UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

COMPARAÇÃO DE MODELOS DE REDES NEURAIS NA SEGMENTAÇÃO DE VASOS SANGUÍNEOS EM IMAGENS MÉDICAS

IVAN DUARTE CALVO

ORIENTADOR: PROF. DR. CESAR HENRIQUE COMIN

São Carlos – SP

2024

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

COMPARAÇÃO DE MODELOS DE REDES NEURAIS NA SEGMENTAÇÃO DE VASOS SANGUÍNEOS EM IMAGENS MÉDICAS

IVAN DUARTE CALVO

Trabalho de Conclusão de Curso apresentado ao curso de Ciência de 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 Bacharel em Ciência de Computação.

Orientador: Prof. Dr. Cesar Henrique Comin

São Carlos – SP

2024

Dedico esse trabalho aos meus pais, pelo apoio incondicional que não só possibilitou toda a jornada, mas a fez agradável e proveitosa.

AGRADECIMENTOS

Agradeço aos meus pais, meu irmão e minha família, pelo possível e impossível feito para possibilitar toda essa jornada.

À minha namorada Clara Saciloto, que sempre me deu apoio, coragem, motivação e dividiu comigo uma parte muito importante do caminho.

Ao meu amigo de longa data Thales Flausino, pelos anos e anos de companheirismo e inspiração. Também aos meus amigos de muito tempo, Ariadne Silva, Bruno Ferrari, Iago Laguna, João Vitor Furlani e Vitor Aredes, por trilharem comigo o começo de tudo.

Ao meu amigo Alberto Seghetto, por todo o conhecimento dividido, pelo acolhimento e amizade compartilhada.

Às grandes amizades que encontrei nesse percurso, que não só tornaram tudo isso possível mas carregam grande parte da responsabilidade por qualquer sucesso alcançado: Ana Paula Chiari, Áquila Oliveira, Carlos Fontaneli, Ingrid Lira, Luís Simas, Matheus Mattioli, Vinícius Borges, Yan Köhler.

Finalmente, agradeço também ao meu orientador, Prof. Dr. Cesar Henrique Comin, por toda a disponibilidade, o auxílio prestado, e a enorme quantidade de conhecimento passado que, com certeza, foi fundamental durante todo o processo desse trabalho.

RESUMO

A análise de vasos sanguíneos é responsável pela extração de diversas informações importantes na área da saúde, e o impacto de análises mais precisas em estudos sobre doenças e diagnósticos possui potencial muito positivo. Entretanto, a realização dessas análises de maneira manual envolve uma grande utilização de tempo e recursos. A segmentação de vasos sanguíneos em imagens representa uma grande parte dessa dificuldade e custo. Avanços tecnológicos têm permitido a implementação de técnicas de aprendizado de máquina na realização dessa tarefa muito importante, principalmente com a utilização das redes neurais, o que tem representado uma grande evolução na área. Neste trabalho são analisados e comparados diversos modelos de redes neurais que têm recebido grande atenção recentemente, como a *ResNet* e *EfficientNet*, por exemplo. Essa análise e comparação visa a obtenção de informações importantes sobre as vantagens e desvantagens da aplicação de cada modelo investigado, como situações nas quais os modelos possuem menor acurácia e sobre a influência das etapas de pré-processamento e treinamento dos modelos. Também é investigado o impacto da variação do tamanho e número de parâmetros dos modelos, visto que os resultados apresentados por modelos menores podem ser muito satisfatórios ao mesmo tempo que consomem menos recursos computacionais e muito menos tempo. Os testes realizados indicaram bons resultados dos modelos implementados com a arquitetura *U-Net*, especialmente nos experimentos envolvendo a combinação da *U-Net* com os modelos *RegNet* e *DenseNet*. Também se destacaram os resultados obtidos pelos modelos menores, como *RegNetY_002*, por exemplo, alcançando em várias ocasiões uma qualidade de segmentação similar a modelos com uma quantidade de parâmetros cerca de cinco vezes maior.

Palavras-chave: segmentação, vasos sanguíneos, redes neurais, convolução, transformers, aprendizado profundo

ABSTRACT

The analysis of blood vessels is responsible for extracting various important pieces of information in the healthcare field, and the impact of more precise analyses on disease studies and diagnoses holds positive potential. However, performing these analyses manually involves significant time and resource consumption. The segmentation of blood vessels in images represents a large part of this difficulty and cost. Technological advancements have enabled the implementation of machine learning techniques to carry out this crucial task, particularly through the use of neural networks, which has marked a significant evolution in the field. This work analyzes and compares various neural network models that have recently garnered significant attention, such as ResNet and EfficientNet, for example. The goal of this analysis and comparison is to obtain important insights into the advantages and disadvantages of applying each investigated model, such as situations where the models have lower accuracy and the influence of preprocessing and training stages on the models. The impact of varying the size and number of parameters in the models is also investigated, as smaller models may deliver very satisfactory results while consuming fewer computational resources and much less time. Finally, the tests indicated a good result with the models implemented with *U-Net*, especially *RegNet* and *DenseNet*. Also noteworthy were the results obtained by smaller models, like *RegNetY_002*, for instance, which in several cases achieved similar results in *IoU* and *clDice* as models with approximately five times more parameters.

Keywords: segmentation, blood vessels, neural networks, convolution, transformers, deep learning

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

CNN	Rede Neural Convolucional
SVM	Máquina de Vetores de Suporte
IoU	Intersecção sobre União
FCN	Rede Totalmente Convolucional
GPU	Unidade de Processamento Gráfico
NLP	Processamento de Linguagem Natural

SUMÁRIO

Capítulo 1
INTRODUÇÃO

O crescimento em popularidade e os avanços tecnológicos nas áreas relacionadas ao aprendizado de máquina têm recebido grande destaque, especialmente nos últimos anos. Em (LECUN et al., 2015), é debatido como o aprendizado profundo possuía um enorme potencial futuro por conta da sua capacidade de realizar tarefas complexas exigindo uma quantidade muito baixa de interação e trabalho humano. Esse potencial previsto tem se tornado parte da realidade em uma velocidade crescente em diversas áreas de estudo e desenvolvimento.

Simultaneamente, há um grande investimento de recursos e de tempo na análise e estudo de vasos sanguíneos, visto o enorme potencial de incrementar a qualidade de diagnósticos e tratamentos. Naturalmente, a segmentação dos vasos em imagens médicas se torna uma parte essencial do processo, incentivando o surgimento de uma enorme área de estudos na tentativa de tornar o processo cada vez mais automatizado (MOCCIA et al., 2018).

Dentre diversos algoritmos e possibilidades, destaca-se atualmente a utilização de tecnologias relacionadas com aprendizado de máquina e mais especificamente a utilização das redes neurais. Em (HAQUE; NEUBERT, 2020) são comparadas diversas implementações em tarefas relacionadas à segmentação semântica de imagens médicas, destacando-se o grande investimento em pesquisas relacionadas à área.

Por conta da possibilidade de adaptação das redes e de não necessitarem de um algoritmo específico para cada tarefa, o estudo da aplicação de redes neurais tem sido cada vez maior, possibilitando o surgimento de novos modelos mais eficientes. Em (SULTANA et al., 2020) é realizada uma análise muito completa em relação à grande evolução tecnológica pela qual a área do aprendizado profundo tem passado nos últimos anos, realizando inúmeros avanços em diversas tarefas.

Com a evolução de tecnologias, em *hardware* e *software*, responsáveis pela implementação e a escalada do processo de convolução, o processamento de imagens por redes neurais obteve um grande avanço e a capacidade das redes em detectar padrões e características têm sido fundamentais no processo de segmentação semântica. Tais desenvolvimentos fomentaram o surgimento de novos modelos e técnicas, alguns inclusive focados na utilização de imagens médicas,

que podem representar um salto importante no processo de estudos dos vasos sanguíneos.

A maior parte dos artigos citados que realizam comparações entre modelos de redes neurais inclui nas comparações modelos específicos, muitas vezes criados visando a tarefa de segmentação de imagens médicas. Esses modelos em geral possuem módulos customizados para as tarefas investigadas. Poucos trabalhos consideram a comparação de arquiteturas de redes padronizadas amplamente disponíveis em bibliotecas de fácil acesso. Dessa forma, é interessante realizar tais comparações para verificar se arquiteturas bem conhecidas, disponíveis em bibliotecas como o *PyTorch*, possuem boa capacidade de segmentação de vasos sanguíneos.

Adicionalmente, é válido também levar em consideração o tamanho dos modelos comparados. Como em (GALDRAN et al., 2022), onde são utilizados modelos muito menores em comparação com os comumente investigados na literatura. Essa comparação pode resultar em conclusões muito interessantes, especialmente no que tange ao custo computacional e de tempo.

1.1 Objetivos

Este trabalho possui o objetivo de realizar uma comparação abrangente de diversos modelos de redes neurais na tarefa de segmentação de vasos sanguíneos em imagens médicas. Devido à alta, e crescente, relevância das áreas relacionadas à aprendizado profundo nos últimos anos, foi decidido realizar uma comparação que aborde modelos com grande notoriedade.

Visto a existência de um elevado número de implementações diferentes de redes, é natural que modelos diferentes possuam vantagens e desvantagens em tarefas distintas. Logo, um grande objetivo a ser desenvolvido nesse trabalho é a busca dos modelos que, de acordo com as métricas e indicativos utilizados, obtenham os melhores resultados na tarefa da segmentação dos vasos sanguíneos.

Outro ponto importante é a grande diversidade nos tamanhos e complexidade dos modelos construídos. Portanto, é também de grande interesse o estudo e comparação desses modelos com grandes diferenças de tamanho, que possam significar uma discrepância de custo e eficiência, possibilitando a compreensão das vantagens ou desvantagens na utilização dos modelos mais custosos.

1.2 Organização do texto

O trabalho contém 6 capítulos, sendo o primeiro o da introdução. Adicionalmente são encontrados os seguintes:

- No capítulo 2 é realizada uma fundamentação teórica, abordando brevemente conceitos necessários para uma melhor compreensão dos temas presentes nos capítulos seguintes.

- No capítulo 3 é construída uma revisão da literatura, com citações de pesquisas relacionadas aos temas presentes e que auxiliaram na construção do trabalho.
- No capítulo 4 é descrita a metodologia presente no desenvolvimento das análises, comparações e conclusões obtidas a partir dos resultados.
- O capítulo 5 possui todos os resultados advindos dos testes realizados com as redes, incluindo a utilização de figuras e outros mecanismos visuais para uma melhor legibilidade e compreensão. Há também a discussão sobre a performance das redes comparadas.
- Por fim, no capítulo 6 está a conclusão desenvolvida a partir dos valores obtidos nas análises e comparações realizadas.

Capítulo 2
FUNDAMENTAÇÃO TEÓRICA

2.1 Segmentação semântica

A segmentação é o processo de divisão ou separação de um conjunto. A segmentação é caracterizada como semântica quando essa divisão é realizada de acordo com o sentido que cada parte tenha dentro do contexto presente nesse conjunto. Portanto, a segmentação semântica de uma imagem consiste em assimilar um rótulo para cada um dos *pixels* que compõem a imagem. Ou seja, a imagem passa a ser dividida em grupos, onde cada grupo representa um rótulo atribuído pela rede (HAO et al., 2020).

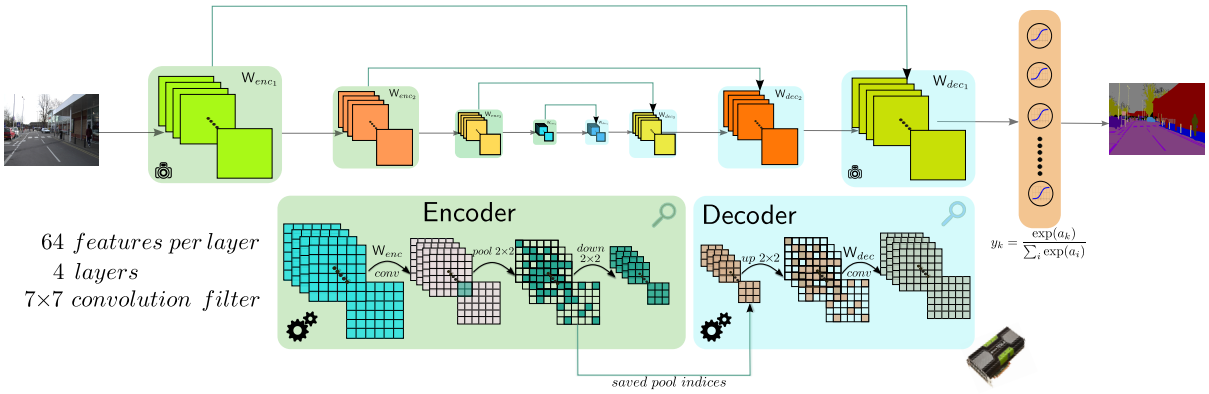
Por exemplo, no caso deste trabalho, cada *pixel* nas imagens utilizadas pode possuir dois valores como rótulo: fundo ou vaso sanguíneo. Dessa forma, a segmentação funciona como um processo de separação entre as possíveis classes que compõem uma imagem gerando no fim uma nova imagem contendo para cada *pixel* um dos valores possíveis a serem assumidos.

2.2 Aprendizado de Máquina e Aprendizado Profundo

A área de aprendizado de máquina tem suas contribuições mais antigas já na década de 50, onde programas de inteligência artificial eram escritos e calculados à mão (SHINDE; SHAH, 2018). Com os grandes avanços tecnológicos no decorrer do tempo as técnicas de aprendizado de máquina foram se tornando muito mais diversas, robustas e eficientes. Atualmente, são aplicadas em tarefas que, muitas vezes, são excessivamente complexas e trabalhosas para seres humanos (MAHESH, 2020).

A maior vantagem que os métodos de aprendizado profundo trazem em relação aos métodos anteriores de aprendizado de máquina tem relação com a capacidade de aprender padrões mais complexos dos dados ou tarefas propostas sem que seja necessário um trabalho manual muito grande para adaptar o modelo (LECUN et al., 2015).

Figura 1 – Exemplo visual do funcionamento da arquitetura codificador-decodificador. O codificador é responsável pela extração das características que são então utilizadas pelo decodificador para formar a imagem segmentada.



Fonte: (BADRINARAYANAN et al., 2015)

2.3 Redes Neurais

As redes neurais utilizadas nesse trabalho para a tarefa da segmentação semântica dos vasos sanguíneos foram construídas com a estrutura codificador-decodificador. Nessa estrutura, um codificador realiza a extração de características e diminuição da dimensionalidade enquanto o decodificador realiza um processo de aumento da amostragem resultando na construção de uma imagem segmentada.

Um exemplo desse tipo de arquitetura pode ser encontrado em (BADRINARAYANAN et al., 2015) e a figura 1 traz uma ilustração da arquitetura proposta. Na imagem pode ser visualizado como a estrutura é montada para que o codificador realize a redução da dimensionalidade espacial, utilizando camadas com filtros de convolução de tamanho 7×7. O codificador extrai características relevantes das imagens, que são então processadas pelo decodificador através de processos de aumento da amostragem. A última camada realiza a classificação de cada pixel do mapa de ativação para gerar a imagem segmentada. Na estrutura exemplificada na imagem, os parâmetros usados em camadas de redução também são utilizados nas camadas de aumento para atingir a mesma dimensionalidade.

2.3.1 *Backpropagation*

Uma parte muito importante para o treinamento bem sucedido de uma rede neural é a aplicação do método de *backpropagation*, que é responsável por ajustar os pesos internos de uma rede através de cálculos envolvendo o gradiente e uma função de perda (também chamada de função de *loss*.)

Especialmente com a implementação de redes com camadas ocultas, torna-se complexa a operação envolvendo os valores dos parâmetros treináveis nessas camadas. A aplicação do

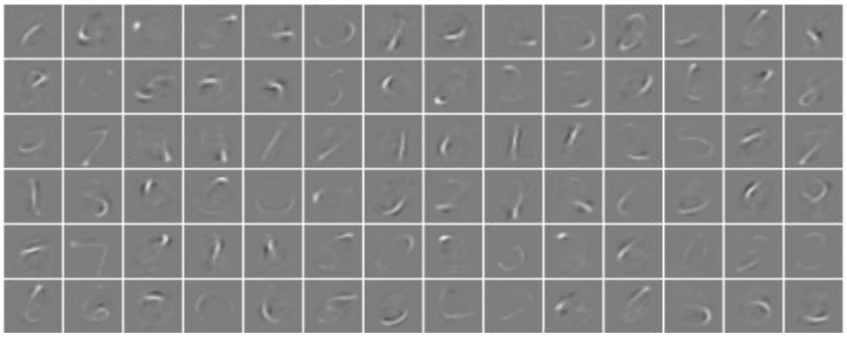
backpropagation auxilia no ajuste dos pesos utilizados através de cálculos envolvendo o gradiente descendente e a aplicação da regra de cadeia. (RUMELHART et al., 1986).

2.3.2 Redes Neurais Convolucionais

Um grande diferencial das redes neurais convolucionais é a capacidade de realizar a extração de características através de filtros que são utilizados em operações de convolução (LI et al., 2020). Os filtros possuem os parâmetros treináveis da rede e cada um deles é capaz de extrair diferentes características. Logo, o treinamento da rede é responsável pelo ajuste dos valores dos filtros, criando então a capacidade de se extrair as características desejadas.

O processo de treinamento permite que essas redes possam identificar padrões nas imagens de maneira semelhante à seres humanos, como pode ser visto na figura 2. Por exemplo, é comum que existam filtros que identifiquem faces, formas geométricas, bordas, dentre outros, como demonstrado em (ZEILER; FERGUS, 2013).

Figura 2 – Exemplo de padrões reconhecidos através de camadas convolucionais no conjunto de dados *MNIST*, que possui dígitos escritos à mão.



Fonte: (O'SHEA; NASH, 2015)

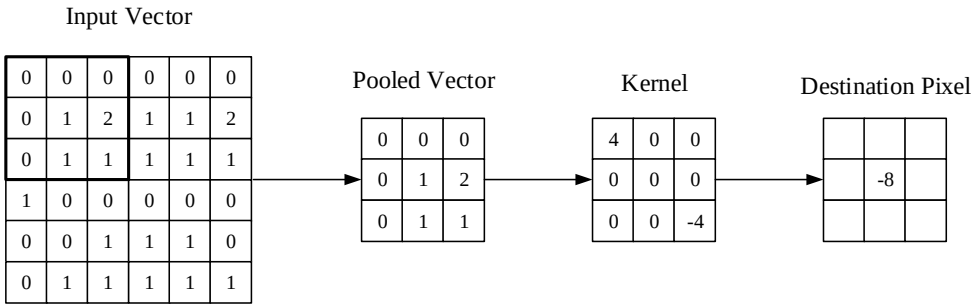
2.3.2.1 Convolução

A convolução é uma operação matemática realizada a partir de um sinal de entrada e um filtro. Quando se analisa o funcionamento da convolução em imagens, pode ser calculado a partir da seguinte equação, descrita em (GU et al., 2017):

$$z_{i,j,k} = \mathbf{w}_k{}^T \mathbf{x}_{i,j} + b_k \tag{2.1}$$

Para o cálculo, é separado um bloco de tamanho $n \times n$ da imagem, representado na equação por um vetor $\mathbf{x}_{i,j}$. Nesse bloco é então realizada a média ponderada com o filtro da convolução, como pode ser visto na figura 3. Os valores i e j se referem às posições do pixel central do bloco selecionado. $\mathbf{w_k}$ é o filtro e b_k o termo de *bias*. A letra k representa o canal, visto que podem ser trabalhadas imagens com múltiplos canais, por exemplo *RGB*.

Figura 3 – Ilustração visual de uma convolução bidimensional



Fonte: (O'SHEA; NASH, 2015)

2.3.3 Codificadores

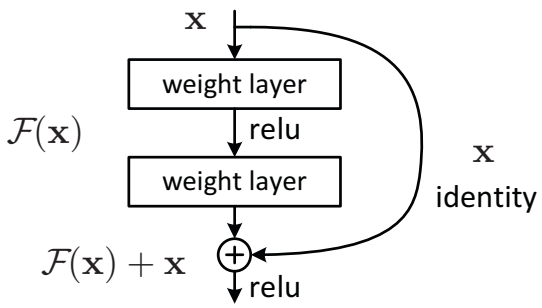
Nesta seção são apresentados os principais codificadores utilizados em arquiteturas codificador-decodificador na literatura.

2.3.3.1 *ResNet*

Definida em (HE et al., 2015), a *ResNet* utiliza um conceito que foi chamado de *Deep Residual Learning framework*. Um ponto comum em implementações de muitos modelos de aprendizado profundo é que o aumento da quantidade de camadas não necessariamente trazia benefícios para a performance da rede. De forma oposta, o aumento do número de camadas ocasionava um problema que foi chamado de problema da degradação, que fazia com que o aumento da quantidade de camadas e complexidade da rede resultasse em uma perda de desempenho.

As *ResNets* foram criadas inicialmente como uma nova forma de evitar esse conhecido problema através de conexões com atalhos entre as camadas, como pode ser observado na figura 4 que ilustra um bloco com duas camadas e o mecanismo de atalho entre elas.

Figura 4 – Arquitetura de um bloco de camadas da *ResNet*



Fonte: (HE et al., 2015)

2.3.3.2 *RegNet*

Como descrito em (RADOSAVOVIC et al., 2020), recentemente o método conhecido como *NAS (Neural Architecture Search)* tem ganhado muita popularidade e atingindo ótimos resultados na construção de redes. Para a construção da *RegNet* foi pensado em combinar vantagens de métodos mais automatizados como o *NAS* com a construção e design manual de aspectos importantes da arquitetura.

Inicialmente, foi criado manualmente um espaço abrangente contendo uma população de redes, chamado de *AnyNet*. Após isso, esse espaço e as respectivas redes passam por restrições de combinações de parâmetros, e é criado um espaço de menor dimensionalidade. Esse novo espaço contém as redes que são chamadas no artigo de redes regulares e o espaço então é chamado de *RegNet*.

Isso significa então que as redes encontradas não foram manualmente geradas uma a uma, mas foram criadas a partir de um espaço de design manualmente pensado e adaptado para conter uma população de redes.

2.3.3.3 *EfficientNet*

A criação da família de modelos que compõem as *EfficientNets* também envolveu a utilização do *NAS*. Nesse caso partiu-se do princípio do aumento do tamanho das redes. Até então, quando se desejava aumentar o tamanho ou complexidade da rede haviam algumas alternativas: largura, profundidade e resolução.

Frequentemente esses aumentos eram realizados de maneira arbitrária e manual, muitas vezes realizado em somente uma das dimensões. No artigo (TAN; LE, 2020) é descrito como foi pensada a ideia de incrementar a rede de modo proporcional e em todas as dimensões, largura, profundidade e resolução, simultaneamente. A figura 5 demonstra visualmente essa abordagem.

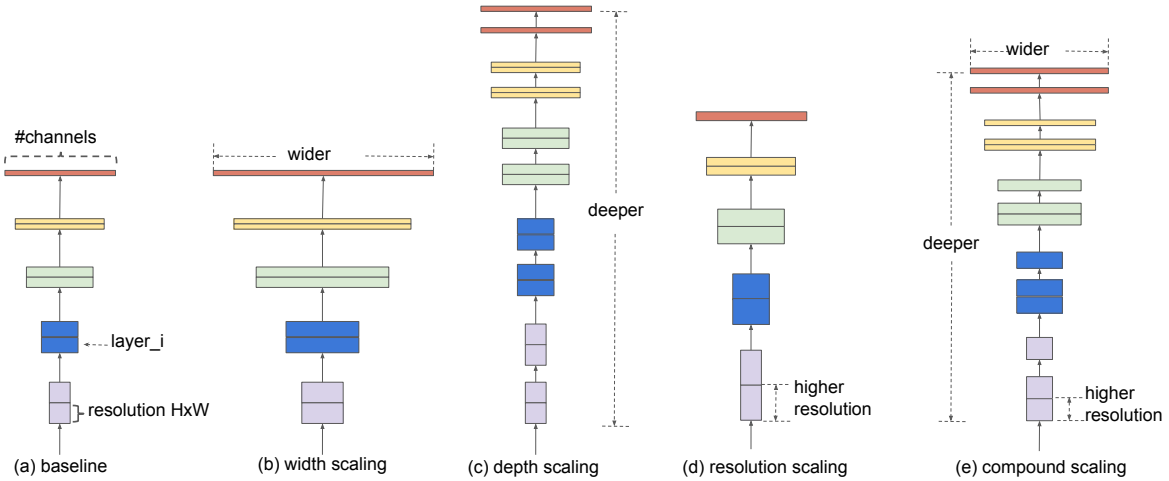
Portanto, foi utilizado o *NAS* para se criar um modelo base que através de um único parâmetro permite definir uma família de modelos chamados de *EfficientNet*.

2.3.3.4 *DenseNet*

O grande diferencial na construção das *DenseNets* é o número elevado de conexões entre as camadas da rede, construindo redes em que todas as camadas são conectadas entre si, como pode ser visto na figura 6.

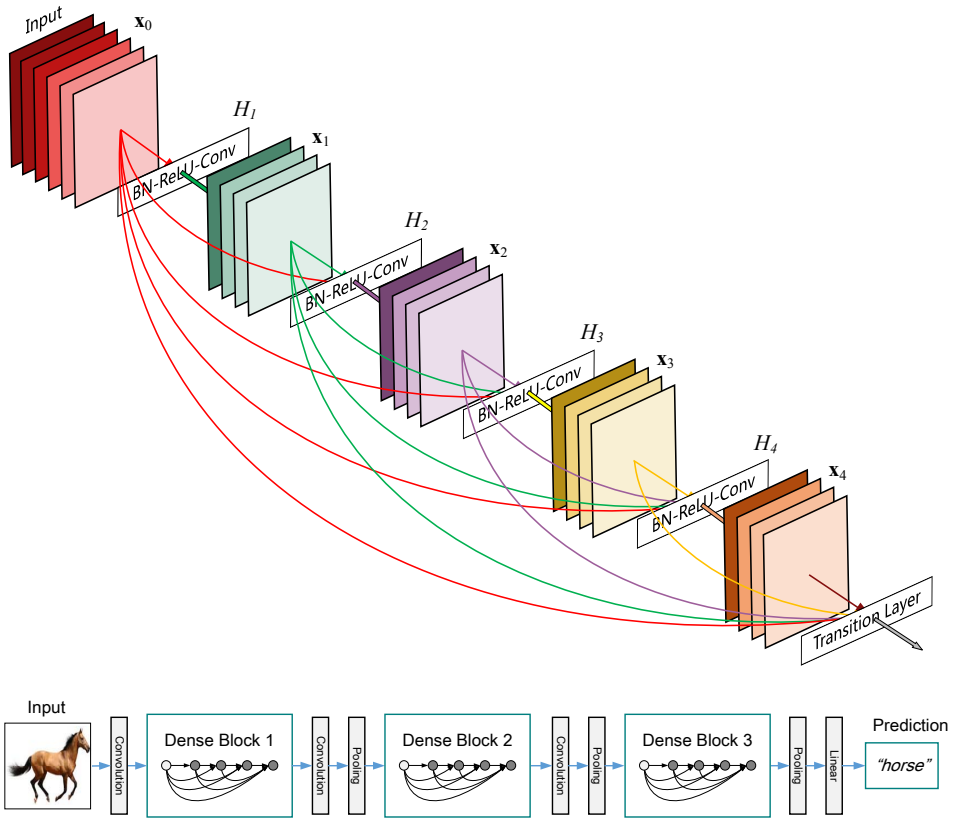
O nome surge por conta da elevada densidade de conexões internas entre as camadas. Um efeito interessante dessa alta conectividade entre as camadas é a necessidade de menos parâmetros na rede em comparação com implementações mais tradicionais de *CNNs*, surgindo assim a possibilidade de criação de redes mais eficientes.

Figura 5 – Comparação de estratégias de aumento de tamanho de redes feitas pela *EfficientNet*.



Fonte: (TAN; LE, 2020)

Figura 6 – Arquitetura *DenseNet*



Fonte: (HUANG et al., 2017)

2.3.3.5 *SwinV2*

O codificador *SwinV2* é o mais discrepante dos escolhidos para esse trabalho pois, diferente de todos os outros selecionados, não é um modelo convolucional e sim um *Transformer*.

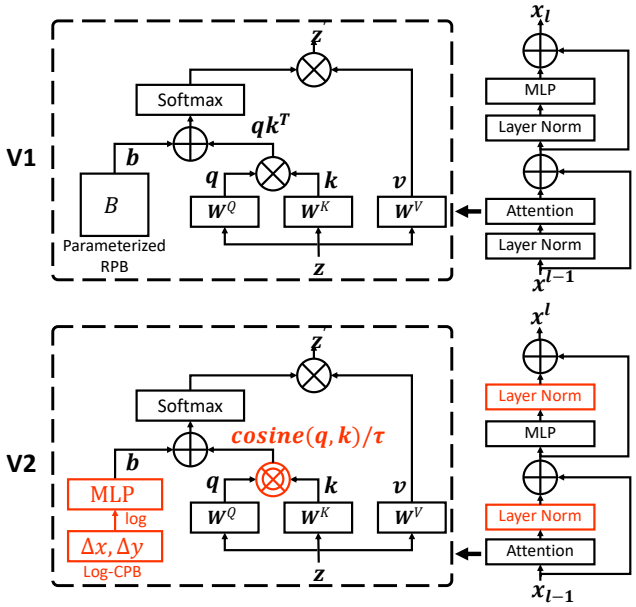
A tecnologia dos *transformers* foi proposta em (VASWANI et al., 2023) e por alguns

anos foi desenvolvida e utilizada principalmente no processamento de textos, especialmente na área de *NLP*. Em (DOSOVITSKIY et al., 2021) é estudado como podem ser utilizados também em tarefas de visão computacional. O principal ponto é a divisão das imagens de entrada em blocos menores que são tratados pela rede da mesma maneira que os *tokens* em *NLP*.

Definido em (LIU et al., 2022), o modelo *SwinV2* é uma evolução do modelo anterior chamado de *SwinV1*. Na figura 7 ambos os modelos podem ser comparados e é possível ver em destaque as adições no modelo mais atual.

Um dos grandes diferenciais do modelo em relação a outras arquiteturas *transformers* é a realização do mecanismo de atenção em janelas, o que permite mitigar o custo computacional de realizar a atenção entre todos os pares de regiões de uma imagem.

Figura 7 – Arquitetura dos modelos *SwinV1* e *SwinV2*.



Fonte: (LIU et al., 2022)

2.3.4 Decodificadores

Nesta seção são apresentados os principais decodificadores utilizados em arquiteturas codificador-decodificador na literatura.

2.3.4.1 *U-Net*

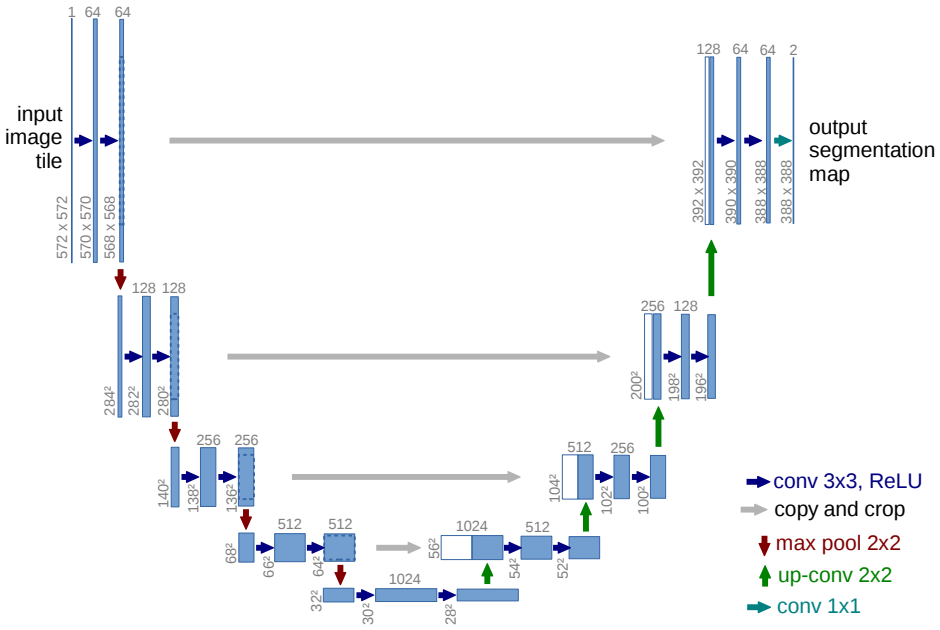
A tarefa de segmentação já era de interesse antes da criação da *U-Net*. Um dos melhores métodos até então utilizava janelas deslizantes ao redor de cada pixel que seria classificado individualmente pela rede formando uma segmentação completa ao final. Esse método foi proposto em (CIRESAN et al., 2012). Havia entretanto problemas em relação à aplicação, que dependia muito do tamanho da janela escolhida. Ao passo que uma janela muito grande impacta

no tempo de execução e recursos utilizados, e uma janela muito pequena pode diminuir demais o contexto para cada pixel.

Na tentativa de se criar soluções mais elegantes, a *U-Net* foi criada a partir de uma arquitetura chamada *FCN* e se constitui em duas partes, a primeira parte é formada por uma rede convolucional padrão, que cria mapas de características e realiza a redução da amostragem. Esses mapas de características passam por um aumento da amostragem na segunda parte da rede para que seja possível formar no fim um mapa da segmentação completa. (RONNEBERGER et al., 2015).

O nome dado à rede representa seu formato final, como pode ser visto na figura 8, onde a junção de um codificador e um decodificador forma uma estrutura semelhante ao formato da letra *U*.

Figura 8 – Arquitetura da *U-Net*



Fonte: (RONNEBERGER et al., 2015)

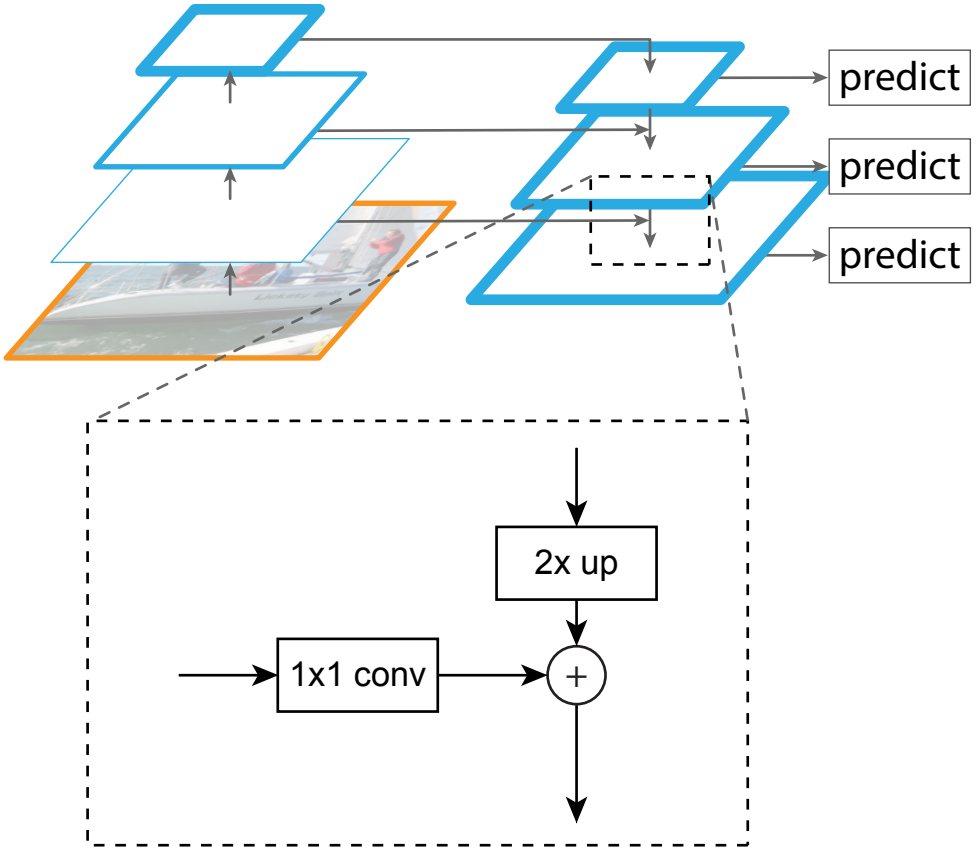
2.3.4.2 *FPN*

O nome *FPN* é uma sigla que significa *Feature Pyramid Network* e representa uma arquitetura de rede construída com base nas pirâmides de características que são muito utilizadas para reconhecimento de objetos.

Como as antigas pirâmides de características foram se tornando mais custosas computacionalmente com o aumento das resoluções das imagens e as redes neurais convolucionais representaram um grande avanço tecnológico, naturalmente, muitas tarefas que antes eram realizadas com as pirâmides passaram a ser realizadas com *CNNs*. Entretanto, ainda nessas *CNNs* havia internamente a utilização de pirâmides.

Com base nisso surgem as *FPNs*, como descrito em (LIN et al., 2017), que implementam *CNNs* mas com uma arquitetura em formato piramidal, que pode ser vista na imagem 9. Essa arquitetura se aproveita do processo interno de extração de características das *CNNs* e conecta lateralmente as pirâmides. Isso proporciona uma utilização da arquitetura de pirâmide mas sem que seja demasiadamente custoso computacionalmente.

Figura 9 – Arquitetura da *FPN*



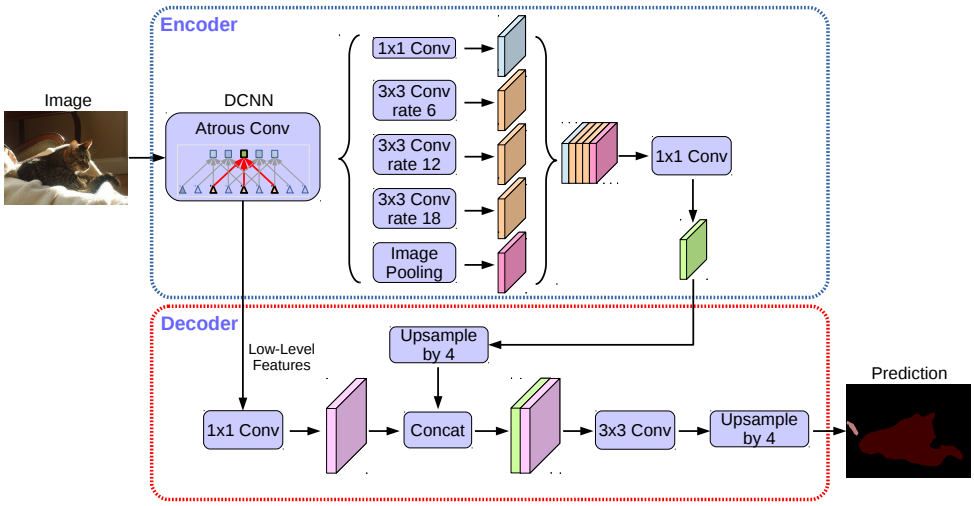
Fonte: (LIN et al., 2017)

2.3.4.3 *DeepLabV3+*

O modelo *DeepLabV3+* surge como uma melhoria do anterior *DeepLabV3*, proposto em (CHEN et al., 2017). O modelo anterior já era relativamente avançado, implementando uma combinação de *spatial pyramid pooling* com uma estrutura de *encoder-decoder*.

A iteração mais recente, criando o *DeepLabV3+*, refere-se à uma adição de um módulo *decoder*, visto na figura 10, especializado em refinar a segmentação para se recuperar melhor os limites do objeto à ser segmentado, como descrito em (CHEN et al., 2018). Os avanços foram suficientes para atingir a maior pontuação na época em *benchmarks* como o *PASCAL VOC 2012*.

Figura 10 – Arquitetura do *DeepLabV3+*



Fonte: (CHEN et al., 2018)

2.4 Métricas

Nessa seção serão apresentadas as métricas utilizadas nas análises, incluindo algumas breves contextualizações.

2.4.1 Número de parâmetros

A contagem de parâmetros é uma métrica muito importante na análise realizada, visto que os valores obtidos na contagem podem ser muito significativos nas comparações de complexidade e tamanho dos modelos.

Uma quantidade maior de parâmetros treináveis em um modelo pode influenciar positivamente na sua capacidade de aprendizado ao mesmo tempo que, em muitos casos, é responsável por uma menor eficiência em consumo de recursos e tempo. Portanto, é muito importante que ao realizar comparações entre modelos se leve em consideração os recursos utilizados, e a quantidade de parâmetros pode ser um grande indicador dos recursos necessários.

2.4.2 Consumo de memória

Assim como o número de parâmetros, o consumo de memória é um dado muito relevante ao tentar analisar a eficiência de um modelo implementado.

Do mesmo modo que a quantidade de parâmetros treináveis presentes em um modelo, o consumo de memória pode ser um indicativo muito valioso de performance e eficiência e sua inclusão nas análises pode ser útil na fundamentação das conclusões obtidas.

2.4.3 *Loss*

O valor utilizado para o cálculo da *loss* no treinamento foi o da entropia cruzada, que é um cálculo que envolve probabilidades entre distribuições distintas. No caso de uma rede neural esse cálculo envolve as predições realizadas e os valores alvo (LI et al., 2020).

A entropia cruzada é calculada através da seguinte equação:

$$l_n = -w_{y_n} log \frac{exp(x_{n,y_n})}{\sum_{c=1}^{C} exp(X_{n,c})} \cdot 1\{y_n \neq ignore\_index\} \tag{2.2}$$

onde x é a entrada, y é o *target*, w é o peso, C o número de classes, n o índice da imagem (em classificação) ou do pixel (em segmentação) e $ignore\_index$ algum valor no *target* que deve ser ignorado. E então é feita uma média dos valores calculados:

$$l(x, y) = \sum_{n=1}^{N} \frac{1}{\sum_{n=1}^{N} w_{y_n} \cdot 1\{y_n \neq ignore\_index\}} l_n \tag{2.3}$$

2.4.4 IoU

Também chamado de índice de Jaccard, o termo *IoU* pode ser traduzido para intersecção sobre união, logo, a métrica de *IoU* é calculada com base na comparação de conjuntos (COSTA, 2021).

Para a segmentação de imagens existem dois conjuntos, chamados de *alvo* e *predição*. Nesse trabalho, o conjunto *alvo* consiste na máscara anotada manualmente no conjunto de dados e o conjunto *predição* contém a segmentação realizada pela rede neural.

Para calcular o índice, a intersecção é formada pelos verdadeiros positivos, isto é, as áreas onde as segmentações se sobrepõem. A união é calculada pela soma das áreas dos verdadeiros positivos e também os falsos positivos e negativos. O valor é calculado a partir da seguinte equação:

$$IoU = \frac{TP}{TP + FP + FN} \tag{2.4}$$

na qual TP são os verdadeiros positivos, FP são os falsos positivos e FN são os falsos negativos.

2.4.5 clDice

Aliado ao *IoU*, o índice *clDice* é uma métrica muito importante na tentativa de avaliar o resultado de uma segmentação. O nome clDice é uma abreviação de *centerlineDice*. É uma métrica criada especificamente para avaliar a segmentação de estruturas tubulares ou lineares. (SHIT et al., 2021).

Para o cálculo são utilizadas duas máscaras. A máscara *target* nomeada V_L, que no caso deste trabalho foi anotada manualmente, e a máscara prevista pelo modelo nomeada V_P. Também

são extraídos, de ambas as máscaras, os esqueletos das estruturas tubulares S_L e S_P. Após isso são calculados os valores da Precisão da Topologia:

$$Tprec(S_P, V_L) = \frac{|S_P \cap V_L|}{S_P} \tag{2.5}$$

E da Sensitividade da Topologia:

$$Tsens(S_L, V_P) = \frac{S_L \cap V_P}{S_L} \tag{2.6}$$

Finalmente, o cálculo final da métrica é feito através da média harmônica entre os dois valores:

$$clDice(V_P, V_L) = 2 \times \frac{Tprec(S_P, V_L) \times Tsens(S_L, V_P)}{Tprec(S_P, V_L) + Tsens(S_L, V_P)} \tag{2.7}$$

Capítulo 3
REVISÃO DA LITERATURA

A segmentação de objetos em imagens é um tópico de grande interesse dentro de muitas áreas da computação. Por exemplo, (XU et al., 2016) desenvolveu um algoritmo para a tarefa de seleção interativa de objetos em imagens, isto é, quando um usuário seleciona com um dispositivo, um *mouse*, por exemplo, um objeto de uma imagem. A implementação, baseada em aprendizado profundo, foi capaz de melhorar a precisão ao mesmo tempo em que reduziu a quantidade necessária de cliques para a seleção do objeto. Apesar de tematicamente diferente desse trabalho, é um estudo interessante pois pode apresentar uma arquitetura capaz de selecionar partes menores e mais complexas dos objetos segmentados. O algoritmo proposto utiliza o cálculo de distâncias euclidianas para realizar um refinamento do modelo baseado em *FCN*.

Ainda sobre a segmentação de objetos, (HE et al., 2018) desenvolveu um método focado em segmentação de instâncias chamado *Mask R-CNN*, que é um tipo de segmentação que envolve detecção de objetos. O método desenvolvido é baseado em redes convolucionais, especialmente em uma abordagem chamada *R-CNN* desenvolvida em (GIRSHICK et al., 2014). A *R-CNN* é focada na utilização das regiões da imagem para a extração de mapas de características que podem auxiliar na tarefa de segmentação.

O método *Mask R-CNN* implementa a utilização das máscaras dos objetos no modelo *R-CNN*, e faz isso em paralelo com a predição de classe. Dessa forma, é criado um modelo eficiente e não muito complexo. Nos testes realizados na publicação, a rede foi implementada junto com um *backbone ResNet* e atingiu ótimos resultados na segmentação de instâncias.

A segmentação de vasos sanguíneos tem sido alvo de um grande interesse há vários anos, mesmo antes do grande aumento na utilização de redes neurais. Por exemplo, (RICCI; PERFETTI, 2007) e (NGUYEN et al., 2013) utilizaram métodos de detecção de linhas através do canal verde invertido nas imagens de retina para realizar a tarefa.

A ideia nesses modelos de detecção de linhas é a seguinte. Para cada *pixel* é criada uma janela de tamanho $W \times W$ e o valor médio do nível de cinza é chamado de I_{avg}^{W}. Após isso, várias linhas com tamanho de W pixels cruzam o *pixel* central por ângulos diferentes e é calculado o valor do nível de cinza nos *pixels* que passam por cada uma das linhas. A linha com maior valor,

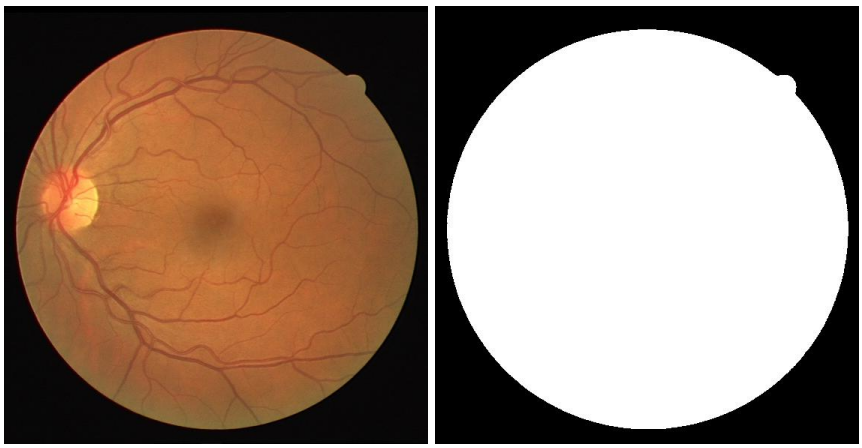
I^W_{max}, é obtida. Por fim, é calculado então $R_W = I^W_{max} - I^W_{avg}$. Caso o resultado seja um valor alto o *pixel* deve ser um *pixel* de vaso sanguíneo, caso contrário deve ser um *pixel* de fundo.

Em 2012, a publicação (FRAZ et al., 2012) realizou um estudo de comparação entre diversos algoritmos no estado da arte na época, envolvendo técnicas de aprendizado supervisionado, como redes neurais e *SVM*, e também técnicas de aprendizado não-supervisionado, como técnicas de agrupamento. Também foram testadas, inclusive, várias outras técnicas que não envolvem necessariamente o aprendizado de máquina.

Dado o grande avanço nas áreas de aprendizado de máquina nos últimos anos, a relevância atual dos métodos que foram testados nessa publicação não é muito grande. No entanto, o artigo ainda traz contribuições importantes, como a padronização das métricas utilizadas para a avaliação dos algoritmos e as bases de dados como a *DRIVE* (STAAL et al., 2004) e *STARE* (HOOVER et al., 2000), que têm sido amplamente utilizadas em diversas publicações e estudos na área.

Em muitas publicações na área de segmentação de vasos sanguíneos são utilizados conjuntos de imagens de retina, como os previamente citados, muitas vezes sendo imagens circulares com uma grande borda preenchida de zeros, como pode ser visto na figura 11. O artigo (KOVÁCS; FAZEKAS, 2022) traz uma abordagem interessante sobre a utilização desses *datasets* e o cálculo de métricas, visto que muitos artigos envolvem os trechos fora da retina nos cálculos de acurácia. A publicação realiza uma compensação nos cálculos de mais de 100 outros artigos na tentativa de balancear os valores publicados.

Figura 11 – Exemplo de imagem de fundo de olho e respectiva máscara indicando a região da retina.



Fonte: (KOVÁCS; FAZEKAS, 2022)

Em (LI et al., 2024) são analisados e comparados métodos propostos de segmentação utilizando *transformers*. É um estudo abrangente, que não somente apresenta e compara dezenas de abordagens, mas também apresenta uma extensa fundamentação teórica que detalha profundamente os modelos e estruturas presentes.

Já em (MOCCIA et al., 2018), podem ser encontradas comparações e métricas de dezenas de modelos que têm tido alta notoriedade nos últimos anos, como por exemplo as *CNNs* que são o principal objeto de estudo deste trabalho. A análise é feita com base no desempenho dos modelos na tarefa de segmentação de vasos sanguíneos em diferentes tipos de imagens, como cérebro, fígado e retina, por exemplo, e são utilizados mais de dez *datasets* diferentes.

Em (HAQUE; NEUBERT, 2020) é feita uma análise de publicações recentes, entre 2015 e 2019, que utilizam técnicas de *Deep Learning* para realizar diferentes tipos de segmentação em imagens médicas, como tomografias, ressonâncias magnéticas, raio-x, dentre outras. Também há uma contribuição muito grande na publicação no que envolve as métricas à serem utilizadas na avaliação dos modelos testados, como o índice de similaridade de *Jaccard*, ou *IoU*, o coeficiente *DICE*, precisão, *recall* e acurácia.

As publicações previamente citadas se especificam na tarefa de segmentação de vasos sanguíneos, que é semelhante à feita neste trabalho. Contudo, ainda é importante a pesquisa de comparações mais abrangentes como (SULTANA et al., 2020), que possui uma análise focada na evolução das técnicas de segmentação de imagens através de *CNNs* e inclui modelos que serão analisados neste trabalho, como por exemplo *U-Net* e *DeepLabV3*. O artigo também realiza um estudo na busca dos melhores hiperparâmetros para cada um dos modelos testados que é uma tarefa muito importante no processo de implementação de um modelo.

Outra comparação de modelos é a encontrada em (GALDRAN et al., 2022), onde são testados vários modelos de publicações populares em *benchmarks* muito conhecidos. Um grande diferencial nessa publicação é a inclusão nos testes de modelos drasticamente menores, como *Little U-Net* e *Little W-Net*, ambos com menos de 100 mil parâmetros treináveis, e que ainda assim são capazes de atingir resultados satisfatórios como por exemplo valores maiores que 80 no coeficiente *DICE* no *benchmark* do conjunto *DRIVE*.

Há também (GOLDBLUM et al., 2024) no qual é feita uma extensa comparação em diferentes categorias como classificação, detecção de objetos e segmentação. Um ponto muito interessante dessa publicação é que, por conta de ser mais recente que as anteriores, é realizada a comparação entre modelos modernos como *ResNet-18*, *EfficientNet-B0* e *SwinV2*. Ainda sobre essa publicação, é válido pontuar que o modelo *EfficientNet-B0* obteve os melhores resultados em algumas categorias e também foi observado que os modelos *ResNet* ainda são capazes de apresentar resultados comparáveis com modelos mais recentes e atuais.

Capítulo 4

METODOLOGIA E DESENVOLVIMENTO

Nesse capítulo são discutidos os tópicos envolvendo o conjunto de dados utilizado, o pré-processamento realizado nesses dados, os modelos de redes neurais e a escolha que levou à cada um desses modelos, assim como todo o processo de treinamento, validação, coleta e exibição de resultados.

4.1 Conjunto de dados

O conjunto de dados escolhido para treinamento e teste foi o elaborado por(SILVA et al., 2024), que é um conjunto que contém 100 imagens de microscopia de tecido cerebral de ratos. Essas 100 imagens foram selecionadas a partir de outros conjuntos pré-existentes. As imagens foram escolhidas para que, apesar do número pequeno no conjunto, possuíssem uma alta representatividade tentando ter um conjunto balanceado com imagens com uma grande variação de características dos vasos.

Com a finalidade de simularmos situações próximas da prática do dia-a-dia, foi decidido utilizar uma fração pequena do conjunto para treinamento, e a maior parte para a avaliação. Por uma questão de eficiência, a quantidade de imagens para cada parte devia ser um número múltiplo de 4, que é o tamanho dos *batches* utilizados no treinamento e teste. Portanto, foi utilizado um conjunto de 12 imagens para treinamento e 88 imagens para teste das redes.

4.1.1 Pré-processamento dos dados

Antes de serem submetidas ao treinamento e teste dos modelos, as imagens passaram por alguns processos incluindo aumento de dados e redimensionamento.

Todas as imagens passaram por um processo chamado de *Random Resize Crop*, onde uma região da imagem é escolhida aleatoriamente e recortada. A região é então redimensionada para o tamanho original da imagem. No caso desse trabalho, foram escolhidas regiões que correspondem a 90% da área original da imagem. As regiões foram redimensionadas para o tamanho 256×256.

As imagens passaram também por espelhamentos horizontais e verticais aleatórios, que fazem um papel muito importante no *data augmentation*, especialmente com o baixo número de imagens utilizadas para treinamento. Por fim, as imagens passaram por um processo de normalização, garantindo que todas as imagens tenham média 0 e variância 1.

4.2 Modelos utilizados

Os modelos foram construídos na linguagem de programação *Python*, com a utilização da biblioteca *PyTorch* (ANSEL et al., 2024) e também a biblioteca *TorchSeg*, que é um *fork* da biblioteca *Segmentation Models* (IAKUBOVSKII, 2019).

Os modelos utilizados são todos com pesos pré-treinados no *dataset ImageNet*, descrito em (RUSSAKOVSKY et al., 2015). Esse pré-treinamento dos pesos é muito importante para facilitar a etapa do treinamento da rede na tarefa de segmentação dos vasos sanguíneos.

O principal critério de escolha para os modelos foi a popularidade e a relevância atual. Dessa forma, foi buscado encontrar codificadores e decodificadores que são frequentemente abordados em estudos na área. Outro fator importante considerado foi o tamanho dos modelos, visto que existe uma gama muito grande de modelos disponíveis de diferentes tamanhos, complexidades e características.

Um dos pontos centrais na análise dos resultados será a comparação dos valores atingidos entre modelos de tamanhos muito diferentes. Dessa forma, foram escolhidos modelos que possuem entre 5 e 15 milhões de parâmetros e modelos que possuem entre 30 e 40 milhões de parâmetros. Na tabela 1 são encontradas todas as combinações de codificadores e decodificadores que foram testadas. Algumas combinações não estão presentes por conta de incompatibilidades nos modelos presentes nas bibliotecas utilizadas. São as combinações envolvendo *DeepLabv3+* com *DenseNets* e *SwinV2*.

A diferença de tamanho e complexidade dos modelos é mostrada visualmente nas figuras 12 e 13. Quase todos os modelos testados são modelos envolvendo *CNNs*, exceto o modelo *SwinV2* que é um modelo *transformer*, como explicado na seção 2.3.3.5.

4.3 Escolha dos hiperparâmetros

Como o número de combinações de modelos testados foi elevado, se tornou inviável a busca pelo conjunto ideal de hiperparâmetros específico para cada um dos modelos. Dessa forma, o método utilizado foi a busca pela combinação de hiperparâmetros que atingisse o melhor desempenho na combinação *U-Net + ResNet50*, pois tanto o codificador quanto decodificador possuem uma grande popularidade atualmente.

Para encontrar os melhores valores, foram testadas 40 combinações possíveis, com o

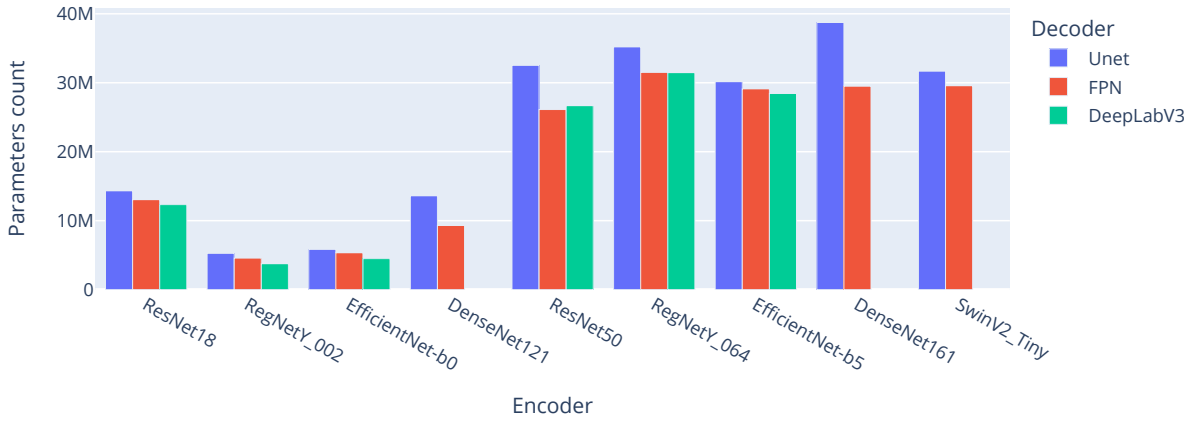
Tabela 1 – Modelos utilizados nos testes. Os nomes dos modelos possuem o padrão “decodificador + codificador".

Nome do modelo	Quantidade de parâmetros (milhões)	Custo de Memória (MiB)
U-Net + ResNet18	14,3	159.6
U-Net + RegNetY_002	5,3	40.4
U-Net + EfficientNet-B0	5,8	45.5
U-Net + DenseNet121	13,6	104.3
U-Net + ResNet50	32,5	251.6
U-Net + RegNetY_064	35,2	271.8
U-Net + EfficientNet-B5	30,2	235.7
U-Net + DenseNet161	38,7	298.9
U-Net + SwinV2_Tiny	31,7	244.7
FPN + ResNet18	13,0	152.7
FPN + RegNetY_002	4,6	84.7
FPN + EfficientNet-B0	5,3	41.1
FPN + DenseNet121	9,3	73.1
FPN + ResNet50	26,1	203.4
FPN + RegNetY_064	31,5	245.8
FPN + EfficientNet-B5	29,1	225.9
FPN + DenseNet161	29,5	231.1
FPN + SwinV2_Tiny	29,6	232.1
DeepLabV3+ + ResNet18	12,3	144.2
DeepLabV3+ + RegNetY_002	3,8	77.5
DeepLabV3+ + EfficientNet-B0	4,5	34.6
DeepLabV3+ + ResNet50	26,7	207.2
DeepLabV3+ + RegNetY_064	31,5	244.4
DeepLabV3+ + EfficientNet-B5	28,4	220.8

Fonte: Próprio Autor

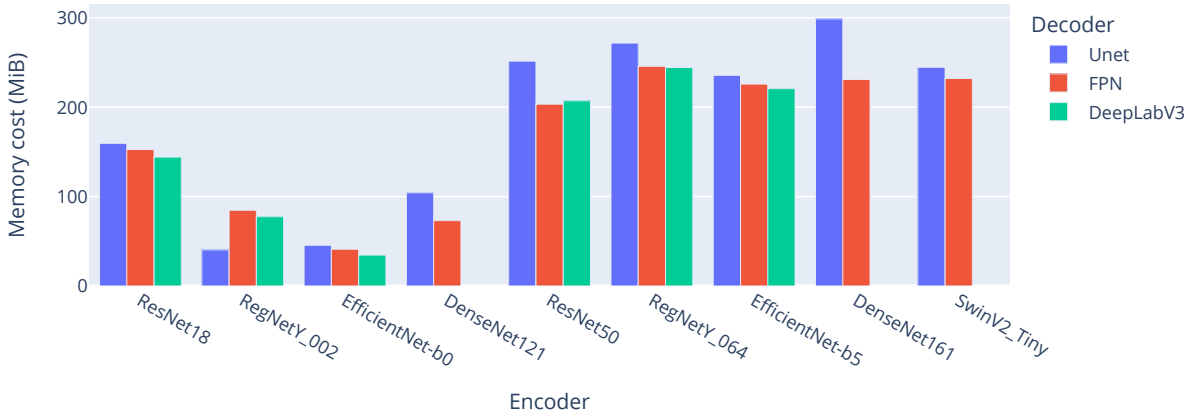
parâmetro *weight_decay* variando de 1×10^{-3} a 0, com os valores sendo multiplicados por 10, e a taxa de aprendizado variando entre 1×10^{-4} e 0.5, com os valores seguindo a progressão $(1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 5 \times 10^{-3}, ...)$. Cada uma das combinações foi testada por 100 épocas de treinamento no modelo escolhido e por fim, o valor final para *weight_decay* foi de

Figura 12 – Comparação da quantidade de parâmetros presentes em cada um dos modelos testados.



Fonte: Próprio Autor

Figura 13 – Comparação do consumo de memória na *GPU* realizado por cada modelo testado.



Fonte: Próprio Autor

1×10^{-6} e o valor para o *learning_rate* foi de 0.1.

4.4 Treinamento e validação

Toda a análise foi estruturada na ideia de simular, na medida do possível, uma aplicação mais próxima da realidade. Portanto, é necessário minimizar o tempo e recursos investidos na segmentação manual das imagens e por conta disso foi utilizado uma baixa porcentagem do *dataset* para o treinamento dos modelos.

O processo foi inteiramente realizado em uma *GPU* modesta, um modelo *NVIDIA GTX 1650* com 4GB de *VRAM*. O maior número de imagens possíveis para os *batches* de treino e teste foi de 4 imagens para cada um. Apesar de algumas redes serem menores e comportarem a utilização de *batches* maiores, foi buscado utilizar valores iguais para todas as redes, dessa forma,

os *batches* com 4 imagens foram os maiores capazes de serem utilizados em todas as redes.

Cada combinação de modelos possui suas próprias características e particularidades de maneira que cada combinação poderia ter um treinamento customizado. Entretanto, por conta das limitações de tempo e escopo do trabalho isso se torna inviável. Assim, todos os testes foram realizados com os mesmos hiperparâmetros e na quantidade fixa de 1500 épocas de treinamento e validação.

Para analisar os resultados obtidos foram escolhidas as métricas de *IoU* e *clDice*. As métricas foram calculadas em três momentos distintos do treinamento: i) na época final de treinamento; ii) na época com o menor valor de *Loss* na validação; iii) após 30 minutos de treinamento. A avaliação após 30 minutos permite uma comparação justa entre modelos mais eficientes que treinam mais rapidamente do que os demais. É importante salientar que todos os treinamentos completos duraram mais do que 30 minutos.

Por fim, foi construído um repositório no *GitHub* com todos os códigos criados e utilizados na elaboração dos testes e análise dos resultados. O repositório foi disponibilizado no endereço: https://github.com/IvanCalvo/segmentacao-vasos-sanguineos/.

4.5 Coleta e exibição de resultados

Os resultados de *IoU* e *clDice* nas épocas finais e épocas com menor *Loss* de cada modelo foram anotados manualmente a partir dos *checkpoints* gerados pelos treinamentos. Já os dados referentes à progressão das métricas de acordo com a época ou tempo foram coletados através do *TensorBoard*, que faz parte da biblioteca *TensorFlow* (ABADI et al., 2015).

Os dados sobre os modelos utilizados foram calculados de acordo com o descrito na seção 2.4, e os gráficos com as informações sobre os modelos e resultados, presentes nesse capítulo e no próximo, foram todos produzidos com a utilização da biblioteca *Plotly* (INC., 2015).

Capítulo 5
RESULTADOS

Nesse capítulo são analisados e comparados os resultados obtidos por cada combinação de modelos na tarefa de segmentação de vasos sanguíneos. As comparações são realizadas através das métricas descritas na seção 2.4. Simultaneamente, são feitas análises comparativas na tentativa de compreender as vantagens e desvantagens de cada modelo.

5.1 Resultados gerais

Na tabela 2 são encontrados os resultados para a última época e para a época com menor valor de *loss* de teste de todas as combinações de modelos testados. A evolução do treinamento de todos os modelos é mostrada nos gráficos das figuras 14 e 15. As curvas mostradas nas figuras representam a progressão de *loss* e *IoU* em todos os modelos testados e também as comparações entre todas as combinações feitas.

Os gráficos apresentados possuem o propósito de auxiliar em uma visão do quadro geral dos treinamentos, mas por conta da alta quantidade de modelos testados, a visualização única de cada modelo não é possível. No entanto, no repositório publicado no *GitHub* (no endereço https://github.com/IvanCalvo/segmentacao-vasos-sanguineos/blob/main/src/code/Plots.ipynb) é possível obter os gráficos em versões interativas.

Na figura 16 são mostrados os valores de *IoU* para a época final de cada modelo. Por último, na figura 17 são comparados os resultados atingidos por todos os modelos após 30 minutos de treinamento. Nessa figura se destaca bastante a superioridade atingida nos testes pelos modelos utilizando *U-Net*.

5.1.1 Valores nulos

O primeiro ponto de destaque são alguns valores nulos encontrados. Eles se devem ao fato de que dependendo da segmentação realizada pela rede, o cálculo do *clDice*, definido na seção 2.4.5, pode resultar em uma tentativa de divisão por zero e por conta disso resulta em valores não-numéricos que, para uma melhor exibição, foram convertidos em zero.

Tabela 2 – Resultados dos treinamentos dos modelos. Em negrito é mostrado o maior valor de cada métrica de qualidade.

Nome do Modelo	Época Final		Época com menor *Loss*		Parâmetros (M)	Memória (MiB)
	IoU	clDice	IoU	clDice		
U-Net + ResNet18	0.7859	0.9355	0.7801	0.9264	14.3	159.6
U-Net + RegNetY_002	0.7623	0.9203	0.7576	0.9214	5.3	40.4
U-Net + EfficientNet-B0	0.7791	0.9232	0.7570	0.9138	5.8	45.5
U-Net + DenseNet121	0.7812	0.9313	0.7817	0.9300	13.6	104.3
U-Net + ResNet50	0.7708	0.9223	0.7615	0.9169	32.5	251.6
U-Net + RegNetY_064	0.7925	**0.9371**	0.7693	0.9249	35.2	271.8
U-Net + EfficientNet-B5	0.7880	0.9336	0.7743	0.9269	30.2	235.7
U-Net + DenseNet161	**0.7932**	0.9357	**0.7902**	**0.9336**	38.7	298.9
U-Net + SwinV2_Tiny	0.7480	0.9254	0.7387	0.9150	31.7	244.7
FPN + ResNet18	0.6739	0.8792	0.6737	0.8830	13.0	152.7
FPN + RegNetY_002	0.6201	0.8204	0.6246	0.8308	4.6	84.7
FPN + EfficientNet-B0	0.7504	0.9212	0.7325	0.9197	5.3	41.1
FPN + DenseNet121	0.5183	0	0.5213	0	9.3	73.1
FPN + ResNet50	0.6122	0	0.6170	0.8156	26.1	203.4
FPN + RegNetY_064	0.6221	0.8155	0.6209	0.8259	31.5	245.8
FPN + EfficientNet-B5	0.5655	0.7477	0.6456	0.8601	29.1	225.9
FPN + DenseNet161	0.6251	0.8391	0.6378	0.8534	29.5	231.1
FPN + SwinV2_Tiny	0.6217	0.8460	0.6264	0.8524	29.6	232.1
DLV3+ + ResNet18	0.7277	0.9073	0.7083	0.8978	12.3	144.2
DLV3+ + RegNetY_002	0.7186	0.8943	0.7104	0.9073	3.8	77.5
DLV3+ + EfficientNet-B0	0.6988	0.8552	0.6456	0.8625	4.5	34.6
DLV3+ + ResNet50	0.7094	0.8924	0.7128	0.8956	26.7	207.2
DLV3+ + RegNetY_064	0.7496	0.9175	0.7200	0.9066	31.5	244.4
DLV3+ + EfficientNet-B5	0.7715	0.9224	0.7280	0.8999	28.4	220.8

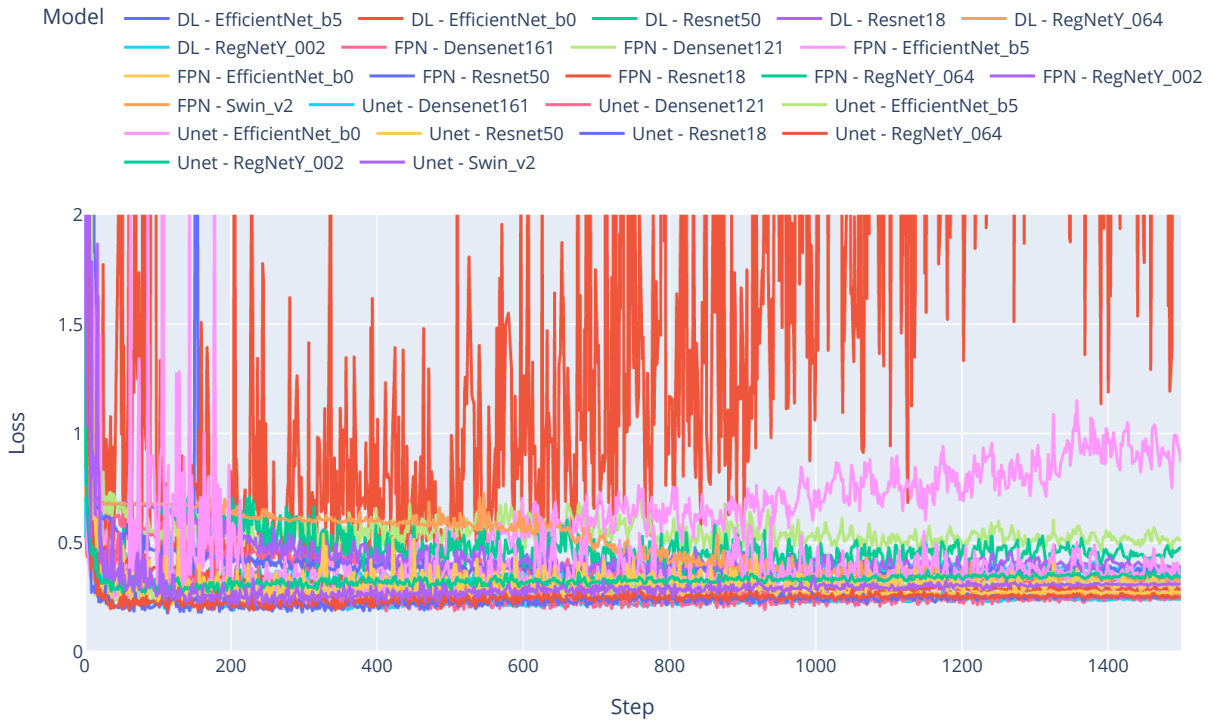
Fonte: Próprio Autor

5.1.2 Melhores resultados

Todas as combinações utilizando modelos *U-Net* atingiram resultados muito positivos, com *IoU* muito próximo a 80 e *clDice* sempre acima de 90. Os melhores resultados foram obtidos em redes construídas com os codificadores *DenseNet161*, *ResNet18* e *RegNetY_064*.

Ainda pontuando os melhores resultados, se destacam as combinações *FPN + EfficientNet-B0*, *DeepLabV3+ + ResNet18* e *DeepLabV3+ + EfficientNet-B5*.

Figura 14 – Gráfico dos valores de *Loss* de validação em cada época nos modelos testados.



Fonte: Próprio Autor

5.1.3 Piores resultados

Os resultados mais baixos foram referentes à combinações utilizando *FPN* como decodificador. Inclusive, os modelos com codificadores *DenseNet121* e *ResNet50* não conseguiram resultados suficientes para o cálculo do *clDice* na maioria das épocas.

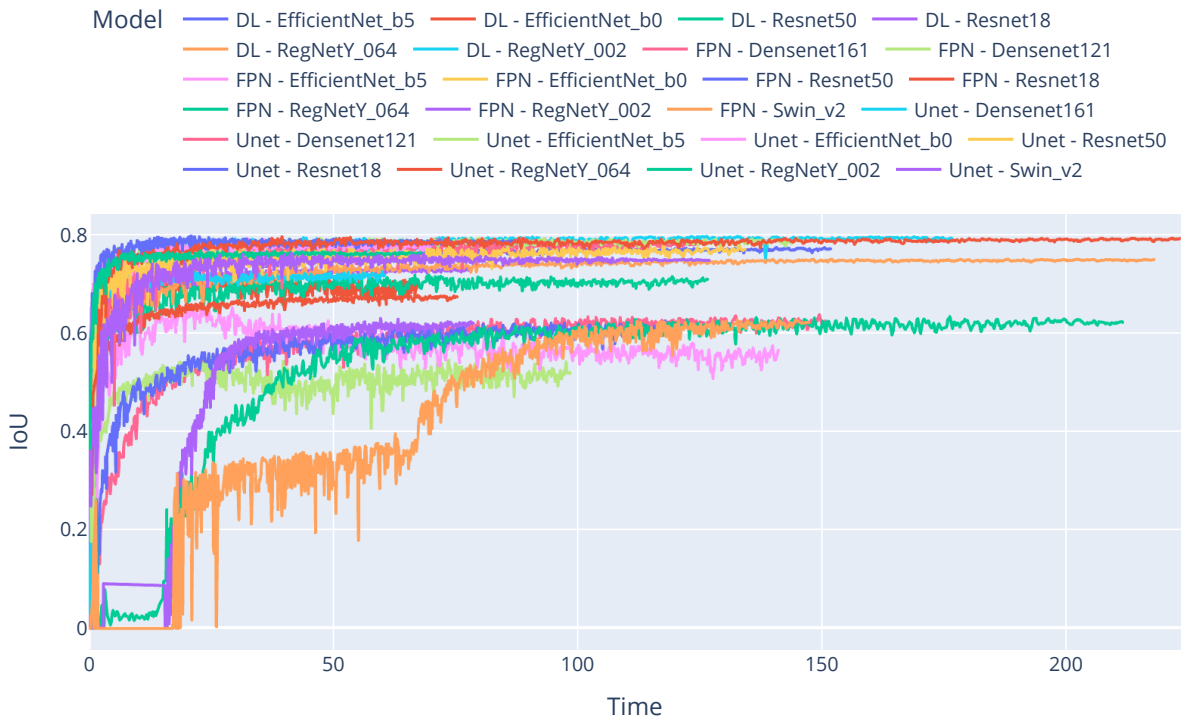
5.2 Tamanho dos modelos

A comparação de resultados entre modelos com quantidades muito discrepantes de parâmetros é de muito interesse e as figuras 18 e 19 representam uma visualização desse ponto. Nessas figuras pode se comprovar como na maior parte dos casos as combinações envolvendo modelos com uma quantidade menor de parâmetros foi suficiente para alcançar resultados muito semelhantes tanto em *IoU* quanto *clDice*.

Através da comparação dos resultados envolvendo combinações de redes da mesma família, o único caso em que há uma notável melhoria nos resultados é com a comparação: *DeepLavV3+ + EfficientNet-B0* e *DeepLavV3+ + EfficientNet-B5*. Em todas as outras situações as redes com menor tamanho parecem muito mais vantajosas na situação testada, chegando a resultados muito próximos ou até melhores que as redes maiores.

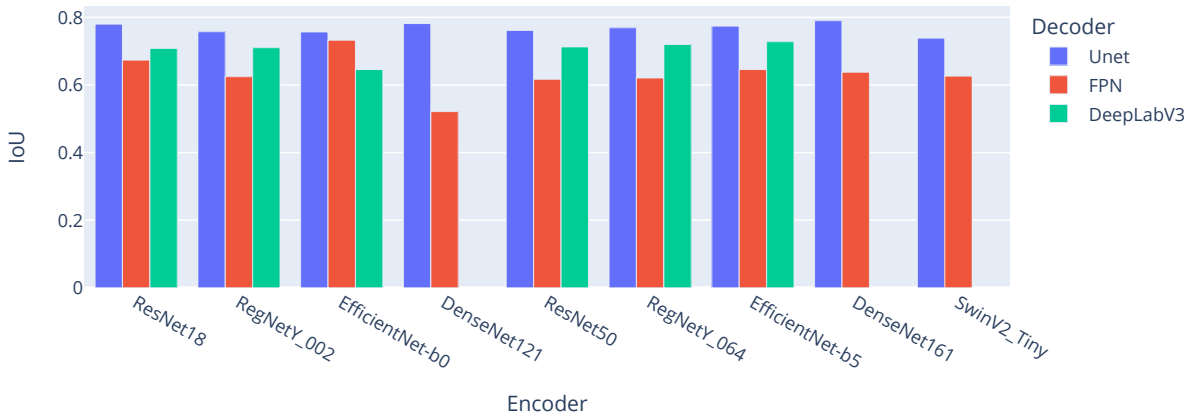
O tamanho e profundidade dos modelos implementados nos testes também possui uma

Figura 15 – Gráfico dos valores de *IoU* obtidos por cada modelo em relação ao tempo de treinamento.



Fonte: Próprio Autor

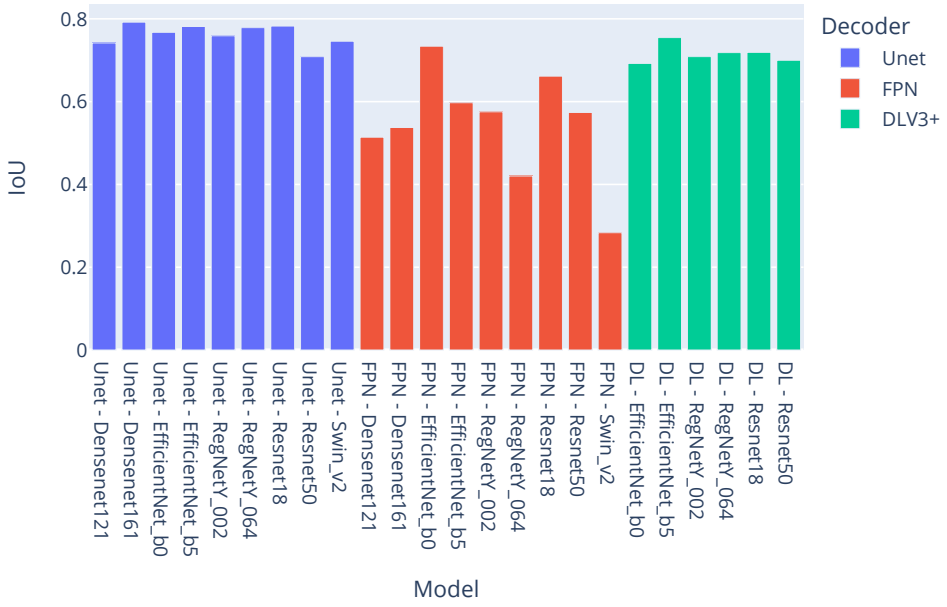
Figura 16 – Comparação dos valores finais de *IoU* entre cada um dos modelos testados.



Fonte: Próprio Autor

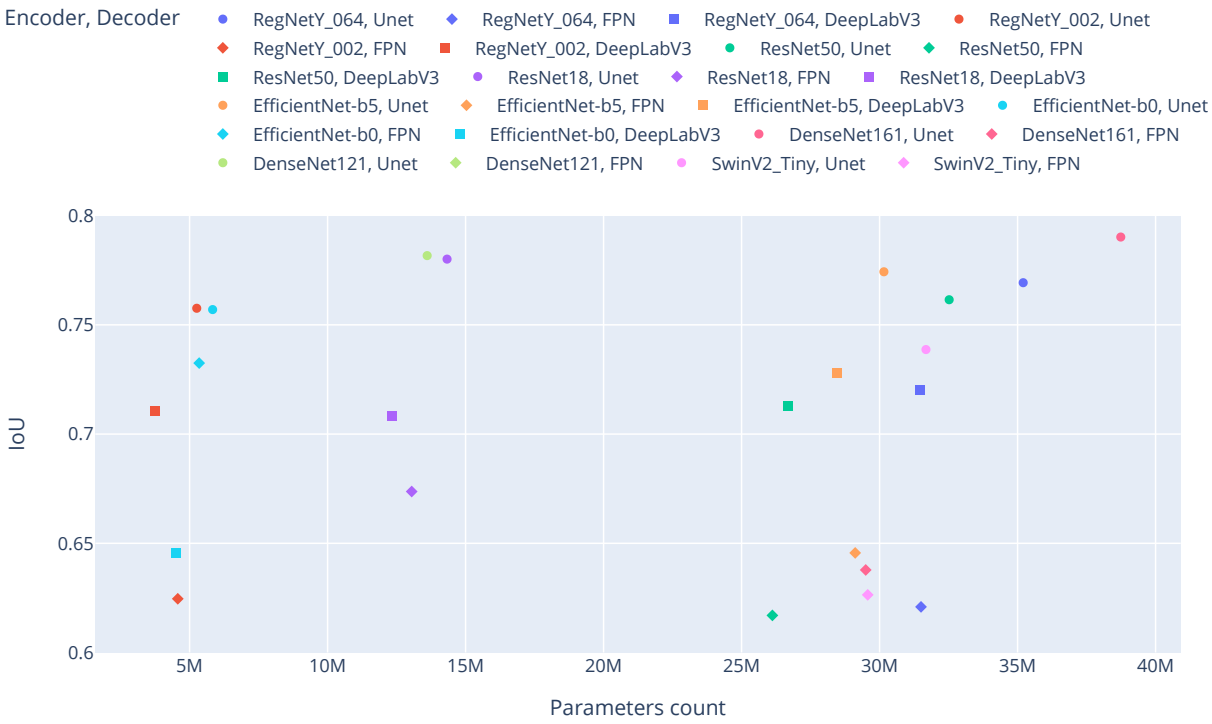
relação com o tempo necessário para o treinamento. Por exemplo, no gráfico presente na figura 15 é possível encontrar modelos que levaram mais de 200 minutos para completar as 1500 épocas de treinamento e outros modelos que levaram menos da metade desse tempo. Por isso é importante analisar o gráfico na figura 17, no qual é possível obter o valor do *IoU* atingido por cada modelo após apenas 30 minutos de treinamento.

Figura 17 – Resultados obtidos de *IoU* em cada um dos modelos após 30 minutos de treinamento.



Fonte: Próprio Autor

Figura 18 – Comparação dos valores finais de *IoU* obtidos e a quantidade de parâmetros presente em cada modelo.



Fonte: Próprio Autor

Figura 19 – Comparação dos valores finais de *clDice* obtidos e a quantidade de parâmetros presente em cada modelo.

Fonte: Próprio Autor

5.3 Desafios encontrados nas imagens

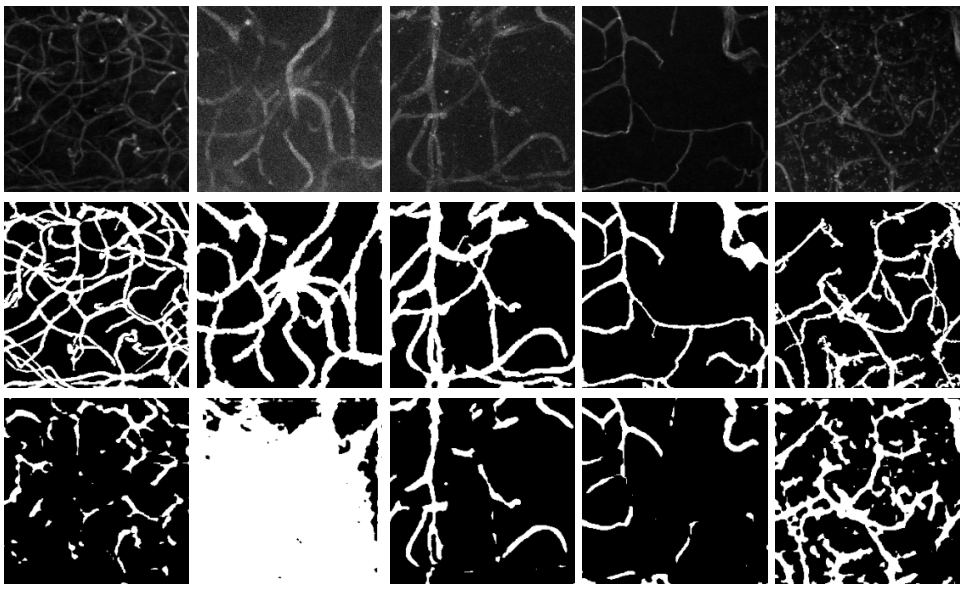
Durante o processo de teste foi possível observar que muitas vezes as mesmas imagens eram segmentadas com menor precisão por todos os modelos. Portanto, é interessante analisar as características dessas imagens. Na figura 20 são encontradas as imagens que obtiveram os piores valores de *IoU* na segmentação com a combinação *U-Net* + *ResNet18*. Para comparação, na figura 21 são mostrados os melhores resultados de segmentação para o mesmo modelo.

Para a análise é importante tentar encontrar características nas imagens que possam representar um desafio na tarefa de segmentação. Em todas as imagens segmentadas se destacam duas características que estão muitas vezes presentes nas imagens com piores resultados, sendo a espessura dos vasos e o contraste entre os vasos sanguíneos e o fundo da imagem.

5.3.1 Contraste entre vasos e fundo

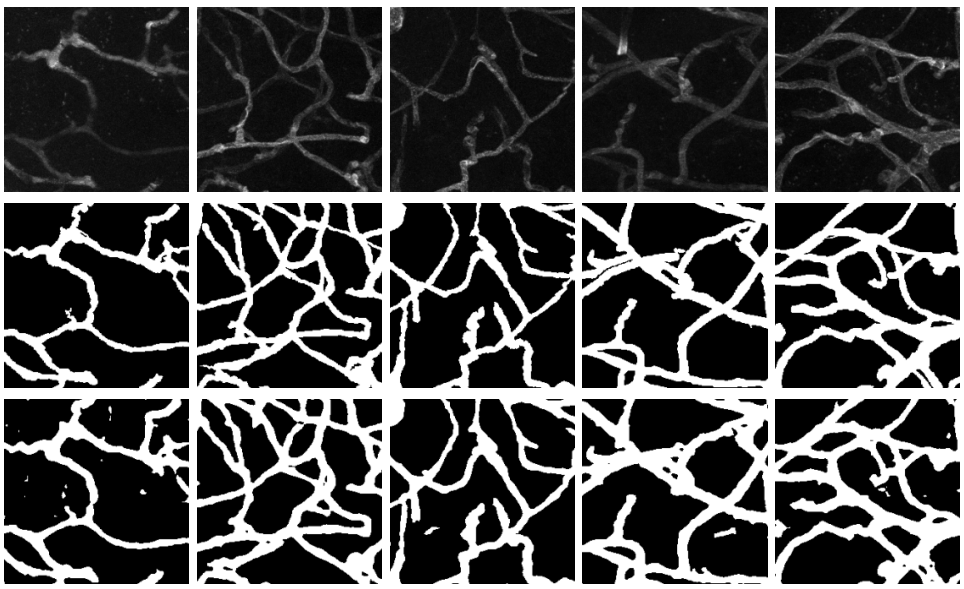
Observando os resultados, torna-se nítido que nas imagens mais desafiadoras o baixo contraste entre o fundo e os vasos sanguíneos é um ponto muito presente, e isso indica que o fundo mais claro nas imagens torna a tarefa muito mais difícil para as redes. Considerando que as imagens e as máscaras são monocromáticas, o contraste entre o fundo e o vaso representa grande parte da extração de características realizadas pelas redes. Logo, em imagens onde o

Figura 20 – Imagens que obtiveram os piores resultados utilizando *U-Net + ResNet18*. As imagens originais, de referência e de resultados da rede são mostradas, respectivamente, na primeira, segunda e terceira linha de figuras.



Fonte: Próprio Autor

Figura 21 – Imagens que obtiveram os melhores resultados utilizando *U-Net + ResNet18*. As imagens originais, de referência e de resultados da rede são mostradas, respectivamente, na primeira, segunda e terceira linha de figuras.



Fonte: Próprio Autor

fundo está quase tão claro quanto os vasos, a quantidade de informações obtidas inevitavelmente diminui e isso reflete nos valores calculados nos resultados de *clDice* e *IoU*.

5.3.2 Espessura e continuidade dos vasos

Outro ponto perceptível é a diferença entre as espessuras dos vasos nas figuras 20 e 21. Nas imagens com melhores resultados os vasos são mais espessos e também não possuem muitas variações drásticas nessa espessura. Já nos piores resultados pode-se ver imagens com vasos mais finos e também com variações mais bruscas de espessura, sendo isso um provável fator de dificuldade na segmentação das imagens.

Além da espessura, outro ponto importante a ser observado é a continuidade dos vasos encontrados nas imagens. Há uma diferença observada entre as imagens presentes nas figuras 21 e 20 no que se refere a encontrar vasos mais contínuos. Logo, a maior irregularidade dos vasos torna a tarefa de segmentação mais desafiadora.

Capítulo 6
CONCLUSÃO

Quando comparada com tarefas mais simples, como a classificação de imagens em conjuntos menores, a tarefa de segmentação de vasos sanguíneos nas imagens obtidas através de microscopia se mostrou razoavelmente desafiadora para algumas das combinações de redes testadas, especialmente nos modelos que utilizaram *FPN*. Logo, é de extrema importância a análise das dificuldades encontradas no processo de construção e treinamento das redes.

Esses desafios se devem a fatores como a quantidade baixa de imagens disponíveis, diminuindo a capacidade de treinamento das redes. Como essa é uma situação comum em imagens biomédicas, nos experimentos realizados foi utilizado um número pequeno de imagens para o conjunto de treinamento. Fatores adicionais incluem o baixo contraste e alto ruído presente em algumas imagens, diminuindo a separação visual entre vaso e fundo, e também a grande variação de espessura e não-continuidade que ocorre em muitos dos vasos sanguíneos analisados.

Ainda há também a dinâmica dos testes realizados, em que os hiperparâmetros foram encontrados com base em uma combinação específica de modelos e utilizados para todas as outras dada a limitação de tempo e poder computacional, visto que o processo de treinamento e validação durou algumas horas para cada uma das combinações. Portanto, houve uma decisão experimental de obter uma possível diminuição dos valores obtidos nos resultados pela capacidade de se incluir o maior número possível de redes neurais nos testes realizados.

Apesar de todas as dificuldades encontradas, ainda foi possível obter um desempenho satisfatório na maior parte das combinações escolhidas. Foi possível concluir que, nas condições impostas, os decodificadores *U-Net* e *DeepLabV3+* aliados à codificadores como *DenseNet*, *RegNet* e *ResNet* foram capazes de atingir resultados promissores nas métricas utilizadas para avaliação.

Através da extensa pesquisa e comparação dos diversos codificadores e decodificadores é possível atestar um bom desempenho especialmente das combinações utilizando a *U-Net*. Um dos pontos de maior interesse foi a comparação de modelos de tamanhos discrepantes, na qual modelos como a *DenseNet121* e *RegNetY* obtiveram grande destaque, pois os resultados obtidos indicam uma provável vantagem na implementação dos modelos menores visto que são capazes

de alcançar resultados semelhantes ou melhores com custos inferiores de tempo e recursos. Dessa maneira, os testes realizados indicam a não existência de uma correlação clara entre a quantidade de parâmetros dos modelos e dos valores atingidos nas métricas de acurácia e *clDice*.

6.1 Trabalhos Futuros

Devido às limitações presentes na construção desse trabalho há um número grande de possibilidades de se incrementar a pesquisa tanto na busca de melhores resultados quanto no incremento da complexidade do assunto abordado. Algumas dessas alternativas são discutidas nessa seção.

6.1.1 Inclusão de novas redes

Dado o rápido avanço da área, novas tecnologias surgem em uma frequência elevada. Logo, é interessante manter atualizada a busca por novos codificadores e decodificadores que possam ser mais eficientes ou atingir resultados melhores na tarefa proposta.

6.1.1.1 Testes com *transformers*

Foi testado nesse trabalho um único codificador *transformer SwinV2*, e dentro da possibilidade de se aumentar e atualizar a gama de redes incluídas nos testes, parece muito importante a inclusão de mais modelos pertencentes à classe dos *transformers* que tem ganhado notoriedade recentemente.

6.1.1.2 Modelos menores

Os menores modelos utilizados nos testes realizados nesse trabalho possuem em torno de quatro milhões de parâmetros e apesar de serem muito menores que modelos mais populares, como a *ResNet50* (mais de 20 milhões de parâmetros), ainda são relativamente maiores que os menores modelos disponíveis atualmente. Como citado previamente no capítulo 2, o artigo (GALDRAN et al., 2022) cita a utilização de modelos com menos de 100 mil parâmetros treináveis e é possível que a adição de modelos desse nível de profundidade possa ser um fator muito positivo para a abrangência das análises realizadas.

6.1.2 Métrica para análise da segmentação

Um dos grandes pontos de interesse que foram levantados nesse trabalho se refere às possibilidade de se medir matematicamente a taxa de sucesso na segmentação. Pode ser muito impactante o desenvolvimento de novas métricas que abordem esse tipo de tarefa computacional como por exemplo o *clDice*, desenvolvido em (SHIT et al., 2021).

6.1.3 Busca mais completa por hiperparâmetros

Por conta das limitações presentes no desenvolvimento, os hiperparâmetros utilizados foram escolhidos com base em uma rede e então fixados. É provável que, caso se busque encontrar os melhores hiperparâmetros para cada uma das combinações, os resultados finais obtidos possam ser superiores para alguns modelos.

REFERÊNCIAS

ABADI, M.; AGARWAL, A.; BARHAM, P.; BREVDO, E.; CHEN, Z.; CITRO, C.; CORRADO, G. S.; DAVIS, A.; DEAN, J.; DEVIN, M.; GHEMAWAT, S.; GOODFELLOW, I.; HARP, A.; IRVING, G.; ISARD, M.; JIA, Y.; JOZEFOWICZ, R.; KAISER, L.; KUDLUR, M.; LEVENBERG, J.; MANé, D.; MONGA, R.; MOORE, S.; MURRAY, D.; OLAH, C.; SCHUSTER, M.; SHLENS, J.; STEINER, B.; SUTSKEVER, I.; TALWAR, K.; TUCKER, P.; VANHOUCKE, V.; VASUDEVAN, V.; VIéGAS, F.; VINYALS, O.; WARDEN, P.; WATTENBERG, M.; WICKE, M.; YU, Y.; ZHENG, X. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Software available from tensorflow.org. Disponível em: <https://www.tensorflow.org/>. Citado na página 34.

ANSEL, J.; YANG, E.; HE, H.; GIMELSHEIN, N.; JAIN, A.; VOZNESENSKY, M.; BAO, B.; BELL, P.; BERARD, D.; BUROVSKI, E.; CHAUHAN, G.; CHOURDIA, A.; CONSTABLE, W.; DESMAISON, A.; DEVITO, Z.; ELLISON, E.; FENG, W.; GONG, J.; GSCHWIND, M.; HIRSH, B.; HUANG, S.; KALAMBARKAR, K.; KIRSCH, L.; LAZOS, M.; LEZCANO, M.; LIANG, Y.; LIANG, J.; LU, Y.; LUK, C.; MAHER, B.; PAN, Y.; PUHRSCH, C.; RESO, M.; SAROUFIM, M.; SIRAICHI, M. Y.; SUK, H.; SUO, M.; TILLET, P.; WANG, E.; WANG, X.; WEN, W.; ZHANG, S.; ZHAO, X.; ZHOU, K.; ZOU, R.; MATHEWS, A.; CHANAN, G.; WU, P.; CHINTALA, S. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In: *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM, 2024. Disponível em: <https://pytorch.org/assets/pytorch2-2.pdf>. Citado na página 31.

BADRINARAYANAN, V.; HANDA, A.; CIPOLLA, R. *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling*. 2015. Disponível em: <https://arxiv.org/abs/1505.07293>. Citado na página 16.

CHEN, L.-C.; PAPANDREOU, G.; SCHROFF, F.; ADAM, H. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. Citado na página 23.

CHEN, L.-C.; ZHU, Y.; PAPANDREOU, G.; SCHROFF, F.; ADAM, H. *Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation*. 2018. Disponível em: <https://arxiv.org/abs/1802.02611>. Citado 2 vezes nas páginas 23 e 24.

CIRESAN, D.; GIUSTI, A.; GAMBARDELLA, L.; SCHMIDHUBER, J. Deep neural networks segment neuronal membranes in electron microscopy images. *Advances in neural information processing systems*, v. 25, 2012. Citado na página 21.

COSTA, L. da F. *Further Generalizations of the Jaccard Index*. 2021. Disponível em: <https://arxiv.org/abs/2110.09619>. Citado na página 25.

DOSOVITSKIY, A.; BEYER, L.; KOLESNIKOV, A.; WEISSENBORN, D.; ZHAI, X.; UNTERTHINER, T.; DEHGHANI, M.; MINDERER, M.; HEIGOLD, G.; GELLY, S.; USZKOREIT,

J.; HOULSBY, N. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. Disponível em: <https://arxiv.org/abs/2010.11929>. Citado na página 21.

FRAZ, M.; REMAGNINO, P.; HOPPE, A.; UYYANONVARA, B.; RUDNICKA, A.; OWEN, C.; BARMAN, S. Blood vessel segmentation methodologies in retinal images – a survey. *Computer Methods and Programs in Biomedicine*, v. 108, n. 1, p. 407–433, 2012. ISSN 0169-2607. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0169260712000843>. Citado na página 28.

GALDRAN, A.; ANJOS, A.; DOLZ, J.; CHAKOR, H.; LOMBAERT, H.; AYED, I. B. State-of-the-art retinal vessel segmentation with minimalistic models. *Scientific Reports*, Nature Publishing Group UK London, v. 12, n. 1, p. 6174, 2022. Citado 3 vezes nas páginas 13, 29 e 44.

GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; MALIK, J. *Rich feature hierarchies for accurate object detection and semantic segmentation*. 2014. Disponível em: <https://arxiv.org/abs/1311.2524>. Citado na página 27.

GOLDBLUM, M.; SOURI, H.; NI, R.; SHU, M.; PRABHU, V.; SOMEPALLI, G.; CHATTOPADHYAY, P.; IBRAHIM, M.; BARDES, A.; HOFFMAN, J. et al. Battle of the backbones: A large-scale comparison of pretrained models across computer vision tasks. *Advances in Neural Information Processing Systems*, v. 36, 2024. Citado na página 29.

GU, J.; WANG, Z.; KUEN, J.; MA, L.; SHAHROUDY, A.; SHUAI, B.; LIU, T.; WANG, X.; WANG, L.; WANG, G.; CAI, J.; CHEN, T. *Recent Advances in Convolutional Neural Networks*. 2017. Disponível em: <https://arxiv.org/abs/1512.07108>. Citado na página 17.

HAO, S.; ZHOU, Y.; GUO, Y. A brief survey on semantic segmentation with deep learning. *Neurocomputing*, Elsevier, v. 406, p. 302–321, 2020. Citado na página 15.

HAQUE, I. R. I.; NEUBERT, J. Deep learning approaches to biomedical image segmentation. *Informatics in Medicine Unlocked*, Elsevier, v. 18, p. 100297, 2020. Citado 2 vezes nas páginas 12 e 29.

HE, K.; GKIOXARI, G.; DOLLáR, P.; GIRSHICK, R. *Mask R-CNN*. 2018. Disponível em: <https://arxiv.org/abs/1703.06870>. Citado na página 27.

HE, K.; ZHANG, X.; REN, S.; SUN, J. *Deep Residual Learning for Image Recognition*. 2015. Disponível em: <https://arxiv.org/abs/1512.03385>. Citado na página 18.

HOOVER, A.; KOUZNETSOVA, V.; GOLDBAUM, M. Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *IEEE Transactions on Medical Imaging*, v. 19, n. 3, p. 203–210, 2000. Citado na página 28.

HUANG, G.; LIU, Z.; MAATEN, L. V. D.; WEINBERGER, K. Q. Densely connected convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 4700–4708. Citado na página 20.

IAKUBOVSKII, P. *Segmentation Models Pytorch*. [S.l.]: GitHub, 2019. <https://github.com/qubvel/segmentation_models.pytorch>. Citado na página 31.

INC., P. T. *Collaborative data science*. Montreal, QC: Plotly Technologies Inc., 2015. Disponível em: <https://plot.ly>. Citado na página 34.

KOVÁCS, G.; FAZEKAS, A. A new baseline for retinal vessel segmentation: Numerical identification and correction of methodological inconsistencies affecting 100+ papers. *Medical Image Analysis*, Elsevier, v. 75, p. 102300, 2022. Citado na página 28.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *nature*, Nature Publishing Group UK London, v. 521, n. 7553, p. 436–444, 2015. Citado 2 vezes nas páginas 12 e 15.

LI, L.; DOROSLOVAčKI, M.; LOEW, M. H. Approximating the gradient of cross-entropy loss function. *IEEE Access*, v. 8, p. 111626–111635, 2020. Citado na página 25.

LI, X.; DING, H.; YUAN, H.; ZHANG, W.; PANG, J.; CHENG, G.; CHEN, K.; LIU, Z.; LOY, C. C. *Transformer-Based Visual Segmentation: A Survey*. 2024. Disponível em: <https://arxiv.org/abs/2304.09854>. Citado na página 28.

LI, Z.; YANG, W.; PENG, S.; LIU, F. *A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects*. 2020. Disponível em: <https://arxiv.org/abs/2004.02806>. Citado na página 17.

LIN, T.-Y.; DOLLÁR, P.; GIRSHICK, R.; HE, K.; HARIHARAN, B.; BELONGIE, S. Feature pyramid networks for object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 2117–2125. Citado na página 23.

LIU, Z.; HU, H.; LIN, Y.; YAO, Z.; XIE, Z.; WEI, Y.; NING, J.; CAO, Y.; ZHANG, Z.; DONG, L.; WEI, F.; GUO, B. *Swin Transformer V2: Scaling Up Capacity and Resolution*. 2022. Disponível em: <https://arxiv.org/abs/2111.09883>. Citado na página 21.

MAHESH, B. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, v. 9, n. 1, p. 381–386, 2020. Citado na página 15.

MOCCIA, S.; De Momi, E.; El Hadji, S.; MATTOS, L. S. Blood vessel segmentation algorithms — review of methods, datasets and evaluation metrics. *Computer Methods and Programs in Biomedicine*, v. 158, p. 71–91, 2018. ISSN 0169-2607. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0169260717313421>. Citado 2 vezes nas páginas 12 e 29.

NGUYEN, U. T.; BHUIYAN, A.; PARK, L. A.; RAMAMOHANARAO, K. An effective retinal blood vessel segmentation method using multi-scale line detection. *Pattern Recognition*, v. 46, n. 3, p. 703–715, 2013. ISSN 0031-3203. Disponível em: <https://www.sciencedirect.com/science/article/pii/S003132031200355X>. Citado na página 27.

O'SHEA, K.; NASH, R. *An Introduction to Convolutional Neural Networks*. 2015. Disponível em: <https://arxiv.org/abs/1511.08458>. Citado 2 vezes nas páginas 17 e 18.

RADOSAVOVIC, I.; KOSARAJU, R. P.; GIRSHICK, R.; HE, K.; DOLLáR, P. *Designing Network Design Spaces*. 2020. Disponível em: <https://arxiv.org/abs/2003.13678>. Citado na página 19.

RICCI, E.; PERFETTI, R. Retinal blood vessel segmentation using line operators and support vector classification. *IEEE Transactions on Medical Imaging*, v. 26, n. 10, p. 1357–1365, 2007. Citado na página 27.

RONNEBERGER, O.; FISCHER, P.; BROX, T. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. Disponível em: <https://arxiv.org/abs/1505.04597>. Citado na página 22.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *nature*, Nature Publishing Group UK London, v. 323, n. 6088, p. 533–536, 1986. Citado na página 17.

RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATHY, A.; KHOSLA, A.; BERNSTEIN, M.; BERG, A. C.; FEI-FEI, L. *ImageNet Large Scale Visual Recognition Challenge*. 2015. Disponível em: <https://arxiv.org/abs/1409.0575>. Citado na página 31.

SHINDE, P. P.; SHAH, S. A review of machine learning and deep learning applications. In: IEEE. *2018 Fourth international conference on computing communication control and automation (ICCUBEA)*. [S.l.], 2018. p. 1–6. Citado na página 15.

SHIT, S.; PAETZOLD, J. C.; SEKUBOYINA, A.; EZHOV, I.; UNGER, A.; ZHYLKA, A.; PLUIM, J. P. W.; BAUER, U.; MENZE, B. H. cldice - a novel topology-preserving loss function for tubular structure segmentation. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021. Disponível em: <http://dx.doi.org/10.1109/CVPR46437.2021.01629>. Citado 2 vezes nas páginas 25 e 44.

SILVA, M. V. da; SANTOS, N. de C.; OUELLETTE, J.; LACOSTE, B.; COMIN, C. H. *A new dataset for measuring the performance of blood vessel segmentation methods under distribution shifts*. 2024. Citado na página 30.

STAAL, J.; ABRAMOFF, M.; NIEMEIJER, M.; VIERGEVER, M.; GINNEKEN, B. van. Ridge-based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging*, v. 23, n. 4, p. 501–509, 2004. Citado na página 28.

SULTANA, F.; SUFIAN, A.; DUTTA, P. Evolution of image segmentation using deep convolutional neural network: A survey. *Knowledge-Based Systems*, Elsevier, v. 201, p. 106062, 2020. Citado 2 vezes nas páginas 12 e 29.

TAN, M.; LE, Q. V. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. 2020. Disponível em: <https://arxiv.org/abs/1905.11946>. Citado 2 vezes nas páginas 19 e 20.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, L.; POLOSUKHIN, I. *Attention Is All You Need*. 2023. Disponível em: <https://arxiv.org/abs/1706.03762>. Citado na página 20.

XU, N.; PRICE, B.; COHEN, S.; YANG, J.; HUANG, T. *Deep Interactive Object Selection*. 2016. Disponível em: <https://arxiv.org/abs/1603.04042>. Citado na página 27.

ZEILER, M. D.; FERGUS, R. *Visualizing and Understanding Convolutional Networks*. 2013. Disponível em: <https://arxiv.org/abs/1311.2901>. Citado na página 17.