



UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Defesa de Dissertação de Mestrado do candidato Pedro Hiroshi Ely Ito, realizada em 20/10/2025.

Comissão Julgadora:

Prof. Dr. Cesar Henrique Comin (UFSCar)

Prof. Dr. Filipi Nascimento Silva (NU)

Prof. Dr. Pedro Henrique Bugatti (UFSCar)

O Relatório de Defesa assinado pelos membros da Comissão Julgadora encontra-se arquivado junto ao Programa de Pós-Graduação em Ciência da Computação.

UNIVERSIDADE FEDERAL DE SÃO CARLOS– UFSCAR
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA– CCET
DEPARTAMENTO DE COMPUTAÇÃO– DC
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO– PPGCC

Pedro Hiroshi Ely Ito

**Desenvolvimento de tarefas auxiliares e
técnicas de pós-processamento para
preservação da topologia em
segmentação de vasos sanguíneos**

Pedro Hiroshi Ely Ito

**Desenvolvimento de tarefas auxiliares e
técnicas de pós-processamento para
preservação da topologia em
segmentação de vasos sanguíneos**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências Exatas e de Tecnologia da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Visão Computacional

Orientador: Prof. Dr. Cesar Henrique Comin

São Carlos

2025

Resumo

Redes neurais convolucionais (CNNs) proporcionaram importantes avanços na segmentação de vasos sanguíneos, mas garantir a correta topologia e a continuidade da vasculatura segmentada continua sendo um desafio significativo. Artefatos estruturais podem comprometer análises subsequentes, criando a necessidade de métodos que abordem explicitamente a conectividade dos vasos. Nesse trabalho, propomos o método SkelAnchor, uma nova tarefa auxiliar projetada para melhorar a conectividade da segmentação através de uma etapa de pós-processamento após a segmentação de uma CNN. Primeiramente, investigamos métodos simples de melhoria da conectividade baseados em grafos e demonstramos suas limitações, evidenciando que eles produzem mapas descontínuos que são difíceis para as redes aprenderem. Para superar isso, o SkelAnchor introduz uma parametrização suave e contínua que codifica a distribuição do esqueleto local do vaso usando pontos âncora. Essa tarefa fornece à rede ricas informações topológicas e geométricas. Nossos resultados mostram que um modelo pode aprender com sucesso a tarefa SkelAnchor. Embora a tarefa auxiliar não altere as métricas de segmentação padrão, demonstramos através de exemplos qualitativos que a representação aprendida pode ser usada com sucesso na etapa de pós-processamento para reparar problemas de conectividade na segmentação final. A análise serve como uma prova de conceito para o uso de tarefas auxiliares baseadas no esqueleto para aprimorar a qualidade topológica da segmentação de vasos sanguíneos.

Palavras-chave: Vasos Sanguíneos, Segmentação, Topologia, Tarefas auxiliares, Redes Neurais.

Abstract

Convolutional neural networks (CNNs) have provided important advances in blood vessel segmentation, but ensuring the correct topology and continuity of the segmented vasculature remains a significant challenge. Structural artifacts can compromise subsequent analyses, creating the need for methods that explicitly address vessel connectivity. In this work, we propose the SkelAnchor method, a new auxiliary task designed to improve segmentation connectivity through a post-processing step after CNN-based segmentation. First, we investigate simple graph-based connectivity improvement methods and demonstrate their limitations, showing that they produce discontinuous maps that are difficult for networks to learn. To overcome this, SkelAnchor introduces a smooth and continuous parameterization that encodes the local vessel skeleton distribution using anchor points. This task provides the network with rich topological and geometrical information. Our results show that a model can successfully learn the SkelAnchor task. Although the auxiliary task does not change standard segmentation metrics, we demonstrate through qualitative examples that the learned representation can be successfully used in the post-processing step to repair connectivity problems in the final segmentation. The analysis serves as a proof of concept for the use of skeleton-based auxiliary tasks to enhance the topological quality of blood vessel segmentation.

Keywords: Blood Vessel, Segmentation, Topology, Auxiliary task, Neural Networks.

Lista de ilustrações

Figura 2.1 – Principais passos da análise de imagens de vasos sanguíneos. A imagem original passa por um processo de segmentação, a máscara binária obtida é utilizada para criar um esqueleto que por sua vez é representado por um grafo.	15
Figura 2.2 – Processo de refinamento para obter um grafo geométrico a partir de um topológico: dado um segmento definido pela aresta (u, v) , amostramos pixels a uma distância r um do outro.	16
Figura 2.3 – A diferença entre uma arquitetura focada em codificador e em decodificador é a existência da interação entre as tarefas no decodificador. . . .	17
Figura 2.4 – Esquema básico de um bloco residual. O segundo bloco convolucional tem sua saída somada ao valor de entrada, permitindo que o bloco se comporte como a função identidade se necessário.	19
Figura 2.5 – Esquema básico de um bloco convolucional. Uma convolução recebe c_{in} canais de entrada, gerando c_{out} canais de saída por meio de convoluções com filtros de tamanho $k \times k$. Quando especificado, a convolução possui passo s . Essa convolução é seguida de um C e a função de ativação <i>ReLU</i>	19
Figura 3.1 – Uma convolução deformável tende a agregar pixels longe da estrutura de interesse. A <i>snake convolution</i> se adapta ao formato tubular da estrutura, permitindo capturar sua complexa geometria.	23
Figura 4.1 – A partir do rótulo de uma imagem (a), obtemos o esqueleto (b) e o grafo topológico (c), dos quais podemos obter um mapa de segmentos dos vasos (d). Por fim, podemos obter o grafo geométrico (e) refinando (c) e utilizando (d).	28
Figura 4.2 – Dado um segmento definido pelos nós n_1 e n_2 , calculamos os deslocamentos entre os pixels desse segmento e os nós.	30
Figura 4.3 – Valores de deslocamento calculados para a imagem de exemplo.	30
Figura 4.4 – Ilustração da região anular e dos pontos âncora.	32

Figura 4.5 – Ilustração do cálculo dos pesos dos setores. Cada ponto $\bar{\theta}_s$ gera pesos correspondentes em dois setores. O valor do peso depende da distância aos respectivos pontos âncora.	34
Figura 4.6 – A arquitetura U-Net com um segundo decodificador para a tarefa auxiliar.	36
Figura 4.7 – Os estágios do codificador reduzem a resolução da imagem usando blocos convolucionais.	36
Figura 4.8 – Os estágios do decodificador aumentam a resolução da imagem (<i>upsample</i>) usando blocos convolucionais e concatenam as saídas dos estágios do codificador para as conexões de salto.	37
Figura 5.1 – Deslocamentos ao grafo topológico. Cada imagem representa uma das quantidades da Equação 4.2, isto é, a diferença das coordenadas de um pixel às coordenadas dos dois nós do grafo associados ao mesmo segmento de vaso. Nesse caso, (a) = $d_x^{n_1}$, (b) = $d_y^{n_1}$, (c) = $d_x^{n_2}$ e (d) = $d_y^{n_2}$.	40
Figura 5.2 – Deslocamentos ao grafo geométrico. Cada imagem representa uma das quantidades da Equação 4.3, isto é, a diferença das coordenadas de um pixel às de um nó do grafo. Nesse caso, (a) = $d_x^{n_1}$, (b) = $d_y^{n_1}$, (c) = $d_x^{n_2}$ e (d) = $d_y^{n_2}$	40
Figura 5.3 – Exemplo dos pesos do SkelAnchor para $n = 8$ setores.	41
Figura 5.4 – Exemplo de valores de deslocamento obtidos para o grafo topológico. As figuras à esquerda mostram o resultado da rede, enquanto que as figuras à direita mostram os valores de referência $d_x^{n_1}$ e $d_y^{n_1}$	42
Figura 5.5 – Exemplo de valores de deslocamento obtidos para o grafo geométrico. As figuras à esquerda mostram o resultado da rede, enquanto que as figuras à direita mostram os valores de referência $d_x^{n_1}$ e $d_y^{n_1}$	42
Figura 5.6 – Pesos SkelAnchor calculados para uma amostra. \hat{w}_0 e \hat{w}_1 indicam os resultados da rede, enquanto que os valores de referência são mostrados à esquerda.	43
Figura 5.7 – Curvas de precisão-revocação para diferentes valores do limiar de pesos l_{skel}	44
Figura 5.8 – Exemplos de pós-processamentos onde foi possível aprimorar a topologia da segmentação inicial. Em verde, pixels recuperados corretamente e em amarelo pontos adicionados erroneamente no pós-processamento.	45

Lista de siglas

CNN Rede Neural Convolutacional

clDice centerline Dice

IoU Índice de Jaccard

ReLU Unidade Linear Retificada

Sumário

1	INTRODUÇÃO	13
1.1	Contextualização e problemática	13
1.2	Objetivos e justificativa	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Grafos e Representação de Vasos Sanguíneos	15
2.2	Aprendizado Multitarefa	16
2.3	Redes Neurais Convolucionais	18
2.3.1	Blocos Básicos de uma Rede Neural Convolucional	18
2.4	Métricas de Qualidade	20
3	REVISÃO BIBLIOGRÁFICA	21
3.1	Segmentação de Vasos Sanguíneos	21
3.2	Preservação de Conectividade em Tarefas de Segmentação	22
3.3	Reconstrução e Pós-processamento	22
3.4	Tarefas Auxiliares para o Treinamento de Redes Neurais	23
3.5	Restrições de formato	24
4	MATERIAL E MÉTODOS	27
4.1	Descrição dos Dados	27
4.2	Construção do Grafo Geométrico	28
4.3	Parametrização por Deslocamentos ao Grafo Topológico	29
4.4	Parametrização por Deslocamentos ao Grafo Geométrico	31
4.5	Parametrização por pontos âncora: SkelAnchor	31
4.5.1	Pontos Âncora	32
4.5.2	Interpolação Linear	33
4.5.3	Cálculo do SkelAnchor	33

4.5.4	Recuperando os Eixos Mediais	35
4.6	Arquitetura da rede e função de perda	36
4.6.1	Função de perda	37
5	RESULTADOS E DISCUSSÃO	39
5.1	Parametrizações por Deslocamentos	39
5.2	Treinando a rede para localizar o esqueleto	41
5.3	Pós-processamento	43
6	CONSIDERAÇÕES FINAIS	47
	REFERÊNCIAS	49

Capítulo 1

Introdução

1.1 Contextualização e problemática

Vasos sanguíneos são estruturas do sistema circulatório responsáveis pelo transporte de sangue. A análise dessas estruturas é de grande interesse da medicina e biologia, permitindo detectar alterações vasculares (KAWATA; NIKI; KUMAZAKI, 1996; COMIN et al., 2021) e doenças como a retinopatia diabética (YANG et al., 2022). A base dessa análise é o processamento de imagens obtidas por diferentes modalidades de imageamento, de forma a obter representações que permitam a realização de uma série de medidas relevantes para caracterizar os vasos e possíveis anomalias. Por sua vez, esse processamento tem como fundamento a detecção e segmentação dos vasos, sendo este um dos primeiros passos da análise.

A identificação manual de vasos é uma tarefa tediosa e demorada, o que dificulta a análise de grandes volumes de imagens. Portanto, faz-se necessário o desenvolvimento de uma análise automatizada de vasos sanguíneos, em especial, métodos de segmentação automática. O problema de segmentação de vasos sanguíneos, no entanto, ainda é um problema em aberto, uma vez que a variabilidade de iluminação e contraste nas amostras, o fato de o problema ser naturalmente desbalanceado (em uma imagem, há mais pixels de fundo do que de vaso) e a presença de ruído nas imagens dificultam a obtenção de segmentações acuradas. A obtenção de segmentações acuradas de forma automática auxiliaria, por exemplo, no diagnóstico e análises posteriores de derrames (SACELEANU et al., 2023), no estudo em organismos modelo do fluxo de sangue do sistema cerebrovascular em indivíduos com deleção de genes relacionada à desordem do espectro autista (OUELLETTE et al., 2020) e na análise do desenvolvimento de organismos modelo sob diferentes condições de nutrição (BORDELEAU et al., 2022).

Nos últimos anos, as redes neurais convolucionais (CNNs) alcançaram um desempenho estado da arte na segmentação de vasos sanguíneos (GALDRAN et al., 2022a; CHEN et al., 2021). Um desafio persistente é garantir a correção topológica da segmentação, particularmente mantendo a continuidade dos vasos (DULAU et al., 2023a; ARAÚJO; CARDOSO; OLIVEIRA, 2019). Tais descontinuidades podem comprometer a precisão de análises subsequentes, tornando esta uma área crítica para aprimoramento.

1.2 Objetivos e justificativa

Uma direção promissora para abordar esse problema envolve o uso de tarefas auxiliares, onde um único modelo é treinado para realizar múltiplas tarefas simultaneamente a fim de melhorar a tarefa principal (CARUANA, 1997; VANDENHENDE et al., 2021). Essa melhoria pode ocorrer tanto guiando o modelo a aprender representações mais robustas durante o treinamento, quanto habilitando uma etapa de pós-processamento que refina o resultado da tarefa principal. Este trabalho explora o desenvolvimento de tarefas auxiliares que podem ser aproveitadas tanto durante o treinamento da rede quanto em uma etapa de pós-processamento para refinar os resultados da segmentação.

Iniciamos investigando duas abordagens que parametrizam os deslocamentos dos pixels do vaso para os nós em uma representação de grafo da vasculatura. Contudo, embora intuitivos, esses métodos geram mapas de deslocamento descontínuos, que são desafiadores para as redes neurais aprenderem com precisão. Para resolver isso, propomos o SkelAnchor, uma nova parametrização que cria uma representação suave e contínua da estrutura do vaso. O SkelAnchor define um referencial local de pontos âncora ao redor de cada pixel do vaso e utiliza um esquema de interpolação para codificar a localização do esqueleto próximo. Esse design evita dependências de longo alcance e produz uma representação mais adequada para o aprendizado de redes neurais, fornecendo ricas informações topológicas que podem ser usadas para refinar a segmentação inicial e melhorar a conectividade.

Demonstramos que nossa representação SkelAnchor pode ser aprendida de forma eficaz por uma rede, e subsequentemente, usada em uma etapa de pós-processamento para reparar erros topológicos, fornecendo uma prova de conceito para nossa abordagem.

Capítulo 2

Fundamentação Teórica

2.1 Grafos e Representação de Vasos Sanguíneos

A análise de imagens de vasos sanguíneos consiste numa sequência de tarefas que permite obter representações úteis da imagem (FREITAS-ANDRADE et al., 2022b). Um exemplo dessa sequência de tarefas pode ser visto na Figura 2.1. Essas representações permitem a realização de uma série de medidas morfológicas utilizadas na caracterização dos vasos sanguíneos, como tortuosidade, ramificação e densidade. Uma representação particularmente relevante é dada por grafos. Grafos são estruturas utilizadas para estudar relações entre pares de objetos. Mais formalmente, podemos definir um grafo como:

Definição 1 (Grafo) *Um grafo é uma tupla $(\mathcal{V}, \mathcal{E})$, onde $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Chamamos \mathcal{V} de conjunto dos nós (ou vértices) e \mathcal{E} de conjunto das arestas.*

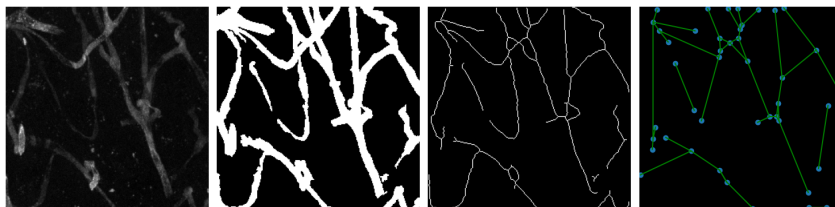


Figura 2.1 – Principais passos da análise de imagens de vasos sanguíneos. A imagem original passa por um processo de segmentação, a máscara binária obtida é utilizada para criar um esqueleto que por sua vez é representado por um grafo.

Grafos serão amplamente utilizados neste trabalho. Em particular, é útil definir duas representações:

Definição 2 (Grafo topológico (de um vaso sanguíneo)) *É possível representar vasos sanguíneos por um grafo (FREITAS-ANDRADE et al., 2022b), com \mathcal{V} sendo o conjunto dos pontos de bifurcação e terminação e \mathcal{E} o conjunto: $(u, v) \in \mathcal{E} \iff$ existe um segmento de vaso entre u e v , com $u, v \in \mathcal{V}$.*

Também definimos um refinamento dessa representação, inspirado pela representação definida em (ZHANG et al., 2024):

Definição 3 (Grafo geométrico (de um vaso sanguíneo)) *Dado um grafo topológico, amostramos pixels ao longo de um segmento definido por uma aresta do grafo topológico e adicionamos esses pixels como nós do grafo. A amostragem é feita de forma a manter uma distância fixa r entre os pixels.*

O processo de definição das arestas do grafo geométrico é ilustrado na Figura 2.2.

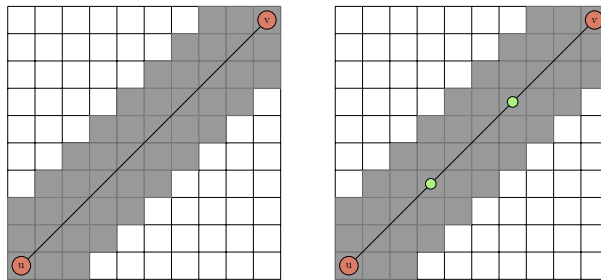


Figura 2.2 – Processo de refinamento para obter um grafo geométrico a partir de um topológico: dado um segmento definido pela aresta (u, v) , amostramos pixels a uma distância r um do outro.

Uma vez que as medidas morfológicas derivam do esqueleto e do grafo, a obtenção de uma segmentação acurada é fundamental para garantir a fidedignidade dessas representações. Em especial, falhas na conectividade da máscara binária alteram diretamente a topologia do grafo final, levando a uma representação imprecisa dos vasos.

2.2 Aprendizado Multitarefa

O aprendizado multitarefa consiste no aprendizado simultâneo de diversas tarefas por um único modelo. Esse tipo de método vem sendo explorado há décadas (CARUANA, 1997) e é observada uma emergência de trabalhos que aplicam aprendizado multitarefa para segmentação de imagens médicas (CONZE et al., 2023). Mais formalmente, seja \hat{f} um modelo que aprende T tarefas, para cada tarefa t , existe um conjunto de treinamento $S_t = (x_i^t, y_i^t)$ para qual definimos um processo de minimização de uma função de perda:

$$\arg \min_{\hat{f}} \sum_{t=1}^T w_t \mathcal{L}_t(\hat{f}(x_i), y_i), \quad (2.1)$$

onde w_t é o peso associado a cada tarefa. Existem alguns trabalhos que garantem teoricamente os benefícios do aprendizado multitarefa. Em (BEN-DAVID; SCHULLER, 2003), é definida uma noção de relacionamento entre tarefas de classificação. Dado um conjunto \mathcal{F} de transformações $f : \mathcal{X} \rightarrow \mathcal{X}$, diz-se que duas tarefas são \mathcal{F} -relacionadas se para uma dada distribuição sobre $\mathcal{X} \times \{0, 1\}$, os dados nessas tarefas podem ser gerados por alguma $f \in \mathcal{F}$ aplicada nessa distribuição. É demonstrado então que se \mathcal{F} é finito, existe uma vantagem em utilizar um modelo multitarefa. Em (MAURER; PONTIL; ROMERA-PAREDES, 2016), uma análise estatística das representações geradas por um modelo multitarefa é usada para gerar limites para a performance dessa classe de modelos, demonstrando rigorosamente sua vantagem computacional quando os dados são suficientemente complexos. Vários trabalhos exploram esses métodos com arquiteturas modernas de aprendizado profundo (RUDER, 2017; VANDENHENDE et al., 2021), também mostrando evidências empíricas dessas vantagens. Em especial, (VANDENHENDE et al., 2021) discute dois tipos principais de arquiteturas: as focadas no codificador e as focadas no decodificador. Ambas as arquiteturas compartilham o codificador, mas nas focadas em codificador não há interação entre as tarefas no decodificador. Já nas focadas em decodificador, há interação entre as tarefas em ambos os processos. Essa diferença é ilustrada na Figura 2.3. Como exemplo, em (XU et al., 2018), o decodificador utiliza atenção espacial (CHEN et al., 2017) para recombinar os atributos de diferentes tarefas.

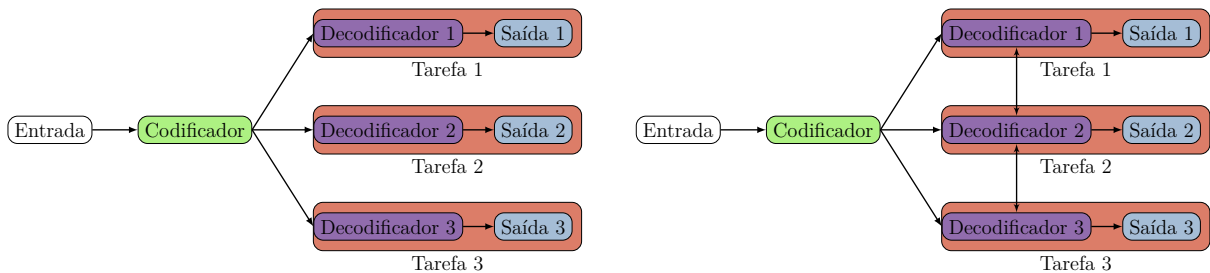


Figura 2.3 – A diferença entre uma arquitetura focada em codificador e em decodificador é a existência da interação entre as tarefas no decodificador.

No contexto de imagens, existem alguns exemplos de aplicação do aprendizado multitarefa. Em (GRAHAM et al., 2023), é utilizado um único modelo para segmentação e classificação de diferentes tipos de tecido em imagens de histologia, sendo notado que o processo de treinamento de tarefas simultâneas aprimora resultados de classificação. Já (HÖRST et al., 2024) realiza a segmentação, identificação da distância ao centro de massa e classificação de núcleos de células em imagens de histologia. Em (CARUANA, 1997) é indicado que o aprendizado multitarefa pode melhorar os resultados para uma tarefa

individual. Como notado por (BAXTER, 2000), o aprendizado de tarefas num mesmo contexto pode melhorar a generalização e convergência do modelo. Em (CARUANA, 1997), também é observado que as tarefas auxiliares podem ajudar a rede a focar em certos atributos, além de guiar o processo de treinamento para pesos úteis para ambas as tarefas, sendo portanto uma forma de impor um viés de representação.

2.3 Redes Neurais Convolucionais

Redes Neurais Convolucionais se tornaram ubíquas em tarefas de visão computacional desde a década passada (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), apresentando o estado da arte em diversas tarefas. Nesta seção, são apresentados os blocos fundamentais de uma arquitetura moderna de rede convolucional, assim como o modelo a ser usado como base para as tarefas propostas neste trabalho.

2.3.1 Blocos Básicos de uma Rede Neural Convolucional

A arquitetura ResNet, introduzida em (HE et al., 2016), tornou-se notória na década passada por expandir a capacidade de criação de redes neurais profundas. A introdução de conexões residuais permite que as camadas da rede aprendam a função de identidade. Isso é feito pela soma da entrada recebida por um bloco de rede à saída deste, isto é, dado uma sequência de camadas $\mathcal{H}(x)$, um bloco residual tem como saída:

$$y = \mathcal{H}(x) + x. \quad (2.2)$$

Existem algumas hipóteses sobre como conexões residuais auxiliam o treinamento de redes neurais. Inicialmente se pensava em termos dos valores dos gradientes, ou seja, as conexões residuais surgem como uma forma de evitar que os gradientes fossem zerados (gerando platôs na superfície da função de perda) ou explodissem. No entanto, foi descoberto que é possível inicializar a rede de forma a sanar esses problemas de gradiente e que conexões residuais por si só não permitem treinar redes mais profundas, pois não evitam a explosão dos gradientes (PRINCE, 2023), sendo conjecturado que redes com muitos parâmetros geram uma superfície da função de perda irregular. Assim, apesar da eficácia de conexões residuais não ser totalmente compreendida, é observado que essa técnica torna a superfície da função de perda mais suave (LI et al., 2018), facilitando o treinamento de redes mais profundas, mesmo sendo necessárias outras regularizações para estabilizar o treinamento da rede. Na Figura 2.4 é ilustrado um bloco de uma rede convolucional com uma conexão residual.



Figura 2.4 – Esquema básico de um bloco residual. O segundo bloco convolucional tem sua saída somada ao valor de entrada, permitindo que o bloco se comporte como a função identidade se necessário.

Outro método comumente aplicado é o *batch norm* (IOFFE; SZEGEDY, 2015), que consiste numa normalização para um lote de dados. O efeito dessa normalização ainda é discutido, sendo a explicação inicial da ligação com o deslocamento interno da covariável, isto é, a mudança na distribuição dos parâmetros da rede devido ao processo de treinamento, bastante questionada nos últimos anos (LIPTON; STEINHARDT, 2019), com alguns trabalhos apontando a inexistência dessa ligação, atribuindo seus efeitos à uma reparametrização do problema de otimização e a suavização da superfície da função de perda (SANTURKAR et al., 2018). Apesar da controvérsia, o *batch norm* é uma componente essencial em arquiteturas modernas de redes neurais, possibilitando estabilidade e uma convergência mais rápida no treinamento. Como averiguado em (YANG et al., 2019), utilizar apenas *batch norm* causa a explosão dos gradientes, no entanto, o uso em conjunto com conexões residuais permite contornar esse problema, possibilitando o treinamento de redes neurais mais profundas. Dado um lote de dados, o *batchnorm* transforma seus elementos de acordo com:

$$y = \frac{x - E[X]}{\sqrt{\text{Var}[x] + \epsilon}} \quad (2.3)$$

A ativação *ReLU* (FUKUSHIMA, 1969) é uma função de ativação amplamente utilizada, oferecendo uma boa relação entre tempo de treinamento e acurácia (DUBEY; SINGH; CHAUDHURI, 2022). A ativação é dada por:

$$\text{ReLU}(x) = \max(0, x) \quad (2.4)$$

Portanto, um bloco básico em uma CNN é dado pela combinação de uma camada de convolução, seguida pelo *batch norm* e *ReLU*, como ilustrado na Figura 2.5.

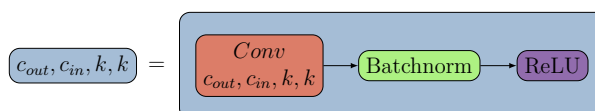


Figura 2.5 – Esquema básico de um bloco convolucional. Uma convolução recebe c_{in} canais de entrada, gerando c_{out} canais de saída por meio de convoluções com filtros de tamanho $k \times k$. Quando especificado, a convolução possui passo s . Essa convolução é seguida de um C e a função de ativação *ReLU*.

2.4 Métricas de Qualidade

Existem diferentes métricas de qualidade em tarefas de segmentação. Aqui apresentamos definições básicas de erros em classificação binária e uma série de métricas que podem ser usadas ao avaliar os modelos propostos neste trabalho.

Definição 4 (Erros em classificação binária) *Seja um problema de classificação binária com previsão \hat{y} , rótulos y e classes $\{0, 1\}$. Temos 4 erros primários:*

$$\begin{aligned} TP &= |\{\hat{y}_i | \hat{y}_i = y_i = 1\}| \\ TN &= |\{\hat{y}_i | \hat{y}_i = y_i = 0\}| \\ FP &= |\{\hat{y}_i | \hat{y}_i = 1, y_i = 0\}| \\ FN &= |\{\hat{y}_i | \hat{y}_i = 0, y_i = 1\}| \end{aligned} \quad (2.5)$$

Esses erros são, em ordem: verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos.

A partir desses erros, definimos algumas métricas:

Definição 5 (Precisão) *A precisão mede a fração de verdadeiros positivos com relação à todas as previsões.*

$$\text{Precisão} = \frac{TP}{TP + FP} \quad (2.6)$$

Definição 6 (Revocação) *A revocação mede a fração de positivos detectados.*

$$\text{Revocação} = \frac{TP}{TP + FN} \quad (2.7)$$

Definição 7 (Coeficiente de Dice) *O coeficiente de Dice (DICE, 1945; SØRENSEN et al., 1948) mede a similaridade entre duas amostras por meio da similaridade entre conjuntos.*

$$\text{Dice} = \frac{2TP}{2TP + FP + FN} \quad (2.8)$$

Definição 8 (clDice) *O clDice (SHIT et al., 2021) mede a similaridade por meio dos eixos mediais (esqueleto) de duas máscaras binárias de uma tarefa de segmentação. Sejam S_y e $S_{\hat{y}}$ os esqueletos do rótulo e da previsão:*

$$\begin{aligned} T_{prec} &= \frac{|S_{\hat{y}} \cap y|}{|S_{\hat{y}}|} \\ T_{sens} &= \frac{|S_y \cap \hat{y}|}{|S_y|} \\ clDice &= 2 \frac{T_{prec} T_{sens}}{T_{prec} + T_{sens}} \end{aligned} \quad (2.9)$$

Capítulo 3

Revisão Bibliográfica

3.1 Segmentação de Vasos Sanguíneos

A segmentação automática de vasos sanguíneos é um problema amplamente explorado atualmente. Nos últimos anos, diversos métodos baseados em aprendizado profundo foram propostos para obter segmentações mais acuradas (KUMAR; SINGH, 2023; MOCCIA et al., 2018). Em particular, CNNs apresentam o estado da arte, recebendo grande atenção nos últimos anos. Dessa forma, desde a introdução da U-Net (RONNEBERGER; FISCHER; BROX, 2015), diversas arquiteturas foram propostas como alterações da U-Net (SIDDIQUE et al., 2021; AZAD et al., 2022). Dentre os mecanismos utilizados no aprimoramento da U-Net, podemos destacar o uso de filtros dilatados (HUSSAIN et al., 2022; PANCHAL; KOKARE, 2024) e operações de atenção (GUO et al., 2021b; GUO et al., 2021a) que visam aumentar o campo receptivo da rede. Apesar dessa variedade de métodos, trabalhos recentes como (ISENSEE et al., 2024) demonstram que U-Nets baseadas em CNNs seguem obtendo performance superior em diversas tarefas quando avaliadas mais rigorosamente. O trabalho aponta a necessidade de utilizar bases de referência bem definidas e da utilização de conjuntos de dados mais diversos. Em (GALDRAN et al., 2022b) é constatado que arquiteturas de U-Net mais minimalistas conseguem performance superior a diversas variantes quando o treinamento e validação são rigorosamente protocolados e aplicados, utilizando mesmas funções de perda, algoritmo de otimização, taxa de aprendizado e fazendo a avaliação em diferentes conjuntos de dados.

3.2 Preservação de Conectividade em Tarefas de Segmentação

Diversos métodos foram propostos na literatura com o objetivo de obter segmentações topologicamente acuradas, especialmente no contexto de estruturas tubulares. Uma das estratégias mais exploradas é a criação de funções de perda que levem em conta a topologia da imagem. Em (HU et al., 2019) é utilizada homologia de persistência (EDELSBRUNNER; HARER, 2022), que permite encontrar valores que caracterizam a topologia de um conjunto de dados, para definir uma função de perda que compara as topologias do rótulo da segmentação e das probabilidades de classe geradas por uma rede neural. Já em (GUPTA et al., 2024) é utilizada Teoria de Morse Discreta (FORMAN, 2002) para explorar a topologia da saída de uma rede neural, reforçando regiões definidas como topologicamente críticas durante o treinamento da rede. Outra estratégia pode ser encontrada em (SHIT et al., 2021), onde a função de perda leva em conta a interseção do esqueleto da previsão de uma rede neural e o rótulo e vice-versa, de forma a garantir uma consistência entre a topologia das duas imagens. Outras abordagens buscam sanar limitações das camadas de convolução. Em (JIANG et al., 2019), são utilizadas convoluções dilatadas para expandir o campo receptivo da rede, agregando informação do contexto global da imagem. Já em (DEARI; OKSUZ; ULUKAYA, 2023) é usado o mecanismo de atenção nas *skip-connections* de uma U-Net para aquisição de informação global durante a tarefa de segmentação. O mecanismo de atenção tem sido explorado no contexto de segmentação de vasos sanguíneos, por exemplo, em (MOU et al., 2019b) dois módulos de atenção são utilizados para modelar as relações espaciais e entre-canais dos atributos extraídos pelo codificador de uma U-Net. Em (QI et al., 2023) ambas essas estratégias são empregadas. Uma camada de convolução modificada, chamada de *snake convolution*, é proposta. A camada é semelhante a convoluções deformáveis (DAI et al., 2017), que permitem que os filtros das camadas de convoluções amostram informação de forma espacialmente irregular por meio de deslocamentos nas posições dos pixels agregados. A camada *snake convolution* restringe os deslocamentos nos filtros de forma a se adaptar à geometria de uma estrutura tubular, como ilustrado na Figura 3.1. Além disso, é utilizada homologia de persistência para definir uma função de perda sensível à topologia da imagem.

3.3 Reconstrução e Pós-processamento

Outros trabalhos também propõem um processo de refinamento da segmentação inicialmente obtida pela rede. Em (MOU et al., 2019a) é proposto identificar obstruções por meio de operações morfológicas e então identificar a direção de um vaso obstruído por meio de uma interpolação por polinômio de segundo grau. A partir da região da obstrução e da direção, o vaso é reconstruído por meio de caminhadas aleatórias guiadas

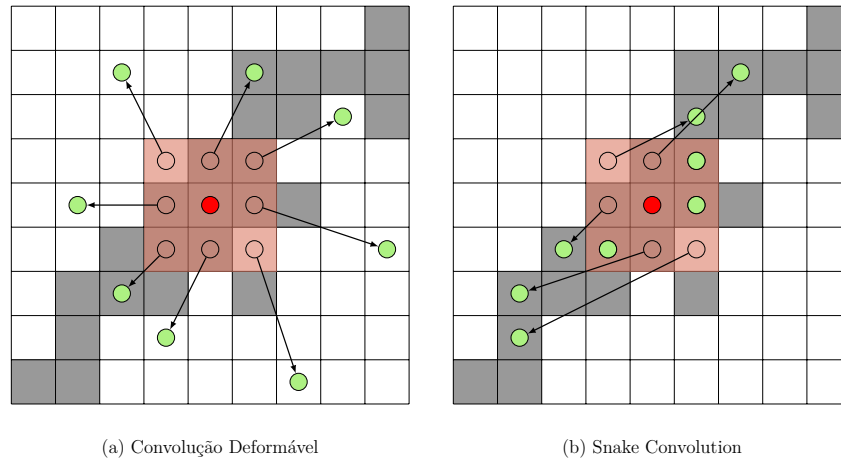


Figura 3.1 – Uma convolução deformável tende a agregar pixels longe da estrutura de interesse. A *snake convolution* se adapta ao formato tubular da estrutura, permitindo capturar sua complexa geometria.

pelos mapas de probabilidade da rede. De forma semelhante, (DU et al., 2023) identifica a direção principal do vaso usando interpolação, mas também levando em conta pontos de bifurcações próximos e realiza o processo de reconstrução por meio de estimativas da espessura do vaso. Em (DULAU et al., 2023b) componentes conexas da máscara binária obtida pela rede são analisadas e a partir de critérios de tamanho e posição relativa a outras componentes são reconectadas. Uma componente menor é reconectada a uma maior ao longo de uma reta entre pontos de terminação por meio da replicação de um bloco 5×5 extraído da componente menor. O processo também remove algumas das componentes que não atendem a esses critérios. Para estruturas curvilíneas, é proposto em (LIU; ZHAO; ZHENG, 2024) que uma rede aprenda atributos geométricos e de imagem para construir uma segmentação final. A rede realiza três tarefas: a extração do eixo medial e a direção e raio (espessura) do vaso em cada ponto. A direção e raio são definidas por um modelo geométrico obtido por meio do triedro de Frenet (uma base ortonormal definida em cada ponto de uma curva suave) e da discretização dos vasos em cilindros. A partir desse modelo geométrico e do eixo medial, é construída uma máscara binária.

3.4 Tarefas Auxiliares para o Treinamento de Redes Neurais

Alguns estudos propõem que o aprendizado multitarefa de redes neurais possibilita aprimorar os resultados de segmentações de imagens médicas. Em (DANGI; LINTE; YANIV, 2019), uma tarefa auxiliar que aprende um mapa de distâncias entre os pixels do vaso e os pixels da borda é utilizada como um regularizador da rede, permitindo que atributos globais sejam aprendidos. A tarefa auxiliar é realizada por um decodificador

paralelo ao decodificador de segmentação, sendo que ambos compartilham um único codificador. Dessa forma, ao aprender uma tarefa que envolve informações como formato, posição e borda do objeto segmentado, o codificador e o decodificador de segmentação também aprendem atributos úteis para a tarefa principal. No contexto de estruturas tubulares em imagens médicas, (WANG et al., 2020) propõe que a rede aprenda, além da segmentação, um mapa de distância que posteriormente é utilizado para refinar a segmentação. O pós-processamento é feito a partir de um pseudoesqueleto obtido a partir dos mapas de probabilidade criados pela rede. Esse pseudoesqueleto é utilizado para criar uma reconstrução da estrutura tubular que por fim é utilizada para refinar o resultado da segmentação original. Em (JIANG et al., 2024) são propostas duas tarefas auxiliares: um mapa de raios, que essencialmente computa um mapa de distâncias, e um mapa de orientações, que codifica localmente a direção de uma estrutura tubular. A partir dessas duas informações, é possível reconstruir a estrutura a partir de um algoritmo que traça esferas entre pontos de bifurcação e terminação com raios estimados pelo mapa de distâncias. Essa reconstrução é então utilizada para refinar o resultado da tarefa principal de segmentação.

3.5 Restrições de formato

A restrição de formato é uma tarefa comum em processamento de imagens, tendo sido desenvolvidas diversas técnicas desde o século passado (ZHANG; LU, 2004). Recentemente, alguns desses métodos têm sido utilizados em conjunto com redes neurais em tarefas de segmentação (KERVADÉC et al., 2021), permitindo adicionar restrições geométricas ao treinamento da rede, possibilitando criar funções de perda que não levem em conta apenas relações pixel a pixel, mas também o formato global do objeto. Em (CHEN et al., 2014), é utilizado um descritor de formato para a detecção automática de pontos de referência e segmentação de formato. São amostradas regiões quadradas da imagem e são calculados os deslocamentos entre um ponto de referência e os centros dessas regiões. A rede então infere para várias regiões aleatórias da imagem o deslocamento a um suposto ponto de interesse, definindo uma distribuição de onde é possível extrair o ponto para o qual mais regiões “concordam” que é um ponto de interesse. Os pontos de interesse são usados como uma representação esparsa da borda do objeto, que pode então ser usada para segmentação. De forma semelhante aos mapas de distância, outro descritor proposto recentemente em (KORDON et al., 2023) utiliza mapas de calor para representar estruturas alongadas. Pontos são uniformemente amostrados ao longo da estrutura e uma interpolação por splines é feita para obter uma curva de referência. O mapa de calor então é definido como a distância local (dentro do fecho convexo da região delimitada por dois pontos consecutivos) dos pontos da estrutura para a curva. Essa representação é proposta como alternativa à máscaras binárias como rótulo em tarefas de segmentação por redes neurais profundas.

Em (KEUTH et al., 2024), uma segunda tarefa que permite a regressão de pontos de referência é realizada em conjunto com a segmentação. A segunda tarefa consiste em aprender um mapa obtido pelos vetores de deslocamento de um conjunto de pontos inicial até o conjunto de pontos de referência. Esses vetores são interpolados em um campo vetorial que é usado para deformar um sistema de coordenadas. A rede então aprende o mapa obtido pela aplicação dessa deformação na identidade. Isso é feito de forma que o mapa seja inversível, permitindo então obter os pontos de referência a partir desse mapa.

Apesar de existirem alguns trabalhos visando utilizar tarefas auxiliares na obtenção de segmentações geometricamente e topologicamente acuradas, observamos que existem poucos trabalhos que utilizam diretamente a informação topológica (grafo) para elaborar a tarefa auxiliar. Além disso, do ponto de vista de pós-processamento e reconstrução, existem processos bastante complexos que algumas vezes exigem suposições sobre a geometria dos vasos, levantando a questão da existência de métodos mais generalistas. Outra observação é a aparente vantagem em manter o modelo próximo a arquiteturas já estabelecidas (U-Net), que aparentam apresentar desempenho superior em configurações experimentais mais rigorosas. Dessa forma, é interessante explorar a elaboração de tarefas auxiliares baseadas na topologia/geometria dos vasos que busquem minimizar as hipóteses sobre seu formato e a definição de modelos que preservem os princípios básicos de arquiteturas que mostram superioridade empírica.

Capítulo 4

Material e Métodos

4.1 Descrição dos Dados

As imagens utilizadas neste projeto fazem parte do conjunto VessMAP (SILVA et al., 2023) e foram obtidas por microscopia confocal do sistema vascular do cérebro de ratos sob diferentes condições. O conjunto usado, contando com 100 imagens de tamanho 256×256 pixels, consiste numa amostragem de um conjunto maior inicial feita de forma a obter um conjunto heterogêneo, possibilitando obter bons resultados no treinamento de redes neurais com poucas amostras de treinamento. Além do conjunto original de imagens, constam também segmentações manuais das imagens. A partir do rótulo, extraímos o esqueleto (ZHANG; SUEN, 1984) e a representação por grafo (FREITAS-ANDRADE et al., 2022a). A partir dos nós do grafo, são definidos segmentos dos vasos e o grafo geométrico. A Figura 4.1 ilustra esses dados. A partir dos grafos e segmentos são calculadas parametrizações da imagem: mapas que atribuem a cada pixel do vaso a diferença entre as coordenadas (deslocamento) a um nó do grafo topológico, geométrico ou ainda à reta que liga esses nós. Essas parametrizações são utilizadas no treinamento multitarefa da rede como tarefas auxiliares.

Nas seções seguintes, são detalhadas as construções do grafo geométrico como um refinamento do grafo topológico, sendo apresentado o algoritmo desenvolvido e os dados necessários. Em seguida, definimos as parametrizações a serem aprendidas pelo modelo como tarefas auxiliares. São propostas três parametrizações baseadas em limitações encontradas no treinamento de uma rede neural, como dependências de longo alcance, uniformidade e suavidade dos dados.

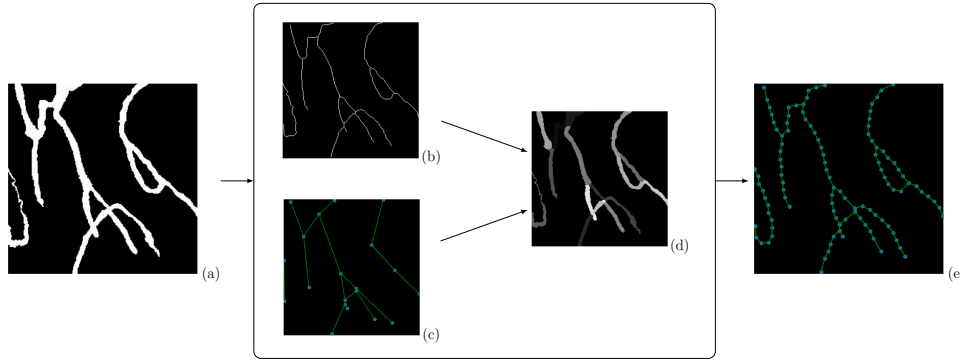


Figura 4.1 – A partir do rótulo de uma imagem (a), obtemos o esqueleto (b) e o grafo topológico (c), dos quais podemos obter um mapa de segmentos dos vasos (d). Por fim, podemos obter o grafo geométrico (e) refinando (c) e utilizando (d).

4.2 Construção do Grafo Geométrico

A obtenção do grafo topológico é uma tarefa já conhecida da literatura (FREITAS-ANDRADE et al., 2022b). Como já estabelecido, é de interesse deste trabalho refinar esse grafo para capturar a geometria dos vasos. Portanto, propomos um algoritmo para esse refinamento. O processo consiste em particionar cada segmento do vaso em sub-segmentos de tamanho r . As terminações de cada sub-segmento são adicionadas como nós do grafo. Dividimos o algoritmo em dois procedimentos: a obtenção dos pixels ao longo de um segmento e a adição dos pixels como nós do grafo.

Algorithm 1 GeraPosiçõesGeométricas

Entrada: P, r

Saída: pos

```

1  $pos \leftarrow \{\}$ 
2  $nó \leftarrow P[0]$ 
3 para  $i, pixel \in enumerar(P)$  faça
4   se  $d(pixel, nó) > r$  então
5      $pos[i] \leftarrow pixel$ 
6      $nó \leftarrow pixel$ 
7   fim
8 fim
9 retorna  $pos$ 

```

A função GeraPosiçõesGeométricas recebe como entrada um caminho P que consiste numa sequência de coordenadas $(x_i, y_i) \in \mathbb{R}^2$ e um raio $r \in \mathbb{R}_{\geq 0}$. Começando pelo primeiro ponto de P , é buscado o ponto mais próximo ao primeiro ponto mas com distância $> r$ a ele. Ao ser encontrado, o ponto e o respectivo índice em P é salvo num dicionário. O processo então se repete para o último nó adicionado ao dicionário. Ao fim, o dicionário com os índices e posições é retornado.

Algorithm 2 GeraGrafoGeométrico**Entrada:** G, r **Saída:** pos

```

10  $arestas \leftarrow \mathcal{E}$ 
11  $postop \leftarrow \mathcal{V}$  para  $n_1, n_2, caminho \in \mathcal{E}$  faça
12    $pos_{geo} = \text{GeraPosiçõesGeométricas}(caminho, r)$ 
13    $nós_{geo} = postop[n_1] \cup val(pos_{geo}) \cup postop[n_2]$ 
14    $nós_{id} = \{n_1\} \cup chave(pos_{geo}) \cup \{n_2\}$     $novo_{id} = \{n_1\} \cup \{|G|, |G| + 1, \dots, |G| + |nós_{geo}|\} \cup$ 
      $\{n_1\}$ 
15   para  $j \in \{1, \dots, |nós_{geo}|\}$  faça
16      $\mathcal{V} \cup (novo_{id}[j], pos_{geo}[j])$ 
17      $\mathcal{E} \cup ((novo_{id}[j], novo_{id}[j - 1]), caminho[nós_{id}[j - 1] : nós_{id}[j]])$ 
18   fim
19 fim
20 retorna  $G$ 

```

A função *GeraGrafoGeométrico* recebe um grafo $G = (\mathcal{V}, \mathcal{E})$ e um raio r . Iterando por todas as arestas de G , é usada a função *GeraPosiçõesGeométricas* para encontrar posições equidistantes ao longo do segmento definido pela aresta. Os nós são então adicionados a \mathcal{V} e arestas são adicionadas sequencialmente a \mathcal{E} , isto é, começando por um nó do grafo original, são criadas arestas entre os nós gerados por *GeraPosiçõesGeométricas*, até que se chegue ao próximo nó do grafo original. Nesses algoritmos, consideramos que um nó do grafo é definido pela tupla (id, pos) , onde id é um inteiro de identificação e pos é a posição do pixel correspondente na imagem, enquanto uma aresta é composta por $((u, v), caminho)$, onde u, v são identificações de nós do grafo e $caminho$ é uma lista com todas as posições dos pixels do segmento definido por u e v .

4.3 Parametrização por Deslocamentos ao Grafo Topológico

A primeira parametrização proposta consiste em, dado um grafo topológico (Definição 2), para cada pixel $p = (p_x, p_y)$ num segmento definido por uma aresta (n_1, n_2) , são calculados os vetores de deslocamento entre o pixel e os nós, isto é:

$$\begin{aligned}
 d_x^{n_i} &= \frac{(n_{ix} - p_x)}{W}, \quad i = 1, 2. \\
 d_y^{n_i} &= \frac{(n_{iy} - p_y)}{H}, \quad i = 1, 2.
 \end{aligned}
 \tag{4.1}$$

Onde n_{ix} e n_{iy} são as coordenadas do nó i e H e W são as dimensões da imagem. Esse processo está ilustrado para um pixel na Figura 4.2.

Esse processo é realizado para todos os pixels do vaso, e um valor espúrio é dado ao restante dos pixels associados ao fundo da imagem. Os valores de deslocamento definem

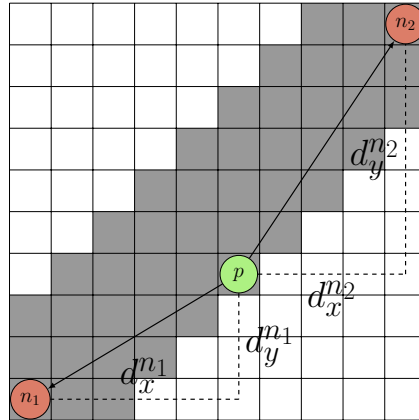


Figura 4.2 – Dado um segmento definido pelos nós n_1 e n_2 , calculamos os deslocamentos entre os pixels desse segmento e os nós.

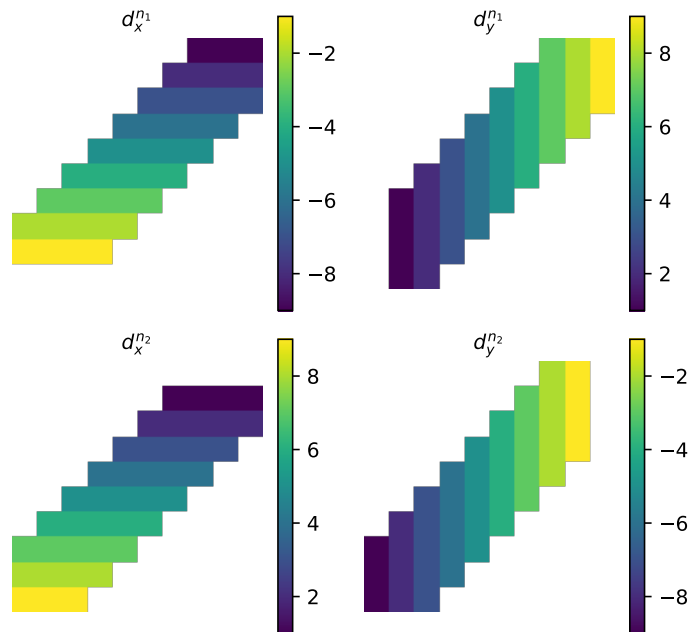


Figura 4.3 – Valores de deslocamento calculados para a imagem de exemplo.

quatro imagens, uma para cada coordenada, vide Figura 4.3.

Essas imagens podem ser aprendidas por uma CNN e a partir delas podemos recuperar os nós correspondentes a cada pixel pelas equações:

$$\begin{aligned} n_{ix} &= W d_x^{n_i} + p_x, \quad i = 1, 2. \\ n_{iy} &= H d_y^{n_i} + p_y, \quad i = 1, 2. \end{aligned} \quad (4.2)$$

A priori, é esperado que adicionar uma segunda tarefa que codifique a posição de pontos chave da estrutura dos vasos faça com que a rede aprenda representações que relacionem

os pixels de forma menos local, como naturalmente convoluções fazem, ajudando na tarefa principal de segmentação.

4.4 Parametrização por Deslocamentos ao Grafo Geométrico

O cálculo dos deslocamentos no processo descrito na seção anterior se baseia no grafo cujos nós são terminações e bifurcações. É observado que os valores obtidos pela Equação 4.1 são muito pequenos em algumas regiões do vaso. Adicionalmente, a dependência de longo alcance entre regiões distantes da imagem pode prejudicar o processo de treinamento da rede. Assim, propomos calcular os deslocamentos com relação ao grafo geométrico, da Definição 3. Nessa parametrização, modificamos a Equação 4.1 para:

$$\begin{aligned} d_x^{n_i} &= \frac{(n_{ix} - p_x)}{r}, \quad i = 1, 2. \\ d_y^{n_i} &= \frac{(n_{iy} - p_y)}{r}, \quad i = 1, 2. \end{aligned} \tag{4.3}$$

Os nós então podem ser recuperados pela equações:

$$\begin{aligned} n_{ix} &= r d_x^{n_i} + p_x, \quad i = 1, 2. \\ n_{iy} &= r d_y^{n_i} + p_y, \quad i = 1, 2. \end{aligned} \tag{4.4}$$

Nessa parametrização, a escala dos deslocamentos é similar para todos os segmentos de vaso.

Com o desenvolvimento dessas tarefas e a subsequente dificuldade em obter bons resultados durante o treinamento, a própria expressividade da rede surge como fator na modelagem da tarefa, isto é, ao invés de simplesmente criar uma tarefa que seja conveniente no pós-processamento, precisamos levar em conta quais tarefas uma rede convolucional pode realizar. Nesse sentido, desenvolvemos uma nova tarefa que permite tratar a parametrização de forma semelhante a um problema de classificação.

4.5 Parametrização por pontos âncora: SkelAnchor

Para lidar com dependências de longo alcance e garantir suavidade, uma parametrização adicional chamada SkelAnchor é proposta. Esta abordagem define um sistema de referência local para cada pixel do vaso, baseado em pontos âncora distribuídos em uma região anular.

4.5.1 Pontos Âncora

Dado um pixel p_r do vaso, definimos uma região anular com centro em p_r como:

$$A_{p_r} = \{q \in I \mid r \leq d(p_r, q) \leq r + \delta r\}, \quad (4.5)$$

onde r é o raio interno e δr é a largura da região anular. Em seguida, o anel é dividido em n setores de $\Delta\theta = \frac{2\pi}{n}$ radianos. Para cada setor s_i , atribuímos um ponto âncora a_i como o ponto médio do setor. Essa construção é ilustrada na Figura 4.4.

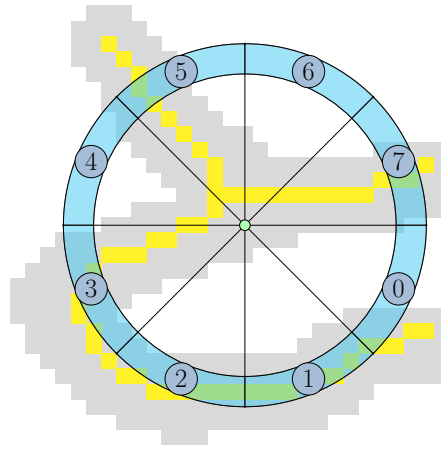


Figura 4.4 – Ilustração da região anular e dos pontos âncora.

Podemos definir uma posição angular normalizada ao longo do anel como:

$$\bar{\theta} = \frac{\theta}{\Delta\theta} \quad (4.6)$$

onde θ é a posição angular de um ponto no anel. Assim, cada ponto âncora tem uma coordenada angular normalizada $\alpha_i = i + 0.5$, com $i \in \{0, 1, \dots, n - 1\}$. Cada ponto $q \in A_p$ com posição angular $\bar{\theta}$ pode ser associado aos dois pontos âncora mais próximos, a_i e a_{i+1} . A posição do ponto âncora a_i é tal que:

$$q \in a_i \iff \lfloor \bar{\theta} - 0.5 \rfloor = i, \quad (4.7)$$

Os pontos âncora definem um referencial local para cada pixel do vaso e podem ser usados para o cálculo de deslocamentos, abordando o problema de dependências de longo alcance. Para resolver o problema da suavidade, definimos uma parametrização apropriada baseada na interpolação de valores intermediários aos pontos âncora.

4.5.2 Interpolação Linear

Como demonstrado em (THÉVENAZ; BLU; UNSER, 2002), a interpolação pode ser definida como uma convolução de coeficientes c_k com uma função de base $h(x)$. Dado um conjunto finito de pontos $P = \{(x_k, y_k), k \in 1, 2, \dots, K\}$, uma função contínua $f(x)$ tal que $f(x_k) = y_k$ é definida como:

$$f(x) = \sum_{k \in P} c_k h(x - x_k), \quad (4.8)$$

Para a interpolação linear, definindo $c_k = f(x_k)$ e a função de base (kernel) como:

$$h(x) = \begin{cases} 1 - |x|, & |x| \leq 1 \\ 0, & |x| > 1 \end{cases} \quad (4.9)$$

e considerando um conjunto de dois pontos $P = \{(x_0, y_0), (x_1, y_1)\}$ com $x_1 - x_0 = 1$, para $x_0 \leq x \leq x_1$, temos:

$$\begin{aligned} f(x) &= y_0(1 - |x - x_0|) + y_1(1 - |x - x_1|) \\ &= y_0(1 - (x - x_0)) + y_1(1 - (x_1 - x)). \end{aligned} \quad (4.10)$$

4.5.3 Cálculo do SkelAnchor

A parametrização é calculada interpolando a posição angular $\bar{\theta}_s$ de um pixel do esqueleto p_s entre os pontos âncora a_i e a_{i+1} . O uso do esqueleto permite ao mesmo tempo reduzir o custo computacional do método, sem comprometer o objetivo original de recuperar a topologia original do vaso.

Definimos a função interpolada $w_i : s_i \rightarrow \mathbb{R}$ com amostras $w_i(\alpha_i) = 1$ e $w_i(\alpha_{i+1}) = 0$. Usando a Equação 4.10:

$$\begin{aligned} w_i(\bar{\theta}_s) &= 1(1 - (\bar{\theta}_s - \alpha_i)) + 0(1 - (\alpha_{i+1} - \bar{\theta}_s)) \\ &= 1 - (\bar{\theta}_s - \alpha_i) \end{aligned} \quad (4.11)$$

O valor w_i representa um peso atribuído ao setor s_i , que está diretamente relacionado à distância entre p_s e o ponto âncora a_i . Um peso similar pode ser definido para o ponto âncora a_{i+1} como:

$$w_{i+1}(\bar{\theta}_s) = 1 - w_i(\bar{\theta}_s) \quad (4.12)$$

Uma ilustração do comportamento de w_i e w_{i+1} pode ser vista na Figura 4.5. Se um ponto âncora a_i estiver associado a múltiplos pixels do esqueleto, apenas o valor máximo de w_i calculado entre os pixels é mantido.

Como nosso objetivo é melhorar a conectividade da segmentação, devemos calcular os pesos considerando a topologia correta do vaso. Tome, por exemplo, o caso simplificado na Figura 4.4. Os pontos do esqueleto no setor 1 não devem ser considerados no cálculo,

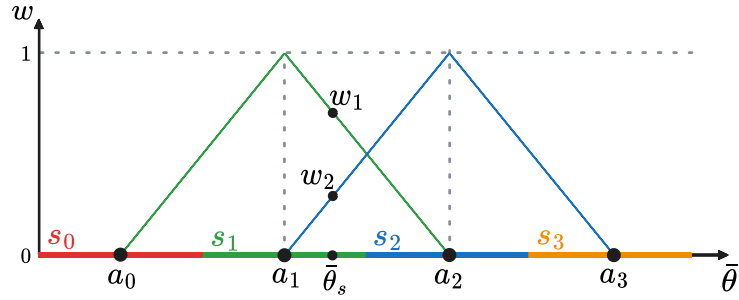


Figura 4.5 – Ilustração do cálculo dos pesos dos setores. Cada ponto $\bar{\theta}_s$ gera pesos correspondentes em dois setores. O valor do peso depende da distância aos respectivos pontos âncora.

pois não estão próximos ao ponto de referência de acordo com a distância do comprimento do caminho ao longo do vaso. Uma situação ainda pior seria ter um segmento de vaso não relacionado, possivelmente desconectado, passando pelo setor.

Para identificar pontos do esqueleto que não estão diretamente relacionados aos pontos de referência do vaso, considera-se a razão entre a distância Euclidiana e a distância geodésica entre os pontos. Distâncias geodésicas não são facilmente calculadas por uma fórmula fechada neste caso, mas é possível obter uma aproximação através da construção de um grafo de vizinhos mais próximos (PREPARATA; SHAMOS, 2012) G dos pixels do vaso. A distância geodésica entre os pixels p e q é o custo do caminho mínimo entre p e q em G . O caminho mínimo pode ser obtido pelo algoritmo de Dijkstra (DIJKSTRA, 2022).

Suponha que p_r e p_s possuam distância euclidiana $d_e(p_r, p_s)$ e distância geodésica $d_g(p_r, p_s)$. Se a razão

$$l = \frac{d_g(p_r, p_s)}{d_e(p_r, p_s)} \quad (4.13)$$

for muito maior que 1, o segmento do vaso entre os pontos é uma curva. Portanto, apenas pontos do esqueleto p_s que possuem $l < t_g$ são incluídos no cálculo. Os pontos restantes são ignorados. O limiar t_g é um parâmetro do método.

O Algoritmo 3 descreve o processo de cálculo dos valores de SkelAnchor.

Algorithm 3 Cálculo do SkelAnchor**Entrada:** $I_s, r, \delta r, n, t_g$ **Saída** : T // Tensor contendo os valores SkelAnchor

```

21 para cada  $p_r$  em  $\mathcal{V}$  faça
22    $S \leftarrow \text{pontos\_esqueleto\_no\_anel}(p_r, r, \delta r)$ 
23   para cada  $p_s$  em  $S$  faça
24      $l \leftarrow d_g(p_r, p_s)/d_e(p_r, p_s)$ 
25     se  $l \leq t_g$  então
26        $\bar{\theta}_s \leftarrow \text{angulo}(p_r, p_s)n/2\pi$ 
27        $i \leftarrow \text{piso}(\bar{\theta} - 0.5)$ 
28        $\alpha \leftarrow i + 0.5$ 
29        $w_i \leftarrow 1 - (\bar{\theta}_s - \alpha)$ 
30        $w_{i+1} \leftarrow 1 - w_i$ 
31        $T[i, p_{rx}, p_{ry}] \leftarrow w_i$ 
32        $T[i + 1, p_{rx}, p_{ry}] \leftarrow w_{i+1}$ 
33     fim
34   fim
35 fim
36 retorna  $T$ 

```

Além de garantir a suavidade dos valores ao longo do vaso, o uso da interpolação permite também localizar a posição do pixel de esqueleto no setor. Essa é uma característica chave no pós-processamento, permitindo recuperar os eixos mediais a partir da parametrização.

4.5.4 Recuperando os Eixos Mediais

Para usar o SkelAnchor no pós-processamento, devemos ser capazes de recuperar os pontos de esqueleto originais a partir da imagem final de n canais, ou seja, devemos encontrar uma inversa para w_i . Apesar de $w_i(x)$ não ser estritamente invertível, podemos reverter os valores com algumas suposições. Dado um pixel $p = (p_{rx}, p_{ry})$, obtemos os valores SkelAnchor $W_p = (w_0, w_1, \dots, w_{n-1})$ para p e aplicamos um limiar l_{skel} , mantendo apenas os valores de W com $w_i \geq l_{skel}$. Para cada valor w_i acima do limiar, temos dois valores possíveis para $\bar{\theta}_s$:

$$\begin{aligned}\bar{\theta}_s^+ &= -(1 - w_i) + 0.5 \\ \bar{\theta}_s^- &= (1 - w_i) + 0.5\end{aligned}\tag{4.14}$$

Para escolher o valor correto, podemos olhar os valores dos setores vizinhos, w_{i-1} e w_{i+1} . Assumindo que o $\bar{\theta}_s$ original está no mesmo lado do maior valor vizinho, se $w_{i+1} \geq w_{i-1}$, então $\bar{\theta}_s = \bar{\theta}_s^+$, caso contrário $\bar{\theta}_s = \bar{\theta}_s^-$. Com $\bar{\theta}_s$, podemos calcular o ângulo original:

$$\theta = (i + \bar{\theta}_s)\Delta\theta\tag{4.15}$$

Finalmente, o ponto do esqueleto original pode ser obtido por:

$$p_s = \left(p_x + \frac{r + \delta r}{2} \sin \theta, p_y + \frac{r + \delta r}{2} \cos \theta \right) \quad (4.16)$$

4.6 Arquitetura da rede e função de perda

Para aprender tanto a segmentação quanto as parametrizações, utilizamos uma arquitetura semelhante à U-Net, consistindo de um codificador compartilhado e dois decodificadores separados, um para cada tarefa.

Como mostrado na Figura 4.6, o codificador e os decodificadores são compostos por estágios, cada estágio composto por uma combinação de blocos básicos. O primeiro bloco é um bloco convolucional, consistindo de uma convolução, seguida por *batch norm* e *ReLU*.

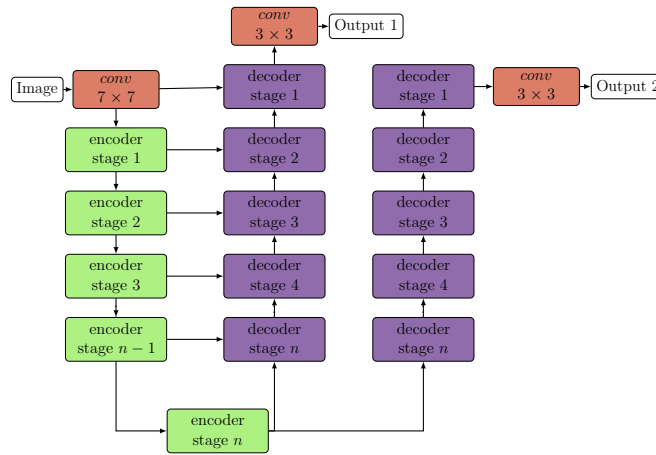


Figura 4.6 – A arquitetura U-Net com um segundo decodificador para a tarefa auxiliar.

Tanto para os estágios do codificador quanto do decodificador, usamos conexões residuais em combinação com blocos convolucionais, como ilustrado na Figura 2.4.

Construímos os estágios do codificador e do decodificador combinando esses blocos básicos. Nas Figuras 4.7 e 4.8, os dois blocos são mostrados. O bloco codificador é uma série de blocos convolucionais que reduzem a resolução da imagem (*downsample*), criando mais canais no processo.

Os estágios do decodificador recebem a imagem com resolução reduzida e gradualmente a retornam à sua forma original, no caso da tarefa de segmentação, e a um tensor $n \times H \times W$

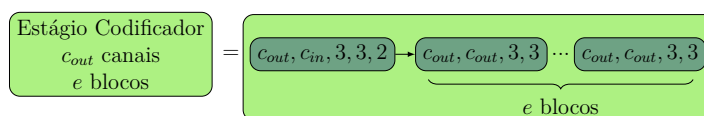


Figura 4.7 – Os estágios do codificador reduzem a resolução da imagem usando blocos convolucionais.

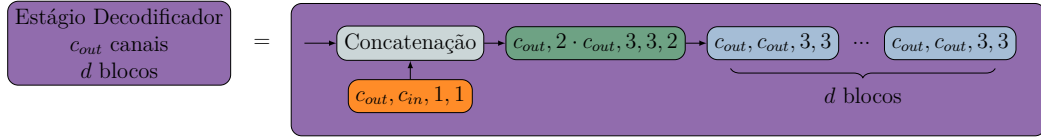


Figura 4.8 – Os estágios do decodificador aumentam a resolução da imagem (*upsample*) usando blocos convolucionais e concatenam as saídas dos estágios do codificador para as conexões de salto.

na tarefa SkelAnchor. Embora mantenhamos as conexões de salto (*skip connections*) da U-Net original na tarefa de segmentação, nossos testes mostram que o decodificador SkelAnchor obtém melhores resultados ao não utilizar este método.

4.6.1 Função de perda

Para realizar o treinamento desses modelos, é preciso definir funções de perda multitasrefas conforme a Equação 2.1. A tarefa de segmentação é comum a todos os modelos e usa a entropia cruzada, assim como o SkelAnchor, pois os valores estão no intervalo $[0, 1]$. Seja y um rótulo e \hat{y} a saída da rede, a perda é dada por:

$$L_{seg}(y, \hat{y}) = - \sum_i y_i \log \hat{y}_i \quad (4.17)$$

As tarefas secundárias relacionadas às parametrizações por grafo são tarefas de regressão, portanto, propomos utilizar a função de perda de Huber (HUBER, 1964):

$$L_d(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{para } |y - f(x)| \leq \delta \\ \delta \left(|y - f(x)| - \frac{1}{2}\delta \right) & \text{caso contrário} \end{cases} \quad (4.18)$$

onde y é o valor real, $f(x)$ é a predição do modelo, e δ é o limiar que define a transição.

$$L(y, \hat{y}) = \lambda L_{seg}(y, \hat{y}) + (1 - \lambda)L_d(y, \hat{y}), \quad (4.19)$$

onde $\lambda \in [0, 1]$ é um termo que ajusta a relevância de cada função de perda.

Capítulo 5

Resultados e Discussão

5.1 Parametrizações por Deslocamentos

De acordo com as parametrizações por deslocamentos propostas, podemos construir os conjuntos de dados para treinamento das redes. Apresentamos aqui exemplos para as três parametrizações definidas.

A parametrização pelo grafo topológico é a proposta inicial de onde derivam as demais parametrizações. Um exemplo de deslocamentos calculados é mostrado na Figura 5.1. É esperado que uma tarefa auxiliar que aprenda essa parametrização seja capaz de utilizar a relação entre pixels de um segmento e os nós para criar representações que levem à segmentações mais acuradas topologicamente. Também será possível utilizar o cálculo dos nós a partir da parametrização (Equação 4.2) como uma forma de pós-processamento para corrigir falhas na segmentação resultante da rede. É observado que essa parametrização tende a criar valores com grande variabilidade e próximos a zero em algumas regiões, o que pode trazer instabilidades numéricas durante o treinamento da rede. Com esse problema em mente, definimos a segunda parametrização a partir do grafo geométrico, restringindo os deslocamentos a regiões menores. Como por ser visto na Figura 5.2, essa parametrização de fato consegue descrever os vasos com valores mais regulares em relação à anterior, permitindo também recuperar os nós posteriormente pela Equação 4.4. Por outro lado, ainda é possível observar valores descontínuos próximos aos nós.

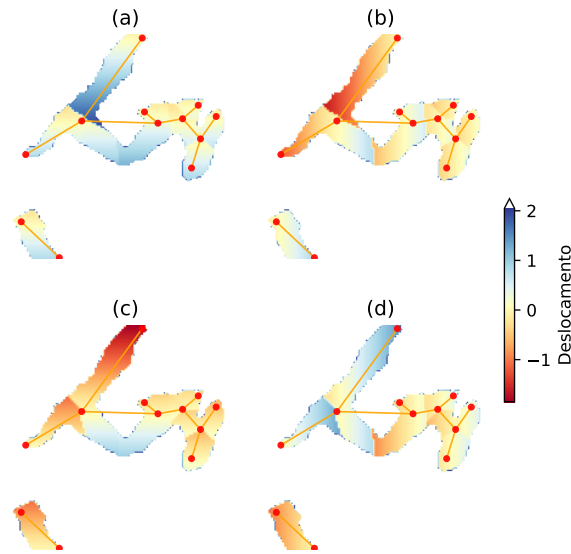


Figura 5.1 – Deslocamentos ao grafo topológico. Cada imagem representa uma das quantidades da Equação 4.2, isto é, a diferença das coordenadas de um pixel às coordenadas dos dois nós do grafo associados ao mesmo segmento de vaso. Nesse caso, (a) = d_x^{n1} , (b) = d_y^{n1} , (c) = d_x^{n2} e (d) = d_y^{n2} .

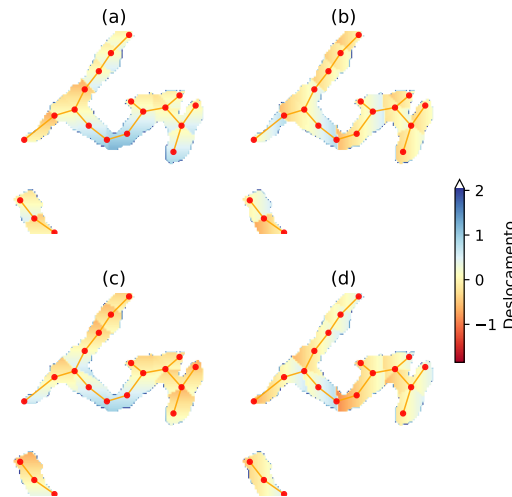


Figura 5.2 – Deslocamentos ao grafo geométrico. Cada imagem representa uma das quantidades da Equação 4.3, isto é, a diferença das coordenadas de um pixel às de um nó do grafo. Nesse caso, (a) = d_x^{n1} , (b) = d_y^{n1} , (c) = d_x^{n2} e (d) = d_y^{n2} .

Em contraste com as abordagens anteriores, o SkelAnchor produz uma parametrização suave dos vasos, superando os problemas de descontinuidade mencionados. Conforme ilustrado na Figura 5.3, essa abordagem gera valores suaves ao longo de todos os canais, o que facilita o treinamento da rede.

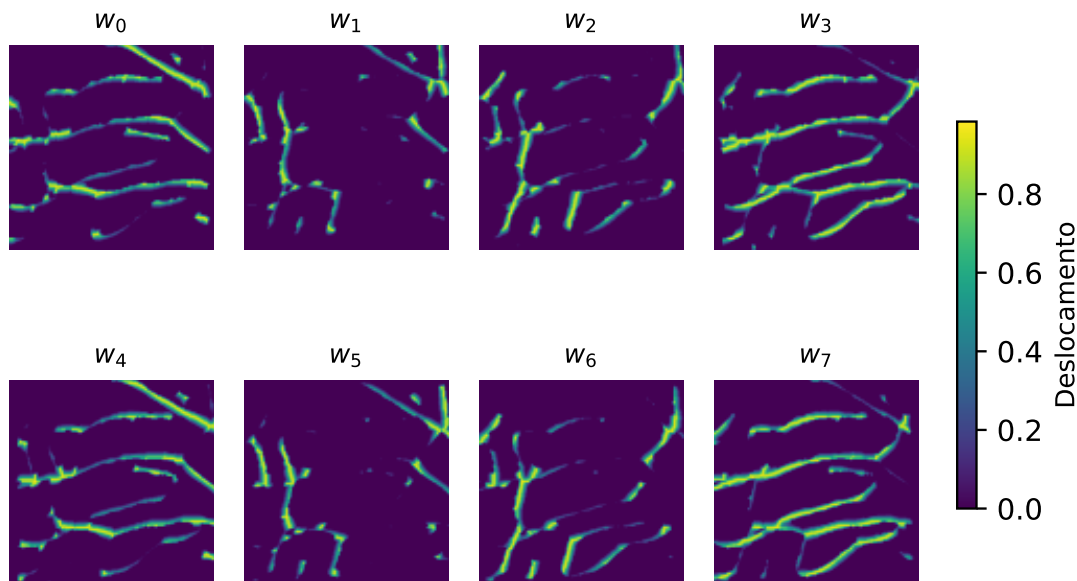


Figura 5.3 – Exemplo dos pesos do SkelAnchor para $n = 8$ setores.

5.2 Treinando a rede para localizar o esqueleto

Implementamos o modelo descrito na seção 4.6 utilizando a biblioteca Pytorch (PASZKE et al., 2019). O modelo foi treinado separadamente para localizar os pontos do esqueleto usando o grafo topológico, os grafos geométricos e o método SkelAnchor. Em todos os casos, o modelo foi treinado por 1000 épocas usando o otimizador Adam (KINGMA; BA, 2014) com um decaimento de peso de 0.0 e um tamanho de lote de 4. Observamos que $\lambda = 0.5$ na Equação 4.19 forneceu os melhores resultados. Em todos os treinamento foram empregados métodos de aumento de dados (escala, rotação e translação), resultando num conjunto final de 100000 imagens. O conjunto de dados VessMAP foi dividido em 80000 imagens para treinamento e 20 para teste.

Após diversos experimentos e buscas de hiperparâmetros do modelo, o modelo não foi capaz de aprender os deslocamentos corretamente usando as parametrizações de grafo topológico e geométrico. As Figuras 5.4 e 5.5 mostram exemplos de resultados para, respectivamente, as parametrizações topológica e geométrica. Observou-se que o modelo tem dificuldade em reproduzir as transições abruptas presentes nos mapas de referência.

Também tentamos variações do modelo, como incluir conexões residuais no decodificador de localização, usar o mesmo decodificador para ambas as tarefas e adicionar camadas de atenção ao modelo. Foram testadas também funções de perda alternativas. Os melhores resultados foram obtidos com a configuração descrita acima.

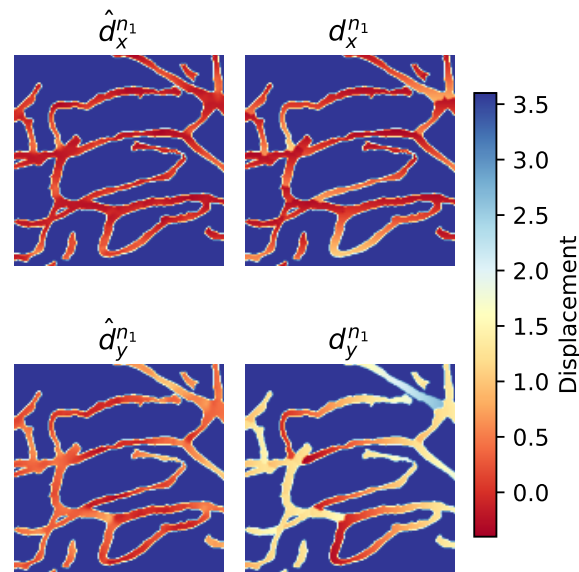


Figura 5.4 – Exemplo de valores de deslocamento obtidos para o grafo topológico. As figuras à esquerda mostram o resultado da rede, enquanto que as figuras à direita mostram os valores de referência $d_x^{n_1}$ e $d_y^{n_1}$.

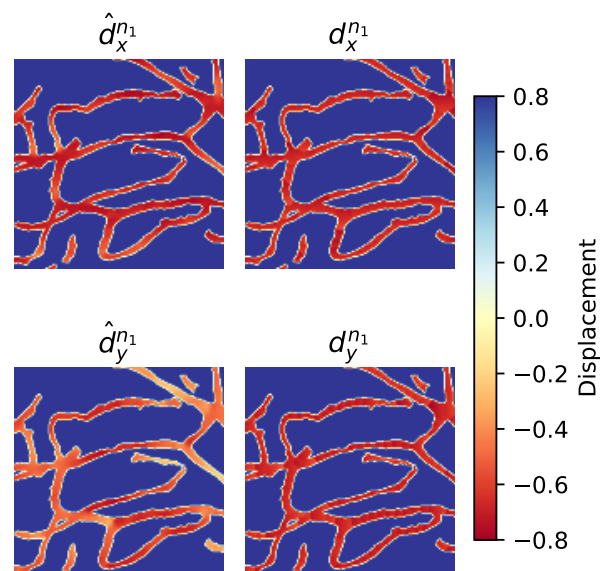


Figura 5.5 – Exemplo de valores de deslocamento obtidos para o grafo geométrico. As figuras à esquerda mostram o resultado da rede, enquanto que as figuras à direita mostram os valores de referência $d_x^{n_1}$ e $d_y^{n_1}$.

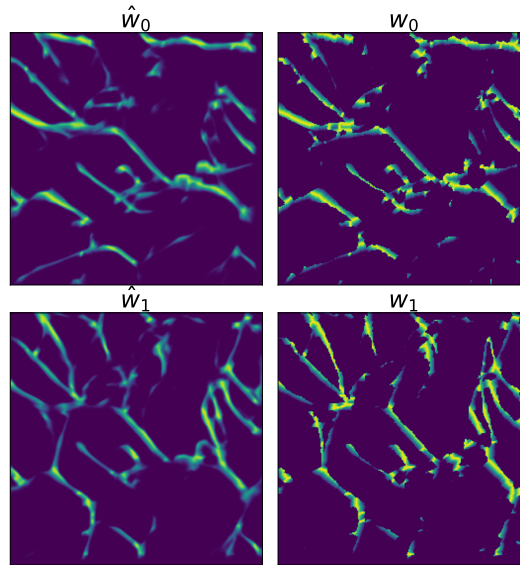


Figura 5.6 – Pesos SkelAnchor calculados para uma amostra. \hat{w}_0 e \hat{w}_1 indicam os resultados da rede, enquanto que os valores de referência são mostrados à esquerda.

Em relação à tarefa SkelAnchor, o mesmo modelo foi capaz de aprender os pesos com sucesso, como mostrado na Figura 5.6.

5.3 Pós-processamento

Para avaliar o impacto da tarefa auxiliar no processo de segmentação, comparamos nosso modelo SkelAnchor completo com um modelo de referência. O modelo de referência é uma arquitetura U-Net idêntica, treinada apenas na tarefa de segmentação, sem o segundo decodificador e o componente de perda SkelAnchor.

A Figura 5.7 mostra as curvas de precisão-revocação para o modelo de referência e para o nosso modelo SkelAnchor (após o pós-processamento) para diferentes parâmetros de pós-processamento. As curvas são, em geral, similares, indicando que em diversos casos a inclusão da tarefa auxiliar SkelAnchor não degrada o desempenho do modelo na tarefa de segmentação principal. Aqui, usamos a precisão e a revocação topológicas conforme definido na Seção 2.4.

O principal benefício do nosso método é percebido na etapa de pós-processamento. A Figura 5.8 fornece um exemplo qualitativo do algoritmo de pós-processamento SkelAnchor em ação. Selecionamos uma região desafiadora de algumas amostras de validação onde o modelo de referência falha em produzir uma segmentação topologicamente correta e aplicamos o método de pós-processamento descrito na Seção 4.5.4. Embora não tenha sido observada uma melhora sistemática nas métricas de qualidade globais, foram identificados diversos casos em que o método proposto aprimorou a topologia da segmentação inicial.

Nos exemplos mostrados nas Figuras 5.8(a) e (b), falhas ao centro do vaso foram

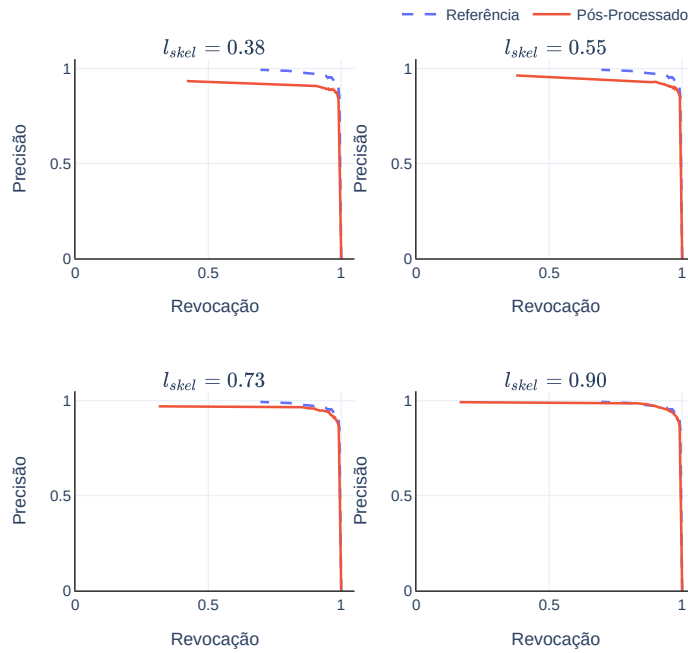


Figura 5.7 – Curvas de precisão-revocação para diferentes valores do limiar de pesos l_{skel} .

propriamente corrigidas, aprimorando a topologia da segmentação inicial. Também é possível encontrar exemplos como mostrados nas Figuras 5.8(c), (d) e (e), onde o pós-processamento consegue reconectar regiões antes desconexas. No entanto, uma limitação do método atual é não tratar os falsos positivos da segmentação inicial, o que impede que o resultado final recupere a topologia exata do padrão-ouro.

Ao aumentar o limiar de segmentação e pós-processar a imagem aproveitando as localizações do esqueleto previstas pela tarefa SkelAnchor, nosso algoritmo de pós-processamento identifica com sucesso os componentes desconectados e restaura a continuidade do vaso. Esta evidência visual serve como uma prova de conceito, demonstrando que a informação topológica aprendida através da tarefa SkelAnchor pode ser usada de forma eficaz para corrigir erros de conectividade na saída de segmentação final. Por outro lado, em diversas situações, o método cria artefatos através da geração de falsos positivos após o pós-processamento. Dessa forma, o método se mostra mais interessante em contextos onde a redução de falsos positivos é mais relevante para a preservação da topologia do vaso.

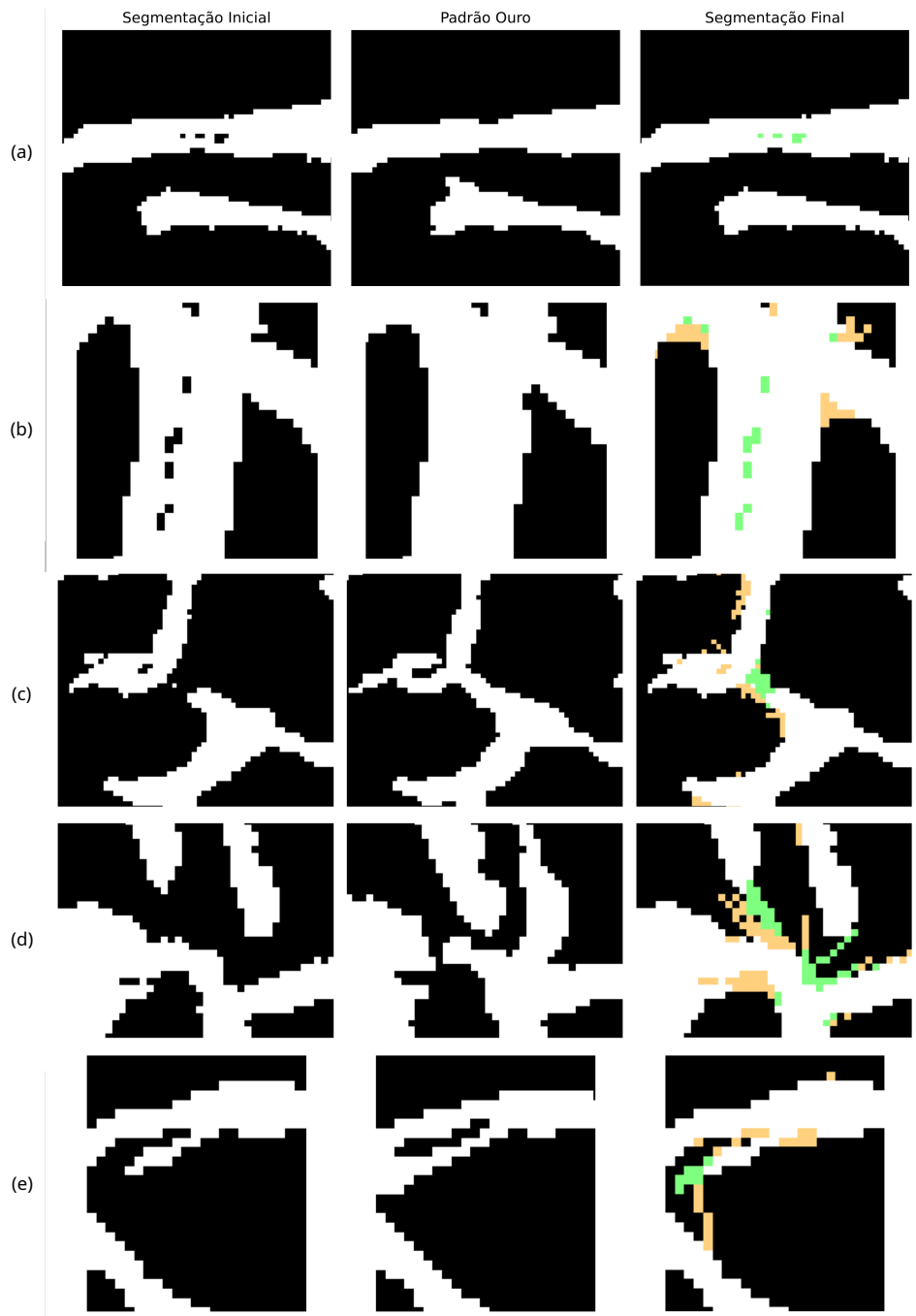


Figura 5.8 – Exemplos de pós-processamentos onde foi possível aprimorar a topologia da segmentação inicial. Em verde, pixels recuperados corretamente e em amarelo pontos adicionados erroneamente no pós-processamento.

Capítulo 6

Considerações Finais

Este trabalho abordou o desafio persistente da preservação da topologia na segmentação de vasos sanguíneos, um problema relevante para análises morfométricas subsequentes. As redes neurais convolucionais, apesar de seu alto desempenho em tarefas de segmentação, frequentemente produzem descontinuidades que comprometem a representação da vasculatura. O objetivo principal desta dissertação foi, portanto, desenvolver e avaliar tarefas auxiliares capazes de fornecer informação topológica à rede, visando aprimorar a conectividade dos vasos segmentados.

A investigação iniciou-se com a exploração de parametrizações baseadas em deslocamentos diretos a pontos chave da imagem. Contudo, os resultados demonstraram que tais abordagens geram mapas de valores descontínuos e com transições abruptas, que se mostraram excessivamente desafiadores para o aprendizado por parte da rede neural. Essas dificuldades motivaram o desenvolvimento do SkelAnchor, uma nova parametrização que codifica a localização do esqueleto vascular de forma suave e contínua, utilizando um referencial local baseado em pontos âncora.

Os experimentos demonstraram que, ao contrário das abordagens de deslocamento, a tarefa SkelAnchor pôde ser aprendida com sucesso pelo modelo. Embora a inclusão desta tarefa auxiliar não tenha resultado em uma melhoria sistemática nas métricas de segmentação padrão, como o *clDice*, sua principal contribuição manifestou-se na etapa de pós-processamento. A análise qualitativa revelou múltiplos casos em que a informação topológica aprendida foi utilizada de forma eficaz para corrigir falhas de conectividade, como lacunas e buracos na segmentação inicial, servindo como uma robusta prova de conceito para a abordagem proposta.

A principal contribuição deste trabalho é, portanto, a demonstração de que uma tarefa auxiliar baseada em esqueleto, como o SkelAnchor, pode codificar informação topológica

relevante e capacitar um modelo para reparar erros de conectividade. Isso estabelece uma via promissora para aprimorar a qualidade topológica de segmentações sem a necessidade de modificar drasticamente arquiteturas de rede já consolidadas ou de desenvolver funções de perda complexas.

Apesar dos resultados promissores, este estudo possui limitações que abrem caminhos para trabalhos futuros. Primeiramente, o método de pós-processamento atual não aborda a remoção de falsos positivos, o que pode, em alguns casos, levar à conexão de estruturas que não deveriam estar conectadas. Futuramente, o algoritmo de pós-processamento poderia ser aprimorado para incluir um critério de filtragem de falsos positivos por meio dos pesos aprendidos pela rede.

Como direções futuras, sugere-se a investigação de formas de refinamento do pós-processamento usando o SkelAnchor, ou ainda o desenvolvimento de tarefas mais complexas com base nos critérios de suavidade estabelecidos neste trabalho. Além disso, as segmentação de outros tipos de estruturas poderia se beneficiar do método, como rachaduras, estrada e outras estruturas curvilíneas. Apesar de não aprimorar o resultado de segmentação, este trabalho valida o potencial das tarefas auxiliares baseadas em esqueleto e estabelece uma base sólida para futuros métodos para segmentações topologicamente acuradas.

Referências

- ARAÚJO, R.; CARDOSO, J.; OLIVEIRA, H. A deep learning design for improving topology coherence in blood vessel segmentation. In: _____. [S.l.: s.n.], 2019. p. 93–101. ISBN 978-3-030-32238-0.
- AZAD, R. et al. Medical image segmentation review: The success of u-net. **arXiv preprint arXiv:2211.14830**, 2022.
- BAXTER, J. A model of inductive bias learning. **Journal of artificial intelligence research**, v. 12, p. 149–198, 2000.
- BEN-DAVID, S.; SCHULLER, R. Exploiting task relatedness for multiple task learning. In: SPRINGER. **Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003. Proceedings**. [S.l.], 2003. p. 567–580.
- BORDELEAU, M. et al. Maternal high-fat diet in mice induces cerebrovascular, microglial and long-term behavioural alterations in offspring. **Communications Biology**, Nature Publishing Group UK London, v. 5, n. 1, p. 26, 2022.
- CARUANA, R. Multitask learning. **Machine learning**, Springer, v. 28, p. 41–75, 1997.
- CHEN, C. et al. Retinal vessel segmentation using deep learning: A review. **IEEE Access**, PP, p. 1–1, 08 2021.
- _____. Automatic x-ray landmark detection and shape segmentation via data-driven joint estimation of image displacements. **Medical image analysis**, Elsevier, v. 18, n. 3, p. 487–499, 2014.
- CHEN, L. et al. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 5659–5667.
- COMIN, C. H. et al. Quantification of retinal blood leakage in fundus fluorescein angiography in a retinal angiogenesis model. **Scientific Reports**, Springer Science and Business Media LLC, v. 11, n. 1, out. 2021. ISSN 2045-2322.
- CONZE, P.-H. et al. Current and emerging trends in medical image segmentation with deep learning. **IEEE Transactions on Radiation and Plasma Medical Sciences**, IEEE, v. 7, n. 6, p. 545–569, 2023.

- DAI, J. et al. Deformable convolutional networks. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2017. p. 764–773.
- DANGI, S.; LINTE, C. A.; YANIV, Z. A distance map regularized cnn for cardiac cine mr image segmentation. **Medical Physics**, Wiley, v. 46, n. 12, p. 5637–5651, out. 2019. ISSN 2473-4209.
- DEARI, S.; OKSUZ, I.; ULUKAYA, S. Block attention and switchable normalization based deep learning framework for segmentation of retinal vessels. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 11, p. 38263–38274, 2023. ISSN 2169-3536.
- DICE, L. R. Measures of the amount of ecologic association between species. **Ecology**, Wiley, v. 26, n. 3, p. 297–302, jul. 1945. ISSN 1939-9170.
- DIJKSTRA, E. W. A note on two problems in connexion with graphs. In: **Edsger Wybe Dijkstra: his life, work, and legacy**. [S.l.: s.n.], 2022. p. 287–290.
- DU, H. et al. Retinal blood vessel segmentation by using the ms-ldnet network and geometric skeleton reconnection method. **Computers in Biology and Medicine**, Elsevier, v. 153, p. 106416, 2023.
- DUBEY, S. R.; SINGH, S. K.; CHAUDHURI, B. B. Activation functions in deep learning: A comprehensive survey and benchmark. **Neurocomputing**, Elsevier, v. 503, p. 92–108, 2022.
- DULAU, I. et al. Ensuring a connected structure for retinal vessels deep-learning segmentation. In: **2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)**. [S.l.: s.n.], 2023. p. 2356–2365.
- _____. Ensuring a connected structure for retinal vessels deep-learning segmentation. In: **Proceedings of the IEEE/CVF International Conference on Computer Vision**. [S.l.: s.n.], 2023. p. 2364–2373.
- EDELSBRUNNER, H.; HARER, J. L. **Computational topology: an introduction**. [S.l.]: American Mathematical Society, 2022.
- FORMAN, R. A user’s guide to discrete morse theory. **Séminaire Lotharingien de Combinatoire [electronic only]**, Universität Wien, Fakultät für Mathematik, v. 48, p. B48c–35, 2002.
- FREITAS-ANDRADE, M. et al. Unbiased analysis of mouse brain endothelial networks from two- or three-dimensional fluorescence images. **Neurophotonics**, SPIE-Intl Soc Optical Eng, v. 9, n. 03, maio 2022. ISSN 2329-423X. Disponível em: <http://dx.doi.org/10.1117/1.NPh.9.3.031916>.
- _____. Unbiased analysis of mouse brain endothelial networks from two-or three-dimensional fluorescence images. **Neurophotonics**, Society of Photo-Optical Instrumentation Engineers, v. 9, n. 3, p. 031916–031916, 2022.
- FUKUSHIMA, K. Visual feature extraction by a multilayered network of analog threshold elements. **IEEE Transactions on Systems Science and Cybernetics**, Institute of Electrical and Electronics Engineers (IEEE), v. 5, n. 4, p. 322–333, 1969. ISSN 0536-1567.

GALDRAN, A. et al. State-of-the-art retinal vessel segmentation with minimalistic models. **Scientific Reports**, v. 12, p. 6174, 04 2022.

_____. State-of-the-art retinal vessel segmentation with minimalistic models. **Scientific Reports**, Nature Publishing Group UK London, v. 12, n. 1, p. 6174, 2022.

GRAHAM, S. et al. One model is all you need: multi-task learning enables simultaneous histology image segmentation and classification. **Medical Image Analysis**, Elsevier, v. 83, p. 102685, 2023.

GUO, C. et al. Channel attention residual u-net for retinal vessel segmentation. In: IEEE. **ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing (ICASSP)**. [S.l.], 2021. p. 1185–1189.

_____. Sa-unet: Spatial attention u-net for retinal vessel segmentation. In: IEEE. **2020 25th international conference on pattern recognition (ICPR)**. [S.l.], 2021. p. 1236–1242.

GUPTA, S. et al. Topology-aware uncertainty for image segmentation. **Advances in Neural Information Processing Systems**, v. 36, 2024.

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

HÖRST, F. et al. Cellvit: Vision transformers for precise cell segmentation and classification. **Medical Image Analysis**, Elsevier, v. 94, p. 103143, 2024.

HU, X. et al. Topology-preserving deep image segmentation. **Advances in neural information processing systems**, v. 32, 2019.

HUBER, P. J. Robust estimation of a location parameter. **The Annals of Mathematical Statistics**, Institute of Mathematical Statistics, v. 35, n. 1, p. 73–101, mar. 1964. ISSN 0003-4851.

HUSSAIN, S. et al. Dilunet: A u-net based architecture for blood vessels segmentation. **Computer Methods and Programs in Biomedicine**, Elsevier, v. 218, p. 106732, 2022.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: PMLR. **International conference on machine learning**. [S.l.], 2015. p. 448–456.

ISENSEE, F. et al. nnu-net revisited: A call for rigorous validation in 3d medical image segmentation. **arXiv preprint arXiv:2404.09556**, 2024.

JIANG, W. et al. Ori-net: Orientation-guided neural network for automated coronary arteries segmentation. **Expert Systems with Applications**, Elsevier BV, v. 238, p. 121905, mar. 2024. ISSN 0957-4174.

JIANG, Y. et al. Retinal vessels segmentation based on dilated multi-scale convolutional neural network. **IEEE Access**, v. 7, p. 76342–76352, 2019.

- KAWATA, Y.; NIKI, N.; KUMAZAKI, T. Measurement of blood vessel characteristics for disease detection based on cone-beam ct images. **IEEE Transactions on Nuclear Science**, v. 43, n. 6, p. 3348–3354, 1996.
- KERVADEC, H. et al. Beyond pixel-wise supervision: semantic segmentation with higher-order shape descriptors. In: **Medical Imaging with Deep Learning**. [S.l.: s.n.], 2021.
- KEUTH, R. et al. Denseseg: Joint learning for semantic segmentation and landmark detection using dense image-to-shape representation. **arXiv preprint arXiv:2405.19746**, 2024.
- KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. **CoRR**, abs/1412.6980, 2014.
- KORDON, F. et al. A principled representation of elongated structures using heatmaps. **Scientific Reports**, Springer Science and Business Media LLC, v. 13, n. 1, set. 2023. ISSN 2045-2322.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. **Advances in neural information processing systems**, v. 25, 2012.
- KUMAR, K. S.; SINGH, N. P. Analysis of retinal blood vessel segmentation techniques: a systematic survey. **Multimedia Tools and Applications**, Springer, v. 82, n. 5, p. 7679–7733, 2023.
- LI, H. et al. Visualizing the loss landscape of neural nets. **Advances in neural information processing systems**, v. 31, 2018.
- LIPTON, Z. C.; STEINHARDT, J. Troubling trends in machine learning scholarship: Some ml papers suffer from flaws that could mislead the public and stymie future research. **Queue**, ACM New York, NY, USA, v. 17, n. 1, p. 45–77, 2019.
- LIU, C.; ZHAO, T.; ZHENG, N. Deepbranchtracer: A generally-applicable approach to curvilinear structure reconstruction using multi-feature learning. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2024. v. 38, n. 4, p. 3548–3557.
- MAURER, A.; PONTIL, M.; ROMERA-PAREDES, B. The benefit of multitask representation learning. **Journal of Machine Learning Research**, v. 17, n. 81, p. 1–32, 2016.
- MOCCIA, S. et al. Blood vessel segmentation algorithms—review of methods, datasets and evaluation metrics. **Computer methods and programs in biomedicine**, Elsevier, v. 158, p. 71–91, 2018.
- MOU, L. et al. Dense dilated network with probability regularized walk for vessel detection. **IEEE transactions on medical imaging**, IEEE, v. 39, n. 5, p. 1392–1403, 2019.

- _____. Cs-net: Channel and spatial attention network for curvilinear structure segmentation. In: SHEN, D. et al. (Ed.). **Medical Image Computing and Computer Assisted Intervention – MICCAI 2019**. Cham: Springer International Publishing, 2019. p. 721–730. ISBN 978-3-030-32239-7.
- OUELLETTE, J. et al. Vascular contributions to 16p11.2 deletion autism syndrome modeled in mice. **Nature Neuroscience**, Springer Science and Business Media LLC, v. 23, n. 9, p. 1090–1101, jul. 2020. ISSN 1546-1726. Disponível em: <<http://dx.doi.org/10.1038/s41593-020-0663-1>>.
- PANCHAL, S.; KOKARE, M. Resmu-net: Residual multi-kernel u-net for blood vessel segmentation in retinal fundus images. **Biomedical Signal Processing and Control**, Elsevier, v. 90, p. 105859, 2024.
- PASZKE, A. et al. Pytorch: An imperative style, high-performance deep learning library. **Advances in neural information processing systems**, v. 32, 2019.
- PREPARATA, F. P.; SHAMOS, M. I. **Computational geometry: an introduction**. [S.l.]: Springer Science & Business Media, 2012.
- PRINCE, S. J. **Understanding Deep Learning**. [S.l.]: The MIT Press, 2023.
- QI, Y. et al. Dynamic snake convolution based on topological geometric constraints for tubular structure segmentation. In: **Proceedings of the IEEE/CVF International Conference on Computer Vision**. [S.l.: s.n.], 2023. p. 6070–6079.
- RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. **Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18**. [S.l.], 2015. p. 234–241.
- RUDER, S. An overview of multi-task learning in deep neural networks. **arXiv preprint arXiv:1706.05098**, 2017.
- SACELEANU, V. M. et al. Integrative approaches in acute ischemic stroke: From symptom recognition to future innovations. **Biomedicines**, v. 11, n. 10, 2023. ISSN 2227-9059. Disponível em: <<https://www.mdpi.com/2227-9059/11/10/2617>>.
- SANTURKAR, S. et al. How does batch normalization help optimization? **Advances in neural information processing systems**, v. 31, 2018.
- SHIT, S. et al. cldice-a novel topology-preserving loss function for tubular structure segmentation. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2021. p. 16560–16569.
- SIDDIQUE, N. et al. U-net and its variants for medical image segmentation: A review of theory and applications. **Ieee Access**, IEEE, v. 9, p. 82031–82057, 2021.
- SILVA, M. V. da et al. A new sampling methodology for defining heterogeneous subsets of samples for training image segmentation algorithms. **arXiv preprint arXiv:2301.04517**, 2023.

SØRENSEN, T. et al. A method of establishing group of equal amplitude in plant sociobiology based on similarity of species content and its application to analyses of the vegetation on danish commons. In: . [S.l.: s.n.], 1948.

THÉVENAZ, P.; BLU, T.; UNSER, M. Interpolation revisited [medical images application]. **IEEE Transactions on medical imaging**, IEEE, v. 19, n. 7, p. 739–758, 2002.

VANDENHENDE, S. et al. Multi-task learning for dense prediction tasks: A survey. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 44, n. 7, p. 3614–3633, 2021.

WANG, Y. et al. Deep distance transform for tubular structure segmentation in ct scans. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2020. p. 3833–3842.

XU, D. et al. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2018. p. 675–684.

YANG, G. et al. A mean field theory of batch normalization. **arXiv preprint arXiv:1902.08129**, 2019.

YANG, Z. et al. Classification of diabetic retinopathy: Past, present and future. **Frontiers in Endocrinology**, Frontiers Media SA, v. 13, dez. 2022. ISSN 1664-2392.

ZHANG, D.; LU, G. Review of shape representation and description techniques. **Pattern recognition**, Elsevier, v. 37, n. 1, p. 1–19, 2004.

ZHANG, T. Y.; SUEN, C. Y. A fast parallel algorithm for thinning digital patterns. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, v. 27, n. 3, p. 236–239, mar. 1984. ISSN 0001-0782. Disponível em: <https://doi.org/10.1145/357994.358023>.

ZHANG, Z. et al. Representation learning of geometric trees. In: **Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining**. [S.l.: s.n.], 2024. p. 4374–4385.