Universidade Federal do Maranhão
Centro de Ciências Exatas e Tecnologias - CCET
Engenharia da Computação

Idemilson dos Anjos Silva

Segmentação de Imagens Hepáticas Utilizando Redes Neurais Convolucionais com Arquitetura UNet

São Luís - MA

2025

Idemilson dos Anjos Silva

Segmentação de Imagens Hepáticas Utilizando Redes Neurais Convolucionais com Arquitetura UNet

Monografia apresentada ao curso de Engenharia da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de bacharel em Engenharia da Computação.

Orientador: Prof. Dr. Bruno Feres de Souza

São Luís - MA

2025

Idemilson dos Anjos Silva

Segmentação de Imagens Hepáticas Utilizando Redes Neurais Convolucionais com Arquitetura UNet

Monografia apresentada ao curso de Engenharia da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de bacharel em Engenharia da Computação.

Aprovado em ___/____/____

_____

Prof. Dr. Bruno Feres de Souza
UFMA - Orientador

_____

Prof. Dr. Paulo Rogério de Almeida Ribeiro
UFMA - Examinador

_____

Prof. Dr. Sergio Souza Costa
UFMA - Examinador

São Luís - MA

2025

Agradecimentos

A conclusão deste trabalho representa não apenas o encerramento de uma etapa acadêmica, mas também o reflexo do apoio de muitas pessoas que, direta ou indiretamente, contribuíram para essa jornada.

Em primeiro lugar, expresso minha profunda gratidão ao Professor Doutor Bruno Feres de Souza, cuja orientação foi essencial para o desenvolvimento deste trabalho. Sua orientação, conhecimento e disponibilidade foram fundamentais para a construção deste trabalho.

Agradeço também a todos os professores do corpo docente da Engenharia da Computação, que, ao longo dessa trajetória, compartilharam seus ensinamentos e experiências, ampliando minha visão e preparando-me para os desafios da área.

À minha companheira, Déborah, que sempre esteve ao meu lado, oferecendo apoio, paciência e compreensão durante todo esse percurso. Seu incentivo foi fundamental nos momentos de desafio e cansaço.

Aos meus amigos e familiares, que sempre me apoiaram, seja com palavras de encorajamento ou simplesmente estando presentes.

A todos vocês, meu mais sincero agradecimento. Sem o apoio de cada um, essa conquista não teria sido possível.

"Se você quer algo que nunca teve, precisa estar disposto a fazer algo que nunca fez."

(*Thomas Jefferson)*

Segmentação de Imagens Hepáticas Utilizando Redes Neurais Convolucionais com Arquitetura UNet

Resumo: Segmentar uma imagem hepática é um grande desafio no que se refere ao processamento de imagens médicas, particularmente o objetivo é melhorar as técnicas automatizadas de análise. Portanto, este trabalho visa propor a segmentação hepática automática utilizando redes neurais convolucionais (CNNs) com arquitetura U-Net. Que abrangerá desde a obtenção, preparação do conjunto de dados até a otimização de hiperparâmetros e avaliação do modelo. Foi utilizado um conjunto de dados públicos contendo imagens hepáticas e suas respectivas máscaras binárias, aplicando técnicas de pré-processamento, como normalização e redimensionamento. A abordagem usando a arquitetura U-Net foi desenvolvida e treinada no Google Colab Pro juntamente com a otimização dos hiperparâmetros com Optuna. Os resultados demonstraram um notável desempenho com Dice Coefficient de 93% no conjunto de validação e 91% no conjunto de teste, demonstrando uma alta precisão na segmentação. A análise qualitativa também demonstrou a capacidade do modelo em segmentar regiões hepáticas com alta credibilidade. Assim, o estudo reforça a viabilidade da U-Net como uma ferramenta eficaz na segmentação de imagens médicas, contribuindo para a melhoria da análise clínica e avanços na medicina computacional.

Palavras chave: segmentação de imagens; redes neurais convolucionais; U-Net; imagens médicas.

Hepatic Image Segmentation Using Convolutional Neural Networks with U-Net Architecture

Abstract: Segmenting hepatic images is a major challenge in medical image processing, particularly when aiming to improve automated analysis techniques. Therefore, this work proposes an automatic hepatic segmentation approach using convolutional neural networks (CNNs) with U-Net architecture. The study encompasses data acquisition, preprocessing, hyperparameter optimization, and model evaluation. A public dataset containing hepatic images and their respective binary masks was used, applying preprocessing techniques such as normalization and data augmentation. The U-Net-based approach was developed and trained on Google Colab Pro, with hyperparameter optimization conducted using Optuna. The results demonstrated excellent performance, achieving a Dice Coefficient of 93% on the validation set and 91% on the test set, indicating high segmentation accuracy. Qualitative analysis also confirmed the model's ability to segment hepatic regions with high reliability. Thus, this study reinforces the feasibility of U-Net as an effective tool for medical image segmentation, contributing to improvements in clinical analysis and advancements in computational medicine.

Keywords: image segmentation; convolutional neural networks; U-Net; medical images.

Lista de Figuras

Lista de Tabelas

SUMÁRIO

1 INTRODUÇÃO

A segmentação de imagens médicas é uma área de pesquisa que tem ganhado destaque nos últimos anos, especialmente com o avanço das técnicas de inteligência artificial. A capacidade de identificar e isolar regiões de interesse em imagens médicas, como órgãos e tecidos, é crucial para diagnósticos precisos e tratamentos personalizados. Segundo Bilic et al. (2023), a automação desse processo por meio de redes neurais convolucionais (CNNs) tem revolucionado a área, permitindo uma segmentação mais rápida e precisa do que os métodos tradicionais.

A visão computacional é um campo da inteligência artificial, capaz de dar às máquinas a habilidade de “enxergar” e interpretar imagens e vídeos. Essa tecnologia desempenha um papel importante em diversas aplicações, como segurança, automação industrial e, principalmente, na área da saúde. De acordo com Szeliski (2022), os avanços recentes em redes neurais profundas melhoram significativamente a capacidade de sistemas de visão computacional na análise de imagens. Na área da saúde, Litjens et al. (2017) destacam que a visão computacional tem sido muito aplicada na análise de exames médicos, permitindo diagnósticos mais rápidos e precisos. Técnicas baseadas em aprendizado profundo, como as Redes Neurais Convolucionais (CNNs), são amplamente utilizadas para segmentação de imagens, facilitando a detecção de órgãos e possíveis anomalias com maior eficiência (RONNEBERGER et al., 2015).

No entanto, a segmentação de imagens médicas ainda enfrenta desafios significativos. Como destacado por kayalibay et al. (2022), a obtenção de grandes volumes de dados rotulados para treinamento é muitas vezes difícil e custosa. Além disso, a qualidade das imagens pode variar dependendo do equipamento utilizado e das condições de aquisição, o que pode comprometer o desempenho dos modelos. Esses desafios mostram a necessidade de avanços contínuos no desenvolvimento de algoritmos mais robustos e adaptáveis.

A arquitetura U-Net, proposta por Ronneberger et al. (2015), tem se destacado como uma das principais ferramentas para a segmentação de imagens médicas. Essa arquitetura, caracterizada por sua estrutura em forma de U, combina um caminho de contração (encoder) para extrair características da imagem e um caminho de expansão (decoder) para reconstruir a segmentação. A U-Net é particularmente eficaz em tarefas de segmentação biomédica devido às suas conexões de salto, que como observado por Wang et al. (2022), essas conexões combinam características de alto e baixo nível fazendo com que elas não sejam perdidas durante o processo de segmentação, o que é crucial para aplicações médicas.

Neste contexto, este trabalho propõe a utilização da arquitetura U-Net para segmentar imagens hepáticas. O objetivo deste estudo é desenvolver e avaliar o modelo U-Net com intuito de contribuir para a melhoria das técnicas automatizadas de segmentação de imagens médicas.

1.1 Justificativa

Este trabalho de conclusão de curso tem como principal motivação aplicar os conhecimentos aprendidos ao longo da graduação em Engenharia da Computação, em especial nas áreas de processamento de imagens e inteligência artificial para resolver problemas reais no campo da saúde: a segmentação de imagens hepáticas usando redes neurais convolucionais com arquitetura U-Net. A escolha deste tema se deu devido à crescente demanda por soluções inovadoras usando tecnologia para otimizar o tempo e melhorar a precisão na análise de imagens médicas.

1.2 Objetivos

1.2.1 Geral

Desenvolver e avaliar um modelo de segmentação de imagens médicas utilizando Redes Neurais Convolucionais (CNN) de arquitetura U-Net para segmentar o fígado em imagens médicas.

1.2.2 Específicos

- Pesquisar na literatura conceitos teóricos sobre redes neurais convolucionais e a arquitetura U-Net.

- Preparar e organizar o dataset das imagens e suas respectivas máscaras, aplicando técnicas de pré-processamento, como normalização e redimensionamento.
- Implementar o modelo de segmentação baseado na arquitetura U-Net e realizar o treinamento.
- Avaliar o modelo, usando métricas utilizadas na literatura.

2 REFERENCIAL TEÓRICO

2.1 Segmentação de Imagens Médicas

2.1.1 Importância na Área médica

Segundo Bilic et al. (2023), a segmentação de imagens usando redes neurais convolucionais (CNNs) são amplamente utilizadas em imagens biomédicas, onde informações de interesses são analisadas e interpretadas a partir de imagens complexas, como ressonâncias magnéticas e tomografias computadorizadas. Esse processo envolve identificar, isolar e analisar detalhadamente as estruturas anatômicas ou regiões de interesse em imagens médicas. A utilização dessas ferramentas baseadas em redes neurais convolucionais (CNNs) trouxe evolução na área, permitindo uma segmentação automatizada e precisa. Conforme destacado por Kayalibay et al. (2017), as CNNs são eficazes na aprendizagem de representações complexas de características, substituindo os métodos tradicionais que dependiam de segmentação manual, muitas vezes passível a erros humanos. Essa automatização não apenas economiza tempo, mas também melhora significativamente os resultados.

A utilização de redes neurais profundas, como as arquiteturas U-Net, possibilitou avanços para a segmentação de imagens. Essas arquiteturas permitem uma análise detalhada de estruturas anatômicas em alta resolução, como apontado por Wang et al. (2022), que fala sobre a ampla aplicação dessas técnicas no diagnóstico assistido por computador e na medicina de precisão. Por outro lado, as redes neurais também enfrentam grandes desafios. Entre eles, está a necessidade de grandes volumes de dados rotulados para o treinamento (Kayalibay et al, 2017), o que nem sempre é fácil de obter. Além disso, Addimulam e Mohammed (2020) apontam que a generalização dos modelos treinados ainda é um desafio, já que variações na qualidade das imagens e nos métodos de aquisição podem

comprometer o desempenho das redes neurais. Esses desafios mostram o quanto é importante avançar no desenvolvimento de algoritmos mais robustos. Fazendo assim, a utilização dessas tecnologias na aplicação em diferentes contextos médicos.

Um aspecto fundamental a ser destacado é como as técnicas de segmentação têm impulsionado os avanços da medicina personalizada. Segundo Addimulam e Mohammed (2020), a segmentação que utiliza aprendizado profundo possibilita a identificação de variações individuais em estruturas anatômicas e nos padrões patológicos, permitindo, assim, a criação de planos de tratamento sob medida para cada paciente. Essa abordagem não apenas melhora os desfechos clínicos, mas também contribui para a redução de custos relacionados a tratamentos genéricos ou inadequados. Além disso, a combinação de transformadores com redes neurais convolucionais (CNNs), como proposto por Yuan et al. (2023), apresenta um enorme potencial para ampliar essas capacidades. Essa integração pode tornar as segmentações ainda mais precisas e adaptáveis, especialmente em imagens tridimensionais que envolvem múltiplos órgãos.

Na perspectiva computacional, a segmentação de imagens médicas é uma área interessante que une ciência da computação e a medicina. Para os especialistas em computação, criar algoritmos robustos para segmentação é uma chance de contribuir diretamente para a saúde. Estudos como o de Asgari Taghanaki et al. (2021) mostram como é essencial tratar a segmentação como um problema interdisciplinar, onde métodos avançados, como redes neurais profundas e transformadores, precisam ser ajustados às necessidades do ambiente médico. Apesar de a segmentação automática já ter mostrado grande potencial em diversas aplicações, ainda há muito trabalho para torná-la acessível e utilizável em grande escala, principalmente em regiões com menos recursos. Por isso, pesquisas nessa área são fundamentais para ampliar os impactos positivos dessa tecnologia na saúde.

2.1.2 Métodos Tradicionais de Segmentação

A segmentação de imagens é uma etapa do processamento digital de imagens, onde divide uma imagem em partes ou regiões distintas, facilitando sua

análise. Entre os métodos tradicionais de segmentação, destacam-se técnicas baseadas em limiarização, detecção de bordas e segmentação por regiões, cada uma com vantagens e limitações específicas (JASIM; MOHAMMED, 2021).

2.1.2.1 Limiarização

A limiarização é uma das técnicas mais simples e muito utilizadas para segmentação de imagens. Essa técnica baseia-se em separar pixels de diferentes regiões da imagem por meio de um limiar T. Essa técnica é muito útil em imagens binárias ou quando existe uma clara diferença de intensidade entre o valor do objeto a ser segmentado e o valor do fundo da imagem (ABDULATEEF;SALMAN, 2021). A fórmula matemática básica da limiarização é dada por:

$$g(x,y) = \begin{cases} 1, & \text{se } f(x,y) \geq T \\ 0, & \text{se } f(x,y) < T \end{cases}$$

Onde:

$f(x,y)$ é a intensidade do pixel na posição (x,y),

T é o valor do limiar,

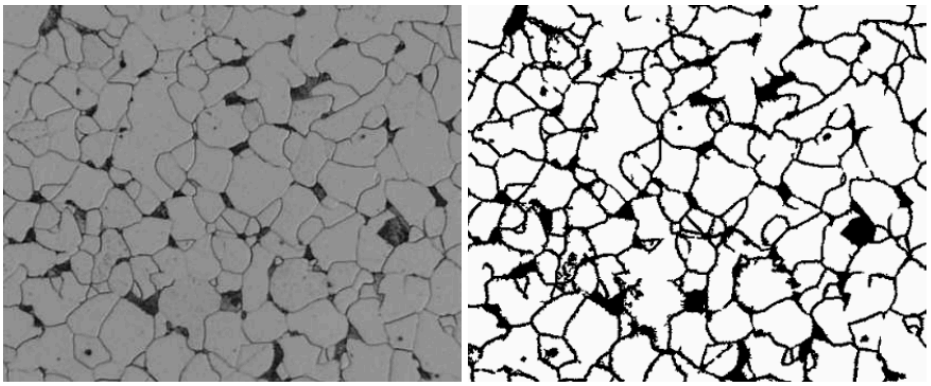
$g(x,y)$ é a imagem segmentada.

Uma das desvantagens dessa técnica é a necessidade de determinar inicialmente o valor de intensidade (Limiar). Porém métodos como de Otsu são muito utilizados por sua capacidade de determinar de forma automática o melhor limiar para uma imagem específica. No entanto, a limiarização apresenta limitações em imagens como iluminação não uniforme ou ruído, onde o contraste entre regiões não é claramente definido (SHARMA; SUJI, 2016).

Para superar essas limitações, surgiram variantes adaptativas que possibilitam o cálculo de limiares locais, em vez de depender de um único valor global. Jasim e Mohammed (2021) ressaltam que, quando aplicada a imagens médicas, essa abordagem pode melhorar de forma significativa a detecção de lesões, especialmente quando combinada com técnicas de pré-processamento voltadas para corrigir problemas de iluminação. No entanto, a limiarização adaptativa

ainda apresenta dificuldades em situações onde as texturas são mais complexas ou os gradientes de intensidade se mostram sutis, o que limita sua aplicação em determinados cenários.

Figura 1 - (a) Imagem de uma superfície. (b) Imagem da superfície limiarizada.



(a) (b)

Fonte: Roncero (2005, p.19).

2.1.2.2 Detecção de bordas

A segmentação baseada na detecção de borda funciona a partir do princípio de que as bordas de um objeto em uma imagem correspondem a mudanças bruscas na intensidade dos pixels. Essas bordas podem ser identificadas analisando as variações de níveis de intensidade, por meio de operações matemáticas que calculam o gradiente de intensidade da imagem. O gradiente, representado por ∇f, é a ferramenta poderosa para localizar essas descontinuidades e determinar a direção da borda em um ponto específico (x,y) de uma imagem f(Gonzalez & Woods, 2010).

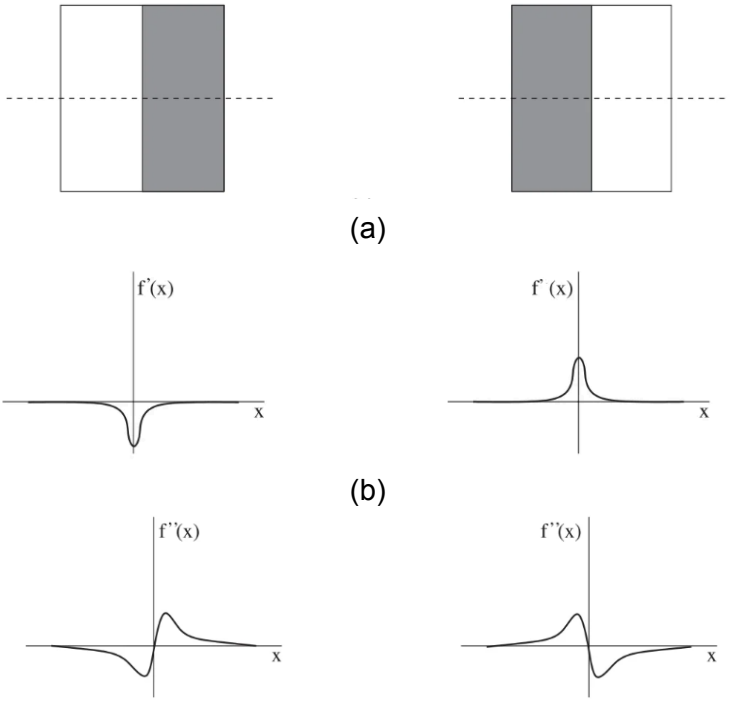
$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

O vetor gradiente de uma função f em um ponto (x,y) aponta a direção onde ocorre a maior variação de intensidade naquele ponto. Já a magnitude do gradiente, calculada como $M(x,y) = \sqrt{g_x^2 + g_y^2}$, indica o valor dessa variação na direção do

gradiente. Tanto g_x , g_y quanto $M(x, y)$ são imagens do mesmo tamanho da imagem original, sendo calculados para cada posições de pixels de f.

Como mostra a figura 2, as derivadas da função que representa os níveis de cinza ao longo de uma linha na imagem são utilizadas para detectar bordas. A primeira derivada identifica a presença de uma borda: ela é positiva ao passar de escuro para claro, e negativa no sentido contrário. Por outro lado, a segunda derivada indica o ponto exato da borda, marcada pelo cruzamento de zero, ou seja, onde ocorre mudança de sinal nos níveis de cinza.

Figura 2 - (a) imagens contendo uma região clara e uma escura; (b) derivada primeira do perfil de níveis de cinza; (c) derivada segunda do perfil de níveis de cinza.



Fonte: (Pedrini;Schwartz,2017).

2.1.2.3 Segmentação por Região

As técnicas de segmentação baseadas em regiões são muito conhecidas na área do processamento de imagens pela sua eficiência em identificar e dividir áreas homogêneas em uma imagem. Essas técnicas se baseiam na ideia de que regiões contínuas de uma imagem compartilham características semelhantes, como intensidade, textura ou cor, permitindo que sejam agrupadas em segmentos distintos. De acordo com Jasim e Mohammed(2021), essa técnica é especialmente útil para identificar objetos específicos em imagens mais complexas, aproveitando a

homogeneidade dos pixels para formar agrupamentos. Contudo, sua precisão pode ser comprometida por fatores de ruídos na imagem ou falta de contraste adequado entre regiões adjacentes.

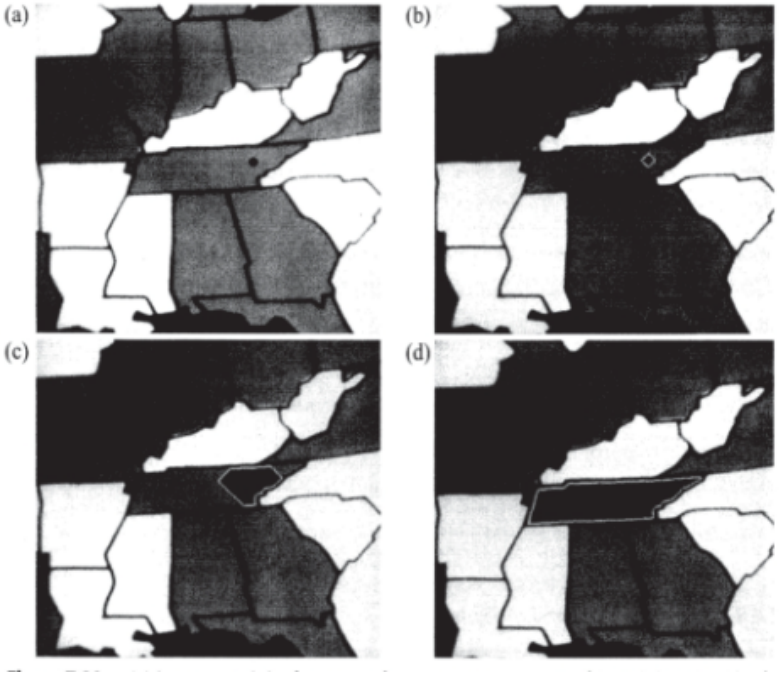
Essas técnicas podem ser classificadas em abordagens como crescimento de região, divisão e fusão de regiões. O crescimento de região, por exemplo, começa com a seleção de um ou mais pixel inicialmente, chamado de pixel semente, e a partir deles o algoritmo adiciona pixel vizinhos que atendam o critério de similaridade. Song e Yan (2017) apontam que, embora seja uma técnica intuitiva e eficiente para imagens com áreas bem delimitadas, ela pode ser influenciada pela escolha da semente e pelos parâmetros definidos para avaliar a similaridade. De outro modo, as técnicas de divisão e fusão seguem uma abordagem mais estruturada, dividindo a imagem em partes menores e depois reunindo essas partes com base na uniformidade dos dados.

Apesar disso tudo, essas técnicas apresentam desafios. Kaur e Kaur (2014) observam que essas técnicas podem demandar muito poder computacional em imagens de alta resolução devido ao volume de comparação de pixels necessários. Além disso, ruídos ou imperfeições nas imagens podem causar segmentações imprecisas, criando regiões que não representam com precisão os objetos reais.

A aplicação dessas técnicas se estende a diferentes áreas, incluindo análise médica, sensoriamento remoto e visão computacional. Em estudos de Jaglan et al. (2019), foi observado que, na segmentação de tumores em imagens de ressonância magnética, a identificação de áreas homogêneas desempenha um papel essencial em diagnósticos e no planejamento de intervenções médicas. A escolha da técnica de segmentação deve considerar as características específicas da imagem e o objetivo da análise para obter os melhores resultados.

Como exemplo, a figura 3 mostra uma imagem de uma seção de mapa contendo a escolha do ponto da semente e a evolução da técnica na região.

Figura 3 - (a) Imagem original mostrando um ponto semente; (b) primeiro estágio de crescimento de uma região; (c) estágio intermediário de crescimento de uma região; (d) região final.



Fonte: (Gonzalez & Woods, 2010).

2.2 Redes Neurais Convolucionais (CNNs)

As Redes Neurais convolucionais são um tipo de redes neurais muito usadas no processamento e análise de dados que possuem estruturas em grade, como imagens. Na área da visão computacional ela é muito utilizada na classificação de imagem, detecção de objetos e até mesmo na segmentação de imagens médicas.

A principal diferença das CNNs para as redes neurais tradicionais é a utilização de convoluções, que é uma operação matemática que extraem e processam características importantes das imagens, mantendo a relação espacial entre os pixels (Malhotra et al., 2022).

2.2.1 Estrutura de uma CNN

A estrutura básica de uma CNN é composta por três camadas principais: camada convolucional, camada de pooling e camada totalmente conectada, cada uma dessas camadas tem um papel específico na extração e aprendizagem de características das imagens (Malhotra et al., 2022). A camada convolução utiliza filtros para extrair características das imagens de entrada. A camada de pooling é

responsável por reduzir as dimensões espaciais da imagem, com intuito de reduzir a complexidade computacional e aumentar a capacidade de reconhecer padrões. Já a camada totalmente conectada é responsável por conectar as características das camadas anteriores.

2.2.1.1 Camada convolucional:

Essa camada, como mencionado anteriormente, usa um filtro para extrair características das imagens, como bordas, texturas e formas. O filtro é uma matriz de tamanho pequeno que é aplicado na imagem de entrada utilizando a convolução, ou seja, a matriz filtro é multiplicada elemento por elementos com a matriz de entrada e depois o resultados da multiplicação são somados, gerando um único valor.

Esse filtro vai deslocando-se pela matriz da imagem de entrada até passar por toda a imagem, o valor somado por cada passagem do filtro da imagem de entrada é inserido no mapa de características, como podemos ver na figura 4.

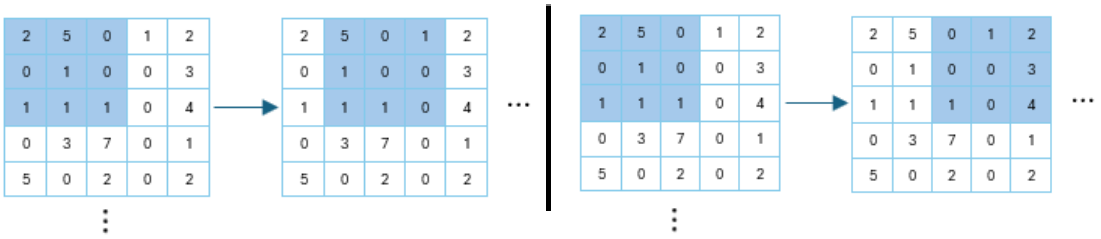
Figura 4 - demonstração de uma matriz de filtro convoluído com a matriz da imagem de entrada, criando um mapa de características.

. Fonte: Autor (2025).

Para deslizar sobre a imagem de entrada, o filtro utiliza o passo, também conhecido como stride, o termo refere-se ao quanto o filtro avança sobre a imagem, fazendo com que percorra toda a imagem e altere de tamanho na saída em relação a imagem de entrada. Um stride de 1 indica que o filtro avança um pixel por vez, enquanto um passo de 2 faz com que ele avance de 2 pixels, e assim por diante,

esse avanço ocorre tanto vertical quanto horizontal, na figura 5 ilustra o movimento horizontal.

Figura 5 - Um filtro de 3x3 com passo 1(esquerda) e com passo 2 (direita) aplicado em uma matriz de imagem de entrada 5x5.



Fonte: Autor (2025).

2.2.1.2 Camada de agrupamento:

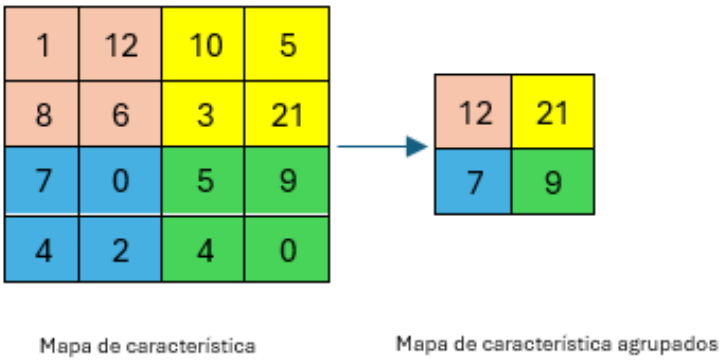
A camada de agrupamento é usada para simplificar as informações dos mapas de características, criadas na camada de convolução. Do mesmo modo que a camada de convolução, a camada de agrupamento utiliza de filtros que “escorrega” por toda a entrada (mapa de características), para fazer a sumarização dos dados e extrair as informações mais importantes, ajudando assim na generalização de rede. A grande diferença é que na camada de agrupamento não é aplicado nenhum filtro aos dados de entrada, mas sim simplificando as informações com uma operação matemática.

Normalmente, utilizam-se dois tipos principais de pooling: o máximo e o médio. Contudo, é bom destacar que existem outras técnicas. Cada uma delas com objetivos distintos, as quais serão discutidas abaixo.

- Max-pooling

O método mais utilizado é o max-pooling, no qual apenas o recurso mais importante (maior valor) dentro de cada janela do filtro. Isso cria um mapa de características agrupadas que será usado como entrada na próxima camada da rede.

Figura 6 - Exemplo do funcionamento do max-pooling com filtro 2X2 em uma imagem 4X4.



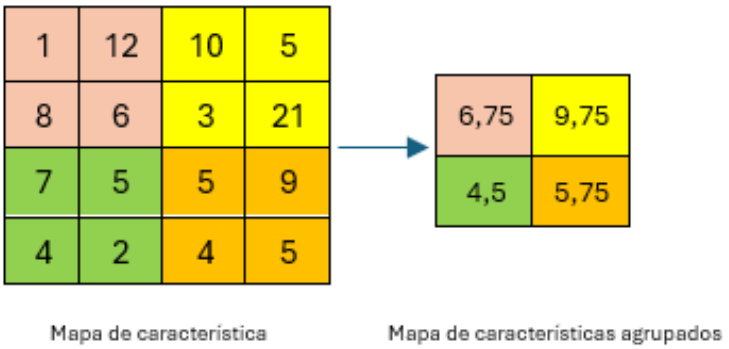
Fonte: Adaptado de Rodrigues, 2018.

Por exemplo, em uma imagem de alta resolução de um rosto humano, o max pooling pode identificar traços importantes como olhos, nariz e boca, dando ênfase a essas regiões enquanto descarta detalhes menos significativos. Fazendo isso, ele permite que as camadas seguintes da rede se concentrem nos aspectos mais importantes para tarefas como reconhecimento ou classificação (Ajmal et al., 2018).

- Average-pooling

Embora o max-pooling seja mais utilizado, o average-pooling funciona de forma semelhante, mas com uma diferença: ao invés de utilizar o valor máximo, ele calcula o valor médio. Isso resulta em um mapa de características agrupadas com base nos valores médios de cada região do filtro.

Figura 7 - Exemplo de uma operação do average-pooling com filtro 2X2 em uma imagem 4X4.



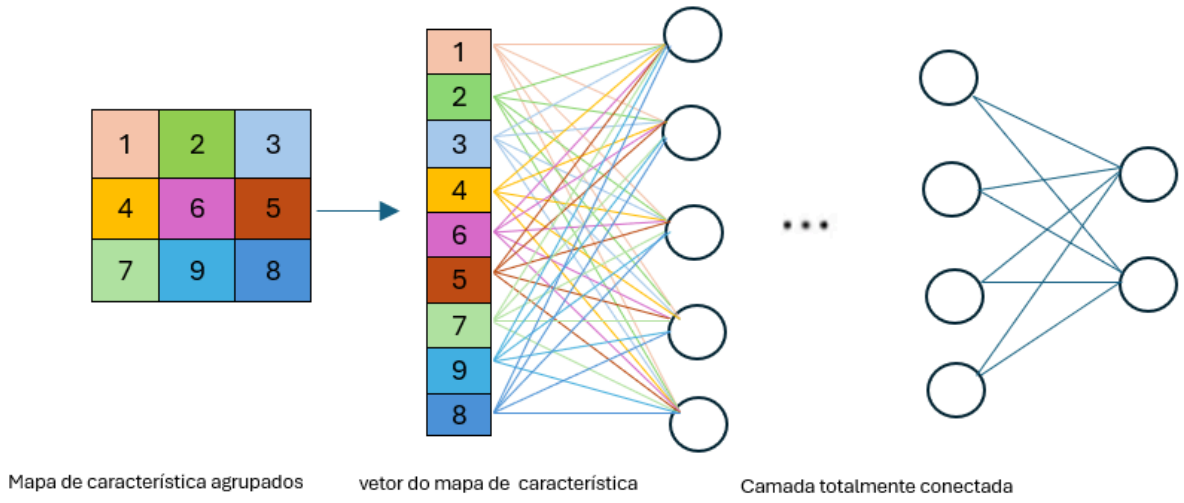
Fonte: Autor (2025).

2.2.1.3 Camada totalmente conectada

No final das redes convolucionais é colocada uma camada totalmente conectada, ou camada densa, como também é conhecida. Essa camada tem como objetivo conectar as características extraídas das camadas anteriores. Além disso, desempenha um papel de "traduzir" as características extraídas pelas camadas anteriores para obter uma saída de classificação da rede (Rodrigues, 2018).

Antes de obter a classificação, cada mapa de características obtidas nas camadas anteriores, passa por um processo de nivelamento, onde todas as características do mapa são convertidas em um vetor coluna, esse vetor é então passado como entrada na totalmente conectada (Jordan, 2017).

Figura 8 - um mapa de características em pool sendo nivelado e inserido em uma camada totalmente conectada (FC).



Fonte: Adaptado de Jordan, 2017.

2.2.2 Aplicabilidade em Segmentação

Atualmente, a segmentação de imagens é uma ferramenta indispensável em diversos contextos, desde medicina, veículos autônomos e estudos ambientais a inúmeras outras áreas de onde outrora parecia inalcançável a possibilidade de avanço. As Redes Neurais Convolucionais têm sido uma das melhores escolhas para realizar esse tipo de análise, a capacidade de isolar áreas específicas de imagem. Isso não apenas fornece aos que a aplicam informações muito mais

precisas do que em algum momento no passado próximo, mas também faz a análise que seria praticamente impossível – ou pelo menos envolveria muito tempo e custo extra – usando técnicas tradicionais. Como destacado por Malhotra et al. (2022),a eficácia das CNNs está relacionada à maneira como processam informações em diferentes níveis: desde detalhes simples, como contornos, até formas mais complexas.

Na área da saúde, a segmentação de imagens por CNNs é usada amplamente para identificar órgãos, diagnosticar doenças e identificar anomalias, como é o caso dos tumores. Como aponta Niyas et al.. em 2022, redes convolucionais tridimensional, ou 3D CNNs, funcionam bem para a interpretação de dados volumétricos, como os adquiridos por ressonâncias magnéticas e tomografias. As tecnologias mais recentes, como o FusionNet utilizado por (Quan et al. em 2021), usam uma abordagem inovadora com conexões residuais para preservar informações nas camadas mais profundas, garantindo maior precisão no resultado da segmentação.

Assim, uma das características mais marcantes das CNNs é sua capacidade de adaptação ao contexto em que são formadas. Sultana et al. mostram que treinadas a partir de um grande número de imagens anotadas, as redes de convolução não apenas ajudam a ver padrões gerais, mas também apresentam como parte sistemática um número de desafios a que geralmente precisam se adaptar, a saber, a alterações de iluminação, ruídos e diferenças de escala e outras. Essas características ajudaram a popularizar o uso de CNNs em áreas como geração de imagem de satélite e monitoramento florestal, onde cada detalhe da imagem pode oferecer informações cruciais sobre mudanças ambientais ou desmatamento.

Embora as Redes Neurais Convolucionais (CNNs) tenham muitas vantagens, elas enfrentam desafios significativos em situações onde existem pouca disponibilidade de dados anotados, o que é comum em áreas especializadas. Para resolver essas limitações, técnicas como aumento de dados têm sido muito utilizadas. Como aponta Ajmal et al. (2018), esses métodos permitem modelos pré-treinados em grandes conjuntos de dados adequados para determinadas tarefas, como segmentação na área da saúde ou segurança. Além de ampliar a área

de aplicação, esses recursos também ajudam a reduzir o custo e o tempo necessários para treinar modelos.

Outro ponto crítico das CNNs está relacionado ao equilíbrio entre precisão e eficiência computacional, especialmente em dispositivos com recursos limitados, como drones e câmeras de baixa capacidade. Han et al. (2022) sugerem alternativas como o ConvUNeXt, uma arquitetura projetada para operar de maneira mais leve e eficiente, tornando-se ideal para cenários onde o desempenho precisa ser otimizado.

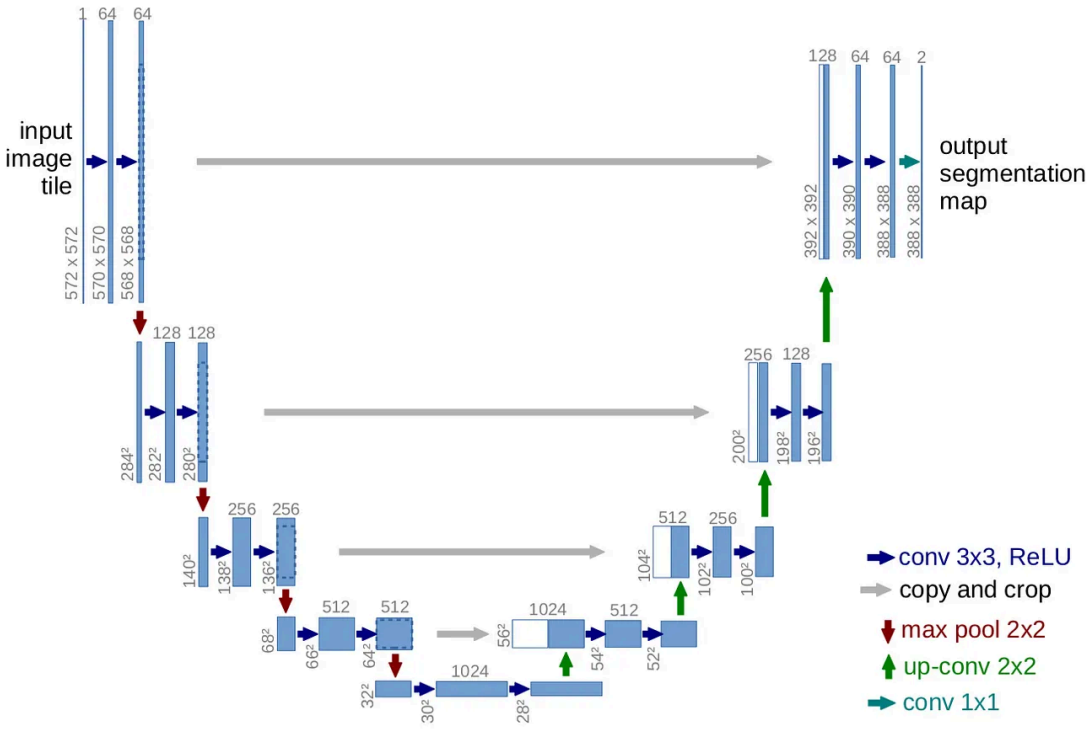
Dessa forma, a segmentação semântica por CNNs continua a evoluir rapidamente, com constantes avanços sendo alcançados. Guo et al. (2018) apontam que o futuro dessa tecnologia pode estar na integração com novas abordagens, como aprendizado não supervisionado e redes generativas adversárias (GANs). Essas combinações têm o potencial de aumentar a precisão e reduzir a necessidade de grandes volumes de dados anotados, expandindo ainda mais o alcance e a aplicabilidade das CNNs.

2.3 Arquitetura U-Net

A U-Net trouxe avanços no campo da segmentação de imagens, uma área que se dedica a identificar objetos e desenhar seus contornos com precisão em imagens. Essa arquitetura inovadora, apresentada pela primeira vez no artigo U-Net: Convolutional Networks for Biomedical Image Segmentation por Ronneberger et al. (2015), tem se destacado por sua grande utilidade, especialmente no processamento e análise de imagens biomédicas.

Como podemos ver na figura 9, a estrutura da U-Net é caracterizada por sua estrutura em forma de U, daí o nome. Ela é composta por dois elementos principais: um caminho de compressão, que reduz os dados para extrair suas características mais importantes, e um caminho de reconstrução, que refina essas informações e as reorganiza.

Figura 9 - Arquitetura U-Net



Fonte: Ronneberger, O.; Fischer, P.; Brox, T. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. (2015).

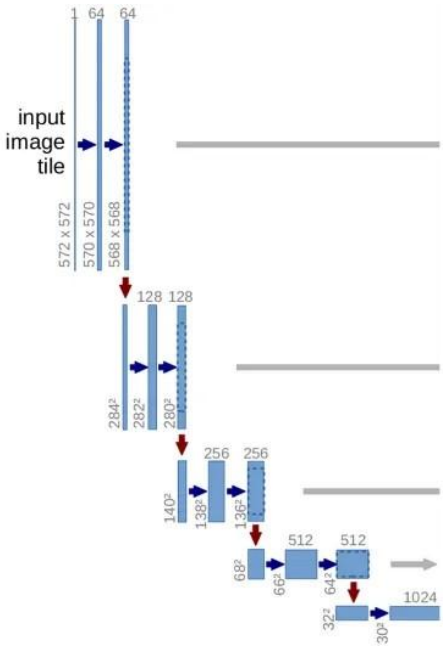
2.3.1 Estrutura da U-Net

2.3.1.1 Caminho de Contração (Encoder):

O caminho de contração ou encoder, é projetado para extrair e capturar informações da imagem. Conforme figura 10, esse caminho consiste em uma sequência de operações convolucionais (3x3), seguidas por funções de ativações ReLU e operações de pooling (2x2). Cada camada convolucional reduz progressivamente a dimensão espacial da imagem (de 572x572 para 32x32), enquanto aumenta a profundidade das características, começando de 64 até 1024 canais.

Essa redução espacial permite que a rede concentre-se em regiões de interesse, enquanto as conexões de salto armazenam informações essenciais para a reconstrução no caminho da expansão. Esse comportamento garante que os detalhes finos não sejam perdidos ao longo do processo (Azad et al., 2024).

Figura 10 - Caminho de contração da U-Net



Fonte: Ronneberger, O.; Fischer, P.; Brox, T. *U-Net: Convolutional Networks for Biomedical Image Segmentation* (com adaptações) (2015).

As conexões de salto são representadas pelas setas cinzas na imagem acima, que conectam cada camada da contração à camada de expansão correspondente. Essas conexões preservam informações espaciais de alta resolução que, de outra forma, seriam perdidas durante o pooling. Isso é especialmente crucial para a precisão em bordas e áreas complexas de segmentação, como observado por Chen et al. (2019).

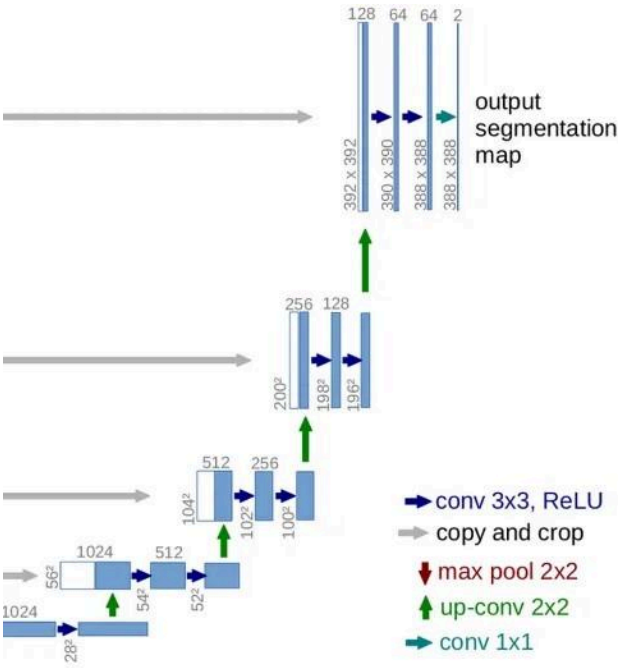
2.3.1.2 Caminho de Expansão (Decoder):

O caminho de expansão ou decoder, mostrado na figura 11, reverte as operações realizadas no caminho de contração. Ele usa up-convoluções (2x2), seguidas pelas convoluções (3x3) e ativações ReLU para aumentar a dimensão espacial da imagem e reduzir a profundidade do canal. Cada camada no caminho de expansão combina os mapas de características de baixo nível com as características de alto nível trazidas pelas conexões de salto.

O processo de upsampling começa com uma dimensão espacial de 32x32 e termina com a reconstrução completa próximo ao tamanho da imagem original (388x388). O uso de convoluções transpostas (setas verdes) é um diferencial que

minimiza perdas de informações frequentemente observados em operações de interpolação simples (Kang et al., 2022).

Figura 11 - Caminho de expansão da U-Net



Fonte: Ronneberger, O.; Fischer, P.; Brox, T. *U-Net: Convolutional Networks for Biomedical Image Segmentation* (com adaptações) (2015).

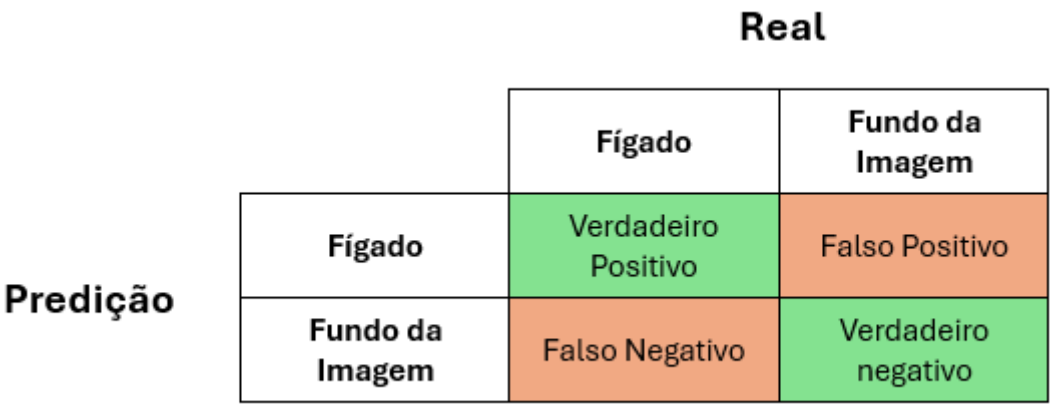
2.4 Métricas de avaliação de desempenho

Para avaliar a precisão do modelo de segmentação, diversas métricas foram desenvolvidas, sendo as mais utilizadas o Dice coefficient , o íntersection over Union (IoU) e a acurácia. Essas métricas são baseadas na matriz de confusão, que é uma ferramenta essencial para análise de desempenho de um modelo classificação e segmentação de imagens como mencionado por (Müller et al., 2022).

A matriz de confusão é uma ferramenta que organiza em formato de tabela a análise de desempenho de um modelo; contém quatro elementos principais : verdadeiros positivos (TP), verdadeiros negativos (TN), falsos positivos (FP) e falsos negativos (FN). Em aplicações, por exemplo, como a segmentação de imagens, verdadeiros positivos correspondem às regiões corretamente identificadas; falsos positivos, regiões classificadas incorretamente classificadas como pertencentes à região de interesse. Porém, falsos negativos representam regiões positivas que não foram detectadas, e os verdadeiros negativos são áreas

corretamente identificadas como não pertencentes ao objeto de interesse (Kofler et al., 2021). Na figura 12, mostra um exemplo de uma matriz de confusão.

Figura 12 - Matriz de confusão



Fonte: Autor (2025).

A métrica Coefficient Dice é muito utilizada na segmentação de imagem para medir a semelhança entre dois conjuntos: a segmentação prevista e a segmentação real, para isso, é utilizado o cálculo da área sobreposta dos dois conjuntos, levando em consideração os falsos negativos e os falsos positivos. Essa métrica é definida pela fórmula:

$$Dice = \frac{2|A \cap B|}{|A|+|B|} ou \frac{2TP}{2TP + FP + FN}$$

Onde A representa a região segmentada pelo modelo e B representa a região de referência. O Dice coefficient varia entre 0 e 1, onde 1 indica uma sobreposição entre os dois conjuntos perfeitamente e, portanto, melhor desempenho de segmentação, enquanto uma 0 indica nenhuma sobreposição. Estudos recentes indicam que essa métrica é sensível a pequenos desvios na segmentação, tornando-a especialmente útil para a avaliação de pequenas regiões de interesse (Tang et al., 2024).

O índice de Jaccard, ou como também é conhecido Intersection over Union (IoU), é semelhante ao coefficient Dice. Como o próprio nome já diz é a área da interseção sobre a união da segmentação prevista e a segmentação real, sendo definida pela fórmula:

$$IoU = \frac{|A \cap B|}{|A \cup B|} \; ou \; \frac{TP}{TP + FP + FN}$$

Essa métrica, assim como o coefficient Dice, varia de 0 a 1, com 0 significando sem nenhuma sobreposição e 1 significando uma sobreposição perfeita. O IoU é especialmente relevante em contextos onde é importante minimizar falsos positivos e falsos negativos (Tang et al., 2024).

A acurácia é uma métrica de mais fácil entendimento, sendo definida como o número de previsões corretas, tanto previsões positivas quanto negativas certas dívidas pelo total de previsões.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Essa métrica, apesar de ser bastante usada, sua precisão pode ser prejudicada quando os pixels não estão distribuídos de maneira equilibrada. Isso particularmente existe onde a área de interesse costuma ser uma região muito menor da imagem em comparação ao restante (Müller et al., 2022).

3 METODOLOGIA

A metodologia utilizada neste trabalho foi estruturada em várias etapas, envolvendo desde a parte da obtenção dos dados até as ferramentas e ambiente de desenvolvimento.

3.1 Obtenção e Preparação do Dataset

Para realização deste trabalho foi utilizado uma base de dados denominada *Segmentation of Liver* disponível na plataforma Kaggle (disponível em:https://www.kaggle.com/datasets/priyamsaha17/segmentation-of-liver/data), contendo um total de 2823 imagens hepáticas e 2823 máscaras correspondentes, totalizando 5.646 imagens. Cada imagem tem dimensão de 512x512, ou seja, 512 pixels de largura e 512 pixels de altura, com extensão jpg, e as máscaras binárias indicam as regiões hepáticas. Para facilitar o processamento de dados e o treinamento do modelo, todas as imagens foram armazenadas no Google Drive em dois diretórios distintos, um contendo as imagens de entradas (*images*), e outro contendo suas respectivas máscaras

(*liver_masks*). Para garantir a correta correspondência entre as imagens e as máscaras, foi utilizada uma ordenação dos arquivos antes da leitura dos dados.

3.2 Pré-processamento das Imagens

Depois da organização do dataset, nesta etapa foi feito o ajuste das imagens e suas respectivas máscaras, deixando-as todas com o mesmo tamanho, sendo redimensionada para 128x128 pixels. Esse ajuste levou em consideração buscar um equilíbrio entre resolução da imagem e a eficiência computacional do modelo, pois uma imagem de tamanho muito grande demandaria muito poder computacional e retardaria no treinamento do modelo. Além disso, foi realizada a normalização dos valores de pixel, convertendo os valores de 0 a 255 para o intervalo de [0,1]. Esse processo auxilia na estabilidade do treinamento da rede neural.

As imagens foram carregadas utilizando a biblioteca OpenCV e convertidas para escala de cinza, já que a segmentação não requer informações de cor. O mesmo procedimento foi realizado para as máscaras, garantindo que as áreas segmentadas fossem corretamente mapeadas.

3.3 Aumento de Dados (Data Augmentation)

Para evitar o overfitting e aumentar a diversidade do dataset, foram aplicadas técnicas de data augmentation no conjunto de treinamento. O objetivo dessa etapa foi criar variações das imagens originais, permitindo que a rede se tornasse mais robusta e capaz de generalizar melhor novos dados. Foram utilizadas transformações da biblioteca Albumentation, incluindo espelhamento horizontal e vertical(HorizontalFlip, VerticalFlip), rotação e escala aleatória (ShiftScaleRotate), e transformações elásticas(ElasticTransform), que simulam pequenas distorções na estrutura das imagens. Essas transformações foram aplicadas em cada uma das imagens de treinamento, aumentando significativamente que passou de 1.976 para 9.880 imagens.

3.4 Construção da Arquitetura U-Net

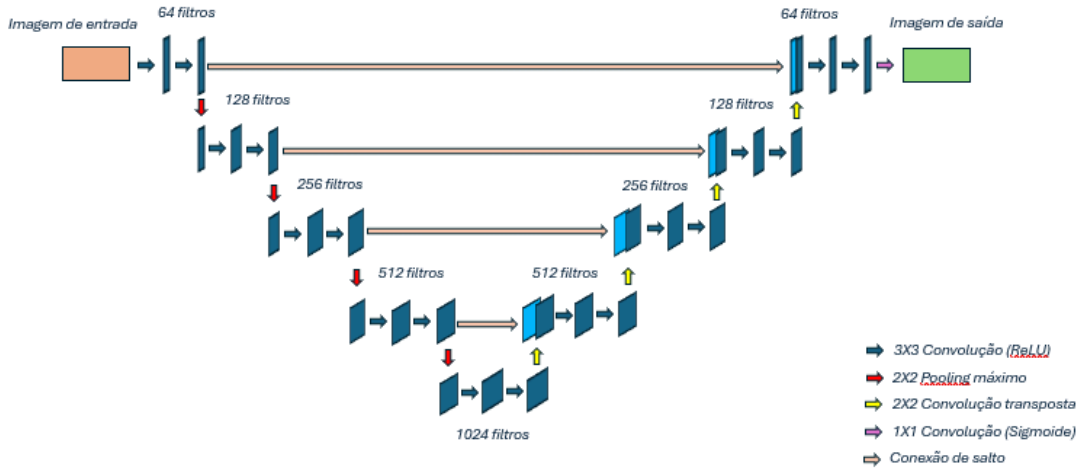
A arquitetura U-Net utilizada na segmentação de imagens hepáticas segue um desenho simétrico composto por um encoder e um decoder, conectados por um *bottleneck* na parte central. O encoder é responsável por extrair características das imagens e reduzir sua dimensão. Ele é composto por

quatro blocos convolucionais, onde o número inicial de filtros é 64 e dobra a cada camada seguinte (128, 256 e 512 filtros). Cada bloco contém duas camadas convolucionais seguidas por uma operação MaxPolling2D, que reduz a dimensão espacial pela metade, permitindo a captação de informações de alto nível. No ponto central da rede, o *bottleneck* atinge 1024 filtros, extraindo padrões essenciais para a segmentação.

O decoder tem a função de reconstruir a segmentação, aumentando gradativamente a resolução da imagem enquanto reduz a quantidade de filtros. Ele utiliza operações de convolução transposta (Conv2DTranspose) para reverter a redução espacial do encoder, espelhando sua estrutura. O número de filtros começa em 512 no primeiro bloco do decoder e diminui progressivamente para 256, 128 e 64 filtros. Em cada etapa, os mapas de características do encoder são concatenados ao decoder através das skip connections, preservando detalhes espaciais críticos para a segmentação do fígado. Essa conexão entre encoder e decoder evita a perda de informações importantes e melhora a precisão do modelo.

Por último, é uma convolução 1×1 com ativação sigmoid, que gera uma máscara segmentada binária, onde cada pixel representa a probabilidade de pertencer ao fígado.

Figura 13 - Arquitetura U-Net da segmentação de imagens hepáticas



O autor (2025).

3.5 Otimização de Hiperparâmetros com Optuna

Para obter os melhores hiperparâmetros do modelo, foi utilizado o framework Optuna, uma ferramenta que encontra automaticamente os melhores parâmetros para o modelo. O objetivo dessa otimização foi minimizar a perda no conjunto de validação. Os hiperparâmetros ajustados foram: (i) número de filtros convolucionais (16, 32 ou 64), (ii) taxa de aprendizado (entre 1e-5 e 1e-2), (iii) tamanho do batch (4, 8 ou 16), (iv) número de épocas (de 10 a 50) e (v) taxa de dropout (entre 0 e 0.5), essas variações foram ajustadas de acordo com experimentos anteriores. Foram realizados 20 experimentos utilizando diferentes combinações desses hiperparâmetros, e o conjunto que minimizou a perda foi utilizado para o treinamento final do modelo. A função de perda utilizada foi a Binary Crossentropy, e as métricas de desempenho monitoradas foram o IoU (Intersection over Union) e o Dice Coefficient.

3.6 Treinamento e Avaliação do Modelo

Após a otimização dos hiperparâmetros, o modelo final foi treinado utilizando os dados aumentados. Inicialmente, o dataset foi dividido em 70% para treinamento, 15% para validação e 15% para teste. O Data Augmentation foi aplicado após a divisão do dataset na base de dados do treinamento, aumentando de 1.976 para 9.880 imagens e logo depois foi embaralhado, para evitar que o modelo aprendesse a ordem. O treinamento foi realizado com o otimizador Adam, e o desempenho do modelo foi observado ao longo das épocas. Para evitar o overfitting, foram utilizados o ModelCheckpoint, que salva automaticamente o modelo com melhor desempenho, e o EarlyStopping, que interrompe o treinamento caso a função de perda no conjunto de validação não melhore.

Tabela 1 - Divisão das imagens inicialmente para treinamento, validação e teste.

Conjunto	Quantidade de imagens
Treinamento	1976
Validação	423
Teste	424
Total	2823

Após o treinamento, o modelo foi avaliado utilizando o conjunto de testes, as métricas IoU, Dice Coefficient, acurácia e perda foram computadas para verificar a qualidade das segmentações geradas. Além disso, foi realizada uma inspeção visual das máscaras preditas pelo modelo em comparação com as máscaras reais. Para isso, foram geradas imagens comparativas, permitindo a verificação da qualidade das predições. Por último, foram plotadas as curvas de perda, acurácia, IoU e Dice Coefficient, monitorando a evolução do treinamento ao longo das épocas.

3.7 Ferramentas e Ambiente de Desenvolvimento

O desenvolvimento deste projeto foi realizado no Google Colab Pro, um ambiente hospedado nas nuvens pelo próprio Google que é muito utilizado para Aprendizagem de Máquina e Inteligência Artificial, a linguagem utilizada para implementar o modelo foi a linguagem de programação Python. A versão Pro do Colab oferece acesso a GPUs mais potentes em comparação com a versão gratuita, isso possibilitou a realização de treinamentos mais rápidos e a execução de otimizações de hiperparâmetros com mais número de experimentos.

As bibliotecas utilizadas na implementação deste trabalho foram:

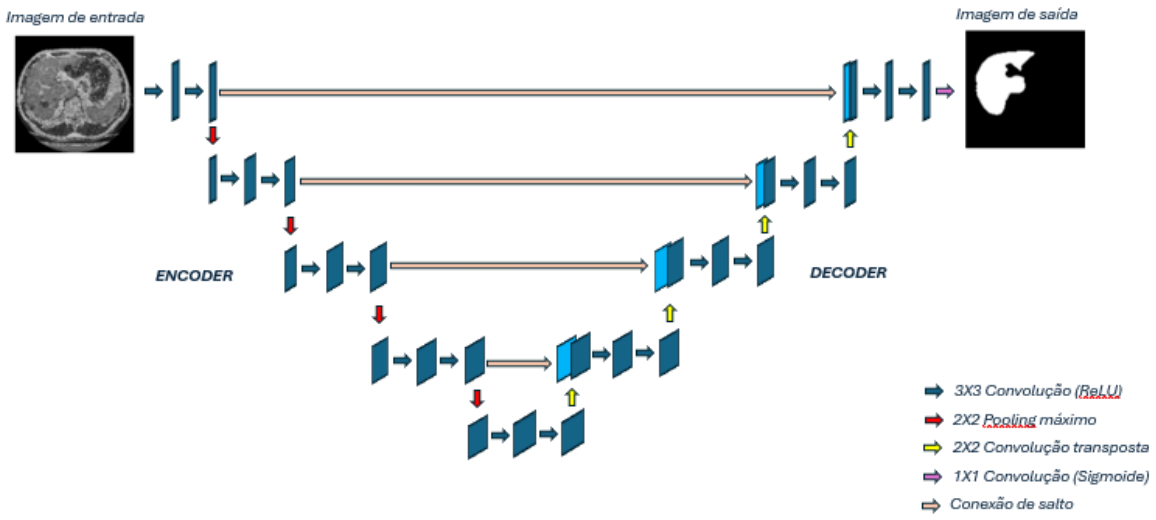
- TensorFlow/Keras - Implementação da rede neural convolucional U-Net.
- numpy - Manipulação de arrays e operações matriciais.
- matplotlib - Biblioteca para visualização de dados.
- scikit-learn - Ferramenta utilizada para a divisão dos dados e embaralhamento.
- os - utilizada para carregamento de imagens do diretório
- openCV - Processamento das imagens
- random - Geração de números aleatórios
- optuna - Ferramenta de otimização de hiperparâmetros
- Albumentations - Implementa algoritmos de Data Augmentation.

4 Resultados e Discussões

4.1 Configuração e Otimização do Modelo

O modelo U-Net implementado neste trabalho é o modelo, proposto por Ronneberger et al. , e compreende os caminhos de contração e expansão. O caminho de contração, ou encoder, extrai características da imagem de entrada convolução seguida de pooling. O caminho de expansão, por sua vez, ou decoder, combina as características extraídas pelo encoder e reconstrói a segmentação. Utiliza-se, ainda, conexões de salto para integração de características extraídas em camadas anteriores, podemos ver sua estrutura na figura 14.

Figura 14 - Estrutura da arquitetura U-Net utilizada na segmentação



Fonte: O autor (2025).

A biblioteca Optuna facilitou a identificação da melhor configuração após a otimização de hiperparâmetros. Os hiperparâmetros selecionados foram de fatores que resultaram ótimo para a taxa de aprendizado de 0.000118, batch size de 4, número de filtros 64, dropout 0.219 e 30 épocas de treinamento. As configurações eleitas garantiram o máximo equilíbrio entre a capacidade de aprendizado e a prevenção de overfitting.

Figura 15 - Resultado da otimização dos hiperparâmetros com optuna resumido

```
study = optuna.create_study(direction='minimize') # minimizar loss
study.optimize(objective, n_trials=20) #numero de experimentos

print("Melhor configuração de hiperparâmetros:")
print(study.best_params)
```

```
[I 2025-01-25 16:06:43,454] A new study created in memory with name: no-name-02094226-aad7-4294-96a6-8ff7ab2cfe71
<ipython-input-20-113652524b96>:4: FutureWarning: suggest_loguniform has been deprecated in v3.0.0. This feature will be removed i
  learning_rate = trial.suggest_loguniform('learning_rate', 1e-5, 1e-2) #Taxa de aprendizado
<ipython-input-20-113652524b96>:7: FutureWarning: suggest_uniform has been deprecated in v3.0.0. This feature will be removed in v
  dropout_rate = trial.suggest_uniform('dropout_rate', 0.0, 0.5)
[I 2025-01-25 16:22:59,619] Trial 0 finished with value: 0.07556312531232834 and parameters: {'n_filters': 16, 'learning_rate': 0.
[I 2025-01-25 16:52:17,739] Trial 1 finished with value: 0.0065549565479159355 and parameters: {'n_filters': 32, 'learning_rate':
[I 2025-01-25 17:33:48,530] Trial 2 finished with value: 0.004033829551190138 and parameters: {'n_filters': 64, 'learning_rate': 0
[I 2025-01-25 17:37:21,873] Trial 3 finished with value: 0.03641658276319504 and parameters: {'n_filters': 16, 'learning_rate': 3.
[I 2025-01-25 17:44:04,827] Trial 4 finished with value: 0.04237291216850281 and parameters: {'n_filters': 32, 'learning_rate': 0.
[I 2025-01-25 17:57:31,787] Trial 5 finished with value: 0.042990658432245255 and parameters: {'n_filters': 16, 'learning_rate': 1
[I 2025-01-25 19:07:26,710] Trial 6 finished with value: 0.06650450080633163 and parameters: {'n_filters': 64, 'learning_rate': 0.
[I 2025-01-25 20:28:42,493] Trial 7 finished with value: 0.00477279257029295 and parameters: {'n_filters': 64, 'learning_rate': 0.
[I 2025-01-25 20:56:25,915] Trial 8 finished with value: 0.005239939782768488 and parameters: {'n_filters': 32, 'learning_rate': 5
[I 2025-01-25 21:14:50,718] Trial 9 finished with value: 0.03820356726646423 and parameters: {'n_filters': 16, 'learning_rate': 0.
[I 2025-01-25 21:48:30,610] Trial 10 finished with value: 0.0042271846905350685 and parameters: {'n_filters': 64, 'learning_rate':
[I 2025-01-25 22:22:17,713] Trial 11 finished with value: 0.004309191834181547 and parameters: {'n_filters': 64, 'learning_rate':
[I 2025-01-25 22:55:44,564] Trial 12 finished with value: 0.004255830775946379 and parameters: {'n_filters': 64, 'learning_rate':
[I 2025-01-25 23:28:39,413] Trial 13 finished with value: 0.00822686217725277 and parameters: {'n_filters': 64, 'learning_rate': 0
[I 2025-01-26 00:30:33,541] Trial 14 finished with value: 0.0037641660310328007 and parameters: {'n_filters': 64, 'learning_rate':
[I 2025-01-26 01:32:41,391] Trial 15 finished with value: 0.00396326370537281 and parameters: {'n_filters': 64, 'learning_rate': 7
[I 2025-01-26 02:34:52,948] Trial 16 finished with value: 0.004467307589948177 and parameters: {'n_filters': 64, 'learning_rate':
[I 2025-01-26 03:37:24,685] Trial 17 finished with value: 0.007400454021990299 and parameters: {'n_filters': 64, 'learning_rate':
[I 2025-01-26 04:39:24,094] Trial 18 finished with value: 0.004030998330563307 and parameters: {'n_filters': 64, 'learning_rate':
[I 2025-01-26 05:01:38,903] Trial 19 finished with value: 0.006579367909580469 and parameters: {'n_filters': 32, 'learning_rate':
Melhor configuração de hiperparâmetros:
{'n_filters': 64, 'learning_rate': 0.00011820317622349755, 'batch_size': 4, 'epochs': 30, 'dropout_rate': 0.21945006193443994}
```

Fonte: O autor (2025).

4.2 Desempenho Quantitativo

O modelo apresentou resultados expressivos tanto no conjunto de validação como também no conjunto de testes. Ao finalizar o treinamento com as 30 épocas, a análise das 423 imagens de validação revelou médias de 93% para a métrica IoU e 95% para Dice coefficient . No conjunto de teste, composto por 424 imagens, os resultados médios foram de 91% para IoU e 93% para Dice coefficient, como podemos ver na tabela 2.

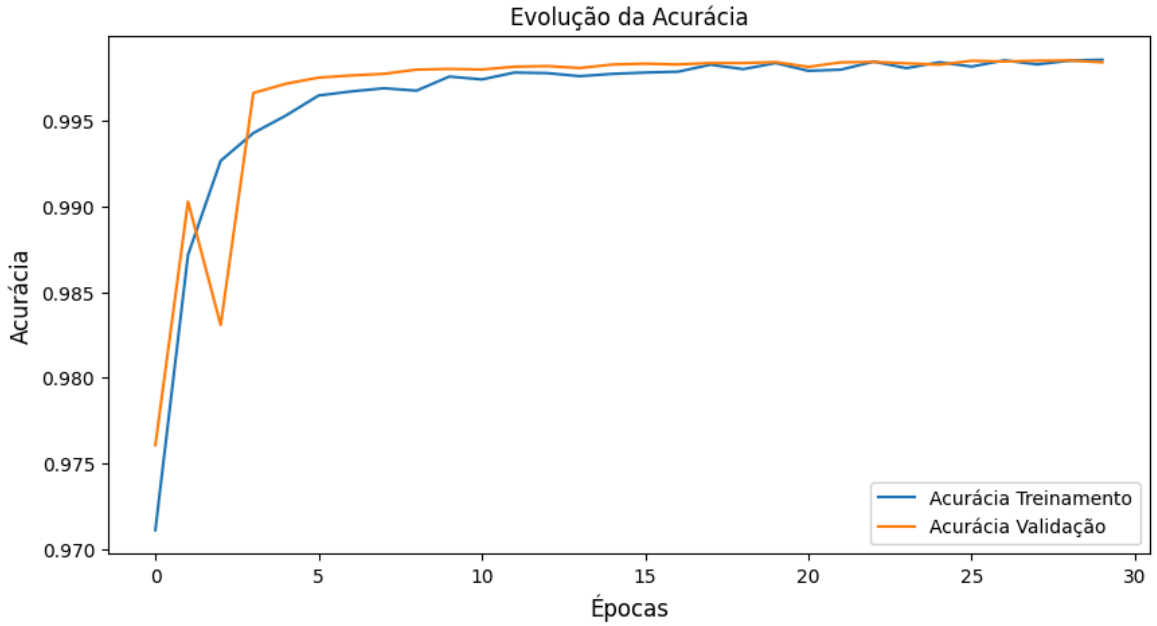
Tabela 2 - Resultado das métricas IoU e Dice Coefficient

Conjunto	IoU(%)	Dice Coefficient (%)
Validação	93	95
Teste	91	93

Nas figuras 16 e 17, que exibe o comportamento das métricas de Perda e Acurácia no treinamento e validação. A acurácia é mostrada como um desempenho excelente de boa generalização do modelo sobre os dados de treinamento e validação. A convergência das curvas é bem nítida, indicando que o modelo está

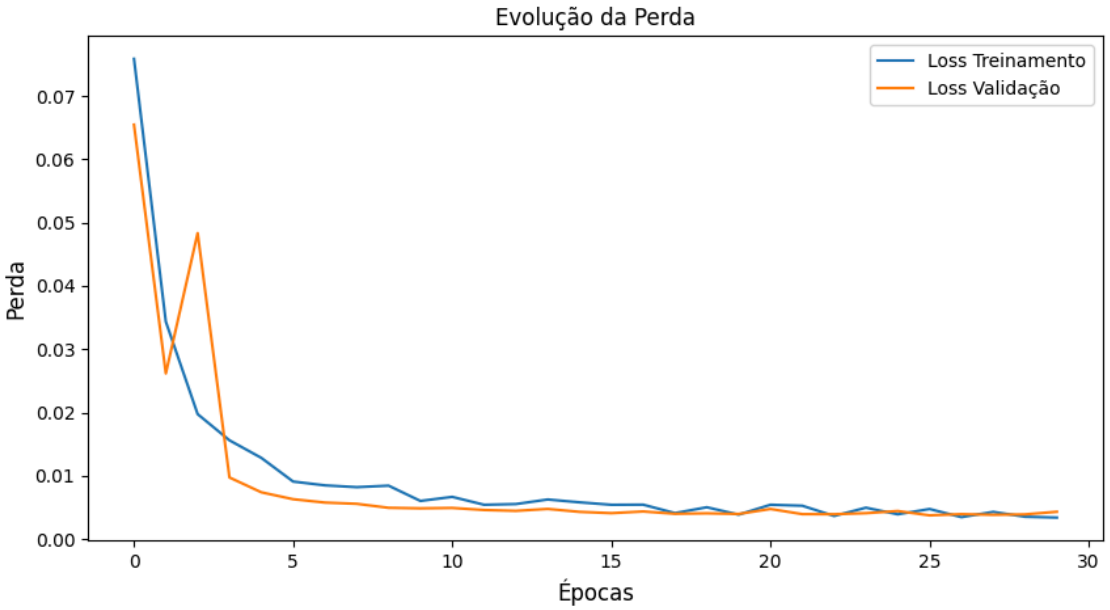
próximo de seu desempenho ideal. Embora inicialmente tenha apresentado uma oscilação, isso pode ser atribuído à forma como os pesos da rede são inicializados, já que, inicialmente, esses pesos são definidos de maneira aleatória. Já o gráfico da Perda, por sua vez, também teve uma oscilação no declínio nas primeiras épocas, e depois convergiu para o mínimo do gráfico. Apesar disso, a proximidade das curvas de perda do treinamento e validação sugere que o modelo generaliza bem e indica a não ocorrência de overfitting.

Figura 16 - Evolução da Acurácia



Fonte: O autor (2025).

Figura 17 - Evolução da Perda



Fonte: O autor (2025).

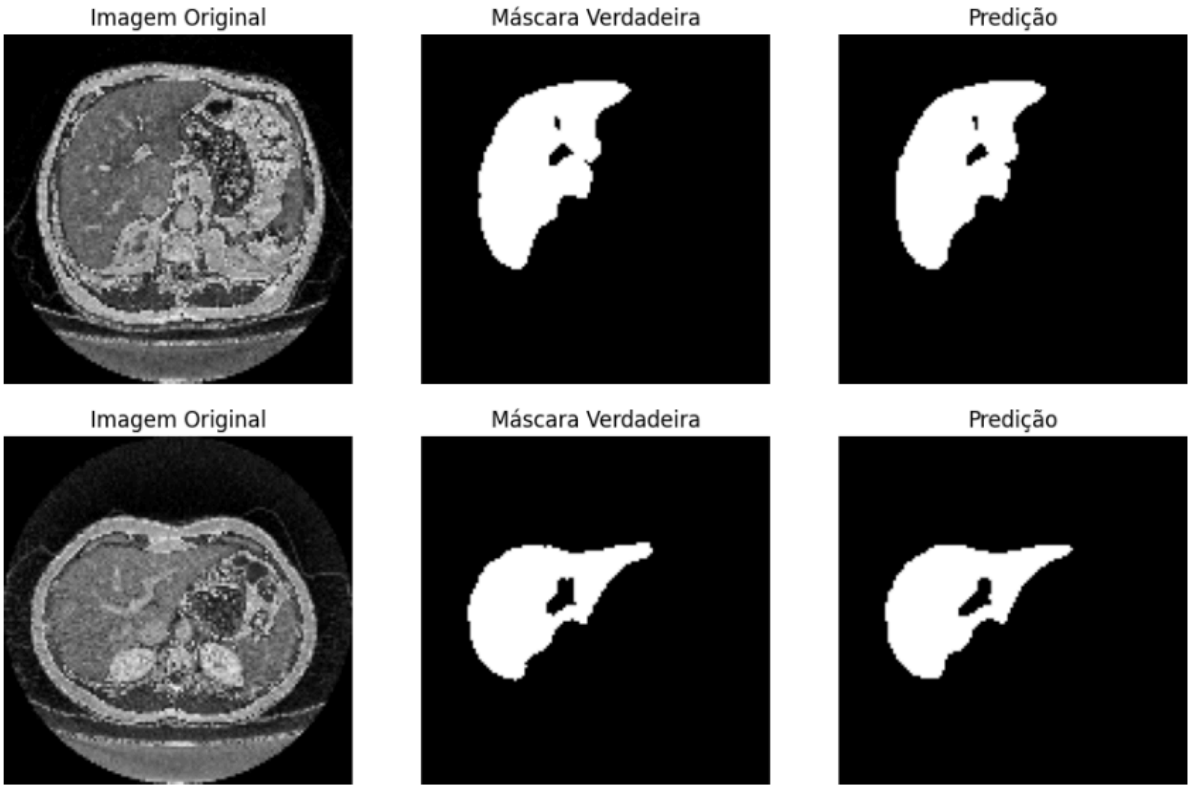
4.3 Avaliação Qualitativa

As figuras 18, 19 e 20 ilustram os resultados qualitativos do modelo nos conjuntos de validação e teste. As imagens servem para fazer uma comparação visual entre a imagem original, a máscara verdadeira e a predição realizada pelo modelo.

4.3.1 Imagens de Validação

Nas imagens de validação, ilustradas na figura 18, observa-se um alinhamento muito parecido entre a predição do modelo e as máscaras reais, com as bordas segmentadas representando quase que fiel aos contornos das regiões do fígado. Esse resultado mostra que o modelo conseguiu identificar padrões aprendidos ao longo do processo de treinamento.

Figura 18 - Imagem original, máscara real e predição da validação



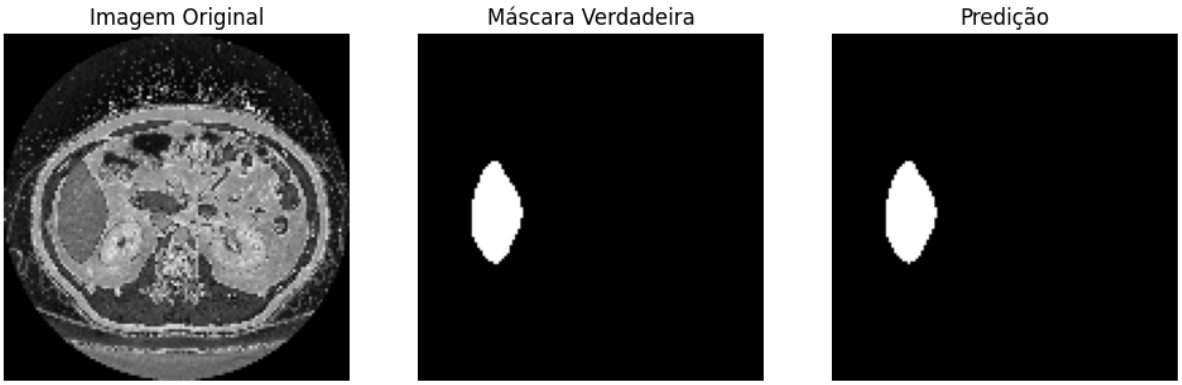
Fonte: O autor (2025).

4.3.2 Imagens de Teste

Para analisar as imagens de teste foram selecionadas as imagens que obtiveram os maiores e menores valores de Dice Coefficiente e IoU. Curiosamente, a mesma imagem teve os maiores valores de Dice Coefficiente (0,99) e IoU (0,98), enquanto outra imagem obteve os menores valores para ambas as métricas (0,0125), o que nos permitiu fazer uma análise comparativa.

Na figura 19, é ilustrada a imagem que obteve os maiores valores de Dice Coefficient (0,99) e IoU (0,98), cujo resultado, como visto na seção de métricas de avaliação, indica que o modelo se aproximou do ideal. Na imagem, o modelo conseguiu classificar corretamente quase todos os pixels. Isso significa que houve pouquíssimos falsos positivos (pixels falsamente classificados como fígado) e falsos negativos (pixels falsamente classificados como “não fígados”). Esse resultado sugere que a imagem possui algumas características que facilitam a segmentação, como contornos bem definidos ou até mesmo o alto contraste entre o fígado e os outros órgãos próximos.

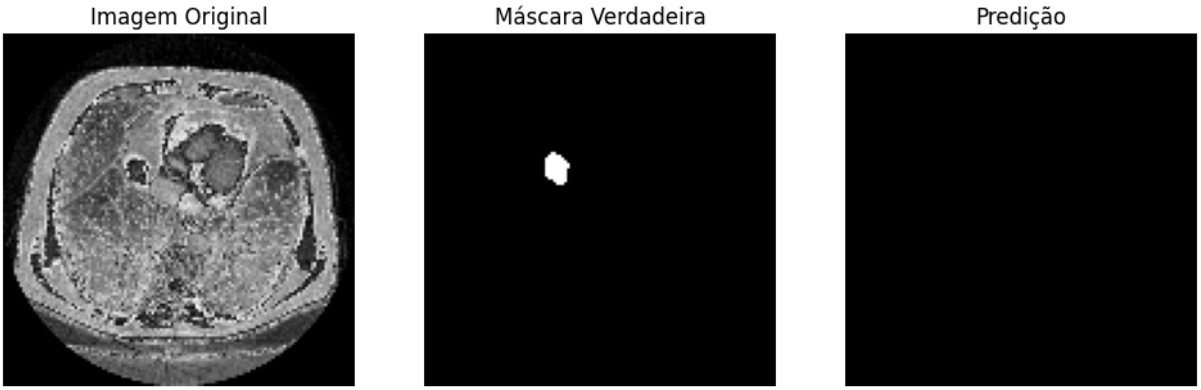
Figura 19 - Imagem original, máscara real e predição do teste com maior Iou e Dice Coefficient



Fonte: O autor (2025).

Por outro lado, a figura 20 ilustra a imagem com o menor Dice Coefficient e IoU, cujo valor foi de 0,0125, demonstrando a dificuldade do modelo de segmentar. Nesse caso, podemos dizer que, como visto na seção de métricas avaliativas, revela que o modelo falhou em classificar a maioria dos pixels do fígado, ou seja, teve muitos falsos negativo e, ao mesmo tempo, classifica os pixels do fígado como pixels do fundo da imagem, ou seja, muitos falsos positivos.

Figura 20 - Imagem original, máscara real e predição do teste com menor Iou e Dice Coefficient



Fonte: O autor (2025).

4.4 Comparação com a Literatura

Ao comparar com Han et al. (2022) e Bilic et al. (2022), percebe-se que ambos os trabalhos sugerem modificações à arquitetura U-Net para melhorar a segmentação de imagens médicas. Han et al. propõem o ConvUNeXt, que usa convoluções de grande kernel e mecanismo leve de atenção para obter

segmentações menos ruidosas e mais estáveis em relação às abordagens tradicionais. Esse modelo demonstrou eficácia particularmente em contextos onde há escassez de dados, proporcionando uma segmentação mais eficiente com menos parâmetros. Em comparação, nosso modelo utiliza a U-Net tradicional com otimização de hiperparâmetros e regularização, obtendo métricas comparáveis sem a necessidade de arquiteturas mais avançadas.

Por outro lado, Bilic et al. (2022) analisaram os melhores algoritmos de uma competição para avaliar modelos de segmentação de fígado e tumores. Os melhores modelos alcançaram um Dice Coefficient de aproximadamente 0.963 para fígado e 0.739 para tumores. Nosso modelo, focado na segmentação hepática, alcançou um Dice Coefficient de 0.95 na validação e 0.93 no teste, ou seja, nosso modelo se saiu tão bem quanto os melhores da competição.

Com isso, nosso estudo demonstrou que, embora algumas variantes da U-Net possam melhorar a eficiência computacional e a segmentação de detalhes anatômicos, a escolha criteriosa de hiperparâmetros e técnicas de regularização pode resultar em uma abordagem altamente eficaz sem a necessidade de redes mais complexas.

4.5 Limitações

Embora o modelo tenha demonstrado um desempenho significativo na segmentação de imagens hepáticas, algumas limitações foram identificadas.

- **Dependência de um conjunto de dados** - O modelo pode ter sua capacidade de generalização reduzida quando não utilizada por imagens de diferentes fontes.
- **Problemas de execução no Colab Pro** - Mesmo utilizando a versão Pro do Colab, enfrentamos dificuldades com o tempo de execução, onde tivemos desconexões inesperadas durante o treinamento, impactando no treinamento do nosso modelo.

5 Conclusões e Trabalhos Futuros

Neste trabalho, desenvolvemos um modelo baseado na arquitetura U-Net para a segmentação automática de imagens hepáticas. A metodologia utilizada compreendeu a aquisição e preparo de um dataset público, pré-processamento das

imagens, otimização de hiperparâmetros e treinamento de um modelo. Os resultados obtidos comprovam a eficácia da abordagem, pois as métricas quantitativas e qualitativas revelam uma notável precisão na segmentação.

Comparando com trabalhos da literatura, vimos que a abordagem adotada apresentou desempenho semelhantes com modelos avançados, demonstrando que uma arquitetura bem ajustada pode atingir resultados de alto nível sem necessidade de redes mais complexas. No entanto, apesar do alto desempenho, algumas limitações foram identificadas, como a dependência de um único conjunto de dados e as dificuldades computacionais enfrentadas durante o treinamento no Google Colab Pro.

Apesar do estudo ter alcançado resultados notáveis, algumas tarefas podem ser exploradas em trabalhos futuros, como:

- Expandir o conjuntos de dados: incluir imagens de diferentes fontes pode melhorar a generalização do modelo;
- Aprimoramento do Pré-processamento: Testar diferentes técnicas de filtragem e realce de contraste para ajudar na qualidade das imagens, antes de passar para a rede neural;
- Experimentação com outras arquiteturas: Apesar da U-Net tradicional apresentar um resultado muito bom, outras arquiteturas podem ser utilizadas para melhorar a precisão da segmentação;
- Segmentação de estruturas adicionais: Além do fígado, poderia expandir o modelo para segmentar outras estruturas anatômicas, como tumores ou lesões.

6 Referências

KAYALIBAY, B.; JENSEN, G.; VAN DER SMAGT, P. CNN-based segmentation of medical imaging data. 2017. Disponível em: https://arxiv.org/pdf/1701.03056. Acesso em: 10 jan. 2025.

ADDIMULAM, S.; MOHAMMED, M. A. Deep Learning-Enhanced Image Segmentation for Medical Diagnostics. *ResearchGate*, 2020. Disponível em: https://www.researchgate.net/publication/381671435. Acesso em: 11 jan. 2025.

JORDAN, Jeremy. Convolutional Neural Networks. Jeremy Jordan. 16 jul. 2017. Disponível em: https://www.jeremyjordan.me/convolutional-neural-networks/ Acesso em: 11 jan. 2025.

YUAN, F.; ZHANG, Z.; FANG, Z. An effective CNN and Transformer complementary network for medical image segmentation. *ScienceDirect*, 2023. Disponível em: https://www.sciencedirect.com/science/article/pii/S0031320322007075. Acesso em: 13 jan. 2025.

ASGARI TAGHANAKI, S.; ABHISHEK, K.; COHEN, J. P. Deep semantic segmentation of natural and medical images: a review. 2021. Disponível em: https://arxiv.org/pdf/1910.07655. Acesso em: 12 jan. 2025.

JASIM, W. N.; MOHAMMED, R. J. A Survey on Segmentation Techniques for Image Processing. Disponível em: https://www.iasj.net/iasj/download/8cadd4e8476b1298. Acesso em: 13 jan. 2025.

ABDULATEEF, S. K.; SALMAN, M. D. A Comprehensive Review of Image Segmentation Techniques. Disponível em: https://www.iasj.net/iasj/download/c1809f5ac0252d40. Acesso em: 13 jan. 2025.

Sharma, P., & Suji, J. (2016). A review on image segmentation with its clustering techniques. Disponível em: https://www.academia.edu/download/89428045/18.pdf. Acesso em: 12 jan. 2025.

PEDRINI, H.; SCHWARTZ, W. R. *Análise de imagens digitais: princípios, algoritmos e aplicações*. 2. ed. São Paulo: Thomson Learning, 2017.

GONZALEZ, Rafael C.; WOODS, Richard E. *Processamento digital de imagens*. 3. ed. São Paulo: Pearson, 2010.

KAUR, D.; KAUR, Y. Various Image Segmentation Techniques: A Review. 2014. Disponível em: V3I5201499a84.pdf. Acesso em: 14 jan. 2025.

JAGLAN, P.; DASS, R.; DUHAN, M. A Comparative Analysis of Various Image Segmentation Techniques. 2019. Disponível em: https://link.springer.com/chapter/10.1007/978-981-13-1217-5_36. Acesso em: 14 jan. 2025.

SONG, Y.; YAN, H. Image Segmentation Techniques Overview. 2017. Disponível em: https://ieeexplore.ieee.org/abstract/document/8424314/. Acesso em: 15 jan. 2025.

RODRIGUES, D. A. Deep learning e redes neurais convolucionais: reconhecimento automático de caracteres em placas de licenciamento automotivo. Universidade Federal da Paraíba, 2018.

AJMAL, H.; REHMAN, S.; FAROOQ, U.; AIN, Q. U. Convolutional neural network based image segmentation: a review. Disponível em: Spiedigitallibrary.org. Acesso em: 15 jan. 2025.

GUO, Y.; LIU, Y.; GEORGIOU, T.; LEW, M. S. A review of semantic segmentation using deep neural networks. Disponível em: https://link.springer.com/article/10.1007/s13735-017-0141-z. Acesso em: 17 jan. 2025.

HAN, Z.; JIAN, M.; WANG, G. G. ConvUNeXt: An efficient convolution neural network for medical image segmentation. Disponível em: https://www.sciencedirect.com/science/article/pii/S0950705122007572. Acesso em: 17 jan. 2025.

MALHOTRA, P.; GUPTA, S.; KOUNDAL, D. Deep Neural Networks for Medical Image Segmentation. Disponível em: https://www.researchgate.net/publication/359154557. Acesso em: 20 jan. 2025.

NIYAS, S.; PAWAN, S. J.; KUMAR, M. A.; RAJAN, J. Medical image segmentation with 3D convolutional neural networks: A survey. Disponível em: https://www.sciencedirect.com/science/article/pii/S0925231222004362. Acesso em: 21 jan. 2025.

QUAN, T. M.; HILDEBRAND, D. G. C.; JEONG, W. K. FusionNet: A deep fully residual convolutional neural network for image segmentation in connectomics. Disponível em: https://www.frontiersin.org/articles/10.3389/fcomp.2021.613981/full. Acesso em: 21 jan. 2025.

SULTANA, F.; SUFIAN, A.; DUTTA, P. Evolution of image segmentation using deep convolutional neural network: A survey. Disponível em: https://www.sciencedirect.com/science/article/pii/S0950705120303464. Acesso em: 21 jan. 2025.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 234-241. DOI: 10.1007/978-3-319-24574-4_28

MÜLLER, D.; SOTO-REY, I.; KRAMER, F. Towards a guideline for evaluation metrics in medical image segmentation. BMC Research Notes, 2022. Disponível em: https://link.springer.com/article/10.1186/s13104-022-06096-y. Acesso em: 03 fev. 2025.

TANG, L. et al**.** Bolstering Performance Evaluation of Image Segmentation Models with Efficacy Metrics in the Absence of a Gold Standard. *IEEE Xplore*, 2024. Disponível em: https://ieeexplore.ieee.org/abstract/document/10643219/. Acesso em: 03 fev. 2025.

KOFLER, F.; EZHOV, I.; ISENSEE, F.; BALSIGER, F.; BERGER, C.; KOERNER, M.; PAETZOLD, J.; LI, H.; SHIT, S.; McKINLEY, R.; BAKAS, S.; ZIMMER, C.; ANKERST, D.; KIRSCHKE, J.; WIESTLER, B.; MENZE, B. Are we using appropriate segmentation metrics? Identifying correlates of human expert perception for CNN training beyond rolling the DICE coefficient. Preprint, mar. 2021. Disponível em: https://www.researchgate.net/publication/349963393. Acesso em: 4 fev. 2025.

BILIC, Patrick et al. The Liver Tumor Segmentation Benchmark (LiTS). *Medical Image Analysis*, v. 84, p. 102680, 2023. DOI: 10.1016/j.media.2022.102680. Acesso em: 5 fev. 2025.

WANG, R.; LEI, T.; CUI, R.; ZHANG, B.; MENG, H. Medical image segmentation using deep learning: A survey. *IET Image Processing*, [s.l.], v. 16, n. 5, p. 1024-1038, 2022. DOI: 10.1049/ipr2.12419.

LITJENS, G. et al. A survey on deep learning in medical image analysis. Medical Image Analysis, v. 42, p. 60-88, 2017.

SZELISKI, R. Computer Vision: Algorithms and Applications. 2. ed. [S. l.]: Springer, 2022.