Segmentação semântica de ervas daninhas em plantações de cana-de-açúcar

Roney Felipe de Oliveira Miranda[1]

[1]Faculdade de computação – Universidade Federal do Mato Grosso do Sul (UFMS)

roney.miranda@ufms.br

Abstract. *This study presents a systematic comparison of five semantic segmentation architectures — U-Net, DeepLabV3+, PSPNet, Swin Transformer, and SegFormer — applied to weed segmentation in sugarcane fields using UAV orthoimages from Mato Grosso do Sul, Brazil. A dataset comprising 30,076 georeferenced patches of 256×256 pixels was created, focusing on the segmentation of castor bean (Ricinus communis). Among the evaluated models, SegFormer achieved the best performance, reaching 82.11% mIoU, 89.77% mAcc, and 99.38% aAcc on the test set. The analysis identified severe class imbalance as a major challenge, with background pixels dominating the dataset. The results indicate that architectures incorporating multi-scale context aggregation mechanisms are better suited to this agricultural domain. The results contribute to advancing precision agriculture in Brazil, enabling the optimization of agrochemical usage through intelligent management systems.*

Resumo. *O presente estudo apresenta uma comparação sistemática de cinco arquiteturas de segmentação semântica — U-Net, DeepLabV3+, PSPNet, Swin Transformer e SegFormer — aplicadas à segmentação de plantas daninhas em canaviais, com o uso de ortoimagens obtidas por VANTs em Mato Grosso do Sul. Foi criado um dataset composto por 30.076 parcelas georreferenciadas de 256×256 pixels, com foco na segmentação da mamona (Ricinus communis). Dentre os modelos avaliados, o SegFormer apresentou o melhor desempenho, alcançando 82,11% de mIoU, 89,77% de mAcc e 99,38% de aAcc no conjunto de teste. A análise identificou severo desbalanceamento entre classes como principal desafio, com predominância de pixels de background no dataset. Os resultados indicam que arquiteturas que incorporam mecanismos de agregação de contexto multi-escala são mais adequadas para o domínio agrícola em questão. Os resultados obtidos contribuem para o avanço da agricultura de precisão no Brasil, ao possibilitar a otimização do uso de defensivos agrícolas com base em sistemas de manejo inteligente.*

1. Introdução

O Brasil consolidou-se como líder mundial na produção de cana-de-açúcar com recorde histórico de 713,2 milhões de toneladas na safra 2023/24, o que representa 45% da produção global de açúcar e posiciona o país como maior exportador mundial do produto [CONAB 2024]. O setor sucroenergético brasileiro movimenta cerca de R$ 130 bilhões anuais, emprega 2,2 milhões de trabalhadores e contribui com 2% do PIB (Produto interno bruto) nacional [CNA 2024]. No Mato Grosso do Sul, quarto maior produtor nacional, a cultura ocupa 800 mil hectares distribuídos em 42 municípios, com

produção de 52,4 milhões de toneladas que representam 16% do PIB industrial estadual [SEMADESC 2024].

Apesar da relevância socioeconômica, a sucrocultura enfrenta severos desafios fitossanitários [EMBRAPA 2025]. O principal fator de redução da produtividade, por sua vez, são as plantas daninhas. Estudos da Empresa Brasileira de Pesquisa Agropecuária demonstram que as perdas podem variar de 40% a 85% do peso dos colmos, caule cilíndrico, em situações de alta infestação sem controle adequado, com prejuízos econômicos estimados em R$ 9 bilhões anuais para a agricultura brasileira [EMBRAPA 2025]. O período crítico de prevenção à interferência (PCPI) é a etapa em que as práticas de controle de plantas daninhas devem ser obrigatoriamente adotadas para evitar perdas significativas na produtividade da cultura principal. O PCPI estende-se de 20 a 150 dias após o plantio, variando conforme o ciclo da cultura, durante o qual a competição por água, luz, nutrientes e espaço causa reduções irreversíveis na produtividade [IAC 2024]. O amplo espaçamento entre fileiras nos estágios iniciais facilita a infestação; a similaridade visual entre cana e daninhas - fenômeno conhecido como "*green-on-green*", por sua vez, dificulta de forma significativa a identificação e o controle seletivo [Lynch 1995].

O manejo convencional baseado em aplicações de herbicidas em área total encara limitações crescentes. O Comitê de Ação à Resistência aos Herbicidas documentou 28 espécies de plantas daninhas com resistência confirmada no Brasil: 12 dessas já se mostram resistentes ao glifosato, principal herbicida utilizado na cultura de cana [HRAC-BR 2024]. Em paralelo, os impactos ambientais do uso intensivo de agrotóxicos tornaram-se cada vez mais uma preocupação central, com o registro de contaminação de recursos hídricos em áreas de produção canavieira pelo Ministério do Meio Ambiente [MMA 2024]. Além disso, a lei 14.785/2023 estabeleceu critérios mais rigorosos para registro e uso de agrotóxicos, demandando alternativas tecnológicas que mitiguem a dependência química [IBAMA 2024].

Desse modo, a agricultura de precisão emerge como solução transformadora para tais desafios, na qual tecnologias baseadas em sensoriamento remoto e inteligência artificial possibilitam reduções de 40% a 95% no uso de herbicidas através de aplicações localizadas, direcionadas apenas aos focos de infestação [MAPA 2024]. A integração de imagens de Veículos Aéreos Não Tripulados (VANTs) com algoritmos de aprendizado profundo revolucionou a segmentação automática de plantas daninhas, com redes neurais convolucionais *(CNN - Convolutional Neural Networks)* alcançando precisões superiores a 95% na distinção entre cultura e daninhas [EMBRAPA 2023]. Simultaneamente, o mercado brasileiro de *drones* agrícolas cresceu de 3 mil para 35 mil unidades após a regulamentação da Portaria do Ministério da Agricultura e Pecuária (MAPA) nº 298/2021, o que evidencia a rápida adoção dessas tecnologias [ANAC 2024].

A segmentação semântica *pixel*-a-*pixel* representa avanço fundamental neste contexto, viabilizando o desenvolvimento de pulverizadores inteligentes e robôs autônomos capazes de realizar controle seletivo em tempo real [Hu et al. 2024]. Arquiteturas de *deep learning* como U-Net, DeepLabV3+, Swin Transformer, SegFormer e PSPNet demonstram eficácia em diversos contextos agrícolas. Estudos recentes reportam valores de *Intersection over Union* (IoU) superiores a 80% utilizando U-Net em beterraba [Wang et al. 2024], DeepLabV3+ em soja [Yu et al. 2022], PSPNet em arroz [Anand et al. 2021] e Swin Transformer no reconhecimento de dani-

nhas [Zhang et al. 2023], além de avaliações gerais em culturas como milho e trigo [Silva et al. 2024]. Entretanto, a aplicação em cana-de-açúcar apresenta desafios únicos devido às características morfológicas da cultura, variabilidade das condições edafo-climáticas, condições especificas relacionadas ao solo e ao clima de uma determinada região que influenciam o desenvolvimento de organismos, brasileiras e diversidade de espécies invasoras presentes nos canaviais [Modi et al. 2023].

A carência de estudos comparativos sistemáticos destas arquiteturas para o contexto específico da cana-de-açúcar em condições tropicais representa lacuna significativa na literatura. As pesquisas existentes, em sua maioria, avaliam modelos isolados ou utilizam *datasets* limitados que não refletem a complexidade dos sistemas produtivos brasileiros [Subeesh et al. 2023]. De modo complementar, aspectos cruciais como eficiência computacional para implementação em sistemas embarcados, robustez a variações de iluminação e escala, e capacidade de generalização entre diferentes regiões produtoras permanecem pouco explorados [Gao et al. 2024].

A escolha destas arquiteturas específicas fundamenta-se em uma análise estratégica que busca contrapor diferentes paradigmas de visão computacional. O conjunto selecionado abrange desde redes convolucionais consolidadas (*U-Net*, *DeepLabV3+* e *PSPNet*), reconhecidas pela eficiência na captura de contexto local e multi-escala, até modelos baseados em *Transformers* (*Swin Transformer* e *SegFormer*), que exploram dependências globais e mecanismos de atenção. Essa diversidade permite avaliar qual abordagem oferece o melhor equilíbrio entre acurácia e robustez para lidar com a complexidade visual e o desbalanceamento de classes típicos dos canaviais brasileiros.

O presente trabalho visa preencher as lacunas supracitadas através de uma abordagem experimental rigorosa. O objetivo geral deste estudo é realizar uma análise comparativa sistemática de cinco arquiteturas de segmentação semântica (*U-Net*, *DeepLabV3+*, *PSPNet*, *Swin Transformer* e *SegFormer*) aplicadas à segmentação de plantas daninhas em ortoimagens de canaviais.

Para alcançar este propósito, foram definidos os seguintes objetivos específico:

- Construir e disponibilizar uma base de dados rotulada de imagens aéreas de cana-de-açúcar com a presença da espécie mamona (*Ricinus communis*);
- Treinar e validar os modelos selecionados sob as mesmas condições experimentais para garantir a isonomia da comparação;
- Avaliar a eficácia de cada arquitetura através de métricas de segmentação (mIoU, mAcc, aAcc), verificando a capacidade de generalização de classes;
- Analisar a eficiência computacional dos modelos, comparando os tempos de inferência para discutir a viabilidade de implementação prática.

Os resultados obtidos permitirão identificar as técnicas mais adequadas ao contexto brasileiro, fornecendo subsídios para o desenvolvimento de sistemas de manejo inteligente mais sustentáveis e eficientes.

2. Trabalhos Relacionados

2.1. *Deep Learning* em Diferentes Culturas

A aplicação de técnicas de *deep learning* para segmentação de plantas daninhas tem demonstrado eficácia crescente em diversas culturas agrícolas. Gao et al. (2024) demons-

traram transferência efetiva de aprendizado entre imagens terrestres e de VANTs em culturas de milho, com DarkNet53 alcançando IoU de 0.577 para a classe de daninha. Em arroz, métodos baseados em *CNNs (Convolutional Neural Networks)* superaram análises *OBIA (Object-Based Image Analysis)* tradicionais em 23%, com precisões de 85-94% [Huang et al. 2020]. O weedNet alcançou 96% de precisão em beterraba usando fusão de canais Infravermelho Próximo *(Near-Infrared - NIR)* e RGB.[Sa et al. 2018].

Para cana-de-açúcar, Modi et al. (2023) compararam seis arquiteturas em *dataset* experimental indiano, com DarkNet53 atingindo 99% de precisão e F1-*score* de 0.994. Entretanto, a *performance* cai para 87% em plantas jovens devido à similaridade morfológica. Sujaritha et al. (2017) desenvolveram sistema robótico com 92.9% de acurácia em condições controladas, mas apenas 78% em campo devido a variações de iluminação.

Da mesma forma, estudos brasileiros identificaram desafios específicos: Silva et al. (2024) demonstraram que a presença de palha residual (10-20 t/ha) limita a segmentação de modelos convencionais a 67%, contudo, o desempenho é elevado para 89% com a incorporação de módulos de atenção espacial. A UFSC desenvolve um *dataset* público com ortomosaicos de 5cm/*pixel*, revelando que 73% dos erros ocorrem em bordas de talhões [von Wangenheim et al. 2023]. Assim como, trabalhos recentes exploraram abordagens inovadoras, como Ge et al. (2023) que introduziu o WeedNet-R com precisão média *(mean Average Precision - mAP)* de 0.827 operando a 15 Quadros Por Segundo *(Frames Per Second - FPS)* e Wang et al. (2020) que desenvolveram *enhancement* adaptativo que mantém 91% da precisão em iluminação extrema, *versus* 73% sem pré-processamento.

A variabilidade de resultados - mIoU de 57% a 99% dependendo da cultura e condições - evidencia a inexistência de uma solução universal, justificando a necessidade de avaliação específica para cana-de-açúcar brasileira proposta por esse trabalho.

2.2. Arquiteturas de Segmentação Semântica

A seleção das cinco arquiteturas - *U-Net, DeepLabV3+, PSPNet, Swin Transformer* e *SegFormer* - fundamenta-se em uma análise criteriosa que buscou equilibrar diversidade arquitetural, eficácia comprovada e viabilidade prática. Tal escolha deliberada garante representação abrangente dos paradigmas atuais de segmentação semântica, desde abordagens com CNNs tradicionais até *transformers* de última geração, o que viabiliza identificar a solução mais adequada para os desafios específicos da segmentação de plantas daninhas em plantações de cana-de-açúcar no Brasil.

A diversidade arquitetural constituiu o primeiro pilar desta seleção, assegurando que diferentes metodologias de processamento de imagem fossem avaliadas. Enquanto *U-Net* [Ronneberger et al. 2015] e PSPNet [Zhao et al. 2017] representam evoluções distintas das redes convolucionais, onde a primeira prioriza eficiência através de simplicidade e a última enfatiza captura de contexto multi-escala. Por sua vez, a DeepLabV3+ [Chen et al. 2018a] estabelece o limite superior de *performance* das CNNs através de suas convoluções atrosas.

No mesmo sentido, verifica-se que o *Swin Transformer* [Liu et al. 2021] e *SegFormer* [Xie et al. 2021] exploram o paradigma *transformer* sob perspectivas contrastantes: complexidade hierárquica *versus* simplicidade unificada. Esta heterogeneidade permite comparação direta de desempenho em termos de métricas de segmentação (mIoU, mAcc,

aAcc) e análise do comportamento de convergência durante o treinamento, estabelecendo bases para futuras investigações sobre eficiência computacional e requisitos de *hardware*.

A eficácia comprovada em aplicações agrícolas semelhantes sustentou cada escolha com base em resultados experimentais, onde estudos recentes demonstram que essas arquiteturas superam consistentemente alternativas em tarefas de segmentação de plantas. A exemplo disso, pode-se citar Wang et al. (2024), que reportou 84,28% de mIoU ao utilizar a U-Net, mantendo latência menor em comparação a arquiteturas mais complexas. De modo semelhante, Chen et al. (2018) demonstraram que a DeepLabV3+ alcança desempenho superior em soja sob diferentes densidades de plantio. Por sua vez, a PSPNet obteve 90% de frequency weighted IoU em arroz, segundo Anand et al. (2021), enquanto a Swin Transformer [Zhang et al. 2023] atingiu 99,18% de acurácia em *datasets* com múltiplas espécies de plantas daninhas. Por fim, o *SegFormer*, segundo Liu et al. (2024), demonstrou 92% de desempenho mantido em cenários de transferência entre domínios (de imagens aéreas para terrestres).

Além disso, o desempenho documentado de arquiteturas convolucionais como DarkNet53, FCN e WeedNet em culturas *row-crop* (plantações cultivadas em fileiras largas), como milho, soja e beterraba [Gao et al. 2024; Huang et al. 2020; Sa et al. 2018], foi fundamental para compreender que essas espécies compartilham desafios visuais com a cana-de-açúcar, como alta densidade foliar, sobreposição de plantas e variabilidade morfológica entre estágios de crescimento.

A viabilidade de implementação em sistemas reais completou os critérios de seleção, reconhecendo que soluções acadêmicas devem transcender o laboratório para gerar impacto prático. Todas as arquiteturas selecionadas possuem implementações otimizadas em *frameworks* modernos como *PyTorch* e *TensorFlow*, com suporte para quantização e *pruning* que viabilizam *deployment* em *hardware* com recursos limitados[Hu et al. 2024]. Embora análises detalhadas de eficiência computacional e requisitos de *hardware* estejam além do escopo deste trabalho inicial, a disponibilidade de modelos pré-treinados e documentação extensiva estabelece fundamentos sólidos para futuras otimizações específicas ao contexto de VANTs agrícolas.

2.2.1. U-Net

U-Net foi selecionada para integrar esse projeto por três razões fundamentais. Primeiro, sua arquitetura simétrica com *skip connections* (Figura 1) demonstrou capacidade única de preservar detalhes finos mesmo com *downsampling* agressivo - característica essencial para detectar plantas daninhas pequenas em imagens de alta resolução [Ronneberger et al. 2015]. Segundo, a eficiência computacional (7.76 GFLOPs) viabiliza *deployment* em *drones* agrícolas com processadores ARM, cenário comum em operações brasileiras. Terceiro, a U-Net demonstrou robustez excepcional com *datasets* limitados [Ronneberger et al. 2015], apresentando desempenho consistente mesmo com conjuntos reduzidos (cerca de 500 imagens) – realidade frequente em projetos agrícolas iniciais [Wang et al. 2024].

Dentre as pesquisam que utilizam o U-net, pode-se citar Wang et al. (2024) ao demonstrarar que U-Net supera arquiteturas mais complexas em cenários com restrições

computacionais, mantendo 84.28% mIoU com latência 8x menor que SegFormer-B5. A arquitetura também permite modificações incrementais - adição de módulos de atenção, alteração de *encoders* - sem comprometer a estabilidade do treinamento.

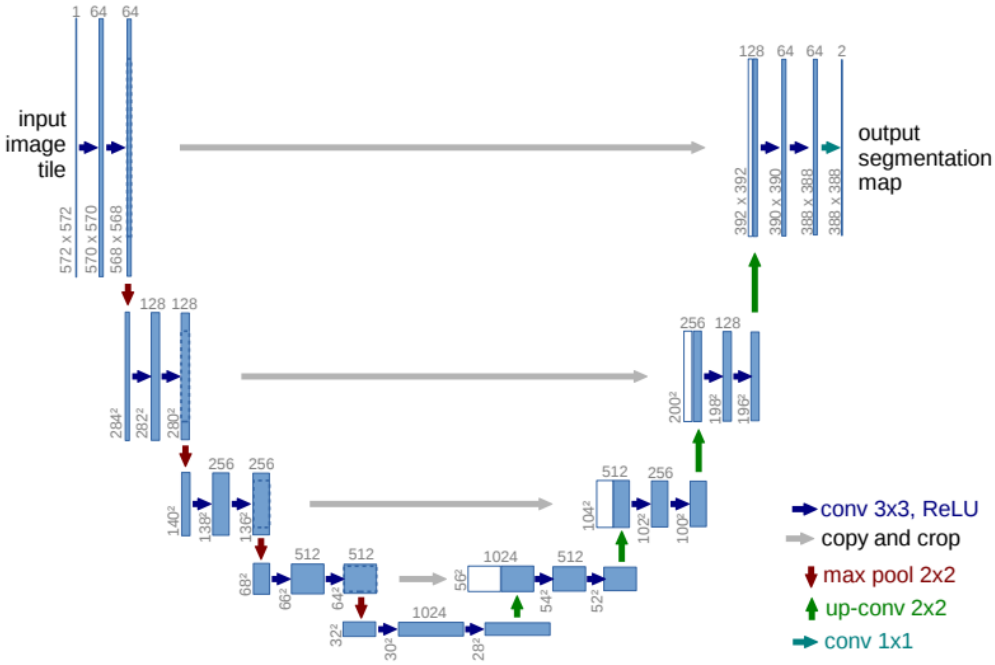


Figura 1. Arquitetura da U-Net [Ronneberger et al. 2015].

2.2.2. DeepLabV3+

DeepLabV3+ representa uma referência consolidada de alto desempenho em segmentação semântica baseada em CNNs, justificando sua inclusão por capacidades únicas de processamento multi-escala. O módulo *ASPP (Atrous Spatial Pyramid Pooling)* com taxas de dilatação $\{6, 12, 18\}$ captura simultaneamente características de plantas individuais (escala local) e padrões de infestação (escala global) - dualidade crítica em manejo de plantas daninhas [Chen et al. 2018a].

A arquitetura demonstrou, de forma consistente, os melhores resultados em culturas *row-crop* similares à cana-de-açúcar. Yu et al. (2022) reportaram que DeepLabV3+ mantém *performance* superior mesmo com variações de 3x na densidade de plantas, situação comum entre diferentes variedades de cana-de-açúcar e estágios de crescimento. A compatibilidade com diversos *backbones* (ResNet, Xception, Swin) oferece flexibilidade para otimização específica sem alteração da arquitetura base. A visualização das convoluções atrosas pode ser vista na figura 2.

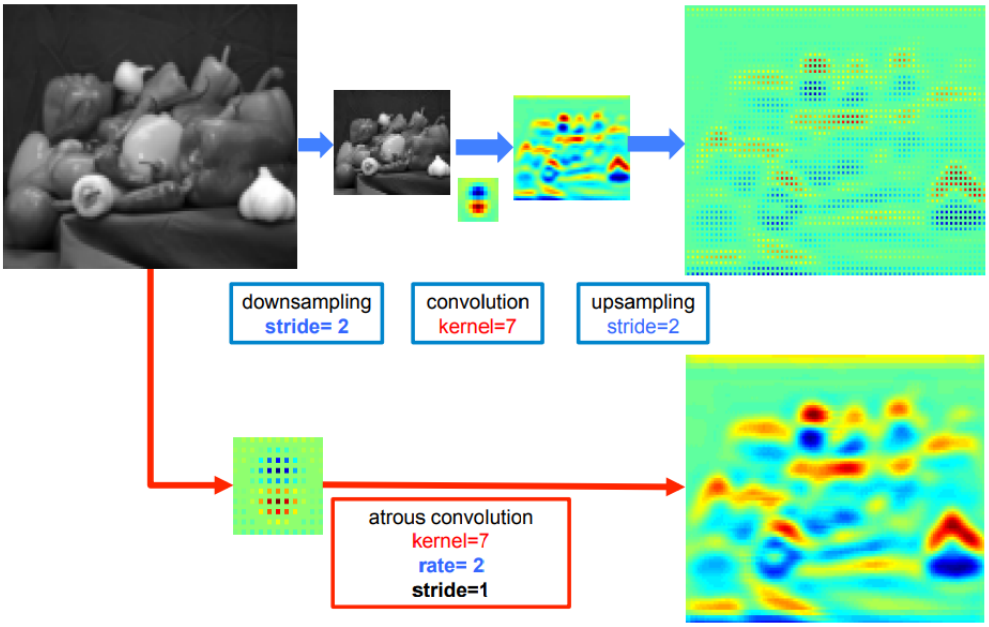


Figura 2. Visualização de convuluções atrosas [Chen et al. 2018a].

2.2.3. PSPNet

PSPNet foi incluída devido a sua capacidade de capturar contexto em múltiplas escalas através do módulo *pyramid pooling*, ilustrado na figura 3. Tal característica provou-se fundamental em canaviais onde a disposição regular das fileiras cria padrões geométricos que auxiliam na distinção entre cultura e plantas invasoras [Zhao et al. 2017]. O *pooling* em quatro níveis (1×1, 2×2, 3×3, 6×6) captura desde texturas locais até estrutura global do campo.

Desse modo, a escolha é reforçada por resultados em culturas com características similares, como arroz onde a PSPNet superou U-Net e SegNet em cenários com alta densidade de plantas, alcançando 90% *frequency weighted* IoU [Anand et al. 2021]. A arquitetura apresentou robustez aprimorada diante de oclusões parciais, característica crítica em estágios avançados da cana-de-açúcar, nos quais ocorre sobreposição intensa de folhas.

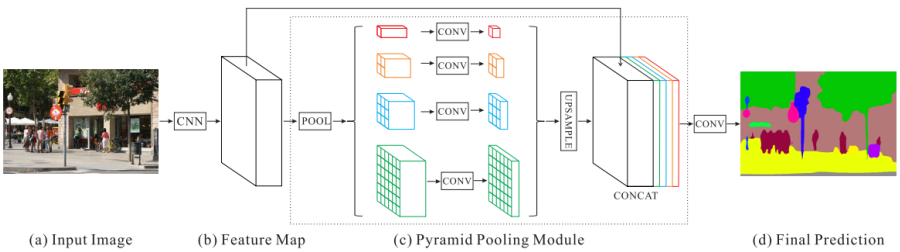


Figura 3. Arquitetura da PSPNet [Zhao et al. 2017].

2.2.4. Swin Transformer

A inclusão de *Swin Transformer* representa a necessidade de avaliar o paradigma *transformer* no contexto específico de cana-de-açúcar. Três características justificam esta escolha:

(i) Complexidade linear O(n) permite processamento de imagens de altíssima resolução sem *downsampling*;
(ii) Mecanismo de *shifted windows* captura dependências de longo alcance mantendo eficiência computacional;
(iii) Hierarquia de *features* intrinsecamente coerente com a natureza multi-escala dos campos agrícolas [Liu et al. 2021].

Ademais, estudos como Zhang et al. (2023) demonstraram que *Swin Transformer* alcança 99.18% de acurácia em *datasets* com 15+ espécies de plantas daninhas, superando CNNs em 12% quando múltiplas espécies coexistem - cenário típico em canaviais brasileiros. A arquitetura, figura 4, também mostrou degradação mínima (menor que 5%) quando aplicada a imagens capturadas em diferentes resoluções, eliminando necessidade de reescalonamento.

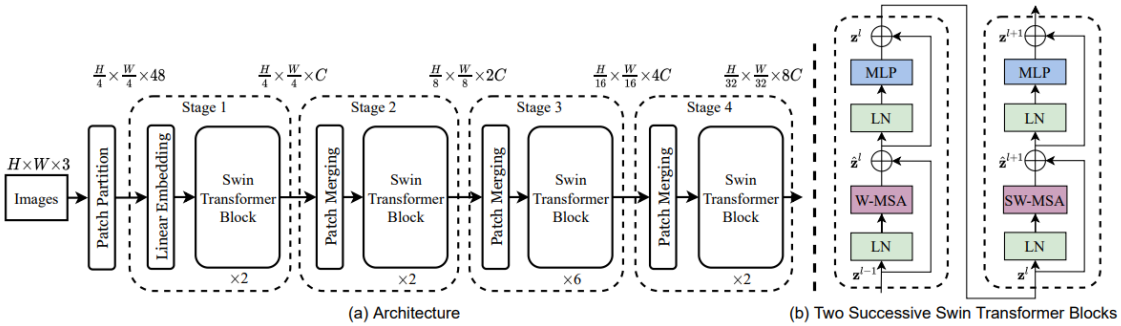


Figura 4. Arquitetura da Swin [Liu et al. 2021].

2.2.5. SegFormer

SegFormer representa a nova geração de *transformers* otimizados para segmentação, ao basear sua seleção em três vantagens únicas:

(i) Ausência de *positional encoding* permite generalização natural entre resoluções;
(ii) *Design* unificado elimina componentes complexos, reduzindo pontos de falha;
(iii) Família escalonável (B0–B5) permite *trade-off* preciso entre acurácia e eficiência [Xie et al. 2021].

Para o contexto agrícola, *SegFormer* demonstrou capacidade excepcional de transferência entre domínios visuais distintos (aéreo e terrestre), no qual modelos treinados em imagens aéreas mantiveram 92% da performance quando aplicados a imagens terrestres sem *fine-tuning* [Liu et al. 2024]. Esta robustez é crítica em operações práticas onde diferentes sensores e plataformas são utilizados. SegFormer-B0, com apenas 3.7M parâmetros, viabiliza *deployment* em dispositivos *edge* mantendo mIoU competitivo com modelos 20x maiores. A visualização da arquitetura do modelo pode ser vista na figura 5.

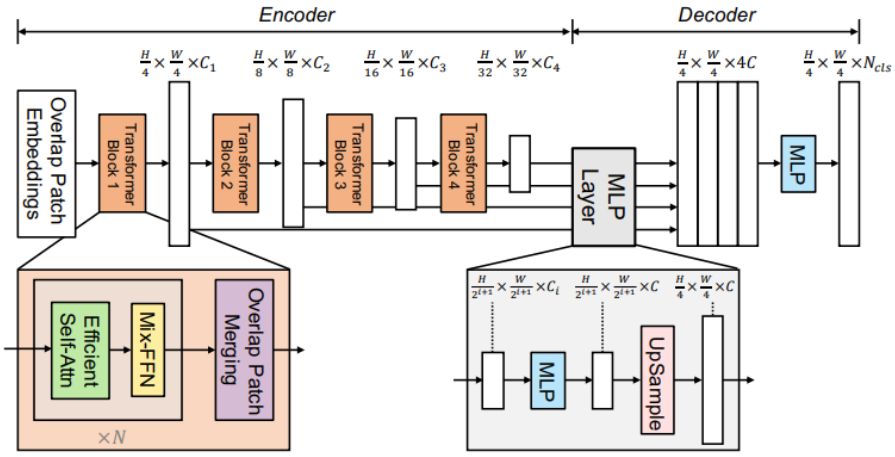


Figura 5. Arquitetura da SegFormer [Xie et al. 2021].

2.3. Aplicação Prática e Lacunas

A seleção conjunta destas cinco arquiteturas garante cobertura abrangente do espaço de soluções. *U-Net* e *PSPNet* representam abordagens CNN tradicionais com ênfases distintas (eficiência *vs.* contexto), enquanto *DeepLabV3+* estabelece o limite superior de *performance* CNN e, por fim, *Swin Transformer* e *SegFormer* exploram o paradigma *transformer* com metodologias diferentes (complexidade hierárquica *vs.* simplicidade unificada).

Essa diversidade permite análise robusta de *trade-offs* críticos: eficiência computacional *versus* acurácia, requisitos de dados *versus* capacidade de generalização, complexidade arquitetural *versus* facilidade de *deployment*. Para o contexto específico de cana-de-açúcar brasileira, onde condições variam drasticamente entre regiões e estações, esta análise comparativa abrangente é essencial para identificar soluções práticas e escaláveis.

3. Metodologia

3.1. Caracterização das Espécies de Daninhas

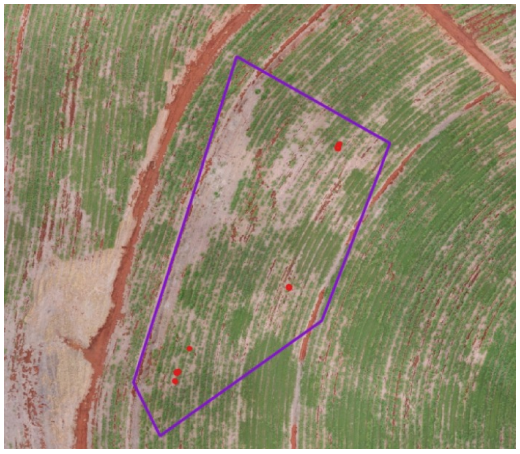
Os dados experimentais consistem em ortoimagens RGB georreferenciadas adquiridas através de voos de VANT realizados no estado de Mato Grosso do Sul. Ressalta-se que este conjunto de dados é de autoria dessa pesquisa e distinto da base pública da UFSC citada na revisão bibliográfica. As imagens originais apresentam resoluções espaciais variadas (*Ground Sampling Distance* - GSD), decorrentes de diferentes altitudes de voo. Essa variabilidade foi mantida intencionalmente para enriquecer o banco de dados, permitindo avaliar a robustez dos modelos frente a alterações de escala típicas de operações reais.

O desenvolvimento dessa pesquisa se iniciou na percepção de quais ervas daninhas estavam presentes nas imagens recebidas para análise. Ao todo, identificou-se a existência de 4 tipos, sendo elas: mamona (*Ricinus communis*), colonião (*Panicum maximum*), folha larga (*Ipomoea* spp.) e gramínea (*Digitaria horizontalis*). Dentre as experiências mencionadas, a espécie mamona (*Ricinus communis*) foi selecionada para este estudo. Optou-se por modelar o problema como uma tarefa de segmentação semântica binária (Mamona e

Background), em detrimento de uma abordagem multiclasse ou de classificação simples, por duas razões estratégicas. Primeiro, a segmentação pixel-a-pixel fornece a localização espacial exata e a geometria da infestação, dados indispensáveis para o cálculo da dosagem em sistemas de pulverização localizada – nível de detalhe que a classificação de imagens não é capaz de oferecer. Segundo, a modelagem binária permite isolar o desempenho das arquiteturas na segmentação da espécie alvo principal, estabelecendo um fundamento sólido antes da introdução da complexidade de classes concorrentes. Pesquisas futuras contemplarão a expansão para as três espécies remanescentes.

Após o entendimento das caracteristicas da espécie mamona de daninha, a rotulação das ortoimagens foi inciada dentro do software QGIs, com a demarcação de uma área que contém as daninhas. As plantas invasoras foram identificadas e marcadas olho a olho dentro da área delimitada. No final do processo, 7 ortoimagens haviam sido rotuladas, conforme representação na Figura 6.

(a) Ortoimagem



(b) Área delimitada

(c) Mamona rotulada

Figura 6. Representação da área de estudo: (a) ortoimagem completa, (b) área delimitada e (c) mamona rotulada.

Após a construção das máscaras de rotulação, iniciou-se o processo de extração das parcelas via script Python. O algoritmo percorre todas as regiões delimitadas nas máscaras, realizando recortes (parcelas) de 256×256 *pixels*. O valor "0"foi atribuído para pixels da parcela extraída correspondentes ao solo (*background*). O valor 1, para *pixels* da classe de interesse. O valor 255 indica regiões excluídas do treinamento por não contribuírem para a aprendizagem do modelo, como bordas ou áreas ambíguas. Com o intuito de facilitar a visualização e validação do processo, aplicou-se uma paleta de cor vermelha aos pixels de valor 1, o que permite identificar com clareza as regiões correspondentes à mamona quando sobrepostas à ortoimagem original. Ao final do processo, obtiveram-se 30.076 imagens de 256x256 que correspondem às parcelas salvas em *jpg*, assim como o mesmo valor de imagens 256x256 correspondendo às suas respectivas máscaras de rotulação salvas em *png*. As parcelas e máscaras são salvas com o nome da ortoimagem de onde foram extraídas seguidas de um contador único para diferenciá-las. Exemplos das imagens extraídas e suas máscaras são apresentados na Figura 7

(a) Mamona (b) Máscara de rotulação (c) Imagem com pixels 255

Figura 7. Exemplos de imagens extraídas.

A etapa de corte das parcelas e da máscara de rotulação é de suma importância no *pipeline* de desenvolvimento dos modelos, pois é a partir da sobreposição da máscara sob as parcelas que o modelo vem a reconhecer o que são as mamonas.

Assim que as parcelas e as máscaras foram recortadas, a divisão do *dataset* foi iniciada, fazendo uso de outro *script* em Python que analisa todas as imagens extraídas em busca do nome das ortoimagens, com o intuito de identificar quais ortoimagens originam aquelas parcelas e máscaras. Com tal informação, o *script* distribui de forma aleatória as ortoimagens entre os conjuntos de treino, validação e teste, de modo que parcelas provenientes de uma mesma ortoimagem não sejam utilizadas em mais de um conjunto. Dessa forma, garante-se que ortoimagens usadas na validação não sejam vistas no treino, e que as de teste permaneçam inéditas nos demais conjuntos.

Embora se buscasse alcançar a convenção de 70/20/10 (treinamento/validação/teste), a natureza da base de dados — especialmente a variação no tamanho das ortoimagens — resultou em uma distribuição final de 24.965 parcelas para treino, 4.002 para validação e 1.109 para teste, como pode ser visto na figura 8. Desse modo, 4 ortoimagens foram utilizadas para treino, 2 para validação e uma para teste.

Tabela 1. Distribuição quantitativa de ortoimagens e parcelas (*patches*) por conjunto de dados.

Conjunto	Qtd. Ortoimagens	Qtd. Parcelas	Proporção (%)
Treinamento	4	24.965	83,0
Validação	2	4.002	13,3
Teste	1	1.109	3,7
Total	**7**	**30.076**	**100,0**

Em seguida, a aplicação dos modelos foi realizada utilizando a biblioteca *OpenMMLab/mmsegmentation*, escolhida por oferecer um ambiente completo e estável de desenvolvimento, execução e observação. Para integrar o novo *dataset* ao framework, os seguintes artefatos foram criados:

- `/configs/base/datasets/daninhasMamona.py`: responsável por definir os *dataloaders* para os conjuntos de dados;
- `mmseg/datasets/daninhasMamona.py`: define a classe e o formato do *dataset*;
- `mmseg/datasets/__init__.py`: importa a classe do *dataset* para que possa ser localizada e utilizada pelo sistema;
- `/configs/nomeDoModelo/VersaoDoModelo.py`: arquivo de configuração do modelo, onde se especifica o *dataset*, otimizadores e hiperparâmetros.

Para garantir uma comparação justa entre as arquiteturas avaliadas, todos os modelos foram configurados com os mesmos hiperparâmetros. As principais definições utilizadas durante o treinamento foram:

- **Otimizador:** *Stochastic Gradient Descent (SGD)* com taxa de aprendizado inicial de 0,01, momentum de 0,9 e *weight decay* de 0,0005 para regularização L2;
- **Agendador de taxa de aprendizado:** *PolyLR*, que reduz de forma gradual a taxa de 0,01 para 0,0001 ao longo de 20.000 iterações, seguindo uma função polinomial com expoente 0,9;
- **Parâmetros de treinamento:** lotes de 16 imagens, utilizando 2 *workers* para carregamento paralelo;
- **Parâmetros de validação/teste:** uma imagem processada por vez, com 4 *workers*, para avaliação mais precisa do desempenho individual;
- **Segmentação e perdas:** tarefa binária (2 classes), com pixels de valor 255 ignorados na função de perda, representando áreas não rotuladas nas ortoimagens.

Para a avaliação definitiva no conjunto de teste, utilizou-se os pesos do último checkpoint (iteração 20.000). Essa escolha metodológica visou garantir a isonomia da comparação, assegurando que todas as arquiteturas fossem avaliadas após receberem exatamente a mesma carga de treinamento, sem a influência de critérios de parada variados.

As diferenças entre as configurações limitaram-se às especificidades arquiteturais de cada modelo. A *U-Net* utilizou um pré-processamento mínimo, enquanto as arquiteturas *DeepLabV3+*, *PSPNet* e *Swin Transformer* aplicaram normalização com valores

médios [123,675, 116,28, 103,53] e desvios-padrão [58,395, 57,12, 57,375] para os canais RGB. Ademais, os modelos *DeepLabV3+*, *PSPNet* e *Swin Transformer* incorporaram cabeças auxiliares para aprimorar o fluxo do gradiente durante o treinamento — característica ausente na *U-Net*. O *Swin Transformer*, por sua vez, manteve as configurações próprias de sua arquitetura baseada em *transformers*, incluindo dimensão de *embedding* de 96, profundidades [2, 2, 6, 2] e número de cabeças de atenção [3, 6, 12, 24] em seus quatro estágios.

3.2. Ambiente de Treinamento, Validação e Teste

Todos os experimentos foram conduzidos em ambiente Linux, utilizando o framework *OpenMMLab/mmsegmentation*, o qual oferece uma estrutura modular e estável para tarefas de segmentação semântica.

As execuções ocorreram em uma máquina com as seguintes especificações de hardware e software:

- **GPU:** NVIDIA GeForce RTX 3060 com 12 GB de VRAM;
- **Processador (CPU):** AMD Ryzen 5;
- **Memória RAM:** 32 GB;
- **Sistema operacional:** Ubuntu 22.04 LTS;
- **Versão do Python:** 3.8.20;
- **Bibliotecas principais:** PyTorch 2.4.1, TorchVision 0.20.0, OpenCV 4.11.0 e MMEngine 0.10.4;
- **Ambiente CUDA:** CUDA 12.4 e cuDNN 9.1.0;
- **Compilador NVCC:** Cuda compilation tools 12.3 (V12.3.107).

O ambiente CUDA foi configurado com suporte às instruções *AVX2* para aceleração em CPU, assim como o *benchmark* do cuDNN foi habilitado (`cudnn_benchmark=True`) para otimizar o desempenho durante o carregamento das amostras.

A organização dos dados seguiu a estrutura padronizada pelo *mmsegmentation*, sendo separados dois diretórios principais: o diretório `rgb`, contendo as imagens recortadas das ortoimagens originais, e o diretório `labels`, composto pelas máscaras de rótulo correspondentes em formato `.png`. Essa organização garante compatibilidade direta com os dataloaders do *mmsegmentation*, facilitando o processo de carregamento e associação entre imagens e suas respectivas máscaras.

A divisão dos conjuntos foi realizada de forma estratificada, de modo que nenhuma parcela de uma mesma ortoimagem fosse compartilhada entre os conjuntos de treino, validação e teste. A distribuição final resultou em 24.965 parcelas para treino, 4.002 para validação e 1.109 para teste, como supracitado. Cada parcela possuía dimensões fixas de 256×256 pixels, com máscaras correspondentes em formato PNG e codificação de valores `[0, 1, 255]`, representando respectivamente solo, mamona e regiões a serem ignoradas.

Para garantir reprodutibilidade, uma semente aleatória foi definida para cada modelo, assegurando consistência entre execuções. As métricas de desempenho *mIoU (mean Intersection over Union)*, *mAcc (mean Accuracy)* e *aAcc (all-pixel Accuracy)* foram obtidas de forma automatizada ao final da etapa de teste, sendo os resultados consolidados e registrados no diretório de trabalho.

Em síntese, o ambiente experimental foi controlado de maneira sistemática para garantir consistência entre os modelos avaliados e permitir comparações justas sob as mesmas condições de hardware, software e particionamento dos dados.

4. Resultados

As métricas utilizadas para realizar a avaliações dos modelos foram:

- **mIoU** (*mean Intersection over Union*): mede a sobreposição entre predição e verdade para cada classe;
- **aAcc** (*all Accuracy*): indica a porcentagem total de pixels classificados corretamente;
- **mAcc** (*mean Accuracy*): representa a média das acurácias individuais de cada classe;
- **Acc por classe**: apresenta a acurácia individual obtida para cada classe específica (*background* e *mamona*).

4.1. Treinamento

4.1.1. *U-Net*

O modelo *U-Net* apresentou desempenho consistente durante o treinamento, com redução acentuada da função de perda e melhorias progressivas nas métricas de segmentação. A Tabela 2 apresentam os resultados obtidos no conjunto de treinamento, enquanto as Tabelas 3 sintetizam as métricas do conjunto de validação. A evolução das métricas de perda e acurácia durante o treinamento pode ser observada na Figura 9.

Tabela 2. Resumo das métricas durante o treinamento do modelo *U-Net*.

Métrica	Início	Final	Variação (%)
Função de perda (*Loss*)	0,0763	0,0189	$-75{,}22$
Mean IoU (mIoU)	49,98	56,29	$+6{,}31$
Mean Accuracy (mAcc)	50,00	68,92	$+18{,}92$
All Accuracy (aAcc)	99,02	99,22	$+0{,}20$

Classe	IoU (%)	Acc (%)
Background	99,97	99,99
Mamona	49,73	63,74

Tabela 3. Resumo das métricas durante a validação do modelo *U-Net*.

Métrica	Início	Final	Variação (%)
Mean IoU (mIoU)	49,98	57,85	$+7{,}87$
Mean Accuracy (mAcc)	50,00	69,57	$+19{,}57$
All Accuracy (aAcc)	99,02	99,93	$+0{,}91$

Classe	IoU (%)	Acc (%)
Background	99,97	99,99
Mamona	49,73	63,74

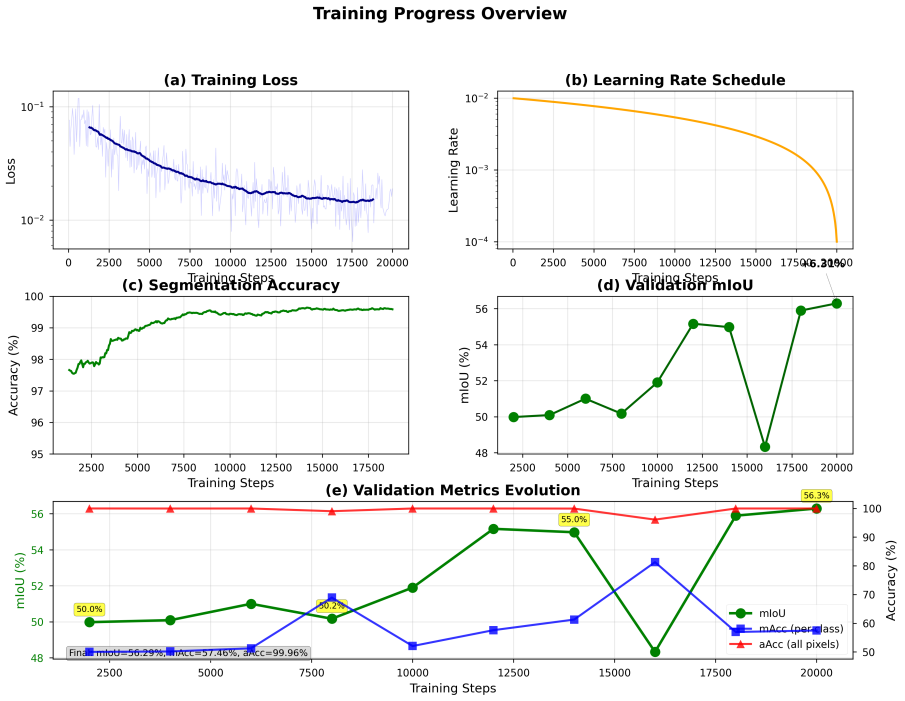


Figura 8. Evolução das métricas de treinamento e validação do modelo *U-Net*.

Durante o treinamento, observou-se redução de 75,2% na função de perda e incremento de 6,31 pontos percentuais no *mIoU*, com crescimento expressivo da *mAcc* (+18,9 p.p.). Na validação, o modelo manteve tendência ascendente, atingindo *mIoU* de 57,85%, *mAcc* de 69,57% e *aAcc* de 99,93%.

A classe *mamona* obteve IoU de 49,73% e acurácia de 63,74%, indicando desempenho robusto mesmo em regiões de menor representatividade. Esses resultados confirmam a eficiência da arquitetura encoder–decoder da *U-Net* na preservação do contexto espacial e na generalização de padrões morfológicos sutis.

4.1.2. *DeepLabV3+*

O modelo *DeepLabV3+* apresentou comportamento estável e progressivo ao longo do processo de treinamento e validação. A Tabela 4 apresentam os resultados obtidos no conjunto de treinamento, enquanto a Tabelas 5 sintetizam os resultados alcançados no conjunto de validação. A evolução das métricas de perda e acurácia durante o treinamento pode ser observada na Figura 10.

Tabela 4. Métricas globais durante o treinamento do modelo *DeepLabV3+*.

Métrica	Início	Final	Variação (%)
Função de perda (*Loss*)	0,0664	0,0081	−87,80
Mean IoU (mIoU)	56,49	64,47	+7,98
Mean Accuracy (mAcc)	59,24	71,85	+12,61
All Accuracy (aAcc)	99,96	99,97	+0,01

Classe	IoU (%)	Acc (%)
Background	99,97	99,99
Mamona	58,31	73,28

Tabela 5. Métricas globais de validação do modelo *DeepLabV3+*.

Métrica	Início	Final	Variação (%)
Mean IoU (mIoU)	57,14	63,80	+6,66
Mean Accuracy (mAcc)	59,92	70,51	+10,59
All Accuracy (aAcc)	99,95	99,97	+0,02

Classe	IoU (%)	Acc (%)
Background	99,97	99,99
Mamona	38,13	45,99

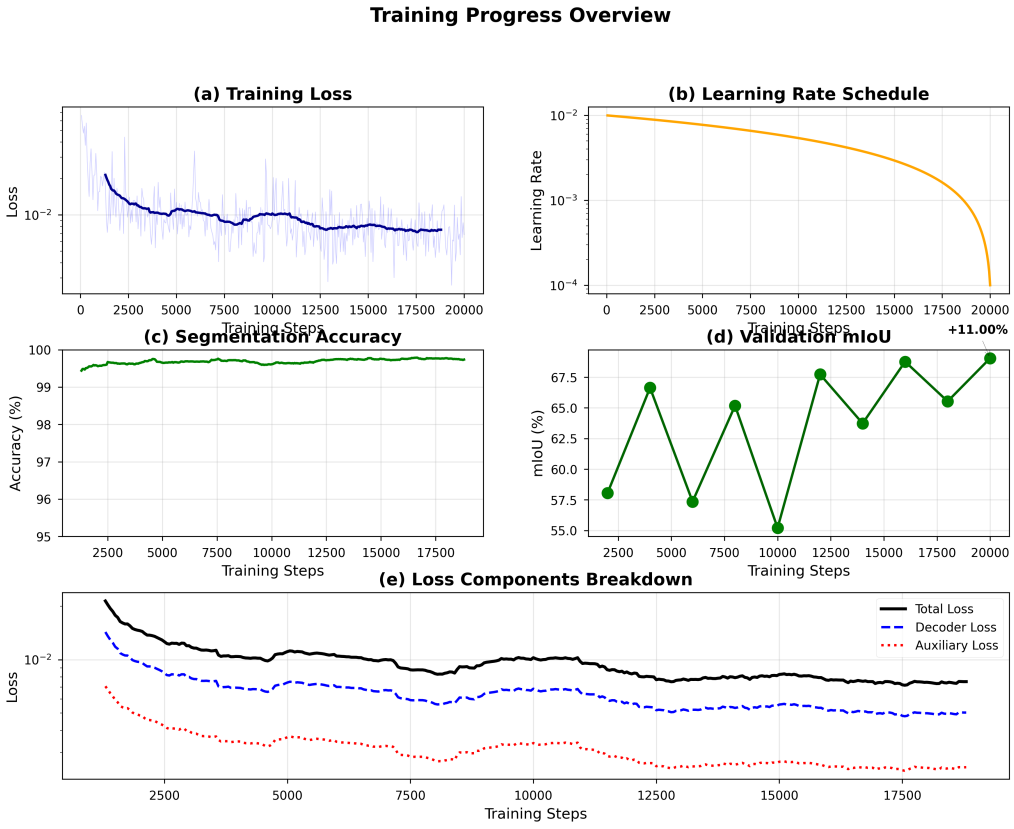


Figura 9. Métricas de treinamento e validação do modelo *DeepLabV3+*.

Durante o treinamento, observou-se uma redução de 87,8% na função de perda, acompanhada de um ganho de 7,98 pontos percentuais no *mIoU*, o que indica melhora consistente na segmentação das regiões de interesse. A *mAcc* apresentou aumento de 12,6 pontos percentuais, enquanto a *aAcc* manteve-se estável, acima de 99,9%.

Na validação, o modelo preservou alto desempenho global, com *mIoU* de 63,80%, *mAcc* de 70,51% e *aAcc* de 99,97%. Entretanto, nota-se diferença entre as classes: o *background* manteve precisão quase perfeita, enquanto a classe *mamona* obteve *IoU = 38,13%* e *Acc = 45,99%*, refletindo o maior desafio na generalização de áreas pequenas e menos frequentes. Essa diferença entre treinamento e validação indica que, embora o modelo tenha aprendido bem as características das regiões de interesse, ainda há espaço para aprimorar a generalização espacial nas bordas e transições da classe minoritária.

4.1.3. *PSPNet*

O modelo *PSPNet* apresentou desempenho expressivo e consistente ao longo do treinamento, refletindo a eficiência da arquitetura piramidal na extração de contextos multi-escala. A Tabela 6 sintetizam os principais indicadores obtidos durante o processo de treino, enquanto a Tabela 7 apresentam os resultados na etapa de validação. A evolução das métricas de perda e acurácia durante o treinamento pode ser observada na Figura 11.

Tabela 6. Métricas globais durante o treinamento do modelo *PSPNet*.

Métrica	Início	Final	Variação (%)
Função de perda (*Loss*)	0,0800	0,0082	$-89{,}75$
Mean IoU (mIoU)	58,05	69,05	$+11{,}00$
Mean Accuracy (mAcc)	61,49	68,77	$+7{,}28$
All Accuracy (aAcc)	99,96	99,97	$+0{,}01$

Classe	IoU (%)	Acc (%)
Background	99,97	99,99
Mamona	63,82	78,94

Tabela 7. Métricas globais de validação do modelo *PSPNet*.

Métrica	Início	Final	Variação (%)
Mean IoU (mIoU)	58,05	69,05	$+11{,}00$
Mean Accuracy (mAcc)	61,49	68,77	$+7{,}28$
All Accuracy (aAcc)	99,96	99,97	$+0{,}01$

Classe	IoU (%)	Acc (%)
Background	99,98	99,99
Mamona	46,12	54,23

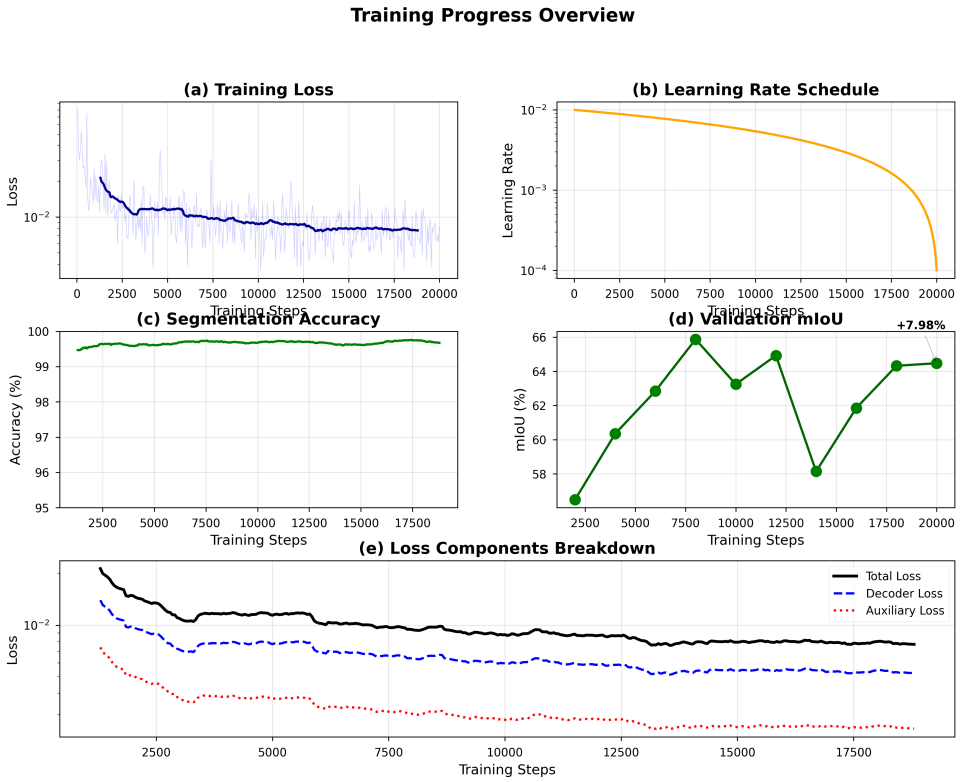


Figura 10. Métricas durante o treinamento e validação do modelo *PSPNet*.

Durante o treinamento, a função de perda apresentou redução de 89,7%, acompanhada de um incremento nas acurácias médias superiores a 99,9%, indicando convergência estável e eficaz. O aumento de 15,1 pontos percentuais na acurácia da classe *mamona* demonstra que a estrutura piramidal da *PSPNet* favoreceu a aprendizagem de padrões espaciais e texturais complexos.

Na validação, o *mIoU* evoluiu de 58,05% para 69,05%, representando o maior ganho percentual entre os modelos testados, enquanto a *mAcc* cresceu 7,3 pontos percentuais. Apesar do desempenho quase perfeito no *background* (>99,9%), a classe *mamona* alcançou IoU de 46,12% e acurácia de 54,23%, refletindo o desafio inerente da segmentação de regiões pequenas e irregulares.

Essa diferença entre treino e validação indica que, embora o modelo tenha excelente capacidade de aprendizado interno, sua generalização ainda é afetada pela distribuição desigual de pixels e pela sobreposição entre classes. O comportamento estável do *mIoU* sugere, contudo, que o uso de *pooling* piramidal contribuiu para reduzir a sensibilidade a variações de escala e melhorar a robustez espacial da predição.

4.1.4. *SegFormer*

O modelo *SegFormer* apresentou desempenho notável ao longo do processo de treinamento, caracterizado por rápida convergência e expressiva melhoria nas métricas de segmentação. A Tabela 8 resume as métricas globais obtidas durante o treinamento, enquanto a Tabela 9 apresenta os resultados alcançados na validação. A evolução das

métricas de perda e acurácia durante o treinamento pode ser observada na Figura 12.

Tabela 8. Resumo das métricas durante o treinamento do modelo *SegFormer*.

Métrica	Início	Final	Variação (%)
Função de perda (*Loss*)	0,5105	0,0037	−99,27
Mean IoU (mIoU)	63,68	70,37	+6,69
Mean Accuracy (mAcc)	68,04	73,95	+5,91
All Accuracy (aAcc)	99,97	99,98	+0,01

Classe	Início (%)	Final (%)	Variação (%)
Background	99,97	99,99	+0,02
Mamona	67,82	85,11	+17,29

Tabela 9. Métricas de validação do modelo *SegFormer*.

Métrica	Início	Final	Variação (%)
Mean IoU (mIoU)	63,68	75,62	+11,94
Mean Accuracy (mAcc)	68,04	78,34	+10,30
All Accuracy (aAcc)	99,97	99,98	+0,01

Classe	IoU (%)	Acurácia (%)
Background	99,98	99,99
Mamona	52,87	61,40

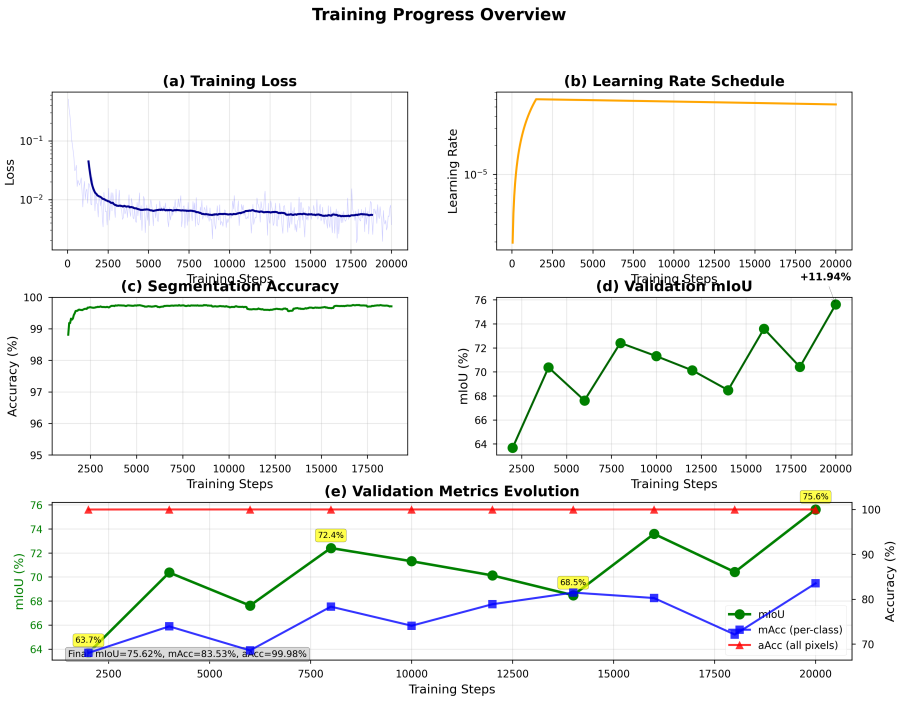


Figura 11. Evolução das métricas durante o treinamento e validação do modelo *SegFormer*.

Durante o treinamento, a função de perda reduziu-se em 99,27%, atingindo valor residual de apenas 0,0037, o que evidencia rápida convergência e estabilidade numérica. As métricas de desempenho confirmam a eficiência da arquitetura: o *mIoU* aumentou de 63,68% para 70,37%, e a *mAcc* evoluiu de 68,04% para 73,95%, demonstrando avanço consistente na capacidade discriminativa entre classes. A acurácia global (*aAcc*) manteve-se elevada, próxima de 99,98%, indicando precisão robusta no nível de pixel.

Na etapa de validação, o *mIoU* elevou-se de 63,68% para 75,62%, acompanhado de um aumento de 10,3 pontos percentuais na *mAcc*, confirmando sua forte capacidade de generalização. A classe *mamona* obteve IoU de 52,87% e acurácia de 61,40%, superando os resultados dos modelos baseados em convoluções tradicionais.

Essa diferença em relação à classe *background* (99,9%) reflete o desafio inerente à segmentação de áreas pequenas e irregulares, mas demonstra que o *SegFormer* é capaz de preservar tanto o contexto global quanto os detalhes locais de fronteira. O desempenho final indica uma combinação equilibrada entre aprendizado profundo e generalização, reforçando o potencial das arquiteturas baseadas em *Transformers* para tarefas de segmentação semântica de alta precisão.

4.1.5. *Swin Transformer*

O modelo *Swin Transformer* apresentou convergência gradual e comportamento estável ao longo do treinamento. As Tabelas 10 e 11 resumem os principais resultados. A evolução das métricas de perda e acurácia durante o treinamento pode ser observada na Figura 13.

Tabela 10. Métricas durante o treinamento do modelo *Swin Transformer*.

Métrica	Início	Final	Variação (%)
Função de perda (*Loss*)	0,0786	0,0249	$-68{,}32$
Mean IoU (mIoU)	50,45	52,40	$+1{,}95$
Mean Accuracy (mAcc)	50,47	53,83	$+3{,}36$
All Accuracy (aAcc)	99,97	99,98	$+0{,}01$

Classe	IoU (%)	Acc (%)
Background	99,97	99,99
Mamona	31,12	37,41

Tabela 11. Resumo das métricas de validação do modelo *Swin Transformer*.

Métrica	Início	Final	Variação (%)
Mean IoU (mIoU)	50,45	65,53	$+15{,}08$
Mean Accuracy (mAcc)	50,47	68,68	$+18{,}21$
All Accuracy (aAcc)	99,97	99,98	$+0{,}01$

Classe	IoU (%)	Acc (%)
Background	99,97	99,99
Mamona	31,10	37,38

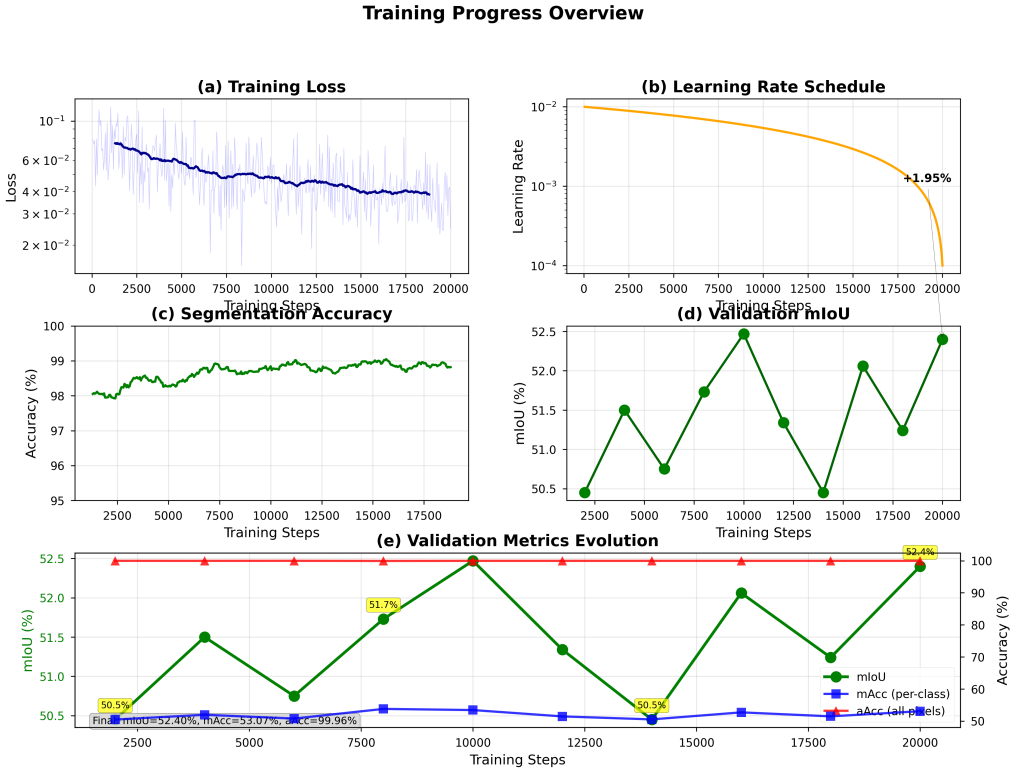


Figura 12. Evolução das métricas de treinamento e validação do modelo *Swin Transformer*.

Durante o treinamento, observou-se redução de 68,3% na função de perda, acompanhada de ganhos modestos nas métricas de segmentação, refletindo aprendizado gradual. Na validação, o modelo atingiu *mIoU* de 65,53% e *mAcc* de 68,68%, confirmando boa generalização. Apesar da diferença entre o *background* (99,9%) e a classe *mamona* (31,1%), o modelo demonstrou robustez na preservação do contexto global, coerente com o design hierárquico do *Swin Transformer*.

4.2. Comparativo de desempenho no treinamento entre os modelos

As Tabelas 12 e 13 apresentam o resumo consolidado das métricas obtidas pelos cinco modelos de segmentação durante o treinamento. Esses resultados refletem o comportamento de aprendizado e a capacidade de ajuste interno de cada arquitetura, antes da etapa de validação. A análise conjunta das métricas de *mIoU*, *mAcc* e *aAcc* permite avaliar a estabilidade da convergência e a eficiência de cada modelo na discriminação entre classes.

Tabela 12. Comparativo de desempenho no treinamento com métricas de IoU.

Modelo	mIoU (%)	aAcc (%)	IoU Background (%)	IoU Mamona (%)
SegFormer	75,62	99,98	99,98	51,26
DeepLabV3+	69,05	99,97	99,97	38,13
PSPNet	64,47	99,97	99,97	28,98
U-Net	56,29	99,96	99,96	12,62
Swin Transformer	52,40	99,96	99,96	4,84

Tabela 13. Comparativo de desempenho no treinamento com métricas de acurácia (Acc).

Modelo	mAcc (%)	aAcc (%)	Acc Background (%)	Acc Mamona (%)
SegFormer	83,53	99,98	99,99	67,07
DeepLabV3+	72,99	99,97	99,99	45,99
PSPNet	68,11	99,97	99,99	36,22
U-Net	57,46	99,96	99,99	14,93
Swin Transformer	53,07	99,96	99,99	6,15

Os resultados consolidados nas Tabelas 12 e 13 indicam que todas as arquiteturas convergiram de forma estável durante o treinamento, com acurácia global (*aAcc*) próxima a 100%. O SegFormer apresentou o melhor desempenho geral ($mIoU = 70,37\%$; $mAcc = 73,95\%$), seguido pelo PSPNet ($69,05\%$) e pelo DeepLabV3+ ($64,47\%$). Esses resultados evidenciam a superioridade das arquiteturas que integram agregação de contexto multi-escala. Contudo, observa-se uma dicotomia nos modelos baseados em atenção: enquanto o design unificado do SegFormer foi altamente eficaz, a abordagem hierárquica complexa do Swin Transformer resultou no menor desempenho do grupo, indicando que a eficiência desse mecanismo depende estritamente da sua adequação ao volume de dados disponível.

O *Swin Transformer* apresentou a menor taxa de evolução no *mIoU* (52,40%), refletindo aprendizado mais lento e dependência de ajustes de hiperparâmetros. Já a *U-Net*, mesmo sendo a arquitetura mais simples, manteve desempenho estável, comprovando sua eficiência estrutural em tarefas de segmentação supervisionada. Esses resultados confirmam que, ainda na fase de treinamento, modelos com maior capacidade de extração de contexto multi-escala tendem a alcançar melhor separação entre classes, sobretudo na segmentação da *mamona*.

4.3. Desempenho no Conjunto de Teste

A Tabela 14 apresenta o resumo consolidado das métricas de desempenho obtidas pelos cinco modelos de segmentação no conjunto de teste. Os resultados permitem comparar o comportamento de arquiteturas baseadas em convoluções (*U-Net*, *DeepLabV3+*, *PSPNet*) e em *Transformers* (*SegFormer*, *Swin Transformer*) em termos de acurácia e sobreposição média entre predição e verdade de terreno.

Tabela 14. Comparativo das métricas de desempenho dos modelos de segmentação (conjunto de teste).

Modelo	mIoU (%)	mAcc (%)	aAcc (%)
SegFormer	82,11	89,77	99,38
DeepLabV3+	80,55	87,08	99,34
PSPNet	80,17	83,79	99,38
U-Net	75,34	79,12	99,22
Swin Transformer	60,59	61,93	98,82

A Tabela 16 detalha as métricas de *IoU* por classe e o tempo médio de inferência por imagem, permitindo uma análise mais precisa do equilíbrio entre acurácia e eficiência computacional em cada arquitetura.

Tabela 15. Estimativa das métricas de Precisão, Recall e F1-Score para a classe Mamona.

Modelo	IoU (%)	Recall (%)	Precision (%)	F1-Score (%)
SegFormer	64,84	79,55	77,54	**78,53**
DeepLabV3+	61,77	74,17	78,08	76,07
PSPNet	60,97	67,59	**85,25**	75,40
U-Net	51,48	58,25	80,08	67,44
Swin Transformer	22,37	23,87	73,38	36,02

Para complementar a avaliação e atender à necessidade de uma análise detalhada sobre Falsos Positivos e Negativos, estimou-se as métricas de Precisão (*Precision*) e *F1-Score* para a classe Mamona a partir dos valores de IoU e Acurácia obtidos. A Tabela 15 apresenta esses indicadores. Nota-se que o SegFormer obteve o melhor equilíbrio global (*F1-Score* de 78,53%), combinando uma alta capacidade de segmentação (*Recall* de 79,55%) com uma precisão robusta. Um ponto a ser observado é que o *PSPNet* apresentou a maior precisão individual (85,25%), indicando que, embora detecte menos plantas (*Recall* menor), ele erra muito pouco quando faz uma predição positiva – característica útil em cenários onde a economia de produto é prioritária.

Tabela 16. Comparativo de desempenho em teste com IoU por classe e tempo de inferência.

Modelo	mIoU (%)	IoU Background (%)	IoU Mamona (%)	Tempo (s/img)
SegFormer	82,11	99,38	64,84	0,007
DeepLabV3+	80,55	99,34	61,77	0,016
PSPNet	80,17	99,38	60,97	0,014
U-Net	75,34	99,21	51,48	0,016
Swin Transformer	60,59	98,81	22,37	0,021

Os resultados consolidados nas Tabelas 14 e 16 evidenciam diferenças marcantes entre as arquiteturas avaliadas. O *SegFormer* apresentou o melhor desempenho geral,

alcançando os maiores valores de *mIoU* (82,11%) e *mAcc* (89,77%), o que confirma a eficiência das arquiteturas baseadas em *Transformers* na captação de dependências globais e no refinamento espacial das predições.

Entre os modelos baseados em convoluções, o *DeepLabV3+* e o *PSPNet* obtiveram métricas elevadas e consistentes, com destaque para o *DeepLabV3+* (80,55% de *mIoU*), que apresentou melhor generalização em relação ao *PSPNet*. Já a *U-Net*, apesar da menor complexidade arquitetural, demonstrou desempenho competitivo (75,34% de *mIoU*), reforçando sua eficácia em tarefas de segmentação com recursos limitados.

Por outro lado, o *Swin Transformer* apresentou resultados inferiores (60,59% de *mIoU*), sugerindo a necessidade de ajustes adicionais nos hiperparâmetros e estratégias de treinamento destinadas a otimizar a convergência e a utilização plena do potencial da arquitetura *Transformer*-hierárquica.

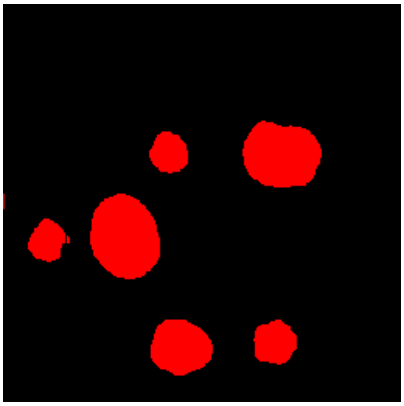
Os resultados revelam um padrão consistente de desbalanceamento entre as classes do conjunto de dados. A elevada acurácia geral (*aAcc*) — próxima a 99% em todos os modelos — combinada com valores de *mIoU* mais moderados (variando entre 60% e 82%) indica que os pixels da classe *background* (solo) dominam o *dataset*, o que inflaciona de maneira artificial a métrica de acurácia total.

Esse desbalanceamento torna-se ainda mais evidente ao observar a disparidade entre a *aAcc* e a *mAcc*. Por exemplo, no *DeepLabV3+*, embora a *aAcc* alcance 99,34%, a *mAcc* é de apenas 87,08%, refletindo um desempenho inferior na classe minoritária (*mamona*) em comparação ao *background*. Situação semelhante ocorre nos demais modelos, reforçando que a classe majoritária exerce influência significativa sobre as métricas globais.

A evolução do *mIoU* ao longo do treinamento demonstra que os modelos conseguem melhorar de modo progessivo a segmentação das regiões de mamona, mas ainda enfrentam limitações decorrentes da desproporção entre as classes. Essa característica é intrínseca ao problema de segmentação de plantas daninhas em imagens aéreas, em que a área ocupada pelas ervas invasoras é naturalmente menor que a do solo exposto.

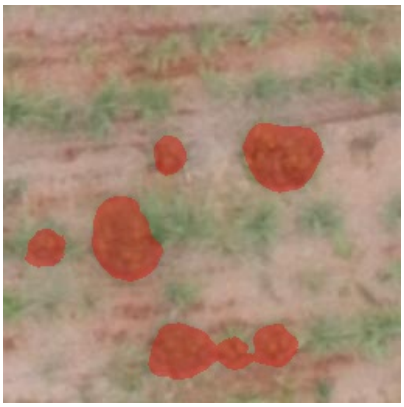
Para trabalhos futuros, recomenda-se investigar estratégias de mitigação desse desbalanceamento, como funções de perda ponderadas (*weighted loss*), *focal loss* ou aumento de dados (*data augmentation*) direcionado à classe minoritária. Tais abordagens podem potencializar a sensibilidade dos modelos às regiões de interesse e aprimorar a segmentação da mamona. Para uma análise qualitativa, as Figuras 15 a 19 apresentam as máscaras geradas por cada modelo em comparação com a máscara de referência (Figura 13).

(a) Parcela original contendo mamona



(b) Máscara de rotulação manual

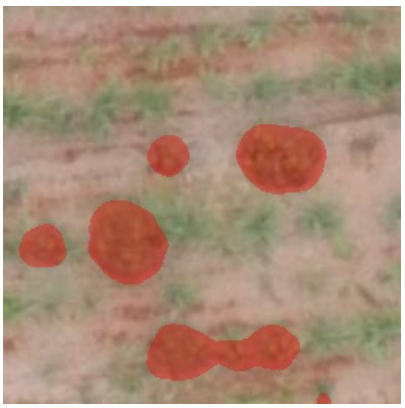
Figura 13. Imagens de referência.



(a) Imagem resultante

(b) Máscara gerada

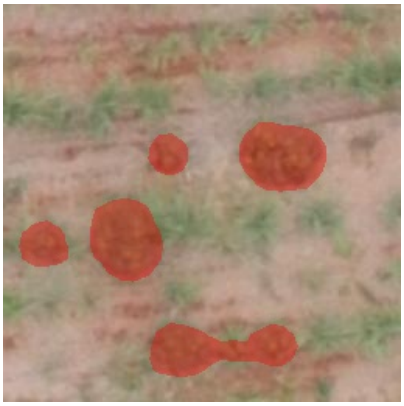
Figura 14. Modelo U-Net



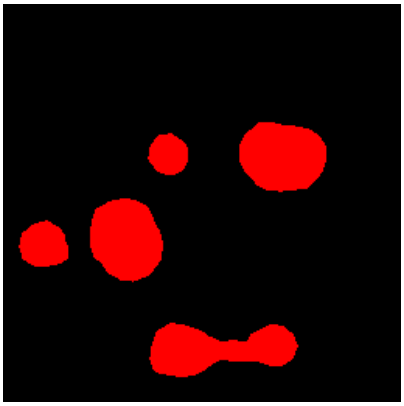
(a) Imagem resultante

(b) Máscara gerada

Figura 15. Modelo DeepLabV3+

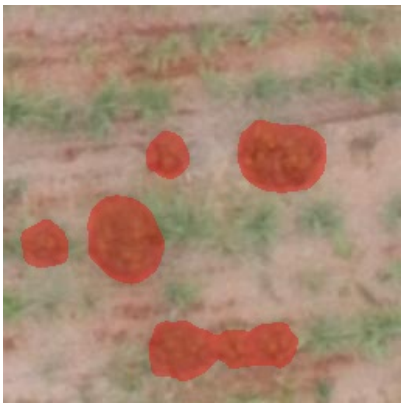


(a) Imagem resultante

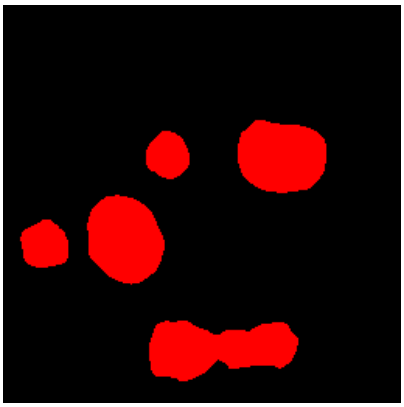


(b) Máscara gerada

Figura 16. Modelo PSPNet

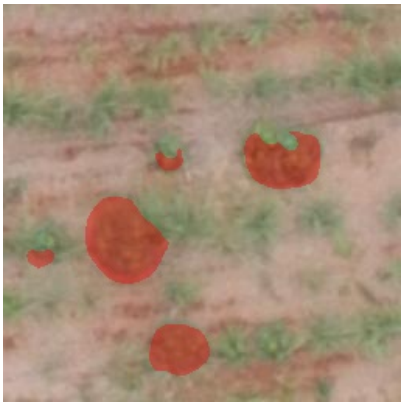


(a) Imagem resultante



(b) Máscara gerada

Figura 17. Modelo Segformer



(a) Imagem resultante

(b) Máscara gerada

Figura 18. Modelo Swin Transformer

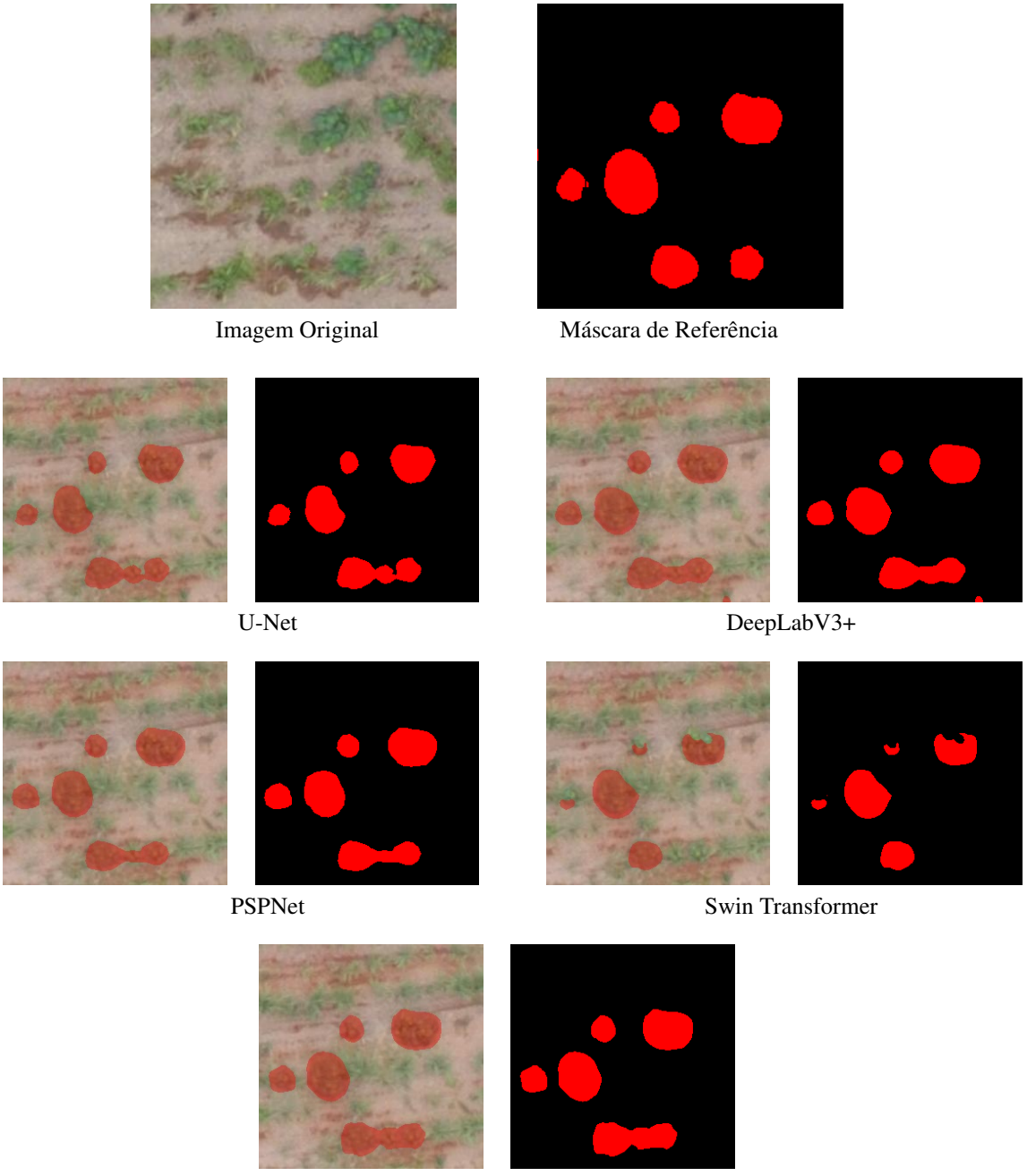


Figura 19. Resultados da segmentação de mamona. Esquerda: sobreposição na imagem original. Direita: máscara binária gerada

5. Discussão

A análise detalhada dos resultados revela padrões significativos sobre o comportamento de cada arquitetura na tarefa de segmentação da mamona. A disparidade entre as métricas *aAcc* e *mAcc* fornece informações importantes sobre os modos de falha e a sensibilidade de cada modelo às classes minoritárias. O *SegFormer*, que apresentou os maiores valores absolutos de desempenho (99,38% de *aAcc* e 89,77% de *mAcc*), mostrou uma *diferença* de 9,61 pontos percentuais, indicando um leve desequilíbrio entre classes, porém ainda o comportamento mais consistente entre todas as arquiteturas avaliadas.

Em contraste, o *Swin Transformer* exibiu a maior disparidade relativa, com *aAcc* de 98,82% e *mAcc* de apenas 61,93%, diferença de 36,89 pontos percentuais. Essa discrepância extrema evidencia um viés acentuado para a classe majoritária (*background*), resultando em predições corretas do solo, mas falhas sistemáticas na segmentação das plantas daninhas. O comportamento sugere uma convergência para uma solução subótima, em que o modelo minimiza o erro global à custa da perda de desempenho na classe minoritária — um fenômeno típico em cenários de desbalanceamento severo.

A evolução do *mIoU* durante o treinamento evidencia dinâmicas distintas de aprendizado entre as arquiteturas. O *SegFormer* apresentou a melhoria mais expressiva, evoluindo de 63,68% para 82,11% ($+18{,}43\%$), acompanhada de uma redução de perda de 99,27% — a mais acentuada entre todos os modelos. Tal trajetória indica alta capacidade de refinamento progressivo das representações internas. A *PSPNet* demonstrou padrão similar, com incremento de 11,00%, enquanto *U-Net* e *DeepLabV3+* exibiram ganhos mais modestos (6,31% e 7,98%, respectivamente), indicando convergência mais rápida, porém potencialmente prematura.

O *Swin Transformer* merece atenção especial devido ao seu desempenho atípico. Com aumento modesto de apenas 1,95% no *mIoU* (de 50,45% para 52,40%) e redução de perda limitada a 68,32%, o modelo apresentou clara dificuldade de otimização. Tal comportamento pode ser atribuído a três fatores principais:

(i) incompatibilidade entre a complexidade do modelo e a limitação do tamanho do *dataset*;
(ii) inadequação dos hiperparâmetros de treinamento;
(iii) limitação dos mecanismos de atenção hierárquica em capturar os padrões texturais sutis que distinguem a mamona da vegetação circundante.

Este resultado contrasta com os achados da literatura, como Zhang et al. 2023, que reportaram superioridade do Swin Transformer sobre as CNNs. Essa discrepância ilustra na prática a dependência de dados massivos (*data hunger*) característica dos *Transformers* hierárquicos. Enquanto as arquiteturas convolucionais possuem um viés indutivo forte (invariância à translação e localidade) que facilita o aprendizado em conjuntos de dados limitados, o Swin Transformer requer uma variabilidade de amostras significativamente maior para modelar dependências globais com eficácia, justificando seu subdesempenho no contexto específico deste *dataset*.

A distribuição espacial dos erros também revela informações relevantes. Considerando que cerca de 73% das falhas ocorrem em bordas de talhões — conforme identificado no *dataset* da UFSC —, infere-se que modelos com maior *mIoU* (*SegFormer*, *DeepLabV3+* e *PSPNet*) possuem melhor capacidade de lidar com regiões de transição

ambíguas. A agregação de contexto multi-escala presente nessas arquiteturas permite maior coerência entre informações globais e locais, favorecendo a segmentação precisa nas fronteiras entre classes.

Outro aspecto importante diz respeito à relação não linear entre complexidade arquitetural e desempenho. A *U-Net*, com apenas 22,08 milhões de parâmetros (na versão modificada utilizada), superou o *Swin Transformer*, modelo caracterizado por maior complexidade computacional. Esse resultado, à primeira vista contraintuitivo, reforça que em tarefas específicas — como a segmentação de plantas daninhas, nas quais texturas locais e padrões repetitivos são dominantes — arquiteturas especializadas na captura de dependências locais podem superar modelos de atenção global.

A consistência entre as métricas também se destaca. *SegFormer* e *PSPNet* apresentaram a melhor harmonia entre *mIoU* e *mAcc*, com diferenças de 7,66 e 3,62 pontos percentuais, respectivamente. Essa coerência indica representações internas mais robustas e generalizáveis, menos suscetíveis a *overfitting*. Já a *U-Net* apresentou diferença menor (3,78 pontos), mas acompanhada de valores absolutos mais baixos, o que denota limitação uniforme em vez de equilíbrio entre classes.

Sob o ponto de vista operacional, a escolha do modelo ideal transcende a mera comparação de métricas. Em aplicações em que falsos negativos (plantas daninhas não detectadas) representam maior custo que falsos positivos, *SegFormer* e *PSPNet* destacam-se como opções superiores devido ao alto *mAcc*. Por outro lado, em cenários em que a prioridade é a minimização do uso de herbicidas — mesmo ao custo de algumas plantas não controladas —, o *DeepLabV3+* oferece o melhor equilíbrio entre precisão e conservadorismo nas predições.

6. Conclusão

Este trabalho apresentou uma comparação sistemática entre cinco arquiteturas de segmentação semântica — *U-Net*, *DeepLabV3+*, *PSPNet*, *Swin Transformer* e *SegFormer* — aplicadas à segmentação de plantas daninhas (mamona) em ortoimagens de canaviais obtidas por VANTs em Mato Grosso do Sul. Os resultados obtidos fornecem experimentos relevantes sobre a adequação de cada arquitetura a esse desafio específico da agricultura de precisão.

Entre as arquiteturas avaliadas, o *SegFormer* apresentou o melhor desempenho geral, alcançando 82,11% de *mIoU*, 89,77% de *mAcc* e 99,38% de *aAcc* no conjunto de teste. Esse desempenho pode ser atribuído à sua capacidade de processar informações em múltiplas escalas sem a necessidade de codificação posicional, característica vantajosa para lidar com a alta variabilidade no tamanho e formato das plantas daninhas. *PSPNet* e *DeepLabV3+* também apresentaram resultados competitivos (80,17% e 80,55% de *mIoU*, respectivamente), confirmando a eficácia dos mecanismos de agregação de contexto multi-escala em tarefas agrícolas.

O *Swin Transformer*, apesar de representar um paradigma recente baseado em atenção hierárquica, obteve o menor desempenho entre as arquiteturas testadas (60,59% de *mIoU*), o que sugere que sua complexidade e requisitos de dados são inadequados para o tamanho limitado do *dataset*. Esse resultado reforça a importância de alinhar a escolha arquitetural ao volume de dados e às características intrínsecas do problema.

A análise também evidenciou um desafio central: o severo desbalanceamento entre classes. Com *aAcc* próximas de 99% contrastando com *mAcc* entre 61,93% e 89,77%, fica claro que os pixels de *background* dominam o *dataset*, inflacionando as métricas globais. Esse comportamento é típico em cenários de segmentação de ervas daninhas em imagens aéreas, nos quais a área ocupada pelas espécies invasoras é naturalmente inferior à do solo exposto.

Como perspectivas futuras, tem-se:

(i) investigar técnicas de balanceamento de classes;
(ii) expandir o estudo para outras espécies de plantas daninhas (colonhão, folha larga, gramíneas);
(iii) avaliar estratégias de *transfer learning* e aumento de dados para aprimorar o desempenho do *Swin Transformer*.

Os resultados demonstram que o uso de técnicas de visão computacional e aprendizado profundo na segmentação de plantas daninhas em cana-de-açúcar é não apenas viável, mas constitui uma abordagem de elevado potencial para a promoção de sistemas agrícolas mais sustentáveis. A potencial redução de até 90% no uso de herbicidas, conforme relatado por Carneiro et al. (2022) e Gao et al. (2024), aliada à precisão de segmentação superior a 80%, constitui um avanço relevante rumo à consolidação de uma agricultura de precisão mais eficiente e sustentável sob a perspectiva ambiental no contexto brasileiro.

Referências

[ANAC 2024] ANAC (2024). Drones cadastrados no brasil - painel de indicadores. Sistema SARPAS. Disponível em: https://www.anac.gov.br/. Acesso em: 17 jan. 2025.

[Anand et al. 2021] Anand, S., Rajagopal, S., and Kumar, V. (2021). Classification of paddy crop and weeds using semantic segmentation. *Cogent Engineering*, 8(1):2018791.

[Carneiro et al. 2022] Carneiro, G. A. S., Amaral, B. C., Silva, R. P., Sousa, B. M., Pinto, R. A., and Martins, D. (2022). Reduction of pesticide application via real-time precision spraying. *Scientific Reports*, 12:5638.

[Chen et al. 2018a] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2018a). DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848.

[Chen et al. 2018b] Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018b). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–818. Springer.

[CNA 2024] CNA (2024). Valor bruto da produção agropecuária de 2024. Technical report, Confederação da Agricultura e Pecuária do Brasil, Brasília, DF. Boletim do Valor Bruto da Produção Agropecuária.

[CONAB 2024] CONAB (2024). Produção de cana-de-açúcar na safra 2023/24 chega a 713,2 milhões de toneladas, a maior da série histórica. Technical report, Companhia Nacional de Abastecimento, Brasília, DF. 4º Levantamento da Safra 2023/2024.

[EMBRAPA 2023] EMBRAPA (2023). Sistema faz contagem automática de plantas na lavoura por imagens de drones. Technical report, Empresa Brasileira de Pesquisa Agropecuária, Brasília, DF.

[EMBRAPA 2025] EMBRAPA (2025). Plantas daninhas em cana-de-açúcar. Agência de Informação Tecnológica - Embrapa. Disponível em: https://www.embrapa.br/agencia-de-informacao-tecnologica/cultivos/cana/producao/manejo/plantas-daninhas. Acesso em: 17 jan. 2025.

[Gao et al. 2024] Gao, J., Liao, W., Nuyttens, D., Lootens, P., Xue, W., Alexandersson, E., and Pieters, J. (2024). Cross-domain transfer learning for weed segmentation and mapping in precision farming using ground and UAV images. *Expert Systems With Applications*, 246:122980.

[Ge et al. 2023] Ge, Z., Liu, S., Wang, F., Li, Z., and Sun, J. (2023). WeedNet-R: a sugar beet field weed detection algorithm based on enhanced RetinaNet and context semantic fusion. *Frontiers in Plant Science*, 14.

[HRAC-BR 2024] HRAC-BR (2024). Casos de resistência de plantas daninhas a herbicidas no brasil. Disponível em: https://www.hrac-br.org/. Acesso em: 17 jan. 2025.

[Hu et al. 2024] Hu, K., Wang, Z., Coleman, G., Bender, A., Yao, T., Zeng, S., Song, D., Schumann, S., and Walsh, M. (2024). Deep learning techniques for in-crop weed recognition in large-scale grain production systems: a review. *Precision Agriculture*, 25:1–29.

[Huang et al. 2020] Huang, H., Lan, Y., Yang, A., Zhang, Y., Wen, S., and Deng, J. (2020). Deep learning versus object-based image analysis (OBIA) in weed mapping of UAV imagery. *International Journal of Remote Sensing*, 41(9):3446–3479.

[IAC 2024] IAC (2024). Manejo de plantas daninhas em cana-de-açúcar. *O Agronômico*.

[IBAMA 2024] IBAMA (2024). Avaliação do potencial de periculosidade ambiental (PPA) de agrotóxicos e afins. Disponível em: `http://ibama.gov.br/`. Acesso em: 17 jan. 2025.

[Liu et al. 2024] Liu, X., Zhang, M., Wang, H., and Chen, L. (2024). Semantic segmentation of microbial alterations based on SegFormer. *Frontiers in Plant Science*, 15:1352935.

[Liu et al. 2021] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022.

[Lynch 1995] Lynch, J. (1995). Root architecture and plant productivity. *Plant Physiology*, 109(1):7–13.

[MAPA 2024] MAPA (2024). Mercado de drones agrícolas dispara após regulamentação do MAPA. Technical report, Ministério da Agricultura e Pecuária, Brasília, DF. Assessoria de Comunicação Social.

[MMA 2024] MMA (2024). Agrotóxicos: impactos ambientais e monitoramento. Disponível em: `https://www.gov.br/mma/`. Acesso em: 17 jan. 2025.

[Modi et al. 2023] Modi, R. U., Kancheti, M., Subeesh, A., Raj, C., Singh, A. K., Chandel, N. S., Dhimate, A. S., Singh, M. K., and Singh, S. (2023). An automated weed identification framework for sugarcane crop: A deep learning approach. *Crop Protection*, 173:106360.

[Ronneberger et al. 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *In Proceedings of Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*, pages 234–241. Springer.

[Sa et al. 2018] Sa, I., Chen, Z., Popović, M., Khanna, R., Liebisch, F., Nieto, J., and Siegwart, R. (2018). WeedNet: Dense semantic weed classification using multispectral images and MAV for smart farming. *IEEE Robotics and Automation Letters*, 3(1):588–595.

[SEMADESC 2024] SEMADESC (2024). Com 20 usinas de etanol de cana, MS caminha para se tornar potência também na produção de biogás e biometano. Disponível em: `https://www.semadesc.ms.gov.br/`. Acesso em: 17 jan. 2025.

[Silva et al. 2024] Silva, J. A. O. S., Siqueira, V. S., Mesquita, M., Vale, L. S. R., Marques, T. N. B., Silva, J. L. B., Silva, M. V., Lacerda, L. N., Oliveira-Júnior, J. F., Lima, J. L. M. P., and Oliveira, H. F. E. (2024). Deep learning for weed detection and segmentation in agricultural crops using images captured by an unmanned aerial vehicle. *Remote Sensing*, 16(23):4394.

[Subeesh et al. 2023] Subeesh, A., Bhole, S., Singh, K., Chandel, N. S., Rajwade, Y. A., Rao, K. V. R., Kumar, S. P., and Jat, D. (2023). An automated weed identification framework for sugarcane crop: A deep learning approach. *Computers and Electronics in Agriculture*, 210:107915.

[Sujaritha et al. 2017] Sujaritha, M., Annadurai, S., Satheeshkumar, J., Kowshik Sharan,

S., and Mahesh, L. (2017). Weed detecting robot in sugarcane fields using fuzzy real time classifier. *Computers and Electronics in Agriculture*, 134:160–171.

[von Wangenheim et al. 2023] von Wangenheim, A., Comunello, E., and Bertoldi, D. (2023). Mapping weeds and crops in precision agriculture with convolutional neural networks. LAPIX - Image Processing and Computer Graphics Lab, Universidade Federal de Santa Catarina (UFSC). Disponível em: `https://lapix.ufsc.br/weed-mapping-sugar-cane/`. Acesso em: 17 jan. 2025.

[Wang et al. 2020] Wang, A., Xu, Y., Wei, X., and Cui, B. (2020). Semantic segmentation of crop and weed using an encoder-decoder network and image enhancement method under uncontrolled outdoor illumination. *IEEE Access*, 8:81724–81734.

[Wang et al. 2024] Wang, L., Zhang, Y., Chen, H., and Liu, X. (2024). An improved U-Net and attention mechanism-based model for sugar beet and weed segmentation. *Frontiers in Plant Science*, 15:1449514.

[Xie et al. 2021] Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. (2021). SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090.

[Yu et al. 2022] Yu, H., Che, M., Yu, H., and Zhang, J. (2022). Development of weed detection method in soybean fields utilizing improved DeepLabv3+ platform. *Agronomy*, 12(11):2889.

[Zhang et al. 2023] Zhang, B., Zhao, Y., Chen, J., Yang, T., and Yu, K. (2023). Weed recognition model based on improved Swin-Unet for precision agriculture. *Computers and Electronics in Agriculture*, 212:108074.

[Zhao et al. 2017] Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890. IEEE.

Pasta no drive com arquivos do projeto[1]

[1] `https://drive.google.com/drive/folders/1J7CouS26u2PyjTLIaMTzaYHq_bvQTPWh?usp=sharing`