0.2824 \pm 0.2324
(\uparrow) RI	0.9800 \pm 0.0099	0.9542 \pm 0.0252	0.9385 \pm 0.0878
(\uparrow) ARI	0.9231 \pm 0.1142	0.8280 \pm 0.1350	0.7863 \pm 0.2861

Tabela 4.1: Comparação quantitativa entre os método RW, LWS e o método proposto (MP). Fonte: Autoria Própria.

As segmentações do RW e LWS obtiveram a melhor avaliação de acordo com as métricas *Precision* e *Recall*, respectivamente. Entretanto, podemos observar que, em ambos os casos, o método proposto apresenta escores muito próximos ao melhor índice alcançado nos testes, enquanto que os métodos RW e o LWS apresentam resultados elevados de *Precision* ao custo do *Recall*, e vice-versa, ou seja, o método proposto promove um balanceamento ideal entre as medidas *Precision* e *Recall*, o que pode ser confirmado pelo seu escore obtido na métrica de sumário, *F-score*. Já nas métricas *VOI*, *RI* e *ARI* o método proposto apresenta resultados amplamente superiores aos obtidos pelos outros dois métodos comparados.

Por fim, nas Figuras 4.12, 4.13 e 4.14, é possível observar que a segmentação do método proposto mantém uma alta qualidade (e constante, sem variações bruscas entre diferentes *slices*) no decorrer das segmentações das fatias de cada um dos três volumes CT avaliados, diferente dos demais métodos, os quais apresentaram grandes discrepâncias no decorrer dos diferentes volumes explorados. Isso indica um aprendizado robusto dos pesos de borda por parte da abordagem proposta.

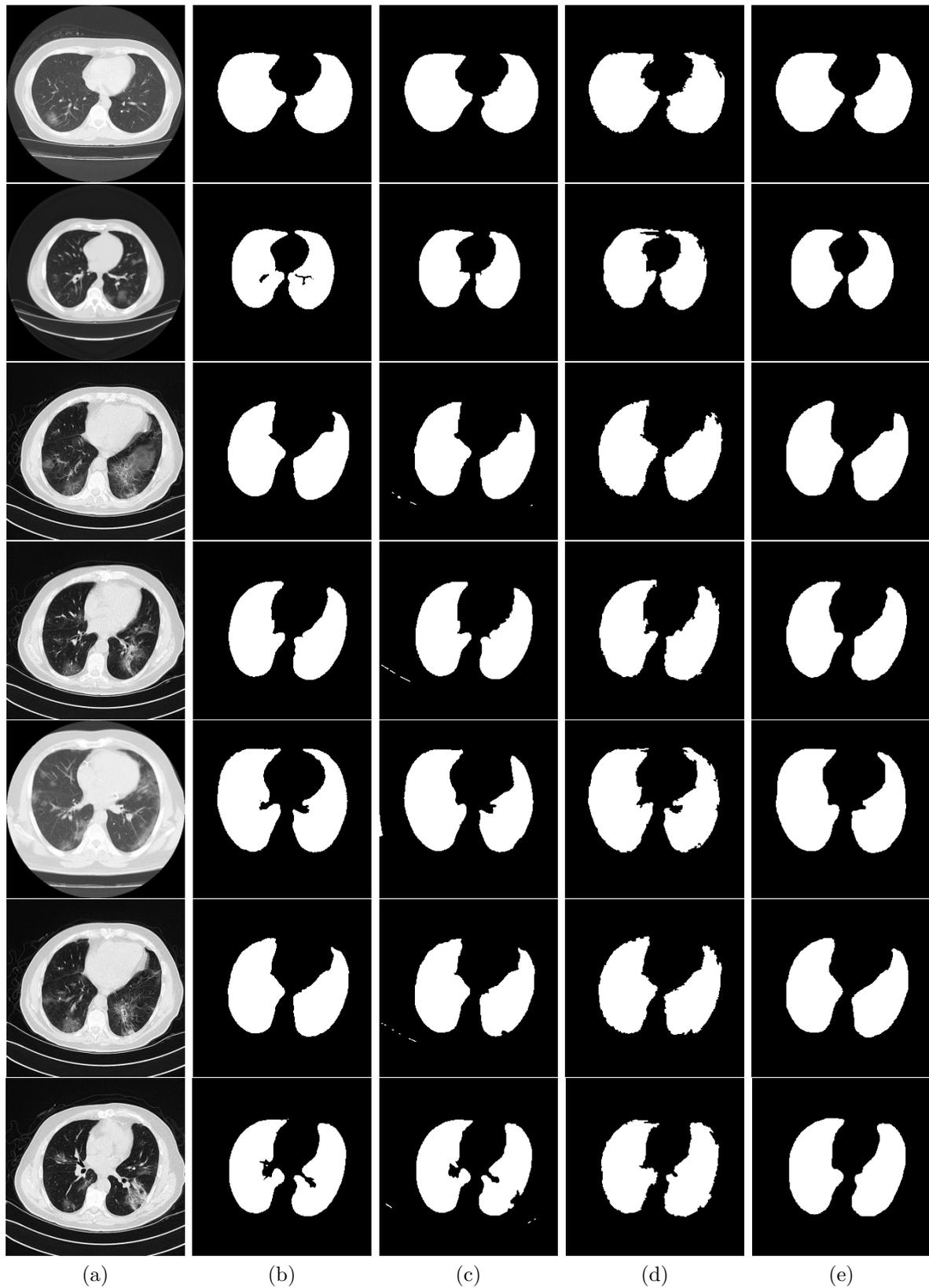


Figura 4.10: Comparação da segmentação de pulmão com lesões aparentes. (a) Imagens de entrada, (b) *ground truth*, (c) segmentações pelo método Random Walker, (d) segmentações pelo método *Learned Watershed* e (e) segmentação pelo método proposto. Fonte: Autoria própria.

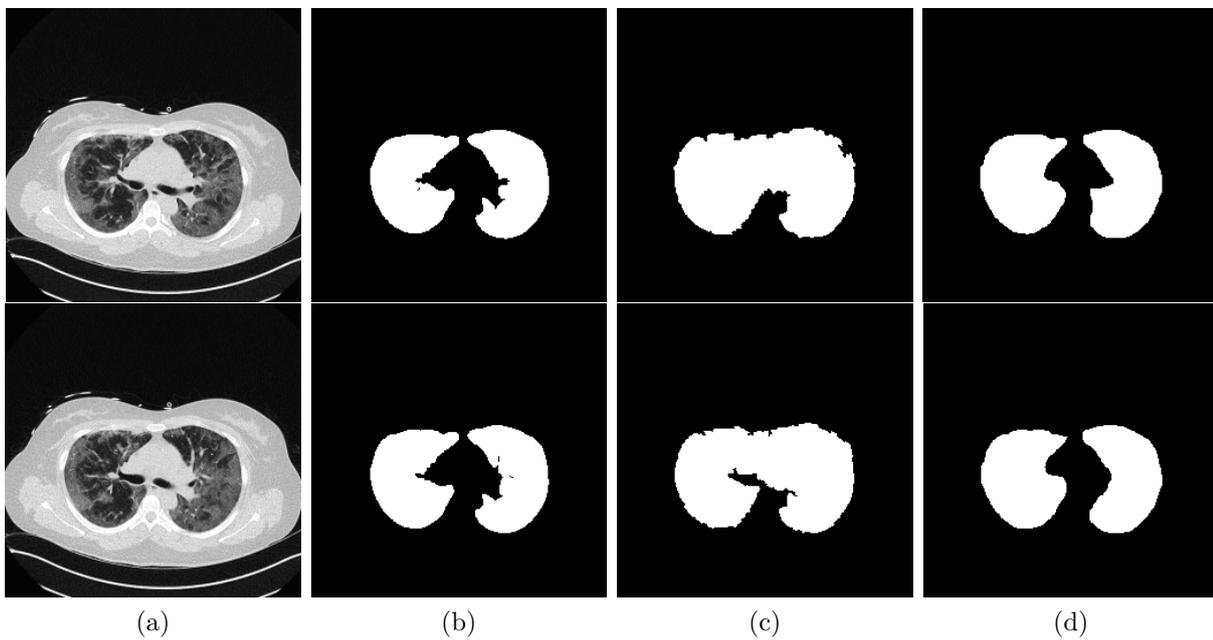


Figura 4.11: Comparação entre o método LWS e o método proposto. (a) Imagens de entrada, (b) *ground truth*, (c) segmentações pelo método Learned Watershed e (d) segmentações pelo método proposto. Fonte: Autoria própria.

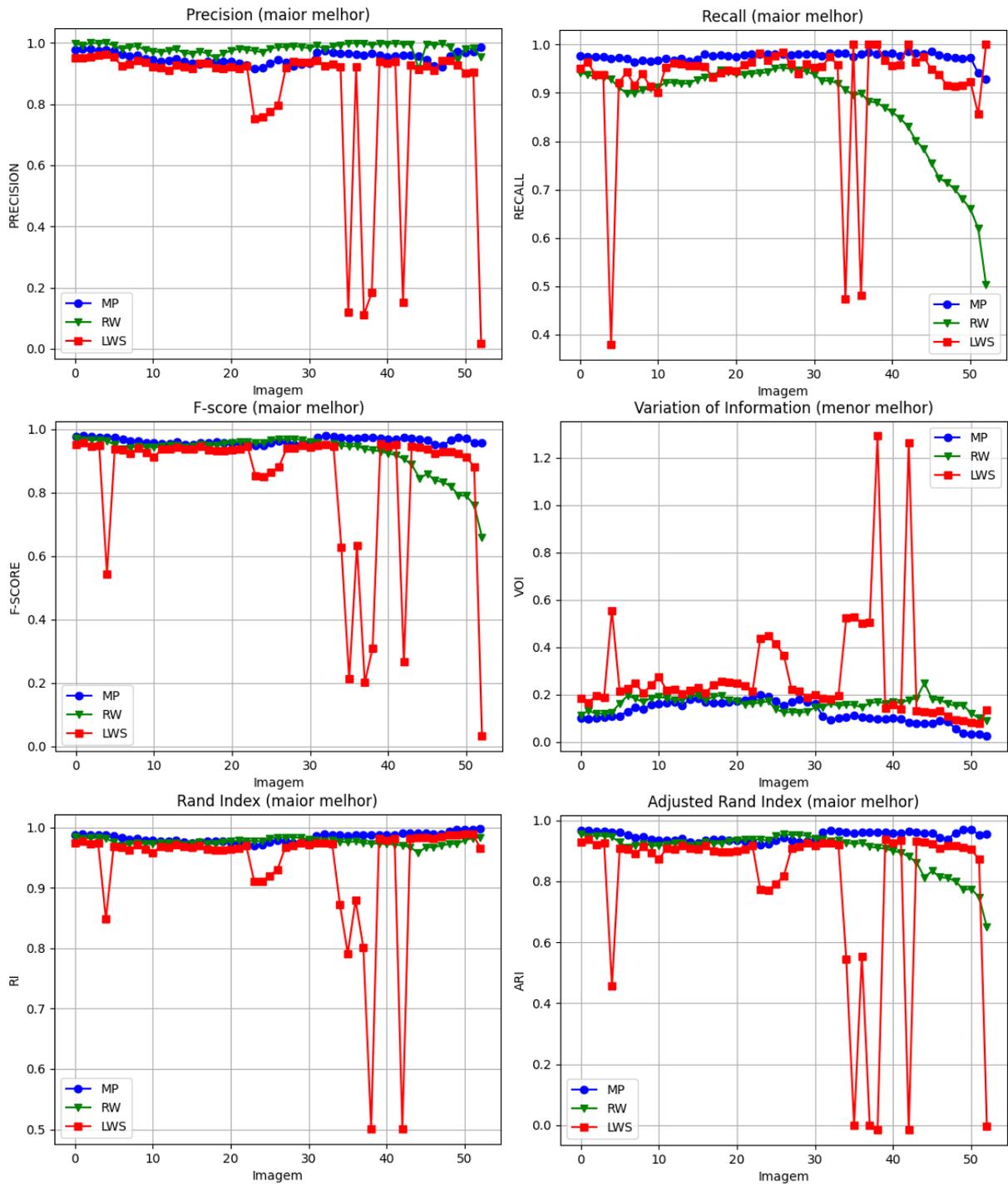


Figura 4.12: Comparação entre o método LWS, RW e o método proposto em cada imagem do volume CT de teste #1, através das métricas Precision, Recall, F-score, VOI, RI e ARI. Fonte: Autoria própria.

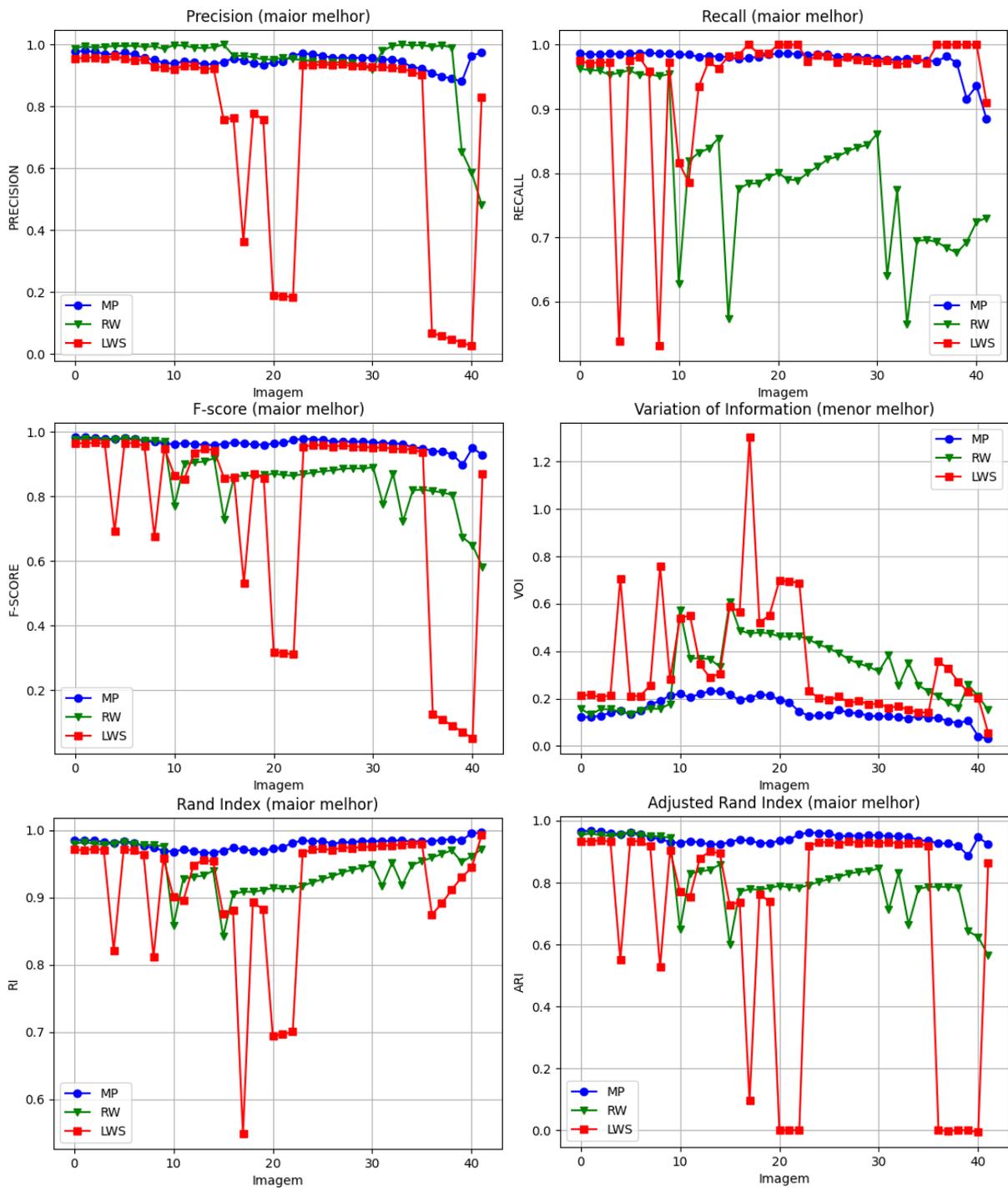


Figura 4.13: Comparação entre o método LWS, RW e o método proposto em cada imagem do volume CT de teste #2, através das métricas Precision, Recall, F-score, VOI, RI e ARI. Fonte: Autoria própria.

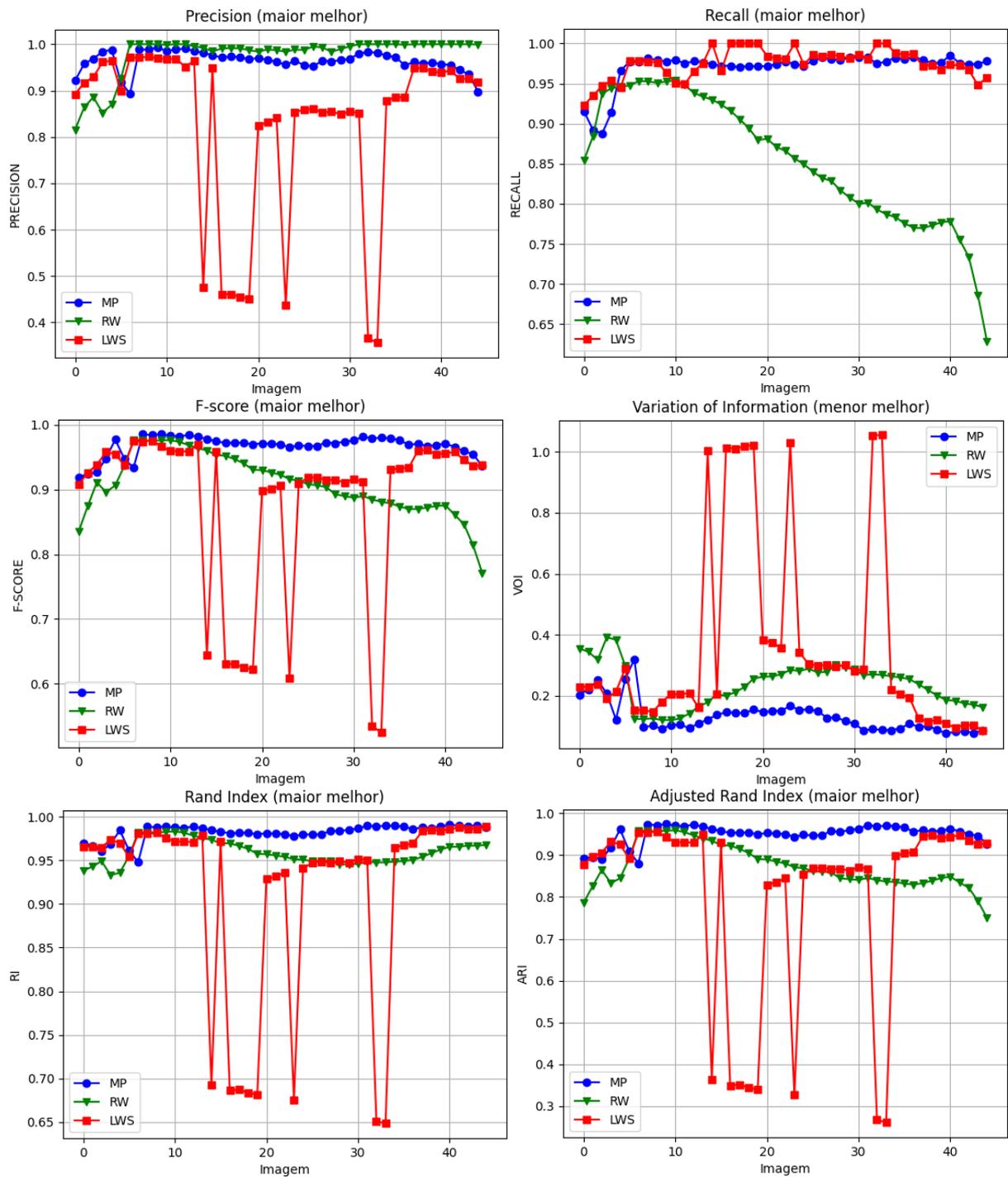


Figura 4.14: Comparação entre o método LWS, RW e o método proposto em cada imagem do volume CT de teste #3, através das métricas Precision, Recall, F-score, VOI, RI e ARI. Fonte: Autoria própria.

5 CONCLUSÃO

Neste dissertação, propusemos um algoritmo de aprendizado profundo, de ponta a ponta, para segmentação interativa de imagens volumétricas CT de pulmão, as quais apresentam danificações ocasionadas pelo novo corona vírus SARS-CoV-2. Foram propostas a otimização das iterações de treinamento do método, para que o mesmo pudesse aprender os pesos das bordas das imagens de maneira mais rápida e, ao mesmo tempo, robusta, além da abordagem proposta promover a criação de mapas de sementes de forma semi-automática, visando assim contornar a tarefa de semear cada fatia dos volumes CT.

A fim de melhorar a qualidade da segmentação, aplicamos uma pré-segmentação nas imagens através do método *Gaussian Mixture Model* (GMM), e então treinamos com sucesso um detector de bordas: uma rede CNN denominada *Convolutional Oriented Boundaries*, que foi integrada ao segmentador *Random Walker*, o que possibilitou preservar as “boas” características do referido segmentador.

A pré-segmentação via GMM permitiu remover ruídos e evitar tecidos que não fazem parte do pulmão, permitindo assim um aprendizado mais preciso. Já o treinamento da CNN integrada ao algoritmo *Random Walker*, juntamente com as técnicas de Gradiente Esperso e Poda do Gradiente apresentadas na Seção 3.1 permitiram reduzir o tempo de treinamento sem perder a qualidade do aprendizado dos pesos de bordas das imagens.

Em geral, Tomografias Computadorizadas Volumétricas são compostas por uma grande quantidade de fatias. Logo, métodos de segmentação semeados, apesar de efetivos, acabam demandando a tarefa repetitiva de se fornecer as sementes para cada uma das fatias. Com o intuito de facilitar essa tarefa massiva de dispêndio de tempo, de se fornecer as sementes para grandes volumes CT, propomos um método de sementeação quase automático, como foi mostrado na Seção 3.2. Na abordagem proposta, o usuário fornece as sementes para apenas uma fatia, e automaticamente as mesmas são expandidas para os demais *slices* da imagem volumétrica. Demonstramos experimentalmente que tal estratégia gera bons resultados, apesar de diferente da forma em que as sementes foram dadas para o treinamento do método.

Por fim, mostramos experimentalmente que o algoritmo proposto, através do aprendizado profundo, e com sementeação quase-automática, supera o desafio de segmentar corretamente o pulmão em imagens volumétricas CT de pacientes com COVID-19, independente do grau, quantidade e locais das lesões apresentadas no tecido do pulmão.

REFERÊNCIAS

- [1] CAMPION, N. R. Wellesley. Boston Globe, p. 1, 1983.
- [2] FACCIO, T. A. d. Análise de Características em Imagens Digitais na Tarefa de Segmentação Interativa. *Dissertação (Master Dissertation) — Universidade Estadual Paulista (UNESP), Faculdade de Ciências e Tecnologia (FCT), 2020.*
- [3] CASACA, W. Graph Laplacian for Spectral Clustering and Seeded Image Segmentation. *Tese (PhD Thesis) — Universidade de São Paulo (USP), Instituto de Ciências Matemáticas e de Computação (ICMC), São Carlos, São Paulo, 2015.*
- [4] ZHENG, S.-W.; LIU, J.; LIU, C.-C. A random-walk based breast tumors segmentation algorithm for mammograms. *International Journal on Computer, Consumer and Control, v. 2, n. 2, p. 66–74, 2013.*
- [5] COUSTY, J. et al. Watershed cuts: Minimum spanning forests and the drop of water principle. *IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 31, n. 8, p. 1362–1374, 2009.*
- [6] WOLF, S. et al. Learned Watershed: End-to-End Learning of Seeded Segmentation. 2017.
- [7] PAULA, D. D. d. Reconstrução de tomossíntese mamária utilizando redes neurais com aprendizado profundo. *Dissertação (Mestrado) — Universidade Estadual Paulista, Câmpus de São José do Rio Preto, 2020.*
- [8] GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep Learning. [S.l.]: MIT Press, 2016. [Http://www.deeplearningbook.org](http://www.deeplearningbook.org).
- [9] MANINIS, K.-K. et al. Convolutional oriented boundaries: From image segmentation to high-level tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), v. 40, n. 4, p. 819 – 833, 2018.*
- [10] CASACA, W. et al. Spectral image segmentation using image decomposition and inner product-based metric. *Journal of Mathematical Imaging and Vision, v. 45, n. 3, p. 227–238, 2013.*
- [11] CASACA, W.; COLNAGO, M.; NONATO, L. Interactive image colorization using laplacian coordinates. *In: Computer Analysis of Images and Patterns (CAIP). [S.l.]: Springer, 2015. p. 675–686.*

- [12] NEGRI, R. G.; SILVA, E.; CASACA, W. *Inducing contextual classifications with kernel functions into support vector machines*. IEEE Geoscience and Remote Sensing Letters, v. 15, n. 6, p. 962–966, 2018.
- [13] BASSO, D. et al. *Combining morphological filtering, anisotropic diffusion and block-based data replication for automatically detecting and recovering unscanned gaps in remote sensing images*. Earth Science Informatics, v. 1, n. 1, p. 1–14, 2021.
- [14] MOTTA, D.; CASACA, W.; PAIVA, A. *Vessel optimal transport for automated alignment of retinal fundus images*. IEEE Transactions on Image Processing, v. 28, n. 12, p. 6154–6168, 2019.
- [15] RAKESH, S.; MAHESH, S. *Nodule segmentation of lung ct image for medical applications*. Global Transitions Proceedings, v. 2, n. 1, p. 80–83, 2021.
- [16] GRADY, L. *Random walks for image segmentation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 28, n. 11, p. 1768–1783, 2006.
- [17] CASACA, W. et al. *Laplacian coordinates: Theory and methods for seeded image segmentation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 43, n. 8, p. 2665–2681, 2021.
- [18] CASACA, W.; NONATO, L. G.; TAUBIN, G. *Laplacian coordinates for seeded image segmentation*. In: IEEE Conference on Computer Vision and Pattern Recognition. [S.l.: s.n.], 2014. p. 384–391.
- [19] LEI, T. et al. *Adaptive morphological reconstruction for seeded image segmentation*. IEEE Transactions on Image Processing, v. 28, n. 11, p. 5510–5523, 2019.
- [20] COUPRIE, C. et al. *Power watershed: A unifying graph-based optimization framework*. IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 33, n. 7, p. 1384–1399, 2011.
- [21] ROGOWSKA, J. *Chapter 5 - overview and fundamentals of medical image segmentation*. In: BANKMAN, I. N. (Ed.). Handbook of Medical Image Processing and Analysis (Second Edition). Second edition. Burlington: Academic Press, 2009. p. 73–90. ISBN 978-0-12-373904-9.
- [22] ARMATO, S. G.; SENSAKOVIC, W. F. *Automated lung segmentation for thoracic CT impact on computer-aided diagnosis*. Academic radiology, v. 11, n. 9, p. 1011–1021, September 2004. ISSN 1076-6332. Disponível em: <<https://doi.org/10.1016/j.acra.2004.06.005>>.
- [23] BAĞCI, U. et al. *Computer-assisted detection of infectious lung diseases: A review*. Computerized Medical Imaging and Graphics, v. 36, n. 1, p. 72–84, 2012. ISSN 0895-6111.
- [24] FOSTER, B. et al. *A computational pipeline for quantification of pulmonary infections in small animal models using serial PET-CT imaging*. EJNMMI research, v. 3, p. 55, 07 2013.

-
- [25] DICHIOTTI, S. et al. Automated segmentation refinement of small lung nodules in CT scans by local shape analysis. *IEEE Transactions on Biomedical Engineering*, v. 58, n. 12, p. 3418–3428, 2011.
- [26] ELINOFF, J. M. et al. Recombinant human factor viia for alveolar hemorrhage following allogeneic stem cell transplantation. *Biology of Blood and Marrow Transplantation*, v. 20, n. 7, p. 969–978, 2014. ISSN 1083-8791.
- [27] GUO, Y. et al. Automated iterative neutrosophic lung segmentation for image analysis in thoracic computed tomography. *Medical Physics*, v. 40, n. 8, p. 081912, 2013.
- [28] NAKAGOMI, K. et al. Multi-shape graph cuts with neighbor prior constraints and its application to lung segmentation from a chest CT volume. *Medical Image Analysis*, v. 17, n. 1, p. 62–77, 2013. ISSN 1361-8415.
- [29] JALALI, Y. et al. Resbcdu-net: A deep learning framework for lung CT image segmentation. *Sensors*, v. 21, n. 1, 2021. ISSN 1424-8220.
- [30] SOFKA, M. et al. Multi-stage learning for robust lung segmentation in challenging CT volumes. *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, v. 14, p. 667–74, 01 2011.
- [31] CAN, Y. B. et al. Learning to segment medical images with scribble-supervision alone. In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Cham: Springer International Publishing, 2018. p. 236–244.
- [32] RAJCHL, M. et al. Deepcut: Object segmentation from bounding box annotations using convolutional neural networks. *IEEE Transactions on Medical Imaging*, v. 36, n. 2, p. 674–683, 2017.
- [33] ZHANG, L. et al. Self-learning to detect and segment cysts in lung ct images without manual annotation. In: *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. [S.l.: s.n.], 2018. p. 1100–1103.
- [34] MEDSEG; JENSSEN, H. B.; SAKINIS, T. MedSeg Covid Dataset 2. 2021.
- [35] KRUSKAL, J. B. On the shortest spanning tree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, v. 7, n. 1, p. 48–50, 1956.
- [36] PRIM, R. C. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, v. 36, n. 6, p. 1389–1401, 1957.
- [37] FU, Z.; WANG, L. Color Image Segmentation Using Gaussian Mixture Model and EM Algorithm. *Berlin, Heidelberg: Springer Berlin Heidelberg*, 2012. 61-66 p.
- [38] DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, [Royal Statistical Society, Wiley], v. 39, n. 1, p. 1–38, 1977.
- [39] BISHOP, C. *Pattern recognition and machine learning*. Spring, v. 16, 2006.
- [40] ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, v. 65, n. 6, p. 386–408, 1958.

- [41] SCHROFF, F.; KALENICHENKO, D.; PHILBIN, J. *Facenet: A unified embedding for face recognition and clustering*. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2015. p. 815–823.
- [42] WERBOS, P. *Beyond regression: new tools for prediction and analysis in the behavioral sciences*. Tese (Doutorado) — Harvard University, 1974.
- [43] HE, K. et al. *Deep residual learning for image recognition*. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2016.
- [44] YU, F.; KOLTUN, V. *Multi-scale context aggregation by dilated convolutions*. In: . [S.l.: s.n.], 2016.
- [45] DONG, C. et al. *An improved random walker with bayes model for volumetric medical image segmentation*. Journal of Healthcare Engineering, v. 2017, p. 1–11, 10 2017.
- [46] PRATONDO, A.; CHUI, C.-K.; ONG, S.-H. *Integrating machine learning with region-based active contour models in medical image segmentation*. Journal of Visual Communication and Image Representation, v. 43, p. 1–9, 2017. ISSN 1047-3203.
- [47] TAPPEN, M. F. et al. *Learning gaussian conditional random fields for low-level vision*. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. [S.l.: s.n.], 2007. p. 1–8.
- [48] KOMODAKIS, N.; TZIRITAS, G. *Image completion using efficient belief propagation via priority scheduling and dynamic pruning*. IEEE Transactions on Image Processing, v. 16, n. 11, p. 2649–2661, 2007.
- [49] AZEVEDO, S. C. de et al. *Shadow detection using object area-based and morphological filtering for very high-resolution satellite imagery of urban areas*. Journal of Applied Remote Sensing, SPIE, v. 13, n. 3, p. 1 – 16, 2019.
- [50] BRADSKI, G. *The OpenCV Library*. Dr. Dobb’s Journal of Software Tools, 2000.
- [51] MEILĂ, M. *Comparing clusterings—an information based distance*. Journal of Multivariate Analysis, v. 98, n. 5, p. 873–895, 2007. ISSN 0047-259X.
- [52] MIGNOTTE, M. *A label field fusion model with a variation of information estimator for image segmentation*. Information Fusion, v. 20, p. 7–20, 2014. ISSN 1566-2535.
- [53] BENEŠ, M.; ZITOVÁ, B. *Performance evaluation of image segmentation algorithms on microscopic image data*. Journal of Microscopy, v. 257, n. 1, p. 65–85, 2015.
- [54] UNNIKRISHNAN, R.; PANTOFARU, C.; HEBERT, M. *Toward objective evaluation of image segmentation algorithms*. IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 29, n. 6, p. 929–944, 2007.
- [55] YANG, A. Y. et al. *Unsupervised segmentation of natural images via lossy data compression*. Computer Vision and Image Understanding, v. 110, n. 2, p. 212–225, 2008. ISSN 1077-3142.
- [56] RAND, W. M. *Objective criteria for the evaluation of clustering methods*. Journal of the American Statistical Association, v. 66, n. 336, p. 846–850, 1971.