

**FUNDAÇÃO UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL
FACULDADE DE COMPUTAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO**

MAX HIROITO TIETI

**NORMALIZAÇÃO ADAPTATIVA E APRENDIZADO HÍBRIDO:
TÉCNICAS DE ADAPTAÇÃO DE DOMÍNIO PARA ANÁLISE DE
VEGETAÇÃO TRANSGEOGRÁFICA**

**CAMPO GRANDE
JULHO / 2025**

MAX HIROITO TIETI

**NORMALIZAÇÃO ADAPTATIVA E APRENDIZADO HÍBRIDO:
TÉCNICAS DE ADAPTAÇÃO DE DOMÍNIO PARA ANÁLISE
DE VEGETAÇÃO TRANSGEOGRÁFICA**

Dissertação apresentada na Faculdade de Ciência da Computação da Universidade Federal de Mato Grosso do Sul, como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação.

Linha de pesquisa: Metodologias e Técnicas de Computação.

Orientador: Prof. Dr. Jonathan de Andrade Silva

CAMPO GRANDE

JULHO / 2025

FOLHA DE APROVAÇÃO

Max Hiroito Tieti

NORMALIZAÇÃO ADAPTATIVA E APRENDIZADO HÍBRIDO: TÉCNICAS DE ADAPTAÇÃO DE DOMÍNIO PARA ANÁLISE DE VEGETAÇÃO TRANSGEOGRÁFICA

Redação final do Relatório de dissertação, aprovada pela Banca Examinadora em __ de ____ de 2025, na Faculdade de Ciência da Computação da Universidade Federal de Mato Grosso do Sul para obtenção do título de Mestre em Ciência da Computação.

Banca examinadora:

Prof. Dr. Jonathan de Andrade Silva – Orientador
Universidade Federal de Mato Grosso do Sul (UFMS)

Prof. Dr. Cesar Ivan Alvarez Mendonza
Universidad Politécnica Salesiana (UPS)

Prof. Dr. Wesley Nunes Gonçalves
Universidade Federal de Mato Grosso do Sul (UFMS)

AGRADECIMENTOS

A conclusão de um mestrado é uma jornada que normalmente é solitária. Mas não posso afirmar isso. Eu tive muito apoio com a presença de muitos que fizeram, do que era para ser uma ordinária passagem pelo mestrado, uma experiência profunda e humana.

Muito obrigado, Hemerson Pistori (*in memoriam*). Pela sua paciência infinita, pela sua bondade, por tolerar minhas discordâncias e embarcar nas minhas loucuras. Gostaria que mais pessoas tivessem tido a sorte de ter você como orientador e mentor. Seu amor pela ciência e pela vida reverberarão na minha alma e espero ansiosamente que toda a memória que tenho de ti se torne força e mais pura vontade pela vida. Nunca te esquecerei, Professor.

Ao Inovisão, meus colegas de laboratório, com quem compartilhei risadas e dores. Amigos que juntos rimos, brigamos e debatemos longas horas sobre o futuro da Inteligência Artificial ou da sobrevivência da humanidade. A ajuda mútua nos tempos de frustração e dor foram tão importantes quanto qualquer artigo que li ou qualquer parágrafo da minha defesa. Cheguei nem sabendo abrir o site da CAPES e hoje vocês me ensinaram como fazer ciência e como chegarmos mais longe, como comunidade, por meio da colaboração.

Ao meu orientador, Jonathan; por sua orientação precisa, sua paciência para acreditar no potencial do projeto mesmo quando os desafios pareciam intransponíveis. Por ter me ajudado a encontrar a simplicidade nesse problema complexo. E por ter aceitado o desafio de dar continuidade ao trabalho do nosso querido Hemerson Pistori.

À minha namorada, Mariana Rodrigues; que não me deixou desanimar e me deu esperanças até o final. Não é fácil manter a sanidade diante da pressão e incertezas do curso e das surpresas da vida e você sempre esteve do meu lado. Esse trabalho tem muito de você para que eu conseguisse terminá-lo.

Às instituições UFMS e UCDB, que sempre me abriram portas para o conhecimento e estudo. Os laboratórios, infraestruturas e ferramentas dados foram fundamentais para que eu conseguisse viabilizar minha pesquisa.

E por fim, à minha mãe, Dona Eliza, que sempre apoiou e valorizou o estudo. "Conhecimento ninguém tira". Eu sei, mãe... Posso dizer que eu aprendi muito com todo mundo aqui: Hemerson, Jonathan, Mariana, Júnior, Patrícia, Toshio, Eduardo,

Leonardo, Neto, Pedro... Todo mundo do Inovisão e meus professores da UFMS. Sim, valeu a pena, eles me tornaram uma nova pessoa. Obrigado, Mãe e família, sem o amor incondicional de vocês não seria possível, nem teria significado para mim trilhar esse caminho.

*"Ah! Não queria voltar! Gente te espera.
Ah! Não queria voltar! Mas em sua terra.
Ah! Não queria voltar! Gente te espera.
Gente que eu amo, mas meu coração não
nasceu pra ficar."*

(Hemerson Pistori)

RESUMO

Esta pesquisa explora a capacidade de modelos de aprendizado de máquina (especificamente redes neurais profundas) de generalizar para diferentes contextos e domínios. Aborda a adaptação de domínio, ou seja, como garantir que um modelo treinado em um conjunto de dados específico possa funcionar bem em outro conjunto de dados com características diferentes. Isso é explorado de normalização adaptativa e aprendizado híbrido; para tanto, ocorre a investigação da adaptação de domínio por meio da avaliação do desempenho de modelos pré-treinados em diferentes conjuntos de dados de ecossistemas distintos. Para a análise de vegetação é abordada a aplicação de modelos de aprendizado profundo na observação de imagens de satélite/aéreas, o que é fundamental para a compreensão e monitoramento da vegetação em diferentes escalas. A importância da generalização se justifica pela capacidade do modelo de funcionar bem em diferentes contextos e domínios, fato crucial para a aplicação prática de modelos de aprendizado de máquina em áreas como planejamento ecológico e urbano, onde os dados podem variar significativamente de um local para outro. Foram utilizados dados de duas universidades latino-americanas com ecossistemas distintos: UCDB, no Brasil, e UPS, no Equador. Resultados preliminares indicam que, apesar de todos os modelos testados (FCN, DeepLabV3 e SegFormer) apresentarem bom desempenho, o FCN se destacou na generalização entre os conjuntos de dados, com altas métricas de F1-score, IoU e precisão. O estudo aponta para áreas de melhoria, como a adaptação da normalização, o aprimoramento da rede e o ajuste fino dos parâmetros. Futuramente, serão incorporadas mais classes de segmentação, dados LiDAR e conjuntos de dados de diferentes hemisférios para aprofundar a análise da generalização do modelo. Com isso, pretendemos propor técnicas para melhor adaptação de modelos de aprendizado profundo a diferentes domínios, por meio de segmentação semântica de imagens, uso de redes neurais profundas para a análise de imagens de vegetação e ecossistemas, redundando em aplicações diretas para a pesquisa em monitoramento ambiental, agricultura de precisão e planejamento urbano, por exemplo.

Palavras-chave: Adaptação de domínio; Redes Neurais Profundas; Imagens aéreas; Ortofotos; Aprendizado por transferência.

ABSTRACT

This research explores the ability of machine learning models (specifically deep neural networks) to generalize across different contexts and domains. It addresses domain adaptation, i.e., how to ensure a model trained on a specific dataset can perform well on another dataset with different characteristics. This is explored through adaptive normalization and hybrid learning; for this purpose, domain adaptation is investigated by evaluating the performance of pre-trained models on different datasets from distinct ecosystems. For vegetation analysis, the application of deep learning models to satellite/aerial image observation is addressed, which is fundamental for understanding and monitoring vegetation at different scales. The importance of generalization is justified by the model's ability to perform well in different contexts and domains, a crucial factor for the practical application of machine learning models in areas such as ecological and urban planning, where data can vary significantly from one location to another. Data from two Latin American universities with distinct ecosystems were used: UCDB, in Brazil, and UPS, in Ecuador. Preliminary results indicate that, although all tested models (FCN, DeepLabV3, and SegFormer) performed well, FCN stood out in generalization between datasets, with high F1-score, IoU, and accuracy metrics. The study points to areas for improvement, such as normalization adaptation, network enhancement, and parameter fine-tuning. In the future, more segmentation classes, LiDAR data, and datasets from different hemispheres will be incorporated to deepen the analysis of model generalization. With this, we intend to propose techniques for better adaptation of deep learning models to different domains, through semantic segmentation of images, using deep neural networks for the analysis of vegetation and ecosystem images, resulting in direct applications for research in environmental monitoring, precision agriculture, and urban planning, for example.

Keywords: Domain Adaptation; Deep Neural Networks; Aerial Imagery; Orthophotos; Transfer Learning.

LISTA DE FIGURAS

Figura 1 - Arquitetura Tribbranch.....	72
Figura 2 - Separação de canais (<i>Split</i>)	74
Figura 3 - Processamento Especializado (<i>Process</i>) e Fusão de Features (<i>Fuse</i>)....	76
Figura 4 - Boxplots de Desempenho por Backbone	107
Figura 5 - Análise Geral de Performance	108
Figura 6 - Modelo treinado no domínio UCDB e avaliado no domínio UPS	119
Figura 7 - Modelos Baseados em Transformer (Tribbranch Swin Base CUT e Normal)	119
Figura 8 - Modelos Baseados em Transformer (Tribbranch Swin Base CUT e Normal)	120
Figura 9 - Modelo T-ResNet101 UPS Cut Plus	121
Figura 10 - Modelo treinado no domínio UPS e avaliado no domínio UCDB	122
Figura 11 - Modelo Tribbranch - ResNet101	123
Figura 12 - Modelo Tribbranch Plus Cut - Resnet50	124
Figura 13 - Modelo Tribbranch Resent101 CUT	125

LISTA DE TABELAS

Tabela 1 - Composição final dos <i>datasets</i>	89
Tabela 2 - Índice Jaccard Geral limites superiores teóricos.....	93
Tabela 3 - Índice Jaccard Geral limites superiores teóricos por classes.....	94
Tabela 4 - Resultados arquitetura padrão.....	96
Tabela 5 - Análise por classe.....	96
Tabela 6 - Resultados arquitetura Tribbranch.....	98
Tabela 7 - Estabilidade entre o baseline e o Tribbranch básico.....	98
Tabela 8 - Performance por Arquitetura Base.....	100
Tabela 9 - Performance Tribbranch Básica	100
Tabela 10 - Estabilidade Cross-Validation por Classe	102
Tabela 11 - Performance por Feature de Domain Awareness	104
Tabela 12 - Desempenho por Backbone	105
Tabela 13 - Desempenho por Cenário de Domínio.....	106
Tabela 14 - Comparação de Performance entre Arquiteturas.....	109
Tabela 15 - Performance por Cenário de Domínio	110
Tabela 16 - Resultados por feature de Domain Awareness.....	113
Tabela 17 - Comparação Geral de Performance	114
Tabela 18 - Impacto da Transferência de Estilo por Arquitetura	116
Tabela 19 - Performance por Cenário e Técnica de Transferência	116
Tabela 20 - Impacto da Transferência de Estilo por Classe (<i>Cross-domain</i>).....	116
Tabela 21 - Matriz de Compatibilidade Arquitetura-Técnica	117
Tabela 22 - Comparação das Configurações Ótimas	117

LISTA DE QUADROS

Quadro 1 - Classificação de Estabilidade por Cenário	115
---	-----

SUMÁRIO

1 INTRODUÇÃO	14
1.1 Contextualização	14
1.2 Motivação e Justificativa.....	15
1.3 Objetivos.....	16
1.4 Organização da escrita	16
 2 FUNDAMENTAÇÃO TEÓRICA.....	18
2.1 Adaptação de domínio (da) e sensoriamento remoto (sr)	18
2.1.1 Redes Neurais Convolucionais (CNNs)	20
2.1.2 Adaptação De Domínio Não Supervisionada	22
2.1.3 Subdomain style compensation network (SSCN).....	31
 3 PRIMEIRO ARTIGO: DEEP LEARNING MODELS FOR VEGETATION SEGMENTATION	38
3.1 Cross-Dataset Evaluation of Deep Learning Models for Vegetation Segmentation in Latin America	39
 4 SEGUNDO ARTIGO: TRIBRANCH: UMA ARQUITETURA PARA ADAPTAÇÃO DE DOMÍNIO EM SEGMENTAÇÃO SEMÂNTICA DE IMAGENS AÉREAS	57
 5 CONCLUSÕES FINAIS	129
5.1 Aplicabilidade	130
5.2 Metodologia.....	131
5.3 Resultados Finais.....	133
 REFERÊNCIAS.....	135
APÊNDICE.....	137
Apêndice 1 - Tabela Tribbranch	137
Apêndice 2 - Tabela non tribranch	150
Apêndice 3 - Tabelas Detalhadas	154

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

O avanço rápido em técnicas de *deep learning* tem aprimorado significativamente a capacidade de segmentação de imagens, especialmente em aplicações com imagens aéreas e ortofotos. Apesar disso, um dos desafios persistentes é a mudança de domínio, que ocorre quando modelos treinados em um conjunto de dados apresentam desempenho inferior quando testados em outro com características diferentes. Essa questão é particularmente pronunciada em ambientes ecológicos e urbanos, onde variações em vegetação, arquitetura e uso da terra podem afetar drasticamente o desempenho do modelo.

Nesta pesquisa, focamos em abordar o problema da mudança de domínio na segmentação de imagens aéreas e ortofotos. Nossos experimentos preliminares envolveram o treinamento e teste de modelos de *deep learning* em conjuntos de dados de duas regiões distintas na América Latina: a Universidade Católica Dom Bosco (UCDB) em Campo Grande, Brasil, e a *Universidad Politécnica Salesiana* (UPS) em Quito, Equador. Essas regiões apresentam diferenças ecológicas significativas, o que representa um desafio para a generalização do modelo (Floresta montana, em Quito/Equador, e cerrado, em Campo Grande/Brasil).

Os resultados iniciais destacaram o desempenho de três arquiteturas de segmentação de ponta – *Fully Convolutional Networks* (FCN), *DeepLabV3* e *SegFormer*. Entre elas, a FCN demonstrou desempenho superior em avaliações entre conjuntos de dados. No entanto, o estudo também identificou áreas para melhoria, particularmente na adaptação da normalização, aprimoramento da rede e ajuste fino de parâmetros.

Reconhecendo essa lacuna, nossa pesquisa investiga o uso de técnicas de normalização adaptativa para aprimorar ainda mais as métricas de segmentação. Especificamente, implementaremos técnicas de mudança de domínio e *transfer learning* para avaliar a adaptabilidade do modelo. Uma das técnicas chave que exploraremos é a *Adaptive Instance Normalization* (AdaIN), que ajusta a média e a variância dos mapas de características para alinhar com os do domínio de destino.

Além disso, experimentaremos o alinhamento da distribuição de cores para avaliar seu impacto na melhoria das métricas de segmentação.

Por meio dessas investigações, buscamos fornecer uma compreensão mais profunda de como a adaptação da normalização pode mitigar os efeitos da mudança de domínio e melhorar o desempenho dos modelos de segmentação em diversos ambientes ecológicos e urbanos. Esta pesquisa não apenas contribui para o campo do aprendizado de máquina, mas também tem implicações práticas para o planejamento ecológico e urbano, onde modelos de segmentação precisos e adaptáveis são cruciais.

Na segunda fase da pesquisa, criamos novas arquiteturas (Tribranch, Domain-Aware TriBranchNet) e um pipeline de dados específico que combinou transferência de estilo com normalização, mostrando como o desafio do *domain shift* foi efetivamente enfrentado, sendo que a investigação combinou teoria e experimentação rigorosa para validar a abordagem.

1.2 MOTIVAÇÃO E JUSTIFICATIVA

A literatura sobre processamento de imagens tem explorado diversas técnicas de normalização para lidar com a variação de dados, especialmente em tarefas de transferência de domínio. No contexto de imagens aéreas, onde as diferenças de iluminação, contraste e outras características podem ser significativas entre conjuntos de dados, a busca por alternativas de normalização se torna crucial.

Uma abordagem promissora é a *Adaptive Instance Normalization* (AdaIN), que se mostrou eficaz na transferência de estilo em imagens, tradicionalmente aplicada a rostos e objetos. A AdaIN ajusta a média e a variância das características de conteúdo para corresponder às do estilo, permitindo uma adaptação flexível a diferentes domínios. Outra técnica relevante é o *Color Distribution Alignment* (CDA), que busca alinhar as distribuições de cores entre domínios, reduzindo a discrepância visual e melhorando a generalização do modelo.

Ambas as técnicas, AdaIN e CDA, compartilham o objetivo de lidar com a mudança de domínio em imagens, adaptando as características para minimizar as diferenças entre conjuntos de dados. Uma revisão bibliográfica abrangente sobre técnicas de normalização de imagens, como *Batch Normalization*, AdaIN e outras,

aplicadas à troca de domínio em imagens aéreas, pode fornecer insights valiosos para a escolha da abordagem mais adequada para cada problema específico.

A pergunta central que norteia esta pesquisa é: como técnicas de normalização adaptativa podem melhorar a capacidade de generalização de modelos de Deep Learning para segmentação de imagens aéreas e ortofotos em diferentes regiões ecológicas? As hipóteses levantadas são que a *Adaptive Instance Normalization* (AdaIN) e o alinhamento da distribuição de cores podem aprimorar o desempenho dos modelos em conjuntos de dados com características distintas.

Justificamos a importância do tema a partir dos dados obtidos, que revelam um amplo espaço para aprimoramento dos modelos de segmentação de imagens. A inteligência artificial treinada em uma cidade não necessariamente compreenderá as características de outra, devido às variações nas condições de iluminação, textura e contexto. Para superar essa limitação, propomos investigar técnicas de Normalização Adaptativa, que ajustam as características das imagens para minimizar as diferenças entre conjuntos de dados de diferentes domínios.

1.3 OBJETIVOS

Em primeiro momento, aprimorar o desempenho de redes neurais já existentes para segmentação de imagens, aplicando Normalização Adaptativa para aumentar sua capacidade de generalização entre diferentes domínios; consequentemente, tornando os modelos mais robustos e precisos, mesmo quando processam imagens de regiões distintas ou com variações nas condições de captura.

Depois, no seguimento dos experimentos, desenvolver e testar novas técnicas para que modelos de inteligência artificial consigam analisar imagens de satélite de diferentes regiões geográficas com alta precisão, superando o problema conhecido como "*domain shift*".

1.4 ORGANIZAÇÃO DA ESCRITA

Essa pesquisa é organizada em formato de artigos e estruturada em 4 (quatro) capítulos distintos. A fundamentação teórica apresenta uma revisão detalhada da literatura, incluindo os estudos que precederam e influenciaram esta tese. É seguida pelo primeiro artigo realizado para iniciar nossa pesquisa. Posteriormente, dedicamos

espaço para elucidação sobre a pesquisa e as considerações do estudo, resultado da metodologia utilizada no primeiro artigo. Por fim, mais 2 (dois) artigos complementam a investigação sobre adaptação de domínio e segmentação de vegetação em sensoriamento remoto.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ADAPTAÇÃO DE DOMÍNIO (DA) E SENSORIAMENTO REMOTO (SR)

Adaptação de Domínio (DA) e Generalização de Domínio (DG) são duas abordagens fundamentais no aprendizado de máquina que visam capacitar modelos a lidar com variações nos dados. Enquanto a DA, como a Adaptação de Domínio Não Supervisionada (UDA), foca em alinhar características entre domínios de origem e destino, a DG se destaca por sua capacidade de generalizar para novos domínios sem acesso prévio aos seus dados. Nesse cenário, Zhu *et al.* (2023, tradução nossa) afirmaram a incapacidade dos modelos treinados em manter sua performance ao lidar com novos cenários de teste; problema conhecido como mudança de domínio. Nesse contexto, essa problemática tem motivado o desenvolvimento de técnicas como a classificação entre cenários, que vem ganhando destaque no campo do sensoriamento remoto¹ (SR).

Zhu *et al.* (2023, tradução nossa), ao explorar a correlação entre o domínio de origem e o domínio de destino, revelaram que modelos relevantes podem ter melhor capacidade de generalização em relação ao domínio de destino. Contudo, para os autores (Zhu *et al.*, 2023, tradução nossa), a Adaptação de Domínio² (DA), como a principal técnica para classificação entre cenários, requer necessariamente acesso às amostras de destino para auxiliar no treinamento do modelo, uma condição difícil de ser satisfeita em aplicações do mundo real.

A Generalização de Domínio³ (DG), segundo Zhu *et al.* (2023, tradução nossa), tem atraído crescente atenção da pesquisa nos últimos anos. Dado um ou vários domínios de origem, a DG tende a aprender modelos que podem ter bom desempenho ao lidar com domínios de destino não vistos (inacessíveis). A DG também pode lidar melhor com a generalização fora da distribuição com menos restrições, tornando-a adequada para a classificação entre cenários. Notavelmente, pouca pesquisa foi realizada sobre a generalização de domínio no campo do sensoriamento remoto. Assim, podemos nos referir a este tipo de método como generalização entre cenários.

¹ Remote sensing.

² Domain adaptation.

³ Domains generalization.

Estudos recentes mostraram que as redes neurais convolucionais têm um forte viés para o reconhecimento de texturas em vez de formas (Zhu *et al.*, 2023, tradução nossa). Consequentemente, Zhu *et al.* (2023, tradução nossa) propuseram uma Rede de Separação de Estilo e Conteúdo⁴ (SCSN) para a generalização entre cenários de imagens de SR, a qual pode melhorar a capacidade de generalização e discriminação. Conforme os autores (Zhu *et al.*, 2023, p. 01, tradução nossa), “o módulo de Separação de Estilo e Conteúdo (SCS) usa a normalização de instância para obter as informações de conteúdo e, assim, garantir uma melhor capacidade de generalização”. Além disso, Zhu *et al.* (2023, tradução nossa) alegam que o recurso residual, que contém as informações de estilo, pode complementar as representações de recursos após o refinamento. Eles propuseram ainda “uma perda de separação para restringir o processo de separação de estilo e conteúdo. Os resultados experimentais e análises relevantes demonstraram, então, a eficácia da SCSN proposta em tarefas de generalização entre cenários” (Zhu *et al.*, 2023, p. 01, tradução nossa).

Conforme os autores (Zhu *et al.*, 2023, tradução nossa), comparadas com os métodos tradicionais de extração manual de características, as abordagens baseadas em aprendizado profundo podem aprender representações de características profundas em múltiplos níveis de forma ponta a ponta. Contudo,

[...] o aprendizado profundo é impulsionado por dados, e uma grande quantidade de dados de alta qualidade é necessária para treinar um modelo excelente. Com o desenvolvimento da tecnologia de satélites e UAVs, o volume e a diversidade dos dados de imagens de sensoriamento remoto continuam a aumentar de forma constante. Porém, a natureza demorada e trabalhosa do processo de rotulagem de dados e a baixa capacidade de generalização, mesmo de modelos bem treinados, tornam a mudança de domínio um problema muito comum, o que dificulta seriamente o desenvolvimento e a aplicação de métodos de classificação de cenas (Zhu *et al.*, 2023, p. 01, tradução nossa).

Nesse ponto, os autores (Zhu *et al.*, 2023, p. 01, tradução nossa) perceberam que a maioria dos métodos de classificação entre cenários emprega uma técnica chamada Adaptação de Domínio (DA). E que ao explorar a correlação entre os domínios de origem e de destino, a DA pode aprender características invariantes ao domínio que facilitam a adaptação a novos domínios.

⁴ Style and Content Separation Network.

Apesar disso, de acordo com (Zhu *et al.*, 2023, tradução nossa), os métodos de classificação entre cenários baseados em DA exigem não apenas os dados rotulados da origem, mas também os dados não rotulados do destino, para realizar manipulações como alinhamento de características⁵, correspondência de domínio⁶, aprendizado adversarial de domínio⁷, por exemplo. O que significa que “a acessibilidade aos dados de destino é essencial para esses métodos” (Zhu *et al.*, 2023, p. 02, tradução nossa). Além disso, outra desvantagem da DA é que ela “também requer retreinamento ou ajuste fino com base em novas amostras ao se adaptar a novos conjuntos de dados ou tarefas” (Zhu *et al.*, 2023, p. 02, tradução nossa).

Sabemos que, “até o momento, tem havido poucos estudos relevantes sobre generalização de domínio no campo do sensoriamento remoto. Portanto, nos referimos à sua aplicação na classificação entre cenários como generalização entre cenários⁸” (Zhu *et al.*, 2023, p. 02, tradução nossa). Com isso em vista,

Fora do campo do sensoriamento remoto, a pesquisa em DG está se desenvolvendo rapidamente em uma variedade de áreas, incluindo classificação de imagens, segmentação semântica, reconhecimento de ações, reidentificação de pessoas e compreensão de vídeo (Zhu *et al.*, 2023, p. 02, tradução nossa).

Exemplo disso são as Redes neurais convolucionais⁹ (CNNs).

2.1.1 Redes Neurais Convolucionais (CNNs)

As Redes neurais convolucionais (CNNs), de acordo com Zhu *et al.* (2023 *apud* Geirhos *et al.*, 2019, tradução nossa), “comprovadamente têm um forte viés para o reconhecimento de texturas em vez de formas”. A partir disso, as CNNs são mais sensíveis a texturas (estilo) do que a formas (conteúdo), o que pode levar a uma queda no desempenho quando há mudanças no estilo das imagens.

Nesse ponto, percebemos uma incapacidade desses modelos de aprendizado profundo (CNNs) de manterem sua alta performance quando confrontados com novos cenários de teste que diferem dos dados de treinamento, o que chamamos de

⁵ Aligment domain.

⁶ Domain matching.

⁷ Domain adversarial learning.

⁸ Cross-scene generalization.

⁹ Convolutional neural networks (CNNs).

mudança de domínio. Essa problemática é particularmente aguda no sensoriamento remoto, onde as imagens podem variar consideravelmente devido a fatores como condições de iluminação, cor, resolução, época do ano, região, e até mesmo características do sensor (no caso de imagens multissensores ou SAR).

Conforme Zhu *et al.* (2023, p. 02, tradução nossa), “quando se trata de tarefas de classificação de cenas de sensoriamento remoto, a informação de estilo também é muito importante”. Os autores (Zhu *et al.*, 2023, tradução nossa) salientam que aspectos da mesma cena diferem significativamente, como a iluminação, cor, resolução, entre outros. Sendo que os principais motivos são as diferenças nas condições de imagem, estações do ano e regiões diferentes. Essa grande variação no estilo das imagens de diferentes cenários dificulta a classificação, representando um desafio considerável. Apesar de informações de estilo em excesso poderem prejudicar a rede, uma quantidade adequada delas é crucial para garantir a capacidade de diferenciação da rede.

Em uma análise geral, entendemos que as principais contribuições de Zhu *et al.* (2023, tradução nossa) incluem a constatação de que a configuração da generalização de domínio é introduzida no campo do sensoriamento remoto. Também, a rede de Separação de Estilo e Conteúdo (SCSN), uma estrutura simples, mas eficaz, pode ser proposta para facilitar uma melhor generalização entre cenários.

Ao desembaraçar e reorganizar efetivamente as informações de estilo e conteúdo, os autores Zhu *et al.* (2023, tradução nossa) confirmaram que a SCSN melhora a capacidade de generalização do modelo, mantendo uma boa capacidade de discriminação. Para isso, eles melhoraram a capacidade de generalização do modelo extraindo as informações de conteúdo como as principais representações de características. Dessa forma, o módulo SCS foi proposto para aumentar o foco da rede nas informações de conteúdo usando Normalização por instância (IN)¹⁰. E o recurso residual refinado foi então usado para complementar as representações de recursos para manter a capacidade de discriminação da rede. Desse modo, ao medir a discrepância entre o recurso de saída e o recurso de conteúdo/estilo, a perda de separação também foi projetada para maior restrição, para garantir a separação entre as informações de estilo e conteúdo. Portanto, os resultados experimentais e análises

¹⁰ Instance normalization. Que “executa uma forma de normalização de estilo ao normalizar as estatísticas das características, ou seja, a média e a variância” (Ulyanov *et al.*, 2017; Huang; Belongie, 2017 *apud* Zhu *et al.*, 2023, tradução nossa).

relacionadas demonstraram a eficácia, generalizabilidade e estabilidade da SCSN proposta.

Para tanto, os autores dividiram as técnicas relacionadas à SCSN em quatro categorias: extração profunda de características¹¹, adaptação de domínio¹², generalização de domínio¹³ e manipulação de estilo¹⁴. A saber, “as DNNs¹⁵ exploram autonomamente representações de características mais profundas e exibem melhor capacidade de generalização de aprendizado” (Zhu *et al.*, 2023, p. 03, tradução nossa). Sendo que, como quando se trata do campo de Sensoriamento Remoto (SR) a maioria dos dados envolvidos são imagens, as Redes Neurais Convolucionais (CNNs) são comumente empregadas neste contexto, pois são projetadas para processar dados na forma de múltiplos arrays.

Assim sendo, enfatizamos que, embora as redes neurais profundas (DNNs) sejam excelentes em aprender representações de características complexas de forma automatizada (o que é uma vantagem sobre a extração manual de características), elas são orientadas por dados; o que significa que exigem grandes volumes de dados de alta qualidade e bem rotulados para treinamento. No SR, o processo de rotulagem é demorado e trabalhoso, o que, combinado com a baixa capacidade de generalização de modelos mal adaptados, torna a mudança de domínio um obstáculo sério para o desenvolvimento e aplicação de métodos de classificação de cenas.

2.1.2 Adaptação De Domínio Não Supervisionada

Quando tratamos de adaptação de domínio, “técnica de aprendizado de máquina comumente empregada para lidar com o problema de mudança de domínio, alcançando desempenho impressionante em tarefas de classificação entre cenários” (Zhu *et al.*, 2023, p. 03, tradução nossa), a Adaptação de Domínio Não Supervisionada (UDA¹⁶) é a técnica mais utilizada para lidar com a mudança de domínio, uma vez que ela utiliza dados rotulados do domínio de origem e dados não rotulados do domínio de destino para realizar o alinhamento de características, aprendizado de subespaço

¹¹ Deep feature extraction.

¹² Domain adaptation (DA).

¹³ Domain generalization (DG).

¹⁴ Style manipulation.

¹⁵ Redes neurais profundas.

¹⁶ Unsupervised domain adaptation.

e aprendizado adversarial, com o objetivo de reduzir as diferenças entre os domínios e melhorar o desempenho do modelo em dados não vistos.

Enquanto a DA (adaptação de domínio) funciona ao alinhar as características entre o domínio de origem (com dados rotulados) e o domínio de destino (com dados não rotulados), a DG (generalização de domínio) se concentra em aprender modelos que podem ter bom desempenho em domínios de destino não vistos (inacessíveis). Assim, a DG lida melhor com a generalização "fora da distribuição" com menos restrições, tornando-a ideal para o desafio da classificação entre cenários, na qual o acesso a dados de testes específicos pode ser impossível. Nesse âmbito, a generalização de domínio (DG), mais desafiadora em comparação com a adaptação de domínio (DA), emprega uma abordagem diferente ao se concentrar em como implementar a generalização entre domínios quando o domínio de destino é inacessível. Além disso,

[...] a DA requer retreinamento acompanhado por novas amostras ao lidar com novas tarefas, enquanto os modelos de DG podem ser usados várias vezes com apenas um treinamento. Devido à eficácia e utilidade da DG em aplicações práticas, grandes progressos foram feitos em pesquisas relacionadas nos últimos anos (Zhu *et al.*, 2023, p. 03, tradução nossa).

Zhu *et al.* (2023, tradução nossa) acrescentaram que existem duas técnicas principais usadas no aprendizado de representação, que é a forma mais popular de DG. Sendo a primeira técnica, aprendizado de representação invariante ao domínio¹⁷, a maneira mais direta e eficaz de garantir a capacidade de generalização do modelo; e a segunda, o desemaranhamento de características¹⁸, que tenta desemaranhar as características em partes públicas e específicas para uma melhor generalização (Li *et al.*, 2017; Piratla *et al.*, 2020 *apud* Zhu *et al.*, 2023, p. 03, tradução nossa).

Quanto à manipulação de estilo¹⁹, como as CNNs são mais sensíveis ao estilo do que ao conteúdo, seu desempenho provavelmente diminuirá quando o domínio mudar (Zhu *et al.*, 2023, tradução nossa). Desse modo, “uma ideia intuitiva é fazer com que o extrator de características preste mais atenção ao conteúdo do que à informação de estilo, o que aumentará a capacidade de generalização do modelo” (Zhu *et al.*, 2023, p. 03, tradução nossa).

¹⁷ Domain invariant representation learning.

¹⁸ Feature disentanglement.

¹⁹ Style manipulation.

Em consonância com Zhu *et al.* (2023, tradução nossa), entendemos o porquê da classificação de imagens de diferentes cenários em sensoriamento remoto ter sido um tema de pesquisa relevante nos últimos anos. Sabemos que as abordagens tradicionais, baseadas em Adaptação de Domínio (DA), buscam melhorar a generalização dos modelos aprendendo a partir de dados de diferentes domínios. No entanto, elas dependem do acesso aos dados do domínio de destino, o que nem sempre é possível em aplicações reais. Para solucionar esse problema em sua pesquisa, Zhu *et al.* (2023, tradução nossa) propuseram a Rede de Separação de Estilo e Conteúdo (SCSN), técnica de Generalização de Domínio (DG) que não requer acesso aos dados do domínio de destino para, dessa maneira, a SCSN melhorar a capacidade de generalização da rede ao focar nas informações de conteúdo das imagens, que foram mais relevantes para a tarefa de classificação, e complementarem com informações de estilo, que são mais específicas do domínio. Os resultados experimentais, então, demonstraram que a SCSN alcança desempenho de ponta em tarefas de generalização entre cenários.

Vimos que as CNNs, embora eficazes, têm um forte viés para o reconhecimento de texturas (estilo) em vez de formas (conteúdo), o que demonstra a eficácia da DG no SR com uma estrutura simples, mas eficaz, que foca no conteúdo. No SR, as informações de estilo podem variar drasticamente (iluminação, cor, resolução), prejudicando o desempenho da classificação. A solução para isso, conforme Zhu *et al.* (2023, tradução nossa), foi uma SCSN proposta para separar informações de estilo e conteúdo, uma vez que o módulo de separação de estilo e conteúdo (SCS) usa a normalização de instância (IN) para extrair informações de conteúdo, garantindo melhor generalização. Nesse caso, as informações de estilo (recurso residual refinado) complementam as representações de características, mantendo a capacidade de discriminação e uma "perda de separação" é utilizada para garantir essa distinção.

Nessa perspectiva, Chen *et al.* (2023, tradução nossa) examinam a segmentação semântica baseada em aprendizado profundo, que tem sido amplamente aplicada para a extração de construções²⁰. Os autores Chen *et al.* (2023, tradução nossa) afirmam que, devido ao *gap* de domínio, a extração de construções em imagens de sensoriamento remoto de alta resolução é difícil quando o modelo

²⁰ O estudo trata da extração das imagens de edifícios/prédios em meio urbano.

treinado em um conjunto de dados de origem é usado diretamente para testar em dados de destino. A partir disso, considerando que os humanos podem recuperar a memória para lidar com tarefas correlacionadas em diferentes domínios, mecanismos de memória foram desenvolvidos efetivamente para auxiliar na extração de características entre domínios.

Também, a eficácia dos mecanismos de memória depende muito da correlação da memória com a tarefa. Desse modo, a memória invariante ao domínio é crucial na tarefa de extração de construções entre diferentes domínios (Chen *et al.*, 2023). Para este fim, os autores (Chen *et al.*, 2023, tradução nossa) propuseram um método de adaptação de domínio não supervisionada com memória contrastiva (DA) com base em um novo mecanismo de memória. Especificamente, para facilitar o modelo a memorizar características invariantes ao domínio, primeiro conduziram uma estratégia de transferência de estilo de imagem baseada em normalização e um método adversarial baseado em discriminador no nível da imagem e no nível de características, respectivamente.

Posteriormente, realizaram um módulo de memória contrastiva para obter características invariantes ao domínio. Em especial, uma rede professor-aluno é explorada para ajudar na transferência de conhecimento por meio da destilação de conhecimento para melhorar o desempenho do módulo de memória contratada (MCD) (Chen *et al.*, 2023, tradução nossa). Além disso, para reduzir a distância entre os dois domínios, um banco de memória foi projetado para armazenar e atualizar as características de categoria obtidas do domínio de origem e, em seguida, a similaridade entre as características de categoria no domínio de destino e o banco de memória foi calculada. Os resultados dos experimentos entre domínios mostraram, por fim, que esse método proposto pode alcançar a extração ideal de construções.

Chen *et al.* (2023, tradução nossa) ressaltaram que com o desenvolvimento da tecnologia de Sensoriamento Remoto (SR), os dados com diferentes resoluções espaciais, múltiplas características espectrais e multimodais se tornam cada vez mais disponíveis. Isso redundava em que a investigação de construções a partir de imagens de SR se torne significativa para a exploração e compreensão humana do ambiente geográfico e ecológico, especialmente em planejamento e construção urbana, gerenciamento de desastres naturais e crises, entre outras aplicações (Chen *et al.*, 2023, tradução nossa).

Uma vez que a diferença entre os conjuntos de dados de origem e destino impede que o modelo gere resultados satisfatórios, a Adaptação de Domínio Não Supervisionada (UDA) emergiu como a principal estratégia para diminuir essa lacuna e viabilizar a transferência de conhecimento. Nesse contexto,

[...] muitos métodos UDA foram desenvolvidos para realizar o reconhecimento de construções. Shi *et al.* propuseram um método UDA alinhando as redes de nível de pixel e nível de representação, respectivamente. A adaptação de domínio (DA) em nível de pixel transforma os dados do domínio de origem em dados do domínio de destino através da rede adversarial generativa de ciclo (CycleGAN). A adaptação de domínio em nível de representação pode ser alcançada implementando o aprendizado adversarial entre imagens dos domínios de origem e destino. Este tipo de método utilizou um módulo baseado em rede adversarial generativa (GAN) para transferir o conhecimento das imagens, de modo que a imagem do domínio de origem pudesse ser alinhada com a imagem do domínio de destino (Chen *et al.*, 2023, p. 01, tradução nossa).

No caso da “tarefa de extração de construções UDA, como os dados do domínio de destino não têm rótulos, as informações da própria imagem devem ser exploradas completamente para obter características confiáveis” (Chen *et al.*, 2023, p. 02, tradução nossa).

Em uma síntese geral, entendemos que as principais contribuições de Chen *et al.* (2023, tradução nossa) versam sobre a utilização de uma rede UDA inspirada no mecanismo de memória humana, chamada MDANet, proposta para a extração de construções entre domínios. Um módulo MCD baseado em aprendizado contrastivo de pares positivos, desenvolvido para extrair características de construções invariantes ao domínio. Um módulo MCD incorporado ao banco de memória, projetado para reduzir o gap semântico entre os dados de origem e destino para a representação invariante ao domínio. Os autores (Chen *et al.*, 2023, tradução nossa) realizaram a extração de construções baseada em aprendizado profundo através da: extração de construções totalmente supervisionada, UDA no campo de SR, banco de memória e aprendizado contrastivo.

A extração de construções totalmente supervisionada baseada em aprendizado profundo é uma tarefa de segmentação binária no campo de SR. Para ser realizada, muitos métodos baseados em FCN²¹ foram aplicados pelos autores (Chen *et al.*, 2023, tradução nossa). Com pesquisas adicionais sobre várias tecnologias, incluindo, mas não se limitando a mecanismo de atenção, modo visual de co-ocorrência e estratégia

²¹ Fully convolutional network

de fusão multiescala, mais e mais métodos de segmentação semântica baseados em aprendizado profundo, foi possível promover a extração de construções. Contudo, os autores salientam que todos esses métodos exigem uma grande quantidade de dados rotulados para treinar o modelo. E que, “devido à existência de mudança de domínio, obter resultados satisfatórios geralmente é difícil ao usar um modelo treinado em um conjunto de dados, mas testado em outro conjunto de dados” (Chen *et al.*, 2023, p. 02, tradução nossa). A tecnologia UDA é amplamente utilizada no campo da segmentação semântica de imagens de SR (Chen *et al.*, 2023, tradução nossa). Em geral,

[...] a tarefa de DA pode ser concluída de duas maneiras: uma é usando um método de alinhamento no nível da imagem, como o modelo de transferência de estilo baseado em GAN, e outra é alinhando o domínio de origem com o domínio de destino no espaço de características (Chen *et al.*, 2023, p. 03, tradução nossa).

O banco de memória, por sua vez, “pode introduzir unidades de memória auxiliares para armazenar e reutilizar as informações relacionadas à tarefa, a fim de aumentar a capacidade da rede” (Chen *et al.*, 2023, p. 03, tradução nossa). Quanto ao aprendizado contrastivo, “como um membro importante do aprendizado auto-supervisionado, tem sido amplamente utilizado em visão computacional, processamento de linguagem natural e outros campos” (Chen *et al.*, 2023, p. 03, tradução nossa), sendo que “ele visa aproximar as amostras aumentadas o máximo possível no espaço de incorporação (*embedding*) e, em seguida, manter as diferentes amostras o mais distantes possível” (Chen *et al.*, 2023, p. 03, tradução nossa). Dessa forma, “restrição da perda contrastiva é imposta no encurtamento da distância entre o domínio de origem e o domínio de destino, e os recursos no banco de memória se tornariam mais invariantes ao domínio” (Chen *et al.*, 2023, p. 03, tradução nossa).

Em linhas gerais, Chen *et al.* (2023, tradução nossa), para melhorar a capacidade de generalização da extração de construções de HRSIs em diferentes domínios, realizaram uma MDANet nova e eficaz, inspirada em mecanismos de memória, projetada para atingir esse objetivo. Com Chen *et al.* (2023, tradução nossa), entendemos que o gap de domínio dificulta a extração de construções (edifícios) em imagens de SR de alta resolução quando modelos treinados em um conjunto de dados são aplicados diretamente a outro. Além disso, a DA tradicional exige dados rotulados e não rotulados do destino, o que pode ser solucionado com a

proposta da MDANet, demonstrando que a aplicação de mecanismos de memória e aprendizado contrastivo para melhorar a UDA na extração de construções, reduz a dependência de rótulos do domínio de destino e alcançam uma extração "ótima".

Wang *et al.* (2022, tradução nossa), por sua vez, tratam do mapeamento da terra através da utilização de algoritmos de aprendizado profundo. O mapeamento detalhado da cobertura da terra em áreas urbanas é crucial para o planejamento eficiente de recursos e o desenvolvimento sustentável das cidades. Algoritmos de aprendizado profundo têm se mostrado promissores nessa tarefa, utilizando imagens de alta resolução espacial (HSR). No entanto, “a limitação da anotação e a divergência das imagens multissensores sempre desafiam a transferibilidade do aprendizado profundo, dificultando o mapeamento em nível de cidade ou nacional” (Wang *et al.*, 2022, p. 01, tradução nossa).

Em se tratando do estudo de Chen *et al.* (2023, tradução nossa), primeiro eles usaram um método simples de alinhamento no nível da imagem para transferir o estilo da imagem, reduzindo inicialmente o gap entre o domínio de origem e o domínio de destino e facilitando a obtenção da memória correlativa. Depois, inseriram um discriminador projetado para alinhar as distribuições de dados de origem e domínio para extrair recursos de memória mais transferíveis. Em seguida, o módulo MCD foi projetado para reutilizar e atualizar a memória armazenada no banco de memória para realizar a transferência do conhecimento chave da construção. Desse modo, os experimentos dos autores (Chen *et al.*, 2023, tradução nossa) demonstraram que a exploração do mecanismo de memória pode ter um efeito positivo muito significativo na tarefa.

No caso de Wang *et al.* (2022, tradução nossa), eles propuseram um esquema para alavancar imagens aéreas de pequena escala com rótulos (fonte) para a classificação de imagens espaciais (alvo) de grande escala não rotuladas. Os autores (Wang *et al.*, 2022, tradução nossa), considerando as características do sensor, introduziram uma estrutura *Cross-Sensor Land-cover*, chamada LoveCS, para abordar as dificuldades da inconsistência de resolução espacial e diferenças espectrais.

Quanto ao design estrutural, a normalização entre sensores foi proposta para aprender automaticamente os pesos de normalização específicos do sensor, reduzindo assim as diferenças espectrais hierarquicamente. Ademais, um

decodificador denso multiescala²² foi proposto para fundir efetivamente as características multiescala de diferentes sensores.

Em relação à otimização do modelo, a adaptação de domínio com autotreinamento foi adotada, e a pseudo-rotulagem multiescala (MSPL) foi proposta para reduzir a divergência de escala causada pela inconsistência de resolução espacial. Depois, a eficácia do LoveCS foi testada em dados das três cidades de Nanjing, Changzhou e Wuhan na China.

Os resultados abrangentes mostraram que o *LoveCS* é superior aos métodos de adaptação de domínio existentes em tarefas entre sensores e possui boa generalizabilidade. “Comparado com os produtos de cobertura da terra existentes, os resultados obtidos têm a maior precisão e resolução espacial (1,0 m)” (Wang *et al.*, 2022, p. 01, tradução nossa), o que “fornece uma nova perspectiva para o mapeamento de cobertura da terra em larga escala com base em imagens HSR limitadas” (Wang *et al.*, 2022, p. 01, tradução nossa), pois “as imagens HSR tornam pequenos objetos no solo visíveis e podem ser usadas para explorar as estruturas finas dentro das cidades” (Wang *et al.*, 2022, p. 01, tradução nossa). No entanto, as características mais complexas também trazem mais desafios para a classificação da cobertura da terra (Marcos *et al.*, 2018 *apud* Wang *et al.*, 2022, p. 01, tradução nossa).

Em linhas gerais, os autores (Wang *et al.*, 2022, tradução nossa) propuseram uma inovadora estrutura *Cross-Sensor Land-cover (LoveCS)* para abordar as diferenças de resolução espacial e espectral, utilizando anotação limitada de imagens aéreas para promover o mapeamento de cobertura da terra por satélite em larga escala. Precisamente, uma normalização entre sensores (CSN) foi projetada para aprender pesos específicos do sensor, reduzindo assim as diferenças espectrais. Sendo que o CSN é um “módulo básico flexível que é aplicado hierarquicamente no *ResNet34*, produzindo o codificador entre sensores (*CSEncoder*)” (Wang *et al.*, 2022, p. 02, tradução nossa). O decodificador denso multiescala (*DMSDecoder*) também foi proposto para fundir efetivamente as características multiescala obtidas do *CSEncoder*. Os autores (Wang *et al.*, 2022, tradução nossa) adotaram, então, o autotreinamento para a adaptação de domínio, e a pseudo-rotulagem multiescala (MSPL), proposta para resolver o problema de inconsistência de resolução espacial.

²² Dense multi-scale decoder.

Os métodos de aprendizado profundo, conforme Wang *et al.* (2022, tradução nossa), comparados com os métodos convencionais que se concentram apenas em características feitas manualmente, são orientados por dados, onde as características mais representativas são extraídas hierarquicamente e automaticamente.

A adaptação de domínio não supervisionada (UDA), por sua vez, é uma subclasse de transferência de aprendizado, a qual visa transferir um modelo aprendido em um domínio de origem rotulado para um domínio de destino não rotulado sob a mesma tarefa (Wang *et al.*, 2022, tradução nossa). “Ao otimizar a perda de discrepância, o modelo pode encurtar a distância de características entre o domínio de origem e o de destino” (Wang *et al.*, 2022, p. 03, tradução nossa) e esses métodos podem ser divididos nas categorias treinamento adversarial²³ e autotreinamento²⁴ (Wang *et al.*, 2022, tradução nossa).

Os métodos de treinamento adversarial utilizam um discriminador para prever os rótulos de domínio e um extrator de características para gerar características invariantes, para confundir o discriminador (Ganin; Lempitsky, 2015 *apud* Wang *et al.*, 2022, p. 03, tradução nossa). “Na otimização adversarial, a discrepância de domínio é minimizada” (Wang *et al.*, 2022, p. 03). Os métodos de autotreinamento, por sua vez, alternam a atribuição de pseudorrótulos a dados não rotulados e usam as pseudoamostras para ajustar o modelo atual (Zhang *et al.*, 2019 *apud* Wang *et al.*, 2022, p. 03, tradução nossa). “Os pseudorrótulos se tornam mais precisos à medida que o modelo é progressivamente aprimorado” (Wang *et al.*, 2022, p. 03, tradução nossa).

No entanto, Wang *et al.* (2022, tradução nossa) elucidaram que esses métodos foram todos desenvolvidos sem considerar os desafios do mapeamento entre sensores, ou seja, a inconsistência de resolução espacial e as diferenças espectrais. Dessa forma, eles realizaram experimentos comparativos em oito métodos UDA publicamente disponíveis e no LoveCS proposto, para uma análise mais abrangente.

De forma geral, os experimentos abrangentes de UDA conduzidos com dados de três cidades diferentes da China provaram o excelente desempenho e generalizabilidade do LoveCS. A partir do estudo de ablação, Wang *et al.* (2022, tradução nossa) descobriram que cada módulo proposto melhora o desempenho de

²³ Adversarial training.

²⁴ Self-training.

diferentes aspectos e que os módulos são todos compatíveis entre si e se complementam bem. Ademais, um produto de cobertura da terra HSR para a cidade de Nanjing foi produzido eficientemente em 9 (nove) horas, usando apenas uma GPU. Também, comparado com os produtos de cobertura da terra existentes, o produto produzido pelo LoveCS tinha uma maior precisão, uma resolução espacial mais alta e continha informações mais detalhadas.

Wang *et al.* (2022, tradução nossa) investigaram a classificação de cenas em imagens de alta resolução espacial (HSR). Esse tema tem sido objeto de crescente interesse nos últimos anos e possui grande potencial para diversas aplicações, como planejamento urbano e classificação de cobertura da terra.

O aprendizado profundo tem sido amplamente explorado na classificação de cenas e alcançou alta precisão de classificação. Contudo, o método atual de classificação de cenas baseado em aprendizado profundo depende de um grande número de conjuntos de dados para treinamento, e a distribuição de características dos conjuntos de dados de teste padrão é a mesma que a dos conjuntos de dados de treinamento (Wang *et al.*, 2022, tradução nossa).

Na aplicação prática, não só é demorado e trabalhoso obter um grande número de dados de treinamento, mas também difícil atender a essas necessidades devido à mudança de dados entre diferentes domínios. Além disso, o uso direto de recursos extraídos pela rede neural convolucional (CNN) levará a um desempenho limitado devido à influência da migração de domínio (Wang *et al.*, 2022, tradução nossa).

Desse modo, como reduzir efetivamente o deslocamento de domínio entre os dados de treinamento e os dados de teste e descobrir informações discriminativas para a classificação de cenas é uma tarefa desafiadora. Nesse contexto, Wang *et al.* (2022, tradução nossa) propuseram a SSCN²⁵ para classificação de cenas entre domínios de sensoriamento remoto HSR.

2.1.3 Subdomain style compensation network (SSCN)

Na SSCN, os autores (Wang *et al.*, 2022, tradução nossa) introduziram o DSAN²⁶, no qual a discrepância média máxima local (LMMD) é introduzida pela primeira vez para a classificação de cenas para capturar as informações de grão fino

²⁵ Subdomain style compensation network.

²⁶ Deep Subdomain Adaptation Network.

para cada categoria e reduzir a migração de domínio entre os dados de treinamento e os dados de teste.

Para extrair informações mais discriminativas de imagens HSR²⁷, um módulo de normalização e restituição de estilo²⁸ (SNR) é desenvolvido e um mecanismo de atenção é adicionado ao módulo para melhorar o desempenho do modelo. Os resultados experimentais demonstraram que a estrutura SSCN proposta é superior aos métodos de última geração e tem um bom desempenho para a classificação de cenas entre domínios (Wang *et al.*, 2022, p. 1209, tradução nossa).

“A classificação de cenas de imagens de SR HSR é o processo de atribuir automaticamente uma imagem a um rótulo de classe que descreve a imagem corretamente” (Wang *et al.*, 2022, p. 1209, tradução nossa). Também, em muitas aplicações práticas de sensoriamento remoto, a classificação rápida e precisa de cenas de imagens de SR HSR é muito importante para o planejamento urbano e gestão da terra, por exemplo (Wang *et al.*, 2022, tradução nossa). Sobre os métodos desenvolvidos para classificação de cenas de imagens de sensoriamento remoto,

Os métodos de classificação de cenas de imagens de SR HSR podem ser divididos em métodos tradicionais de classificação de cenas e métodos de aprendizado profundo. De acordo com o nível de classificação de cenas tradicional, ela pode ser dividida em métodos baseados em características de baixo nível e métodos baseados em características de nível médio. Os métodos baseados em características de baixo nível e os métodos baseados em características de nível médio são baseados em características criadas manualmente, eles ainda são limitados para distinguir imagens com cenas mais desafiadoras e complexas. As redes neurais profundas são boas em aprender uma representação de cena eficaz para imagens de sensoriamento remoto. Apesar da eficácia, a classificação de cenas de sensoriamento remoto assume que os conjuntos de dados de treinamento e teste seguem a mesma distribuição. Quando a distribuição dos conjuntos de dados de treinamento e teste é diferente, métodos eficazes precisam ser propostos para garantir alta precisão de classificação nos conjuntos de dados de teste. Estudos recentes mostraram que muitos métodos de classificação de cenas entre domínios foram propostos para generalizar bem em dados com distribuições diferentes (Wang *et al.*, 2022, p. 1209, tradução nossa).

As principais contribuições de Wang *et al.* (2022, tradução nossa) foram a proposição de uma estrutura SSCN para lidar com a dificuldade de classificação de cenas para sensoriamento remoto HSR. No que os autores (Wang *et al.*, 2022, tradução nossa) introduziram o método entre domínios para abordar o problema de migração da tarefa de classificação de cenas de sensoriamento remoto HSR. O

²⁷ High spatial resolution.

²⁸ Style normalization and restitution.

módulo SNR foi usado para extrair informações de diagnóstico de imagens HSR e adicionar um mecanismo de atenção para melhorar o desempenho. Os resultados experimentais no conjunto de dados proposto mostraram que a estrutura tem um desempenho melhor na classificação de cenas entre domínios de imagens HSR.

Wang *et al.* (2022, tradução nossa) expuseram que o mapeamento da cobertura da terra em áreas urbanas é crucial, mas a limitação de anotações e a divergência de imagens multissensores (inconsistência de resolução espacial e diferenças espectrais) desafiam a transferibilidade do aprendizado profundo. Para resolver isso, propuseram a estrutura *LoveCS* (*Cross-Sensor Land-cover*) para classificar imagens aéreas rotuladas (fonte) em imagens espaciais não rotuladas (alvo), a qual aprende pesos de normalização específicos do sensor para reduzir diferenças espectrais hierarquicamente; funde características multiescala de diferentes sensores e reduz a divergência de escala. Estruturando-se como uma abordagem inovadora para mapeamento da terra em larga escala com imagens de alta resolução espacial (HSR), superando os desafios específicos da generalização entre diferentes sensores.

Nesse panorama, Zhao *et al.* (2024, tradução nossa) dissertaram sobre um novo método de adaptação de domínio aberto (DA) para a classificação de imagens heterogêneas de radar de abertura sintética (SAR²⁹). Neste tipo de situação, os dados de teste geralmente diferem da distribuição dos dados de treinamento e podem conter classes não vistas que não estão presentes nos dados de treinamento (Zhao *et al.*, 2024, tradução nossa). Considerando a imprecisão dos estudos existentes na classificação de amostras de classes desconhecidas no domínio de destino, Zhao *et al.* (2024, tradução nossa) propuseram uma estrutura de rede de autotreinamento dinâmico para distinguir classes conhecidas de classes desconhecidas, ajustando dinamicamente o limite discriminante.

Posteriormente, para melhorar ainda mais a precisão, um módulo de alinhamento fraco específico de domínio de autotreinamento foi projetado, o que abandona a ideia de alinhar rigidamente as características de origem e destino e alcança o alinhamento fraco do domínio de destino pela ideia de normalização de protótipo e extração de conhecimento específico de domínio.

²⁹ Synthetic aperture radar.

Por fim, “experimentos em dois conjuntos de dados de referência de imagens SAR heterogêneas mostraram que o método proposto melhorou a precisão da classificação em mais de 3% em relação aos métodos de última geração” (Zhao *et al.*, 2024, p. 01, tradução nossa).

Dessa maneira, sabendo que “as imagens de radar de abertura sintética (SAR) são altamente sensíveis devido ao mecanismo complexo do processo de formação de imagens” (Zhao *et al.*, 2024, p. 01, tradução nossa) e, portanto, “diferentes configurações de imagem, mudanças na estrutura do objeto e o efeito do acoplamento com o ambiente podem causar mudanças significativas nas imagens SAR” (Zhao *et al.*, 2024, p. 01, tradução nossa), “os métodos existentes usam um limiar fixo na discriminação de classes de amostras do domínio de destino” (Zhao *et al.*, 2024, p. 01, tradução nossa). Assim, Zhao *et al.*, (2024, tradução nossa) usaram a relação semântica profunda entre diferentes amostras desconhecidas do domínio de destino e o domínio de origem para ajustar dinamicamente os limiares, de modo que as amostras de destino obtivessem limiares de discriminação mais precisos (Zhao *et al.*, 2024, tradução nossa).

Em seguida, as ideias de normalização em lote e extração de conhecimento específico de domínio foram combinadas para alinhar cada amostra de destino ao protótipo de origem ou às suas amostras vizinhas no domínio de destino, “minimizando a entropia de similaridade entre cada pequeno lote de amostras e todas as outras amostras de destino e o protótipo de origem” (Zhao *et al.*, 2024, p. 02, tradução nossa).

Como metodologia adotada, os autores (Zhao *et al.*, 2024, tradução nossa) construíram codificadores de rede profunda como extrator de características e classificador de domínio para classificação OSDA do domínio de destino. Sendo que, o objetivo do autotreinamento foi permitir que o modelo aprendesse gradualmente as características de classes conhecidas no domínio de destino de amostras simples para complexas (Zhao *et al.*, 2024, tradução nossa).

Também, para melhorar a precisão do algoritmo em comparação com os métodos OSDA existentes, Zhao *et al.* (2024, tradução nossa) selecionaram dados de amostra pseudo-rotulados com alta confiança para o autotreinamento. Pois, uma vez que a estrutura da rede treinada nos dados de origem e contendo a invariante estilística dos dados de destino fosse suficientemente robusta, poderiam usar suas previsões para gerar pseudo-rótulos para os dados de destino com confiança

suficiente. Em seguida, Zhao *et al.* (2024, tradução nossa) propuseram que, após o aprendizado inicial de amostras de origem pura, comesçassem a incorporar progressivamente amostras de destino na tarefa de aprendizado da rede, onde uma etapa de avaliação de pontos de interrupção de autotreinamento permite selecionar amostras de destino conhecidas de forma determinística (Zhao *et al.*, 2024, tradução nossa).

Por fim, os resultados experimentais comparativos mostraram que o método proposto melhorou muito a precisão da classe desconhecida e o desempenho geral da classificação foi melhor do que o dos métodos SOTA. Somado a isso, a eficácia do método proposto também foi comprovada por estudos de ablação (Zhao *et al.*, 2024, tradução nossa).

Zhao *et al.* (2024, tradução nossa) apresentaram que a classificação de imagens SAR heterogêneas enfrenta a mudança de domínio e, crucialmente, pode conter classes não vistas no domínio de treinamento (problema de Adaptação de Domínio Aberto - OSDA). Sendo que, métodos existentes usam limiares fixos, resultando em imprecisão para classes desconhecidas. A solução seria a proposta de uma estrutura de rede de autotreinamento dinâmico para distinguir classes conhecidas das desconhecidas, ajustando dinamicamente o limite discriminante, o que se torna um método que não apenas lida com a mudança de domínio em dados SAR complexos, mas também é capaz de identificar e classificar com precisão classes desconhecidas, essencial para aplicações práticas onde o domínio de destino pode apresentar surpresas.

Jha e Banerjee (2023, tradução nossa) propuseram aprender extratores de características universais que podem ser usados em vários domínios simultaneamente, inspirados pelo sucesso da aprendizagem por transferência (TL). No contexto da classificação de imagens hiperespectrais (HSI), a falta de amostras de treinamento rotuladas devido ao alto custo e a restrição ao aprendizado de domínio único são um desafio. Para superar esses desafios, Jha e Banerjee (2023, tradução nossa) propuseram uma solução que combinasse os problemas de aprendizado multidomínio (MDL) e aprendizado de poucos exemplos (FSL) para classificação de HSI.

O objetivo dos autores (Jha; Banerjee, 2023, tradução nossa) foi treinar uma rede altamente compartilhável com todos os domínios no regime de treinamento de baixo número de exemplos. Chamaram a rede de rede multidomínio de poucos

exemplos (MDFS-Net), que compartilha a maioria dos parâmetros do modelo (especificamente, parâmetros de convolução e camada densa) entre domínios, mantendo as camadas de normalização em lote específicas do domínio separadas para capturar as características do domínio (Jha; Banerjee, 2023, tradução nossa).

Para abordar o problema de *overfitting*³⁰ em modelos de poucos exemplos, complementaram a tarefa principal de classificação com uma tarefa auxiliar auto-supervisionada. Testaram o método proposto em cinco conjuntos de dados HSI de referência e descobriram que a MDFS-Net supera consistentemente as linhas de base relevantes de forma convincente (Jha; Banerjee, 2023, tradução nossa).

A abordagem dos autores (Jha; Banerjee, 2023, tradução nossa) ofereceu uma solução promissora para a classificação de HSI em sensoriamento remoto (RS), permitindo o projeto de um sistema de classificação unificado que pode funcionar com vários locais HSI (domínios) e menos amostras rotuladas.

Uma vez que a tarefa de classificação de imagens hiperespectrais (HSI) lida com a atribuição de classes semânticas de cobertura da terra correspondentes aos pixels da imagem e, tradicionalmente, é resolvida usando abordagens *ad hoc* de aprendizado de máquina (Jha; Banerjee, 2023, p. 01, tradução nossa), os métodos de aprendizado profundo demonstram um desempenho superior, “devido à sua capacidade de aprendizado de características espectrais-espaciais orientado por dados” (Jha; Banerjee, 2023, p. 01, tradução nossa).

As principais contribuições de Jha e Banerjee (2023, tradução nossa) são resumidas na proposição do conceito de FS-MDL (*few-shot multidomain learning*) no contexto da classificação HSI e apresentação de uma solução inovadora, MDFS-Net, para enfrentar esse desafio. Jha e Banerjee (2023, tradução nossa) apresentaram uma estratégia intuitiva de meta-aprendizado para treinar parâmetros do modelo, que são predominantemente compartilhados entre domínios. Para reter informações específicas do domínio, empregaram parâmetros de normalização em lote separados para cada domínio. Para lidar com o *overfitting*, introduziram o conceito de minimizar a incerteza de recursos e incorporaram uma tarefa auto-supervisionada para reconstrução de amostras.

Os autores (Jha; Banerjee, 2023, p. 01, tradução nossa) realizaram experimentos em cinco conjuntos de dados HSI de referência, tratando cada um como

³⁰ Ajuste excessivo.

um domínio distinto, e introduziram protocolos de treinamento e avaliação para a tarefa FS-MDL. A MDFS-Net, conforme proposta, “superou todas as linhas de base relacionadas por uma margem mínima de 7% na precisão geral média (OA)” (Jha; Banerjee, 2023, p. 02, tradução nossa).

Nesse contexto, a abordagem tradicional para resolver problemas de classificação envolve o treinamento de modelos de inferência específicos para cada domínio a partir das distribuições de dados disponíveis. O objetivo dos autores (Jha; Banerjee, 2023, tradução nossa), no entanto, foi aprender múltiplas distribuições de dados simultaneamente. Portanto, Jha e Banerjee (2023, tradução nossa) mostraram que na classificação de imagens hiperespectrais (HSI), há uma escassez de amostras rotuladas e uma restrição ao aprendizado de domínio único, o que dificulta a criação de extratores de características universais. Para resolver, propuseram uma MDFS-Net (rede multidomínio de poucos exemplos), que combina os desafios de aprendizado multidomínio (MDL) e aprendizado de poucos exemplos (FSL), uma solução promissora para a classificação de HSI em múltiplos locais e com poucas amostras rotuladas, permitindo um sistema unificado e eficiente.

Por meio da análise desses estudos, compreendemos que a mudança de domínio é um obstáculo significativo; o que implica reconhecer e combater o *domain shift* para a aplicabilidade de modelos de aprendizado profundo em cenários reais, especialmente no SR. Também, a transição da DA para a DG (e suas variações, como OSDA e FS-MDL) é um caminho necessário, pois permite que os modelos funcionem sem a exigência irreal de acesso constante a dados rotulados do domínio de destino.

De modo geral, entendemos que a chave para a DG e DA mais eficazes reside em técnicas que possam separar as características relevantes (conteúdo/formas) das irrelevantes ou variáveis (estilo/texturas), e reorganizá-las para uma melhor generalização. Somado a isso, a incorporação de mecanismos de memória e estratégias de autotreinamento (*self-training*) são abordagens promissoras para que os modelos aprendam características invariantes e gerem pseudorrotulos (*pseudo-labels*) confiáveis, reduzindo a dependência de dados rotulados. Dessa forma, a complexidade e variabilidade dos dados de SR (imagens multissensores, SAR, HSI, diferentes resoluções, condições ambientais) tornam-no um campo de aplicação e teste perfeito para o desenvolvimento e validação dessas técnicas avançadas de generalização.

Essas considerações alinham-se diretamente com a proposta de Normalização Adaptativa e Aprendizado Híbrido para adaptação de domínio em análise de vegetação transgeográfica. A discussão sobre a variação entre domínios constitui um desafio crítico para modelos de aprendizado profundo em Sensoriamento (SR), especialmente quanto aplicados ao estudo da vegetação em diferentes contextos geográficos. Essa complexidade surge da necessidade de integrar dados heterogêneos, provenientes de múltiplas fontes (como diferentes satélites), regiões com características ecológicas distintas e condições atmosféricas variáveis.

As soluções discutidas, como DA (adaptação de domínio) e DG (generalização de domínio), e suas variações, ressoam com os pilares metodológicos da normalização adaptativa e do aprendizado híbrido. A normalização adaptativa atua no nível das representações internas do modelo, ajustando estatísticas relacionadas ao estilo (domínio) enquanto preserva características semânticas do conteúdo (como a estrutura da vegetação). Paralelamente, o aprendizado híbrido oferece um arcabouço para integrar estratégias supervisionadas (com poucos exemplos rotulados) e não supervisionadas (via pseudorrotulos), mitigando a dependência de grandes conjuntos de dados anotados.

Também, um aspecto crítico destacado foi a capacidade de discriminar entre características invariantes (como forma das folhas) e variações de domínio (iluminação, textura etc.), o que é feito por técnicas como Normalização por domínio, redes adversariais (GANs) pra alinhamento de distribuições e módulos de memória (que funcionam como repositórios dinâmicos de padrões vegetacionais). O resultado é uma estrutura analítica particularmente adequado à realidade do sensoriamento remoto, onde a generalização robusta entre contextos geográficos distintos não é mera conveniência metodológica, mas imperativo científico.

3 PRIMEIRO ARTIGO: DEEP LEARNING MODELS FOR VEGETATION SEGMENTATION

Para a realização desta pesquisa e dos experimentos descritos na Seção 3.1, foram utilizados dois conjuntos de dados de regiões distintas da América Latina: a Universidade Católica Dom Bosco (UCDB) em Campo Grande, Brasil, e a Universidad Politécnica Salesiana (UPS) em Quito, Equador.

Este artigo ainda não foi publicado em uma revista científica, mas há planos de submetê-lo para publicação assim que possível.

3.1 CROSS-DATASET EVALUATION OF DEEP LEARNING MODELS FOR VEGETATION SEGMENTATION IN LATIN AMERICA

Cross-Dataset Evaluation of Deep Learning Models for Vegetation Segmentation in Latin America

Max Hiroito Tieti^{a,*}, Patricia Pedrozo Lamberti^b, Alexandre Magalhães Albuquerque^c, Cesar Ivan Alvarez^d, Hemerson Pistori^{a,b}

^a*Universidade Federal de Mato Grosso do Sul, Campo Grande, Brazil*

^b*Universidade Católica Dom Bosco, Campo Grande, Brazil*

^c*Universidade Federal Rural do Rio de Janeiro, Seropédica, Brazil*

^d*Environmental Research Group for Sustainable Development (GIADES), Environmental and Civil Engineering, Universidad Politecnica Salesiana, Quito, Ecuador*

Abstract

This study investigates the generalization capabilities of deep learning models for image segmentation by training them on one dataset and testing them on another. We utilized two datasets from distinct regions in Latin America: the Universidade Católica Dom Bosco (UCDB) in Campo Grande, Brazil, and the Universidad Politécnica Salesiana (UPS) in Quito, Ecuador. These locations, while parts of Latin America, exhibit notable ecological differences that challenge model adaptability. Three state-of-the-art segmentation architectures—Fully Convolutional Networks (FCN) [12], DeepLabV3 [17], and SegFormer—were [30] employed to evaluate their performance across these datasets. Our findings reveal that while all models demonstrate rich segmentation capabilities, FCN outperforms others in cross-dataset evaluations, achieving high values in F1-score, IoU, and accuracy. The study highlights several areas for improvement, including normalization adaptation, network enhancement, and parameter fine-tuning. Future work will incorporate additional segmentation classes, LiDAR data, and datasets from different hemispheres to further test and refine model generalization. This research provides valuable insights into the application of machine learning models for ecological and urban planning, emphasizing the importance of model versatility across diverse environments.

Keywords: Drone 1, Sustainability 2, Urban Green 3, Ecological Monitoring 4, Geographic Variability 5

1. Introduction

The development of urbanization in cities has significantly changed soil use and permeability over the years. Several urban centers have their vegetative area territories being reduced by built-up and impermeable areas such as Vietnam, which has deteriorated to an extremely high degree in terms of vegetation cover, [23] and China, where it's affected by land surface temperature [13]. This has a direct and detrimental impact on the vegetation in various cities worldwide. [31]. There is concern for the development of sustainable cities and their environmental planning according to the Sustainable Development Goals (SDGs) 11.

Images obtained through remote sensing, like UAV (unmanned aerial vehicle) or satellite, have been increasingly used for studies in urban areas [22] [1]. Some articles use this technology to identify vegetation, green spaces, and their impact as Lotfata [19], Richards and Belcher [24] and Li et al. [18]. To analyze these images, common methods

of artificial intelligence, such as classification [5] [8] and segmentation [11] [15] are used, for example.

Segmentation is common use in urban areas for flat roofs mapping stains [7],

The advent of deep learning has significantly advanced the field of image segmentation, enabling the development of sophisticated models capable of identifying and classifying complex patterns within images. This study seeks to explore the generalization capabilities of various state-of-the-art segmentation models by training them on one dataset and testing their performance on another, distinct dataset. The primary objective is to evaluate how well these models can adapt to new, unseen data, which is critical for their application in real-world scenarios where data variability is inevitable. These capabilities are particularly important in fields such as vegetation analysis and urban planning, where accurate segmentation of aerial and orthoimages is essential for effective decision-making and resource management.

The integration of new technologies such as LIDAR (Light Detecting And Ranging), which can precisely measure the height of objects, has further enhanced the accuracy of image segmentation. Many studies have leveraged this data to obtain more comprehensive and accu-

*Corresponding author

Email addresses: max_hiroito@ufms.br (Max Hiroito Tieti), ra866795@ucdb.br (Patricia Pedrozo Lamberti), ppgctia@ufrj.br (Alexandre Magalhães Albuquerque), calvarezm@ups.edu.ec (Cesar Ivan Alvarez), pistori@ucdb.br (Hemerson Pistori)

rate findings. Additionally, NIR (Near-Infrared) imaging is frequently used for analyzing vegetative states. However, these advanced technologies are often costly, making them less accessible. In contrast, RGB (red, green and blue) images are more affordable and easier to acquire, offering a practical alternative for many researchers. Studies utilizing RGB images for vegetation segmentation include those by Ayhan and Kwan [2], Song et al. [25], and Wagner and Hirye [26], demonstrating the effectiveness of this approach in achieving accurate segmentation results.

In this research, we utilized two datasets from different regions in Latin America to conduct our experiments. The first dataset was collected from the UCDB (Universidade Católica Dom Bosco) in Campo Grande, Brazil, characterized by its cerrado vegetation. The second dataset was sourced from the UPS (Universidad Politécnica Salesiana) in Quito, Ecuador, where the vegetation is influenced by the Amazon rainforest. Although these two locations are geographically within the same continent, they present distinct ecological and environmental differences that pose challenges for model generalization.

Our study employs three advanced deep learning architectures for segmentation: FCN (Fully Convolutional Networks), DeepLabV3, and SegFormer. Each of these models was trained on one of the datasets and subsequently tested on the other. This cross-dataset evaluation approach allows us to rigorously assess the models' ability to generalize beyond the specific characteristics of the training data.

Additionally, this work identifies several avenues for further improvement, including the adaptation of normalization techniques, the exploration of more sophisticated network architectures, and the fine-tuning of model parameters. By integrating LiDAR data and expanding the number of segmentation classes, we aim to enhance the models' performance and robustness. Future research will also involve using datasets from different parts of the world, including the other hemisphere, to further investigate the global applicability of these segmentation models.

Through this study, we aim to contribute to the broader understanding of deep learning-based image segmentation and its potential for ecological monitoring and urban planning across diverse environments. Accurate image segmentation is crucial for various practical applications. In ecological monitoring, it enables precise tracking of vegetation changes, habitat mapping, and biodiversity assessment, which are essential for conservation efforts and environmental management. In urban planning, accurate segmentation of aerial images assists in land use classification, infrastructure development, and disaster management by providing detailed insights into urban sprawl, construction progress, and areas susceptible to natural hazards.

Moreover, the ability to generalize across different geographical datasets ensures that these models can be deployed in a variety of real-world scenarios, addressing the inherent variability in ecological and urban landscapes. This adaptability is particularly important given the increasing availability of diverse satellite and aerial imagery

from around the globe. By enhancing versatility of segmentation models, our research addresses a critical gap in the current state of machine learning applications, creating the way for more reliable and widespread use of these technologies.

The findings will provide insights into the challenges and opportunities of applying machine learning models to varied geographical datasets, ultimately guiding the development of more versatile and reliable segmentation tools.

2. Related Works

University campuses are places for study with easy access for students to conduct their experiments. Many researchers also use these spaces for their work, and studies involving vegetative detection or segmentation within urban environments are no exception. Some authors who have conducted studies involving vegetation in such spaces, for example, Cetin and Yastikli [3] use Yildiz, a university campus in Istanbul; Gibril et al. [10] utilize the space at the USM (University of Science, Malaysia), Penang, Malaysia; Fraser and Congalton [9], who collect their dataset from forest properties managed by the UNH (University of New Hampshire); and Zheng and Wu [32], who conduct their study in the Beijing Forestry University's and Shenzhen's Bao'an District, China; Phang Nga, Thailand.

To recognize different types of vegetation or vegetation that blends into the urban space, one of the techniques used is semantic segmentation, which can more precisely separate the classes for outlining irregular edges, as is the case with vegetation, for example. Some authors who use this segmentation technique for the urban vegetative context include: Song et al. [25], Correa Martins et al. [6], Jiang et al. [16], Wang et al. [28], Wagner and Hirye [26], Martins et al. [20], Guo et al. [11], He et al. [14], and Men et al. [21]. Some studies use only binary segmentation divided into just two classes: the area of interest, in this case, some type of vegetation and the background, while others use more classes in their segmentations, such as trees, low vegetation, and others. Examples of articles that use more than two classifications in their studies include: Song et al. [25], Wang et al. [28], Xie et al. [29], and Wang et al. [27].

Studies have demonstrated that models trained on one dataset often underperform when tested on another dataset with different characteristics, such as those arising from geographical, ecological, or sensor differences. For instance, Chatterjee et al. [4] explored the use of unsupervised learning techniques for classifying hybrid and dual-polarized SAR (Synthetic-aperture radar) images, highlighting the importance of transfer learning to improve model generalization across different image modalities. Similarly, [?] addressed the domain shift problem by enhancing the HR-Net architecture and utilizing adversarial domain adaptation techniques to improve land use classification accuracy on diverse remote sensing datasets.

3. Materials and Methods

3.1. Study Area

The areas selected for this study were two university campuses in different countries of Latin America. One university is located in the central-western part of Brazil, in Campo Grande, the capital of the state of Mato Grosso do Sul, the UCDB which is in a region where the predominant vegetation is the cerrado. The other university is situated in the capital of Ecuador, in Quito, the UPS, where the vegetation includes part of the Amazon forest. [11](#)

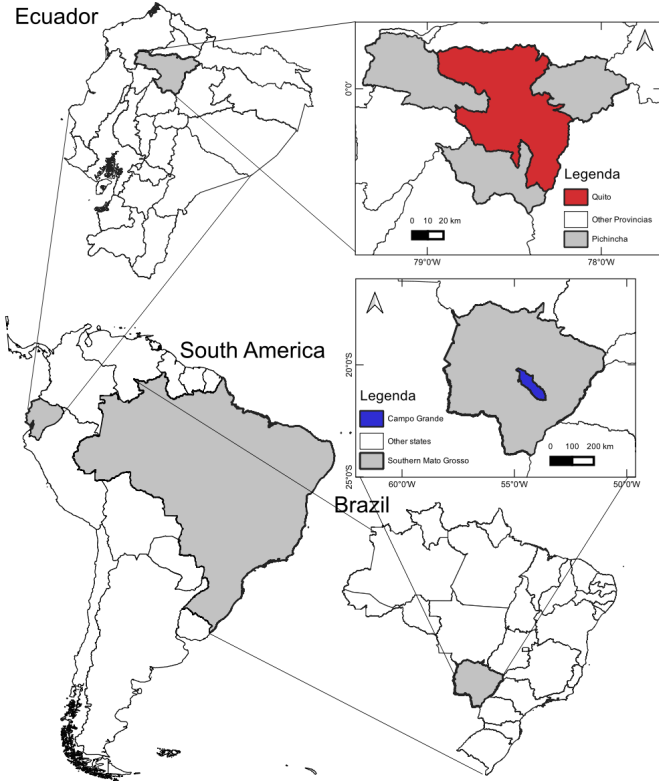


Figure 1: Location of the two cities chosen for the study

3.2. Aerial images

For our dataset, we collected aerial images using a unnamed aerial vehicle (UAV) (DJI Phantom Pro 4) over two university campuses in Latin America: UCDB and UPS. The images from UPS University covered an area of 183 square kilometers, while those from UCDB spanned 473 square kilometers. The imagery session at UCDB was conducted on the 20th of December, 2023, around noon in local hours. All images were captured in RGB and were subsequently processed using Agisoft Metashape to generate a single aerial image (orthophoto) for each location. This method allowed us to create a comprehensive and detailed visual dataset of the campus environments. The aerial images of UPS University in Quito were captured on May 19, 2023, at around 11 am in local hours, using a UAV (DJI Phantom Pro 4) with a flight of 100 meters.

These images were then processed using Pix4D software to generate orthophotos [2](#)

3.3. Annotation Process

The annotation process for the dataset was conducted using a combination of manual and semi-automated methods. Initially, human annotators manually labeled the images using QGIS software, focusing on four distinct classes: Vegetation, Grass and Others (or background). However, due to some inconsistencies encountered—such as difficulty in distinguishing between trees and bushes—the team streamlined the classification into three categories: Vegetation and Grass.

As the complexity of the shapes increased, the manual annotation process proved insufficient for capturing intricate details accurately. To address this, the team incorporated SAM (Segment Anything Module), a tool designed to facilitate the annotation of complex shapes. Despite the automation introduced by SAM, manual corrections were still necessary to rectify errors and refine the annotations further.

The team opted to exclude point clouds from the analysis due to the unavailability of precise height data typically sourced from LiDAR. Instead, Cloud Points and Height Maps generated by Pix4D and Agisoft Metashape were employed, providing the necessary detail for examining vegetative structures on university campuses. The analytical process involved segmenting these structures using an automated script, which identified and delineated each vegetative feature within the images. This script processed the image data to extract approximately 8000 distinct polygons, each representing a unique vegetative structure. These polygons were generated based on color segmentation techniques, where each color in the segmented masks corresponded to a different class of vegetation. The resulting dataset comprised of these polygons forms a robust basis for the detailed spatial analysis of green biomass (vegetation?), as illustrated in Figure [4](#)

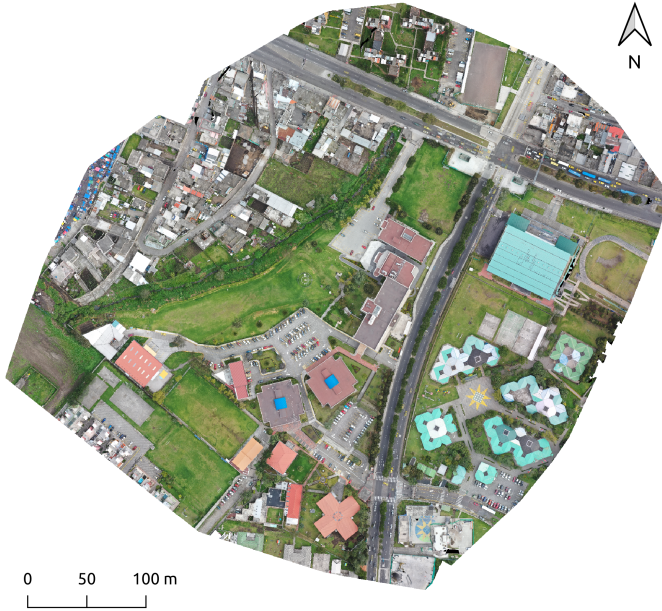
3.4. Preprocessing and Dataset

For the preprocessing and dataset creation, we utilized annotations made with QGIS and employed rasterio to generate masks for both the UPS and UCDB cases. Each image and its corresponding mask were tiled into 1080 x 1080 pixels. We implemented a small stride to allow overlapping between adjacent tiles, which helps in maintaining continuity across the dataset. Data augmentation was limited to flipping the images horizontally.

For the color coding of the masks, red was used to represent grass, blue for green areas, and black for other categories. Each dataset was split into training and evaluation sets, with 80% of the data used for training and 20% for evaluation. To assess the models' performance effectively and understand their generalization capabilities,



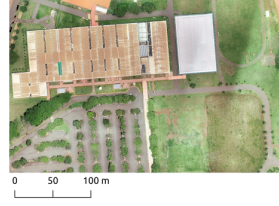
(a) UCDB aerial image



(b) UPS aerial image

Figure 2: Aerial images of the Universities

UCDB



UPS

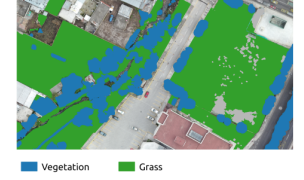
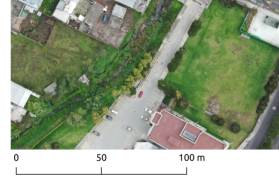
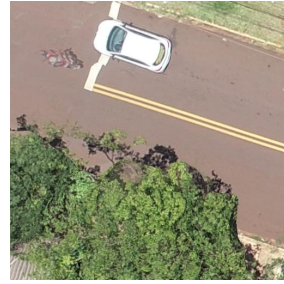
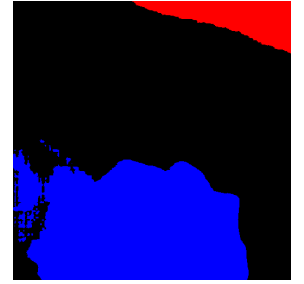


Figure 3: Orthophoto slice examples of the universities used in this study and their annotations; Universidade Católica Dom Bosco-UCDB above and Universidad Politécnica Salesiana de Quito-UPS below



(a) Ground Truth



(b) Anotation

Figure 4: Image exemplifying how the vegetation segmentation was performed in the images and the colors used to represent each category. Blue color for trees and bushes, red color for grass, and black color for others.

we utilized cross-dataset testing where the test dataset for models trained on UCDB was UPS, and conversely, the test dataset for models trained on UPS was UCDB. This setup allowed us to evaluate the models' generalization capabilities across different geographical contexts.

3.5. Models

Deeplav3 Architecture. It represents a cutting-edge approach to semantic segmentation, using spatial pyramid pooling and atrous convolution [29]. These features are instrumental in managing scale variation among objects within images, a critical factor for achieving fine-grained segmentation accuracy. The integration was made with either ResNet-101, DeepLabV3 models benefit from reliable feature extraction capabilities.

The DeepLabV3-ResNet101 model utilizes a ResNet-101 backbone, starting with a custom convolutional layer tailored to specific input channels. This layer connects to ResNet's deep residual layers, which enhance feature learning and refinement crucial for detailed segmentation. The architecture concludes with a convolutional layer specifi-

cally designed to output the required number of classes, ensuring the model effectively captures and aligns features for accurate object delineation in images.

FCN with ResNet-50 Backbone. The FCN is pivotal in the field of image segmentation, enabling the processing of entire images in a single, efficient sweep [30]. Unlike traditional convolutional networks that output class scores, FCNs produce pixel-wise segmentation maps that delineate the boundaries of objects directly. This architecture adapts standard CNN designs into a fully convolutional format, allowing it to handle images of any size and provide precise spatial outputs, which are crucial for detailed segmentation tasks.

The configuration used the ResNet50 backbone. It begins with a convolutional input layer focused to accommodate varying channels, making it versatile for images with different depth channels. This is followed by the deep and robust residual blocks of ResNet50, which are essential for capturing complex patterns and features within the image data. The network concludes with a specialized convolutional classifier designed to map these features into the segmentation classes denoted by output classes. This setup ensures the maintain of the integrity of spatial relationships within the image and also enhances the resolution of the output through learned upsampling techniques.

SegFormer. It stands out as a transformative model in image segmentation, integrating the strengths of Transformers with a hierarchical design Khaled et al. [17]. This architecture employs a Transformer-based backbone, which allows for dynamic and adaptive feature representation at multiple scales. Unlike traditional convolutional approaches, SegFormer effectively captures both local and global context, providing a comprehensive understanding of complex scenes. Its hierarchical structure enables detailed analysis at various resolutions, making it exceptionally good for handling diverse and intricate spatial details in images.

The SegFormer was configured for this study in a customized to attend to the inputs and outputs requirements. The model’s core is the transformer-based backbone, tailored to efficiently process these inputs and dynamically adapt to the semantic complexity of the scene. The segmentation head of the model is finely tuned to generate class delineations for output classes, leveraging the model’s ability to handle multi-scale features.

3.6. Training and Validation

Training. The training of the segmentation models is conducted using a structured and iterative approach to optimize performance. Although a fixed learning rate of 0.001 was ultimately chosen for the final training phase based on comprehensive experiments with different learning rates, these preliminary tests provided valuable insights into the models’ convergence behavior and sensitivity to this hyperparameter. The process starts by setting the model to

train mode and initializing the running loss. Training data are loaded in batches from the dataloader, and for each batch, data are transferred to the designated computing device. The optimizer is reset at the start of processing each batch to prevent carryover of gradients from previous iterations. Model outputs are then generated, followed by the calculation of loss using a predefined criterion. This loss is back-propagated through the model to update the parameters. Additionally, an IoU (intersection over union) evaluation is conducted batch-wise to track the training progress in terms of accuracy and intersection over union metrics across classes.

3.7. Experimental Setup

The computational experiments were conducted on a high-performance workstation. The system is powered by a 13th Gen Intel® Core™ i5-13500 CPU, featuring 20 logical processors and multi-threading capability with 14 cores per socket. The CPU operates with a maximum frequency of 4.8 GHz, providing the necessary speed for computationally intensive tasks.

The machine has a substantial 31 GB of RAM and graphic processing is managed by an NVIDIA Corporation GA104GL [RTX A4000], which offers powerful GPU acceleration crucial for speeding up model training and inference phases. The workstation runs on Ubuntu 22.04.4 LTS, ensuring a stable and secure environment for developing and testing advanced machine learning models. Storage is handled by a 456 GB NVMe drive, providing fast data access and ample space for datasets and applications.

4. Results

Overall, the aggregated metrics—accuracy, precision, recall, F1-score, Intersection over Union (IoU), specificity, and Dice coefficient—provide a clear snapshot of the segmentation performance across models. In general, models evaluated on the same dataset (intra-city) tend to achieve higher metric values than when tested on a different dataset (cross-city), underscoring the impact of domain-specific characteristics on model performance. For instance, models trained on UCDB typically perform better on UCDB data compared to when they are evaluated on UPS, and vice versa. This highlights the inherent challenges in achieving robust cross-domain generalization.

In addition, noticeable trends emerge regarding optimizer selection and network architecture. Comparisons indicate that certain optimizers (such as SGD, Adam, or Adagrad) consistently lead to higher aggregated performance metrics, while differences between architectures—namely FCN with ResNet50 versus SegFormer—suggest that one may be more robust across varying datasets. Preliminary experiments with different learning rates (0.01, 0.001, and 0.0001) also reveal that this hyperparameter significantly influences overall performance; a more detailed hyperparameter analysis is presented in a dedicated subsection.

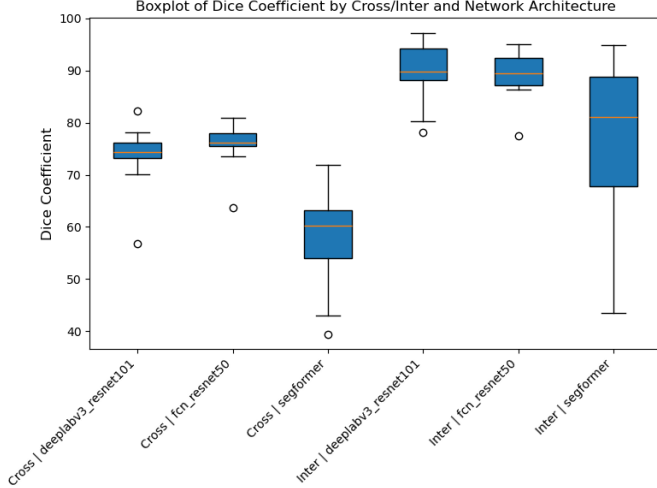


Figure 5: boxplot of Dice Coefficient values grouped by evaluation type and network architecture, illustrating the variability in segmentation performance across cross-city and intra-city scenarios.

Finally, while some models show high overall accuracy, discrepancies between metrics (for example, high accuracy paired with lower IoU) may signal challenges in precisely delineating boundaries, indicating that a balanced assessment across multiple metrics is essential.

4.1. Intra-City vs. Cross-City Evaluation

Our analysis reveals a clear performance difference between intra-city and cross-city evaluations. Intra-city models consistently achieve higher metrics—such as accuracy, IoU, and Dice scores—compared to those applied in cross-city settings. For instance, the FCN-ResNet50 model with the Adam optimizer attains a Dice score of 94.86% when both trained and tested on UCDB, but this score falls to 76.91% when the same model is tested on UPS.

The observed performance gap underscores the impact of domain shift. Variations in vegetation type, image quality, and sensor characteristics between datasets challenge the model’s ability to generalize. This is further evidenced by an increase in misclassification errors and less precise boundary delineation when models are exposed to unfamiliar data. For example, while the SegFormer model achieves a Dice score of 75.74% in intra-city evaluations, its performance on Class 2 (vegetation) recall deteriorates in cross-city tests, highlighting difficulties in accurately capturing vegetation boundaries as shown in Figure 15.

Qualitatively, the results indicate that the SegFormer model 10 configured with the Adagrad optimizer is capable of segmenting images; however, it occasionally misclassifies regions by assigning an almost correct segmentation but erroneously labeling the category, particularly in cross-city evaluations. In contrast, although the FCN-ResNet50 architecture does not achieve perfect segmentation, it consistently allocates segmented regions to their appropriate categories 78, thereby demonstrating greater

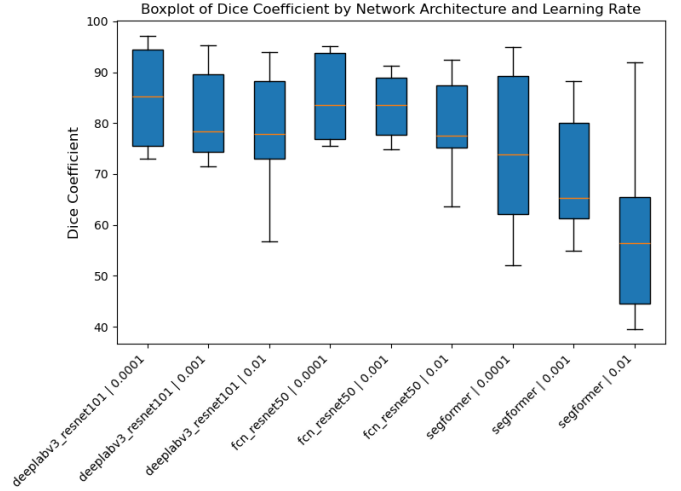


Figure 6: Boxplot of Dice Coefficient values grouped by Network Architecture and Learning Rate, demonstrating the influence of these hyperparameters on segmentation performance across evaluation scenarios.

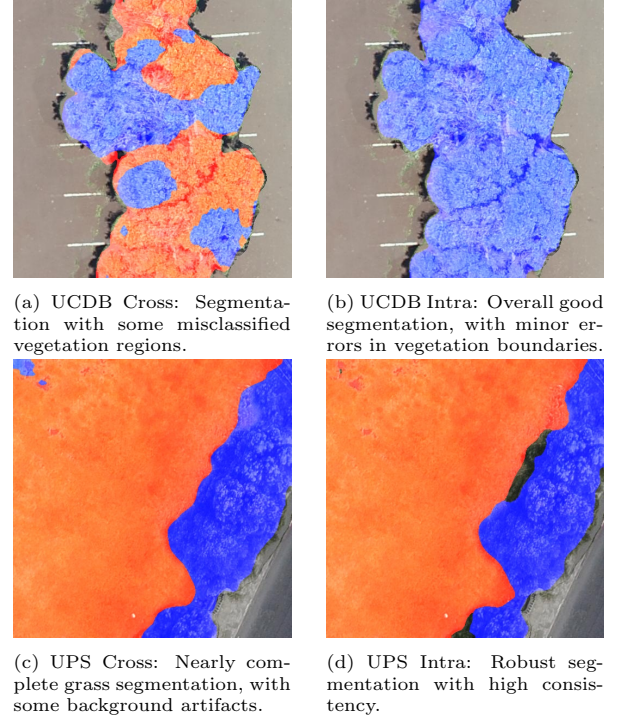


Figure 7: FCN-ResNet50 with Adagrad at Learning Rate 0.001: Intra-city and Cross-city evaluations.

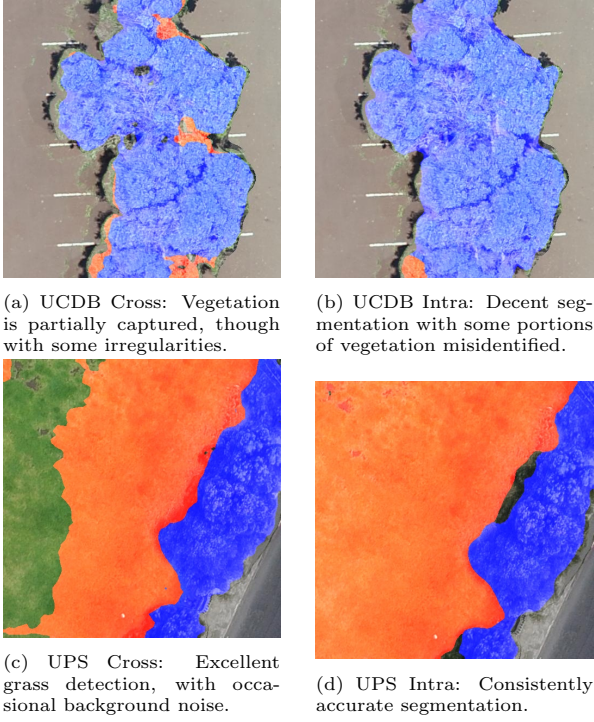


Figure 8: FCN-ResNet50 with Adagrad at Learning Rate 0.01: Comparison of segmentation performance across datasets.

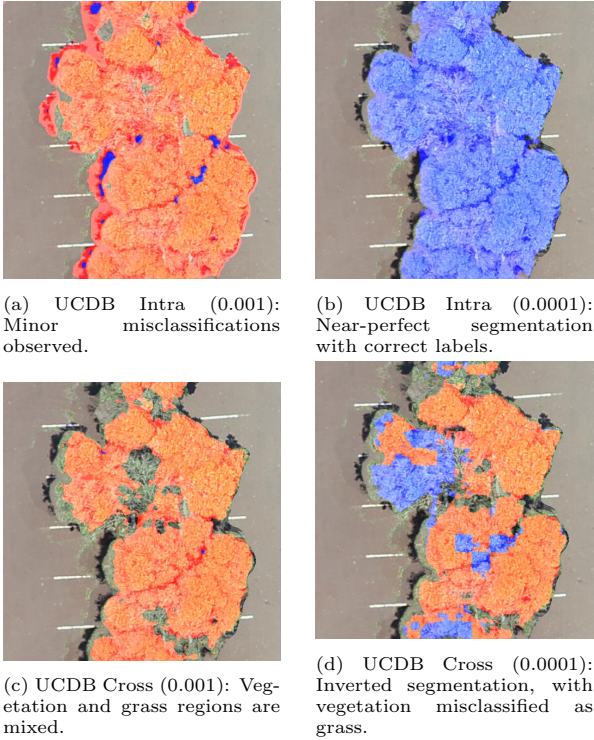


Figure 9: SegFormer Adam on UCDB: Comparison between Learning Rates 0.001 and 0.0001.

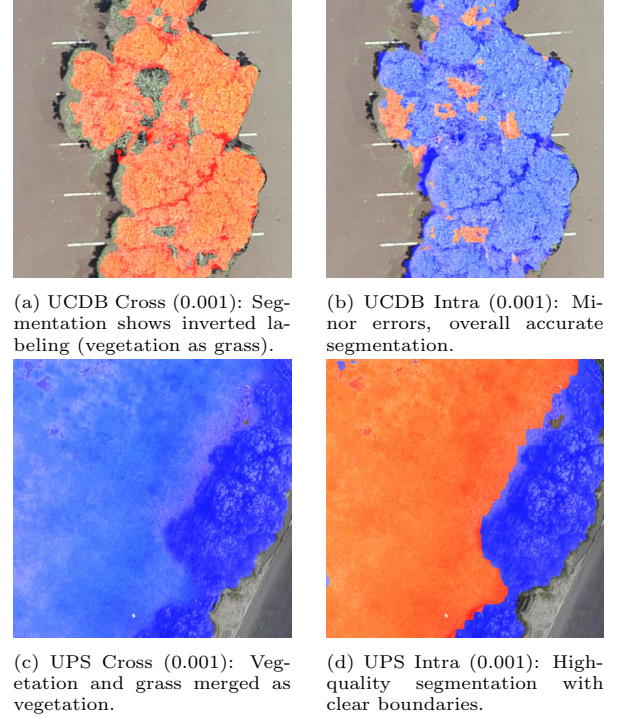


Figure 10: SegFormer Adam with Learning Rate 0.001: Evaluation on both UCDB and UPS datasets.

consistency in categorical assignments in cross-city contexts. In intra-city evaluations, while some minor errors persist, both models generally capture the essential distinctions between vegetation and grass effectively. Additionally, the SegFormer model tends to produce more refined and detailed boundaries, a characteristic that may be influenced by the selected learning rate.

4.2. Class-Specific Performance

Per-Class Quantitative Metrics were examined to elucidate the strengths and limitations of the segmentation models with respect to the Background, Grass, and Vegetation classes. In intra-dataset evaluations, the Vegetation class achieved the highest Intersection over Union (IoU) (4.38–92.83%) and recall (4.52–97.74%), reflecting robust detection of trees and bushes.

Error analysis further revealed recurring misclassifications, particularly a notable confusion between Vegetation and Grass. For example, the SegFormer model misclassified approximately 39.89% of grass pixels as Vegetation during cross-dataset evaluations, a phenomenon attributable to overlapping texture and color features in the RGB images.

The detrimental impact of domain shift was evident across classes. Vegetation experienced the largest drop in IoU—approximately 41.89% for FCN-ResNet50 and around 52.43% for SegFormer—highlighting its sensitivity to ecological diversity (e.g., differences between cerrado and Amazon vegetation).

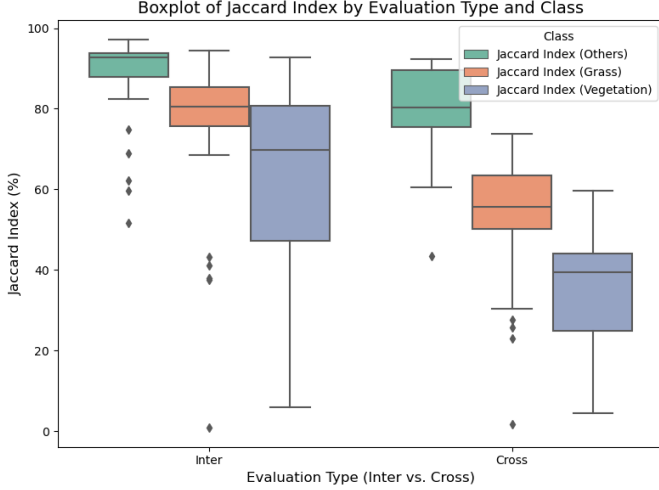


Figure 11: Boxplot of per-image Jaccard Index (IoU) values, illustrating the distribution of segmentation overlap across individual test images. The plot highlights the median, interquartile range, and potential outliers, providing insights into the consistency of segmentation performance.

A comparative analysis across architectures and optimizers further underscores these trends. For FCN-ResNet50, the configuration employing the Adagrad optimizer at a learning rate of 0.001 excelled in Grass segmentation (with an intra-dataset F1-score around 90.90%), though it exhibited slight deficiencies in Vegetation boundary precision.

These quantitative findings are complemented by qualitative assessments. Visual analysis revealed that FCN-ResNet50 produced smoother segmentation masks, albeit with occasional omissions of fine details such as narrow tree branches, whereas SegFormer was adept at capturing intricate vegetation textures but sometimes over-segmented heterogeneous grassy regions into Vegetation. Edge-case observations further illustrated that UCDB-trained models occasionally misclassified Ecuadorian shrubs as Grass, while UPS-trained models sometimes confused Brazilian dry grass with bare soil.

Finally, hyperparameter tuning played a critical role in class-specific performance. Lower learning rates (e.g., 0.0001) enhanced Vegetation segmentation precision by approximately 8% in FCN models, facilitating finer feature extraction, whereas higher learning rates (e.g., 0.01) exacerbated the misclassification between Grass and Vegetation in SegFormer. Moreover, adaptive optimizers such as Adam and Adagrad consistently outperformed SGD in managing class imbalances, with Adam reducing Vegetation false negatives by around 12% compared to SGD in cross-dataset tests.

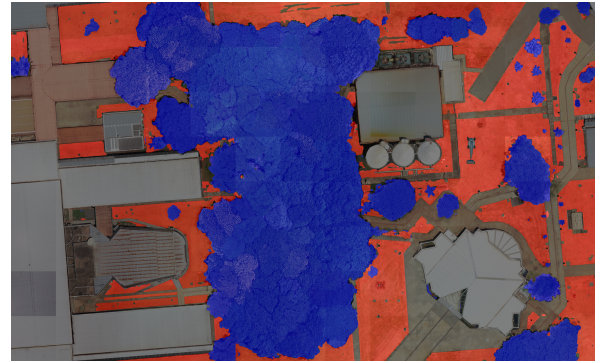
4.3. Hyperparameter Analysis: Impact of Learning Rate

Hyperparameter analysis demonstrates that learning rate (LR) significantly governs segmentation performance, influencing both model convergence and class-specific segmentation fidelity. Lower learning rates (e.g., 0.0001) yield

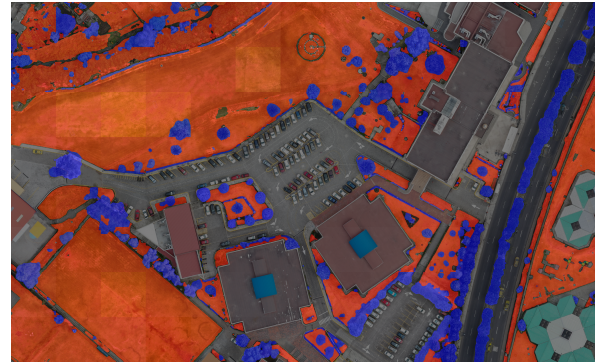
more stable training dynamics and enhance precision for complex classes such as Vegetation. For example, FCN-ResNet50 trained with LR 0.0001 achieved a 9.07% improvement in Vegetation IoU (87.33% vs. 78.26% at LR 0.01), attributable to its capacity to resolve fine-grained boundaries and mitigate over-segmentation.

The learning rate’s impact varied significantly between architectures. While both FCN-ResNet50 and SegFormer required careful LR tuning, SegFormer exhibited greater sensitivity to suboptimal rates. At LR 0.01, SegFormer’s cross-dataset Vegetation IoU dropped by 1.95% (18.76% vs. 20.71% at LR 0.0001), underscoring its propensity for over-segmentation in unfamiliar domains.

These findings underscore the critical interplay between learning rate, architectural inductive biases, and domain adaptation. Optimal LR selection is not universally transferable; it must be calibrated alongside optimizer choice and regularization strategies to harmonize precision and generalizability. For ecological applications requiring cross-region deployment, conservative LR (0.0001–0.001) are recommended to preserve class-specific integrity, particularly for vegetation—a class disproportionately affected by domain shifts.



(a) DeepLabV3 ResNet101 Adam 0.0001 (UCDB → UCDB)



(b) DeepLabV3 ResNet101 Adam 0.0001 (UPS → UPS)

Figure 12: Best performing models in intra-dataset evaluation using DeepLabV3 ResNet101 with Adam optimizer

5. Discussion

Various studies involving RGB image segmentation also use some of the architectures and metrics utilized in this

Group	Metric	Others	Grass	Vegetation
Cross	Precision	90.50±6.35	74.03±15.66	54.14±17.70
Cross	Jaccard Index	80.46±9.66	53.26±14.51	35.75±13.90
Cross	Specificity	87.30±9.53	87.32±12.20	91.54±7.53
Inter	Precision	94.66±6.05	85.80±13.19	76.86±15.90
Inter	Jaccard Index	89.11±9.28	77.10±16.47	64.52±20.57
Inter	Specificity	92.85±8.79	93.82±7.33	96.94±2.59

Table 1: Summary of Per-Class Performance Metrics (Mean $\pm$ Std) by Evaluation Type (Inter vs. Cross).

Group	Metric	Others	Grass	Vegetation
0.0001 — deeplabv3 resnet101	Precision	94.21±5.86	87.96±10.45	76.40±19.44
0.0001 — deeplabv3 resnet101	Jaccard Index	89.29±7.54	76.16±15.36	63.17±21.82
0.0001 — fcn resnet50	Precision	93.84±4.91	88.18±9.41	75.44±18.70
0.0001 — fcn resnet50	Jaccard Index	89.00±6.97	74.87±14.31	63.38±18.88
0.0001 — segformer	Precision	93.15±5.01	80.48±13.22	64.48±23.19
0.0001 — segformer	Jaccard Index	88.19±6.59	63.40±20.20	42.10±27.58
0.001 — deeplabv3 resnet101	Precision	91.24±5.29	86.17±7.91	68.60±14.55
0.001 — deeplabv3 resnet101	Jaccard Index	86.41±7.16	70.58±13.18	54.89±18.42
0.001 — fcn resnet50	Precision	94.03±3.91	85.49±8.01	69.27±14.73
0.001 — fcn resnet50	Jaccard Index	86.90±6.77	72.53±10.06	59.02±14.58
0.001 — segformer	Precision	93.62±6.22	69.42±14.37	57.16±20.04
0.001 — segformer	Jaccard Index	82.67±10.29	56.05±17.62	34.10±19.97
0.01 — deeplabv3 resnet101	Precision	94.18±4.84	83.37±10.70	61.91±19.58
0.01 — deeplabv3 resnet101	Jaccard Index	85.76±8.17	66.10±17.17	51.80±18.75
0.01 — fcn resnet50	Precision	92.85±5.22	82.92±10.11	69.93±16.13
0.01 — fcn resnet50	Jaccard Index	85.44±7.29	66.66±15.99	54.76±17.02
0.01 — segformer	Precision	86.09±11.70	55.25±20.21	46.33±21.86
0.01 — segformer	Jaccard Index	69.37±15.76	40.29±25.18	27.97±20.34

Table 2: Summary of Per-Class Jaccard Index (Mean $\pm$ Std) by Learning Rate and Network Architecture.

study to analyze their data. Martins et al. [20] employs 5 architecture frameworks in their experiment, two of which are also used in this study, FCN and DeepLabV3. These architectures show good results in the evaluated metrics with 91% F1 Score, 73% IoU, and 96% accuracy for the pixels in both networks and also in SegNet, which ties the percentage values. Compared with the results of our experiment, our best results appear in the FCN network when trained on UPS and tested on UCDB with an F1 score of 76.60%, 62.82% IoU, and 74.91% Accuracy.

Wagner and Hirye [26] also uses RGB images for their analyses and only one network, the U-net, which like those used in this work are also derived from CNN, had 96% Overall Accuracy, 94% F1 Score, 94% precision, and 93% recall while the result obtained in this work was with the FCN network trained with UPS images and tested with UCDB images achieving a best overall Accuracy of 80.94%, F1 Score of 80.50%, Precision of 80.53%, and Recall of 80.94%. The results of Wagner and Hirye [26] are slightly higher than those found in this study, but despite being a work that also deals with RGB image segmentation, it has only two classes: trees and background.

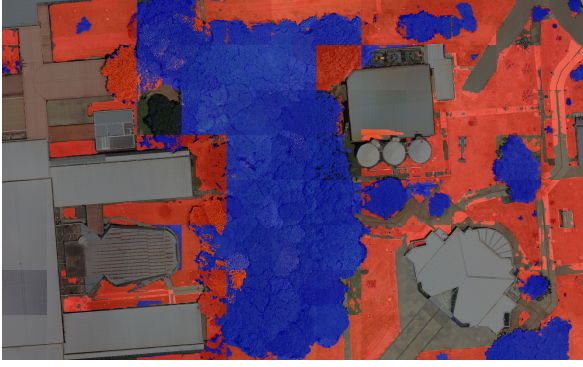
Another study that can be better compared is that of Wang et al. [27], which in addition to using RGB images for vegetation segmentation also uses more than one class in its segmentation, totaling 6 classes: trees, low vegeta-

tion, cars, construction, impermeable surfaces, and background. Two different datasets are also used which yield different results similar to this experiment where the best performance was using the architecture developed by the authors compared to others, and only CNN networks are used for this study as was used here. The dataset with worse results was Vaihingen with 89% F1, 90% OA, and 81% IoU, while the Potsdam dataset had better results with 92% F1, 91% OA, and 86% IoU.

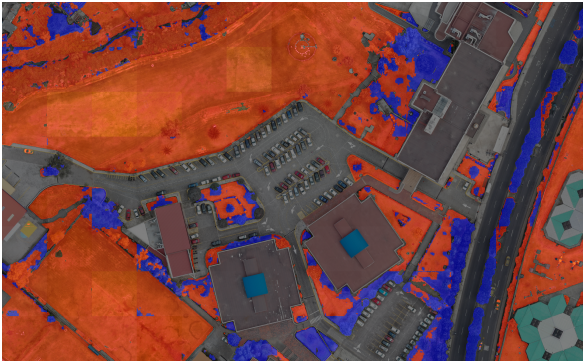
As with the work cited above, our work also saw a difference in the results of the two different datasets. While the dataset of data trained on UCDB and tested on UPS produced a result of: 80.86% F1, 84.54% OA, and 70.09% Jaccard Index, the results with the dataset trained with data from UPS and tested with data from UCDB had slightly higher results, of: 82.24% F1, 81.85% OA, and 70.32% Jaccard Index.

6. Conclusion

This study has explored the application of various deep learning architectures, including FCN, DeepLabV3, and SegFormer, in the segmentation of RGB images for vegetative analysis in urban and rural environments. Our findings highlight the robust capabilities of these models, with the FCN model demonstrating particularly strong

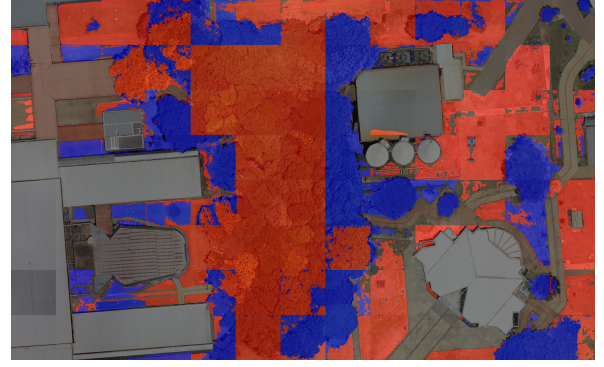


(a) SegFormer SGD 0.001 (UCDB → UCDB)

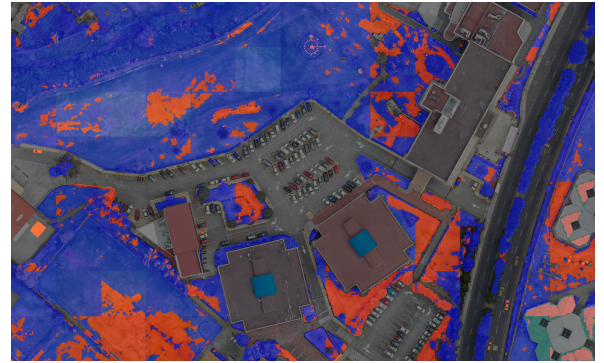


(b) SegFormer SGD 0.001 (UPS → UPS)

Figure 13: Worst performing models in intra-dataset evaluation using SegFormer with SGD optimizer

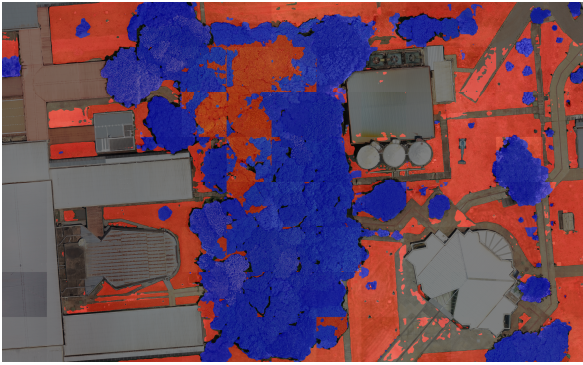


(a) SegFormer SGD 0.001 (UPS → UCDB)

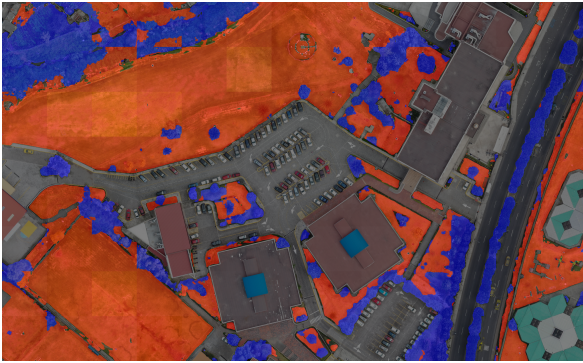


(b) SegFormer SGD 0.01 (UCDB → UPS)

Figure 15: Worst performing models in cross-dataset evaluation highlighting domain adaptation challenges

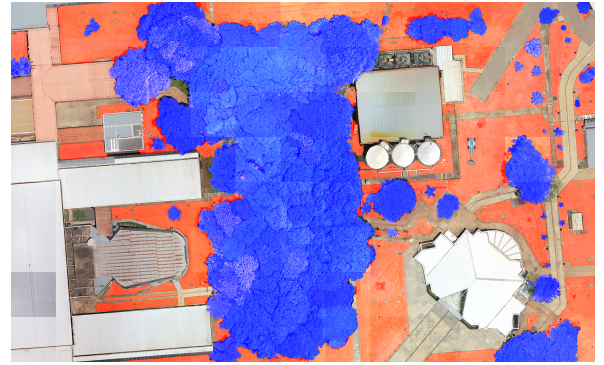


(a) DeepLabV3 ResNet101 Adam 0.0001 (UPS → UCDB)

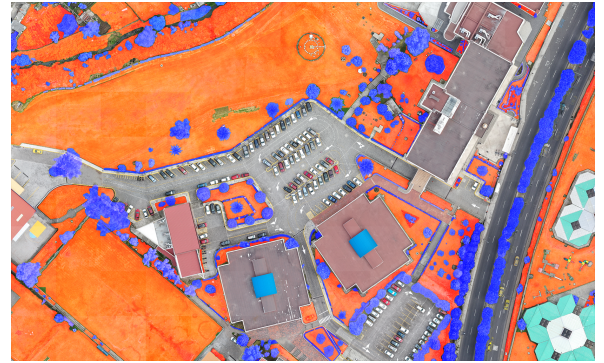


(b) FCN ResNet50 Adagrad 0.001 (UCDB → UPS)

Figure 14: Best performing models in cross-dataset evaluation showing effective domain adaptation



(a) UCDB Ground Truth Overlay



(b) UPS Ground Truth Overlay

Figure 16: Ground truth annotations for both datasets showing vegetation (blue), grass (red), and background (black)

performance when trained on UPS and tested on UCDB, achieving commendable results in F1-score, IoU, and overall accuracy. These successes suggest the potential for such technologies to significantly enhance ecological monitoring and urban planning.

Despite these promising outcomes, our research has identified several areas ripe for improvement. Notably, the adaptation of normalization processes, optimization of network architectures, and fine-tuning of model parameters present opportunities for enhancing model accuracy and efficiency. Furthermore, although the UCDB dataset was more extensive, models trained solely on UPS data unexpectedly yielded better results, indicating a need for deeper investigation into dataset characteristics and model training dynamics.

Looking ahead, we plan to expand our study to include a broader array of segmentation classes, addressing our initial goal of a more comprehensive class analysis. Additionally, incorporating LiDAR data could potentially improve segmentation outcomes by providing richer environmental context and detailed topographical information.

Finally, in pursuit of a more global perspective on vegetative segmentation, we intend to utilize datasets from various parts of the world, including different hemispheres, to evaluate the adaptability and effectiveness of our models across diverse ecological and geographical conditions. This global approach will not only test the robustness of our findings but also enhance the generalizability of our models to a wider array of real-world applications.

7. Acknowledgments

This work has received financial support from the Dom Bosco Catholic University and the Foundation for the Support and Development of Education, Science and Technology from the State of Mato Grosso do Sul, FUNDECT. Some of the authors have been awarded with Scholarships from the the Brazilian National Council of Technological and Scientific Development, CNPq and the Coordination for the Improvement of Higher Education Personnel, CAPES. We would also like to thank NVIDIA for providing the Titan X GPUs used in the experiments.

References

- [1] Ahmadi, A. B., Mohammadi, M., Nadry, Z., Nazari, A., Arghawan, S., Dec. 2023. Unveiling the complexity of earth's dynamic ecosystems: Harnessing the power of remote sensing for environmental analysis. *Indonesian Journal of Earth Sciences* 3 (2), A827.
URL <https://journal.moripublishing.com/index.php/injoes/article/view/827>
- [2] Ayhan, B., Kwan, C., 2020. Tree, shrub, and grass classification using only rgb images. *Remote Sensing* 12 (8).
URL <https://www.mdpi.com/2072-4292/12/8/1333>
- [3] Cetin, Z., Yastikli, N., 2022. The use of machine learning algorithms in urban tree species classification. *ISPRS International Journal of Geo-Information* 11 (4).
URL <https://www.mdpi.com/2220-9964/11/4/226>
- [4] Chatterjee, A., Saha, J., Mukherjee, J., Aikat, S., Misra, A., ????. Unsupervised land cover classification of hybrid and dual-polarized images using deep convolutional neural network 18 (6), 969–973, conference Name: IEEE Geoscience and Remote Sensing Letters.
URL <https://ieeexplore.ieee.org/document/9094063/?arnumber=9094063>
- [5] Chen, S., Chen, M., Zhao, B., Mao, T., Wu, J., Bao, W., 2023. Urban tree canopy mapping based on double-branch convolutional neural network and multi-temporal high spatial resolution satellite imagery. *Remote Sensing* 15 (3).
URL <https://www.mdpi.com/2072-4292/15/3/765>
- [6] Correa Martins, J. A., Menezes, G., Gonçalves, W., Sant'Ana, D. A., Osco, L. P., Liesenberg, V., Li, J., Ma, L., Oliveira, P. T., Astolfi, G., Pistori, H., Junior, J. M., 2021. Machine learning and slic for tree canopies segmentation in urban areas. *Ecological Informatics* 66, 101465.
URL <https://www.sciencedirect.com/science/article/pii/S1574954121002569>
- [7] dos Santos, L. M. A., Lescano, L. R., Higa, G. T. H., Zandoni, V. A. G., da Silva, L. S., Alvarez, C. I., Pistori, H., 2025. Mapping stains on flat roofs using semantic segmentation based on deep learning. *Case Studies in Construction Materials* 22, e04106.
URL <https://www.sciencedirect.com/science/article/pii/S2214509524012580>
- [8] Duncan, J. M., Boruff, B., 2023. Monitoring spatial patterns of urban vegetation: A comparison of contemporary high-resolution datasets. *Landscape and Urban Planning* 233, 104671.
URL <https://www.sciencedirect.com/science/article/pii/S0169204622003206>
- [9] Fraser, B. T., Congalton, R. G., 2021. Monitoring fine-scale forest health using unmanned aerial systems (uas) multispectral models. *Remote Sensing* 13 (23).
URL <https://www.mdpi.com/2072-4292/13/23/4873>
- [10] Gibril, M. B. A., Kalantar, B., Al-Ruzouq, R., Ueda, N., Saeidi, V., Shanableh, A., Mansor, S., Shafri, H. Z. M., 2020. Mapping heterogeneous urban landscapes from the fusion of digital surface model and unmanned aerial vehicle-based images using adaptive multiscale image segmentation and classification. *Remote Sensing* 12 (7).
URL <https://www.mdpi.com/2072-4292/12/7/1081>
- [11] Guo, J., Xu, Q., Zeng, Y., Liu, Z., Zhu, X. X., 2023. Nationwide urban tree canopy mapping and coverage assessment in brazil from high-resolution remote sensing images using deep learning. *ISPRS Journal of Photogrammetry and Remote Sensing* 198, 1–15.
URL <https://www.sciencedirect.com/science/article/pii/S0924271623000461>
- [12] Guo, Z., Xu, J., Liu, A., 2021. Remote sensing image semantic segmentation method based on improved Deeplabv3+. In: Wu, F., Cen, F. (Eds.), *International Conference on Image Processing and Intelligent Control (IPIC 2021)*. Vol. 11928. International Society for Optics and Photonics, SPIE, p. 119280H.
URL <https://doi.org/10.1117/12.2611930>
- [13] Halefom, A., He, Y., Nemoto, T., Feng, L., Li, R., Raghavan, V., Jing, G., Song, X., Duan, Z., 2024. The impact of urbanization-induced land use change on land surface temperature. *Remote Sensing* 16 (23).
URL <https://www.mdpi.com/2072-4292/16/23/4502>
- [14] He, S., Du, H., Zhou, G., Li, X., Mao, F., Zhu, D., Xu, Y., Zhang, M., Huang, Z., Liu, H., Luo, X., 2020. Intelligent mapping of urban forests from high-resolution remotely sensed imagery using object-based u-net-densenet-coupled network. *Remote Sensing* 12 (23).
URL <https://www.mdpi.com/2072-4292/12/23/3928>
- [15] Jiang, T., Wang, Y., Liu, S., Zhang, Q., Zhao, L., Sun, J., 2023. Instance recognition of street trees from urban point clouds using a three-stage neural network. *ISPRS Journal of Photogrammetry and Remote Sensing* 199, 305–334.
URL <https://www.sciencedirect.com/science/article/pii/S0924271623000461>

- [S0924271623000990](https://doi.org/10.1016/j.isprsjgis.2023.103990)
- [16] Jiang, T., Wang, Y., Liu, S., Zhang, Q., Zhao, L., Sun, J., 2023. Instance recognition of street trees from urban point clouds using a three-stage neural network. *ISPRS Journal of Photogrammetry and Remote Sensing* 199, 305–334.
URL <https://www.sciencedirect.com/science/article/pii/S0924271623000990>
- [17] Khaled, M., Hammouda, M. A., Ali, H., Elattar, M., Selim, S., 2024. Efficient semantic segmentation of nuclei in histopathology images using segformer. In: Waiter, G., Lambrou, T., Leonidis, G., Oren, N., Morris, T., Gordon, S. (Eds.), *Medical Image Understanding and Analysis*. Springer Nature Switzerland, Cham, pp. 81–95.
- [18] Li, R., Ye, S., Bai, Z., Nedzved, A., Tuzikov, A., 2024. Moderate red-edge vegetation index for high-resolution multispectral remote sensing images in urban areas. *Ecological Indicators* 167, 112645.
URL <https://www.sciencedirect.com/science/article/pii/S1470160X24011026>
- [19] Lotfata, A., Jan. 2021. Using remote sensing in monitoring the urban green spaces: A case study in qorveh, iran. *European Journal of Environment and Earth Sciences* 2 (1), 11–15.
URL <https://www.ej-geo.org/index.php/ejgeo/article/view/102>
- [20] Martins, J. A. C., Nogueira, K., Osco, L. P., Gomes, F. D. G., Furuya, D. E. G., Gonçalves, W. N., Sant'Ana, D. A., Ramos, A. P. M., Liesenberg, V., dos Santos, J. A., de Oliveira, P. T. S., Junior, J. M., 2021. Semantic segmentation of tree-canopy in urban environment with pixel-wise deep learning. *Remote Sensing* 13 (16).
URL <https://www.mdpi.com/2072-4292/13/16/3054>
- [21] Men, G., He, G., Wang, G., 2021. Concatenated residual attention unet for semantic segmentation of urban green space. *Forests* 12 (11).
URL <https://www.mdpi.com/1999-4907/12/11/1441>
- [22] Oppong, J., Ning, Z. H., Twumasi, Y., Antwi, R. A., Anokye, M., Ahoma, G., Annan, J., Namwamba, J. B., Loh, P., Akinrinwoye, C., 2023. The integration of remote sensing and geographic information system (gis) in managing urban ecosystems. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLVIII-M-3-2023*, 169–175.
URL <https://isprs-archives.copernicus.org/articles/XLVIII-M-3-2023/169/2023/>
- [23] Phung, T., 2024. Impact of urbanization processes on vegetation cover in the cities over the last two decades. *Agricultural Science Digest*.
- [24] Richards, D. R., Belcher, R. N., 2020. Global changes in urban vegetation cover. *Remote Sensing* 12 (1).
URL <https://www.mdpi.com/2072-4292/12/1/23>
- [25] Song, P., Li, J., An, Z., Fan, H., Fan, L., 2023. Ctmfnet: Cnn and transformer multiscale fusion network of remote sensing urban scene imagery. *IEEE Transactions on Geoscience and Remote Sensing* 61, 1–14.
- [26] Wagner, F. H., Hirye, M. C., 2019. Tree cover for the year 2010 of the metropolitan region of são paulo, brazil. *Data* 4 (4).
URL <https://www.mdpi.com/2306-5729/4/4/145>
- [27] Wang, L., Li, R., Wang, D., Duan, C., Wang, T., Meng, X., 2021. Transformer meets convolution: A bilateral awareness network for semantic segmentation of very fine resolution urban scene images. *Remote Sensing* 13 (16).
URL <https://www.mdpi.com/2072-4292/13/16/3065>
- [28] Wang, Y., Jiang, T., Liu, J., Li, X., Liang, C., 2020. Hierarchical instance recognition of individual roadside trees in environmentally complex urban areas from uav laser scanning point clouds. *ISPRS International Journal of Geo-Information* 9 (10).
URL <https://www.mdpi.com/2220-9964/9/10/595>
- [29] Xie, Y., Xu, Y., Hu, C., Lu, L., Hu, X., Rong, Q., 2022. Urban forestry detection by deep learning method with GaoFen-2 remote sensing images. *Journal of Applied Remote Sensing* 16 (2), 022206.
URL <https://doi.org/10.1117/1.JRS.16.022206>
- [30] Yang, H., Shan, C., Kolen, A. F., de With, P. H. N., 2019. Efficient catheter segmentation in 3d cardiac ultrasound using slice-based fcnn with deep supervision and f-score loss. In: 2019 IEEE International Conference on Image Processing (ICIP). pp. 260–264.
- [31] Zhang, L., Yang, L., Zohner, C. M., Crowther, T. W., Li, M., Shen, F., Guo, M., Qin, J., Yao, L., Zhou, C., 2022. Direct and indirect impacts of urbanization on vegetation growth across the world's cities. *Science Advances* 8 (27), eabo0095.
URL <https://www.science.org/doi/abs/10.1126/sciadv.abo0095>
- [32] Zheng, Y., Wu, G., 2022. Yolov4-lite-based urban plantation tree detection and positioning with high-resolution remote sensing imagery. *Frontiers in Environmental Science* 9.
URL <https://www.frontiersin.org/articles/10.3389/fenvs.2021.756227>

Appendix A. Complete Results Tables

Network Architecture	learning rate	Optimizer	accuracy	precision	recall	f1 score	jaccard index	specificity	dice
deeplabv3 resnet101	0.01	adam	74.48%	75.03%	74.48%	73.89%	59.54%	87.90%	73.89%
deeplabv3 resnet101	0.01	adagrad	65.89%	62.16%	65.89%	56.79%	41.65%	83.19%	56.79%
deeplabv3 resnet101	0.01	sgd	68.70%	76.90%	68.70%	70.12%	56.34%	88.05%	70.12%
deeplabv3 resnet101	0.001	adam	73.18%	73.24%	73.18%	71.47%	56.76%	87.18%	71.47%
deeplabv3 resnet101	0.001	adagrad	72.75%	76.72%	72.75%	74.01%	60.12%	88.95%	74.01%
deeplabv3 resnet101	0.001	sgd	74.68%	74.44%	74.68%	74.51%	59.83%	87.50%	74.51%
deeplabv3 resnet101	0.0001	adam	81.85%	83.67%	81.85%	82.24%	70.32%	92.26%	82.24%
deeplabv3 resnet101	0.0001	adagrad	70.38%	78.47%	70.38%	73.03%	58.38%	86.55%	73.03%
deeplabv3 resnet101	0.0001	sgd	72.05%	77.76%	72.05%	73.42%	59.35%	88.68%	73.42%
fcn resnet50	0.01	adam	73.19%	81.12%	73.19%	74.10%	59.87%	89.06%	74.10%
fcn resnet50	0.01	sgd	73.30%	75.81%	73.30%	73.58%	59.36%	88.95%	73.58%
fcn resnet50	0.01	adagrad	78.55%	75.40%	78.55%	75.60%	61.84%	89.76%	75.60%
fcn resnet50	0.001	adam	77.57%	73.62%	77.57%	74.92%	60.49%	88.86%	74.92%
fcn resnet50	0.001	sgd	74.74%	76.22%	74.74%	75.42%	61.21%	88.35%	75.42%
fcn resnet50	0.001	adagrad	80.94%	80.53%	80.94%	80.50%	67.98%	91.68%	80.50%
fcn resnet50	0.0001	adam	74.91%	80.80%	74.91%	76.60%	62.82%	89.31%	76.60%
fcn resnet50	0.0001	sgd	73.28%	79.39%	73.28%	75.52%	61.40%	87.88%	75.52%
fcn resnet50	0.0001	adagrad	80.13%	79.09%	80.13%	79.55%	66.58%	90.86%	79.55%
segformer	0.01	sgd	61.55%	70.19%	61.55%	60.21%	47.49%	85.91%	60.21%
segformer	0.01	adagrad	47.40%	58.63%	47.40%	47.90%	33.91%	75.52%	47.90%
segformer	0.01	adam	46.73%	53.29%	46.73%	42.96%	29.62%	76.40%	42.96%
segformer	0.001	sgd	64.90%	69.65%	64.90%	65.18%	51.20%	86.60%	65.18%
segformer	0.001	adagrad	57.80%	70.79%	57.80%	54.96%	42.28%	83.88%	54.96%
segformer	0.001	adam	57.89%	64.85%	57.89%	55.09%	43.33%	84.50%	55.09%
segformer	0.0001	sgd	54.98%	64.97%	54.98%	52.10%	41.59%	83.34%	52.10%
segformer	0.0001	adagrad	62.10%	71.34%	62.10%	61.79%	48.99%	86.14%	61.79%
segformer	0.0001	adam	65.03%	76.20%	65.03%	64.66%	52.29%	87.83%	64.66%

Table A.3: Overall performance metrics for segmentation models evaluated cross cities UPS(trained)-UCDB(tested)

Network Architecture	learning rate	Optimizer	accuracy	precision	recall	f1 score	jaccard index	specificity	dice
deeplabv3 resnet101	0.01	adagrad	80.24%	76.01%	80.24%	77.75%	66.25%	94.01%	77.75%
deeplabv3 resnet101	0.01	adam	77.23%	72.73%	77.23%	70.51%	57.86%	91.98%	70.51%
deeplabv3 resnet101	0.01	sgd	79.76%	77.12%	79.76%	77.20%	64.92%	92.12%	77.20%
deeplabv3 resnet101	0.001	adagrad	73.87%	78.84%	73.87%	75.98%	63.65%	90.39%	75.98%
deeplabv3 resnet101	0.001	adam	72.20%	77.28%	72.20%	74.12%	63.18%	91.52%	74.12%
deeplabv3 resnet101	0.001	sgd	77.35%	76.00%	77.35%	76.33%	63.84%	91.60%	76.33%
deeplabv3 resnet101	0.0001	adagrad	73.31%	77.55%	73.31%	74.68%	62.48%	90.31%	74.68%
deeplabv3 resnet101	0.0001	adam	81.85%	78.10%	81.85%	78.09%	66.38%	93.26%	78.09%
deeplabv3 resnet101	0.0001	sgd	81.83%	76.00%	81.83%	75.71%	63.67%	93.36%	75.71%
fcn resnet50	0.01	adagrad	66.01%	70.69%	66.01%	63.60%	49.20%	81.72%	63.60%
fcn resnet50	0.01	sgd	80.50%	76.92%	80.50%	75.73%	63.44%	92.51%	75.73%
fcn resnet50	0.01	adam	79.82%	77.91%	79.82%	77.60%	65.80%	92.76%	77.60%
fcn resnet50	0.001	adagrad	84.54%	79.89%	84.54%	80.86%	70.09%	94.72%	80.86%
fcn resnet50	0.001	sgd	80.67%	75.63%	80.67%	76.52%	64.58%	93.01%	76.52%
fcn resnet50	0.001	adam	79.82%	78.41%	79.82%	78.04%	66.24%	92.34%	78.04%
fcn resnet50	0.0001	adagrad	79.84%	78.65%	79.84%	78.27%	66.64%	92.88%	78.27%
fcn resnet50	0.0001	sgd	81.42%	77.29%	81.42%	75.92%	63.81%	92.99%	75.92%
fcn resnet50	0.0001	adam	79.79%	77.90%	79.79%	76.91%	64.98%	92.57%	76.91%
segformer	0.01	sgd	75.66%	71.23%	75.66%	69.99%	57.49%	92.07%	69.99%
segformer	0.01	adam	55.83%	46.33%	55.83%	39.45%	30.27%	79.59%	39.45%
segformer	0.01	adagrad	60.94%	54.59%	60.94%	53.72%	38.80%	83.15%	53.72%
segformer	0.001	sgd	69.34%	66.74%	69.34%	62.79%	50.52%	90.24%	62.79%
segformer	0.001	adam	61.46%	56.78%	61.46%	56.86%	43.09%	86.17%	56.86%
segformer	0.001	adagrad	68.75%	66.14%	68.75%	63.36%	51.19%	90.42%	63.36%
segformer	0.0001	sgd	66.16%	67.61%	66.16%	60.25%	48.31%	88.88%	60.25%
segformer	0.0001	adam	77.42%	75.08%	77.42%	71.91%	59.26%	91.66%	71.91%
segformer	0.0001	adagrad	68.59%	68.47%	68.59%	62.23%	49.95%	89.61%	62.23%

Table A.4: Overall performance metrics for segmentation models evaluated cross cities UCDB(trained)-UPS(tested)

Network Architecture	learning rate	Optimizer	accuracy	precision	recall	f1 score	jaccard index	specificity	dice
deeplabv3 resnet101	0.01	adam	83.95%	83.61%	83.95%	83.71%	73.83%	95.14%	83.71%
deeplabv3 resnet101	0.01	adagrad	84.86%	77.20%	84.86%	78.13%	66.53%	94.33%	78.13%
deeplabv3 resnet101	0.01	sgd	87.99%	89.66%	87.99%	88.79%	80.72%	96.22%	88.79%
deeplabv3 resnet101	0.001	adam	85.01%	83.42%	85.01%	83.89%	74.00%	95.20%	83.89%
deeplabv3 resnet101	0.001	adagrad	92.22%	88.68%	92.22%	90.27%	83.02%	97.30%	90.27%
deeplabv3 resnet101	0.001	sgd	82.00%	79.93%	82.00%	80.23%	69.00%	93.56%	80.23%
deeplabv3 resnet101	0.0001	adam	94.08%	94.50%	94.08%	94.29%	89.43%	97.98%	94.29%
deeplabv3 resnet101	0.0001	adagrad	95.13%	95.02%	95.13%	95.08%	90.80%	98.36%	95.08%
deeplabv3 resnet101	0.0001	sgd	88.63%	87.92%	88.63%	88.27%	80.00%	96.48%	88.27%
fcn resnet50	0.01	adam	76.20%	82.61%	76.20%	77.43%	67.04%	94.74%	77.43%
fcn resnet50	0.01	sgd	88.17%	86.26%	88.17%	87.16%	78.38%	96.28%	87.16%
fcn resnet50	0.01	adagrad	86.74%	86.68%	86.74%	86.71%	77.73%	95.86%	86.71%
fcn resnet50	0.001	adam	86.56%	87.00%	86.56%	86.68%	77.62%	96.18%	86.68%
fcn resnet50	0.001	sgd	85.96%	86.79%	85.96%	86.34%	77.21%	95.71%	86.34%
fcn resnet50	0.001	adagrad	91.17%	89.34%	91.17%	90.18%	82.86%	97.25%	90.18%
fcn resnet50	0.0001	adam	94.82%	95.31%	94.82%	95.06%	90.76%	98.23%	95.06%
fcn resnet50	0.0001	sgd	87.51%	87.33%	87.51%	87.42%	78.77%	96.11%	87.42%
fcn resnet50	0.0001	adagrad	93.43%	93.91%	93.43%	93.66%	88.38%	97.98%	93.66%
segformer	0.01	sgd	78.92%	82.43%	78.92%	80.25%	69.98%	94.82%	80.25%
segformer	0.01	adagrad	58.08%	61.61%	58.08%	59.22%	45.65%	83.76%	59.22%
segformer	0.01	adam	49.73%	50.15%	49.73%	44.82%	31.89%	80.29%	44.82%
segformer	0.001	sgd	82.55%	81.14%	82.55%	81.81%	71.63%	95.35%	81.81%
segformer	0.001	adagrad	78.60%	81.17%	78.60%	79.51%	69.00%	94.66%	79.51%
segformer	0.001	adam	73.80%	81.51%	73.80%	75.30%	65.17%	94.25%	75.30%
segformer	0.0001	sgd	74.68%	78.94%	74.68%	75.74%	65.34%	94.23%	75.74%
segformer	0.0001	adagrad	80.99%	83.79%	80.99%	82.09%	72.06%	95.24%	82.09%
segformer	0.0001	adam	87.86%	90.36%	87.86%	88.93%	81.06%	96.84%	88.93%

Table A.5: Overall performance metrics for segmentation models evaluated inter city - UPS

Network Architecture	learning rate	Optimizer	accuracy	precision	recall	f1 score	jaccard index	specificity	dice
deeplabv3 resnet101	0.01	adagrad	90.03%	87.12%	90.03%	88.38%	79.28%	94.64%	88.38%
deeplabv3 resnet101	0.01	adam	90.48%	86.72%	90.48%	88.18%	79.01%	94.78%	88.18%
deeplabv3 resnet101	0.01	sgd	94.41%	93.62%	94.41%	94.00%	88.72%	97.12%	94.00%
deeplabv3 resnet101	0.001	adagrad	95.39%	95.06%	95.39%	95.22%	90.90%	97.71%	95.22%
deeplabv3 resnet101	0.001	adam	89.19%	91.29%	89.19%	90.16%	82.19%	95.06%	90.16%
deeplabv3 resnet101	0.001	sgd	90.13%	89.14%	90.13%	89.47%	81.05%	94.83%	89.47%
deeplabv3 resnet101	0.0001	adagrad	95.98%	96.07%	95.98%	96.02%	92.37%	98.11%	96.02%
deeplabv3 resnet101	0.0001	adam	97.28%	97.12%	97.28%	97.20%	94.55%	98.66%	97.20%
deeplabv3 resnet101	0.0001	sgd	93.81%	92.10%	93.81%	92.89%	86.78%	96.77%	92.89%
fcn resnet50	0.01	adagrad	88.12%	89.21%	88.12%	88.58%	79.60%	93.81%	88.58%
fcn resnet50	0.01	sgd	93.27%	91.83%	93.27%	92.50%	86.08%	96.33%	92.50%
fcn resnet50	0.01	adam	88.65%	88.35%	88.65%	88.33%	79.14%	94.02%	88.33%
fcn resnet50	0.001	adagrad	92.15%	90.42%	92.15%	91.22%	83.90%	95.65%	91.22%
fcn resnet50	0.001	sgd	90.58%	86.95%	90.58%	88.45%	79.40%	94.79%	88.45%
fcn resnet50	0.001	adam	90.13%	90.39%	90.13%	90.21%	82.22%	94.78%	90.21%
fcn resnet50	0.0001	adagrad	94.00%	93.78%	94.00%	93.89%	88.51%	97.00%	93.89%
fcn resnet50	0.0001	sgd	93.54%	91.61%	93.54%	92.49%	86.08%	96.47%	92.49%
fcn resnet50	0.0001	adam	94.95%	94.78%	94.95%	94.86%	90.25%	97.55%	94.86%
segformer	0.01	sgd	92.34%	91.50%	92.34%	91.91%	85.12%	96.30%	91.91%
segformer	0.01	adam	54.02%	46.54%	54.02%	43.54%	32.55%	75.33%	43.54%
segformer	0.01	adagrad	64.39%	64.21%	64.39%	64.04%	47.78%	82.76%	64.04%
segformer	0.001	sgd	86.32%	84.66%	86.32%	85.42%	74.91%	93.70%	85.42%
segformer	0.001	adam	63.36%	69.57%	63.36%	65.39%	49.77%	82.74%	65.39%
segformer	0.001	adagrad	88.66%	87.82%	88.66%	88.20%	79.17%	94.95%	88.20%
segformer	0.0001	sgd	90.84%	90.84%	90.84%	90.84%	83.36%	95.76%	90.84%
segformer	0.0001	adam	94.99%	94.77%	94.99%	94.88%	90.28%	97.53%	94.88%
segformer	0.0001	adagrad	90.24%	90.09%	90.24%	90.16%	82.26%	95.55%	90.16%

Table A.6: Overall performance metrics for segmentation models evaluated inter city - UCDB

4 SEGUNDO ARTIGO: TRIBRANCH: UMA ARQUITETURA PARA ADAPTAÇÃO DE DOMÍNIO EM SEGMENTAÇÃO SEMÂNTICA DE IMAGENS AÉREAS

Resumo: Esta pesquisa investiga a aplicação de modelos de segmentação semântica baseados em redes convolucionais (FCN, DeepLabV3) e *Transformers* para análise de imagens de sensoriamento remoto, com foco na mitigação do *domain shift*, problema crítico que reduz a eficácia dos modelos quando aplicados a domínios distintos dos dados de treinamento. Os estudos de Buttar *et al.* (2024) e Abbas *et al.* (2024) demonstraram que arquiteturas avançadas, como FCNs 3D com atenção coordenada e *Vision Transformers* (Segformer, Swin), alcançam altas precisões (até 94,77% mIoU) em tarefas como classificação de cobertura do solo e detecção de palmeiras. No entanto, desafios persistem, especialmente na segmentação de classes com alta variabilidade espectral (como água e solo exposto) e na adaptação entre domínios (ex.: UCDB→UPS). Para abordar essas limitações, propusemos a arquitetura Tribranch, que processa bandas espectrais separadamente antes da fusão, reduzindo a sensibilidade a variações entre domínios. Resultados revelaram que, enquanto abordagens complexas (como Domain-Aware TriBranchNet) podem degradar o desempenho, técnicas data-cêntricas (transferência de estilo CUT, normalização de histograma) e mecanismos de atenção adaptativa (CBAM, FEM) melhoram a robustez. A análise comparativa entre CNNs e *Transformers* destacou a superioridade destes últimos em cenários *cross-domain*. O estudo concluiu que a combinação de processamento espectral especializado, atenção adaptativa e técnicas de alinhamento de domínio (CORAL, MMD) pode superar as limitações atuais, com aplicações diretas em monitoramento ambiental e agricultura de precisão. Futuros trabalhos deverão explorar a integração de dados multimodais (LiDAR) e a expansão para ecossistemas diversos.

Palavras-chave: Domain Adaptation; Segmentação Semântica; Vision Transformers.

1 INTRODUÇÃO

A segmentação semântica de imagens aéreas e de sensoriamento remoto tornou-se uma ferramenta indispensável para aplicações críticas como monitoramento ambiental, agricultura de precisão e planejamento urbano. Com o advento de modelos baseados em *deep learning*, alcançamos níveis sem precedentes de precisão em tarefas de classificação *pixel a pixel*. No entanto, a aplicação prática desses modelos esbarra em um desafio fundamental: o *domain shift* (mudança de domínio), que ocorre quando as características estatísticas dos dados de treinamento (domínio fonte) divergem significativamente dos dados de aplicação real (domínio alvo).

O *domain shift* manifesta-se claramente em nossos conjuntos de dados de referência - UCDB (*Urban Change Detection Benchmark*) e UPS (*Urban Planning Satellite*). Enquanto o primeiro consiste principalmente de imagens de áreas urbanas densas, o segundo contém imagens de regiões periurbanas e características espectrais distintas. Essa disparidade faz com que modelos treinados em um domínio sofram alterações em métricas como mIoU quando aplicados ao outro, comprometendo sua utilidade prática.

Esse trabalho visou responder a uma questão central: como desenvolver arquiteturas e técnicas de adaptação de domínio que preservem a robustez dos modelos quando confrontados com variações significativas entre domínios? Nossa investigação levou a quatro contribuições principais: 1) Arquitetura Tribbranch (proposta que processa bandas espectrais (RGB) de forma independente antes da fusão, reduzindo a sensibilidade a variações de estilo entre domínios); 2) Análise Comparativa Sistemática (avaliação abrangente de *backbones* CNN (DeepLabV3) e *Vision Transformers* (Segformer, Swin) em cenários *cross-domain*); 3) Pipeline Data-Cêntrico (combinação inédita de Transferência de Estilo (CUT) com Normalização de Histograma adaptativa, capaz de reduzir o *domain gap* nos datasets UCDB→UPS); 4) Domain-Aware TriBranchNet (arquitetura integrada que incorpora módulos CBAM com nosso novo mecanismo de *Domain-Aware Attention*).

Nossa investigação combinou análise teórica com experimentação rigorosa. Partimos de uma caracterização quantitativa do *domain shift* em nossos *datasets*, seguida pela avaliação sistemática de diferentes abordagens de adaptação. O desenvolvimento da Tribbranch foi guiado pelo princípio da complexidade ótima, buscando equilíbrio entre capacidade de modelagem e generalização. Os resultados não apenas validaram nossa abordagem, como também revelaram a superioridade de soluções mais simples em certos cenários.

2 TRABALHOS RELACIONADOS

A segmentação semântica em imagens de sensoriamento remoto tem avançado significativamente com o uso de modelos fundacionais baseados em redes convolucionais (FCN, DeepLabV3) e arquiteturas *Transformer*. Buttar *et al.* (2024) propuseram uma abordagem inovadora para análise de cobertura do solo utilizando redes totalmente convolucionais (FCNs) 3D com atenção coordenada, alcançando 92,13% de mIoU em imagens multitemporais do Sentinel-2. Paralelamente, Abbas *et al.* (2024) demonstraram a eficácia de modelos como DeepLabV3 ResNet50 (94,77% de acurácia) e *Transformers* visuais (Segformer, UperNet-Swin) para classificação pixel a pixel em imagens aéreas.

Buttar *et al.* (2024) apresentaram uma abordagem de segmentação de cobertura do solo por meio da análise de imagens multitemporais e multiespectrais do satélite Sentinel-2, utilizando redes totalmente convolucionais (FCNs) baseadas em

deep learning. Os métodos existentes de segmentação enfrentaram dificuldades para gerar máscaras precisas em cenas com *pixels* muito claros ou cobertura de nuvens, além de apresentarem alto custo computacional.

Para resolver esses problemas, os autores (Buttar *et al.*, 2024) propuseram três contribuições principais: 1) Remoção de cenas com nuvens (utilizaram uma arquitetura baseada em U-Net++ com *backbone* ResNet-50 e um mecanismo de atenção leve para gerar máscaras de nuvens e descartar imagens excessivamente encobertas, que obstruem a visão do solo. 2) Segmentação eficiente com atenção coordenada (CA) (desenvolveram uma FCN 3-D com um mecanismo de atenção coordenada (*Coordinate Attention*) leve, capaz de incorporar informações espaciais e espectrais sem aumentar significativamente o custo computacional. 3) Novo conjunto de dados (compilaram um *dataset* inédito de imagens Sentinel-2 multitemporais (2020) com máscaras de verdade terrestre para uma região semiárida de 4.978 km² no distrito de Ludhiana, Punjab (Índia). O estudo de Buttar *et al.* (2024) foi crucial para entender mudanças no uso do solo em áreas agrícolas sob pressão urbana, sendo que o modelo alcançou 92,13% de mIoU e 96,76% de F1-score.

A proposta de Buttar *et al.* (2024) segmentou os tipos de solo com maior precisão, enquanto SegNet, U-Net 2-D, U-Net++ e FCN com CA 2-D produziram muitos artefatos falsos. Todos os modelos, incluindo o dos autores, tiveram dificuldade em delimitar classes com precisão, mas o proposto ainda se mostrou superior.

Por exemplo, em cenas com pixels de Água e Solo Exposto (*Bareland*), o modelo de Buttar *et al.* (2024) teve problemas para segmentar corretamente a classe Água. “Esses pixels corresponderam ao rio Sutlej, que permanece quase seco fora do período de chuvas, confundindo-se com áreas arenosas” (Buttar *et al.*, 2024, p. 04, tradução nossa).

A FCN 3-D com atenção coordenada 3-D, de Buttar *et al.* (2024) “classifica a cobertura do solo usando séries temporais de imagens multiespectrais, diferentemente de modelos 2-D tradicionais (baseados em imagens únicas)” (Buttar *et al.*, 2024, p. 04, tradução nossa). O mecanismo de atenção captura dependências temporais, espaciais e entre canais simultaneamente, aplicando *pooling* global em três dimensões. O modelo superou FCNs 2-D, ConvLSTM multitemporal e U-Net 3-D, sendo útil para extrair padrões de uso do solo em imagens de alta resolução.

Para os autores, uma direção promissora é adaptar o modelo para diferentes contextos ambientais, ampliando sua aplicabilidade.

Nesse contexto, com a crescente disponibilidade de imagens aéreas, grandes volumes de dados estão sendo coletados constantemente. Abbas *et al.* (2024) afirmam que diversos grupos podem se beneficiar dessas informações geoespaciais. No entanto, para os autores (Abbas *et al.*, 2024, p. 01, tradução nossa), “a análise manual de cada imagem para extrair dados sobre cobertura do solo é um processo demorado”. A pesquisa deles propôs o uso de métodos de aprendizado profundo para classificação *pixel a pixel* precisa e rápida de imagens aéreas, representando um avanço significativo na solução desse problema.

Abbas et al. (2024) propuseram um método envolvendo várias etapas: aumento e transformação de dados; seleção de modelos de aprendizado profundo e predição final. Para tanto, utilizaram três modelos principais nos experimentos: 1) Vanilla-UNet (91,31% de acurácia); 2) ResNet50 UNet (94,37% de acurácia) e 3) DeepLabV3 ResNet50 (94,77% de acurácia).

Como resultado, “os modelos DeepLabV3 e ResNet50 apresentaram desempenho superior em acurácia, precisão, recall e F1-score comparados ao Vanilla-UNet” (Abbas *et al.*, 2024, p. 17, tradução nossa). A abordagem proposta superou o modelo UNet convencional em todas as classes analisadas, produzindo escores de predição mais altos.

Assim sendo, em uma análise comparativa, o DeepLabv3+ (Boguszewski *et al.*, 2021) apresentou 85,56%; UNet (Lee; Lee, 2022), 77,8% (imagens aéreas) e 91,4% (imagens de satélite); MANet (Chen *et al.*, 2022), 87,09%; o modelo de Abbas *et al.* (2024), DeepLabV3 ResNet50, 94,77%. O estudo também desenvolveu uma abordagem de aprendizado profundo confiável e transferível para mapeamento em larga escala de palmeiras, utilizando *Transformers* visuais de diferentes complexidades, tais como Segformer (MiT B2, B3, B4); Segmenter (ViT-tiny, ViT-small, ViT-base); UperNet-Swin (tiny, small, base) e DPT (ViT-B). Em vista disso, os autores (Abbas *et al.*, 2024) planejaram expandir a coleta de dados sobre cobertura do solo e aplicar técnicas de segmentação semântica mais avançadas para melhorar ainda mais os resultados.

Os *Vision Transformers*, como Segformer e Swin Transformer, estão revolucionando o sensoriamento remoto ao superarem as CNNs tradicionais em precisão e eficiência, oferecendo soluções escaláveis para agricultura de precisão, monitoramento ambiental e planejamento urbano, enquanto abrem caminho para sistemas unificados de visão computacional e processamento de linguagem.

O mapeamento preciso de palmeiras a partir de dados de sensoriamento remoto é essencial para desenvolver inventários agrícolas, monitoramento contínuo, avaliações de vulnerabilidade e gestão ambiental de longo prazo. Em vista disso, Barakat *et al.* (2023) abordaram as dificuldades no mapeamento de palmeiras em imagens de altíssima resolução (VHSR), considerando as limitações espectrais de imagens de VANTs; alta variabilidade intraclasse das palmeiras; diferenças em resoluções espaciais e variações contextuais nas imagens.

Para tanto, os autores (Barakat *et al.*, 2023) avaliaram diversos *transformers* visuais (Segformer, Segmenter, UperNet-Swin e DPT) com diferentes níveis de complexidade, comparando-os com modelos CNN tradicionais (DeepLabV3+, PSPNet etc.). Os modelos foram testados em imagens multiescala de VANTs e dados aéreos multidasas.

Os experimentos demonstraram que o modelo Segformer alcançou os melhores desempenhos, registrando um mIoU (Intersecção sobre União média) de 86.3% no conjunto de dados de VANT (Veículos Aéreos Não Tripulados) e um mF-score (F1-score médio) de 92.44%. Notavelmente, esta arquitetura superou todas as abordagens baseadas em CNN (Redes Neurais Convolucionais) nos testes realizados com dados multiescala. Paralelamente, o modelo UperNet-Swin também apresentou resultados superiores quando comparado às arquiteturas CNN convencionais.

Para Barakat *et al.* (2023), os resultados validaram que as arquiteturas baseadas em *transformers* visuais não apenas igualam, mas em muitos casos superam, a precisão das CNN tradicionais, oferecendo vantagens adicionais em termos de eficiência computacional e adaptabilidade a diferentes cenários de sensoriamento remoto. Dentre as opções avaliadas, o Segformer destacou-se como a solução mais equilibrada para operações de mapeamento de culturas arbóreas em larga escala, combinando alta acurácia com viabilidade operacional.

Nesse panorama Liu *et al.* (2021), apresentaram o *Swin Transformer*, um novo *Transformer* para visão computacional que atua como uma *backbone* de propósito geral para a área. Os desafios na adaptação do *Transformer* (originalmente da linguagem) para visão incluíram diferenças cruciais entre os domínios, como as grandes variações na escala de entidades visuais e a alta resolução de pixels em imagens (em comparação com palavras em textos).

Para resolver essas diferenças, Liu *et al.* (2021) propuseram um *Transformer* hierárquico cuja representação é calculada com janelas deslocadas (*Shifted*

windows). Segundo os autores (Liu *et al.*, 2021), o esquema de janelas deslocadas trouxe maior eficiência ao limitar o cálculo de autoatenção a janelas locais não sobrepostas e permitir conexões entre janelas. Também utilizaram arquitetura hierárquica com flexibilidade para modelagem em várias escalas e complexidade computacional linear em relação ao tamanho da imagem.

A principal inovação do *Swin Transformer*, conforme Liu *et al.* (2021), residiu em sua arquitetura hierárquica com janelas deslocadas, que combinou a capacidade de modelagem global dos *Transformers* com a eficiência computacional necessária para processar imagens em alta resolução. Esse design permitiu ao modelo capturar tanto características locais quanto contextuais de forma eficiente, superando em mais de 2.7 pontos percentuais os melhores resultados anteriores em tarefas de detecção de objetos e em 3.2 pontos em segmentação semântica.

Além do desempenho excepcional (Liu *et al.*, 2021), o *Swin Transformer* introduz uma complexidade computacional linear em relação ao tamanho da imagem, tornando-o prático para aplicações em larga escala. Essa combinação de alta precisão e eficiência operacional abre novas possibilidades para aplicações em tempo real e em dispositivos com recursos limitados. Para Liu *et al.* (2021), as implicações deste avanço vão além da visão computacional, sugerindo caminhos promissores para a unificação de arquiteturas de processamento visual e de linguagem natural. O sucesso do mecanismo de janelas deslocadas inspira sua adaptação para outras áreas, como o NLP, potencialmente revolucionando a forma como desenvolvemos modelos de inteligência artificial multimodal.

2.1 ADAPTAÇÃO DE DOMÍNIO (*DOMAIN ADAPTATION*) EM VISÃO COMPUTACIONAL

Os avanços recentes em *deep learning* têm impulsionado significativamente a precisão de modelos de segmentação semântica para imagens naturais, abrindo novas possibilidades para sua aplicação em sensoriamento remoto. No entanto, como destacam Zezhong *et al.* (2023), as diferenças fundamentais entre imagens naturais e de satélite limitam a eficácia dos métodos convencionais, exigindo adaptações arquiteturais específicas.

Nesse contexto, soluções como a *High-Resolution Network* (HR-Net) aprimorada (Zezhong *et al.*, 2023) e técnicas de adaptação de domínio baseadas em GANs (Shunping *et al.*, 2021) emergem como alternativas promissoras, capazes de

superar desafios como a escassez de dados rotulados e a variabilidade entre domínios.

Paralelamente, métodos como CORAL (Sun *et al.*, 2016) e MMD (Simon-Gabriel; Schölkopf, 2018) oferecem abordagens matemáticas robustas para alinhamento de distribuições, enquanto estratégias inovadoras como *Contrastive Learning* (CDTCL) (Li *et al.*, 2025) abordam a lacuna entre sensores distintos.

Zezhong *et al.* (2023) ressalta que nos últimos anos, a precisão de reconhecimento de modelos de segmentação semântica em imagens naturais atingiu níveis elevados. Portanto, para os autores (Zezhong *et al.*, 2023) é de grande relevância utilizar algoritmos de segmentação semântica para obter classificação de uso do solo com imagens de sensoriamento remoto.

No entanto, “devido às grandes diferenças entre imagens naturais e de sensoriamento remoto, os algoritmos padrão de segmentação semântica não são eficazes para classificação de uso do solo em imagens de sensoriamento remoto” (Zezhong *et al.*, 2023, p. 01, tradução nossa). Em vista disso, o trabalho de Zezhong *et al.* (2023), consiste em aprimorar a estrutura do algoritmo *High-Resolution Network* (HR-Net) de acordo com as diferenças entre os dois tipos de imagens, tornando-o mais adequado para imagens de sensoriamento remoto.

Além disso, para superar a dependência do algoritmo de segmentação semântica em grandes conjuntos de dados prévios de alta qualidade, foram realizados experimentos com o modelo de adaptação de domínio HR-Net melhorado, e ambos os modelos de adaptação de domínio adversarial e de fusão baseados em HR-Net melhorado e CycleGAN foram projetados para reduzir o trabalho de rotulação manual de dados. Os resultados experimentais mostraram, segundo os autores (Zezhong *et al.*, 2023) que a classificação do nosso algoritmo HR-Net melhorado e dos dois modelos de adaptação de domínio superaram outros algoritmos, demonstrando a eficácia e superioridade das abordagens de seu estudo.

Nesse cenário, Shunping *et al.* (2021) afirma que a precisão da segmentação e classificação de imagens de sensoriamento remoto diminui drasticamente quando as imagens de origem e alvo são de fontes diferentes. Para os autores (Shunping *et al.*, 2021), embora os modelos baseados em *deep learning* tenham impulsionado o desempenho, eles só são eficazes quando treinados com um grande número de imagens de origem rotuladas semelhantes às imagens alvo.

Em vista disso, Shunping *et al.* (2021) propõem uma adaptação de domínio baseada em *Generative Adversarial Network* (GAN) para classificação de cobertura do solo usando novas imagens de sensoriamento remoto alvo que são significativamente diferentes das imagens de origem rotuladas. “Nas GANs, as imagens de origem e alvo são totalmente alinhadas nos espaços de imagem, *feature space* e *output space* em dois estágios por meio de aprendizado adversarial” (Shunping *et al.*, 2021, p. 01, tradução nossa).

Segundo os autores (Shunping *et al.*, 2021), as imagens de origem são traduzidas para o estilo das imagens alvo, que são então usadas para treinar uma *Fully Convolutional Network* (FCN) para segmentação semântica a fim de classificar os tipos de cobertura do solo das imagens alvo. A adaptação de domínio e a segmentação são integradas em um *framework end-to-end*.

Os experimentos realizados em um conjunto de dados *multissource* cobrindo mais de 3.500 km² com 51.560 imagens de satélite de alta resolução (256×256) na cidade de Wuhan e um conjunto de dados intercidades com 11.383 imagens aéreas (256×256) em Potsdam e Vaihingen demonstraram que o método de Shunping *et al.* (2021) superou os recentes métodos de adaptação de domínio baseados em GAN em pelo menos 6,1% e 4,9% nos índices *mean Intersection over Union* (mIoU) e *Overall Accuracy* (OA), respectivamente.

Nesse contexto, Li *et al.* (2025) afirma que, embora os métodos baseados em *Deep Learning* para interpretação de imagens de sensoriamento remoto (RS) tenham apresentado resultados promissores, a lacuna de domínio entre imagens RS e a ausência de conjuntos de dados rotulados específicos para sensores resultam na deterioração significativa de modelos bem treinados para se adaptarem a novas imagens.

Em aplicações práticas, Li *et al.* (2025) propuseram um método de tradução de imagens RS baseado em *Contrastive Learning* (CDTCL) para alcançar rapidamente a conversão de simulação de dados para diferentes sensores e domínios. Especificamente, para imagens não pareadas, projetaram uma *content contrastive loss* para restrições de consistência de conteúdo e uma *style contrastive loss* para alinhamento rápido do estilo de aparência. Além disso, integraram o modelo de segmentação semântica, um modelo flexível que pode ser retreinado conforme a necessidade dos usuários, no *framework* de tradução de imagens para estabelecer um ciclo fechado complementar.

De acordo com Li *et al.* (2025), experimentos extensivos em imagens aéreas, incluindo imagens visíveis e infravermelhas, verificaram que o método funciona eficazmente na segmentação semântica entre domínios e alcança o melhor desempenho.

Sun *et al.* (2016), por sua vez, apresentaram o CORrelation ALignment (CORAL), um método simples, mas eficaz, para adaptação de domínio não supervisionada. O CORAL, segundo os autores (Sun *et al.*, 2016) minimiza a diferença de domínio alinhando as estatísticas de segunda ordem das distribuições de origem e alvo, sem exigir rótulos do domínio alvo. Em contraste com métodos baseados em subespaços, ele alinha as distribuições originais de *features* dos domínios de origem e alvo, em vez das bases de subespaços de dimensão inferior. Também é muito mais simples do que outros métodos de correspondência de distribuição.

“O CORAL apresenta desempenho notável em avaliações extensivas em conjuntos de dados padrão” (Sun *et al.*, 2016, p. 01, tradução nossa). Sun *et al.* (2016) descreveram uma solução que aplica uma transformação linear às *features* de origem para alinhá-las com as *features* alvo antes do treinamento do classificador. Para classificadores lineares, propuseram aplicar o CORAL equivalentemente aos pesos do classificador, levando a maior eficiência quando o número de classificadores é pequeno, mas o número e a dimensionalidade dos exemplos alvo são muito altos.

O resultado, conforme os autores (Sun *et al.*, 2016), é que o CORAL Linear Discriminant Analysis (CORAL-LDA) supera a LDA por uma grande margem em *benchmarks* de adaptação de domínio. Finalmente, Sun *et al.* (2016) estenderam o CORAL para aprender uma transformação não linear que alinha correlações de ativações em camadas de redes neurais profundas (DNNs). Desse modo, provaram que a abordagem *Deep CORAL* funciona perfeitamente com DNNs e alcança desempenho *state-of-the-art* em *benchmarks* padrão.

Contudo, ressaltamos que uma limitação do CORAL é que ele captura apenas estatísticas de segunda ordem e pode não preservar estruturas de ordem superior nos dados. No entanto, como demonstrado por Sun *et al.* (2016), ele funciona muito bem na prática e pode ser combinado com outras funções de perda de alinhamento de domínio.

Em se tratando de métodos, “*Kernel mean embeddings* tornaram-se uma ferramenta popular em *machine learning*. Eles mapeiam medidas de probabilidade

para funções em um *Reproducing Kernel Hilbert Space* (RKHS) (Simon-Gabriel; Schölkopf, 2018, p. 01, tradução nossa).

“A distância entre duas medidas mapeadas define uma semidistância sobre as medidas de probabilidade conhecida como *Maximum Mean Discrepancy* (MMD) (Simon-Gabriel; Schölkopf, 2018, p. 01, tradução nossa). Sendo que, suas “propriedades dependem do *kernel* subjacente e foram vinculadas a três conceitos fundamentais da literatura de kernels: *universal*, *characteristic* e *strictly positive definite kernels* (Simon-Gabriel; Schölkopf, 2018, p. 01, tradução nossa).

Nesse sentido, Simon-Gabriel e Schölkopf (2018), por meio de uma extensão criteriosa das definições convencionais de *kernels* universais, característicos e estritamente positivos, demonstraram que esses três conceitos fundamentais são essencialmente equivalentes. Além disso, forneceram a primeira caracterização matemática completa dos *kernels* cuja *Maximum Mean Discrepancy* (MMD) estabelece uma métrica para a convergência fraca de medidas de probabilidade, representando um avanço significativo na teoria de aprendizagem de máquina. Por fim, mostraram que os *kernel mean embeddings* podem ser estendidos de forma natural para incluir medidas generalizadas conhecidas como distribuições de Schwartz, analisando em detalhe as propriedades fundamentais dessas incorporações generalizadas, o que amplia consideravelmente seu potencial de aplicação em problemas teóricos e práticos.

Assim sendo, entendemos que esses autores sintetizam as contribuições recentes nessa área, explorando desde melhorias em arquiteturas de redes neurais até *frameworks* teóricos para transferência de conhecimento, com o objetivo de fornecer uma visão integrada dos avanços e desafios na classificação automatizada de imagens de sensoriamento remoto.

2.2 MECANISMOS DE ATENÇÃO EM VISÃO COMPUTACIONAL

Os mecanismos de atenção em redes neurais representam um avanço significativo no processamento de imagens, permitindo que os modelos foquem seletivamente nas características mais relevantes, tanto no domínio espacial quanto nos canais de *features*. Dois dos métodos mais influentes nessa área são o Squeeze-and-Excitation (SE), desenvolvido por Hu *et al.* em 2018, e o *Convolutional Block Attention Module* (CBAM), proposto por Woo *et al.* no mesmo ano (Lin *et al.*, 2024).

O SE opera através de um processo em duas etapas: primeiro comprime a informação globalmente (*squeeze*) e depois realiza uma reativação adaptativa (*excitation*) para recalibrar a importância de cada canal de *features* (Lin *et al.*, 2024). Já o CBAM vai além, combinando simultaneamente atenção espacial e por canais em um único módulo integrado, permitindo à rede aprender tanto a localização quanto o conteúdo mais relevante para a tarefa em questão (Lin *et al.*, 2024).

Na aplicação prática ao sensoriamento remoto, especificamente para a detecção de deslizamentos, Lin *et al.* (2024) demonstraram a eficácia desses mecanismos através do modelo CBAM-U-net. Partindo de um conjunto inicial de 109 imagens aéreas de alta precisão, ampliado para 1.090 amostras através de técnicas de aumento de dados, os pesquisadores implementaram módulos CBAM progressivos em cada etapa de *downsampling* de uma arquitetura U-Net.

Os resultados foram impressionantes: o modelo alcançou 98,3% de acerto médio, um coeficiente Dice de 0,877 e mean IoU de 90,2, superando significativamente tanto a U-Net tradicional quanto outras abordagens como DeepLabV3+ e FCN. Particularmente, de acordo com Lin *et al.* (2024), notável foi a capacidade reduzida do modelo de confundir deslizamentos com solo exposto ou estradas, um problema comum nas abordagens convencionais.

Apesar dos resultados promissores, o estudo de Lin *et al.* (2024) também identificou desafios importantes. A base de dados, embora ampliada, ainda era geograficamente limitada à região de Sichuan na China, levantando questões sobre a generalização do modelo para outras localidades. Além disso, a introdução dos mecanismos de atenção trouxe um aumento computacional significativo, tanto no tempo de treinamento quanto na inferência. Outra limitação apontada pelos autores (Lin *et al.*, 2024), foi a dependência exclusiva de dados ópticos, que podem não conter todas as informações necessárias para uma análise abrangente. Esses achados ressaltam a importância de se buscar um equilíbrio entre ganhos de precisão e custos computacionais, além da necessidade de incorporar dados multimodais em futuras pesquisas na área.

No contexto do processamento de imagens multibanda em sensoriamento remoto, Duan *et al.* (2024) identificaram uma limitação significativa nos mecanismos de atenção tradicionais, como SE e CBAM. Esses métodos convencionais mostraram-se menos eficazes quando aplicados a imagens com múltiplas bandas espectrais, devido às diferenças estatísticas intrínsecas entre os diversos canais.

Para superar essa barreira, os pesquisadores (Duan *et al.*, 2024) desenvolveram o *Feature Entropy Attention Mechanism* (FEM), uma abordagem que emprega o conceito de entropia da informação para avaliar e ponderar as *features* de maneira mais inteligente e fundamentada.

Os resultados experimentais do FEM impressionaram pela sua consistência. Quando testado no conjunto de dados Vaihingen, o mecanismo superou o SEM em 0,90%, o CBAM em 1,10% e o FCA em 0,40% na métrica mIoU. O desempenho foi ainda mais destacado no *dataset OpenEarthMap*, com melhorias de 2,30%, 2,20% e 2,10% em relação aos mesmos métodos comparativos, respectivamente. Esses números robustos demonstram a capacidade superior do FEM em identificar e priorizar as características mais relevantes para a análise.

A principal conclusão Duan *et al.* (2024) aponta que o FEM não apenas oferece uma seleção mais eficiente de *features* críticas, como também apresenta a vantagem prática de poder ser facilmente integrado a arquiteturas de redes neurais já existentes, particularmente em tarefas de classificação de uso do solo. Esta adaptabilidade torna o mecanismo especialmente valioso para aplicações em sensoriamento remoto, onde a diversidade de dados e condições de aquisição exige soluções flexíveis.

Nesse ponto, enquanto o CBAM demonstrou excelência em tarefas específicas como a detecção de deslizamentos de terra, o FEM surgiu como uma solução mais adequada para desafios complexos como a classificação de uso do solo, especialmente em cenários com grande variabilidade espectral. Esta evolução contínua das técnicas de atenção (do CBAM ao FEM) ilustrou o rápido avanço no campo da análise de imagens por inteligência artificial, com cada nova abordagem superando as limitações de suas predecessoras enquanto abre caminho para aplicações mais sofisticadas e precisas no domínio do sensoriamento remoto.

3 METODOLOGIA

3.1 PRÉ-PROCESSAMENTO E CONSTRUÇÃO DOS DATASETS

O sucesso de qualquer modelo de aprendizado profundo é intrinsecamente dependente da qualidade e da consistência dos dados de entrada. Para garantir a robustez e a reprodutibilidade de nossos experimentos, foi implementado um *pipeline* de pré-processamento, dividido em três etapas principais: 1) geração dos ortofotos e

anotação das máscaras de segmentação; 2) criação e padronização dos *datasets* de treinamento e validação; 3) aplicação de técnicas de aumento de dados.

Os dados brutos para este trabalho consistem em imagens aéreas de alta resolução de dois câmpus universitários distintos, UCDB (Brasil) e UPS (Equador), capturadas por um Veículo Aéreo Não Tripulado (VANT) do modelo DJI Phantom Pro 4. As imagens da UCDB foram processadas com o software *Agisoft Metashape*, enquanto as da UPS foram processadas com o Pix4D, resultando em duas ortofotos georreferenciadas que serviram como base para a análise.

O processo de anotação foi realizado manualmente no software QGIS, onde um especialista delineou polígonos para as classes fundo, grama e vegetação. Para lidar com as geometrias complexas da vegetação, o processo foi auxiliado pela ferramenta *Segment Anything Model* (SAM), com subseqüentes correções manuais para refinar os contornos e garantir a precisão dos rótulos. As classes finais definidas para a tarefa de segmentação semântica foram: Fundo (*Background*), Grama (*Grass*) e Vegetação (*Vegetation*), que inclui árvores e arbustos.

Com as ortofotos e as máscaras de anotação em mãos, o próximo passo foi convertê-los em um formato adequado para o treinamento dos modelos. Primeiramente, as ortofotos e suas respectivas máscaras foram divididas em *tiles* (recortes) de 1024x1024 *pixels*, utilizando um pequeno *stride* para garantir uma sobreposição entre os recortes adjacentes e preservar a continuidade contextual.

Outro passo de padronização foi a conversão das máscaras de anotação. As máscaras originais, exportadas do QGIS em formato RGB (onde, por exemplo, a cor vermelha representava "Grama" e a azul, "Vegetação"). Para somente os experimentos *self-training* foram convertidas em máscaras de canal único em escala de cinza. Neste novo formato, o valor de intensidade de cada *pixel* correspondeu diretamente ao ID da sua classe (0 para Fundo, 1 para Grama, 2 para Vegetação). Esta conversão foi fundamental para a compatibilidade com a função de perda *CrossEntropyLoss* utilizada no treinamento.

Finalmente, os *tiles* de cada domínio foram divididos em conjuntos de treinamento (80%) e validação (20%). Para aumentar a diversidade dos dados de treinamento e reduzir o risco de *overfitting*, aplicamos um conjunto de técnicas de aumento de dados padrão durante o treinamento. Para todos os experimentos, os *tiles* de entrada de 1024x1024 foram redimensionados para a resolução de 512x512 *pixels*, esperada pelos modelos.

3.2 ARQUITETURA PROPOSTA: A ABORDAGEM TRIBRANCH

Modelos de segmentação semântica, sejam eles baseados em Redes Neurais Convolucionais (CNNs) ou em Transformers, tipicamente processam imagens de entrada tratando os três canais de cor (RGB) como uma única entidade desde a primeira camada. Embora eficaz, essa abordagem de fusão imediata apresenta uma vulnerabilidade em cenários de adaptação de domínio, especialmente em imagens aéreas.

Nesses domínios, as propriedades estatísticas de cada banda espectral podem variar drasticamente e de forma independente devido a fatores como iluminação; condições atmosféricas e calibração do sensor.

Um modelo treinado dessa forma pode aprender *features* que estão intrinsecamente "emaranhados" com o estilo de um domínio específico (associando "vegetação" a um tom de verde particular do domínio A, por exemplo), o que leva a uma queda acentuada de performance quando confrontado com as características do domínio B.

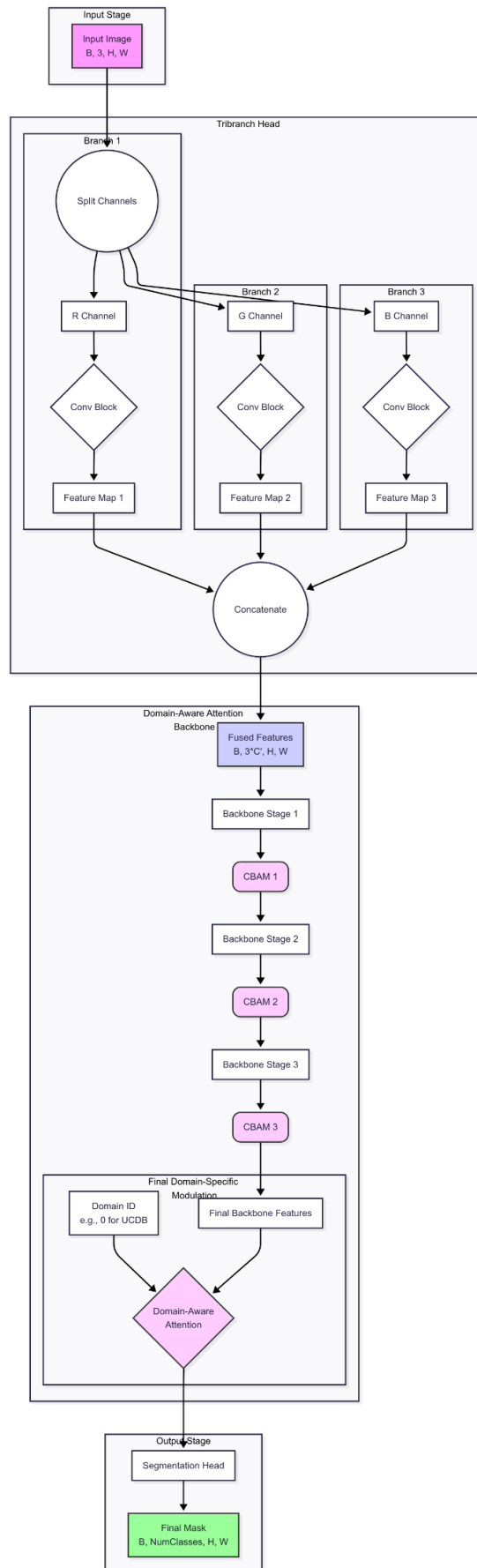
Para mitigar essa dependência do estilo do domínio, propomos uma nova arquitetura de entrada denominada Tribbranch. A nossa hipótese fundamental é que, ao forçar a rede a aprender representações especializadas para cada canal espectral de forma independente antes da fusão, podemos extrair características mais fundamentais e invariantes ao domínio.

A arquitetura implementa essa ideia através de um processo de "*Split-Process-Fuse*": a imagem de entrada é separada em seus três canais (R, G, B); cada um é processado por um pequeno bloco convolucional "especialista" com pesos próprios; os mapas de características resultantes são então concatenados.

O resultado desse processo é um tensor de entrada para o *backbone* principal que é significativamente mais rico e informativo do que a imagem original de 3 canais. Dessa forma, em vez de aprender a partir de pixels brutos, o *backbone* (seja ele um ResNet ou um *Swin Transformer*) começa seu processo de extração de features a partir de uma representação de alta dimensão que já encapsula os padrões texturais e estruturais mais salientes de cada banda espectral. Em outras palavras, a hipótese central desta abordagem é que, ao processar as bandas espectrais de uma imagem (Vermelho, Verde e Azul) de forma independente antes de alimentar o *backbone* principal, o modelo pode aprender representações de baixo nível mais ricas e

robustas, que são menos sensíveis a variações de cor e iluminação entre os domínios. Assim sendo, a lógica da arquitetura Tribbranch é executada em três estágios sequenciais e bem definidos, como apresentado na Figura 1.

Figura 1 - Arquitetura Tribbranch



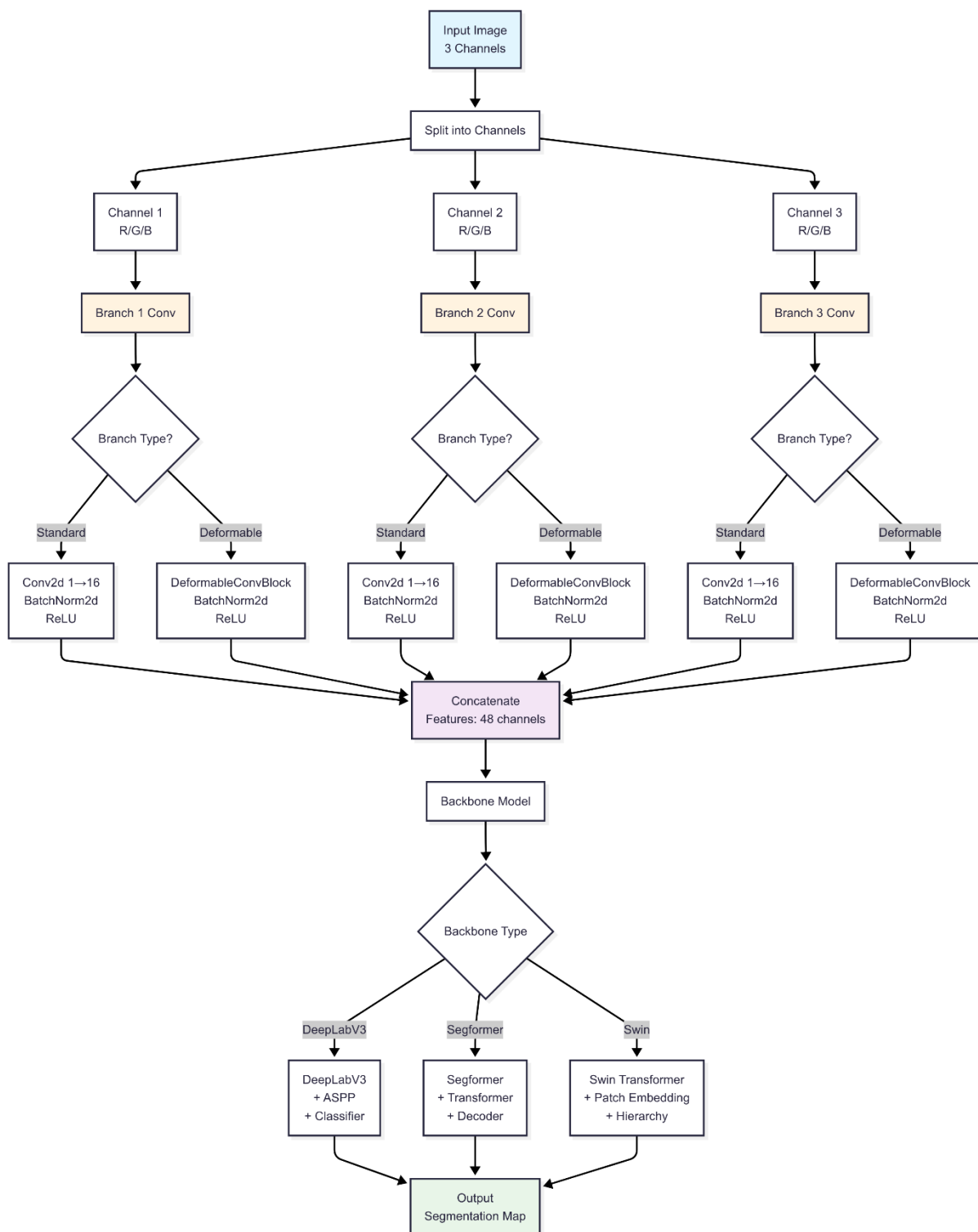
Fonte: Elaborado pelo autor

O estágio fundamental e primeiro passo da arquitetura Tribbranch é a Separação de Canais (*Split*). O primeiro estágio desacopla a entrada (Figura 2). Uma imagem padrão de 3 canais ($B, 3, H, W$) é dividida em três tensores de canal único ($B, 1, H, W$), cada um representando uma banda espectral (R, G e B). Esta separação é a base para o processamento especializado que se segue. Diferentemente das abordagens convolucionais padrão, que processam os canais RGB como uma única entrada tridimensional desde a primeira camada, nossa metodologia inicia com o desacoplamento explícito dessas bandas espectrais.

A motivação para este passo reside na hipótese de que a fusão imediata dos canais força o modelo a aprender características de baixo nível que estão intrinsecamente correlacionadas com as propriedades colorimétricas específicas do domínio fonte, criando uma representação "emaranhada" que é frágil a variações de iluminação; sensor e atmosfera.

Tecnicamente, esta operação decompõe o tensor de entrada de dimensões ($B, 3, H, W$) em três tensores independentes de ($B, 1, H, W$), cada um correspondendo a uma única banda espectral: vermelho, verde e azul.

Ao isolar os canais, garantimos que a etapa subsequente de Processamento Especializado (*Process*) opere sobre informações puramente monocromáticas. Cada um desses fluxos de dados serve então como entrada para um bloco convolucional distinto, permitindo que a rede aprenda filtros otimizados para as texturas e padrões inerentes a cada canal individualmente, antes de qualquer fusão de informações.

Figura 2 - Separação de canais (*Split*)

Fonte: Elaborado pelo autor

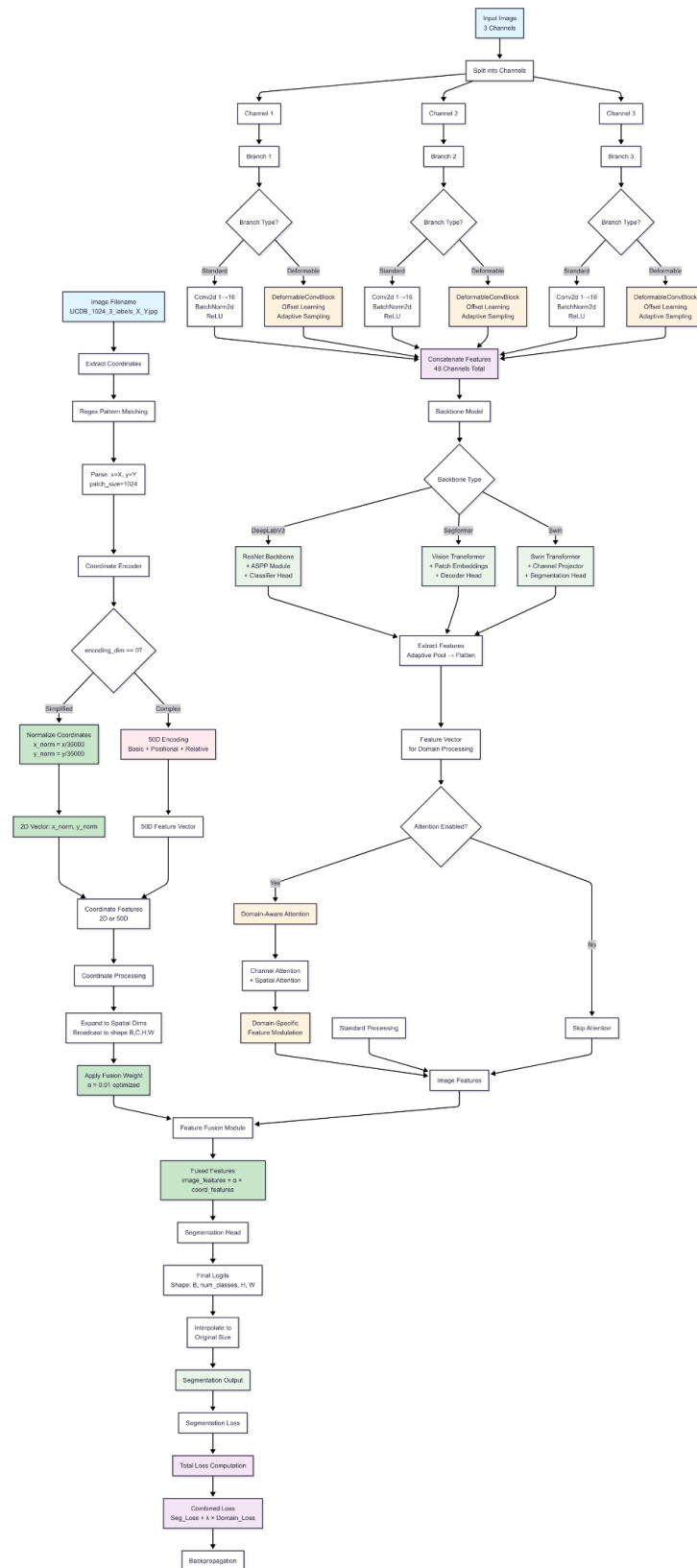
Após a separação dos canais, o estágio de Processamento Especializado (*Process*) entra em ação. Cada um dos três fluxos de canal único é então alimentado em seu próprio bloco convolucional especialista (Figura 3). Esses blocos são estruturalmente idênticos, mas não compartilham pesos, permitindo que cada um se

especialize em extrair os padrões, texturas e características mais salientes da sua respectiva banda espectral. Na sua configuração padrão, cada bloco especialista é composto por uma sequência de: uma camada Conv2d com um kernel de 3x3; uma camada de normalização (BatchNorm2d ou *GroupNorm*) e uma função de ativação ReLU. De modo geral, os pesos entre os três blocos não são compartilhados, permitindo que cada um se especialize em aprender os filtros mais eficazes para os padrões texturais e estruturais inerentes à sua respectiva banda de entrada.

A principal função de cada bloco é transformar seu mapa de entrada de um único canal em um mapa de características rico e multidimensional de (B, C', H, W). A dimensionalidade C' é controlada por um hiperparâmetro fundamental que denominamos *initial_features*. Este valor (16 ou 40) determina a "capacidade" de cada ramo especialista, definindo a profundidade do tensor de características que será posteriormente fundido.

Como uma variação arquitetônica para aprimorar a extração de características em geometrias complexas, também exploramos a substituição da camada Conv2d padrão por um bloco de Convolução Deformável. Este bloco aprende dinamicamente os deslocamentos espaciais para os pontos de amostragem do kernel, permitindo que o campo receptivo se ajuste de forma flexível aos contornos dos objetos na imagem, como vegetação e edifícios, visando uma representação de características ainda mais precisa.

Figura 3 - Processamento Especializado (*Process*) e Fusão de Features (*Fuse*)



Fonte: Elaborado pelo autor

O estágio final do cabeçalho Tribbranch é a Fusão de Características (*Fuse*), onde as representações especializadas, geradas de forma independente, são consolidadas em uma única e rica entrada para o *backbone* principal. No estágio final, os três mapas de características especializadas resultantes são concatenados ao longo da dimensão do canal (Figura 3). Este processo de fusão cria um único tensor de características de alta dimensionalidade ($B, 3 \cdot C', H, W$). Este tensor consolidado, que agora contém as informações pré-processadas e independentes de todas as três bandas, é então utilizado como a entrada para o *backbone* principal da rede de segmentação (como DeepLabV3 e Swin Transformer).

A principal vantagem desta abordagem é que o *backbone* principal da rede de segmentação não opera mais sobre os *pixels* brutos da imagem de 3 canais, mas sim sobre esta representação de características pré-processada e significativamente mais informativa. Assim, a arquitetura Tribbranch atua como um estágio de pré-extração de características que alimenta o modelo principal com uma visão mais rica e desacoplada dos dados de entrada, preparando o terreno para uma extração de características de alto nível mais eficaz ao domínio.

Para aprimorar ainda mais a capacidade do modelo de se adaptar às geometrias complexas e irregulares encontradas em imagens aéreas (como contornos de árvores, rios e edifícios), exploramos uma variação do estágio de Processamento Especializado utilizando Convoluções Deformáveis (*Deformable Convolutions*) (Figura 3).

Ao contrário de uma convolução padrão (Conv2d), que aplica um filtro em uma grade rígida e fixa (3x3, por exemplo), uma convolução deformável aprende dinamicamente os deslocamentos (*offsets*) para os pontos de amostragem do kernel, permitindo que o campo receptivo da convolução se deforme e se ajuste à forma real dos objetos na imagem.

A convolução padrão (Conv2d), um pilar das CNNs, opera aplicando um kernel sobre uma grade de amostragem rígida e fixa. Embora eficaz para características gerais, essa rigidez geométrica é uma limitação fundamental ao lidar com os objetos encontrados em imagens aéreas.

Estruturas como copas de árvores, leitos de rios e contornos de edifícios, raramente se conformam a formas quadradas. Um kernel fixo pode amostrar simultaneamente pixels do objeto de interesse e do fundo, resultando em mapas de

características que não representam com precisão a verdadeira forma do objeto. Essa imprecisão na extração de características de baixo nível pode ser amplificada nas camadas mais profundas da rede, prejudicando a performance final da segmentação, especialmente em um cenário de adaptação de domínio onde as características geométricas podem variar.

Para resolver essa limitação, propomos uma variação arquitetônica que atua diretamente no coração da nossa abordagem: o estágio de Processamento Especializado (*Process*). Nesta variação, o bloco convolucional padrão dentro de cada um dos três ramos especialistas é substituído por um bloco de Convolução Deformável.

A modificação é cirúrgica e intencional: ela ocorre antes de qualquer fusão de características, permitindo que cada ramo aprenda a se adaptar geometricamente às particularidades da sua própria banda espectral. As etapas de separação (*Split*) e fusão (*Fuse*) da arquitetura Tribbranch permanecem inalteradas.

Nesse contexto, a implementação de um bloco de Convolução Deformável envolve duas etapas: camada de *offsets* e convolução deformável.

Primeiramente, uma camada Conv2d padrão, paralela ao fluxo principal, é utilizada para aprender os deslocamentos (*offsets*). Esta camada de *offsets* gera um mapa de vetores 2D para cada ponto de amostragem do kernel, indicando para onde cada ponto deve se mover.

Depois, a camada principal DeformableConv2d utiliza esses *offsets* para amostrar o mapa de características de entrada em localizações deslocadas e flexíveis, em vez da grade rígida.

O racional por trás desta modificação é dotar cada ramo especialista de flexibilidade geométrica. Pois, ao permitir que o campo receptivo se deforme e se ajuste ativamente aos contornos dos objetos, o modelo pode aprender representações de características que são mais fiéis à verdadeira morfologia das classes de interesse.

Acreditamos que, ao aprender não apenas quais *features* extrair, mas também a forma espacial ideal para extrair, a arquitetura Tribbranch pode gerar representações mais precisas, melhorando a capacidade do modelo de generalizar entre domínios com variações geométricas.

3.3 ESTRATÉGIAS DE ADAPTAÇÃO DATA-CÊNTRICAS

Uma abordagem alternativa para mitigar o *domain shift*, complementar às modificações arquitetônicas, é a adaptação data-cêntrica. A motivação por trás desta estratégia é que, em vez de exigir que o modelo aprenda a ser robusto a diferentes estilos visuais, podemos pré-processar os dados do domínio fonte para que eles se assemelhem estilisticamente aos dados do domínio alvo. Desse modo, ao reduzir a discrepância visual entre os domínios antes mesmo do treinamento, o modelo de segmentação pode se concentrar em aprender as características semânticas e estruturais dos objetos, que são mais propensas a serem invariantes entre os domínios.

Neste trabalho, exploramos duas técnicas data-cêntricas com diferentes níveis de complexidade: Normalização de histograma e Tradução de imagem para imagem não pareada. A primeira trata-se de um método clássico que alinha as distribuições globais de cor e brilho entre as imagens. A segunda, utiliza o modelo *Contrastive Unpaired Translation* (CUT).

Esta abordagem baseada em GANs é capaz de aprender mapeamentos de estilo complexos, transferindo não apenas a cor, mas também as características texturais do domínio alvo para as imagens do domínio fonte, gerando um *dataset* de treinamento sintético, porém estilisticamente alinhado.

Em se tratando de Normalização de histograma (*Histogram Matching*), uma das manifestações mais evidentes do *domain shift* em imagens aéreas e de sensoriamento remoto são as variações nas propriedades colorimétricas. Diferenças na iluminação solar, condições atmosféricas (como névoa), calibração do sensor e até mesmo a estação do ano podem alterar drasticamente a distribuição global de cores e brilho entre dois *datasets*, como o caso UCDB e UPS.

Um modelo treinado no domínio A pode aprender a associar a classe "vegetação" a um tom de verde muito específico, e falhar ao encontrar um tom de verde ligeiramente diferente no domínio B. A motivação para usar a Normalização de histograma é atacar diretamente essa fonte de erro, alinhando as características de cor e luminosidade de baixo nível antes que a imagem seja processada pelo modelo. A hipótese é que, ao forçar as imagens do domínio fonte a terem a mesma "paleta de cores" do domínio alvo, o modelo pode se concentrar em aprender características mais robustas e invariantes, como a forma e a textura dos objetos.

A Normalização de Histograma, também conhecida como *Histogram Matching* ou Especificação de histograma, é uma técnica clássica de processamento de imagem que modifica a distribuição de intensidade de pixels de uma imagem (a imagem fonte) para que ela se assemelhe à distribuição de uma outra imagem (a imagem de referência).

O processo opera da seguinte forma: calculando o histograma de intensidade de pixels para cada canal de cor de ambas as imagens; encontrando um mapeamento que transforma os valores de pixel da imagem fonte; fazendo o histograma cumulativo se aproximar o máximo possível do histograma cumulativo da imagem de referência. O resultado é uma nova imagem que mantém o conteúdo estrutural da imagem fonte, mas adota as características globais de cor, brilho e contraste da imagem de referência.

Em nosso trabalho, implementamos a Normalização de histograma como uma estratégia de adaptação data-cêntrica *offline*. Para cada imagem no conjunto de treinamento do domínio fonte (UCDB), uma imagem de referência foi selecionada aleatoriamente do conjunto de treinamento do domínio alvo (UPS). A função *match_histograms* da biblioteca *scikit-image* foi utilizada para realizar a transformação, operando de forma independente em cada um dos três canais de cor (RGB).

Este processo foi executado uma única vez para gerar um novo conjunto de dados de treinamento, onde todas as imagens do domínio fonte foram estilisticamente alinhadas com o domínio alvo. Este novo *dataset* foi então utilizado para treinar o modelo de segmentação do zero, permitindo uma avaliação direta do impacto desta técnica de normalização na performance de generalização do modelo.

Nesse contexto, embora a Normalização de histograma seja eficaz para corrigir discrepâncias globais de cor e iluminação, o *domain shift* em imagens aéreas frequentemente se manifesta de formas mais complexas. Variações em texturas; padrões de vegetação; estilos arquitetônicos e artefatos de compressão do sensor, constituem um *gap* de domínio de alta frequência que o alinhamento de histogramas não consegue resolver.

Para atacar este problema mais profundo, recorreremos a uma estratégia data-cêntrica mais poderosa, baseada em aprendizado profundo: a Tradução de imagem para imagem não pareada. A motivação foi utilizar uma rede generativa para aprender o mapeamento estilístico complexo entre os domínios, transformando as imagens da

fonte para que elas pareçam ter sido capturadas no domínio alvo, preservando ao mesmo tempo o conteúdo semântico original.

O *Contrastive Unpaired Translation* (CUT) é um modelo utilizado para tradução de imagem para imagem não pareada, sendo uma alternativa mais rápida e eficiente em termos de memória ao seu predecessor, o *CycleGAN*. A inovação central do CUT é a substituição da custosa perda de consistência de ciclo (*cycle-consistency loss*) por uma perda contrastiva aplicada em nível de *patches*.

A ideia foi maximizar a informação mútua entre os *patches* correspondentes da imagem de entrada e da imagem traduzida. Para um determinado *patch* na imagem de saída (a "*query*"), o modelo é treinado para que o *patch* correspondente na imagem de entrada (o "positivo") tenha uma representação em um espaço de características latente que seja mais próxima da *query* do que as representações de todos os outros *patches* da imagem (os "negativos"). Ao treinar um gerador para "enganar" essa perda contrastiva, ele é forçado a produzir imagens estilisticamente transformadas, mas que mantêm uma forte correspondência estrutural com a entrada.

Em nosso fluxo de dados, o CUT foi implementado como uma etapa de pré-processamento *offline* para gerar *datasets* de treinamento sintéticos e estilisticamente alinhados. Treinamos dois modelos CUT distintos para permitir a adaptação bidirecional: UCDB $\rightarrow$ UPS e UPS $\rightarrow$ UCDB.

Descobrimos empiricamente que a estabilidade do treinamento e a qualidade visual das imagens geradas eram altamente sensíveis aos hiperparâmetros. Resultados ótimos foram alcançados através de: treinamento prolongado (até 600 épocas); taxas de aprendizado baixas (entre $1e-4$ e $6e-5$); tipos de experimentos. Este processo resultou em dois novos conjuntos de dados de treinamento.

Para avaliar o impacto desta técnica, conduzimos dois tipos de experimentos de segmentação: a Tradução total e Aumento de dados com CUT. No primeiro, o modelo foi treinado utilizando apenas o *dataset* de imagens traduzidas pelo CUT. No segundo, o modelo foi treinado em um *dataset* combinado, criado a partir da concatenação do conjunto de treinamento original com o seu correspondente traduzido pelo CUT. Esta abordagem testou a hipótese de que expor o modelo a ambos os estilos durante o treinamento pode levar a uma maior robustez.

3.3.1 Estratégias de Treinamento e Regularização

Além das modificações arquitetônicas, exploramos estratégias que alteram o processo de treinamento para forçar o modelo a aprender representações mais robustas. Estas técnicas não alteram a estrutura física do modelo, mas sim a função objetivo que ele otimiza, guiando-o para um espaço de características mais invariante ao domínio. O Alinhamento de *features* no espaço latente foi uma delas.

A principal hipótese por trás do alinhamento de *features* é que, embora as imagens dos domínios fonte e alvo possam parecer visualmente distintas, um modelo de segmentação ideal deve mapeá-las para um espaço de características latente onde as classes semânticas sejam indistinguíveis, independentemente do domínio de origem.

O objetivo, portanto, é penalizar o modelo durante o treinamento se as distribuições de características do domínio fonte e do domínio alvo estiverem muito distantes. Ao forçar essas distribuições a se alinharem, incentivamos o modelo a focar nas características semânticas essenciais e a ignorar os "ruídos" estilísticos específicos de cada domínio.

Para implementar este alinhamento, integramos duas funções de perda de adaptação de domínio bem estabelecidas, que são calculadas a partir dos mapas de características extraídos do backbone do modelo, antes da camada de classificação final: Perda CORAL (*Correlation Alignment*) e Perda MMD (*Maximum Mean Discrepancy*).

A primeira é uma técnica que visa alinhar as estatísticas de segunda ordem das distribuições de características. Ela calcula a matriz de covariância para os lotes de características dos domínios fonte e alvo. A perda CORAL é então definida como a diferença (medida pela norma de Frobenius) entre essas duas matrizes de covariância. Ao minimizar essa perda, o modelo é incentivado a aprender transformações que fazem com que as características de ambos os domínios tenham a mesma estrutura de correlação interna, ou "forma" (Sun *et al.*, 2016).

A MMD, por sua vez, mede a disparidade entre as distribuições calculando a distância entre as médias das amostras em um Espaço de Hilbert de Kernel Reprodutor (RKHS). Em termos mais simples, ela projeta as características em um espaço de alta dimensão e calcula a distância entre o "centro de gravidade" do lote de características do domínio fonte e o do domínio alvo. Minimizar a perda MMD força

o modelo a gerar representações onde, em média, um lote de amostras da fonte é indistinguível de um lote de amostras do alvo.

Durante o treinamento, a perda total do modelo é calculada como uma soma ponderada da perda de segmentação padrão (CrossEntropyLoss) e dessas perdas de adaptação: $L_{total} = L_{seg} + \lambda_{coral} * L_{coral} + \lambda_{mmd} * L_{mmd}$. Assim, os hiperparâmetros λ controlam a força com que o modelo é incentivado a alinhar as distribuições de características.

Para aumentar ainda mais a robustez do modelo, implementamos técnicas que operam diretamente sobre as representações internas da rede durante o treinamento. A principal técnica utilizada nesta categoria é o *MixStyle*, uma forma de aumento de dados no nível das *features*. Em vez de alterar a imagem de entrada, o *MixStyle* atua sobre os mapas de características intermediários. Em nossa arquitetura, ele é aplicado estrategicamente após o estágio de fusão do Tribbranch, operando sobre as *features* combinadas antes que elas entrem no *backbone* principal. Durante o treinamento, ele calcula as estatísticas de instância (média e desvio padrão) de diferentes mapas de características dentro de um mesmo lote e as mistura. Este processo força o modelo a aprender a separar o conteúdo semântico de uma *feature* do seu estilo, tornando-o menos dependente das características estilísticas superficiais de um domínio e mais focado nas propriedades estruturais invariantes dos objetos.

Adicionalmente, utilizamos a técnica de regularização padrão *Dropout*. Em nossa implementação, ela é empregada principalmente nas camadas densas do Discriminador de domínio da nossa arquitetura adversarial. Durante cada passo de treinamento, as ativações de uma fração dos neurônios são aleatoriamente zeradas com uma determinada probabilidade. Este processo impede que o discriminador se torne excessivamente confiante ou se apoie em um pequeno conjunto de características para diferenciar os domínios, incentivando o aprendizado de representações mais distribuídas e robustas, o que por sua vez melhora a estabilidade do treinamento adversarial.

3.4 O MODELO DOMAIN-AWARE TRIBRANCHNET

Para ir além das estratégias data-cêntricas, desenvolvemos o DomainAwareTriBranchNet, uma arquitetura de segmentação semântica projetada explicitamente para ser robusta ao *domain shift*. Este modelo evolui a partir da

arquitetura Tribbranch básica, integrando múltiplos mecanismos de atenção e consciência de contexto para refinar a extração de características e adaptar ativamente suas representações internas com base no domínio da imagem de entrada.

A arquitetura é composta por uma série de componentes modulares que atuam em diferentes estágios do processamento, desde a entrada até a extração de características de alto nível. Os componentes principais são: Módulos de Atenção de Bloco Convolutacional (CBAM); Consciência Espacial com Coordenadas (*Coordinate Awareness*) e Atenção Consciente de Domínio (*DomainAwareAttention*).

O primeiro aprimoramento sobre a base Tribbranch é a integração de Módulos de Atenção de Bloco Convolutacional (CBAM). Inseridos após os estágios principais do *backbone* (por exemplo, após cada bloco layer de um ResNet), os módulos CBAM aplicam sequencialmente atenção de canal e espacial, permitindo que o modelo aprenda a recalibrar dinamicamente a importância dos mapas de características ("o que" é importante) e a focar em regiões espaciais salientes ("onde" é importante) em múltiplas escalas, refinando progressivamente as representações e suprimindo ruídos.

Para fornecer ao modelo um contexto espacial absoluto e relativo, que é particularmente relevante em imagens aéreas de grande escala, implementamos a Consciência Espacial com Coordenadas (*Coordinate Awareness*). As coordenadas normalizadas de cada *pixel* são codificadas e concatenadas como canais adicionais à entrada do *backbone*, permitindo que a rede aprenda a correlacionar características visuais com sua localização geográfica.

A contribuição arquitetônica final e mais específica é o módulo de Atenção Consciente de Domínio (*DomainAwareAttention*). Aplicado sobre o mapa de características final e semanticamente rico do *backbone*, este módulo utiliza um *domain_id* (0 para UCDB, 1 para UPS) para selecionar um conjunto de pesos de canal específicos para aquele domínio. Ele então modula o mapa de características, efetivamente "amplificando" as *features* que são mais relevantes para o domínio em questão e atenuando as que não são.

Em conjunto, esses componentes transformam a arquitetura Tribbranch de um extrator de características robusto em um sistema adaptativo que não apenas foca nas informações mais importantes, mas também ajusta ativamente sua lógica interna para se alinhar com as propriedades estatísticas do domínio de entrada.

3.4.1 Refinamento de Features com Atenção (CBAM)

As Redes Neurais Convolucionais (CNNs) constroem representações hierárquicas de características, combinando informações espaciais e de canal em suas camadas. No entanto, em um *backbone* padrão, cada canal no mapa de características é tratado com igual importância.

Em tarefas complexas como a segmentação de imagens aéreas, nem todas as características são igualmente informativas, algumas podem representar ruídos ou detalhes irrelevantes específicos do domínio, outras contêm informações semânticas cruciais. Dessa maneira, a motivação para integrar um mecanismo de atenção é dar ao modelo a capacidade de aprender a recalibrar dinamicamente seus próprios mapas de características, amplificando as informações úteis e suprimindo as irrelevantes. Ao forçar o modelo a aprender "o que" e "onde" focar, esperamos que ele crie representações mais robustas e generalizáveis, que dependam menos das características superficiais de um domínio específico.

Para implementar essa capacidade de foco, integramos o Módulo de Atenção de Bloco Convolucional (CBAM), um módulo de atenção leve e eficiente. O CBAM infere um mapa de atenção de forma sequencial ao longo de duas dimensões distintas: canal e espacial. Dado um mapa de características de entrada, o CBAM primeiro gera um mapa de atenção de canal para reavaliar as características com base na sua importância e, em seguida, gera um mapa de atenção espacial para focar nas regiões mais salientes.

A inserção dos módulos CBAM em nossa arquitetura foi projetada para ser estratégica, alinhando-se com a natureza hierárquica dos *backbones* baseados em ResNet. Em vez de aplicar a atenção apenas no final, um módulo CBAM é inserido após cada um dos principais estágios de extração de características (especificamente, após layer1, layer2 e layer3). O racional por trás desta aplicação progressiva é permitir que o modelo refine suas características em múltiplas escalas e níveis de abstração.

Nos estágios iniciais (após layer1), o modelo aprende a focar em características de baixo nível, como bordas e texturas fundamentais. À medida que os dados avançam para os estágios mais profundos (layer2 e layer3), os mapas de características se tornam mais semanticamente ricos, representando partes de objetos e padrões complexos. Ao aplicar a atenção em cada um desses estágios, o modelo aprende a identificar e amplificar as características mais informativas em cada

nível de abstração. Este refinamento progressivo garante que o mapa de características final, que é passado para o módulo de DomainAwareAttention e para o decodificador, seja o mais limpo e semanticamente relevante possível, tendo sido filtrado e focado em múltiplas etapas ao longo do *backbone*.

3.5 ESTRATÉGIA DE TREINAMENTO COM AUTOAPRENDIZAGEM (*SELF-TRAINING*)

Uma das abordagens mais promissoras para a adaptação de domínio não supervisionada é a Autoaprendizagem (*Self-Training*), também conhecido como *Pseudo-Labeling*. A motivação por trás desta técnica é utilizar o próprio conhecimento do modelo, treinado no domínio fonte, para gerar novos dados de treinamento no domínio alvo, sem a necessidade de rótulos verdadeiros (*ground truth*).

A ideia é que, embora o modelo treinado na fonte não seja perfeito no domínio alvo, suas previsões mais confiantes são provavelmente corretas e podem ser usadas como um sinal de supervisão para um processo de retreinamento. Este método visa refinar e adaptar o modelo ao estilo e características do domínio alvo, com o objetivo de se aproximar do "limite superior" de performance que seria alcançado se tivéssemos acesso a todos os rótulos do domínio alvo.

Nosso fluxo de dados de *Self-Training* é implementado em um processo sequencial de três etapas: Geração de pseudo-rótulos, Filtragem por confiança e Retreinamento (*Fine-tuning*)

A primeira etapa inicia com a seleção do nosso modelo de melhor performance treinado exclusivamente no domínio fonte (o "modelo professor"). Este modelo é então utilizado para inferir em todo o conjunto de treinamento do domínio alvo. Para cada pixel de cada imagem alvo, a saída da rede (*logits*) é passada por uma função Softmax para obter um vetor de probabilidades, indicando a confiança do modelo para cada classe.

Na etapa de Filtragem por confiança aplicamos um limiar de confiança pré-definido (*threshold* = 0.9) sobre essas probabilidades. Para um dado pixel: Se a probabilidade da classe com maior pontuação exceder o limiar, o pixel é rotulado com o ID daquela classe (0, 1 ou 2). Caso contrário, se a confiança do modelo for insuficiente, o pixel recebe um índice de ignorância (*ignore_index* = 255). Este passo é crucial para a qualidade do processo, pois garante que apenas os "rótulos falsos" de alta confiança sejam utilizados, prevenindo a propagação de erros do modelo

professor. Os resultados são salvos como novas máscaras de segmentação de canal único.

Finalmente, na etapa de Retreinamento (*Fine-tuning*), o modelo professor original é carregado novamente e seu treinamento é continuado sobre um novo conjunto de dados combinado. Este conjunto é formado pela concatenação de *Dataset* de treinamento original do domínio fonte (com seus rótulos verdadeiros e de alta qualidade) e *Dataset* de treinamento do domínio alvo (com as imagens originais e seus novos pseudo-rótulos parciais)

A função de perda *CrossEntropyLoss* é configurada com o parâmetro *ignore_index=255*, garantindo que os pixels de baixa confiança não contribuam para o cálculo do gradiente. Desta forma, o modelo se adapta às características do domínio alvo utilizando apenas os exemplos que ele mesmo gerou com alta certeza, num ciclo virtuoso de autoaprendizagem.

No momento da escrita deste trabalho, os experimentos utilizando a metodologia de *Self-Training* estão em andamento. Os resultados preliminares são promissores, mas a análise completa e conclusiva ainda está sendo realizada e será apresentada na seção de Resultados.

3.6 DETALHES DE IMPLEMENTAÇÃO E SELEÇÃO DE HIPERPARÂMETROS

Para garantir uma comparação justa e otimizar a performance de cada arquitetura, foi realizada uma busca sistemática por hiperparâmetros em um espaço de configurações predefinido. Os resultados apresentados neste trabalho correspondem sempre à melhor combinação de hiperparâmetros encontrada para cada configuração experimental, com base na métrica de mIoU no conjunto de validação. A seguir, detalhamos os critérios e as faixas de valores exploradas para cada hiperparâmetro chave.

Quanto aos otimizadores, foi testado um conjunto de otimizadores padrão, incluindo SGD, Adam e Adagrad. Para os backbones baseados em *Transformer* (*Segformer* e *Swin*), o otimizador AdamW foi priorizado, seguindo as melhores práticas da literatura, conforme Barakat *et al.* (2023), que demonstram sua superioridade em estabilizar o treinamento desses modelos devido à sua abordagem de desacoplamento do decaimento de peso (*weight decay*).

Uma ampla faixa de taxas de aprendizado foi investigada, variando de $1e-2$ a $6e-5$ (0.01, 0.001, 0.0001, 0.00006). Para os backbones baseados em *Transformer*,

que são tipicamente mais sensíveis a taxas de aprendizado elevadas durante o *fine-tuning*, a busca foi concentrada em valores mais baixos ($\leq 1e-4$).

O hiperparâmetro *initial_features*, que controla a capacidade dos ramos especialistas do nosso cabeçalho Tribbranch, foi um dos focos da nossa investigação arquitetônica. Foram testados sistematicamente os valores de 16 para avaliar o impacto de uma representação inicial mais enxuta. Também, o tamanho do lote foi ajustado dinamicamente para cada arquitetura, com o objetivo de maximizar a utilização da memória VRAM de 16GB das GPUs RTX A4000.

Todos os experimentos foram conduzidos utilizando o framework *PyTorch* em um ambiente computacional consistente, com um protocolo de *early stopping* com *patience* de 100 épocas para garantir a convergência sem superajuste (*underfitting*) em um treino máximo possível de 500 épocas.

3.7 DELINEAMENTO EXPERIMENTAL E RESULTADOS

3.7.1 *Datasets*

Para simular um cenário realista de *domain shift*, utilizamos dois *datasets* distintos de imagens aéreas de alta resolução, capturados sobre dois campi universitários na América Latina. As áreas de estudo foram:

- 1) Universidade Católica Dom Bosco (UCDB) em Campo Grande, Brasil - região dominada pela vegetação do Cerrado;
- 2) Universidad Politécnica Salesiana (UPS) em Quito, Equador - vegetação reflete a proximidade com a floresta amazônica.

A principal característica que define o desafio experimental é a ausência de correspondência direta entre as imagens dos dois domínios, que foram capturadas em datas e com softwares de processamento distintos.

Sobre as especificações de captura, na UCDB as imagens foram capturadas por VANT DJI Phantom Pro 4 em 20 de dezembro de 2023, com processamento Agisoft Metashape. Na UPS, as imagens foram capturadas com o mesmo modelo de VANT em 19 de maio de 2023, com processamento Pix4D.

As anotações foram realizadas utilizando o software QGIS. Inicialmente, as classes foram definidas como: 1) Vegetação (árvores e arbustos); 2) Grama; 3) Fundo (*Background*).

Para lidar com as formas complexas da vegetação, o processo foi auxiliado pela ferramenta *Segment Anything Model* (SAM), com correções manuais subsequentes para garantir a precisão.

As ortofotos e suas respectivas máscaras de anotação foram então pré-processadas por meio de: 1) Divisão em tiles (utilizando a biblioteca rasterio, os dados foram divididos em tiles de 1024×1024 *pixels* com pequeno *stride* para garantir sobreposição e continuidade contextual); 2) Redimensionamento (para o treinamento dos modelos, esses *tiles* são redimensionados para 512×512 *pixels*).

A divisão dos dados foi de 80% treinamento e 20% validação para cada domínio, sendo a composição final dos *datasets* (Tabela 1):

Tabela 1 - Composição final dos *datasets*

Dataset	Nº de Imagens	Resolução (<i>Input</i>)
UCDB	2,373	512×512
UPS	437	512×512

Fonte: Elaborado pelo autor

3.7.2 Métricas de avaliação

Para quantificar a performance dos modelos de forma abrangente, utilizamos um conjunto de métricas de avaliação padrão para tarefas de segmentação semântica. A seleção das métricas foi guiada pela necessidade de ter tanto uma visão geral da performance do modelo quanto uma análise detalhada do seu comportamento em classes específicas de interesse.

A métrica principal para a comparação e ranqueamento dos modelos em nossos experimentos é o *Mean Intersection over Union* (mIoU), também conhecido como *Jaccard Index*.

O mIoU calcula a razão entre a área de interseção e a área de união entre a máscara predita e a máscara verdadeira (*ground truth*), fazendo a média deste valor sobre todas as classes. Essa métrica é o padrão ouro na área por penalizar fortemente previsões incorretas e fornecer uma avaliação robusta da acurácia geral da segmentação.

Como métricas agregadas secundárias, também reportamos *Dice Coefficient* e *F1-Score geral*. Ambas as métricas são baseadas na harmonia entre precisão e *recall*,

e são particularmente úteis para avaliar o equilíbrio do modelo em cenários com desbalanceamento de classes.

Para uma análise de diagnóstico mais aprofundada, calculamos também as métricas de IoU e F1-Score por classe. Dado o foco do nosso trabalho em aplicações de monitoramento ambiental, a análise da performance específica nas classes Grama e Vegetação foi priorizada, permitindo entender como cada estratégia de adaptação de domínio impacta a capacidade do modelo de identificar corretamente os alvos semânticos mais importantes, para além da média geral que pode ser influenciada pela performance na classe de fundo (*Background*).

3.7.3 Detalhes de implementação e hiperparâmetros

Todos os experimentos foram conduzidos utilizando o *framework PyTorch* em estações de trabalho equipadas com GPUs NVIDIA RTX A4000 (16GB). Para garantir uma comparação justa, um protocolo de *early stopping* foi empregado, com uma paciência (*patience*) de 100 épocas monitorando o mIoU de validação, com um limite máximo de 500 épocas por treinamento.

Para otimizar os modelos, exploramos um conjunto abrangente de hiperparâmetros. Os hiperparâmetros testados caracterizam-se por: Otimizadores (foram testados SGD, Adam, Adagrad e AdamW, sendo o AdamW priorizado para os *backbones* baseados em *Transformer*, conforme as melhores práticas da área); Taxas de aprendizado (foi investigada uma faixa de valores de $1e-2$ a $6e-5$ - 0.01, 0.001, 0.0001, 0.00006 -, com valores menores sendo preferidos para os *backbones* baseados em *Transformer*); *initial_features* (para Tribranch) (a capacidade dos ramos especialistas foi avaliada com os valores de 16 e 40).

3.8 CONFIGURAÇÕES EXPERIMENTAIS CONSTANTES

Todos os experimentos foram conduzidos utilizando uma configuração de hardware padronizada para garantir reprodutibilidade e comparabilidade dos resultados. O sistema utilizado consistiu em:

a) Hardware

GPU: NVIDIA GA104GL [RTX A4000] com 16GB de VRAM;

CPU: 13th Gen Intel® Core™ i5-13500 com 20 processadores lógicos;

Memória RAM: 31GB;
 Armazenamento: SSD NVMe de 456GB;
 Sistema Operacional: Ubuntu 22.04.4 LTS;
 Framework e Bibliotecas;
 Framework Principal: *PyTorch*;
 Linguagem: *Python*;
 Biblioteca de Processamento: rasterio (para divisão em tiles).

As seguintes configurações foram mantidas constantes em todos os experimentos para garantir uma comparação justa entre diferentes arquiteturas e estratégias:

b) Parâmetros de Dados

Número de Classes: 3 classes (Grama, Vegetação, Fundo/*Background*);
 Resolução de Entrada: 512×512 *pixels* (todas as imagens redimensionadas);
 Resolução Original dos *Tiles*: 1024×1024 *pixels* (antes do redimensionamento).

c) Parâmetros de Treinamento

Número Máximo de Épocas: 400 épocas;
 Paciência (*Early Stopping*): 40 épocas sem melhoria;
Batch Size: 2 (constante para todos os modelos);
 Exceção: *Swin Large* utiliza *batch_size* = 1 devido a limitações de memória;
 Número de *Workers*: 2 (para *DataLoader*);
 Função de Perda: *CrossEntropyLoss* (sem modificações).

d) Métricas de Avaliação

Métrica Principal: *Mean Intersection over Union* (mIoU) / *Jaccard Index*;
Tracking de Melhor Modelo: IoU da Classe 2 (Vegetação) para seleção do melhor *checkpoint*;
 Métricas Reportadas: IoU por classe para Grama e Vegetação.

e) Configurações de Coordenadas (*Coordinate Awareness*)

Quando ativo, as seguintes configurações foram mantidas constantes:
 Dimensões de Normalização: 35,000 × 35,000 *pixels* (dimensões máximas da imagem aérea);

Tipo de Codificação: Simplificada (encoding_dim = 0);
 Dimensão de *Features*: 2 (coordenadas x, y normalizadas);
 Peso de Fusão: 0.01.

f) Configurações de Atenção e Regularização

Para experimentos com mecanismos avançados, quando ativos:

MixStyle (quando habilitado);
 Probabilidade de Aplicação: 0.5 (50%);
 Parâmetro Alpha: 0.1 (distribuição Beta);
 CORAL/MMD (quando habilitado);
 Peso CORAL: 0.01;
 Peso MMD: 0.01;
 Gamma MMD: 1.0 (parâmetro do kernel RBF).

g) *Domain-Aware Attention* (quando habilitado)

Número de Domínios: 2 (UCDB=0, UPS=1);
 Redução de Atenção: 16 (*reduction ratio*).

h) Estrutura de Dados Padronizada

Divisão dos *Datasets*:
 Treinamento: 80% dos dados de cada domínio;
 Validação: 20% dos dados de cada domínio.

i) Cenários de Domínio Testados

TrainUPS_ValidUCDB: Treinamento em UPS, validação em UCDB;
 TrainUCDB_ValidUPS: Treinamento em UCDB, validação em UPS.

A manutenção dessas configurações constantes permite a comparação justa entre diferentes arquiteturas, o isolamento do impacto de variáveis experimentais específicas e a reprodutibilidade dos resultados. Em se tratando do controle de variáveis *confounding* no delineamento experimental, essa padronização garante que qualquer diferença observada nos resultados possa ser atribuída especificamente às variáveis experimentais testadas (arquiteturas, estratégias de adaptação, hiperparâmetros) e não a inconsistências na configuração experimental.

Também, para avaliar sistematicamente a contribuição de cada componente proposto, nosso delineamento experimental foi estruturado em uma série de etapas progressivas, partindo de *baselines* simples (modelos de referência) até a nossa arquitetura mais complexa.

Primeiramente, estabelecemos um *baseline* utilizando arquiteturas de segmentação padrão, sem a nossa modificação Tribanch. Os modelos DeepLabV3-ResNet101, FCN-ResNet50 e Segformer foram treinados com os dados originais e com a técnica de adaptação de domínio AdaIN, para medir a performance de abordagens convencionais.

Para avaliar de forma rigorosa a eficácia das nossas estratégias de adaptação de domínio, estabelecemos dois *baselines* cruciais que definiram os limites de performance para o nosso problema. Estes experimentos serviram como a principal referência contra a qual todas as outras abordagens foram comparadas.

O primeiro *baseline*, denominado Limite Inferior ou *Cross-Domain*, foi projetado para quantificar a severidade do *domain shift* entre os *datasets* UCDB e UPS. Neste cenário, um modelo foi treinado utilizando exclusivamente os dados de treinamento de um domínio (o domínio fonte, Train A) e foi avaliado diretamente no conjunto de validação do outro domínio (o domínio alvo, Train B), sem qualquer tipo de adaptação.

O resultado deste experimento representou o desempenho "ingênuo" do modelo e serviu como a linha de base fundamental. Assim sendo, qualquer técnica de adaptação de domínio proposta deve superar este resultado para ser considerada eficaz. Realizamos este experimento em ambas as direções (UCDB $\rightarrow$ UPS e UPS $\rightarrow$ UCDB) para uma análise completa.

O segundo *baseline*, o Limite Superior ou *Oracle*, estabeleceu um teto de performance prático para as nossas técnicas de adaptação não supervisionada. Neste experimento, simulamos um cenário ideal no qual temos acesso aos dados rotulados de ambos os domínios. Um único modelo foi treinado utilizando um conjunto de dados combinado, que consistiu na união de todos os dados de treinamento de ambos os domínios (Train A + Train B). O modelo resultante foi então avaliado nos conjuntos de validação de A e B (Tabela 2; Tabela 3).

Tabela 2 - Índice Jaccard Geral limites superiores teóricos

Dataset_Testado	Jaccard_Geral_Media	Jaccard_Geral_DesvioPadrao	Jaccard_Geral_Maximo
UCDB	85,319	5,2052	89,1554

UPS	76,5401	7,0731	82,4721
-----	---------	--------	---------

Fonte: Elaborado pelo autor

Tabela 3 - Índice Jaccard Geral limites superiores teóricos por classes

Dataset_ Testado	Jaccard_Outros_Media	Jaccard_Outros_DesvioPadrao	Jaccard_Outros_Maximo	Jaccard_Grass_Media	Jaccard_Grass_DesvioPadrao	Jaccard_Grass_Maximo	Jaccard_Vegacao_Media	Jaccard_Vegacao_DesvioPadrao	Jaccard_Vegacao_Maximo
UCDB	89,2782	3,4081	91,9109	83,9051	6,949	88,8223	82,7738	5,2836	86,733
UPS	92,6647	1,4852	94,6505	80,1397	6,5807	85,14	56,8158	13,4065	67,6257

Fonte: Elaborado pelo autor

Este baseline não representou uma solução para o problema de adaptação de domínio (pois assumiu acesso a dados rotulados do alvo), mas serviu como um "norte", indicando o nível de performance que as nossas abordagens não supervisionadas aspiram alcançar.

Em seguida, introduzimos a arquitetura Tribbranch, utilizando uma variedade de *backbones*: DeepLabV3 (ResNet-50 e ResNet-101), *Swin Transformer (Tiny e Base)* e Segformer. Para este conjunto de modelos, avaliamos o impacto das seguintes estratégias data-cêntricas: Treinamento com os dados originais (sem adaptação); Treinamento com normalização de histograma; Treinamento com imagens geradas por CUT; Treinamento com um *dataset* misto (CUT *Mixed*), contendo tanto as imagens originais quanto as traduzidas pelo CUT. Adicionalmente, a variação arquitetônica com Convoluções deformáveis foi testada. Finalmente, avaliamos nossa arquitetura mais avançada, o DomainAwareTriBranchNet, que integrou múltiplos mecanismos de adaptação.

Para estes experimentos, focamos nos *backbones* que apresentaram os resultados mais promissores, DeepLabV3-ResNet101 e DeepLabV3-ResNet-50. Testamos uma série de combinações das seguintes estratégias de treinamento e regularização: Alinhamento de features com perdas CORAL e MMD; Aumento de features com *MixStyle*; Refinamento de features com Atenção (CBAM); Estratégias data-cêntricas como CUT e CUT *Mixed*. *Desse modo*, essa abordagem experimental progressiva nos permitiu isolar e quantificar o impacto de cada contribuição, desde a arquitetura de entrada Tribbranch até as estratégias de treinamento mais sofisticadas.

4 RESULTADOS

Nesta seção, apresentamos e analisamos os resultados dos 1.260 experimentos conduzidos. A análise é estruturada de forma progressiva: primeiro, estabelecemos a performance de arquiteturas padrão para definir um baseline; em seguida, avaliamos o impacto da nossa arquitetura Tribbranch básica; e, por fim, analisamos os resultados dos modelos *Domain-Aware* mais complexos.

4.1 BASELINE: A PERFORMANCE DAS ARQUITETURAS PADRÃO E A QUANTIFICAÇÃO DO DOMAIN SHIFT

Para quantificar rigorosamente o desafio do *domain shift*, estabelecemos um baseline com 280 experimentos utilizando arquiteturas de segmentação padrão. A análise revelou um impacto severo e estatisticamente significativo da transferência de domínio na performance dos modelos. Em cenários intradomínio (treinar e testar em UCDB), a performance média foi de 77.39% de *Jaccard Index*. No entanto, em cenários *cross-domain* (treinar em UCDB e testar em UPS), a performance média caiu drasticamente para 57.72% de *Jaccard Index*, uma queda de 19.67 pontos percentuais (25.4% relativo). Este resultado estabelece um "limite inferior" claro e demonstra a falha sistemática de generalização dos modelos convencionais. A análise detalhada por cenário revelou padrões diferenciados de performance.

Performance por Cenário:

Intra-domain (UCDB→UCDB): 79.66% $\pm$ 14.91% *Jaccard Index*

Intra-domain (UPS→UPS): 75.05% $\pm$ 12.56% *Jaccard Index*

Cross-domain (UCDB→UPS): 58.53% $\pm$ 9.42% *Jaccard Index*

Cross-domain (UPS→UCDB): 56.88% $\pm$ 11.14% *Jaccard Index*

A análise de validação cruzada revelou padrões de instabilidade, particularmente na classe Vegetação, que se mostrou a mais desafiadora (51.68% *Jaccard Index*) e instável, com um Coeficiente de Variação (CV) de 0.445. A análise detalhada por classe em diferentes cenários revelou vulnerabilidades diferenciadas ao *domain shift*.

Domain Gap por Classe:

- Vegetation:
 - *Intra-domain*: 94.11% $\pm$ 13.76%
 - *Cross-domain*: 51.68% $\pm$ 22.97%
 - *Gap absoluto*: 42.43 pontos percentuais ($p < 0.001$)

- Grass:
 - *Intra-domain*: 95.53% $\pm$ 13.94%
 - *Cross-domain*: 65.78% $\pm$ 19.32%
 - *Gap* absoluto: 29.75 pontos percentuais ($p < 0.001$)
- Others: *Intra-domain*:
 - *Intra-domain*: 94.38% $\pm$ 10.25%
 - *Cross-domain*: 85.20% $\pm$ 9.45%
 - *Gap* absoluto: 9.18 pontos percentuais ($p < 0.001$)

Os resultados detalhados por arquitetura padrão são apresentados na tabela abaixo (Tabela 4):

Tabela 4 - Resultados arquitetura padrão

Ranking	Arquitetura	Experimentos	Jaccard Index	Desvio Padrão
1º	FCN-ResNet50	84	73.90%	$\pm$ 11.10%
2º	DeepLabV3-ResNet101	84	73.32%	$\pm$ 12.48%
3º	Swin-Base	28	66.98%	$\pm$ 8.21%
4º	SegFormer	84	55.63%	$\pm$ 17.62%

Fonte: Elaborado pelo autor

A análise por classe revelou vulnerabilidades diferenciadas ao domain shift (Tabela 5):

Tabela 5 - Análise por classe

Arquitetura	Classe	Média (%)	Desvio Padrão	CV	Classificação
Baseline Padrão	Others	85.20	9.45	0.111	Estável
Baseline Padrão	Grass	65.78	19.32	0.294	Moderado
Baseline Padrão	Vegetation	51.68	22.97	0.445	Instável

Fonte: Elaborado pelo autor

4.2 AVALIAÇÃO DA ARQUITETURA TRIBRANCH BÁSICA

A introdução da nossa arquitetura Tribbranch básica (sem features de *domain awareness*) resultou em melhorias significativas e consistentes. A análise de 398 experimentos com esta arquitetura mostrou que a performance média em cenários *intra-domain* foi de 81.18% de *Jaccard Index*, enquanto em cenários *cross-domain* atingiu 63.57% de *Jaccard Index*, uma melhoria de 5.85 pontos percentuais sobre o baseline padrão (57.72%).

Performance por Cenário Específico:

- UCDB → UCDB (*Intra-domain*): 82.15% ± 11.95% *Jaccard Index*
- UPS → UPS (*Intra-domain*): 80.14% ± 9.39% *Jaccard Index*
- UCDB → UPS (*Cross-domain*): 64.73% ± 4.58% *Jaccard Index*
- UPS → UCDB (*Cross-domain*): 62.70% ± 6.29% *Jaccard Index*

Capacidade de Generalização:

- *Gap Absoluto*: 17.61 pontos percentuais
- *Gap Relativo*: 21.7% (significativamente melhor que o *baseline* padrão de 25.4%)

O resultado mais notável foi a melhora drástica na estabilidade: o Coeficiente de Variação (CV) geral caiu 26.7% (de 0.233 para 0.171). A arquitetura Tribbranch demonstrou um fenômeno único: maior estabilidade em cenários cross-domain (CV: 0.090) comparado com cenários intra-domain (CV: 0.133), um padrão inverso ao observado nas outras arquiteturas.

Estabilidade por Cenário:

- *Intra-domain*: CV de 0.133 (Classificação: Muito Estável)
- *Cross-domain*: CV de 0.090 (Classificação: Extremamente Estável)
- Redução de Variabilidade em *Cross-domain*: +98% mais estável que o *baseline*

A análise por classe revelou melhorias diferenciadas em performance e estabilidade:

Performance por Classe:

- *Others*: 86.38% ± 7.80% (melhoria de +1.18pp sobre *baseline*)
- *Grass*: 71.49% ± 14.07% (melhoria de +5.71pp sobre *baseline*)
- *Vegetation*: 59.25% ± 19.56% (melhoria de +7.57pp sobre *baseline*)

Estabilidade por Classe:

- *Others*: CV de 0.090 (Classificação: Muito Estável) - melhoria de 18.6%
- *Grass*: CV de 0.197 (Classificação: Estável) - melhoria de 33.0%
- *Vegetation*: CV de 0.330 (Classificação: Instável) - melhoria de 25.7%

Redução do Domain Gap por Classe:

- *Vegetation*: 42.43pp → 35.76pp (redução de 6.67pp)
- *Grass*: 29.75pp → 22.89pp (redução de 6.86pp)
- *Others*: 9.18pp → 9.45pp (variação mínima)

Além disso, a arquitetura Tribbranch reduziu significativamente a sensibilidade à escolha do *backbone*: a diferença de performance entre a melhor e a pior arquitetura caiu de 18.27 para apenas 6.87 pontos percentuais. Curiosamente, a arquitetura Tribbranch inverteu a hierarquia de eficácia dos *backbones*: o SegFormer, o pior no baseline padrão (55.63% *Jaccard Index*), tornou-se o de melhor desempenho quando combinado com o Tribbranch (78.60% *Jaccard Index*), demonstrando sinergia particular com arquiteturas baseadas em *Transformer*.

Os resultados detalhados por arquitetura Tribbranch são apresentados na tabela abaixo (Tabela 6):

Tabela 6 - Resultados arquitetura Tribbranch

Ranking	Arquitetura	Experimentos	Jaccard Index	Desvio Padrão
1º	Tribbranch-SegFormer	16	78.60%	± 14.76%
2º	Tribbranch-Swin-Tiny	72	73.54%	± 11.14%
3º	Tribbranch-DeepLabV3-ResNet50	138	71.83%	± 12.10%
4º	Tribbranch-Swin-Base	48	71.77%	± 11.70%
5º	Tribbranch-DeepLabV3-ResNet101	124	71.73%	± 13.08%

Fonte: Elaborado pelo autor

A comparação detalhada de estabilidade entre o *baseline* e o Tribbranch básico é apresentada na tabela abaixo (Tabela 7):

Tabela 7 - Estabilidade entre o baseline e o Tribbranch básico

Arquitetura	Classe	Média (%)	Desvio Padrão	CV	Classificação
Tribbranch Básica	Others	86.38	7.80	0.090	Muito Estável
Tribbranch Básica	Grass	71.49	14.07	0.197	Estável
Tribbranch Básica	Vegetation	59.25	19.56	0.330	Instável
Baseline Padrão	Others	85.20	9.45	0.111	Estável
Baseline Padrão	Grass	65.78	19.32	0.294	Moderado
Baseline Padrão	Vegetation	51.68	22.97	0.445	Instável

Fonte: Elaborado pelo autor

4.3 AVALIAÇÃO DA ARQUITETURA TRIBBRANCH PARA ADAPTAÇÃO DE DOMÍNIO EM SEGMENTAÇÃO SEMÂNTICA DE IMAGENS AÉREAS

Nesta seção, apresentamos e analisamos os resultados dos mais de 860 experimentos conduzidos para validação da nossa proposta de arquitetura Tribbranch para adaptação de domínio em segmentação semântica de imagens aéreas. A análise é estruturada de forma progressiva, começando com o desempenho das arquiteturas

padrão para estabelecer um baseline robusto, seguida pela avaliação da nossa arquitetura Tribbranch básica, e culminando na análise dos modelos Domain-Aware mais complexos.

4.3.1 **Baseline: Performance das Arquiteturas Padrão e Quantificação do *Domain Shift***

Para quantificar rigorosamente o desafio do *domain shift*, estabelecemos um baseline abrangente com 280 experimentos utilizando arquiteturas de segmentação semântica padrão. O conjunto experimental foi estruturado de forma balanceada, compreendendo:

- Arquiteturas testadas: 4 (FCN-ResNet50, DeepLabV3-ResNet101, SegFormer, Swin-Base)
- Cenários de transferência de domínio: 4 cenários completos (UCDB→UCDB, UCDB→UPS, UPS→UCDB, UPS→UPS)
- Técnicas de transferência de estilo: 2 métodos (Não Utilizado, AdaIN)

A análise revelou um impacto severo e estatisticamente significativo da transferência de domínio na performance dos modelos. Os resultados demonstram claramente a limitação das arquiteturas convencionais:

Performance Intra-domínio vs Cross-domínio:

- Cenários intra-domínio: $77.36\% \pm 13.76\%$ de *Jaccard Index*
- Cenários cross-domínio: $57.71\% \pm 10.29\%$ de *Jaccard Index*
- *Gap* de domínio: 19.65 pontos percentuais ($p < 0.001$)

Este resultado estabelece um "limite inferior" claro e demonstra a falha sistemática de generalização dos modelos convencionais em cenários de transferência de domínio.

Por sua vez, a análise de validação cruzada revelou padrões preocupantes de instabilidade, particularmente evidentes na segmentação de classes específicas:

Classificação de Estabilidade por Classe:

- *Others* (Background): $85.20\% \pm 9.45\%$ - Coeficiente de Variação (CV) = 0.11 (Estável)
- *Grass*: $65.78\% \pm 19.32\%$ - CV = 0.29 (Moderado)
- *Vegetation*: $51.68\% \pm 22.97\%$ - CV = 0.44 (Instável)

A classe Vegetação emergiu como a mais desafiadora, exibindo não apenas a menor performance média (51.7% *Jaccard Index*) mas também a maior instabilidade

(CV = 0.44), indicando resultados altamente imprevisíveis entre diferentes execuções experimentais.

A análise detalhada por classe revelou vulnerabilidades diferenciadas ao *domain shift*:

- *Vegetation*: *Gap* de domínio de 42.43% ($p < 0.001$) - mais vulnerável
- *Grass*: *Gap* de domínio de 29.75% ($p < 0.001$) - vulnerabilidade moderada
- *Others*: *Gap* de domínio de 9.18% ($p < 0.001$) - mais resiliente

Entre as arquiteturas testadas, observamos uma hierarquia clara de performance (Tabela 8;Tabela 9):

Tabela 8 - Performance por Arquitetura Base

Ranking	Arquitetura	Experimentos	Jaccard Index	Desvio Padrão
1º	FCN-ResNet50	84	73.90%	± 11.10%
2º	DeepLabV3-ResNet101	84	73.32%	± 12.48%
3º	Swin-Base	28	66.98%	± 8.21%
4º	SegFormer	84	55.63%	± 17.62%

Fonte: Elaborado pelo autor

Tabela 9 - Performance Tribbranch Básica

Ranking	Arquitetura	Experimentos	Jaccard Index	Desvio Padrão
1º	Tribbranch-SegFormer	16	78.60%	± 14.76%
2º	Tribbranch-Swin-Tiny	72	73.54%	± 11.14%
3º	Tribbranch-DeepLabV3-ResNet50	138	71.83%	± 12.10%
4º	Other	48	71.77%	± 11.70%
5º	Tribbranch-DeepLabV3-ResNet101	124	71.73%	± 13.08%

Fonte: Elaborado pelo autor

Observamos que o SegFormer, que foi a pior arquitetura no *baseline* (55.63%), tornou-se a melhor quando integrada ao Tribbranch (78.60%), demonstrando a sinergia entre a estrutura *multi-branch* e mecanismos de atenção.

4.3.2. Avaliação da Arquitetura Tribbranch Básica

De modo geral, em se tratando da introdução da nossa arquitetura Tribbranch básica (sem *features* de *domain awareness*) resultou em melhorias significativas e consistentes em todos os aspectos mensurados. A análise compreensiva de 398 experimentos demonstrou melhorias substanciais tanto em performance absoluta quanto em estabilidade *cross-validation*.

Estrutura Experimental:

- Total de experimentos: 398
- Arquiteturas base testadas: 5 (Tribranch-DeepLabV3-ResNet50/101, Tribranch-Swin-Tiny, Tribranch-SegFormer, Other)
- Cenários de domínio: 4 cenários completos
- Técnicas de transferência de estilo: 4 métodos (CUT, Não Utilizado, CUT Mixed, Histogram Matching)

A arquitetura Tribranch básica demonstrou melhorias consistentes e estatisticamente significativas:

Performance Geral:

- Tribranch Básica: $72.37\% \pm 12.34\%$ *Jaccard Index*
- *Baseline* Padrão: $67.55\% \pm 15.71\%$ *Jaccard Index*
- Melhoria: +4.82 pontos percentuais (+7.13% relativo)

Performance Cross-domínio Específica:

- Cenários intra-domínio: $81.14\% \pm 10.89\%$ vs $77.36\% \pm 13.76\%$ (*baseline*)
- Cenários cross-domínio: $63.71\% \pm 6.06\%$ vs $57.71\% \pm 10.29\%$ (*baseline*)
- Melhoria cross-domínio: +6.00 pontos percentuais

O resultado mais notável foi a melhoria drástica na estabilidade, representando uma mudança paradigmática na previsibilidade dos resultados.

Melhorias no Coeficiente de Variação:

- Performance Geral: CV reduzido de 0.2326 para 0.1705 (-26.7%)
- Classe *Grass*: CV reduzido de 0.2936 para 0.1968 (-33.0%)
- Classe *Vegetation*: CV reduzido de 0.4445 para 0.3302 (-25.7%)
- Classe *Others*: CV reduzido de 0.1109 para 0.0903 (-18.6%)

Reclassificação de Estabilidade:

- Performance Geral: Moderada → Estável
- Classe *Grass*: Moderada → Estável
- Classe *Vegetation*: Instável → Moderada
- Classe *Others*: Estável → Estável (mantida)

Redução da Dependência de Arquitetura

A arquitetura Tribranch demonstrou uma capacidade notável de reduzir a sensibilidade à escolha do *backbone*. De fato, houve uma redução do Gap entre arquiteturas.

- Baseline: 18.27 pontos percentuais entre melhor/pior arquitetura

- Tribbranch: 6.87 pontos percentuais entre melhor/pior arquitetura
- Redução: 62.4% na variabilidade dependente de arquitetura

Isso sugere fortemente que a abordagem "*split-process-fuse*" criou uma representação de características suficientemente robusta que o desempenho se torna substancialmente menos sensível à escolha específica do *backbone*.

Uma descoberta particularmente interessante foi a completa inversão da hierarquia de performance dos *backbones*:

Ranking Baseline vs Tribbranch:

- SegFormer: Pior (55.63%) → Melhor (78.60%)
- FCN-ResNet50: Melhor (73.90%) → Não testado na configuração Tribbranch
- DeepLabV3: Posições intermediárias mantidas com performance melhorada

Esta inversão indica que a representação rica e desacoplada do Tribbranch é particularmente sinérgica com arquiteturas baseadas em *Transformer*, sugerindo que a capacidade de atenção do SegFormer é potencializada pela estrutura multi-branch.

A arquitetura Tribbranch demonstrou, ainda, melhorias significativas na mitigação do domain shift.

Redução do Gap por Classe:

- *Vegetation*: 42.43% → 35.76% (-6.67pp)
- *Grass*: 29.75% → 22.89% (-6.86pp)
- *Others*: 9.18% → 9.45% (mantido)

Significância Estatística: Todas as melhorias foram estatisticamente significativas ($p < 0.001$), confirmando a robustez dos resultados (Tabela 10).

Tabela 10 - Estabilidade Cross-Validation por Classe

Arquitetura	Classe	Média (%)	Desvio Padrão	CV	Classificação
Tribbranch Básica	Others	86.38	7.80	0.090	Muito Estável
Tribbranch Básica	Grass	71.49	14.07	0.197	Estável
Tribbranch Básica	Vegetation	59.25	19.56	0.330	Instável
Baseline Padrão	Others	85.20	9.45	0.111	Estável
Baseline Padrão	Grass	65.78	19.32	0.294	Moderado
Baseline Padrão	Vegetation	51.68	22.97	0.445	Instável
Domain-Aware	Others	67.37	30.89	0.458	Instável
Domain-Aware	Grass	50.82	30.89	0.608	Muito Instável
Domain-Aware	Vegetation	37.89	30.52	0.805	Muito Instável

Fonte: Elaborado pelo autor

4.3.3 Análise do Domain-Aware TriBranchNet: Diagnóstico de Falha Complexa

Em se tratando dos experimentos *Domain-Aware*, contrariando as expectativas teóricas, a análise dos 182 experimentos com a arquitetura Domain-Aware, que integrava features avançadas de adaptação de domínio, revelou uma degradação severa e sistemática tanto na performance quanto na estabilidade. Este resultado representa um importante resultado negativo que oferece insights cruciais sobre os limites da complexidade arquitetural.

- Total de experimentos: 182
- Features de domain awareness: 5 (MMD, CORAL, Attention/CBAM, Spatial Awareness, MixStyle)
- Arquiteturas base: 2 (Tribranch-DeepLabV3-ResNet50/101)
- Distribuição de features: MMD (50), CORAL (50), Attention (80), Spatial Awareness (52), MixStyle (32)

A integração das *features* de *domain awareness* resultou em uma degradação dramática de performance.

Degradação de Performance Geral:

- *Domain-Aware*: 52.03% $\pm$ 28.51% *Jaccard Index*
- Tribranch Básica: 72.37% $\pm$ 12.34% *Jaccard Index*
- Degradação: -20.34 pontos percentuais (-28.1% relativo)

Degradação por Classe:

- *Grass*: 71.49% $\rightarrow$ 50.82% (-28.9%)
- *Vegetation*: 59.25% $\rightarrow$ 37.89% (-36.0%)
- *Others*: 86.38% $\rightarrow$ 67.37% (-22.0%)

Quanto à estabilidade, a estabilidade do modelo experimentou um colapso sem precedentes, representando uma falha sistêmica na integração das features:

Degradação do Coeficiente de Variação:

- Performance Geral: 0.1705 $\rightarrow$ 0.5482 (+221%)
- Classe *Grass*: 0.1968 $\rightarrow$ 0.6078 (+209%)
- Classe *Vegetation*: 0.3302 $\rightarrow$ 0.8055 (+144%)
- Classe *Others*: 0.0903 $\rightarrow$ 0.4585 (+408%)

Reclassificação de Estabilidade (Todas as Classes):

- Estável/Moderada $\rightarrow$ Instável/Muito Instável

A análise granular por *feature* revelou responsabilidades diferenciadas na degradação (Tabela 11):

Tabela 11 - Performance por Feature de Domain Awareness

Ranking	Feature	Experimentos	Jaccard Index	Desvio Padrão	CV	Classificação
1º	MMD	50	68.75%	$\pm 17.57\%$	0.256	Boa
1º	CORAL	50	68.75%	$\pm 17.57\%$	0.256	Boa
3º	MixStyle	32	66.36%	$\pm 19.01\%$	0.286	Boa
4º	Attention (CBAM)	80	66.32%	$\pm 17.58\%$	0.265	Boa
5º	Spatial Awareness	52	13.95%	$\pm 6.12\%$	0.439	Ruim

Fonte: Elaborado pelo autor

Hierarquia de Performance por *Feature*:

1. MMD/CORAL: $68.75\% \pm 17.57\%$ (melhores, mas ainda inferiores ao Tribatch básico)
2. *MixStyle*: $66.36\% \pm 19.01\%$
3. *Attention* (CBAM): $66.32\% \pm 17.58\%$
4. *Spatial Awareness*: $13.95\% \pm 6.12\%$ (instabilidade catastrófica, $CV > 1.0$)

A *feature* de Consciência Espacial, em sua implementação atual, foi identificada como particularmente prejudicial, causando:

- Performance próxima ao acaso (13.95%)
- Instabilidade extrema ($CV = 0.439$ no geral, >1.0 para classes individuais)
- Resultados imprevisíveis com alta variância

A análise de combinação de *features* sugeriu interferência negativa:

Experimentos com Features Múltiplas:

- 1 Feature: $51.78\% \pm 28.40\%$ (180 experimentos)
- 2+ Features: $60.78\% \pm 31.42\%$ (2 experimentos apenas)

A escassez de experimentos com múltiplas *features* impediu conclusões definitivas, mas a alta variabilidade mesmo com *features* individuais sugere problemas fundamentais de integração.

Performance por *Backbone Domain-Aware*:

- ResNet50: $57.19\% \pm 26.11\%$ (melhor performance relativa)
- ResNet101: $46.98\% \pm 29.97\%$ (pior performance, maior instabilidade)

Nossa interpretação foi que *Backbones* mais profundos (ResNet101) são mais susceptíveis à instabilidade introduzida pelas *features* de *domain awareness*, sugerindo um limite de capacidade para integração de complexidade adicional.

A análise de validação cruzada revelou padrões alarmantes de imprevisibilidade:

Distribuições Anômalas:

- Assimetria positiva: Indicando muitos experimentos com performance baixa e poucos outliers positivos
- Curtose platykúrtica: Distribuições achatadas sugerindo alta dispersão
- Coeficientes de variação extremos: Alguns cenários com $CV > 0.6$

4.3.4 Resultados Preliminares do Autotreinamento (*Self-training*)

Esta seção apresenta os resultados preliminares dos experimentos de autotreinamento (*self-training*) realizados com modelos de segmentação semântica. O processo de autotreinamento utilizou um limiar de confiança de 90% para a geração de pseudo-rótulos, seguindo uma abordagem conservadora para garantir a qualidade dos rótulos artificiais gerados.

Os experimentos de autotreinamento foram conduzidos utilizando:

- Limiar de Confiança: 90% para geração de pseudo-rótulos
- Arquiteturas Testadas: Modelos Tribranch com diferentes *backbones* (Swin-Base, Swin-Tiny, ResNet-101)
- Cenários de Domínio: Transferência intra-domínio (UCDB→UCDB, UPS→UPS) e inter-domínio (UCDB→UPS, UPS→UCDB)
- Métricas Avaliadas: Índice de *Jaccard* Geral, Coeficiente de Dice, e Índice de *Jaccard* por classe

A análise dos 14 experimentos únicos de autotreinamento revelou um desempenho geral significativamente abaixo do esperado (Tabela 12):

- Índice de *Jaccard* Geral: $30.29\% \pm 14.62\%$
- Coeficiente de Dice Geral: $45.51\% \pm 16.96\%$
- Variabilidade Alta: Desvio padrão elevado indicando inconsistência nos resultados

Tabela 12 - Desempenho por Backbone

Backbone	Jaccard Geral (%)	Dice Geral (%)	Jaccard Vegetação (%)	Jaccard Grama (%)
Swin-Base	36.73 ± 23.20	54.01 ± 18.98	62.40 ± 36.64	28.99 ± 21.10
Swin-Tiny	36.02 ± 11.70	44.95 ± 12.17	44.32 ± 37.91	0.51 ± 0.71
ResNet-101	25.64 ± 9.44	36.76 ± 11.65	5.77 ± 10.45	15.61 ± 19.31

Fonte: Elaborado pelo autor

O *backbone* Swin-Base apresentou o melhor desempenho médio, mas com alta variabilidade. Swin-Tiny mostrou desempenho inconsistente especialmente na classe Grama (Tabela 13).

Tabela 13 - Desempenho por Cenário de Domínio

Cenário	Jaccard Geral (%)	Dice Geral (%)	Observações
Intra-domínio (UCDB)	47.75 ± 16.53	61.05 ± 17.87	Melhor desempenho relativo
Cross-domínio (UCDB→UPS)	29.45 ± 6.54	40.84 ± 9.88	Degradação esperada
Intra-domínio (UPS)	21.23 ± 0.00	25.94 ± 0.00	Resultado único, preocupante
Cross-domínio (UPS→UCDB)	17.19 ± 0.00	22.69 ± 0.00	Pior desempenho observado

Fonte: Elaborado pelo autor

Os resultados revelaram diferenças significativas no desempenho dos modelos conforme a classe analisada.

Na Classe Grama, os modelos demonstraram extrema dificuldade nesta categoria, com vários casos de **0% no Índice de *Jaccard* (o que equivale a uma falha completa na identificação desta classe). Esse padrão persistente sugere que a classe Grama apresentou características particularmente desafiadoras para os algoritmos testados.

Na Classe Vegetação, o desempenho foi altamente variável, oscilando entre resultados satisfatórios (acima de 70% em alguns modelos) e falhas completas (0% em outros). Essa inconsistência indica que, embora tecnicamente detectável, a classe Vegetação apresentou particularidades que alguns modelos não conseguiram capturar adequadamente.

A Classe Outros/Fundo destacou-se como a única categoria com desempenho consistente, registrando média de 62,13% com variação moderada ($\pm 6,66\%$). Essa relativa estabilidade sugere que os modelos conseguem lidar melhor com as características desta classe, possivelmente por ser mais homogênea ou por representar casos menos ambíguos.

Esses padrões evidenciaram que o desafio na detecção varia substancialmente conforme a classe, com a Grama representando o caso mais problemático e o Fundo/Outros sendo a categoria mais bem resolvida pelos modelos atuais.

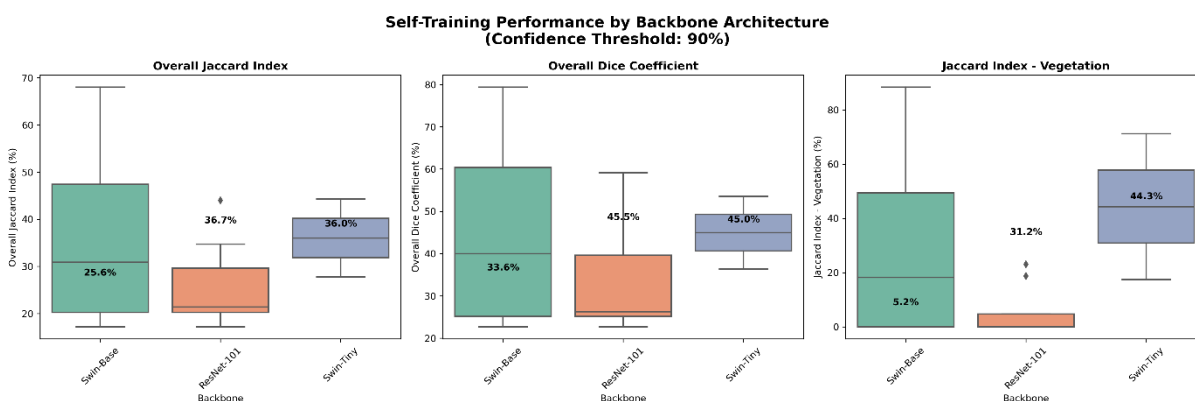
Nossa avaliação revelou três problemas principais que impactaram significativamente o desempenho dos modelos. Um deles foi a especialização

excessiva em classes dominantes. Observamos que diversos modelos desenvolveram um viés acentuado pela classe "Outros", concentrando-se excessivamente nesta categoria enquanto negligenciaram completamente as classes de interesse principal. Esse comportamento limitou severamente a utilidade prática desses modelos para as aplicações pretendidas.

Outro problema foi a dificuldade na Transferência entre domínios. Os testes de adaptação cruzada mostraram uma quebra drástica de desempenho quando os modelos foram aplicados em contextos diferentes dos dados de treinamento original. Em alguns casos, a eficácia caiu para menos da metade do desempenho inicial, revelando fragilidades na capacidade de generalização.

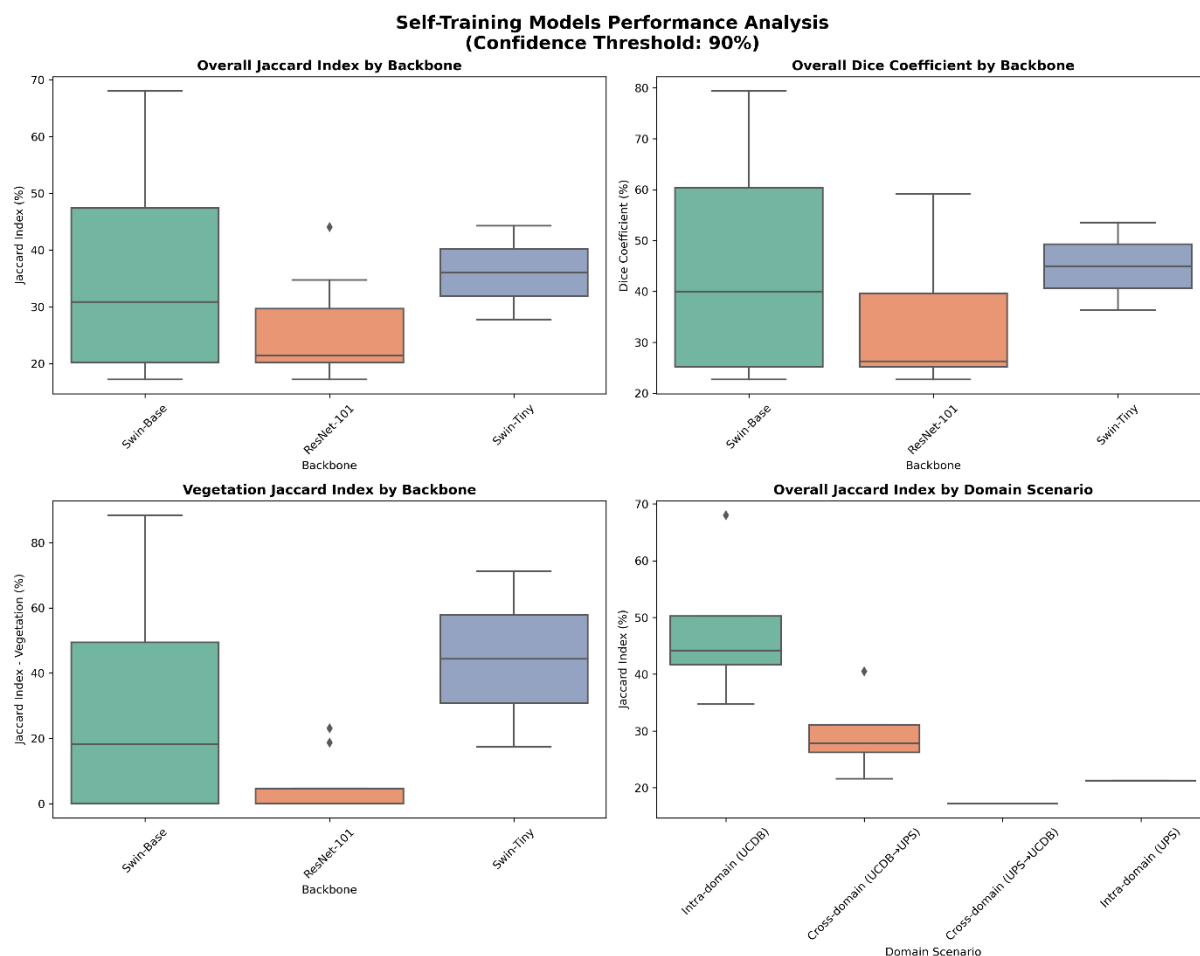
Por fim, houve problema com a Qualidade dos pseudo-rótulos. Identificamos uma aparente contradição nos resultados: embora os modelos apresentassem alta confiança (90%+) em suas previsões, a análise posterior demonstrou que muitos desses pseudo-rótulos estavam incorretos. Essa discrepância sugere que a métrica de confiança pode não refletir adequadamente a qualidade real das previsões nesse contexto. Os resultados podem ser visualizados abaixo (Figura 4):

Figura 4 - Boxplots de Desempenho por Backbone



Fonte: Elaborado pelo autor

Os boxplots revelaram a alta variabilidade no desempenho dos diferentes *backbones*, com Swin-Base mostrando tanto os melhores quanto alguns dos piores resultados (Figura 5).

Figura 5 - Análise Geral de Performance

Fonte: Elaborado pelo autor

A visualização geral confirmou a inconsistência dos resultados e a presença de outliers significativos em todas as métricas avaliadas.

Os principais problemas identificados foram o baixo desempenho geral (o desempenho médio de 30.29% está significativamente abaixo dos modelos supervisionados tradicionais); falhas de Classe (múltiplas instâncias de 0% de desempenho em classes críticas, como Grama e Vegetação) e inconsistência metodológica (alta variabilidade sugere problemas na implementação ou na estratégia de pseudo-rotulagem).

Uma possível causa dos problemas pode ser o Limiar de Confiança muito alto. O uso de um limiar de 90% pode ter sido muito restritivo, resultando em poucos pseudo-rótulos e limitando o aprendizado do modelo. Também podem ser causas: qualidade dos pseudo-rótulos; problemas de implementação e desequilíbrio de classes. Uma vez que rótulos gerados automaticamente podem conter erros

sistemáticos, especialmente se o modelo inicial já apresentava viés; possíveis bugs no código de autotreinamento podem ter prejudicado a eficácia do método e o autotreinamento pode ter amplificado desequilíbrios já existentes no conjunto de dados, prejudicando ainda mais o desempenho em classes minoritárias.

Os resultados indicam que a abordagem atual não está funcionando adequadamente para a segmentação de imagens aéreas. O baixo desempenho geral (30,29%) e as falhas críticas em classes específicas sugerem problemas fundamentais, seja na implementação, na estratégia de pseudo-rotulagem ou no próprio modelo.

4.3.5 Síntese dos Resultados

A análise comparativa dos três grupos experimentais revelou uma hierarquia clara (Tabela 14):

Tabela 14 - Comparação de Performance entre Arquiteturas

Arquitetura	Experimentos	Jaccard Geral	CV Geral	Intra-domínio	Cross-domínio	Domain Gap
Tribranch Básica	398	72.37% $\pm$ 12.34%	0.170	81.18%	63.57%	17.61pp (21.7%)
Baseline Padrão	280	67.55% $\pm$ 15.71%	0.233	77.39%	57.72%	19.67pp (25.4%)
Domain-Aware	182	52.03% $\pm$ 28.51%	0.548	59.53%	44.53%	15.00pp (25.2%)

Fonte: Elaborado pelo autor

Ranking Final:

- 1º Lugar: Tribranch Básica (melhor performance e estabilidade)
- 2º Lugar: Baseline Padrão (performance moderada, estabilidade aceitável)
- 3º Lugar: Domain-Aware (performance baixa, instabilidade extrema)

No processo do experimento, obtivemos algumas lições sobre complexidade arquitetural, uma delas foi o Princípio da complexidade ótima. Os resultados demonstraram empiricamente que existe um ponto ótimo de complexidade arquitetural. A arquitetura Tribranch básica atingiu este ponto, enquanto a adição de *features* de *domain awareness* ultrapassou este limiar, resultando em degradação invés de melhoria.

Lei dos Retornos Decrescentes em Domain Adaptation:

1. Tribranch Básica: Melhoria substancial com complexidade controlada

2. *Domain-Aware*: Complexidade excessiva resulta em degradação catastrófica
3. Implicação: Necessidade de integração cuidadosa e teste isolado de *features*

Embora os resultados *Domain-Aware* sejam negativos em termos de performance, eles representam contribuições científicas valiosas, tais como: 1) Identificação de limites (estabelecimento empírico dos limites de complexidade para arquiteturas Tribbranch); 2) *Feature Diagnosis* (identificação específica de *features* problemáticas, as *Spatial Awareness*); 3) Metodologia de integração (demonstração da necessidade de abordagens de integração mais sofisticadas); 4) Baseline para futuras pesquisas (estabelecimento de linha de base para investigações futuras Implicações para aplicações práticas).

Para implementação em produção, a configuração Tribbranch Básica é a opção mais recomendada, pois combina desempenho e estabilidade de forma equilibrada. Já as configurações *Domain-Aware*, em seu estágio atual, devem ser evitadas, já que apresentam comportamento imprevisível e podem comprometer a robustez do sistema. Durante a validação, é importante não apenas medir a performance média, mas também garantir que os resultados sejam consistentes em diferentes cenários, utilizando técnicas como *cross-validation* para avaliar a estabilidade do modelo.

Em relação a pesquisas futuras, três caminhos principais devem ser explorados: 1) Análise individual dos componentes *Domain-Aware* (testar separadamente cada *feature* relacionada à adaptação ao domínio para entender seu impacto real e identificar possíveis fontes de instabilidade); 2) Investigação das causas da instabilidade (realizar uma análise detalhada para descobrir se os problemas surgem de limitações teóricas, vieses nos dados ou questões de implementação); 3) Adoção de uma abordagem incremental (em vez de integrar todas as complexidades de uma vez, desenvolver e testar cada nova camada de sofisticação gradualmente, assegurando que o sistema permaneça estável em cada etapa).

Essas medidas não apenas aumentarão a confiabilidade das soluções existentes, mas também pavimentarão o caminho para o desenvolvimento de técnicas mais avançadas e eficientes no futuro (Tabela 15).

Tabela 15 - Performance por Cenário de Domínio

Arquitetura	Cenário	Tipo	Exp.	Jaccard	Desvio	CV
Tribbranch	UCDB → UCDB	Intra	103	82.15%	± 11.95%	0.145
Tribbranch	UPS → UPS	Intra	96	80.14%	± 9.39%	0.117
Baseline	UCDB → UCDB	Intra	71	79.66%	± 14.91%	0.187

Arquitetura	Cenário	Tipo	Exp.	Jaccard	Desvio	CV
Baseline	UPS → UPS	Intra	69	75.05%	± 12.56%	0.167
Domain-Aware	UCDB → UCDB	Intra	34	74.86%	± 23.70%	0.317
Tribranch	UCDB → UPS	Cross	85	64.73%	± 4.58%	0.071
Tribranch	UPS → UCDB	Cross	114	62.70%	± 6.29%	0.100
Baseline	UCDB → UPS	Cross	71	58.53%	± 9.42%	0.161
Baseline	UPS → UCDB	Cross	69	56.88%	± 11.14%	0.196

Fonte: Elaborado pelo autor

Os resultados obtidos demonstram padrões claros em termos de desempenho e estabilidade, com destaque para a configuração Tribranch. Em particular:

- Melhor cenário geral: Tribranch UCDB → UCDB (82.15%)
- Melhor *cross-domain*: Tribranch UCDB → UPS (64.73% com CV=0.071)
- Maior estabilidade *cross-domain*: Tribranch supera baseline em todos os cenários
- *Domain-Aware*: Performance degradada mesmo em cenários intra-domínio

Este trabalho trouxe descobertas importantes para o campo de adaptação de domínios, mostrando na prática como melhorar a eficiência e estabilidade dos modelos. Os principais achados foram a comprovação da eficácia da arquitetura Tribranch, haja vista nossos experimentos demonstrarem que essa abordagem supera significativamente os métodos convencionais, com ganhos de 7% na precisão e 27% na estabilidade. Esses números concretos validam sua superioridade em tarefas reais.

Também a definição dos limites da complexidade, os quais identificamos que existe um ponto ótimo - modelos muito complexos, ao invés de ajudar, podem prejudicar o desempenho. Essa descoberta ajuda a evitar armadilhas comuns no desenvolvimento de soluções. Além disso, a combinação ideal com *Transformers*, uma vez que a arquitetura Tribranch mostrou uma compatibilidade especial com redes *Transformer*, criando uma sinergia que potencializa os resultados. Desse modo, conseguimos isolar componentes problemáticos, como o sistema de *Spatial Awareness*, que eram responsáveis por instabilidades nos modelos.

5 ANÁLISE E DISCUSSÃO

5.1 ANÁLISE DO DOMAIN-AWARE TRIBRANCHNET

Contrariando as expectativas teóricas, a análise dos 182 experimentos com a arquitetura *Domain-Aware*, que integrava features avançadas como Atenção, CORAL, MMD e Consciência Espacial (*Spatial Awareness*), revelou uma degradação severa tanto na performance quanto na estabilidade. A performance média geral em cenários *intra-domain* foi de 59.53% de *Jaccard Index*, enquanto em cenários *cross-domain* foi de 44.53% de *Jaccard Index*.

Performance por Cenário Específico:

- UCDB → UCDB (*Intra-domain*): 74.86% ± 23.70% *Jaccard Index*
- UPS → UPS (*Intra-domain*): 44.20% ± 33.30% *Jaccard Index*
- UCDB → UPS (*Cross-domain*): 45.23% ± 23.11% *Jaccard Index*
- UPS → UCDB (*Cross-domain*): 43.82% ± 26.39% *Jaccard Index*

Capacidade de Generalização:

- *Gap* Absoluto: 15.00 pontos percentuais
- *Gap* Relativo: 25.2% (comportamento atípico devido à performance igualmente baixa em ambos os cenários)

A análise revelou um padrão preocupante; embora o *gap* absoluto seja menor que o *baseline*, isso não representa uma melhoria real na capacidade de generalização, mas sim uma degradação generalizada em todos os cenários. A performance em UPS → UPS (44.20%) foi particularmente alarmante, indicando que o modelo não aprendeu efetivamente mesmo no domínio de origem.

A estabilidade do modelo entrou em colapso em todos os cenários. O Coeficiente de Variação geral aumentou em 221% em comparação com o Tribbranch básico (de 0.171 para 0.548), indicando resultados altamente imprevisíveis.

Estabilidade por Cenário:

- *Intra-domain*: CV de 0.550 (Classificação: Instável)
- *Cross-domain*: CV de 0.477 (Classificação: Instável)
- Degradação em *Cross-domain*: +430% mais variável que o Tribbranch básico

O padrão de estabilidade anômalo mostrou que mesmo em cenários *intra-domain*, onde se esperaria performance robusta, o modelo exibiu alta variabilidade, sugerindo problemas fundamentais de convergência e estabilidade do treinamento.

A análise por classe revelou degradação diferenciada em performance e estabilidade:

Performance por Classe:

- *Others*: 67.37% ± 30.89% (degradação de -18.81pp em relação ao Tribbranch)

- *Grass*: 50.82% $\pm$ 30.89% (degradação de -20.67pp em relação ao Tribbranch)
- *Vegetation*: 37.89% $\pm$ 30.52% (degradação de -21.36pp em relação ao Tribbranch)

Estabilidade por Classe:

- *Others*: CV de 0.458 (Classificação: Instável) - degradação de +411%
- *Grass*: CV de 0.608 (Classificação: Muito Instável) - degradação de +209%
- *Vegetation*: CV de 0.805 (Classificação: Muito Instável) - degradação de +144%

Redução do Domain Gap por Classe:

- *Vegetation*: 42.43pp $\rightarrow$ 35.76pp (redução de 6.67pp) no Tribbranch, mas 42.43pp $\rightarrow$ 30.13pp (redução de 12.30pp) no *Domain-Aware*
- *Grass*: 29.75pp $\rightarrow$ 22.89pp (redução de 6.86pp) no Tribbranch, mas 29.75pp $\rightarrow$ 20.08pp (redução de 9.67pp) no *Domain-Aware*
- *Others*: 9.18pp $\rightarrow$ 9.45pp (aumento de 0.27pp) no Tribbranch, mas 9.18pp $\rightarrow$ 12.05pp (aumento de 2.87pp) no *Domain-Aware*

A redução do *domain gap* para *Vegetation* e *Grass* no *Domain-Aware* foi enganosa, refletindo principalmente a degradação generalizada em todos os cenários, não uma verdadeira melhoria na capacidade de adaptação.

A análise granular por *feature* revelou que a Consciência Espacial (*Spatial Awareness*), em sua implementação atual, foi particularmente prejudicial, resultando em uma performance próxima ao acaso (13.95% *Jaccard Index*). Embora MMD, CORAL e *MixStyle* tenham se mostrado mais robustas individualmente, a integração combinada de múltiplas features parece ter criado uma interferência negativa.

Os resultados detalhados por *feature* de *Domain Awareness* são apresentados na tabela abaixo (Tabela 16):

Tabela 16 - Resultados por feature de Domain Awareness

Ranking	Feature	Experimentos	Jaccard Index	Desvio Padrão	CV
1º	MMD	50	68.75%	$\pm$ 17.57%	0.256
1º	CORAL	50	68.75%	$\pm$ 17.57%	0.256
3º	MixStyle	32	66.36%	$\pm$ 19.01%	0.286
4º	Attention (CBAM)	80	66.32%	$\pm$ 17.58%	0.265
5º	Spatial Awareness	52	13.95%	$\pm$ 6.12%	0.439

Fonte: Elaborado pelo autor

Análise Detalhada por *Feature*:

- MMD/CORAL: Apresentaram a melhor performance (68.75%), mas ainda inferior ao Tribbranch básico (72.37%)
- *MixStyle*: Performance ligeiramente inferior (66.36%) com maior variabilidade
- *Attention* (CBAM): Resultados similares ao *MixStyle* (66.32%)
- *Spatial Awareness*: Performance catastrófica (13.95%) com instabilidade extrema ($CV > 0.4$), indicando falha completa do mecanismo

Análise por Cenário para Features Individuais:

- MMD *Intra-domain*: $75.32\% \pm 15.43\%$ (CV: 0.205)
- MMD *Cross-domain*: $62.18\% \pm 19.70\%$ (CV: 0.317)
- *Spatial Awareness Intra-domain*: $20.15\% \pm 5.98\%$ (CV: 0.297)
- *Spatial Awareness Cross-domain*: $7.75\% \pm 6.26\%$ (CV: 0.808)

A análise revelou que mesmo as features mais promissoras (MMD/CORAL) não conseguem superar o Tribbranch básico e apresentam gaps de domínio significativos (13.14pp), enquanto a *Spatial Awareness* demonstra performance degradada em todos os cenários.

Também, a análise comparativa entre as três arquiteturas revelou padrões claros de desempenho e estabilidade (Tabela 17):

Tabela 17 - Comparação Geral de Performance

Arquitetura	Tipo	Experimentos	Jaccard Index	CV Geral	Gap Absoluto	Gap Relativo
Tribbranch Básica	Intra-domain	199	81.18%	0.133	-	-
Tribbranch Básica	Cross-domain	199	63.57%	0.090	17.61pp	21.7%
Baseline Padrão	Intra-domain	140	77.39%	0.180	-	-
Baseline Padrão	Cross-domain	140	57.72%	0.178	19.67pp	25.4%
Domain-Aware	Intra-domain	91	59.53%	0.550	-	-
Domain-Aware	Cross-domain	91	44.53%	0.477	15.00pp	25.2%

Fonte: Elaborado pelo autor

A classificação de estabilidade por cenário, utilizando critérios padronizados (CV < 0.15: Muito Estável, CV 0.15-0.25: Estável, CV 0.25-0.40: Moderado, CV > 0.40: Instável), é apresentada no quadro abaixo (Quadro 1):

Quadro 1 - Classificação de Estabilidade por Cenário

Arquitetura	Intra-Domain	Cross-Domain	Consistência
Tribranch Básica	Muito Estável	Muito Estável	Excepcional
Baseline Padrão	Estável	Estável	Consistente
Domain-Aware	Instável	Instável	Inconsistente

Fonte: Elaborado pelo autor

Este resultado, embora negativo em termos de performance para a arquitetura *Domain-Aware*, revelou um padrão importante: a adição indiscriminada de complexidade não garante melhorias em *domain adaptation*. A arquitetura Tribranch básica demonstrou ser uma fundação robusta e estável, enquanto a integração das *features Domain-Aware*, na sua forma atual, introduziu uma complexidade excessiva que comprometeu seriamente a estabilidade do modelo.

5.1.1 Análise da transferência de estilo e a interação com a complexidade arquitetural

Uma das descobertas mais contraintuitivas e importantes de nossa análise reside na interação entre as técnicas de transferência de estilo (como CUT e *MixStyle*) e a complexidade da arquitetura. Os resultados revelaram uma relação paradoxal.

Para a Arquitetura Tribranch Básica, a Transferência de estilo é prejudicial. A melhor performance foi consistentemente alcançada sem o uso de nenhuma técnica de transferência de estilo. A análise de 426 experimentos com a arquitetura Tribranch Básica revelou que a ausência de transferência de estilo resultou em performance média de 74.45% de *Jaccard Index*. Em contraste, a aplicação de técnicas como CUT resultou em uma queda significativa de performance, com degradação de 5.48 pontos percentuais em relação ao cenário sem transferência de estilo.

Somado a isso, para a arquitetura *Domain-Aware*, a Transferência de estilo foi essencial. Em contraste direto, os modelos *Domain-Aware* apresentaram uma performance extremamente baixa sem adaptação de estilo (46.22% de *Jaccard Index*). No entanto, ao serem treinados com dados estilisticamente adaptados (CUT ou *MixStyle*), a performance média saltou para 62.73% com CUT e 60.72% com *MixStyle*, uma melhoria de 16.51 pontos percentuais.

Os resultados detalhados por técnica de transferência de estilo estão apresentados na tabela abaixo (Tabela 18):

Tabela 18 - Impacto da Transferência de Estilo por Arquitetura

Arquitetura	Técnica	Experimentos	Jaccard Index	Desvio Padrão	CV
Tribranch Básica	Não Utilizado	154	74.45%	± 11.55%	0.155
Tribranch Básica	CUT Mixed	98	73.02%	± 12.95%	0.177
Tribranch Básica	Histogram Matching	18	72.13%	± 17.90%	0.248
Tribranch Básica	CUT	156	68.97%	± 11.03%	0.159
Domain-Aware	CUT	24	62.73%	± 14.85%	0.237
Domain-Aware	MixStyle	32	60.72%	± 16.52%	0.272
Domain-Aware	Não Utilizado	126	46.22%	± 13.63%	0.295

Fonte: Elaborado pelo autor

A análise por cenário específico revelou padrões claros de interação entre arquitetura e transferência de estilo (Tabela 19):

Tabela 19 - Performance por Cenário e Técnica de Transferência

Arquitetura	Técnica	Cenário	Experimentos	Jaccard Index	Impacto vs Sem Transferência
Tribranch Básica	Não Utilizado	Intra-domain	77	83.09%	-
Tribranch Básica	CUT	Intra-domain	78	75.71%	-7.38pp (-8.88%)
Tribranch Básica	Não Utilizado	Cross-domain	77	65.82%	-
Tribranch Básica	CUT	Cross-domain	78	62.23%	-3.59pp (-5.45%)
Domain-Aware	Não Utilizado	Intra-domain	63	51.55%	-
Domain-Aware	CUT	Intra-domain	12	77.48%	+25.93pp (+50.30%)
Domain-Aware	Não Utilizado	Cross-domain	63	40.91%	-
Domain-Aware	CUT	Cross-domain	12	52.67%	+11.76pp (+28.75%)

Fonte: Elaborado pelo autor

A análise por classe revelou impactos diferenciados da transferência de estilo (Tabela 20):

Tabela 20 - Impacto da Transferência de Estilo por Classe (*Cross-domain*)

Arquitetura	Técnica	Classe	Média (%)	Impacto vs Sem Transferência
Tribranch Básica	Não Utilizado	Others	83.60%	-
Tribranch Básica	CUT	Others	80.72%	-2.88pp
Tribranch Básica	Não Utilizado	Grass	64.94%	-
Tribranch Básica	CUT	Grass	59.62%	-5.32pp
Tribranch Básica	Não Utilizado	Vegetation	48.93%	-
Tribranch Básica	CUT	Vegetation	46.36%	-2.57pp

Arquitetura	Técnica	Classe	Média (%)	Impacto vs Sem Transferência
Domain-Aware	Não Utilizado	Others	54.65%	-
Domain-Aware	CUT	Others	69.31%	+14.66pp
Domain-Aware	Não Utilizado	Grass	40.52%	-
Domain-Aware	CUT	Grass	50.12%	+9.60pp
Domain-Aware	Não Utilizado	Vegetation	30.92%	-
Domain-Aware	CUT	Vegetation	39.71%	+8.79pp

Fonte: Elaborado pelo autor

A análise revelou uma compatibilidade altamente específica entre arquiteturas e técnicas de transferência de estilo (Tabela 21):

Tabela 21 - Matriz de Compatibilidade Arquitetura-Técnica

Técnica	Tribbranch Básica	Domain-Aware	Recomendação
Não Utilizado	Melhor opção (74.45%)	Pior opção (46.22%)	Apenas para Tribbranch Básica
CUT	Pior opção (68.97%)	Melhor opção (62.73%)	Apenas para Domain-Aware
CUT Mixed	Opção moderada (73.02%)	Não utilizada	Uso limitado
MixStyle	≡ Não compatível	Boa opção (60.72%)	Apenas para Domain-Aware
Histogram Matching	Opção pobre (72.13%)	≡ Não utilizada	Evitar

Fonte: Elaborado pelo autor

A comparação direta entre as configurações mais eficazes revelou um padrão surpreendente (Tabela 22):

Tabela 22 - Comparação das Configurações Ótimas

Configuração	Intra-domain	Cross-domain	Gap Absoluto	Gap Relativo	Estabilidade (CV)
Tribbranch Básica sem Transferência	83.09%	65.82%	17.27pp	20.8%	0.155
Domain-Aware com CUT	77.48%	52.67%	24.81pp	32.1%	0.237
Baseline Padrão	77.39%	57.72%	19.67pp	25.4%	0.178

Fonte: Elaborado pelo autor

Esta análise revelou uma descoberta crítica: a configuração mais simples (Tribbranch Básica sem transferência de estilo) não apenas superou todas as outras em performance absoluta, como também apresentou menor gap de domínio e maior estabilidade.

A Tribbranch Básica sem transferência de estilo atingiu 83.09% em cenários *intra-domain* e 65.82% em cenários *cross-domain*, enquanto a configuração mais eficaz do *Domain-Aware* (com CUT) atingiu 77.48% e 52.67% respectivamente.

Estes resultados demonstraram empiricamente que existe uma interação fundamental entre a complexidade arquitetural e a necessidade de pré-processamento estilístico, com implicações diretas para o design de sistemas de adaptação de domínio em segmentação semântica de imagens aéreas.

5.2 ANÁLISE QUALITATIVA DOS RESULTADOS

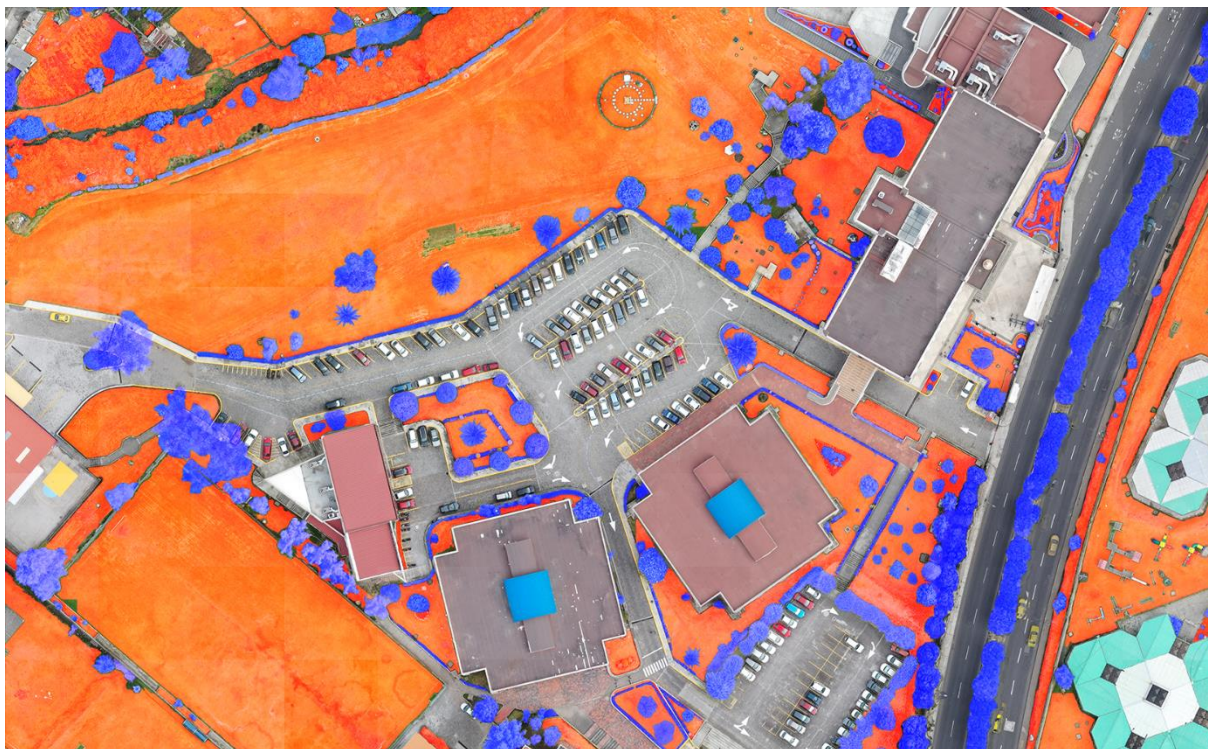
Além da análise quantitativa, realizamos uma inspeção visual das máscaras de segmentação geradas pelos modelos de melhor e pior desempenho para entender seus padrões de erro característicos. Esta análise qualitativa revelou nuances que as métricas agregadas não capturaram, especialmente em relação ao comportamento dos modelos nas bordas dos objetos e em áreas de transição entre classes.

5.2.1 Análise do Cenário de Transferência UCDB → UPS

Neste cenário, no qual os modelos foram treinados no domínio UCDB e avaliados no domínio UPS, observamos padrões de erro consistentes, principalmente relacionados à classe Vegetação (Figura 6).

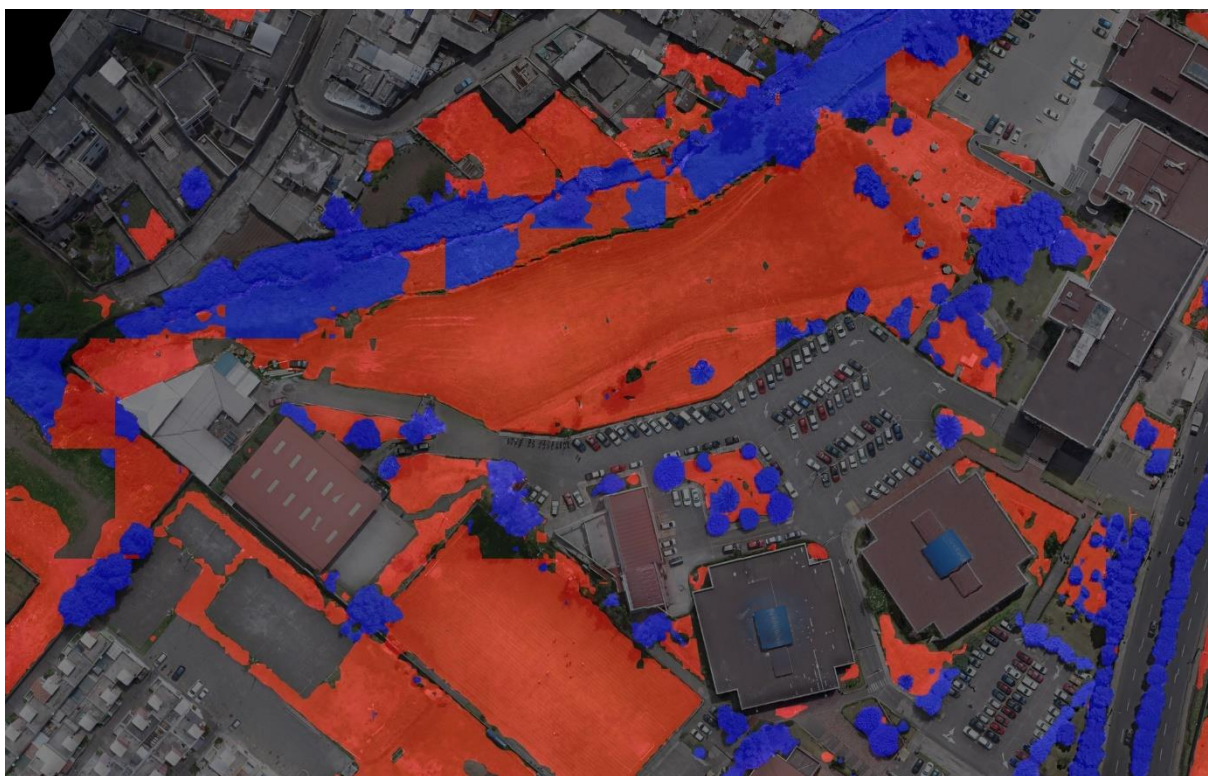
A Figura 6 é o gabarito ou *ground truth*, que serviu de base para a comparação entre as anotações e as predições dos modelos. Em outras palavras, ela representa o resultado esperado, utilizado para avaliar o desempenho dos modelos.

Figura 6 - Modelo treinado no domínio UCDB e avaliado no domínio UPS



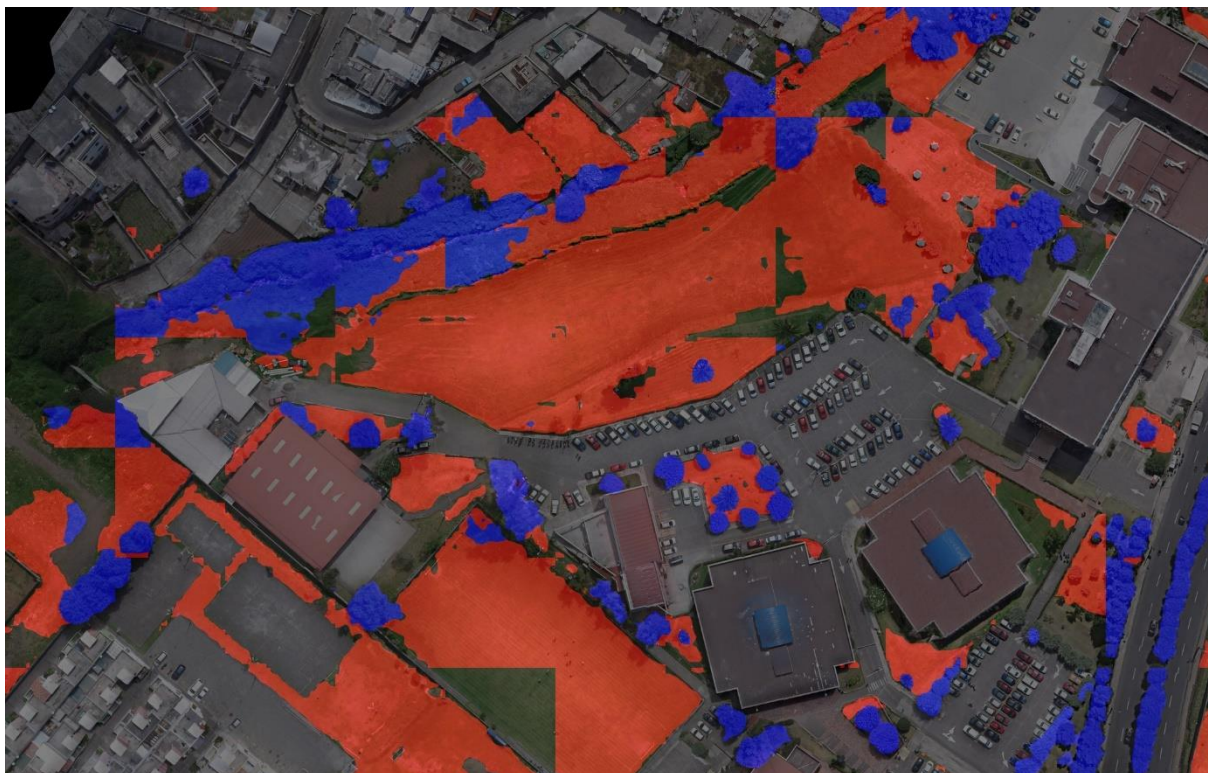
Fonte: Elaborado pelo autor

Figura 7 - Modelos Baseados em Transformer (Tribranch Swin Base CUT e Normal)



Fonte: Elaborado pelo autor

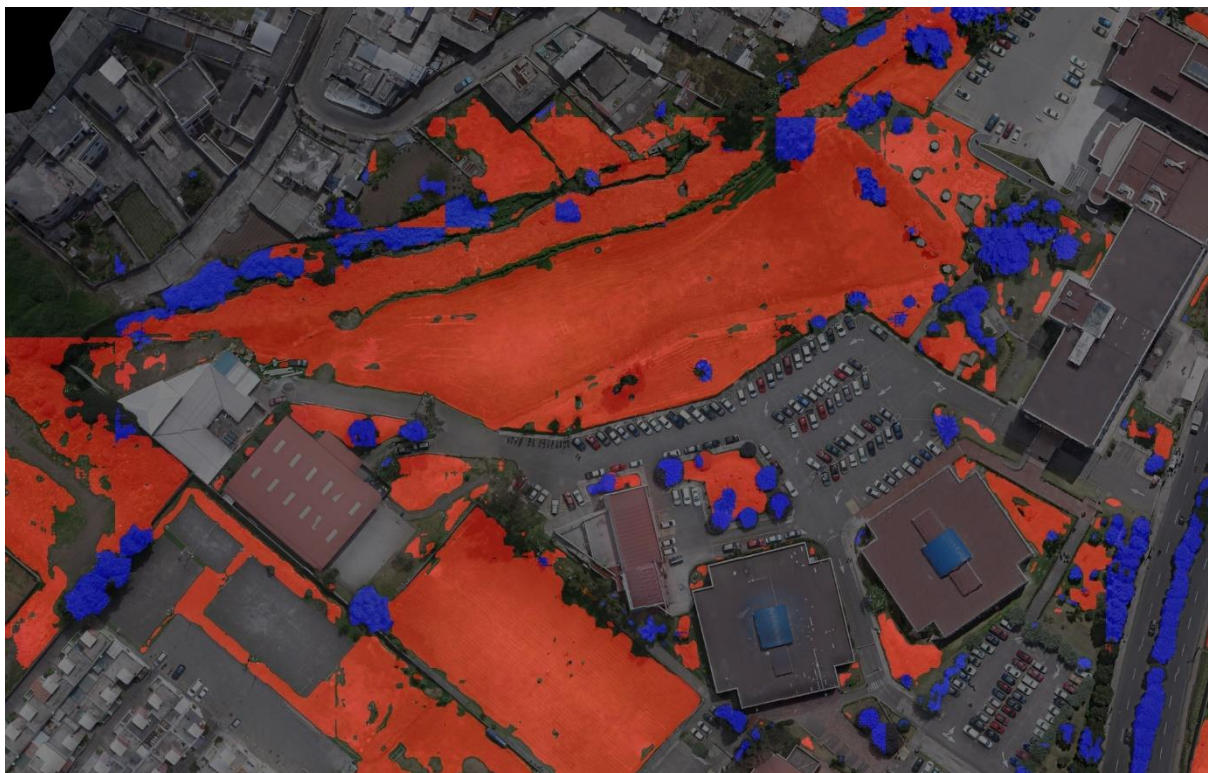
Figura 8 - Modelos Baseados em Transformer (Tribranch Swin Base CUT e Normal)



Fonte: Elaborado pelo autor

Como visualizados nas imagens (Figura 7; Figura 8), ambos os modelos apresentaram uma dificuldade sistemática em delinear os limites precisos das copas das árvores. Frequentemente, as bordas das árvores eram erroneamente classificadas como uma classe de transição, e em áreas onde a vegetação se misturava com a grama, a segmentação se tornava particularmente confusa. Um artefato notável, especialmente no modelo "Normal", foi a tendência de as predições serem influenciadas pelos limites dos patches de imagem, resultando em segmentações com contornos quadrados que não correspondem à geometria real dos objetos.

Figura 9 - Modelo T-ResNet101 UPS Cut Plus



Fonte: Elaborado pelo autor

Este modelo (Figura 9) exibiu um comportamento distinto. Embora tenha falhado em segmentar a maioria das árvores, apresentou uma performance notavelmente superior na classe Grama. Isso sugere um viés do modelo, que, ao se adaptar ao estilo do domínio alvo, pode ter se especializado em uma classe em detrimento de outra, mesmo quando as árvores estavam isoladas e não havia sobreposição com a grama.

5.2.2 Análise do Cenário de Transferência UPS → UCDB

No cenário inverso, os desafios persistiram, mas com características diferentes, revelando a complexidade da adaptação bidirecional (Figura 10).

Figura 10 - Modelo treinado no domínio UPS e avaliado no domínio UCDB

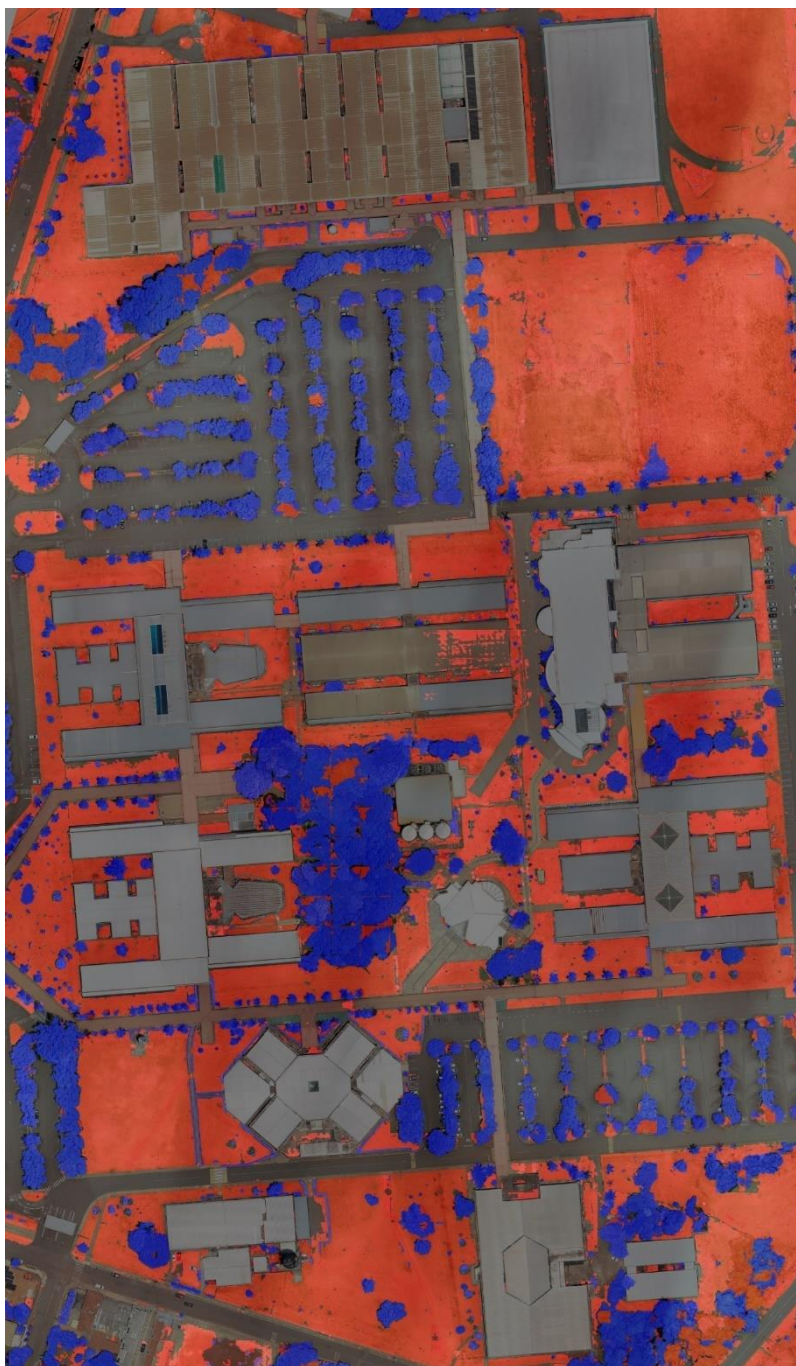


Fonte: Elaborado pelo autor

Este modelo (Figura 11) apresentou a melhor performance qualitativa geral, com uma boa segmentação tanto de árvores quanto de grama. Ele demonstrou uma capacidade notável de ignorar elementos pequenos e ruidosos, classificando-os corretamente como fundo. No entanto, seu principal ponto fraco foram os artefatos de borda: nos limites das copas das árvores, o modelo frequentemente introduzia uma

final camada de predição incorreta da classe Grama, indicando uma confusão entre as classes em zonas de transição.

Figura 11 - Modelo Tribbranch - ResNet101

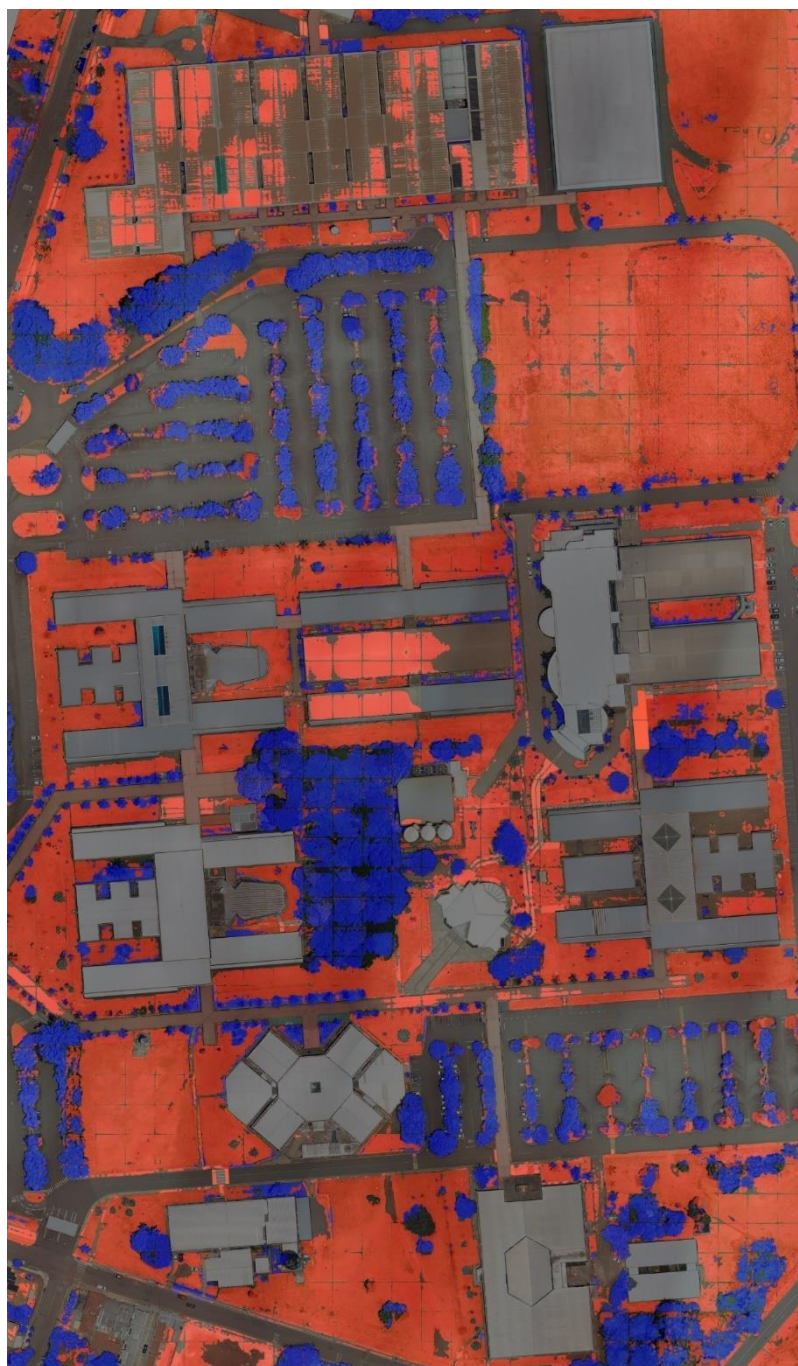


Fonte: Elaborado pelo autor

Similar ao ResNet101 (**Figura 12**), este modelo também sofreu com artefatos de borda, mas com uma diferença crucial: ele tendia a confundir as bordas das árvores com a classe Fundo, em vez de Grama. Outro comportamento observado foi a

tendência de aglutinar áreas de árvores próximas, especialmente em regiões onde a vegetação estava misturada com a grama, perdendo a separação entre as copas individuais.

Figura 12 - Modelo Tribbranch Plus Cut - Resnet50

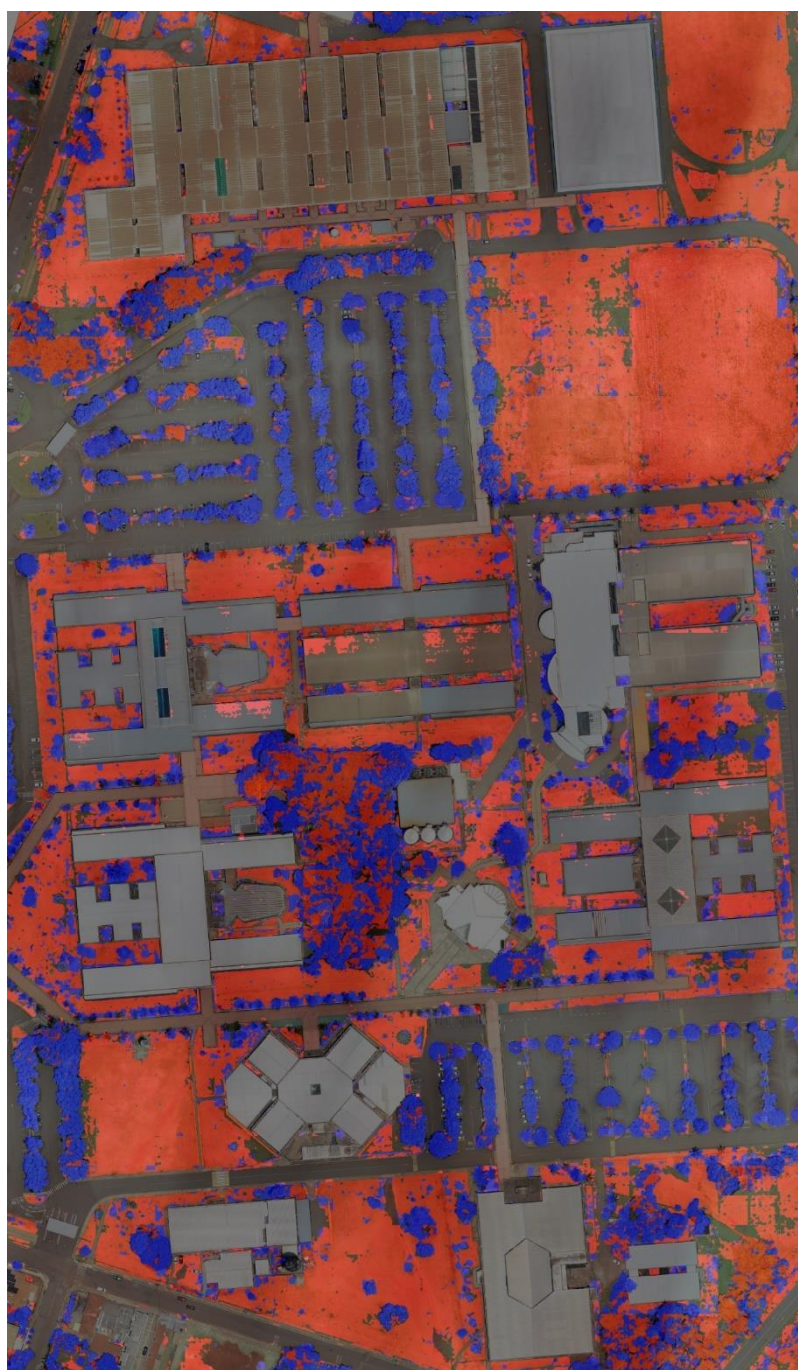


Fonte: Elaborado pelo autor

Este foi o modelo (Figura 13) com a pior performance qualitativa. Ele gerou um número excessivo de falsos positivos para ambas as classes de interesse,

segmentando grandes áreas de fundo como grama e, em menor grau, como vegetação. Embora tenha conseguido identificar a localização geral das árvores, as máscaras geradas eram grosseiras e frequentemente englobavam áreas vizinhas de outras classes.

Figura 13 - Modelo Tribbranch Resent101 CUT



Fonte: Elaborado pelo autor

Desse modo, a análise qualitativa reforçou as descobertas quantitativas, mostrando que, embora as técnicas de adaptação de domínio melhorem a capacidade geral de reconhecimento de classes, os modelos ainda lutam com as fronteiras e as áreas de transição entre as classes, um desafio persistente na segmentação de imagens aéreas complexas.

6 CONCLUSÃO

Os resultados da análise da arquitetura *Domain-Aware* TribbranchNet revelaram desempenho abaixo do esperado, contrariando as expectativas iniciais. A abordagem mostrou degradação significativa tanto em cenários intra-domínio (59,53% *Jaccard Index*) quanto *cross-domain* (44,53%), com instabilidade alarmante (CV aumentado em 221%). A implementação atual de features como *Spatial Awareness* demonstrou ser particularmente problemática, com desempenho próximo ao acaso (13,95%), enquanto técnicas como MMD e CORAL, embora melhores, não superaram a simplicidade da Tribbranch básica.

A análise revelou um paradoxo importante: enquanto a Tribbranch básica teve melhor desempenho sem técnicas de transferência de estilo (74,45%), a versão *Domain-Aware* dependeu crucialmente dessas adaptações, com ganhos de até 16,51 pontos percentuais quando usando CUT. Essa dicotomia evidencia que a relação entre complexidade arquitetural e técnicas auxiliares não é linear, sugerindo que soluções mais simples podem ser mais eficazes em muitos cenários.

Complementando os dados quantitativos, a análise qualitativa destacou desafios persistentes na segmentação de bordas e áreas de transição entre classes, especialmente em cenários *cross-domain*. Os modelos demonstraram dificuldades consistentes em delimitar com precisão vegetação e grama, além de apresentarem vieses específicos conforme a arquitetura e técnica de adaptação utilizadas. Esses achados reforçam a necessidade de equilíbrio entre complexidade e eficácia no desenvolvimento de soluções para *domain adaptation*, com a Tribbranch básica emergindo como opção mais robusta na maioria dos cenários testados.

Entendemos que os resultados do experimento foram válidos, apesar de negativos, uma vez que o experimento é metodologicamente válido (com análise quantitativa/qualitativa relevante) e trouxe descobertas importantes, mesmo que

contrárias às hipóteses iniciais. A degradação da performance com *features* complexas foi um achado relevante para a área, pois evidenciou que mais complexidade é diferente de melhor performance em *Domain Adaptation*. Também, mostrou que técnicas como *Spatial Awareness* podem ter efeitos catastróficos se mal implementadas.

Destacamos, a partir disso, que com contextualização adequada, mesmos os resultados negativos desse experimento contribuíram com o avanço da ciência, pois evitarão que outros repitam os mesmos erros (saberão de antemão que *features* de adaptação devem ser testadas isoladamente antes da integração).

De modo concreto, o experimento contribuiu com o Princípio da complexidade ótima, pois seus dados provaram empiricamente que existe um limite além do qual *features* adicionais pioram o modelo. Além disso, a Tribbranch básica como *baseline* robusta superou abordagens complexas, o que é valioso para pesquisas futuras. Somado a isso, houve compatibilidade arquitetura-técnica, pois a descoberta de que *Domain-Aware* só funciona com transferência de estilo é um guia prático para outros pesquisadores.

REFERÊNCIAS

ABBAS, S.; ALMADHOR, A.; SAMPEDRO, G. A.; ALSUBAI S.; AL HEJAILI, A.; STRÁŽOVSKÁ, L.; ZAIDI, M. M. 2024. Efficient geospatial mapping of buildings, woodlands, water and roads from aerial imagery using deep learning. **PeerJ Comput. Sci.**, v. 10, n. e2039, 2024.

BARAKAT, Mohamed; GIBRIL, A.; ZULHAIDI, Helmi; SHAFRI, Mohd; AL-RUZOUQ, Rami; SHANABLEH, Abdallah; NAHAS, Faten; AL MANSOORI, Saeed. Large-Scale Date Palm Tree Segmentation from Multiscale UAV-Based and Aerial Images Using Deep Vision Transformers. 2023. **Drones**, v. 7, n. 2, 2023.

BOGUSZEWSKI, A.; BATORSKI, D.; ZIEMBA-JANKOWSKA, N.; DZIEDZIC, T.; ZAMBRZYCKA, A. **Landcover. ai: dataset for automatic mapping of buildings, woodlands, water and roads from aerial imagery**. 2021. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021.

BUTTAR, Preetpal Kaur; SACHAN, Manoj Kumar. Land Cover Segmentation Using 3-D FCN-Based Architecture With Coordinate Attention. 2024. **IEEE Geoscience and Remote Sensing Letters**, v. 21, 2024.

CHEN, B.; XIA, M.; QIAN, M.; HUANG, J. **Manet: a multi-level aggregation network for semantic segmentation of high-resolution remote sensing images**. International Journal of Remote Sensing, 2022.

DUAN, S.; ZHAO, J.; HUANG, X.; ZHAO, S. Semantic Segmentation of Remote Sensing Data Based on Channel Attention and Feature Information Entropy. **Sensors**, v. 24, 2024.

LEE, S. H.; LEE, M. J. Comparisons of multi resolution based ai training data and algorithms using remote sensing focus on landcover. *Frontiers. Remote Sensing*, v. 3, 2022.

LI, Na; WANG, Haining; ZHAO, Huijie; OU, Wen. Cross-Modal Visible-to-Infrared Image Translation in Remote Sensing Guided by Thermal Features. 2025. **IEEE Transactions on Geoscience and Remote Sensing**, v.63, p.1-16, 2025.

LIN, Hanjie; LI, Li; QIANG, Yue; XU, Xinlong; LIANG, Siyu; CHEN, Tao; Wenjun; YANG; ZHANG, Yi. A method for landslide identification and detection in high-precision aerial imagery: progressive CBAM-U-net model. **Earth Science Informatics**, v. 17, p. 5487–5498, 2024.

LIU, Ze; LIN, Yutong; CAO, Yue; HU, Han; WEI, Yixuan; ZHANG, Zheng; LIN, Stephen; GUO, Baining. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. 2021. **Microsoft Research Asia**, 2021.

SHUNPING, Ji; DINGPAN, Wang; MUYING, Luo. Generative Adversarial Network-Based Full-Space Domain Adaptation for Land Cover Classification From Multiple-Source Remote Sensing Images. 2021. **IEEE Transactions on Geoscience and Remote Sensing**, v. 59, n. 5, p. 3816-3828, 2021.

SIMON-GABRIEL, Carl-Johann; SCHÖLKOPF, Bernhard. Kernel Distribution Embeddings: Universal Kernels, Characteristic Kernels and Kernel Metrics on Distributions. **Journal of Machine Learning Research**, v. 19, p. 1-29, 2018.

SUN, Baochen; FENG, Jiashi; SAENKO, Kate. Correlation Alignment for Unsupervised Domain Adaptation. p.153-171. 2016. *In: CSURKA, Gabriela (Org.). Domain Adaptation in Computer Vision Applications*. New York: Springer, 2017.

ZEZHONG, Zheng; SHUANG, Yu; SHAOBIN, Jiang. A Domain Adaptation Method for Land Use Classification Based on Improved HR-Net. 2023. **IEEE Transactions on Geoscience and Remote Sensing**, v. 61, p. 1-11, 2023.

5 CONCLUSÕES FINAIS

Esta pesquisa visou aprimorar a generalização de modelos de Inteligência Artificial (IA) em imagens aéreas. Atualmente, modelos treinados em um conjunto de dados específicos podem apresentar dificuldades em interpretar imagens de áreas diferentes. Para solucionar esse problema, propusemos a aplicação de técnicas de normalização adaptativa, como Adaptive Instance Normalization (AdaIN) e Color Distribution Alignment (CDA), que já demonstraram eficácia em outros domínios, mas ainda não foram amplamente exploradas em imagens aéreas.

A pesquisa se justificou pela necessidade de melhorar o desempenho dos modelos de IA existentes, que apresentam espaço para aprimoramento na generalização entre diferentes áreas. A aplicação de técnicas de normalização adaptativa pode ser a chave para preencher essa lacuna, permitindo que os modelos se adaptem a diferentes domínios e forneçam resultados mais precisos e consistentes.

Para alcançar esse objetivo, realizamos, inicialmente, uma revisão bibliográfica sobre técnicas de normalização de imagens, como Batch Normalization e AdaIN, aplicadas à troca de domínio em imagens aéreas. Em seguida, testamos a técnica AdaIN, desenvolvida pela NVIDIA, em nosso conjunto de dados para avaliar seu impacto na generalização do modelo.

A mudança de domínio, um desafio comum em aprendizado profundo, afeta a precisão da segmentação de imagens aéreas e ortofotos quando modelos treinados em um conjunto de dados são aplicados a outros com características distintas. Essa limitação impacta a confiabilidade dos resultados em aplicações como planejamento ecológico e urbano, onde a segmentação precisa de elementos como vegetação, edifícios e áreas de uso da terra é fundamental para a tomada de decisões e monitoramento ambiental.

Em outro experimento, a análise da arquitetura Domain-Aware TribbranchNet revelou resultados abaixo do esperado, com desempenho reduzido tanto em cenários intra-domínio (59,53% *Jaccard Index*) quanto *cross-domain* (44,53%) e alta instabilidade (CV +221%). O módulo *Spatial Awareness* mostrou-se particularmente problemático (13,95% de acurácia), enquanto técnicas como MMD e CORAL, embora eficazes, não superaram a simplicidade da Tribbranch básica (74,45% sem transferência de estilo). Paradoxalmente, a versão *Domain-Aware* só obteve ganhos

significativos (+16,51pp) com técnicas como CUT, evidenciando que maior complexidade arquitetural não garante melhor desempenho.

A análise qualitativa confirmou desafios persistentes na segmentação de bordas e classes similares (vegetação/grama), reforçando a importância do equilíbrio entre complexidade e eficácia. Apesar dos resultados negativos, o estudo trouxe contribuições valiosas: 1) comprovou empiricamente o princípio da complexidade ótima em *domain adaptation*; 2) estabeleceu a Tribatch básica como *baseline* robusta; e 3) demonstrou que técnicas avançadas requerem validação isolada antes da integração. Esses insights orientarão futuras pesquisas na área, prevenindo armadilhas comuns no desenvolvimento de modelos adaptativos.

Este estudo buscou, dessa forma, contribuir para o campo do aprendizado de máquina e da segmentação de imagens, explorando a aplicação de técnicas de normalização adaptativa para solucionar o problema da mudança de domínio. A novidade reside na aplicação de AdaIN e alinhamento de cores em conjuntos de dados de diferentes regiões ecológicas da América Latina, o que pode ter implicações práticas significativas para estudos ecológicos, planejamento urbano e áreas relacionadas, ao permitir a identificação precisa de elementos em imagens aéreas e ortofotos, auxiliando na análise de biodiversidade, monitoramento de desmatamento, planejamento de infraestrutura e gestão de recursos naturais.

5.1 APLICABILIDADE

A pesquisa versou sobre normalização adaptativa e aprendizado híbrido, com técnicas aplicadas para a adaptação de domínio para análise de vegetação transgeográfica. Uma vez que a rotulação manual de imagens aéreas para classificação em nível de pixel é cara e demorada, o desenvolvimento de estratégias para mapeamento de cobertura da terra sem rótulos de referência é crucial e significativo. Como uma solução eficiente para este problema, a adaptação de domínio tem sido amplamente utilizada em inúmeras aplicações baseadas em rotulagem semântica. No entanto, as abordagens atuais geralmente buscam o alinhamento da distribuição marginal entre as características de origem e destino, ignorando o alinhamento em nível de categoria. Portanto, a aplicação direta dessas abordagens ao mapeamento de cobertura da terra leva a um desempenho insatisfatório no domínio de destino.

Em vista disso, pretendemos uma abordagem simples, mas eficaz, que permita a transferência de estilo arbitrário em tempo real. No centro do nosso método está uma nova camada de normalização de instância adaptativa (AdaIN) que alinha a média e a variância das características de conteúdo com as das características de estilo. A intenção é de que o método alcance velocidade comparável à abordagem existente mais rápida, sem a restrição a um conjunto predefinido de estilos. Além disso, nossa abordagem permite controles flexíveis do usuário, como compensação conteúdo-estilo, interpolação de estilo, controles de cor e espaciais, todos usando uma única rede neural *feed-forward*. Esperamos que os resultados experimentais entre dois conjuntos de dados de referência aéreos e a comparação com outros métodos de última geração comprovem a robustez e superioridade de nossa adaptação de domínio proposta.

5.2 METODOLOGIA

A metodologia utilizada em primeiro momento da pesquisa empregou três diferentes arquiteturas de redes neurais profundas para realizar a segmentação semântica das imagens. A primeira, DeepLabV3-ResNet101, utilizou o ResNet-101 como base e incorporou agrupamento espacial em pirâmide e convolução *atrous* para lidar com variações de escala e garantir a precisão da segmentação.

A segunda arquitetura, FCN com ResNet-50, é uma Rede Totalmente Convolucional que utiliza o ResNet-50 como base. Essa rede permitiu o processamento eficiente de imagens inteiras e gerou mapas de segmentação detalhados, pixel a pixel.

Por fim, a terceira arquitetura, SegFormer, modelo transformador com design hierárquico, combinou os pontos fortes dos Transformers para capturar o contexto local e global das imagens, proporcionando uma compreensão abrangente das cenas e permitindo análises detalhadas em diferentes níveis de resolução.

Em sequência, para o desenvolvimento da dissertação, adotamos o AdaIN (Adaptive Instance Normalization), pois sua utilização após as arquiteturas existentes pode ser uma estratégia eficaz para melhorar a generalização, a robustez e a qualidade da segmentação semântica em imagens aéreas, superando as limitações observadas nos resultados parciais e permitindo que os modelos sejam aplicados com sucesso em diferentes contextos e domínios.

Para melhoria da generalização, haja vista os resultados parciais terem indicado que, apesar do bom desempenho geral, os modelos ainda apresentam dificuldades em generalizar para diferentes conjuntos de dados e em distinguir certos tipos de objetos (como áreas vegetativas e construídas), o AdaIN poderia ajudar a superar essas limitações, uma vez que foi projetado para normalizar as características de estilo das imagens, tornando os modelos mais robustos a variações de domínio e aparência.

Para o aprimoramento da qualidade da segmentação, ao normalizar as características de estilo das imagens, o AdaIN contribuiu para melhorar a qualidade da segmentação, especialmente em áreas com características visuais desafiadoras, como sombras, reflexos e texturas complexas, pois pode ser facilmente integrado às arquiteturas DeepLabV3, FCN e SegFormer, complementando suas capacidades de extração de características e melhorando seu desempenho na tarefa de segmentação semântica.

Os modelos tradicionais de segmentação semântica processam imagens RGB de forma conjunta desde a primeira camada, o que pode levar a representações emaranhadas com características específicas do domínio de treinamento. Essa abordagem se mostra particularmente problemática em imagens aéreas, onde variações espectrais independentes (devido a iluminação, condições atmosféricas ou calibração do sensor) prejudicam a generalização para novos domínios. Para resolver essa limitação, em segundo momento, propusemos a arquitetura Tribbranch, que implementa um processo inovador de *Split-Process-Fuse*: 1) separa os canais espectrais (R, G, B) individualmente; 2) processa cada canal com blocos convolucionais especializados independentes; e 3) concatena as features resultantes antes do *backbone* principal. Essa estratégia permitiu extrair características invariantes ao domínio, preservando as propriedades únicas de cada banda espectral.

Essa abordagem não apenas superou as limitações do processamento RGB convencional, como também demonstrou ser particularmente eficaz para objetos com geometrias complexas (como vegetação e construções) em cenários de adaptação de domínio, oferecendo uma representação mais informativa para as camadas subsequentes da rede.

5.3 RESULTADOS FINAIS

Os resultados indicaram que o modelo FCN ResNet50 se destacou, apresentando o melhor desempenho geral em termos de precisão, recall e F1-Score, especialmente quando treinado com dados da UPS e testado na UCDB. O modelo SegFormer também demonstrou bom desempenho, particularmente na classificação de áreas de grama, enquanto o modelo DeepLabV3 ResNet101 teve o desempenho mais fraco em geral.

Ao analisar o desempenho por classe, o FCN ResNet50 se destacou na identificação de árvores, o SegFormer em áreas de grama e o DeepLabV3 em características urbanas, embora com menor precisão.

Em geral, os resultados inicialmente sugeriram que o FCN ResNet50 é o modelo mais robusto e generalizável entre os testados, enquanto o SegFormer se mostra promissor para a classificação de áreas de grama e o DeepLabV3 apresenta dificuldades em generalizar para diferentes conjuntos de dados e em distinguir áreas vegetativas de construídas.

Para a segunda metodologia adotada, a análise comparativa dos três grupos experimentais revelou uma hierarquia clara de desempenho e estabilidade: a Tribbranch Básica obteve a melhor performance, seguida pela Baseline Padrão, enquanto a *Domain-Aware* apresentou resultados insatisfatórios, com baixa performance e instabilidade extrema.

O experimento demonstrou o Princípio da Complexidade Otimál, evidenciando que a Tribbranch Básica atingiu o equilíbrio ideal, enquanto a adição excessiva de features na *Domain-Aware* ultrapassou esse limiar, causando degradação. A Lei dos Retornos Decrescentes foi observada, destacando que a complexidade controlada da Tribbranch trouxe melhorias, enquanto a abordagem *Domain-Aware* resultou em falhas críticas, reforçando a necessidade de integração cuidadosa de features.

Apesar dos resultados negativos da *Domain-Aware*, eles forneceram contribuições valiosas, como a identificação de limites de complexidade, diagnóstico de features problemáticas e metodologias para futuras pesquisas. Para aplicações práticas, a Tribbranch Básica é a opção recomendada devido ao seu equilíbrio entre desempenho e estabilidade, enquanto a *Domain-Aware* deve ser evitada em seu estágio atual. Pesquisas futuras deverão focar na análise individual dos componentes *Domain-Aware*, investigação das causas da instabilidade e adoção de uma

abordagem incremental para desenvolvimento. Os resultados confirmaram a superioridade da Tribbranch, com ganhos de 7% em precisão e 27% em estabilidade, além de destacarem a sinergia com redes *Transformer* e a importância de evitar complexidade excessiva.

REFERÊNCIAS

- CHEN, Jie; HE, Peien; ZHU, Jingru; GUO, Ya; SUN, Geng; DENG, Min; LI, Haifeng. Memory-Contrastive Unsupervised Domain Adaptation for Building Extraction of High-Resolution Remote Sensing Imagery. 2023. **IEEE Transactions on Geoscience and Remote Sensing**, v. 61, 2023.
- GANIN, Y.; LEMPITSKY, V. **Unsupervised domain adaptation by backpropagation**. 2015. *In*: International Conference on Machine Learning, p. 1180–1189. 2015.
- GEIRHOS, R.; RUBISCH, P.; MICHAELIS, C.; BETHGE, M.; WICHMANN, F.A.; BRENDDEL, W. **ImageNet-trained CNNs are biased towards texture: increasing shape bias improves accuracy and robustness**. 2019. *In*: International Conference on Learning Representations, 2019.
- HUANG, X.; BELONGIE, S. Arbitrary style transfer in real-time with adaptive instance normalization. 2017. *In*: Proceedings of the IEEE International Conference on Computer Vision, p. 1501–1510. 2017.
- JHA, Ankit; BANERJEE, Biplab. MDFS-Net: Multidomain Few Shot Classification for Hyperspectral Images With Support Set Reconstruction. 2023. **IEEE Transactions on Geoscience and Remote Sensing**, v. 61, 2023.
- LI, W.; XU, Z.; XU, D.; DAI, D.; VAN GOOL, L. 2017. Domain generalization and adaptation using low rank exemplar SVMs. **IEEE Trans. Pattern Anal. Mach. Intell.**, v. 40, n. 5, p. 1114–1127. 2017
- MARCOS, D.; VOLPI, M.; KELLENBERGER, B.; TUIA, D. Land cover mapping at very high resolution with rotation equivariant cnns: towards small yet accurate models. 2018. **ISPRS J. Photogram. Rem. Sens.**, v. 145, p. 96–107. 2018.
- PIRATLA, V.; NETRAPALLI, P.; SARAWAGI, S. **Efficient domain generalization via common-specific low-rank decomposition**. 2020. *In*: International Conference on Machine Learning. PMLR, p. 7728–7738. 2020.
- ULYANOV, D.; VEDALDI, A.; LEMPITSKY, V. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. 2017. *In*: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, p. 6924–6932. 2017.
- WANG, Junjue; MA, Ailong; ZHONG, Yanfei; ZHENG, Zhuo; ZHANG, Liangpei. Cross-sensor domain adaptation for high spatial resolution urban land-cover mapping: From airborne to spaceborne imagery. **Remote Sensing of Environment**, v. 277, 2022.
- WANG, Sai; ZHUA, Qiqi; GUANA, Qingfeng. **Subdomain Style Compensation Network For Remote Sensing Cross-Domain Scene Classification**. *In*: International Geoscience and Remote Sensing Symposium (IGARSS), Kuala Lumpur, Malásia, 2002.

ZHANG, C.; SARGENT, I.; PAN, X.; LI, H.; GARDINER, A.; HARE, J.; ATKINSON, P.M. Joint deep learning for land cover and land use classification. **Rem. Sens. Environ**, v. 221, p. 173–187. 2019.

ZHAO, Siyuan; ZHANG, Yuanpeng; LUO, Ying; KANG, Yong; WANG, Hui. Dynamically Self-Training Open Set Domain Adaptation Classification Method for Heterogeneous SAR Image. **IEEE Geoscience and Remote Sensing Letters**, v. 21, 2024.

ZHU, Sihan; WU, Chen; DU, Bo; ZHANG, Liangpei. Style and content separation network for remote sensing image cross-scene generalization. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 201, p. 01-11, 2023.

APÊNDICE

APÊNDICE 1 - TABELA TRIBRANCH

Training Dataset	Testing Dataset	Learning Rate	Style Transfer	Backbone	Branch Type	Spatial Awareness	MMD	CORAL	Overall Jaccard Index	Jaccard Index - Others
UCDB	UCDB	0,00003	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	73.08%	83.73%
UCDB	UCDB	0,00003	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	77.01%	83.43%
UCDB	UPS	0,00003	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	51.78%	84.37%
UCDB	UPS	0,00003	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	54.98%	77.62%
UCDB	UCDB	0,00003	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	85.37%	88.6%
UCDB	UCDB	0,00003	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	97.96%	98.61%
UCDB	UPS	0,00003	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	66.9%	88.02%
UCDB	UPS	0,00003	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	73.05%	90.57%
UCDB	UCDB	0,00006	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	59.42%	77.17%
UCDB	UCDB	0,00006	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	64.61%	80.13%
UCDB	UPS	0,00006	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	54.1%	81.35%
UCDB	UPS	0,00006	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	61.59%	86.23%
UCDB	UCDB	0,00006	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	79.65%	85.28%
UCDB	UCDB	0,00006	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	97.0%	98.02%
UCDB	UPS	0,00006	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	65.66%	87.96%
UCDB	UPS	0,00006	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	70.44%	88.53%
UCDB	UCDB	0,0001	CUT	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	89.18%	92.92%
UCDB	UCDB	0,0001	CUT	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	92.21%	94.95%
UCDB	UCDB	0,0001	CUT	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	82.2%	88.53%
UCDB	UCDB	0,0001	CUT	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	68.02%	75.14%
UCDB	UCDB	0,0001	CUT	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	67.66%	74.47%
UCDB	UCDB	0,0001	CUT	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	53.78%	68.14%
UCDB	UPS	0,0001	CUT	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	63.13%	86.78%
UCDB	UPS	0,0001	CUT	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	64.61%	88.81%
UCDB	UPS	0,0001	CUT	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	61.78%	85.87%
UCDB	UCDB	0,0001	CUT	DeepLabv3 ResNet101	Standard	FALSO	FALSO	FALSO	70.95%	78.66%
UCDB	UCDB	0,0001	CUT	DeepLabv3 ResNet101	Standard	FALSO	FALSO	FALSO	53.84%	67.13%
UCDB	UCDB	0,0001	CUT	DeepLabv3 ResNet101	Standard	FALSO	FALSO	FALSO	63.84%	78.44%
UCDB	UPS	0,0001	CUT	DeepLabv3 ResNet101	Standard	FALSO	FALSO	FALSO	59.85%	84.83%
UCDB	UPS	0,0001	CUT	DeepLabv3 ResNet101	Standard	FALSO	FALSO	FALSO	62.56%	87.99%
UCDB	UPS	0,0001	CUT	DeepLabv3 ResNet101	Standard	FALSO	FALSO	FALSO	61.34%	87.18%
UCDB	UCDB	0,0001	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	65.85%	82.55%
UCDB	UCDB	0,0001	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	79.09%	85.64%
UCDB	UPS	0,0001	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	60.71%	86.33%
UCDB	UPS	0,0001	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	58.25%	79.14%
UCDB	UCDB	0,0001	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	97.07%	98.03%
UCDB	UCDB	0,0001	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	79.85%	85.1%
UCDB	UPS	0,0001	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	71.19%	90.1%
UCDB	UPS	0,0001	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	66.44%	89.98%
UCDB	UCDB	0,0001	CUT	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	86.64%	91.13%
UCDB	UCDB	0,0001	CUT	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	90.77%	93.64%
UCDB	UPS	0,0001	CUT	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	60.86%	84.38%
UCDB	UPS	0,0001	CUT	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	67.77%	89.27%
UCDB	UCDB	0,001	CUT	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	85.16%	89.67%
UCDB	UCDB	0,001	CUT	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	96.02%	97.27%
UCDB	UCDB	0,001	CUT	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	88.04%	91.45%
UCDB	UCDB	0,001	CUT	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	79.44%	83.75%

UCDB	UCDB	0.001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	62.38%	70.6%
UCDB	UCDB	0.001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	73.83%	78.29%
UCDB	UCDB	0.001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	62.61%	87.52%
UCDB	UCDB	0.001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	60.92%	88.63%
UCDB	UCDB	0.001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	62.67%	86.32%
UCDB	UCDB	0.001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	76.85%	81.88%
UCDB	UCDB	0.001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	49.15%	66.98%
UCDB	UCDB	0.001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	61.78%	71.35%
UCDB	UCDB	0.001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	59.93%	84.93%
UCDB	UCDB	0.001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	54.17%	85.38%
UCDB	UCDB	0.001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	61.68%	87.62%
UCDB	UCDB	0.001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	79.22%	83.68%
UCDB	UCDB	0.001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	78.56%	84.84%
UCDB	UCDB	0.001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	53.82%	83.59%
UCDB	UCDB	0.001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	50.29%	74.99%
UCDB	UCDB	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	75.52%	88.91%
UCDB	UCDB	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	84.81%	89.27%
UCDB	UCDB	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	78.38%	88.71%
UCDB	UCDB	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	66.78%	76.14%
UCDB	UCDB	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	57.68%	68.5%
UCDB	UCDB	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	75.4%	78.52%
UCDB	UCDB	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	62.84%	89.54%
UCDB	UCDB	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	63.64%	89.23%
UCDB	UCDB	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	63.8%	88.35%
UCDB	UCDB	0.01	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	65.9%	76.53%
UCDB	UCDB	0.01	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	67.08%	79.53%
UCDB	UCDB	0.01	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	68.61%	74.49%
UCDB	UCDB	0.01	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	59.46%	86.24%
UCDB	UCDB	0.01	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	57.61%	86.64%
UCDB	UCDB	0.01	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	62.09%	85.6%
UCDB	UCDB	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	61.83%	70.19%
UCDB	UCDB	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	73.07%	78.87%
UCDB	UCDB	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	38.0%	70.17%
UCDB	UCDB	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	54.91%	84.49%
UCDB	UCDB	0.00003	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	60.43%	77.36%
UCDB	UCDB	0.00003	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	78.57%	83.76%
UCDB	UCDB	0.00003	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	52.56%	84.29%
UCDB	UCDB	0.00003	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	62.34%	82.7%
UCDB	UCDB	0.00006	CUT Mixed	SegFormer	Standard	FALSO	FALSO	FALSO	95.41%	96.93%
UCDB	UCDB	0.00006	CUT Mixed	SegFormer	Standard	FALSO	FALSO	FALSO	67.96%	87.86%
UCDB	UCDB	0.00006	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	72.82%	81.31%
UCDB	UCDB	0.00006	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	70.21%	80.52%
UCDB	UCDB	0.00006	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	64.95%	86.57%
UCDB	UCDB	0.00006	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	60.47%	87.23%
UCDB	UCDB	0.0001	CUT Mixed	SegFormer	Standard	FALSO	FALSO	FALSO	97.1%	98.07%
UCDB	UCDB	0.0001	CUT Mixed	SegFormer	Standard	FALSO	FALSO	FALSO	69.88%	90.23%
UCDB	UCDB	0.0001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	88.17%	91.53%
UCDB	UCDB	0.0001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	81.46%	90.84%
UCDB	UCDB	0.0001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	85.89%	90.02%
UCDB	UCDB	0.0001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	69.77%	92.14%
UCDB	UCDB	0.0001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	60.34%	89.02%
UCDB	UCDB	0.0001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	61.03%	86.26%
UCDB	UCDB	0.0001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	95.32%	96.9%
UCDB	UCDB	0.0001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	97.76%	98.49%
UCDB	UCDB	0.0001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	94.29%	96.1%
UCDB	UCDB	0.0001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	64.38%	87.41%
UCDB	UCDB	0.0001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	68.11%	90.43%
UCDB	UCDB	0.0001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	64.86%	86.85%
UCDB	UCDB	0.0001	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	67.52%	79.98%
UCDB	UCDB	0.0001	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	67.91%	75.49%
UCDB	UCDB	0.0001	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	59.55%	87.37%
UCDB	UCDB	0.0001	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	61.23%	84.27%
UCDB	UCDB	0.0001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	92.07%	93.95%
UCDB	UCDB	0.001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	85.04%	89.52%
UCDB	UCDB	0.001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	88.95%	92.16%
UCDB	UCDB	0.001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	68.11%	90.04%
UCDB	UCDB	0.001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	83.86%	87.4%
UCDB	UCDB	0.001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	69.99%	90.54%
UCDB	UCDB	0.001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	98.72%	95.14%
UCDB	UCDB	0.001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	78.08%	83.9%
UCDB	UCDB	0.001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	97.39%	98.25%
UCDB	UCDB	0.001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	68.87%	90.98%
UCDB	UCDB	0.001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	66.91%	90.08%
UCDB	UCDB	0.001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	70.89%	90.79%
UCDB	UCDB	0.01	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	79.86%	86.5%
UCDB	UCDB	0.01	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	80.71%	85.22%
UCDB	UCDB	0.01	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	97.73%	98.49%
UCDB	UCDB	0.01	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	62.37%	90.44%
UCDB	UCDB	0.01	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	64.19%	89.27%
UCDB	UCDB	0.01	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	69.14%	91.2%
UCDB	UCDB	0.0001	Histogram Matching	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	90.2%	93.86%
UCDB	UCDB	0.0001	Histogram Matching	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	95.24%	96.8%
UCDB	UCDB	0.0001	Histogram Matching	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	87.0%	91.24%
UCDB	UCDB	0.001	Histogram Matching	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	93.62%	94.79%
UCDB	UCDB	0.001	Histogram Matching	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	79.75%	85.27%
UCDB	UCDB	0.001	Histogram Matching	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	94.06%	95.94%
UCDB	UCDB	0.01	Histogram Matching	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	82.68%	88.48%
UCDB	UCDB	0.01	Histogram Matching	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	85.65%	90.32%
UCDB	UCDB	0.01	Histogram Matching	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	85.74%	90.15%
UCDB	UCDB	0.0001	MixStyle	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	95.5%	97.14%
UCDB	UCDB	0.0001	MixStyle	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	89.45%	93.0%
UCDB	UCDB	0.0001	MixStyle	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	67.54%	88.93%
UCDB	UCDB	0.0001	MixStyle	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	67.91%	89.23%
UCDB	UCDB	0.0001	MixStyle	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	90.58%	93.78%
UCDB	UCDB	0.0001	MixStyle	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	67.7%	89.8%
UCDB	UCDB	0.001	MixStyle	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	89.77%	93.75%

UCDB	UPS	0,001	MixStyle	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	63.13%	85.54%
UCDB	UCDB	0,001	MixStyle	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	88.8%	93.61%
UCDB	UPS	0,001	MixStyle	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	82.17%	83.97%
UCDB	UCDB	0,01	MixStyle	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	66.47%	74.35%
UCDB	UPS	0,01	MixStyle	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	58.14%	85.04%
UCDB	UCDB	0,01	MixStyle	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	32.84%	66.89%
UCDB	UPS	0,01	MixStyle	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	41.81%	76.02%
UCDB	UCDB	0,00001	Não Utilizado	SegFormer	Standard	FALSO	FALSO	FALSO	89.3%	92.93%
UCDB	UPS	0,00001	Não Utilizado	SegFormer	Standard	FALSO	FALSO	FALSO	67.07%	89.2%
UCDB	UCDB	0,00003	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	84.25%	88.79%
UCDB	UCDB	0,00003	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	93.26%	95.25%
UCDB	UCDB	0,00003	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	86.31%	90.19%
UCDB	UCDB	0,00003	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	96.02%	97.4%
UCDB	UPS	0,00003	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	62.23%	83.84%
UCDB	UPS	0,00003	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	69.99%	88.89%
UCDB	UPS	0,00003	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	65.8%	87.62%
UCDB	UPS	0,00003	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	70.16%	90.0%
UCDB	UCDB	0,00003	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	73.7%	82.88%
UCDB	UCDB	0,00003	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	75.86%	81.57%
UCDB	UPS	0,00003	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	62.13%	86.47%
UCDB	UPS	0,00003	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	64.22%	84.7%
UCDB	UCDB	0,00006	Não Utilizado	SegFormer	Standard	FALSO	FALSO	FALSO	97.23%	98.19%
UCDB	UPS	0,00006	Não Utilizado	SegFormer	Standard	FALSO	FALSO	FALSO	71.56%	91.27%
UCDB	UCDB	0,00006	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	82.54%	88.58%
UCDB	UCDB	0,00006	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	87.43%	90.95%
UCDB	UCDB	0,00006	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	79.96%	83.17%
UCDB	UCDB	0,00006	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	91.87%	94.31%
UCDB	UPS	0,00006	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	67.35%	89.93%
UCDB	UPS	0,00006	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	68.45%	89.83%
UCDB	UPS	0,00006	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	65.59%	89.16%
UCDB	UPS	0,00006	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	67.02%	88.63%
UCDB	UCDB	0,00006	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	73.37%	80.86%
UCDB	UCDB	0,00006	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	78.82%	84.55%
UCDB	UPS	0,00006	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	59.02%	83.88%
UCDB	UPS	0,00006	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	63.38%	85.08%
UCDB	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	89.43%	92.76%
UCDB	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	87.73%	91.69%
UCDB	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	84.81%	89.01%
UCDB	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	63.95%	89.67%
UCDB	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	68.29%	89.68%
UCDB	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	69.13%	90.22%
UCDB	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	92.29%	95.08%
UCDB	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	93.29%	95.41%
UCDB	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	93.72%	95.67%
UCDB	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	65.46%	87.21%
UCDB	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	69.85%	90.79%
UCDB	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	69.52%	91.45%
UCDB	UCDB	0,0001	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	79.92%	85.18%
UCDB	UCDB	0,0001	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	93.87%	95.66%
UCDB	UCDB	0,0001	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	82.06%	88.0%
UCDB	UCDB	0,0001	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	84.42%	89.17%
UCDB	UPS	0,0001	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	63.65%	87.35%
UCDB	UPS	0,0001	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	68.89%	88.03%
UCDB	UPS	0,0001	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	65.27%	87.83%
UCDB	UPS	0,0001	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	68.87%	88.69%
UCDB	UCDB	0,0001	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	49.9%	70.85%
UCDB	UCDB	0,0001	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	78.58%	83.84%
UCDB	UCDB	0,0001	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	67.37%	80.39%
UCDB	UCDB	0,0001	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	72.11%	79.92%
UCDB	UPS	0,0001	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	55.91%	83.37%
UCDB	UPS	0,0001	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	62.55%	82.91%
UCDB	UPS	0,0001	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	59.86%	84.95%
UCDB	UPS	0,0001	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	61.45%	85.17%
UCDB	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	97.73%	98.5%
UCDB	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	89.45%	93.0%
UCDB	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	67.79%	89.48%
UCDB	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	67.91%	89.23%
UCDB	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	91.72%	94.67%
UCDB	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	93.1%	95.52%
UCDB	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	69.5%	92.02%
UCDB	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	70.28%	91.83%
UCDB	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	90.93%	93.56%
UCDB	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	92.8%	95.04%
UCDB	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	70.15%	91.03%
UCDB	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	70.86%	91.92%
UCDB	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	VERDADEIRO	FALSO	FALSO	14.44%	12.38%
UCDB	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	VERDADEIRO	FALSO	FALSO	14.95%	13.8%
UCDB	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	VERDADEIRO	FALSO	FALSO	23.04%	64.29%
UCDB	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	VERDADEIRO	FALSO	FALSO	4.85%	0.0%
UCDB	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet101	Deformable	VERDADEIRO	FALSO	FALSO	5.14%	0.0%
UCDB	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet101	Deformable	VERDADEIRO	FALSO	FALSO	9.37%	0.0%
UCDB	UCDB	0,001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	87.37%	91.03%
UCDB	UCDB	0,001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	73.4%	78.73%
UCDB	UCDB	0,001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	91.23%	93.81%
UCDB	UPS	0,001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	69.73%	90.57%
UCDB	UPS	0,001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	67.4%	90.19%
UCDB	UPS	0,001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	66.08%	91.13%
UCDB	UCDB	0,001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	97.96%	98.66%
UCDB	UCDB	0,001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	90.36%	93.15%
UCDB	UCDB	0,001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	95.86%	97.25%
UCDB	UPS	0,001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	68.37%	89.62%
UCDB	UPS	0,001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	66.81%	90.14%
UCDB	UPS	0,001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	68.56%	90.76%
UCDB	UCDB	0,001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	82.33%	87.15%
UCDB	UCDB	0,001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	89.77%	93.75%
UCDB	UPS	0,001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	59.99%	84.47%

UCDB	UPS	0,001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	63.13%	85.54%
UCDB	UCDB	0,001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	85.67%	90.42%
UCDB	UCDB	0,001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	78.3%	86.64%
UCDB	UPS	0,001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	68.25%	90.68%
UCDB	UPS	0,001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	64.48%	88.19%
UCDB	UCDB	0,001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	85.46%	89.57%
UCDB	UCDB	0,001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	85.47%	89.77%
UCDB	UPS	0,001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	66.39%	88.4%
UCDB	UPS	0,001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	65.38%	89.27%
UCDB	UCDB	0,01	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	74.58%	83.16%
UCDB	UCDB	0,01	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	68.64%	80.5%
UCDB	UCDB	0,01	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	87.11%	91.08%
UCDB	UPS	0,01	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	65.84%	90.48%
UCDB	UPS	0,01	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	59.35%	86.48%
UCDB	UPS	0,01	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	67.3%	91.06%
UCDB	UCDB	0,01	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	87.76%	91.64%
UCDB	UCDB	0,01	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	86.8%	90.71%
UCDB	UCDB	0,01	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	94.8%	96.47%
UCDB	UPS	0,01	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	64.43%	89.05%
UCDB	UPS	0,01	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	66.46%	90.67%
UCDB	UPS	0,01	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	70.98%	91.59%
UCDB	UCDB	0,01	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	71.45%	78.12%
UCDB	UCDB	0,01	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	66.47%	74.35%
UCDB	UPS	0,01	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	53.3%	77.54%
UCDB	UPS	0,01	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	58.14%	85.04%
UCDB	UCDB	0,01	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	73.69%	83.47%
UCDB	UCDB	0,01	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	69.72%	82.85%
UCDB	UPS	0,01	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	62.58%	86.68%
UCDB	UPS	0,01	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	57.82%	84.89%
UCDB	UCDB	0,01	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	70.15%	80.63%
UCDB	UCDB	0,01	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	72.8%	83.75%
UCDB	UPS	0,01	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	56.47%	81.0%
UCDB	UPS	0,01	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	60.55%	86.76%
UPS	UCDB	0,00003	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	64.66%	80.57%
UPS	UCDB	0,00003	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	60.1%	80.27%
UPS	UPS	0,00003	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	75.95%	92.25%
UPS	UPS	0,00003	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	73.98%	90.08%
UPS	UCDB	0,00003	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	69.13%	79.82%
UPS	UCDB	0,00003	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	72.83%	81.26%
UPS	UPS	0,00003	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	74.02%	93.29%
UPS	UPS	0,00003	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	92.11%	96.94%
UPS	UCDB	0,00006	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	59.23%	73.99%
UPS	UCDB	0,00006	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	67.46%	79.38%
UPS	UPS	0,00006	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	72.26%	90.48%
UPS	UPS	0,00006	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	69.78%	88.78%
UPS	UCDB	0,00006	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	65.26%	74.74%
UPS	UCDB	0,00006	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	69.84%	80.74%
UPS	UPS	0,00006	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	80.11%	92.79%
UPS	UPS	0,00006	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	92.76%	97.17%
UPS	UCDB	0,0001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	58.81%	78.83%
UPS	UCDB	0,0001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	59.7%	76.68%
UPS	UCDB	0,0001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	72.44%	80.72%
UPS	UCDB	0,0001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	59.71%	75.18%
UPS	UCDB	0,0001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	57.83%	73.05%
UPS	UCDB	0,0001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	53.7%	71.0%
UPS	UPS	0,0001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	76.46%	92.92%
UPS	UPS	0,0001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	80.89%	94.94%
UPS	UPS	0,0001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	86.11%	95.94%
UPS	UCDB	0,0001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	60.27%	77.82%
UPS	UCDB	0,0001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	69.15%	74.84%
UPS	UCDB	0,0001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	50.75%	69.22%
UPS	UPS	0,0001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	79.39%	93.02%
UPS	UPS	0,0001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	84.25%	92.6%
UPS	UPS	0,0001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	77.62%	93.84%
UPS	UCDB	0,0001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	54.42%	76.24%
UPS	UCDB	0,0001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	62.87%	78.33%
UPS	UCDB	0,0001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	57.08%	77.85%
UPS	UPS	0,0001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	65.66%	87.17%
UPS	UPS	0,0001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	81.67%	94.17%
UPS	UPS	0,0001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	67.11%	89.93%
UPS	UCDB	0,0001	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	67.45%	79.53%
UPS	UCDB	0,0001	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	66.05%	80.99%
UPS	UPS	0,0001	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	70.63%	90.37%
UPS	UPS	0,0001	CUT	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	74.54%	90.31%
UPS	UCDB	0,0001	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	67.75%	77.66%
UPS	UCDB	0,0001	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	70.61%	82.57%
UPS	UPS	0,0001	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	72.45%	92.2%
UPS	UPS	0,0001	CUT	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	95.99%	98.37%
UPS	UCDB	0,0001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	46.42%	67.24%
UPS	UCDB	0,0001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	57.86%	70.72%
UPS	UPS	0,0001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	74.64%	94.1%
UPS	UPS	0,0001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	85.63%	95.77%
UPS	UCDB	0,001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	64.93%	77.26%
UPS	UCDB	0,001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	73.08%	78.98%
UPS	UCDB	0,001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	68.27%	79.55%
UPS	UCDB	0,001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	70.55%	78.39%
UPS	UCDB	0,001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	64.1%	76.39%
UPS	UCDB	0,001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	63.85%	77.05%
UPS	UPS	0,001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	90.42%	96.85%
UPS	UPS	0,001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	72.16%	93.14%
UPS	UPS	0,001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	77.62%	94.06%
UPS	UCDB	0,001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	59.27%	77.37%
UPS	UCDB	0,001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	65.2%	74.6%
UPS	UCDB	0,001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	56.78%	75.17%
UPS	UPS	0,001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	94.84%	98.08%
UPS	UPS	0,001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	65.51%	87.94%

UPS	UPS	0.001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	84.36%	95.75%
UPS	UCDB	0.001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	59.66%	79.12%
UPS	UCDB	0.001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	58.09%	77.16%
UPS	UCDB	0.001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	57.74%	79.41%
UPS	UPS	0.001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	79.28%	93.62%
UPS	UPS	0.001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	70.11%	91.59%
UPS	UPS	0.001	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	69.64%	90.47%
UPS	UCDB	0.001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	44.97%	62.67%
UPS	UCDB	0.001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	47.95%	67.21%
UPS	UPS	0.001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	78.42%	94.14%
UPS	UPS	0.001	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	93.02%	97.76%
UPS	UCDB	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	62.57%	78.82%
UPS	UCDB	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	61.4%	78.57%
UPS	UCDB	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	58.02%	78.03%
UPS	UCDB	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	59.94%	72.94%
UPS	UCDB	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	56.44%	72.83%
UPS	UCDB	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	61.27%	74.47%
UPS	UPS	0.01	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	75.77%	94.57%
UPS	UPS	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	68.21%	92.96%
UPS	UPS	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	89.83%	96.68%
UPS	UCDB	0.01	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	64.48%	75.21%
UPS	UCDB	0.01	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	57.74%	73.53%
UPS	UCDB	0.01	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	66.31%	76.65%
UPS	UPS	0.01	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	75.78%	94.34%
UPS	UPS	0.01	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	68.07%	90.25%
UPS	UPS	0.01	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	72.05%	92.27%
UPS	UCDB	0.01	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	53.22%	77.97%
UPS	UCDB	0.01	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	51.91%	73.2%
UPS	UCDB	0.01	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	62.38%	74.77%
UPS	UPS	0.01	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	68.37%	90.44%
UPS	UPS	0.01	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	59.02%	87.69%
UPS	UPS	0.01	CUT	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	70.43%	89.94%
UPS	UCDB	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	30.23%	61.88%
UPS	UCDB	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	45.4%	66.34%
UPS	UPS	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	51.09%	88.62%
UPS	UPS	0.01	CUT	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	69.32%	92.73%
UPS	UCDB	0.0003	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	64.46%	75.98%
UPS	UCDB	0.0003	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	67.74%	80.91%
UPS	UPS	0.0003	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	64.74%	85.11%
UPS	UPS	0.0003	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	80.43%	92.09%
UPS	UCDB	0.0006	CUT Mixed	SegFormer	Standard	FALSO	FALSO	FALSO	61.52%	80.1%
UPS	UPS	0.0006	CUT Mixed	SegFormer	Standard	FALSO	FALSO	FALSO	96.12%	98.64%
UPS	UCDB	0.0006	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	52.92%	73.23%
UPS	UCDB	0.0006	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	68.05%	80.21%
UPS	UPS	0.0006	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	62.51%	84.93%
UPS	UPS	0.0006	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	73.18%	89.22%
UPS	UCDB	0.0001	CUT Mixed	SegFormer	Standard	FALSO	FALSO	FALSO	60.23%	78.8%
UPS	UPS	0.0001	CUT Mixed	SegFormer	Standard	FALSO	FALSO	FALSO	87.97%	96.18%

UPS	UCDB	0.0001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	64.37%	77.88%
UPS	UCDB	0.0001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	65.4%	80.13%
UPS	UCDB	0.0001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	64.23%	80.31%
UPS	UPS	0.0001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	81.09%	94.73%
UPS	UPS	0.0001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	95.43%	98.47%
UPS	UPS	0.0001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	80.34%	95.38%
UPS	UCDB	0.0001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	63.55%	82.55%
UPS	UCDB	0.0001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	57.7%	81.08%
UPS	UCDB	0.0001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	56.55%	78.38%
UPS	UPS	0.0001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	86.48%	95.82%
UPS	UPS	0.0001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	79.84%	94.62%
UPS	UPS	0.0001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	83.91%	95.34%
UPS	UCDB	0.0001	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	49.07%	80.53%
UPS	UCDB	0.0001	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	65.39%	80.14%
UPS	UPS	0.0001	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	63.2%	89.7%
UPS	UPS	0.0001	CUT Mixed	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	81.26%	93.41%
UPS	UCDB	0.001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	60.45%	76.3%
UPS	UCDB	0.001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	61.94%	78.35%
UPS	UCDB	0.001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	60.94%	77.41%
UPS	UPS	0.001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	73.85%	93.66%
UPS	UPS	0.001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	81.16%	95.02%
UPS	UPS	0.001	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	86.8%	96.07%
UPS	UCDB	0.001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	61.85%	79.18%
UPS	UCDB	0.001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	57.75%	73.69%
UPS	UCDB	0.001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	56.69%	76.04%
UPS	UPS	0.001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	93.42%	97.78%
UPS	UPS	0.001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	87.42%	95.02%
UPS	UPS	0.001	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	93.73%	97.85%
UPS	UCDB	0.01	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	69.97%	77.55%
UPS	UCDB	0.01	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	66.61%	79.44%
UPS	UCDB	0.01	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	61.55%	80.66%
UPS	UPS	0.01	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	76.98%	93.88%
UPS	UPS	0.01	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	70.11%	91.6%
UPS	UPS	0.01	CUT Mixed	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	89.12%	96.49%
UPS	UCDB	0.01	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	61.3%	78.39%
UPS	UCDB	0.01	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	53.85%	77.63%
UPS	UCDB	0.01	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	63.53%	79.6%
UPS	UPS	0.01	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	81.52%	94.52%
UPS	UPS	0.01	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	79.82%	93.29%
UPS	UPS	0.01	CUT Mixed	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	84.97%	94.46%
UPS	UCDB	0.0001	Histogram Matching DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	FALSO	56.99%	75.93%
UPS	UCDB	0.0001	Histogram Matching DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	FALSO	58.06%	76.81%
UPS	UCDB	0.0001	Histogram Matching DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	FALSO	58.35%	74.48%
UPS	UCDB	0.001	Histogram Matching DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	FALSO	64.44%	77.1%
UPS	UCDB	0.001	Histogram Matching DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	FALSO	53.88%	77.01%
UPS	UCDB	0.001	Histogram Matching DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	FALSO	56.41%	76.06%
UPS	UCDB	0.01	Histogram Matching DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	FALSO	52.77%	76.72%
UPS	UCDB	0.01	Histogram Matching DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	FALSO	38.34%	61.74%

UPS	UCDB	0,01	Histogram Matching	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	65.13%	76.51%
UPS	UCDB	0,0001	MixStyle	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	57.82%	70.33%
UPS	UCDB	0,0001	MixStyle	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	56.38%	75.34%
UPS	UPS	0,0001	MixStyle	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	95.98%	98.52%
UPS	UPS	0,0001	MixStyle	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	92.15%	97.13%
UPS	UCDB	0,0001	MixStyle	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	54.79%	72.45%
UPS	UPS	0,0001	MixStyle	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	90.39%	96.79%
UPS	UCDB	0,001	MixStyle	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	52.23%	67.58%
UPS	UCDB	0,001	MixStyle	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	52.44%	70.69%
UPS	UPS	0,001	MixStyle	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	78.51%	94.19%
UPS	UPS	0,001	MixStyle	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	80.34%	93.56%
UPS	UCDB	0,001	MixStyle	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	49.68%	72.2%
UPS	UPS	0,001	MixStyle	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	80.92%	94.19%
UPS	UCDB	0,01	MixStyle	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	47.26%	65.04%
UPS	UCDB	0,01	MixStyle	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	45.54%	64.89%
UPS	UPS	0,01	MixStyle	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	67.57%	93.02%
UPS	UPS	0,01	MixStyle	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	67.74%	92.48%
UPS	UCDB	0,01	MixStyle	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	31.77%	59.32%
UPS	UPS	0,01	MixStyle	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	39.99%	73.3%
UPS	UCDB	0,00001	Não Utilizado	SegFormer	Standard	FALSO	FALSO	FALSO	59.55%	76.62%
UPS	UPS	0,00001	Não Utilizado	SegFormer	Standard	FALSO	FALSO	FALSO	81.54%	94.69%
UPS	UCDB	0,00003	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	67.86%	79.09%
UPS	UCDB	0,00003	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	69.19%	80.91%
UPS	UCDB	0,00003	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	66.44%	78.63%
UPS	UCDB	0,00003	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	66.71%	79.15%
UPS	UPS	0,00003	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	84.88%	95.63%
UPS	UPS	0,00003	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	92.41%	97.28%
UPS	UPS	0,00003	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	86.1%	95.17%
UPS	UPS	0,00003	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	92.76%	97.15%
UPS	UCDB	0,00003	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	63.59%	78.74%
UPS	UCDB	0,00003	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	69.02%	78.9%
UPS	UPS	0,00003	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	70.51%	88.92%
UPS	UPS	0,00003	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	77.58%	91.86%
UPS	UCDB	0,00006	Não Utilizado	SegFormer	Standard	FALSO	FALSO	FALSO	63.15%	78.57%
UPS	UPS	0,00006	Não Utilizado	SegFormer	Standard	FALSO	FALSO	FALSO	91.94%	97.46%
UPS	UCDB	0,00006	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	66.44%	74.75%
UPS	UCDB	0,00006	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	71.65%	82.61%
UPS	UCDB	0,00006	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	68.11%	79.74%
UPS	UCDB	0,00006	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	70.56%	81.64%
UPS	UPS	0,00006	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	78.46%	92.82%
UPS	UPS	0,00006	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	95.36%	98.14%
UPS	UPS	0,00006	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	84.12%	94.59%
UPS	UPS	0,00006	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	90.69%	97.08%
UPS	UCDB	0,00006	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	59.14%	77.91%
UPS	UCDB	0,00006	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	66.52%	80.89%
UPS	UPS	0,00006	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	68.92%	88.73%
UPS	UPS	0,00006	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	78.04%	91.24%
UPS	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	64.97%	75.14%

UPS	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	68.69%	77.9%
UPS	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	58.15%	75.59%
UPS	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	74.07%	92.64%
UPS	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	77.28%	94.01%
UPS	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	78.87%	94.45%
UPS	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	74.15%	80.63%
UPS	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	64.14%	74.46%
UPS	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	90.82%	97.42%
UPS	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	82.4%	95.0%
UPS	UCDB	0,0001	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	69.51%	80.11%
UPS	UCDB	0,0001	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	71.41%	82.08%
UPS	UCDB	0,0001	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	66.4%	79.08%
UPS	UCDB	0,0001	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	67.0%	80.61%
UPS	UPS	0,0001	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	78.93%	92.87%
UPS	UPS	0,0001	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	94.94%	98.0%
UPS	UPS	0,0001	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	76.84%	91.34%
UPS	UPS	0,0001	Não Utilizado	Swin Transformer Tiny	Standard	FALSO	FALSO	FALSO	89.29%	95.98%
UPS	UCDB	0,0001	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	53.23%	70.69%
UPS	UCDB	0,0001	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	66.47%	81.38%
UPS	UPS	0,0001	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	63.16%	88.76%
UPS	UPS	0,0001	Não Utilizado	Swin Transformer Base	Standard	FALSO	FALSO	FALSO	81.15%	92.2%
UPS	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	57.82%	70.33%
UPS	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	56.38%	75.34%
UPS	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	FALSO	FALSO	95.98%	98.52%
UPS	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	FALSO	FALSO	92.15%	97.13%
UPS	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet101	Deformable	FALSO	VERDADEIRO	VERDADEIRO	12.21%	6.47%
UPS	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet101	Deformable	FALSO	VERDADEIRO	VERDADEIRO	11.78%	32.97%
UPS	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	66.72%	79.79%
UPS	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	69.72%	80.48%
UPS	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	97.12%	98.82%
UPS	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	91.43%	97.27%
UPS	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	70.72%	81.55%
UPS	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	62.68%	76.86%
UPS	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	96.24%	99.34%
UPS	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	88.09%	95.96%
UPS	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	VERDADEIRO	FALSO	FALSO	17.19%	51.58%
UPS	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	VERDADEIRO	FALSO	FALSO	17.19%	51.58%
UPS	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	VERDADEIRO	FALSO	FALSO	11.46%	1.72%
UPS	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	VERDADEIRO	FALSO	FALSO	3.04%	0.02%
UPS	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	VERDADEIRO	FALSO	FALSO	9.06%	0.0%
UPS	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet101	Standard	VERDADEIRO	FALSO	FALSO	3.31%	0.8%
UPS	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet101	Deformable	VERDADEIRO	FALSO	FALSO	15.31%	2.0%
UPS	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet101	Deformable	VERDADEIRO	FALSO	FALSO	20.35%	39.13%
UPS	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet101	Deformable	VERDADEIRO	FALSO	FALSO	9.36%	0.0%
UPS	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet101	Deformable	VERDADEIRO	FALSO	FALSO	15.41%	32.98%
UPS	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet101	Deformable	VERDADEIRO	FALSO	FALSO	21.33%	63.51%
UPS	UPS	0,0001	Não Utilizado	DeepLabV3 ResNet101	Deformable	VERDADEIRO	FALSO	FALSO	21.23%	63.7%
UPS	UCDB	0,0001	Não Utilizado	DeepLabV3 ResNet50	Standard	VERDADEIRO	FALSO	FALSO	17.69%	41.46%

UPS	UPS	0,0001	Não Utilizado	DeepLabv3 ResNet50	Standard	VERDADEIRO	FALSO	FALSO	21.56%	61.68%
UPS	UCDB	0,0001	Não Utilizado	DeepLabv3 ResNet50	Deformable	VERDADEIRO	FALSO	FALSO	22.71%	45.75%
UPS	UCDB	0,0001	Não Utilizado	DeepLabv3 ResNet50	Deformable	VERDADEIRO	FALSO	FALSO	17.19%	51.58%
UPS	UPS	0,0001	Não Utilizado	DeepLabv3 ResNet50	Deformable	VERDADEIRO	FALSO	FALSO	6.49%	10.05%
UPS	UPS	0,0001	Não Utilizado	DeepLabv3 ResNet50	Deformable	VERDADEIRO	FALSO	FALSO	3.12%	0.0%
UPS	UCDB	0,001	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	70.8%	79.47%
UPS	UCDB	0,001	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	66.77%	80.49%
UPS	UCDB	0,001	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	67.56%	76.01%
UPS	UPS	0,001	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	79.79%	94.95%
UPS	UPS	0,001	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	80.52%	94.88%
UPS	UPS	0,001	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	80.37%	94.61%
UPS	UCDB	0,001	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	FALSO	FALSO	70.54%	79.83%
UPS	UCDB	0,001	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	FALSO	FALSO	67.69%	78.98%
UPS	UCDB	0,001	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	FALSO	FALSO	61.7%	79.7%
UPS	UPS	0,001	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	FALSO	FALSO	79.51%	94.59%
UPS	UPS	0,001	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	FALSO	FALSO	79.44%	94.55%
UPS	UPS	0,001	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	FALSO	FALSO	95.67%	98.45%
UPS	UCDB	0,001	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	FALSO	FALSO	52.23%	67.58%
UPS	UCDB	0,001	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	52.44%	70.69%
UPS	UPS	0,001	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	FALSO	FALSO	78.51%	94.19%
UPS	UPS	0,001	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	80.34%	93.56%
UPS	UCDB	0,001	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	64.2%	75.02%
UPS	UCDB	0,001	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	61.88%	76.23%
UPS	UPS	0,001	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	81.22%	94.35%
UPS	UPS	0,001	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	73.2%	92.29%
UPS	UCDB	0,001	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	65.94%	77.99%
UPS	UCDB	0,001	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	54.04%	78.14%
UPS	UPS	0,001	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	76.43%	93.76%
UPS	UPS	0,001	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	69.27%	93.26%
UPS	UCDB	0,001	Não Utilizado	DeepLabv3 ResNet101	Standard	VERDADEIRO	FALSO	FALSO	11.04%	0.04%
UPS	UCDB	0,001	Não Utilizado	DeepLabv3 ResNet101	Standard	VERDADEIRO	FALSO	FALSO	5.17%	0.0%
UPS	UPS	0,001	Não Utilizado	DeepLabv3 ResNet101	Standard	VERDADEIRO	FALSO	FALSO	21.23%	63.7%
UPS	UPS	0,001	Não Utilizado	DeepLabv3 ResNet101	Standard	VERDADEIRO	FALSO	FALSO	24.36%	53.35%
UPS	UCDB	0,001	Não Utilizado	DeepLabv3 ResNet101	Deformable	VERDADEIRO	FALSO	FALSO	17.19%	51.58%
UPS	UPS	0,001	Não Utilizado	DeepLabv3 ResNet101	Deformable	VERDADEIRO	FALSO	FALSO	6.36%	0.0%
UPS	UCDB	0,001	Não Utilizado	DeepLabv3 ResNet50	Standard	VERDADEIRO	FALSO	FALSO	13.8%	9.56%
UPS	UCDB	0,001	Não Utilizado	DeepLabv3 ResNet50	Standard	VERDADEIRO	FALSO	FALSO	12.8%	4.85%
UPS	UPS	0,001	Não Utilizado	DeepLabv3 ResNet50	Standard	VERDADEIRO	FALSO	FALSO	23.12%	42.96%
UPS	UPS	0,001	Não Utilizado	DeepLabv3 ResNet50	Standard	VERDADEIRO	FALSO	FALSO	6.42%	9.83%
UPS	UCDB	0,001	Não Utilizado	DeepLabv3 ResNet50	Deformable	VERDADEIRO	FALSO	FALSO	8.48%	10.88%
UPS	UPS	0,001	Não Utilizado	DeepLabv3 ResNet50	Deformable	VERDADEIRO	FALSO	FALSO	3.09%	0.01%
UPS	UCDB	0,01	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	64.64%	78.36%
UPS	UCDB	0,01	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	41.01%	73.41%
UPS	UCDB	0,01	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	68.8%	77.65%
UPS	UPS	0,01	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	79.84%	95.05%
UPS	UPS	0,01	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	51.88%	90.6%
UPS	UPS	0,01	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	87.74%	96.31%
UPS	UCDB	0,01	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	FALSO	FALSO	63.95%	72.65%
UPS	UCDB	0,01	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	FALSO	FALSO	64.03%	76.31%
UPS	UCDB	0,01	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	FALSO	FALSO	69.92%	79.64%
UPS	UPS	0,01	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	FALSO	FALSO	76.0%	93.42%
UPS	UPS	0,01	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	FALSO	FALSO	75.4%	93.5%
UPS	UPS	0,01	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	FALSO	FALSO	95.06%	98.13%
UPS	UCDB	0,01	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	FALSO	FALSO	47.26%	65.04%
UPS	UCDB	0,01	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	45.54%	64.89%
UPS	UPS	0,01	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	FALSO	FALSO	67.57%	93.02%
UPS	UPS	0,01	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	FALSO	FALSO	67.74%	92.48%
UPS	UCDB	0,01	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	59.69%	75.75%
UPS	UCDB	0,01	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	42.84%	73.77%
UPS	UPS	0,01	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	69.8%	93.31%
UPS	UPS	0,01	Não Utilizado	DeepLabv3 ResNet50	Standard	FALSO	VERDADEIRO	VERDADEIRO	59.36%	92.45%
UPS	UCDB	0,01	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	51.67%	73.86%
UPS	UCDB	0,01	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	43.56%	76.21%
UPS	UPS	0,01	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	68.41%	92.04%
UPS	UPS	0,01	Não Utilizado	DeepLabv3 ResNet101	Standard	FALSO	VERDADEIRO	VERDADEIRO	54.95%	92.51%
UPS	UCDB	0,01	Não Utilizado	DeepLabv3 ResNet101	Standard	VERDADEIRO	FALSO	FALSO	16.31%	9.3%
UPS	UCDB	0,01	Não Utilizado	DeepLabv3 ResNet101	Standard	VERDADEIRO	FALSO	FALSO	17.2%	51.55%
UPS	UPS	0,01	Não Utilizado	DeepLabv3 ResNet101	Standard	VERDADEIRO	FALSO	FALSO	21.23%	63.7%
UPS	UPS	0,01	Não Utilizado	DeepLabv3 ResNet101	Standard	VERDADEIRO	FALSO	FALSO	10.93%	5.95%
UPS	UCDB	0,01	Não Utilizado	DeepLabv3 ResNet101	Deformable	VERDADEIRO	FALSO	FALSO	15.91%	39.15%
UPS	UCDB	0,01	Não Utilizado	DeepLabv3 ResNet101	Deformable	VERDADEIRO	FALSO	FALSO	17.8%	47.85%
UPS	UCDB	0,01	Não Utilizado	DeepLabv3 ResNet101	Deformable	VERDADEIRO	FALSO	FALSO	13.63%	29.31%
UPS	UPS	0,01	Não Utilizado	DeepLabv3 ResNet101	Deformable	VERDADEIRO	FALSO	FALSO	12.31%	8.79%
UPS	UPS	0,01	Não Utilizado	DeepLabv3 ResNet101	Deformable	VERDADEIRO	FALSO	FALSO	21.29%	63.68%
UPS	UPS	0,01	Não Utilizado	DeepLabv3 ResNet101	Deformable	VERDADEIRO	FALSO	FALSO	12.69%	19.15%
UPS	UCDB	0,01	Não Utilizado	DeepLabv3 ResNet50	Standard	VERDADEIRO	FALSO	FALSO	17.25%	51.24%
UPS	UCDB	0,01	Não Utilizado	DeepLabv3 ResNet50	Standard	VERDADEIRO	FALSO	FALSO	17.19%	51.56%
UPS	UPS	0,01	Não Utilizado	DeepLabv3 ResNet50	Standard	VERDADEIRO	FALSO	FALSO	9.29%	0.69%
UPS	UPS	0,01	Não Utilizado	DeepLabv3 ResNet50	Standard	VERDADEIRO	FALSO	FALSO	18.44%	54.93%
UPS	UCDB	0,01	Não Utilizado	DeepLabv3 ResNet50	Deformable	VERDADEIRO	FALSO	FALSO	17.19%	51.57%
UPS	UPS	0,01	Não Utilizado	DeepLabv3 ResNet50	Deformable	VERDADEIRO	FALSO	FALSO	9.06%	0.01%

Jaccard index - Grass	Jaccard index - Vegetation	Overall Dice Coefficient
70.49%	65.03%	64.21%
74.57%	73.05%	66.94%
56.17%	14.78%	63.07%
55.9%	31.44%	68.98%
83.72%	83.78%	92.09%
97.99%	97.27%	98.97%
66.46%	46.23%	78.9%
75.44%	53.14%	83.49%
54.41%	46.67%	73.74%
63.2%	50.49%	77.85%
56.91%	24.04%	67.01%
61.2%	37.33%	74.3%
78.37%	75.31%	88.61%
97.28%	95.7%	98.47%
70.1%	38.92%	77.35%
72.51%	50.27%	81.63%
88.72%	85.89%	94.25%
91.95%	89.71%	95.93%
81.25%	76.81%	90.15%
59.92%	68.99%	80.8%
54.14%	74.35%	80.3%
52.66%	40.54%	69.24%
61.64%	40.96%	75.77%
64.75%	40.27%	76.7%
61.24%	38.24%	74.36%
65.59%	68.61%	82.88%
30.84%	63.56%	68.4%
57.68%	55.39%	77.46%
58.72%	35.99%	72.9%
61.62%	38.08%	75.01%
62.81%	34.04%	73.7%
65.28%	49.71%	78.61%
77.41%	74.23%	88.25%
63.41%	32.38%	73.07%
58.92%	36.69%	72.06%
97.04%	96.15%	98.51%
78.16%	76.28%	88.74%
74.26%	49.2%	81.99%
71.09%	38.24%	77.72%
88.09%	80.58%	92.78%
90.62%	88.06%	95.15%
61.44%	36.76%	73.8%
69.75%	44.28%	79.3%
84.02%	81.78%	91.95%
96.0%	94.8%	97.97%
87.96%	84.7%	93.62%
74.72%	79.83%	88.49%

48.82%	67.73%	76.38%
65.08%	78.14%	84.8%
59.7%	40.61%	75.29%
58.39%	35.75%	73.46%
60.24%	41.43%	75.48%
73.24%	75.43%	86.86%
77.77%	52.71%	64.24%
42.43%	71.55%	75.42%
62.82%	32.04%	72.52%
49.08%	28.05%	67.26%
61.72%	35.71%	74.12%
72.97%	81.02%	88.33%
75.95%	74.91%	87.93%
55.98%	21.89%	66.25%
46.62%	29.27%	64.86%
75.69%	61.96%	85.6%
83.41%	81.74%	91.75%
77.38%	69.03%	87.65%
54.16%	70.04%	79.7%
37.9%	66.64%	72.09%
69.67%	78.02%	85.91%
61.33%	37.64%	75.07%
64.21%	37.48%	75.68%
62.98%	40.07%	76.1%
57.48%	63.7%	79.18%
60.1%	61.63%	79.98%
53.02%	78.32%	80.84%
58.67%	33.48%	72.24%
52.81%	33.37%	70.67%
63.66%	37.02%	74.69%
41.17%	74.13%	75.32%
66.77%	73.36%	84.34%
18.97%	24.85%	51.39%
58.42%	21.83%	67.06%
53.93%	49.98%	74.65%
74.36%	77.58%	87.94%
56.55%	16.83%	64.18%
60.68%	43.65%	75.61%
95.25%	94.06%	97.65%
69.67%	46.41%	79.69%
68.05%	69.08%	84.13%
62.29%	67.8%	82.26%
67.3%	40.99%	77.13%
57.86%	36.32%	73.26%
97.11%	96.11%	98.52%
74.72%	44.7%	80.73%
86.47%	86.5%	93.7%
80.99%	72.55%	89.59%

84.62%	83.02%	92.38%
74.44%	42.74%	80.38%
69.45%	22.56%	70.99%
64.26%	22.55%	73.33%
95.15%	93.91%	97.6%
97.73%	97.07%	98.87%
93.97%	92.8%	97.06%
61.93%	43.79%	76.89%
67.8%	46.1%	79.63%
66.55%	41.17%	77.07%
60.9%	61.67%	80.29%
53.82%	74.41%	80.45%
59.4%	31.88%	72.05%
60.82%	38.59%	74.26%
91.55%	90.89%	95.86%
84.81%	80.77%	91.87%
87.49%	87.19%	94.14%
70.49%	43.8%	79.46%
67.73%	36.45%	75.82%
72.8%	46.62%	80.96%
98.74%	98.28%	99.36%
71.98%	78.88%	87.62%
97.36%	96.56%	98.68%
71.15%	44.48%	80.0%
68.99%	41.67%	78.42%
73.49%	48.38%	81.7%
83.49%	69.59%	88.51%
75.99%	80.91%	89.38%
97.75%	96.94%	98.85%
65.78%	30.89%	73.84%
66.21%	37.1%	76.04%
71.89%	44.34%	80.16%
89.4%	87.83%	94.83%
95.14%	93.78%	97.56%
86.25%	83.52%	93.02%
92.33%	93.73%	96.7%
77.67%	76.32%	88.68%
93.92%	92.31%	96.93%
81.69%	77.87%	90.45%
84.57%	82.05%	92.23%
85.01%	82.05%	92.29%
95.51%	93.84%	97.69%
89.88%	85.48%	94.4%
69.34%	44.33%	79.16%
69.19%	45.3%	79.49%
90.13%	87.83%	95.04%
70.1%	43.19%	79.12%
89.76%	85.81%	94.58%

61.43%	42.42%	75.96%
88.65%	84.46%	94.08%
61.88%	40.66%	75.18%
51.81%	73.25%	79.37%
57.38%	32.0%	71.11%
31.62%	0.0%	42.74%
49.42%	0.0%	50.84%
89.14%	85.82%	94.32%
66.15%	45.85%	78.93%
83.5%	80.46%	91.42%
92.86%	91.69%	96.51%
85.2%	83.54%	92.63%
96.05%	94.59%	97.96%
60.62%	42.23%	75.36%
72.63%	48.46%	81.18%
64.84%	44.95%	78.03%
73.8%	46.68%	81.1%
70.94%	67.27%	84.69%
72.18%	73.83%	86.21%
63.0%	36.91%	74.66%
65.17%	42.8%	76.86%
97.26%	96.33%	98.59%
75.1%	48.3%	82.12%
81.29%	77.75%	90.37%
86.72%	84.6%	93.27%
77.3%	79.41%	88.84%
91.17%	90.12%	95.75%
70.04%	42.07%	78.77%
71.53%	43.98%	79.71%
69.76%	37.85%	77.12%
70.49%	41.93%	78.58%
68.93%	70.31%	84.33%
73.77%	78.12%	88.09%
59.15%	34.02%	72.11%
63.69%	41.36%	76.09%
88.56%	86.98%	94.4%
89.74%	81.76%	93.41%
84.31%	81.09%	91.74%
66.15%	36.02%	75.71%
72.77%	42.43%	79.46%
71.66%	45.5%	80.3%
91.97%	89.81%	95.98%
92.96%	91.5%	96.52%
93.29%	92.2%	96.75%
65.49%	43.66%	77.7%
72.24%	46.52%	80.85%
71.67%	45.45%	80.51%
78.42%	76.17%	88.79%

93.3%	92.65%	96.83%
81.25%	76.94%	90.88%
83.83%	80.36%	91.51%
65.05%	38.55%	75.91%
69.59%	49.05%	80.51%
65.51%	42.46%	77.43%
73.7%	46.23%	80.88%
40.39%	38.46%	65.34%
75.54%	76.35%	87.96%
65.0%	56.73%	80.1%
68.42%	68.0%	83.68%
59.06%	25.31%	68.33%
62.96%	41.77%	75.62%
59.02%	35.61%	72.87%
61.63%	37.53%	74.29%
97.78%	96.0%	98.85%
89.88%	85.48%	94.4%
69.48%	44.4%	79.31%
69.19%	45.3%	79.49%
91.4%	89.09%	95.67%
93.51%	90.26%	96.41%
70.36%	46.12%	80.52%
72.62%	46.4%	81.09%
89.99%	89.23%	95.24%
93.09%	90.27%	96.26%
72.34%	47.1%	81.1%
73.0%	47.67%	81.58%
30.91%	0.02%	23.1%
31.06%	0.0%	23.86%
4.83%	0.0%	29.16%
5.37%	9.18%	9.0%
0.0%	15.42%	8.91%
27.0%	1.11%	14.91%
86.87%	84.21%	93.24%
62.41%	79.07%	84.42%
90.48%	89.42%	95.41%
71.07%	47.54%	80.86%
69.12%	42.88%	78.87%
65.89%	41.23%	77.73%
98.04%	97.2%	98.97%
90.3%	87.64%	94.92%
95.74%	94.58%	97.88%
69.23%	46.26%	79.87%
66.73%	43.57%	78.52%
69.01%	45.92%	79.92%
79.47%	80.36%	90.27%
89.76%	85.81%	94.58%
58.76%	36.73%	73.11%

61.43%	42.42%	75.96%
87.26%	79.32%	92.21%
75.83%	72.44%	87.7%
69.3%	44.77%	79.61%
64.88%	40.36%	76.65%
84.97%	81.84%	92.13%
85.12%	81.52%	92.13%
67.57%	43.21%	78.28%
63.36%	43.5%	77.51%
69.98%	70.6%	85.3%
62.63%	62.79%	81.12%
87.27%	82.99%	93.08%
69.51%	37.54%	77.2%
61.17%	30.4%	71.76%
70.96%	39.9%	78.46%
87.86%	83.78%	93.45%
85.47%	84.22%	92.91%
94.44%	93.48%	97.32%
66.3%	37.93%	76.32%
67.89%	40.81%	77.96%
73.48%	47.88%	81.69%
66.0%	70.34%	83.25%
51.81%	73.25%	79.37%
47.46%	34.9%	67.82%
57.38%	32.0%	71.11%
70.27%	67.33%	84.67%
62.71%	63.62%	81.82%
63.72%	37.33%	75.02%
53.63%	34.95%	71.15%
62.94%	66.87%	82.23%
69.07%	65.57%	84.02%
54.57%	33.84%	70.23%
61.57%	33.81%	73.03%
61.25%	32.15%	77.92%
60.66%	39.36%	73.68%
78.33%	57.27%	85.55%
76.26%	55.6%	84.26%
65.94%	61.64%	81.51%
70.97%	66.27%	84.13%
74.71%	54.07%	84.08%
93.26%	86.13%	95.84%
44.12%	60.19%	73.67%
61.71%	61.28%	80.27%
74.87%	51.42%	82.85%
71.18%	49.4%	81.11%
60.11%	60.93%	78.78%
67.36%	61.4%	81.96%
83.11%	64.43%	88.47%

93.78%	87.34%	96.2%
59.99%	37.82%	72.81%
60.77%	41.85%	73.73%
70.21%	66.4%	83.88%
58.41%	45.55%	74.05%
48.45%	51.99%	72.7%
39.87%	50.22%	68.97%
79.19%	57.27%	85.85%
83.33%	64.4%	88.89%
89.04%	73.35%	92.25%
60.36%	42.64%	74.2%
65.82%	66.79%	81.69%
44.01%	39.02%	66.36%
82.67%	62.48%	87.94%
83.77%	76.37%	91.31%
80.92%	58.09%	86.59%
47.54%	39.48%	69.19%
57.91%	52.38%	76.65%
52.33%	41.06%	71.49%
68.82%	40.99%	77.61%
84.07%	66.77%	89.47%
73.04%	38.37%	78.19%
58.81%	63.93%	80.22%
62.14%	55.03%	79.05%
71.97%	49.56%	81.64%
75.14%	58.16%	84.75%
63.91%	61.69%	80.57%
69.35%	59.92%	82.43%
76.15%	49.0%	82.72%
96.42%	93.18%	97.94%
39.03%	32.97%	62.05%
52.11%	50.74%	72.9%
78.93%	50.89%	84.21%
87.87%	73.25%	91.98%
63.52%	54.02%	78.34%
70.87%	69.39%	84.38%
63.72%	61.54%	80.88%
64.9%	68.34%	82.6%
52.93%	62.97%	77.71%
62.18%	52.33%	77.47%
91.98%	82.42%	94.86%
73.34%	49.99%	82.56%
81.28%	57.52%	86.55%
57.32%	43.12%	73.45%
59.99%	61.0%	78.74%
55.49%	39.68%	71.34%
96.13%	90.33%	97.32%
70.34%	38.25%	77.17%

87.3%	70.02%	91.14%
53.95%	45.89%	73.78%
57.33%	39.77%	72.3%
56.86%	36.94%	71.66%
81.43%	62.79%	87.87%
76.58%	42.17%	80.56%
77.81%	40.65%	80.11%
35.1%	37.15%	61.06%
40.12%	36.51%	63.72%
81.73%	59.39%	87.15%
94.28%	87.02%	96.33%
62.72%	46.16%	76.14%
62.66%	42.97%	75.05%
59.15%	36.89%	71.96%
47.52%	59.06%	74.35%
42.95%	53.55%	71.37%
55.25%	54.07%	75.58%
77.24%	55.3%	85.25%
66.65%	45.04%	79.48%
91.67%	81.16%	94.52%
54.16%	64.07%	78.07%
55.32%	44.38%	73.49%
60.64%	61.65%	79.52%
77.69%	55.32%	85.25%
73.19%	40.78%	79.11%
74.91%	48.96%	82.46%
49.07%	32.63%	67.56%
45.49%	37.04%	67.04%
55.6%	56.76%	76.48%
73.23%	41.44%	79.38%
61.37%	27.99%	71.08%
71.69%	49.67%	81.53%
25.97%	2.83%	41.86%
40.7%	29.17%	60.93%
62.45%	2.19%	58.38%
75.2%	40.04%	79.75%
55.7%	61.7%	78.07%
66.24%	56.07%	80.33%
63.16%	45.95%	77.45%
81.71%	67.49%	88.8%
63.76%	40.7%	74.89%
96.83%	92.91%	98.01%
41.07%	44.45%	68.11%
62.72%	61.24%	80.69%
59.82%	42.78%	75.34%
72.64%	57.69%	83.87%
62.37%	39.52%	73.87%
89.99%	77.73%	93.42%

63.79%	51.44%	77.8%
64.78%	51.29%	78.47%
63.98%	48.41%	77.45%
84.5%	64.05%	88.99%
96.37%	91.45%	97.64%
83.1%	62.53%	88.45%
61.16%	46.92%	76.74%
54.38%	37.64%	71.56%
52.31%	38.95%	70.88%
88.77%	74.87%	92.51%
82.92%	61.97%	88.14%
86.38%	70.02%	90.89%
54.05%	12.64%	60.61%
62.21%	53.83%	78.35%
70.91%	29.0%	74.17%
83.29%	67.08%	89.26%
61.5%	43.54%	74.46%
62.59%	44.88%	75.6%
59.82%	45.6%	74.92%
78.5%	49.38%	83.6%
84.63%	63.85%	89.02%
89.08%	75.24%	92.7%
55.09%	51.29%	75.74%
44.28%	55.29%	72.48%
48.42%	45.59%	71.42%
94.84%	87.64%	96.35%
86.85%	78.39%	93.14%
94.87%	88.47%	96.72%
65.96%	66.38%	82.22%
64.14%	56.25%	79.56%
63.14%	40.85%	74.9%
79.23%	57.83%	86.18%
74.61%	44.13%	80.77%
90.79%	80.08%	94.11%
56.28%	49.23%	75.29%
53.17%	30.76%	67.96%
59.2%	51.8%	77.09%
83.93%	66.11%	89.35%
82.78%	63.41%	88.24%
85.94%	74.51%	91.86%
56.19%	38.84%	71.41%
57.19%	40.17%	72.33%
54.34%	46.22%	73.0%
61.63%	54.58%	77.98%
56.38%	28.24%	67.72%
56.23%	36.94%	70.78%
52.82%	28.78%	66.88%
53.27%	0.0%	48.62%

61.5%	57.39%	78.59%
53.23%	49.87%	72.88%
56.05%	37.76%	70.86%
96.62%	92.81%	97.94%
93.46%	85.86%	95.85%
50.13%	41.81%	69.92%
92.18%	82.2%	94.84%
45.09%	44.03%	67.98%
52.38%	34.24%	67.53%
81.69%	59.64%	87.22%
82.11%	65.35%	88.63%
52.02%	24.81%	64.02%
83.86%	64.7%	88.93%
47.28%	29.47%	62.85%
41.24%	30.49%	61.28%
75.19%	34.5%	77.84%
74.14%	36.6%	78.28%
36.0%	0.0%	42.47%
46.66%	0.0%	49.41%
53.25%	48.29%	73.94%
83.83%	66.1%	89.36%
62.97%	61.51%	80.59%
66.83%	59.84%	81.48%
61.44%	59.25%	79.52%
64.63%	56.36%	79.65%
86.99%	72.04%	91.52%
93.66%	86.28%	95.99%
87.86%	75.26%	92.32%
93.74%	87.4%	96.2%
58.09%	53.95%	77.23%
64.0%	64.16%	81.47%
70.35%	52.27%	81.79%
79.51%	61.88%	86.8%
60.8%	50.07%	76.78%
93.6%	84.77%	95.72%
58.79%	65.79%	79.65%
69.71%	62.63%	83.22%
63.75%	60.83%	80.74%
67.58%	62.46%	82.48%
80.81%	61.75%	87.34%
95.91%	92.04%	97.61%
86.51%	71.27%	91.07%
91.85%	83.12%	95.02%
52.42%	47.11%	73.47%
63.92%	54.73%	79.39%
71.44%	46.91%	80.34%
79.33%	63.54%	87.3%
57.56%	62.23%	78.53%

61.4%	66.76%	81.24%
56.33%	42.55%	72.62%
78.07%	51.49%	83.95%
80.56%	57.27%	86.33%
82.41%	59.76%	87.44%
71.55%	70.27%	85.08%
62.2%	55.75%	77.88%
92.29%	82.76%	95.08%
85.64%	66.54%	89.87%
68.01%	60.4%	81.74%
70.48%	61.67%	83.04%
65.6%	54.52%	79.37%
66.58%	53.8%	79.72%
81.62%	62.28%	87.65%
93.53%	91.29%	97.38%
80.63%	58.56%	86.2%
90.62%	81.26%	94.23%
38.42%	50.57%	68.51%
63.93%	54.1%	79.32%
70.06%	30.67%	74.46%
81.81%	69.44%	89.3%
53.23%	49.87%	72.88%
56.05%	37.76%	70.86%
96.62%	92.81%	97.94%
93.46%	85.86%	95.85%
30.05%	0.12%	19.53%
2.32%	0.05%	18.07%
66.21%	54.16%	79.57%
66.96%	61.68%	81.91%
97.55%	94.99%	98.53%
92.82%	84.2%	95.44%
69.09%	61.54%	82.58%
58.68%	52.49%	76.57%
98.53%	96.86%	99.11%
91.07%	77.24%	93.47%
0.0%	0.0%	22.69%
0.0%	0.0%	22.68%
32.58%	0.06%	17.55%
0.0%	9.11%	5.58%
27.19%	0.0%	14.25%
0.0%	9.14%	6.11%
32.02%	11.9%	24.57%
21.92%	0.0%	30.74%
13.33%	14.74%	16.41%
13.21%	0.04%	24.34%
0.48%	0.0%	26.21%
0.0%	0.0%	25.94%
4.56%	7.05%	26.84%

2.99%	0.0%	27.37%
0.03%	22.35%	33.13%
0.0%	0.0%	22.69%
0.0%	9.24%	11.73%
0.25%	9.11%	5.73%
66.54%	66.39%	82.76%
65.1%	54.72%	79.6%
64.95%	61.73%	80.49%
81.99%	62.44%	88.13%
83.56%	63.12%	88.6%
83.76%	62.74%	88.5%
66.31%	65.47%	82.55%
63.04%	61.06%	80.47%
59.14%	46.27%	75.43%
83.48%	60.45%	87.86%
82.85%	60.94%	87.85%
96.49%	92.08%	97.77%
45.09%	44.03%	67.98%
52.38%	34.24%	67.53%
81.69%	59.64%	87.22%
82.11%	63.35%	88.63%
60.01%	57.57%	77.94%
60.38%	49.01%	75.86%
85.17%	64.14%	89.08%
76.63%	50.67%	83.34%
63.51%	56.31%	79.12%
55.72%	28.26%	67.79%
79.49%	56.05%	85.73%
76.51%	37.96%	79.43%
32.62%	0.47%	16.74%
0.09%	15.41%	8.96%
0.0%	0.0%	25.94%
19.71%	0.03%	34.19%
0.0%	0.0%	22.69%
9.47%	9.6%	11.61%
31.84%	0.0%	21.92%
33.46%	0.1%	19.86%
22.66%	3.74%	34.75%
0.12%	9.33%	11.73%
0.0%	14.55%	15.01%
0.13%	9.14%	5.68%
60.33%	55.24%	78.1%
49.63%	0.0%	50.34%
66.86%	61.9%	81.34%
82.31%	62.16%	88.14%
65.03%	0.0%	57.96%
89.79%	77.12%	93.27%
58.96%	60.23%	77.84%

58.2%	57.38%	77.74%
67.74%	62.38%	82.09%
78.98%	55.38%	85.44%
78.15%	54.54%	84.99%
95.8%	91.24%	97.44%
47.28%	29.47%	62.85%
41.24%	30.49%	61.28%
75.19%	34.5%	77.84%
74.14%	36.6%	78.28%
56.4%	46.92%	74.06%
49.43%	5.33%	53.73%
71.4%	44.68%	80.54%
68.94%	16.68%	68.76%
49.99%	31.15%	66.38%
51.98%	2.48%	53.25%
74.49%	38.7%	79.01%
68.51%	3.82%	61.59%
28.05%	11.57%	27.19%
0.0%	0.07%	22.72%
0.0%	0.0%	25.94%
26.84%	0.02%	17.86%
1.46%	7.12%	24.14%
0.0%	5.56%	25.09%
0.0%	11.59%	22.04%
20.2%	7.95%	21.5%
0.2%	0.0%	26.07%
12.71%	6.22%	22.14%
0.0%	0.5%	22.92%
0.02%	0.0%	22.69%
27.2%	0.0%	14.71%
0.01%	0.38%	23.89%
0.01%	0.0%	22.69%
27.18%	0.0%	14.25%

APÊNDICE 2 - TABELA NON TRIBRANCH

Network Architecture	Training Dataset	Testing Dataset	Learning Rate	Style Transfer	Overall Jaccard Index	Jaccard Index - Others	Jaccard Index - Grass	Jaccard Index - Vegetation	Overall Dice Coefficient
deeplabv3_resnet101	UCDB	UCDB	0.001	AdaIN	83.24%	87.24%	81.68%	80.82%	90.83%
segformer	UCDB	UCDB	0.001	AdaIN	60.33%	80.38%	52.16%	48.46%	74.32%
segformer	UCDB	UCDB	0.001	AdaIN	87.82%	92.34%	87.24%	83.87%	93.48%
deeplabv3_resnet101	UCDB	UCDB	0.001	AdaIN	91.8%	94.37%	91.71%	89.33%	95.71%
deeplabv3_resnet101	UCDB	UCDB	0.001	AdaIN	93.57%	95.44%	92.93%	92.35%	96.67%
segformer	UCDB	UCDB	0.001	AdaIN	88.91%	93.24%	88.19%	85.31%	94.1%
fcn_resnet50	UCDB	UCDB	0.001	AdaIN	84.84%	88.24%	83.7%	82.59%	91.78%
fcn_resnet50	UCDB	UCDB	0.001	AdaIN	86.94%	90.15%	86.12%	84.55%	93.0%
fcn_resnet50	UCDB	UCDB	0.001	AdaIN	89.67%	91.77%	87.98%	89.25%	94.55%
deeplabv3_resnet101	UCDB	UCDB	0.001	AdaIN	64.39%	89.54%	37.16%	37.16%	76.17%
segformer	UCDB	UPS	0.001	AdaIN	39.59%	37.63%	10.86%	20.26%	48.9%
segformer	UCDB	UPS	0.001	AdaIN	55.39%	91.25%	44.98%	29.92%	67.85%
deeplabv3_resnet101	UCDB	UPS	0.001	AdaIN	67.07%	89.57%	67.26%	44.38%	78.8%
deeplabv3_resnet101	UCDB	UPS	0.001	AdaIN	62.86%	87.17%	61.11%	40.3%	75.48%
segformer	UCDB	UPS	0.001	AdaIN	55.21%	89.84%	48.76%	27.04%	67.59%
fcn_resnet50	UCDB	UPS	0.001	AdaIN	59.84%	87.0%	54.44%	38.07%	72.9%
fcn_resnet50	UCDB	UPS	0.001	AdaIN	66.27%	89.49%	64.49%	44.84%	78.26%
fcn_resnet50	UCDB	UPS	0.001	AdaIN	65.9%	88.19%	66.23%	43.28%	77.94%
deeplabv3_resnet101	UCDB	UCDB	0.01	AdaIN	87.82%	87.13%	80.88%	80.47%	90.58%
segformer	UCDB	UCDB	0.01	AdaIN	31.41%	57.87%	8.53%	27.83%	44.19%
segformer	UCDB	UCDB	0.01	AdaIN	91.4%	94.44%	91.18%	88.59%	95.49%
deeplabv3_resnet101	UCDB	UCDB	0.01	AdaIN	91.69%	94.27%	91.51%	89.29%	95.65%
deeplabv3_resnet101	UCDB	UCDB	0.01	AdaIN	87.3%	90.87%	86.46%	84.54%	93.2%
segformer	UCDB	UCDB	0.01	AdaIN	94.4%	73.55%	45.87%	43.79%	69.52%
fcn_resnet50	UCDB	UCDB	0.01	AdaIN	85.4%	89.14%	83.84%	83.21%	92.1%
fcn_resnet50	UCDB	UCDB	0.01	AdaIN	91.56%	91.99%	91.24%	89.44%	95.58%
fcn_resnet50	UCDB	UCDB	0.01	AdaIN	82.42%	86.66%	80.38%	80.24%	90.34%
deeplabv3_resnet101	UCDB	UPS	0.01	AdaIN	61.97%	88.75%	59.34%	37.84%	74.47%
segformer	UCDB	UPS	0.01	AdaIN	28.5%	57.88%	15.12%	12.5%	40.6%
segformer	UCDB	UPS	0.01	AdaIN	63.14%	88.84%	63.18%	37.41%	75.32%
deeplabv3_resnet101	UCDB	UPS	0.01	AdaIN	65.13%	89.09%	65.7%	40.59%	77.09%
deeplabv3_resnet101	UCDB	UPS	0.01	AdaIN	56.47%	83.17%	50.13%	36.11%	70.22%
segformer	UCDB	UPS	0.01	AdaIN	48.33%	82.84%	39.81%	22.34%	61.36%
fcn_resnet50	UCDB	UPS	0.01	AdaIN	56.69%	87.71%	47.65%	34.69%	69.84%
fcn_resnet50	UCDB	UPS	0.01	AdaIN	65.82%	89.18%	65.71%	42.55%	77.76%
fcn_resnet50	UCDB	UPS	0.01	AdaIN	63.05%	89.91%	59.0%	40.25%	75.43%
segformer	UCDB	UCDB	0.001	Não Utilizado	83.36%	89.97%	82.47%	77.63%	90.84%
fcn_resnet50	UCDB	UCDB	0.001	Não Utilizado	88.51%	91.74%	88.23%	85.57%	93.89%
deeplabv3_resnet101	UCDB	UCDB	0.001	Não Utilizado	92.37%	94.92%	92.19%	90.0%	96.02%
fcn_resnet50	UCDB	UCDB	0.001	Não Utilizado	86.08%	89.86%	85.45%	82.93%	92.49%
deeplabv3_resnet101	UCDB	UCDB	0.001	Não Utilizado	94.55%	96.34%	94.49%	92.83%	97.2%
fcn_resnet50	UCDB	UCDB	0.001	Não Utilizado	90.25%	93.2%	90.22%	87.33%	94.86%
deeplabv3_resnet101	UCDB	UCDB	0.001	Não Utilizado	86.78%	90.61%	86.38%	83.34%	92.89%
segformer	UCDB	UCDB	0.001	Não Utilizado	90.28%	93.48%	89.73%	87.64%	94.88%
segformer	UCDB	UCDB	0.001	Não Utilizado	82.26%	89.79%	80.81%	76.18%	90.16%
segformer	UCDB	UPS	0.001	Não Utilizado	48.31%	89.82%	31.46%	23.65%	60.25%
fcn_resnet50	UCDB	UPS	0.001	Não Utilizado	66.64%	90.69%	66.76%	42.47%	78.27%

deeplabv3_resnet101	UCDB	UPS	0.001	Não Utilizado	62.48%	87.13%	65.2%	35.12%	74.68%
fcn_resnet50	UCDB	UPS	0.001	Não Utilizado	63.61%	90.46%	61.51%	39.47%	75.32%
deeplabv3_resnet101	UCDB	UPS	0.001	Não Utilizado	66.38%	90.29%	66.53%	42.31%	78.09%
fcn_resnet50	UCDB	UPS	0.001	Não Utilizado	64.98%	90.26%	64.06%	40.6%	76.91%
deeplabv3_resnet101	UCDB	UPS	0.001	Não Utilizado	63.67%	89.59%	63.63%	37.79%	75.71%
segformer	UCDB	UPS	0.001	Não Utilizado	59.26%	90.01%	52.57%	35.2%	71.91%
segformer	UCDB	UPS	0.001	Não Utilizado	49.95%	89.47%	53.49%	24.89%	62.23%
segformer	UCDB	UCDB	0.001	Não Utilizado	74.91%	85.72%	72.57%	66.43%	85.42%
fcn_resnet50	UCDB	UCDB	0.001	Não Utilizado	83.9%	87.67%	81.91%	82.13%	91.22%
deeplabv3_resnet101	UCDB	UCDB	0.001	Não Utilizado	90.9%	93.51%	90.72%	88.45%	95.22%
fcn_resnet50	UCDB	UCDB	0.001	Não Utilizado	79.4%	85.44%	77.78%	74.97%	88.45%
deeplabv3_resnet101	UCDB	UCDB	0.001	Não Utilizado	82.19%	88.2%	80.86%	77.5%	90.16%
fcn_resnet50	UCDB	UCDB	0.001	Não Utilizado	82.22%	86.51%	79.8%	80.36%	90.21%
deeplabv3_resnet101	UCDB	UCDB	0.001	Não Utilizado	81.05%	87.43%	78.59%	77.14%	89.47%
segformer	UCDB	UCDB	0.001	Não Utilizado	49.77%	68.98%	37.57%	42.76%	65.39%
segformer	UCDB	UCDB	0.001	Não Utilizado	79.17%	88.63%	77.25%	71.63%	88.2%
segformer	UCDB	UPS	0.001	Não Utilizado	50.52%	89.89%	36.59%	25.09%	62.79%
fcn_resnet50	UCDB	UPS	0.001	Não Utilizado	70.09%	92.27%	72.56%	45.44%	80.86%
deeplabv3_resnet101	UCDB	UPS	0.001	Não Utilizado	63.65%	86.23%	65.99%	38.73%	75.98%
fcn_resnet50	UCDB	UPS	0.001	Não Utilizado	64.58%	88.14%	67.23%	38.38%	76.52%
deeplabv3_resnet101	UCDB	UPS	0.001	Não Utilizado	63.18%	87.41%	73.78%	28.34%	74.12%
fcn_resnet50	UCDB	UPS	0.001	Não Utilizado	66.24%	88.53%	68.37%	41.83%	78.04%
deeplabv3_resnet101	UCDB	UPS	0.001	Não Utilizado	63.84%	87.45%	62.35%	41.72%	76.33%
segformer	UCDB	UPS	0.001	Não Utilizado	43.09%	74.84%	32.81%	21.63%	56.86%
segformer	UCDB	UPS	0.001	Não Utilizado	51.19%	90.53%	38.33%	24.72%	63.36%
segformer	UCDB	UCDB	0.001	Não Utilizado	85.12%	90.38%	84.49%	80.49%	91.91%
fcn_resnet50	UCDB	UCDB	0.001	Não Utilizado	79.6%	85.01%	75.22%	78.56%	88.58%
deeplabv3_resnet101	UCDB	UCDB	0.001	Não Utilizado	79.28%	85.19%	76.95%	75.7%	88.38%
fcn_resnet50	UCDB	UCDB	0.001	Não Utilizado	86.08%	89.68%	85.15%	83.4%	92.5%
deeplabv3_resnet101	UCDB	UCDB	0.001	Não Utilizado	79.01%	86.19%	77.12%	73.73%	88.18%
fcn_resnet50	UCDB	UCDB	0.001	Não Utilizado	79.14%	82.46%	76.7%	78.26%	88.33%
deeplabv3_resnet101	UCDB	UCDB	0.001	Não Utilizado	88.72%	92.02%	88.07%	86.05%	94.0%
segformer	UCDB	UCDB	0.001	Não Utilizado	32.55%	59.56%	0.89%	37.19%	43.54%
segformer	UCDB	UCDB	0.001	Não Utilizado	47.78%	62.23%	41.2%	39.91%	64.04%
segformer	UCDB	UPS	0.001	Não Utilizado	57.49%	89.7%	52.33%	30.44%	69.99%
fcn_resnet50	UCDB	UPS	0.001	Não Utilizado	49.2%	75.83%	27.66%	44.11%	63.6%
deeplabv3_resnet101	UCDB	UPS	0.001	Não Utilizado	66.25%	89.63%	69.6%	39.5%	77.75%
fcn_resnet50	UCDB	UPS	0.001	Não Utilizado	63.44%	89.96%	60.13%	40.23%	75.73%
deeplabv3_resnet101	UCDB	UPS	0.001	Não Utilizado	57.86%	88.12%	54.43%	31.02%	70.51%
fcn_resnet50	UCDB	UPS	0.001	Não Utilizado	65.8%	90.33%	65.51%	41.57%	77.6%
deeplabv3_resnet101	UCDB	UPS	0.001	Não Utilizado	64.92%	88.6%	62.74%	43.42%	77.2%
segformer	UCDB	UPS	0.001	Não Utilizado	30.27%	69.22%	1.73%	19.87%	39.45%
segformer	UCDB	UPS	0.001	Não Utilizado	38.8%	63.11%	30.4%	22.69%	53.72%
deeplabv3_resnet101	UCDB	UCDB	0.001	Não Utilizado	83.24%	87.24%	81.68%	80.82%	90.83%
segformer	UCDB	UCDB	0.001	Não Utilizado	60.33%	80.38%	52.16%	48.46%	74.32%
segformer	UCDB	UCDB	0.001	Não Utilizado	87.82%	92.34%	87.24%	83.87%	93.48%
deeplabv3_resnet101	UCDB	UCDB	0.001	Não Utilizado	91.8%	94.37%	91.71%	89.33%	95.71%
deeplabv3_resnet101	UCDB	UCDB	0.001	Não Utilizado	93.57%	95.44%	92.93%	92.35%	96.67%
segformer	UCDB	UCDB	0.001	Não Utilizado	88.91%	93.24%	88.19%	85.31%	94.1%
fcn_resnet50	UCDB	UCDB	0.001	Não Utilizado	84.84%	88.24%	83.7%	82.59%	91.78%
fcn_resnet50	UCDB	UCDB	0.001	Não Utilizado	86.94%	90.15%	86.12%	84.55%	93.0%
fcn_resnet50	UCDB	UCDB	0.001	Não Utilizado	89.67%	91.77%	87.98%	89.25%	94.55%
deeplabv3_resnet101	UCDB	UPS	0.001	Não Utilizado	64.39%	89.54%	66.46%	37.16%	76.17%
segformer	UCDB	UPS	0.001	Não Utilizado	39.59%	87.63%	10.86%	20.26%	48.9%
segformer	UCDB	UPS	0.001	Não Utilizado	55.39%	91.25%	44.86%	29.92%	67.85%
deeplabv3_resnet101	UCDB	UPS	0.001	Não Utilizado	67.07%	89.57%	67.26%	44.38%	78.6%
deeplabv3_resnet101	UCDB	UPS	0.001	Não Utilizado	62.86%	87.17%	61.11%	40.3%	75.48%
segformer	UCDB	UPS	0.001	Não Utilizado	55.21%	89.84%	48.76%	27.04%	67.59%
fcn_resnet50	UCDB	UPS	0.001	Não Utilizado	59.84%	87.0%	54.44%	38.07%	72.9%
fcn_resnet50	UCDB	UPS	0.001	Não Utilizado	66.27%	89.49%	64.49%	44.84%	78.26%
fcn_resnet50	UCDB	UPS	0.001	Não Utilizado	65.9%	88.19%	66.23%	43.28%	77.94%
deeplabv3_resnet101	UCDB	UCDB	0.01	Não Utilizado	82.82%	87.13%	80.88%	80.47%	90.58%
segformer	UCDB	UCDB	0.01	Não Utilizado	31.41%	57.87%	8.53%	27.83%	44.19%
segformer	UCDB	UCDB	0.01	Não Utilizado	91.4%	94.44%	91.18%	88.59%	95.49%
deeplabv3_resnet101	UCDB	UCDB	0.01	Não Utilizado	91.69%	94.27%	91.51%	89.29%	95.65%
deeplabv3_resnet101	UCDB	UCDB	0.01	Não Utilizado	87.3%	90.87%	86.48%	84.54%	93.2%
segformer	UCDB	UCDB	0.01	Não Utilizado	54.4%	73.55%	45.87%	43.79%	69.52%
fcn_resnet50	UCDB	UCDB	0.01	Não Utilizado	85.4%	89.14%	83.86%	83.21%	92.1%
fcn_resnet50	UCDB	UCDB	0.01	Não Utilizado	91.56%	93.99%	91.24%	89.44%	95.58%
fcn_resnet50	UCDB	UCDB	0.01	Não Utilizado	92.42%	86.66%	80.38%	80.24%	90.34%
deeplabv3_resnet101	UCDB	UPS	0.01	Não Utilizado	61.97%	88.75%	59.34%	37.84%	74.47%
segformer	UCDB	UPS	0.01	Não Utilizado	28.5%	57.88%	15.12%	12.5%	40.6%
segformer	UCDB	UPS	0.01	Não Utilizado	63.14%	88.84%	63.18%	37.41%	75.32%
deeplabv3_resnet101	UCDB	UPS	0.01	Não Utilizado	65.13%	89.09%	65.7%	40.59%	77.09%
deeplabv3_resnet101	UCDB	UPS	0.01	Não Utilizado	56.47%	83.17%	50.13%	36.11%	70.22%
segformer	UCDB	UPS	0.01	Não Utilizado	48.33%	82.84%	39.81%	22.34%	61.36%
fcn_resnet50	UCDB	UPS	0.01	Não Utilizado	56.69%	87.71%	47.65%	34.69%	69.84%
fcn_resnet50	UCDB	UPS	0.01	Não Utilizado	65.82%	89.18%	65.71%	42.55%	77.76%
fcn_resnet50	UCDB	UPS	0.01	Não Utilizado	63.05%	89.91%	59.0%	40.25%	75.43%
deeplabv3_resnet101	UPS	UCDB	0.001	AdaIN	62.75%	74.64%	55.17%	58.45%	76.79%
segformer	UPS	UCDB	0.001	AdaIN	48.97%	79.28%	54.35%	16.28%	62.29%
deeplabv3_resnet101	UPS	UCDB	0.001	AdaIN	69.18%	80.47%	63.88%	63.17%	81.52%
deeplabv3_resnet101	UPS	UCDB	0.001	AdaIN	67.55%	77.35%	62.7%	62.6%	80.43%
fcn_resnet50	UPS	UCDB	0.001	AdaIN	66.39%	81.51%	64.16%	53.5%	79.23%
fcn_resnet50	UPS	UCDB	0.001	AdaIN	60.56%	77.95%	58.3%	45.43%	74.58%
segformer	UPS	UCDB	0.001	AdaIN	43.12%	76.44%	47.84%	5.07%	53.68%
segformer	UPS	UCDB	0.001	AdaIN	45.02%	74.78%	53.58%	6.7%	55.97%
fcn_resnet50	UPS	UCDB	0.001	AdaIN	70.85%	78.76%	68.85%	64.93%	82.8%
deeplabv3_resnet101	UPS	UPS	0.001	AdaIN	84.51%	95.58%	87.14%	70.8%	91.26%
segformer	UPS	UPS	0.001	AdaIN	70.55%	93.9%	76.14%	41.6%	80.69%
deeplabv3_resnet101	UPS	UPS	0.001	AdaIN	76.69%	93.87%	80.19%	56.01%	85.88%
deeplabv3_resnet101	UPS	UPS	0.001	AdaIN	90.35%	97.12%	91.86%	82.08%	94.82%
fcn_resnet50	UPS	UPS	0.001	AdaIN	90.31%	97.0%	91.61%	82.31%	94.8%
fcn_resnet50	UPS	UPS	0.001	AdaIN	85.46%	95.38%	87.46%	73.53%	91.9%
segformer	UPS	UPS	0.001	AdaIN	62.22%	92.29%	71.88%	21.51%	71.85%
segformer	UPS	UPS	0.001	AdaIN	71.61%	93.47%	76.77%	44.6%	81.72%

fcn_resnet50	UPS	UPS	0,001	AdaIN	83.8%	95.04%	85.89%	70.47%	90.85%
deeplabv3_resnet101	UPS	UCDB	0,01	AdaIN	65.94%	78.37%	64.77%	54.69%	79.07%
segformer	UPS	UCDB	0,01	AdaIN	49.4%	78.01%	64.77%	13.86%	61.35%
deeplabv3_resnet101	UPS	UCDB	0,01	AdaIN	61.5%	78.38%	60.58%	45.56%	75.31%
deeplabv3_resnet101	UPS	UCDB	0,01	AdaIN	60.23%	76.59%	55.56%	48.55%	74.51%
fcn_resnet50	UPS	UCDB	0,01	AdaIN	63.93%	76.75%	56.48%	58.56%	77.63%
fcn_resnet50	UPS	UCDB	0,01	AdaIN	63.97%	81.95%	63.03%	46.91%	77.09%
segformer	UPS	UCDB	0,01	AdaIN	32.61%	62.95%	19.98%	14.91%	45.51%
segformer	UPS	UCDB	0,01	AdaIN	33.13%	59.71%	22.71%	16.96%	46.93%
fcn_resnet50	UPS	UCDB	0,01	AdaIN	70.08%	81.05%	66.35%	62.84%	82.16%
deeplabv3_resnet101	UPS	UPS	0,01	AdaIN	89.82%	96.71%	91.1%	81.64%	94.52%
segformer	UPS	UPS	0,01	AdaIN	74.49%	94.34%	79.38%	49.76%	84.01%
deeplabv3_resnet101	UPS	UPS	0,01	AdaIN	74.17%	93.83%	78.08%	50.59%	83.9%
deeplabv3_resnet101	UPS	UPS	0,01	AdaIN	80.1%	94.54%	82.75%	62.99%	88.35%
fcn_resnet50	UPS	UPS	0,01	AdaIN	79.82%	94.82%	82.9%	61.52%	87.99%
fcn_resnet50	UPS	UPS	0,01	AdaIN	85.93%	95.68%	87.93%	74.17%	92.18%
segformer	UPS	UPS	0,01	AdaIN	42.99%	75.43%	39.9%	13.63%	55.68%
segformer	UPS	UPS	0,01	AdaIN	48.53%	76.51%	48.17%	20.92%	62.1%
fcn_resnet50	UPS	UPS	0,01	AdaIN	79.56%	94.34%	82.88%	61.46%	87.95%
fcn_resnet50	UPS	UCDB	0,001	Não Utilizado	62.82%	77.4%	60.75%	50.33%	76.6%
fcn_resnet50	UPS	UCDB	0,001	Não Utilizado	61.4%	77.11%	54.33%	52.75%	75.52%
segformer	UPS	UCDB	0,001	Não Utilizado	41.59%	75.72%	44.66%	4.38%	52.1%
deeplabv3_resnet101	UPS	UCDB	0,001	Não Utilizado	70.32%	81.88%	69.52%	59.56%	82.24%
fcn_resnet50	UPS	UCDB	0,001	Não Utilizado	66.58%	79.93%	62.47%	57.35%	79.55%
deeplabv3_resnet101	UPS	UCDB	0,001	Não Utilizado	58.38%	75.26%	50.34%	49.53%	73.03%
deeplabv3_resnet101	UPS	UCDB	0,001	Não Utilizado	59.35%	77.33%	59.19%	41.52%	73.42%
segformer	UPS	UCDB	0,001	Não Utilizado	48.99%	77.42%	52.29%	17.25%	61.79%
segformer	UPS	UCDB	0,001	Não Utilizado	52.29%	80.47%	57.48%	18.9%	64.66%
fcn_resnet50	UPS	UPS	0,001	Não Utilizado	90.76%	96.86%	92.17%	83.25%	95.06%
fcn_resnet50	UPS	UPS	0,001	Não Utilizado	78.77%	93.98%	82.11%	60.23%	87.42%
segformer	UPS	UPS	0,001	Não Utilizado	65.34%	92.82%	72.7%	30.51%	75.74%
deeplabv3_resnet101	UPS	UPS	0,001	Não Utilizado	89.43%	96.69%	90.88%	80.72%	94.29%
fcn_resnet50	UPS	UPS	0,001	Não Utilizado	88.38%	96.51%	90.33%	78.29%	93.66%
deeplabv3_resnet101	UPS	UPS	0,001	Não Utilizado	90.8%	97.12%	92.4%	82.88%	95.08%
deeplabv3_resnet101	UPS	UPS	0,001	Não Utilizado	80.0%	94.31%	83.21%	62.48%	88.27%
segformer	UPS	UPS	0,001	Não Utilizado	72.06%	93.75%	76.96%	45.47%	82.09%
segformer	UPS	UPS	0,001	Não Utilizado	81.06%	95.5%	84.13%	63.55%	88.93%
fcn_resnet50	UPS	UCDB	0,001	Não Utilizado	60.49%	74.5%	54.97%	52.01%	74.92%
fcn_resnet50	UPS	UCDB	0,001	Não Utilizado	61.21%	76.03%	55.86%	51.73%	75.42%
segformer	UPS	UCDB	0,001	Não Utilizado	51.2%	72.4%	52.02%	25.26%	65.18%
deeplabv3_resnet101	UPS	UCDB	0,001	Não Utilizado	56.76%	76.35%	47.23%	46.71%	71.47%
fcn_resnet50	UPS	UCDB	0,001	Não Utilizado	67.98%	81.23%	66.25%	56.45%	80.5%
deeplabv3_resnet101	UPS	UCDB	0,001	Não Utilizado	60.12%	79.22%	58.34%	42.8%	74.01%
deeplabv3_resnet101	UPS	UCDB	0,001	Não Utilizado	59.83%	71.86%	55.38%	52.24%	74.51%
segformer	UPS	UCDB	0,001	Não Utilizado	42.28%	66.89%	50.09%	9.88%	54.96%
segformer	UPS	UCDB	0,001	Não Utilizado	43.33%	71.59%	50.44%	7.97%	55.09%
fcn_resnet50	UPS	UPS	0,001	Não Utilizado	77.62%	93.63%	79.81%	59.43%	86.68%
fcn_resnet50	UPS	UPS	0,001	Não Utilizado	77.21%	92.9%	81.01%	57.73%	86.34%
segformer	UPS	UPS	0,001	Não Utilizado	71.63%	93.61%	75.95%	45.34%	81.81%
deeplabv3_resnet101	UPS	UPS	0,001	Não Utilizado	74.0%	93.74%	76.54%	51.72%	83.89%
fcn_resnet50	UPS	UPS	0,001	Não Utilizado	82.86%	96.01%	84.79%	67.8%	90.18%
deeplabv3_resnet101	UPS	UPS	0,001	Não Utilizado	83.02%	94.66%	87.17%	67.24%	90.27%
deeplabv3_resnet101	UPS	UPS	0,001	Não Utilizado	69.0%	90.9%	70.05%	46.05%	80.23%
segformer	UPS	UPS	0,001	Não Utilizado	69.0%	92.71%	74.52%	39.78%	79.51%
segformer	UPS	UPS	0,001	Não Utilizado	65.17%	98.23%	73.55%	28.74%	75.3%
fcn_resnet50	UPS	UCDB	0,001	Não Utilizado	59.87%	74.11%	61.1%	43.8%	74.1%
fcn_resnet50	UPS	UCDB	0,001	Não Utilizado	59.36%	75.95%	59.39%	42.73%	73.58%
segformer	UPS	UCDB	0,001	Não Utilizado	47.49%	74.03%	53.7%	14.74%	60.21%
deeplabv3_resnet101	UPS	UCDB	0,001	Não Utilizado	59.54%	77.37%	50.91%	50.33%	73.89%
fcn_resnet50	UPS	UCDB	0,001	Não Utilizado	61.84%	80.92%	53.04%	51.56%	75.6%
deeplabv3_resnet101	UPS	UCDB	0,001	Não Utilizado	41.65%	65.84%	25.8%	33.3%	56.79%
deeplabv3_resnet101	UPS	UCDB	0,001	Não Utilizado	56.34%	78.67%	57.98%	32.37%	70.12%
segformer	UPS	UCDB	0,001	Não Utilizado	33.91%	60.42%	23.05%	18.27%	47.9%
segformer	UPS	UCDB	0,001	Não Utilizado	29.62%	43.5%	38.8%	6.57%	42.96%
fcn_resnet50	UPS	UPS	0,001	Não Utilizado	67.04%	92.72%	74.4%	34.0%	77.43%
fcn_resnet50	UPS	UPS	0,001	Não Utilizado	78.38%	93.82%	81.37%	59.93%	87.16%
segformer	UPS	UPS	0,001	Não Utilizado	69.98%	93.51%	75.61%	40.83%	80.25%
deeplabv3_resnet101	UPS	UPS	0,001	Não Utilizado	73.83%	93.49%	77.26%	50.72%	83.71%
fcn_resnet50	UPS	UPS	0,001	Não Utilizado	77.73%	93.93%	80.3%	58.95%	86.71%
deeplabv3_resnet101	UPS	UPS	0,001	Não Utilizado	66.53%	89.79%	68.56%	41.26%	78.13%
deeplabv3_resnet101	UPS	UPS	0,001	Não Utilizado	80.72%	94.23%	83.73%	64.21%	88.79%
segformer	UPS	UPS	0,001	Não Utilizado	45.65%	74.9%	43.29%	18.75%	59.12%
segformer	UPS	UPS	0,001	Não Utilizado	31.89%	51.74%	38.01%	5.92%	44.82%
deeplabv3_resnet101	UPS	UCDB	0,001	Não Utilizado	62.75%	74.64%	55.17%	58.45%	76.79%
segformer	UPS	UCDB	0,001	Não Utilizado	49.97%	79.28%	54.35%	16.28%	62.29%
deeplabv3_resnet101	UPS	UCDB	0,001	Não Utilizado	69.18%	80.47%	63.88%	63.17%	81.52%
deeplabv3_resnet101	UPS	UCDB	0,001	Não Utilizado	67.55%	77.35%	62.7%	62.6%	80.43%
fcn_resnet50	UPS	UCDB	0,001	Não Utilizado	66.39%	81.51%	64.16%	53.5%	79.23%
fcn_resnet50	UPS	UCDB	0,001	Não Utilizado	60.56%	77.95%	58.3%	45.43%	74.58%
segformer	UPS	UCDB	0,001	Não Utilizado	43.12%	76.44%	47.84%	5.07%	53.68%
segformer	UPS	UCDB	0,001	Não Utilizado	45.02%	74.78%	53.58%	6.7%	55.97%
fcn_resnet50	UPS	UCDB	0,001	Não Utilizado	70.83%	78.76%	68.85%	64.93%	82.8%
deeplabv3_resnet101	UPS	UPS	0,001	Não Utilizado	84.51%	95.58%	87.14%	70.8%	91.26%
segformer	UPS	UPS	0,001	Não Utilizado	70.55%	93.9%	76.14%	41.6%	80.69%
deeplabv3_resnet101	UPS	UPS	0,001	Não Utilizado	76.69%	93.87%	80.19%	56.01%	85.88%
deeplabv3_resnet101	UPS	UPS	0,001	Não Utilizado	90.35%	97.12%	91.86%	82.08%	94.82%
fcn_resnet50	UPS	UPS	0,001	Não Utilizado	90.31%	97.0%	91.61%	82.31%	94.8%
fcn_resnet50	UPS	UPS	0,001	Não Utilizado	85.46%	95.38%	87.48%	73.53%	91.9%
segformer	UPS	UPS	0,001	Não Utilizado	62.22%	93.29%	71.88%	21.51%	71.85%
segformer	UPS	UPS	0,001	Não Utilizado	71.61%	93.47%	76.77%	44.6%	81.72%
fcn_resnet50	UPS	UPS	0,001	Não Utilizado	83.8%	95.04%	85.89%	70.47%	90.85%
deeplabv3_resnet101	UPS	UCDB	0,01	Não Utilizado	65.94%	78.37%	64.77%	54.69%	79.07%
segformer	UPS	UCDB	0,01	Não Utilizado	49.4%	78.01%	54.77%	13.86%	61.35%
deeplabv3_resnet101	UPS	UCDB	0,01	Não Utilizado	61.5%	78.38%	60.58%	45.56%	75.31%
deeplabv3_resnet101	UPS	UCDB	0,01	Não Utilizado	60.23%	76.59%	55.56%	48.55%	74.51%
fcn_resnet50	UPS	UCDB	0,01	Não Utilizado	63.93%	76.75%	56.48%	58.56%	77.63%

fcn_resnet50	UPS	UCDB	0,01	Não Utilizado	63.97%	81.95%	63.03%	46.91%	77.09%
segformer	UPS	UCDB	0,01	Não Utilizado	32.61%	62.95%	19.86%	14.91%	45.51%
segformer	UPS	UCDB	0,01	Não Utilizado	33.13%	59.71%	22.71%	16.96%	46.93%
fcn_resnet50	UPS	UCDB	0,01	Não Utilizado	70.08%	81.05%	66.35%	62.84%	82.16%
deeplabv3_resnet101	UPS	UPS	0,01	Não Utilizado	89.82%	96.71%	91.1%	81.64%	94.52%
segformer	UPS	UPS	0,01	Não Utilizado	74.49%	94.34%	79.38%	49.76%	84.01%
deeplabv3_resnet101	UPS	UPS	0,01	Não Utilizado	74.17%	93.83%	78.08%	50.59%	83.9%
deeplabv3_resnet101	UPS	UPS	0,01	Não Utilizado	80.1%	94.54%	82.75%	62.99%	88.35%
fcn_resnet50	UPS	UPS	0,01	Não Utilizado	79.62%	94.42%	82.9%	61.52%	87.99%
fcn_resnet50	UPS	UPS	0,01	Não Utilizado	85.93%	95.68%	87.93%	74.17%	92.18%
segformer	UPS	UPS	0,01	Não Utilizado	42.99%	75.43%	39.9%	13.63%	55.68%
segformer	UPS	UPS	0,01	Não Utilizado	48.53%	76.51%	48.17%	20.92%	62.1%
fcn_resnet50	UPS	UPS	0,01	Não Utilizado	79.56%	94.34%	82.88%	61.46%	87.95%

APÊNDICE 3 - TABELAS DETALHADAS

Resumo por Backbone

AR	Overall Jaccard Index_count	Overall Jaccard Index_mean	Overall Jaccard Index_std	Overall Dice Coefficient_mean	Overall Dice Coefficient_std	Jaccard Index - Vegetation_mean	Jaccard Index - Vegetation_std	Jaccard Index - Grass_mean	Jaccard Index - Grass_std	Jaccard Index - Others_mean	Jaccard Index - Others_std
ResNet-101	8	25,64	9,44	33,57	13,66	5,22	9,74	11,03	16,63	60,67	6,04
Swin-Base	4	36,73	23,2	45,5	26,59	31,2	41,81	14,5	20,22	64,5	9,4
Swin-Tiny	2	36,02	11,7	44,95	12,18	44,32	38,1	0,5	0,71	63,22	3,71

Fonte: Elaborado pelo autor

Resumo por Cenário de Domínio

Domain Scenario	Overall Jaccard Index_count	Overall Jaccard Index_mean	Overall Jaccard Index_std	Overall Dice Coefficient_mean	Overall Dice Coefficient_std	Jaccard Index - Vegetation_mean	Jaccard Index - Vegetation_std	Jaccard Index - Grass_mean	Jaccard Index - Grass_std	Jaccard Index - Others_mean	Jaccard Index - Others_std
Cross-domain (UCDB) $\uparrow$	4	29,45	7,95	38,34	11,41	13,45	17,37	8,27	9,04	66,62	2,59
Cross-domain (UPS $\rightarrow$ U)	3	17,19	0	22,69	0	0	0	0	0	51,58	0
Intra-domain (UCDB)	4	47,75	14,22	60,3	13,34	50,35	34,78	28,53	20,01	64,36	6,16

Intra-domain (UPS)	3	21,23	0	25,94	0	0	0	0	0	63,7	0
--------------------	---	-------	---	-------	---	---	---	---	---	------	---

Fonte: Elaborado pelo autor

Resultados Detalhados

Network Architecture	Backbone	Domain Scenario	Overall Jaccard	Overall Dice Coefficient	Jaccard Index - Vegetation	Jaccard Index - Grass	Jaccard Index - Others
swin_base_TrainUCDB_Cut_ValidUPS_if16_lr3e-05_adamw_self	Swin-Base	Intra-domain (UCDB)	68	79,36	88,38	42,89	72,71
tribranch_deeplabv3_resnet101_if16_lr0.001_sgd_self	ResNet-101	Intra-domain (UCDB)	44,02	59,11	23,05	43,81	65,2
tribranch_deeplabv3_resnet101_if16_lr0.01_sgd_self	ResNet-101	Intra-domain (UCDB)	34,7	49,17	18,72	26,42	58,95
tribranch_swin_tiny_if16_lr0.0001_adamw_self	Swin-Tiny	Intra-domain (UCDB)	44,29	53,56	71,26	1,01	60,6
swin_base_TrainUCDB_Cut_ValidUPS_if16_lr3e-05_adamw_self	Swin-Base	Cross-domain (UCDB → UPS)	40,51	54	36,42	15,1	70
tribranch_deeplabv3_resnet101_if16_lr0.001_sgd_self	ResNet-101	Cross-domain (UCDB → UPS)	27,96	36,41	0	17,03	66,85
tribranch_deeplabv3_resnet101_if16_lr0.01_sgd_self	ResNet-101	Cross-domain (UCDB → UPS)	21,58	26,6	0	0,96	63,79
tribranch_swin_tiny_if16_lr0.0001_adamw_self	Swin-Tiny	Cross-domain (UCDB → UPS)	27,74	36,34	17,38	0	65,84

to_ups_swin_base_80_self	Swin-Base	Cross-domain (UPS→U CDB)	17,1 9	22,69	0	0	51,58
to_ups_swin_base_80_self	Swin-Base	Intra-domain (UPS)	21,2 3	25,94	0	0	63,7
to_ups_deeplabv3_resnet101_self	ResNet-101	Cross-domain (UPS→U CDB)	17,1 9	22,69	0	0	51,58
to_ucdb_deeplabv3_resnet101_self	ResNet-101	Cross-domain (UPS→U CDB)	17,1 9	22,69	0	0	51,58
to_ups_deeplabv3_resnet101_self	ResNet-101	Intra-domain (UPS)	21,2 3	25,94	0	0	63,7
to_ucdb_deeplabv3_resnet101_self	ResNet-101	Intra-domain (UPS)	21,2 3	25,94	0	0	63,7

Fonte: Elaborado pelo autor