Segmentação de Fissuras Estruturais Utilizando uma Arquitetura Híbrida SegFormer-U-Net com Atenção Seletiva

Wesley Vieira de Santana
Universidade de Pernambuco
Recife, Brasil
wvs1@poli.br

Verusca Severo de Lima
Universidade de Pernambuco
Recife, Brasil
verusca.severo@poli.br

Francisco Madeiro
Universidade de Pernambuco
Recife, Brasil
madeiro@poli.br

***Resumo*—Neste trabalho, apresenta-se um modelo híbrido para segmentação de fissuras em imagens estruturais, combinando o *encoder SegFormer*-b2 com o *decoder* U-Net e integrando mecanismos de atenção seletiva. Foram avaliadas diferentes configurações de atenção, aplicadas de forma pontual nas camadas do *decoder*, com o objetivo de investigar seu impacto no desempenho da segmentação. A abordagem proposta alcançou o melhor desempenho com a configuração que aplica atenção de canal na primeira camada e atenção espacial na última camada do *decoder*, superando modelos de referência como o *Hybrid-Segmentor*. O modelo híbrido obteve Pontuação F1 de 0,789, *IoU* de 0,654 e mais de 52 FPS, apresentando bom compromisso entre desempenho de segmentação e velocidade de inferência. Os resultados demonstram que a atenção seletiva é uma estratégia eficaz para melhorar a segmentação sem aumentar significativamente a complexidade computacional do modelo.**

***Palavras-chaves*—Segmentação de imagem, *Transformer*, U-Net, *SegFormer*, Fissuras estruturais**

I. Introdução

Fissuras em pavimentos, paredes e estruturas de concreto representam um risco significativo à segurança e à durabilidade das infraestruturas civis. A identificação precoce dessas fissuras é essencial para o planejamento de ações de manutenção preventiva e para evitar falhas estruturais que possam acarretar elevados custos financeiros e riscos à integridade física da população [1], [2].

Tradicionalmente, a detecção de fissuras é realizada por inspeções visuais conduzidas por especialistas, de forma não automatizada, sendo um processo intensivo em mão de obra, sujeito à subjetividade e propenso a erros humanos [3].

Com os avanços em tecnologias de aquisição de imagens e o crescente uso de sistemas automatizados, como veículos aéreos não tripulados (UAVs - *Unmanned Aerial Vehicles*), câmeras embarcadas e sensores móveis, tornou-se viável aplicar técnicas de visão computacional e aprendizado profundo (DL - *Deep Learning*) para automatizar o processo de detecção e segmentação de fissuras [4]–[6].

A segmentação de fissuras, baseada em DL, especialmente por meio de Redes Neurais Convolucionais (CNNs - *Convolutional Neural Network*), destaca-se como uma abordagem eficaz para esse tipo de tarefa, pois permite reconhecer as fissuras em nível de *pixel* e extrair informações geométricas detalhadas, como largura, extensão e padrão da trinca [1]. Tais características são essenciais para estimar a severidade do dano e avaliar com precisão o estado de deterioração da estrutura. A segmentação utilizando modelos *Transformer*, também pertencentes ao contexto de DL, destaca-se pela capacidade de modelar contextos globais utilizando mecanismos de autoatenção, mas, em geral, demanda grandes quantidades de dados e recursos computacionais elevados [1].

Para explorar as vantagens dessas duas arquiteturas, isto é, CNNs com sua precisão espacial local, e *Transformers*, com sua capacidade de capturar contextos globais, abordagens híbridas vêm sendo propostas recentemente, combinando ambas para alcançar um equilíbrio entre detalhamento fino e compreensão global [3], [7]–[9].

Neste trabalho, apresenta-se uma arquitetura híbrida baseada em *SegFormer* (como *encoder*) e U-Net (como *decoder*), com o objetivo de aliar as vantagens das arquiteturas CNN e *Transformers*, visando alcançar bom desempenho de segmentação e complexidade computacional moderada na tarefa de segmentação automática de fissuras estruturais. O modelo híbrido apresentado busca atender tanto aos requisitos práticos quanto às restrições computacionais, tornando-o adequado para aplicações em dispositivos embarcados e inspeções em tempo real.

Este trabalho está estruturado da seguinte forma: a Seção II apresenta os fundamentos teóricos sobre segmentação de fissuras, modelos baseados em CNNs, *Transformers* e arquiteturas híbridas com mecanismos de atenção. A Seção III descreve a arquitetura proposta, detalhando os componentes do modelo híbrido e os módulos de atenção aplicados. A Seção IV apresenta as simulações realizadas, incluindo as configurações de treinamento, as métricas utilizadas e os resultados obtidos com diferentes configurações de atenção. Por fim, a Seção V apresenta as conclusões e sugestões de trabalhos.

II. Fundamentação Teórica

A. Segmentação Semântica de Fissuras

Na segmentação semântica, todos os *pixels* de uma imagem são rotulados com uma classe, por exemplo, fissura ou não

fissura, sem diferenciar entre diferentes instâncias de uma mesma classe [10]. Sendo assim, todas as fissuras presentes na mesma imagem são classificadas como pertencentes à mesma classe. Esse método é especialmente útil em tarefas práticas, pois fornece informações detalhadas como largura, extensão e padrões das fissuras, essenciais para avaliações estruturais e decisões de manutenção preventiva. A segmentação de fissuras pode ser denominada de segmentação binária, em que o modelo prevê apenas as classes fissura e não fissura [1]. A Figura 1 ilustra um exemplo típico de segmentação semântica binária, onde os *pixels* são rotulados como pertencentes ou não à classe de fissura.

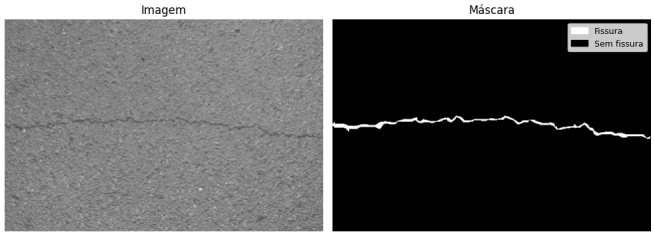


Figura 1: Exemplo de segmentação semântica binária: à esquerda, a imagem original do pavimento contendo fissura; à direita, a máscara de referência indicando os *pixels* classificados como fissura (branco) e sem fissura (preto). Imagem obtida a partir do *Dataset CrackForest* (CFD) [11].

Trabalhos recentes têm se concentrado na segmentação binária de fissuras devido à simplicidade e eficácia dessa abordagem em contextos práticos, especialmente quando a principal preocupação é identificar rapidamente a presença e localização geral das fissuras [3], [12]–[14].

B. Modelos Baseados em Redes Neurais Convolucionais (CNNs)

CNNs são especialmente projetadas para o processamento de dados de imagem devido à sua capacidade de capturar informações espaciais locais por meio da operação de convolução. Entre as arquiteturas baseadas em CNN amplamente utilizadas para segmentação de fissuras destacam-se a U-Net [15], [16], a FCN (*Fully Convolutional Network*) [17], [18] e a DeepLab [19], [20]. Dentre essas, arquiteturas como a U-Net e o DeepLabv3+ empregam uma estrutura *encoder-decoder*, visando recuperar as informações espaciais perdidas durante as operações de redução de resolução.

As FCNs foram pioneiras ao eliminar camadas totalmente conectadas, permitindo a produção de mapas de segmentação com a mesma resolução da imagem de entrada [18]. A U-Net, por sua vez, aprimora essa estrutura com conexões de atalho entre o *encoder* e *decoder*, o que permite preservar detalhes espaciais perdidos [15].

Estudos têm demonstrado a eficácia da U-Net na segmentação de fissuras, tanto em imagens completas quanto em pequenos blocos processados com janelas deslizantes [21], [22]. Contudo, CNNs tradicionais enfrentam limitações no campo receptivo, o que dificulta a captura de dependências espaciais mais extensas e contextos globais relevantes em imagens de fissuras complexas [1], [23].

C. Modelos Baseados em Transformers

Inicialmente introduzidos por Vaswani *et al.* [24] no contexto de tradução automática, os *Transformers* foram projetados como uma substituição para redes neurais recorrentes (RNNs - *Recurrent Neural Network*), com a vantagem de processar *tokens* de entrada em paralelo por meio de um mecanismo de autoatenção [24], [25].

Com o sucesso em tarefas de Processamento de Linguagem Natural (NLP – *Natural Language Processing*), surgiram modelos como o BERT (*Bidirectional Encoder Representations from Transformers*) [26] e o GPT-3 (*Generative Pre-trained Transformer 3*) [27], que escalaram massivamente o número de parâmetros e foram pré-treinados em grandes volumes de texto não rotulado, alcançando desempenho expressivo em diversas tarefas. *Transformers* têm ganhado destaque na área de segmentação de imagens devido ao mecanismo de autoatenção que lhes permite capturar dependências espaciais globais eficazmente [25].

Esse sucesso incentivou a adaptação dos *Transformers* para o domínio da visão computacional, resultando no desenvolvimento de modelos como o *Vision Transformer* (ViT) [28], *Swin-Transformer* [29], *Segmenter* [30] e *SegFormer* [31], adaptados especificamente para tarefas de segmentação semântica.

A seguir, destacam-se os principais componentes de um *Transformer* aplicado à visão computacional [25]:

- **Divisão em *Patches*:** A imagem de entrada é dividida em pequenos blocos (*patches*), que são achatados e transformados em vetores. Cada *patch* é tratado como um *token* visual, análogo a uma "palavra" em tarefas de processamento de linguagem natural.
- **Codificação Posicional:** Como os *Transformers* não modelam diretamente a posição dos elementos na entrada, vetores de posição são adicionados aos *embeddings* dos *patches* para incorporar informações espaciais à representação.
- **Camadas de Autoatenção (*Self-Attention*):** A autoatenção permite que cada *patch* interaja com os demais, ponderando sua importância relativa. Esse mecanismo possibilita capturar dependências de longo alcance na imagem. Para um vetor de entrada $\mathbf{x} \in \mathbb{R}^d$ (em que d é a dimensionalidade do *embedding*), são computados os vetores $\mathbf{Q}$ (*query*), $\mathbf{K}$ (*key*) e $\mathbf{V}$ (*value*) por meio de projeções lineares, com matrizes de pesos ajustadas durante o treinamento $W^Q, W^K, W^V \in \mathbb{R}^{d \times d_k}$, em que d_k é a dimensão da projeção e

$$\mathbf{Q} = \mathbf{x}\mathbf{W}^{\mathbf{Q}}, \quad \mathbf{K} = \mathbf{x}\mathbf{W}^{\mathbf{K}} \quad e \quad \mathbf{V} = \mathbf{x}\mathbf{W}^{\mathbf{V}}. \quad (1)$$

A pontuação de atenção é então dada por

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}(\mathbf{Q}\mathbf{K}^T)\mathbf{V}, \quad (2)$$

em que $\mathbf{K}^T$ representa a transposição da matriz *key* para calcular a similaridade entre pares de *patches*, resultando em um mapa de atenção que guia a ponderação dos valores.

- ***Feed-Forward Networks* (FFN):** Após a etapa de autoatenção, os vetores resultantes passam por redes neurais totalmente conectadas aplicadas individualmente a cada *token*, permitindo transformações não lineares adicionais.
- **Normalização e Conexões Residuais:** Tanto a autoatenção quanto a FFN são envoltas por camadas de normalização e conexões residuais, o que estabiliza o treinamento e facilita a propagação do gradiente.
- **Cabeças de Atenção Múltiplas (*Multi-Head Self-Attention*):** A atenção é dividida em múltiplas "cabeças", onde cada uma aprende a capturar diferentes padrões de interação entre os *patches*. Cada cabeça possui suas próprias projeções W_i^Q, W_i^K, W_i^V, sendo a saída agregada como:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O, \tag{3}$$

com

$$\text{head}_i = \text{Attention}(\mathbf{Q}W_i^Q, \mathbf{K}W_i^K, \mathbf{V}W_i^V), \tag{4}$$

em que W^O é a matriz de projeção final que transforma a concatenação das cabeças de volta para o espaço original de *embedding*.

Esse mecanismo permite ao modelo capturar simultaneamente múltiplas perspectivas de atenção, melhorando sua capacidade de representação sem perda da estrutura espacial global.

- **Decodificação para Segmentação:** As saídas dos *patches* são reorganizadas e decodificadas para formar o mapa de segmentação da imagem.

Diferentemente das CNNs, os *Transformers* não possuem um campo receptivo limitado, o que lhes permite capturar contextos globais em imagens de forma mais direta. O *SegFormer* é um exemplo eficiente dessa abordagem, projetado para segmentação com menor complexidade computacional e alta precisão contextual [3].

Os componentes dos *Transformers* são eficazes para modelar contexto global e interações de alto nível em imagens. No entanto, seu uso isolado pode demandar muitos dados de treinamento, o que motivou o surgimento de arquiteturas híbridas que conciliam sua capacidade contextual com a eficiência espacial das CNNs [25], [32].

D. Modelos Híbridos com Mecanismos de Atenção

Tanto as redes convolucionais quanto os *Transformers* puros têm se mostrado eficazes em tarefas de visão computacional, incluindo a segmentação de fissuras [3]. A combinação de ambas as abordagens tem mostrado resultados promissores em termos de métricas de desempenho na segmentação de fissuras. Por isso, muitos trabalhos estão focando em abordagens híbridas para atacar o problema da segmentacão de fissuras [7]–[9].

Além da combinação entre modelos CNN e *Transformers*, integrar mecanismos de atenção à estrutura de CNNs também apresenta bons resultados de segmentação. Três tipos principais de mecanismos de atenção são empregados às CNNs: atenção espacial (SA - *Spatial Attention*) [33], atenção por canal (CA - *Channel Attention*) [34] e sua combinação, como o *Dual Attention Network* [35]. Esses mecanismos focam em redistribuir os pesos das características extraídas, enfatizando regiões específicas da imagem (SA) ou canais mais informativos (CA).

Os cálculos utilizados nesses mecanismos de atenção são distintos. As Equações de (5) a (7) descrevem o módulo de atenção por canal, enquanto as Equações (8) e (9) apresentam o módulo de atenção espacial implementado neste trabalho, seguindo a estrutura proposta por [36].

Para o módulo de atenção por canal, aplica-se o *pooling* global médio ao longo das dimensões espaciais, seguido por duas camadas convolucionais com ativações ReLU e sigmoide

$$\mathbf{CA} = \sigma\left(\text{Conv}_2\left(\phi\left(\text{Conv}_1\left(\text{GAP}(\mathbf{F})\right)\right)\right)\right), \tag{5}$$

em que $\mathbf{F} \in \mathbb{R}^{C\times H\times W}$ representa o tensor de entrada (com C canais e dimensões espaciais $H \times W$), GAP é o *global average pooling*, ϕ é a função de ativação ReLU e σ é a função de ativação sigmoide. As camadas Conv_1 e Conv_2 são convoluções 1D que ajustam as dimensões intermediárias para gerar o vetor de atenção por canal $\mathbf{CA} \in \mathbb{R}^C$, com valores entre 0 e 1 para cada canal. O operador $\text{GAP}(\mathbf{F})$ calcula a média global por canal, isto é,

$$\text{GAP}(\mathbf{F})_c = \frac{1}{H \cdot W}\sum_{i=1}^{H}\sum_{j=1}^{W}\mathbf{F}_{c,i,j}. \tag{6}$$

O vetor $\mathbf{CA}$ é então aplicado ao tensor original por meio de produto elemento a elemento (denotado por $\odot$) para realçar os canais mais relevantes,

$$\mathbf{F}' = \mathbf{F} \odot \mathbf{CA}, \tag{7}$$

em que $\mathbf{F}' \in \mathbb{R}^{C\times H\times W}$ representa o tensor refinado após a aplicação da atenção por canal, com os canais ponderados de acordo com os pesos aprendidos em $\mathbf{CA}$.

Para a atenção espacial, aplicam-se médias e máximos ao longo da dimensão dos canais, concatenam-se os mapas resultantes e em seguida aplica-se uma convolução 2D com *kernel* 7×7, seguida de ativação sigmoide,

$$\mathbf{SA} = \sigma\left(\text{Conv}_{7\times 7}\left([\text{AvgPool}_c(\mathbf{F}); \text{MaxPool}_c(\mathbf{F})]\right)\right), \tag{8}$$

em que AvgPool_c e MaxPool_c realizam operações de *pooling* médio e máximo ao longo da dimensão dos canais. O mapa de atenção espacial $\mathbf{SA} \in \mathbb{R}^{H\times W}$ é então aplicado ao tensor original para enfatizar regiões espaciais relevantes,

$$\mathbf{F}'' = \mathbf{F} \odot \mathbf{SA}, \tag{9}$$

em que $\mathbf{F}'' \in \mathbb{R}^{C\times H\times W}$ representa o tensor final, que preserva a estrutura original, mas com regiões espaciais ponderadas segundo os pesos do mapa de atenção $\mathbf{SA}$.

O operador $\odot$ denota o produto elemento a elemento entre os tensores correspondentes.

III. Modelo Híbrido Proposto

O modelo proposto corresponde a uma arquitetura híbrida que combina o *encoder* baseado em *Transformer* da família *SegFormer*-B2 [31], com um *decoder* convolucional inspirado na U-Net [15], mantendo a estrutura hierárquica com camadas de *upsampling* e *skip connections* entre os estágios correspondentes do *encoder* e do *decoder*.

Com o intuito de reforçar a capacidade de realce de regiões de fissuras, são incorporados mecanismos de atenção no *decoder*. A atenção por canal (CA) é aplicada à camada mais profunda do *decoder*, enquanto a atenção espacial (SA) é empregada à saída do *decoder*, conforme descrito na Figura 2. A atenção por canal permite destacar os mapas de ativação mais relevantes entre os canais, enquanto a atenção espacial enfatiza as regiões mais informativas da imagem em cada etapa de reconstrução.

O *encoder* é responsável pela extração de características hierárquicas da imagem de entrada por meio de camadas com atenção eficiente presentes no *SegFormer*. A estrutura do *SegFormer* possibilita o processamento em múltiplas escalas, mantendo baixo custo computacional. Neste trabalho, é utilizada a versão *SegFormer*-B2 como *backbone* do *encoder*, por apresentar bom equilíbrio entre precisão de segmentação e viabilidade computacional. O *decoder*, por sua vez, reconstrói o mapa de segmentação em alta resolução a partir dessas características, utilizando blocos de *upsampling* com conexões de atalho.

Cada estágio do *decoder* é composto por duas etapas principais: uma operação de *upsampling* por convolução transposta (ConvTranspose2D), que dobra a resolução espacial do mapa de ativação, seguida por concatenação com o mapa de ativação correspondente do *encoder (skip connection)*. Após a concatenação, aplica-se um bloco convolucional composto por uma camada Conv2D, normalização por BatchNorm e ativação ReLU. Essa estrutura é repetida nos três estágios de decodificação. A atenção por canal é aplicada na camada mais profunda do *decoder*, logo após a primeira operação de *upsampling*, onde as informações extraídas pelo *encoder* são combinadas com as do primeiro *skip connection*. A atenção espacial, por sua vez, é empregada na saída do *decoder*, antes da última etapa de *upsampling* e da camada de previsão.

IV. Resultados

As simulações foram realizados em ambiente Google Colab Pro com GPU NVIDIA Tesla T4, 16 GB de memória, utilizando o *framework* PyTorch versão 2.0 e a biblioteca HuggingFace *Transformers* para acesso ao *encoder SegFormer*-B2.

O modelo foi avaliado sobre o *dataset* CrackVision12K [3], composto por 12.000 imagens com diferentes tipos de pavimento, iluminação e texturas e suas respectivas máscaras binárias de segmentação. A origem e estruturação desse *dataset* são descritas em [3], que propôs sua utilização em tarefas de segmentação fina de fissuras em infraestrutura civil. O

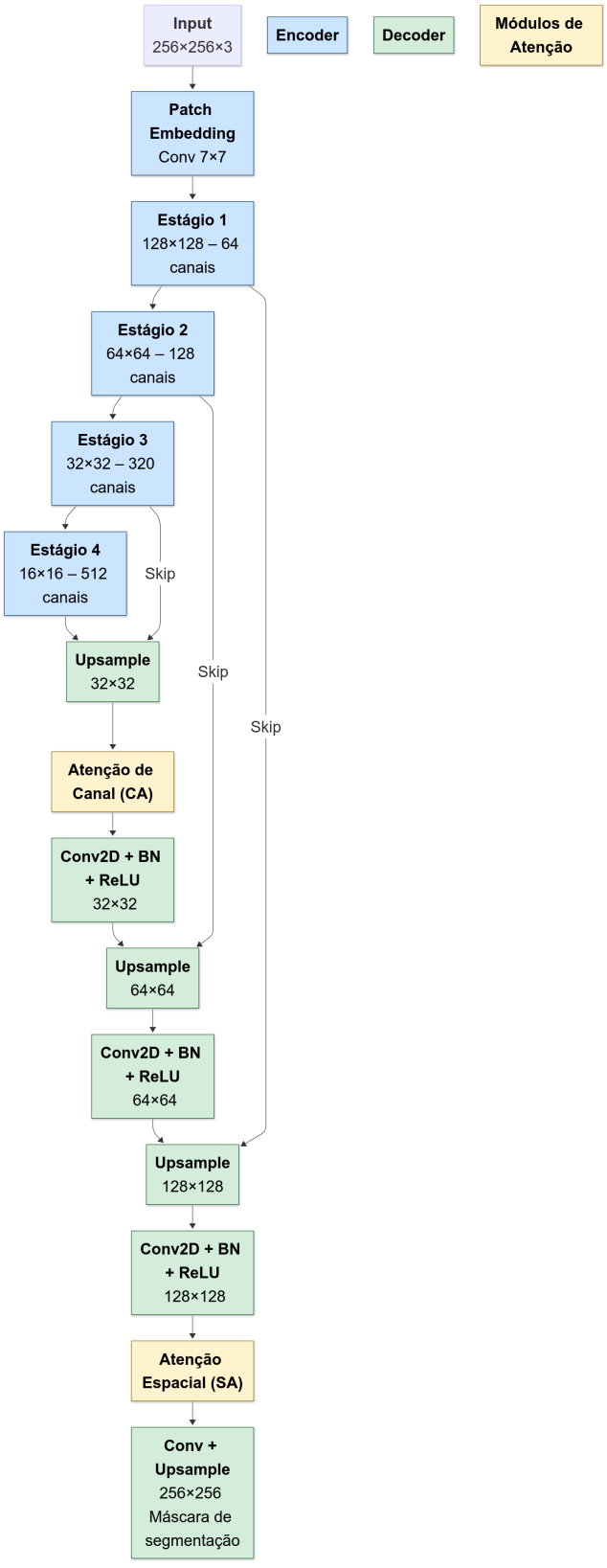


Figura 2: Fluxo da arquitetura proposta, composta por *encoder SegFormer*-B2 e *decoder* convolucional tipo U-Net. Os blocos do *encoder* são representados pela cor azul, *decoder* verde e mecanismos de atenção amarelo.

dataset foi aleatoriamente particionado em três subconjuntos: 9.600 imagens para treinamento (80%), 1.200 para validação (10%) e 1.200 para teste (10%), conforme proposto em [3].

O treinamento foi conduzido com taxa de aprendizado inicial de 1×10^{-4}, utilizando o otimizador AdamW com *weight decay* de 1×10^{-4}. O tamanho do lote foi fixado em 16, e o número total de épocas foi de 100, com *early stopping* aplicado com paciência de 10 épocas. Durante o treinamento, foi utilizada a técnica ReduceLROnPlateau para ajuste dinâmico da taxa de aprendizado com fator 0,5 e paciência de 3 épocas. Esses valores foram definidos com base no trabalho apresentado em [3] e confirmados empiricamente por meio de experimentação preliminar.

A função de perda utilizada foi a combinação composta pela ponderação do *Dice Loss* [37] (peso 0,8) e *Binary Cross-Entropy* (BCE) [38] (peso 0,2). Essa estratégia foi adotada com base nos resultados apresentados no trabalho de Goo *et al.* [3], buscando manter a comparabilidade dos resultados e replicar uma configuração eficaz para o *dataset* CrackVision12k.

A formulação matemática da função total de perda utilizada é dada por

$$\mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}_{\text{Dice}} + \beta \cdot \mathcal{L}_{\text{BCE}}, \tag{10}$$

em que $\alpha = 0{,}8$ e $\beta = 0{,}2$.

A função *Dice Loss* utilizada neste trabalho é definida como

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2 \cdot \sum_{i=1}^{N} p_i g_i + \epsilon}{\sum_{i=1}^{N} p_i + \sum_{i=1}^{N} g_i + \epsilon}, \tag{11}$$

em que $p_i \in [0, 1]$ representa a predição para o *pixel* i, $g_i \in [0, 1]$ é o valor correspondente na máscara de referência, N é o número total de *pixels*, e ϵ é um termo de estabilidade numérica, adicionado para evitar divisão por zero.

Essa função busca maximizar a sobreposição entre os *pixels* previstos como positivos e os reais, sendo especialmente eficaz em tarefas de segmentação com forte desbalanceamento entre classes.

A *Binary Cross-Entropy* (BCE) é expressa como

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N} \sum_{i=1}^{N} \left[g_i \cdot \log(p_i) + (1 - g_i) \cdot \log(1 - p_i) \right], \tag{12}$$

em que p_i é a probabilidade prevista para o *pixel* i, g_i o valor da máscara binária e N o número total de *pixels*.

A Tabela I resume as principais configurações adotadas durante o processo de treinamento, incluindo informações sobre a plataforma, hiperparâmetros e bibliotecas utilizadas.

Tabela I: Configurações utilizadas no treinamento dos modelos.

Configuração	**Parâmetro**
Plataforma	Google Colab Pro (GPU Tesla T4, 16GB RAM)
Tamanho das imagens	256×256 *pixels*
Tamanho do lote	16
Épocas	100
Interrupção	Paciência de 10 épocas
Otimizador	AdamW
Agendador	ReduceLROnPlateau (fator=0,5, paciência=3)
Função de perda	0,8 Dice + 0,2 BCE

Para avaliação do modelo híbrido proposto, foram considerados para comparação os modelos U-Net [15], *SegFormer*-B2 [31] e o *Hybrid-Segmentor*. O modelo *Hybrid-Segmentor*, proposto por Goo *et al.* [3], foi originalmente desenvolvido para segmentação refinada de fissuras para o *dataset* CrackVision12K. Sua arquitetura é composta por um *encoder* híbrido que combina a ResNet50 e uma adaptação do *SegFormer*.

Para avaliar o desempenho dos modelos de segmentação propostos neste trabalho, utilizamos um conjunto de métricas amplamente adotadas na literatura, a saber: Acurácia (*Accuracy*), Precisão (*Precision*), Sensibilidade (*Recall*), Pontuação F1 (*F1 Score*) e Interseção sobre União (IoU). Além disso, para fins de análise da complexidade computacional, também consideramos as métricas FPS (Quadros por segundos - *frames per second*), número de parâmetros do modelo (em milhões) e GFLOPs (giga operações de ponto flutuante por segundo - *giga floating point operations per second*).

Com base na arquitetura híbrida proposta, composta por um *encoder SegFormer*-b2 e um *decoder* do tipo U-Net, foram conduzidas simulações para avaliar o impacto da aplicação seletiva de mecanismos de atenção (CA e SA) nas camadas do *decoder*. Para este estudo, consideramos apenas as métricas de desempenho de segmentação.

A Tabela II apresenta os resultados obtidos considerando cinco diferentes combinações de atenção. As configurações de atenção avaliadas no *decoder* do modelo híbrido estão atribuídas da seguinte forma:

- **A1:** CA na 1ª camada, SA na última
- **A2:** CA apenas na 1ª camada
- **A3:** CA nas duas primeiras camadas e SA nas duas últimas
- **A4:** Apenas SA na última camada
- **A5:** Sem nenhum mecanismo de atenção

Tabela II: Desempenho do modelo híbrido com diferentes configurações de atenção (descritas de A1 a A5).

ID	**Acurácia**	**Precisão**	**Sensibilidade**	**F1**	**IoU**
A1	0,972	0,803	0,781	**0,789**	**0,654**
A2	0,972	0,796	0,784	0,787	0,652
A3	0,971	0,788	0,797	0,788	**0,654**
A4	0,972	0,800	0,781	0,788	0,653
A5	0,970	0,786	0,786	0,776	0,637

Com base nos resultados apresentados na Tabela II, observa-se que a configuração A1 apresentou o melhor desempenho, atingindo a maior Pontuação F1 (0,789) e um empate no IoU (0,654) com a configuração A3. Essa combinação seletiva de atenção parece contribuir para um melhor equilíbrio entre extração de características discriminativas e preservação de detalhes espaciais relevantes para a tarefa de segmentação.

As configurações A3 e A4 também demonstraram resultados competitivos, com Pontuação F1 de 0,788, sugerindo que o uso da atenção espacial, mesmo de forma isolada, contribui para a melhoria da segmentação.

Por outro lado, o modelo sem nenhum mecanismo de atenção A5 apresentou desempenho inferior, com o menor

Pontuação F1 (0,776) e menor *IoU* (0,637), evidenciando que a ausência de atenção impacta negativamente o desempenho do modelo quando comparado aos demais, até mesmo aos que aplicaram atenção de forma unitária, como em A2 e A4.

Esses resultados indicam que a aplicação seletiva de atenção no *decoder* pode ser uma estratégia eficaz para aprimorar a capacidade de segmentação do modelo.

A fim de validar a eficácia do modelo híbrido proposto, foi realizada uma comparação com outras arquiteturas de referência da literatura. A Tabela III apresenta os resultados em termos de Pontuação F1, IoU, FPS, número de parâmetros (em milhões) e GFLOPs.

- **M1:** Modelo híbrido proposto
- **M2:** *Hybrid-Segmentor*
- **M3:** SegFormerb2
- **M4:** SegFormerb2+U-Net (sem atenção)
- **M5:** U-Net

Tabela III: Resultados obtidos pelos modelos avaliados.

Modelo	Pontuação F1	IoU	FPS	Parâmetros (M)	GFLOPs
M1	**0,789**	**0,654**	52,34	27,4	5,8
M2	0,770	0,630	16,38	200	14,0
M3	0,731	0,580	27,39	27,3	14,0
M4	0,776	0,637	53,80	**25,4**	**4,8**
M5	0,735	0,584	**286,00**	31,0	48,0

A partir dos resultados apresentados na Tabela III, observa-se que o modelo proposto M1, obteve o melhor desempenho de segmentação entre os modelos avaliados, com destaque para a maior Pontuação F1 (0,789) e o maior IoU (0,654). Esses resultados evidenciam a efetividade da arquitetura híbrida baseada em *encoder SegFormer*-b2 e *decoder* U-Net com atenção seletiva.

O modelo M2, *Hybrid-Segmentor*, apresentou Pontuação F1 (0,770) próximo à do modelo proposto, porém com elevado custo computacional, 200 milhões de parâmetros e apenas 16,38 FPS. Já o modelo M3 (*SegFormer*-b2) obteve os piores resultados em termos de Pontuação F1 e IoU, destacando a importância do *decoder* e da introdução de atenção na arquitetura para tarefas de segmentação.

A Figura 3 apresenta um gráfico do tipo *scatter plot* que relaciona o desempenho dos modelos em termos de Pontuação F1, com sua velocidade de inferência, medida em FPS. O tamanho dos marcadores, que representam os modelos, é proporcional ao número de parâmetros do modelo, permitindo uma visualização conjunta dos principais aspectos que influenciam tanto o desempenho da segmentação quanto o custo computacional.

Observa-se que o modelo híbrido proposto ocupa uma posição privilegiada no gráfico, combinando alta Pontuação F1 com eficiência computacional. O modelo U-Net, apesar de ser o mais rápido em tempo de inferência, apresenta desempenho inferior em termos de segmentação. Por outro lado, o *Hybrid-Segmentor* atinge bons valores de Pontuação F1, mas seu elevado número de parâmetros e baixo FPS o torna menos viável para aplicações em tempo real.

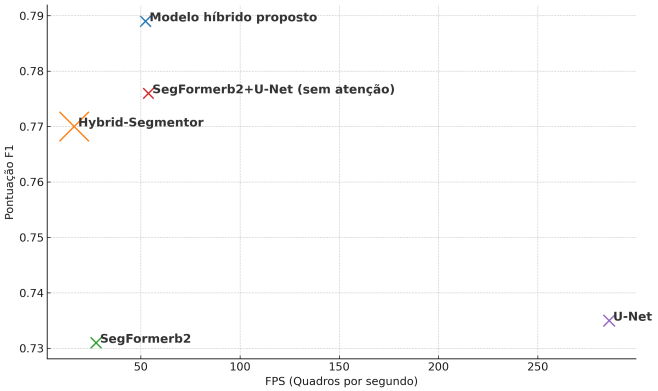


Figura 3: Relação entre Pontuação F1 e FPS.

A Figura 4 apresenta os resultados visuais de segmentação obtidos pelos modelos U-Net, *SegFormer*-b2 e o modelo híbrido proposto, considerando quatro exemplos com características visuais distintas. Observa-se que o modelo proposto apresenta máscaras com maior fidelidade às estruturas reais das fissuras comparando com a máscara de referência, com contornos mais contínuos e menos ruídos, quando comparado aos demais. O modelo U-Net, embora apresente bom desempenho em alguns casos, demonstra maior propensão a falsos positivos, especialmente em regiões com texturas irregulares. O *SegFormer*-b2, por sua vez, apresentou, em alguns casos, máscaras fragmentadas, o que pode comprometer a interpretação da extensão real da fissura.

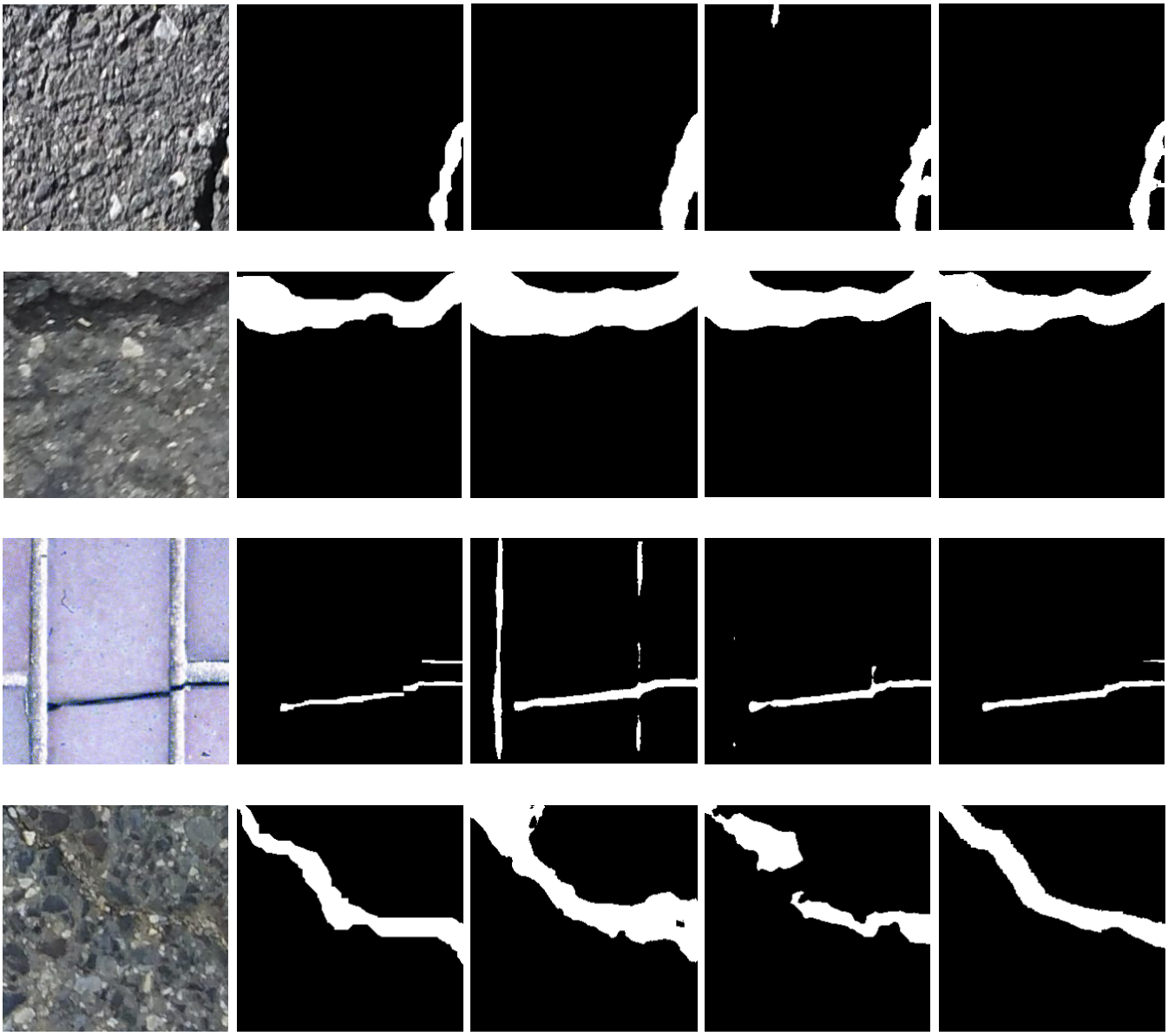
Por fim, nota-se que o modelo híbrido proposto consegue equilibrar a precisão na detecção com boa regularidade espacial, reforçando sua capacidade de generalização em diferentes tipos de pavimentos e superfícies.

V. Conclusões

Neste trabalho, foi proposto um modelo híbrido baseado na combinação do *encoder SegFormer*-b2 com o *decoder* U-Net, integrando mecanismos de atenção seletiva com o objetivo de aprimorar o desempenho na tarefa de segmentação de fissuras em imagens estruturais. A proposta foi avaliada por meio de diferentes configurações de atenção aplicadas seletivamente nas camadas do *decoder*.

Os resultados mostraram que a aplicação combinada de atenção de canal na primeira camada e atenção espacial na última (configuração A1) atingiu bom desempenho (maior Pontuação F1), mantendo uma complexidade computacional reduzida, superando inclusive modelos com maior número de parâmetros como o *Hybrid-Segmentor*. Além disso, verificou-se que mesmo configurações simplificadas com atenção parcial já proporcionam ganhos importantes quando comparado ao modelo sem o uso dos mecanismos de atenção.

Na comparação com outras arquiteturas de referência, o modelo híbrido proposto destacou-se com a maior Pontuação F1 (0,789) e IoU (0,654), mantendo baixo custo computacional e velocidade de inferência com mais de 52 FPS. Tais resultados o posicionam como uma alternativa para a tarefa



(a) Imagem original (b) Máscara de referência (c) U-Net (d) *SegFormer*-b2 (e) Proposto

Figura 4: Resultados visuais de segmentação para quatro exemplos distintos. As imagens da esquerda para a direita: (a) imagem original, (b) máscara de referência, (c) predição do modelo U-Net, (d) *SegFormer*-b2 e (e) o modelo híbrido proposto.

de segmentação de fissuras, especialmente para aplicações em dispositivos embarcados e sistemas com restrição de recursos.

Como trabalho futuro, pretende-se explorar a aplicação de módulos de atenção adaptativos, que aprendam automaticamente onde e quando aplicar atenção, bem como realizar a quantização e a poda do modelo para reduzir ainda mais a complexidade computacional e o espaço por ele ocupado em memória, visando a implantação em ambientes com recursos computacionais limitados.

Referências

[1] H. Gong, L. Liu, H. Liang, Y. Zhou, e L. Cong, "A state-of-the-art survey of deep learning models for automated pavement crack segmentation," *International Journal of Transportation Science and Technology*, vol. 13, pp. 44–57, 2024.

[2] J. König, M. D. Jenkins, M. Mannion, P. Barrie, e G. Morison, "Optimized deep encoder-decoder methods for crack segmentation," *Digital Signal Processing*, vol. 108, p. 102907, 2021.

[3] J. M. Goo, X. Milidonis, A. Artusi, J. Boehm, e C. Ciliberto, "Hybrid-segmentor: Hybrid approach for automated fine-grained crack segmentation in civil infrastructure," *Automation in Construction*, vol. 170, p. 105960, 2025.

[4] K. Teixeira, G. Miguel, H. S. Silva, e F. Madeiro, "A survey on applications of unmanned aerial vehicles using machine learning," *IEEE Access*, vol. 11, pp. 117 582–117 621, 2023.

[5] Y. Li, R. Ma, H. Liu, e G. Cheng, "Real-time high-resolution neural network with semantic guidance for crack segmentation," *Automation in Construction*, vol. 156, p. 105112, 2023.

[6] A. Neyestani, I. Ahmed, P. Daponte, e L. D. Vito, "Concrete crack detection and segmentation in civil infrastructures using UAVs and deep

learning," in *2023 7th International Conference on Internet of Things and Applications (IoT)*, 2023, pp. 1–6.

[7] H. Dong, Y. Du, D. Feng, Q. Hu, M. Zhou, J. Xing, Z. Long, W. Shu, e Y. Liu, "CSegNet: a hybrid transformer-CNN network for road crack image segmentation," *Insight - Non-Destructive Testing and Condition Monitoring*, vol. 66, pp. 737–746, 2024.

[8] M. He e T. L. Lau, "A novel network fusing transformer and CNN for road crack segmentation," *IEEE Access*, 2024.

[9] Y. Zhou, R. Ali, N. Mokhtar, S. W. Harun, e M. Iwahashi, "MixSegNet: A novel crack segmentation network combining CNN and transformer," *IEEE Access*, 2024.

[10] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, e D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, nº. 7, pp. 3523–3542, 2022.

[11] Y. Shi, L. Cui, Z. Qi, F. Meng, e Z. Chen, "Automatic road crack detection using random structured forests," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, nº. 12, pp. 3434–3445, 2016.

[12] Y. Kondo e N. Ukita, "Joint learning of blind super-resolution and crack segmentation for realistic degraded images," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–16, 2024.

[13] ——, "Crack segmentation for low-resolution images using joint learning with super- resolution," in *2021 17th International Conference on Machine Vision and Applications (MVA)*, 2021, pp. 1–6.

[14] H. Tao, B. Liu, J. Cui, e H. Zhang, "A convolutional-transformer network for crack segmentation with boundary awareness," in *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2023, pp. 86–90.

[15] O. Ronneberger, P. Fischer, e T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference*. Springer, 2015, pp. 234–241.

[16] X. Gao e B. Tong, "MRA-UNet: Balancing speed and accuracy in road crack segmentation network," *Signal, Image and Video Processing*, vol. 17, nº. 5, pp. 2093–2100, 2023.

[17] S. Wang, X. Wu, Y. Zhang, X. Liu, e L. Zhao, "A neural network ensemble method for effective crack segmentation using fully convolutional networks and multi-scale structured forests," *Machine Vision and Applications*, vol. 31, nº. 7, p. 60, 2020.

[18] J. Long, E. Shelhamer, e T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[19] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, e A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFS," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, nº. 4, pp. 834–848, 2017.

[20] X. Sun, Y. Xie, L. Jiang, Y. Cao, e B. Liu, "DMA-Net: Deeplab with multi-scale attention for pavement crack segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, nº. 10, pp. 18 392–18 403, 2022.

[21] M. D. Jenkins, T. A. Carr, M. I. Iglesias, T. Buggy, e G. Morison, "A deep convolutional neural network for semantic pixel-wise segmentation of road and pavement surface cracks," in *2018 26th European signal processing conference (EUSIPCO)*. IEEE, 2018, pp. 2120–2124.

[22] J. Cheng, W. Xiong, W. Chen, Y. Gu, e Y. Li, "Pixel-level crack detection using U-Net," in *TENCON 2018-2018 IEEE Region 10 Conference*. IEEE, 2018, pp. 0462–0466.

[23] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, e J. Garcia-Rodriguez, "A survey on deep learning techniques for image and video semantic segmentation," *Applied Soft Computing*, vol. 70, pp. 41–65, 2018.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, e I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[25] X. Li, H. Ding, H. Yuan, W. Zhang, J. Pang, G. Cheng, K. Chen, Z. Liu, e C. C. Loy, "Transformer-based visual segmentation: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[26] J. Devlin, M.-W. Chang, K. Lee, e K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 4171–4186.

[27] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[28] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, e N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[29] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, e B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[30] R. Strudel, R. Garcia, I. Laptev, e C. Schmid, "Segmenter: Transformer for semantic segmentation," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7262–7272.

[31] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, e P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 12 077–12 090.

[32] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, e M. Shah, "A comprehensive review of vision transformer-based methods for image segmentation," *Computer Vision and Image Understanding*, vol. 226, p. 103545, 2022.

[33] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[34] J. Hu, L. Shen, e G. Sun, "Squeeze-and-excitation networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[35] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, e H. Lu, "Dual attention network for scene segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3146–3154.

[36] S. Woo, J. Park, J.-Y. Lee, e I. S. Kweon, "CBAM: Convolutional block attention module," in *European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[37] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, e M. Jorge Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2017, pp. 240–248.

[38] M. Yi-de, L. Qing, e Q. Zhi-Bai, "Automated image segmentation using improved PCNN model based on cross-entropy," in *International Symposium on Intelligent Multimedia, Video and Speech Processing*. IEEE, 2004, pp. 743–746.