

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Um estudo experimental sobre aplicação prática de
métodos de *deep learning* à segmentação semântica de
áreas navegáveis e não navegáveis em tempo real

William Luis Alves Ferreira



São Carlos – SP

Um estudo experimental sobre aplicação prática de métodos de *deep learning* à segmentação semântica de áreas navegáveis e não navegáveis em tempo real

William Luis Alves Ferreira

Orientador: Prof. Dr. Fernando Osório

Coorientador: Me. Felipe Manfio Barbosa

Monografia final de conclusão de curso do Departamento de Sistemas de Computação do Instituto de Ciências Matemáticas e de Computação – ICMC-USP, para obtenção do título de Bacharel em Engenharia de Computação.

Área de Concentração: Ciência de Dados

USP – São Carlos
Outubro de 2024

Ferreira, William Luis Alves

Um estudo experimental sobre aplicação prática
de métodos de *deep learning* à segmentação semântica
de áreas navegáveis e não navegáveis em tempo real /
William Luis Alves Ferreira. - São Carlos - SP, 2024.

137 p.; 29,7 cm.

Orientador: Fernando Osório.

Coorientador: Felipe Manfio Barbosa.

Monografia (Graduação) - Instituto de Ciências
Matemáticas e de Computação (ICMC/USP), São Carlos -
SP, 2024.

1. Aprendizagem profunda. 2. Redes Neurais Convolucionais. 3. Aprendizagem supervisionada.
4. Aprendizagem semi-supervisionada. 5. Adaptação de domínio não supervisionada. 6. Aprendizagem por consistência. 7. Direção Autônoma. 8. Transformadores de Visão. 9. Dataset CityScapes. 10. Dataset GTA5
- . I. Osório, Fernando. II. Instituto de Ciências Matemáticas e de Computação (ICMC/USP). III. Título.

*Dedico este trabalho à minha mãe, **Elisabete Alves Laurindo**, e aos meus irmãos, pelo apoio emocional e incentivo ao longo desta jornada.*

*Também dedico ao meu cônjuge, **Roger Gregório Marcondes**, cujo apoio foi fundamental para a concretização deste sonho, graduado da Universidade de São Paulo.*

AGRADECIMENTOS

Agradeço aos meus orientadores, **Dr. Fernando Osório e Me. Felipe Manfio Barbosa**, que, com sabedoria e paciência, me guiaram ao longo deste percurso. Suas contribuições foram essenciais para o desenvolvimento desta monografia.

Agradeço ao **Dr. Wilk de Oliveira**, meu orientador de iniciação científica, pela oportunidade de vivenciar o trabalho acadêmico.

Sou igualmente grato aos professores do curso de Engenharia de Computação, oferecido pelo Instituto de Ciências Matemáticas e de Computação / Escola de Engenharia de São Carlos - USP, por ampliarem meus horizontes e aprofundarem meus conhecimentos.

Também agradeço aos meus amigos e colegas de curso – **Axel Costa, Bruner Eduardo Augusto Albrecht, Caio Botelho Naves, Carlos Roberto dos Santos Junior, Clayton Miccas Junior e Flavio Alves Conti** – pela troca de experiências e pelo suporte nas horas de dificuldade. Vocês tornaram essa caminhada mais leve e prazerosa.

Minha gratidão se estende à **Aline Fernanda da Conceição**, pelo apoio acadêmico, profissional e emocional ao longo desta jornada.

A todos, meu sincero obrigado e meus votos de uma vida repleta de sucesso.

RESUMO

FERREIRA, W. **Um estudo experimental sobre aplicação prática de métodos de *deep learning* à segmentação semântica de áreas navegáveis e não navegáveis em tempo real.** 2024. 137 f. Monografia (Graduação) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

A direção autônoma tem potencial para transformar o transporte urbano ao aumentar a **segurança e eficiência**, mas enfrenta desafios relacionados à interpretação do ambiente em **tempo real**. A segmentação semântica é essencial nesse processo, permitindo que sistemas autônomos identifiquem áreas navegáveis e obstáculos. No Brasil, há desafios específicos devido às particularidades das vias e à dificuldade de generalizar modelos treinados com dados internacionais, como o conjunto **CityScapes**. Este trabalho propõe um modelo **eficiente e leve** de segmentação semântica, baseado no **LiteSeg** e integrado às arquiteturas **CCT** e **PixMatch**, adaptado ao contexto brasileiro. O conjunto de dados **CityScapesBrazil**, com 21.485 imagens, foi gerado para validar o contexto de cenas urbanas brasileiras. O modelo alcançou **95,4% de IoU** na classe “*road*” e **30 FPS em alta resolução**, comprovando sua eficiência e capacidade de operação em tempo real. A proposta oferece uma solução robusta e prática para a navegação autônoma em vias brasileiras, com alta performance e compatível com dispositivos de recursos limitados. O repositório com os artefatos desta monografia é encontrado em Light_Domain_Adaptation@illiamw.

Palavras-chave: Aprendizagem profunda, Redes Neurais Convolucionais, Aprendizagem supervisionada, Aprendizagem semi-supervisionada, Adaptação de domínio não supervisionada, Aprendizagem por consistência, Direção Autônoma, Transformadores de Visão, Dataset CityScapes, Dataset GTA5 .

ABSTRACT

FERREIRA, W. **Um estudo experimental sobre aplicação prática de métodos de deep learning à segmentação semântica de áreas navegáveis e não navegáveis em tempo real.** 2024. 137 f. Monografia (Graduação) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Autonomous driving has the potential to transform urban transportation by enhancing **safety** and **efficiency**, but it faces challenges related to **real-time** environmental interpretation. Semantic segmentation is essential in this process, enabling autonomous systems to identify navigable areas and obstacles. In Brazil, specific challenges arise due to the particular characteristics of roads and the difficulty of generalizing models trained on international datasets, such as **CityScapes**. This study proposes an efficient and **LiteSeg** semantic segmentation model, based on LiteSeg and integrated with **CCT** and **PixMatch** architectures, adapted to the Brazilian context. The **CityScapesBrazil** dataset, comprising 21,485 images, was created to validate scenes from Brazilian urban environments. The model achieved **95.4% IoU** in the "road" class and **30 FPS at high resolution**, demonstrating its efficiency and real-time operation capability. This proposal offers a robust and practical solution for autonomous navigation on Brazilian roads, delivering high performance and compatibility with resource-constrained devices. The repository of the artifacts of this monograph is found in Light_Domain_Adaptation@illiamw.

Key-words: Deep Learning, Convolutional Neural Networks (CNNs), Supervised Learning, Semi-supervised Learning, Unsupervised Domain Adaptation, Consistency Learning, Autonomous Driving, Vision Transformers (ViTs), CityScapes Dataset, GTA5 Dataset.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de segmentação semântica CityScapes	24
Figura 2 – Convolução separável em profundidade em comparação com convolução padrão.	34
Figura 3 – Arquitetura do modelo LiteSeg.	35
Figura 4 – Módulo DASPP.	37
Figura 5 – Representação gráfica do processo de adaptação de domínio não supervisionado.	40
Figura 6 – Arquitetura e representação das perdas do PixMatch.	43
Figura 7 – Exemplo das perturbações geradas, para este trabalho apenas o Augmentation será aplicado.	44
Figura 8 – Ilustração dos hiperespaços para definir a classificação de dados rotulados e não rotulados.	44
Figura 9 – Arquitetura e estratégia de treinamento com perda supervisionada e não supervisionada do CCT.	45
Figura 10 – CCT aplicado em múltiplos domínios.	51
Figura 11 – Diagrama do modelo PixMatchLite para os treinamento GTA5 para CityScapes, GTA5 para CityScapesBrazil e CityScapes para CityScapesBrazil	55
Figura 12 – Diagrama do modelo CCTLite para os treinamento GTA5 para CityScapes, GTA5 para CityScapesBrazil e CityScapes para CityScapesBrazil	56
Figura 13 – Diagrama do modelo CCTLite para os treinamento GTA5 e CityScapes para CityScapesBrazil	56
Figura 14 – Comparando conjunto de validação CityScapes como Supervisionado em semi-supervisionado e semi-supervisionado em múltiplos domínios, além da estratégia não supervisionada em adaptação de domínio.	60
Figura 15 – Comparando conjunto de validação GTA5 treinado de forma supervisionada para CityScapes e CityScapesBrazil como semi-supervisionado e múltiplos domínios.	61
Figura 16 – Comparando conjunto de validação CityScapesBrazil treinado de forma não supervisionado em semi-supervisionado a partir do CityScapes e GTA5, e múltiplos domínios.	61
Figura 17 – Comparando conjunto de validação CityScapes como Supervisionado (PIXCB) e Não Supervisionado (PIXGC).	62
Figura 18 – Comparando conjunto de validação GTA5 como Supervisionado adaptando os domínios CityScapes e CityScapesBrazil.	62

Figura 19 – Comparando conjunto de validação CityScapesBrazil como Não Supervisionado adaptado a partir dos domínios CityScapes e GTA5.	62
Figura 20 – Exemplos do conjunto de teste do CityScapesBrazil na primeira coluna, e suas predições nos modelos CCTCB, CTTMDGCB, PIXCB e PIXGB nas respectivas colunas seguintes.	67
Figura 21 – Arquiteturas ao longo do tempo.	87
Figura 22 – Taxonomia das estratégias de treinamento semi-supervisionado e não supervisionado.	88
Figura 23 – Efeito do stride na convolução.	89
Figura 24 – Convolução bidimensional em imagens.	90
Figura 25 – Aplicações de diferentes kernels em uma imagem.	91
Figura 26 – Funcionamento do MaxPooling.	91
Figura 27 – Aplicação de pooling em uma imagem de bordas.	92
Figura 28 – Convolução transposta.	92
Figura 29 – Projeção das convoluções padrão e dilatada lado a lado.	93
Figura 30 – Ilustração da convolução dilatada bidimensional.	93
Figura 31 – Agrupamento de pirâmide espacial dilatada (ASPP).	94
Figura 32 – Ilustra diferentes topologias com e sem ASPP.	94
Figura 33 – Arquitetura DeepLabV3+ estendendo DeepLabV3.	95
Figura 34 – Exemplos do conjunto de dados obtido.	97
Figura 35 – Trajetos percorridos no streetview, nas cidades de Fortaleza, Porto Alegre, Curitiba, Rio de Janeiro e São Paulo.	98
Figura 36 – Trajetória por rodovias de nordeste a sul do Brasil.	99
Figura 37 – Amostra com 50 imagens do conjunto de validação do CityScapesBrazil para 100 épocas no modelo PixMatch: GTA5 para CityScapes	105
Figura 38 – Amostra com 50 imagens do conjunto de validação do CityScapesBrazil para 100 épocas no modelo PixMatch: CityScapes para CityScapesBrazil	106
Figura 39 – Amostra com 50 imagens do conjunto de validação do CityScapesBrazil para 100 épocas no modelo PixMatch: GTA5 para CityScapesBrazil	107
Figura 40 – Amostra com 50 imagens do conjunto de validação do CityScapesBrazil para 40 épocas no modelo CCT: GTA5 para CityScapes	108
Figura 41 – Amostra com 50 imagens do conjunto de validação do CityScapesBrazil para 40 épocas no modelo CCT: CityScapes para CityScapesBrazil	109
Figura 42 – Amostra com 50 imagens do conjunto de validação do CityScapesBrazil para 40 épocas no modelo CCT: GTA5 para CityScapesBrazil	110
Figura 43 – Amostra com 50 imagens do conjunto de validação do CityScapesBrazil para 40 épocas no modelo CCT: GTA5 para CityScapes e CityScapesBrazil	111

Figura 44 – Exemplos do conjunto de validação do CityScapesBrazil na primeira coluna, e suas predições dos modelos CCTCB, CCTGC, CCTGB e CTTMDGCB, nas respectivas colunas seguintes.	112
Figura 45 – Exemplos do conjunto de validação do CityScapes na primeira coluna, e suas predições nos modelos CCTCB,CCTGC, CCTGB e CTTMDGCB, nas respectivas colunas seguintes.	113
Figura 46 – Exemplos do conjunto de validação do CityScapesBrazil na primeira coluna, e suas predições nos modelos PIXCB, PIXGB, e PIXGC, nas respectivas colunas seguintes.	114
Figura 47 – Exemplos do conjunto de validação do CityScapes na primeira coluna, e suas predições nos modelos PIXCB, PIXGB, e PIXGC, nas respectivas colunas seguintes.	115
Figura 48 – Métricas Padrão - PixMatch: GTA5 para CityScapes	117
Figura 49 – Exemplos de predição ao longo das épocas para o modelo PixMatch: GTA5 para CityScapes, com amostras a cada 20 épocas, primeira linha imagem original, segunda linha rótulo verdadeira, demais linhas são as predições . . .	118
Figura 50 – Métricas Padrão - PixMatch: GTA5 para CityScapes (19 classes)	119
Figura 51 – Exemplos de predição ao longo das épocas para o modelo PixMatch: GTA5 para CityScapes (19 classes), com amostras a cada 20 épocas, primeira linha imagem original, segunda linha rótulo verdadeira, demais linhas são as predições	120
Figura 52 – Métricas Padrão - PixMatch: CityScape para CityScapesBrazil	121
Figura 53 – Exemplos de predição ao longo das épocas para o modelo PixMatch: CityS- cape para CityScapesBrazil, com amostras a cada 20 épocas, primeira linha imagem original, segunda linha rótulo verdadeira, demais linhas são as predi- ções	122
Figura 54 – Métricas Padrão - PixMatch: GTA5 para CityScapesBrazil	123
Figura 55 – Exemplos de predição ao longo das épocas para o modelo PixMatch: GTA5 para CityScapesBrazil, com amostras a cada 20 épocas, primeira linha ima- gem original, segunda linha rótulo verdadeira, demais linhas são as predições	124
Figura 56 – Métricas Padrão - CCT: GTA5 para CityScapes	125
Figura 57 – Exemplos de predição ao longo das épocas para o modelo CCT: GTA5 para CityScapes, conjunto GTA5, com amostras a cada 10 épocas, primeira coluna imagem, segunda coluna predição e terceira coluna rótulo verdadeiro . . .	126
Figura 58 – Exemplos de predição ao longo das épocas para o modelo CCT: GTA5 para CityScapes, conjunto CityScapes, com amostras a cada 10 épocas, primeira coluna imagem, segunda coluna predição e terceira coluna rótulo verdadeiro	127
Figura 59 – Métricas Padrão - CCT: CityScapes para CityScapesBrazil	128

Figura 60 – Exemplos de predição ao longo das épocas para o modelo CCT: CityScapes para CityScapesBrazil, conjunto CityScapes, com amostras a cada 10 épocas, primeira coluna imagem, segunda coluna predição e terceira coluna rótulo verdadeiro	129
Figura 61 – Exemplos de predição ao longo das épocas para o modelo CCT: CityScapes para CityScapesBrazil, conjunto CityScapesBrazil, com amostras a cada 10 épocas, primeira coluna imagem, segunda coluna predição e terceira coluna rótulo verdadeiro	130
Figura 62 – Métricas Padrão - CCT: GTA5 para CityScapesBrazil	131
Figura 63 – Exemplos de predição ao longo das épocas para o modelo CCT: GTA5 para CityScapesBrazil, conjunto GTA5, com amostras a cada 10 épocas, primeira coluna imagem, segunda coluna predição e terceira coluna rótulo verdadeiro	132
Figura 64 – Exemplos de predição ao longo das épocas para o modelo CCT: GTA5 para CityScapesBrazil, conjunto CityScapesBrazil, com amostras a cada 10 épocas, primeira coluna imagem, segunda coluna predição e terceira coluna rótulo verdadeiro	133
Figura 65 – Métricas Padrão - CCTMD: GTA5 para CityScapes e CityScapesBrazil . . .	134
Figura 66 – Exemplos de predição ao longo das épocas para o modelo CCTMD: GTA5 para CityScapes e CityScapesBrazil, conjunto GTA5, com amostras a cada 10 épocas, primeira coluna imagem, segunda coluna predição e terceira coluna rótulo verdadeiro	135
Figura 67 – Exemplos de predição ao longo das épocas para o modelo CCTMD: GTA5 para CityScapes e CityScapesBrazil, conjunto CityScapes, com amostras a cada 10 épocas, primeira coluna imagem, segunda coluna predição e terceira coluna rótulo verdadeiro	136
Figura 68 – Exemplos de predição ao longo das épocas para o modelo CCTMD: GTA5 para CityScapes e CityScapesBrazil, conjunto CityScapesBrazil, com amostras a cada 10 épocas, primeira coluna imagem, segunda coluna predição e terceira coluna rótulo verdadeiro	137

LISTA DE TABELAS

Tabela 1 – Separação dos subconjuntos de dados de treino, validação e teste, conjuntos CityScapes e GTA5.	53
Tabela 2 – Separação dos subconjuntos de dados de treino, validação e teste, conjunto CityScapesBrazil.	53
Tabela 3 – Hiperparâmetros de treinamento para o modelo LiteSeg.	57
Tabela 4 – Hiperparâmetros peso da perda supervisionada e não supervisionada, além do batch size e época limite.	57
Tabela 5 – Hiperparâmetros das funções de perda, assim como a quantidade de decodificadores auxiliares, função de perda, e peso da perda supervisionada e não supervisionada, além do batch size e época limite.	57
Tabela 6 – Siglas dos modelos avaliados, descrição e respectivos apêndices.	59
Tabela 7 – Entropia de Shannon para CityScapesBrazil aplicado aos modelos CCTGC, CCTCB e CTTMDGCB.	63
Tabela 8 – MIoU e IoU sobre as classes Navegável, Inavegável e Obstáculos para CityScapes conjunto de validação aplicado aos modelos CCTGB, CCTGC, CCTCB e CTTMDGCB.	63
Tabela 9 – Entropia de Shannon para CityScapesBrazil aplicado aos modelos PixGC e PixCB.	64
Tabela 10 – MIoU e IoU sobre as classes Navegável, Inavegável e Obstáculos para CityScapes conjunto de validação aplicado aos modelos PixGB, PixGC e PixCB.	64
Tabela 11 – Conjunto CityScape aplicada como <i>source (label)</i> e <i>target(unlabel)</i> nos modelos avaliados sobre as métricas MIoU e Shannon no conjunto de validação	65
Tabela 12 – Conjunto GTA5 aplicada como <i>source</i> nos modelos avaliados sobre as métricas MIoU e Shannon no conjunto de validação e teste	66
Tabela 13 – Resultado obtido para o conjunto CityScapeBrazil avaliado pela entropia de Shannon, sobre o conjunto de teste	66
Tabela 14 – MIoU e IoU por classe para o modelo PIXGC para 19 classes CityScapes conjunto de teste	68
Tabela 15 – MIoU e IoU por classe para o modelo PIXGC para 7 categorias CityScapes conjunto de teste	68
Tabela 16 – MIoU e IoU por classe para o modelo PIXGC para 19 classes agrupamento Navegável, Inavegável e Obstáculos para CityScapes conjunto de teste	68

Tabela 17 – Especificações dos sistema de aplicação dos modelos.	69
Tabela 18 – Tempo de predição em Frames por Segundo (FPS), métrica de computação em Flops e número de parâmetros treináveis dos modelos.	69
Tabela 19 – Tempo de treinamento aproximado por modelo.	69
Tabela 20 – Volumetria de amostras por trajeto.	97
Tabela 21 – Nome e codificação das classes padrão e para esta monografia	101
Tabela 22 – Comparação do estado da arte de segmentação semântica para o conjunto CityScape para teste, sobre MIoU e classe Navegável correspondente a classe padrão “road”, id = 7.	102
Tabela 23 – Resultados obtidos para os conjuntos CityScapes, GTA5 e CityScapesBrazil, sobre o conjunto de teste e validação considerando as melhores épocas verificadas em validação ao decorrer do treinamento.	103
Tabela 24 – Média da intersecção da união, e IoU para cada uma das classes Navegável, Inavegável e Obstáculos, sobre o conjunto de validação para as melhores épocas respectivas a cada modelo e base.	104
Tabela 25 – Média da intersecção da união, e IoU para cada uma das classes Navegável, Inavegável e Obstáculos, sobre o conjunto de teste para as melhores épocas respectivo a cada modelo e base.	104

SUMÁRIO

1	INTRODUÇÃO	21
2	OBJETIVOS	23
3	REVISÃO BIBLIOGRÁFICA	25
3.1	Segmentação Semântica: CNNs, ViT, Aprendizado e Adaptação de Domínio	25
3.2	Aprendizado semi-supervisionado e adaptação de domínio	26
4	TRABALHOS RELACIONADOS	29
4.1	Execução em tempo real	29
4.2	Aprendizagem semi-supervisionado e adaptação de domínio não supervisionado	30
5	FUNDAMENTAÇÃO TEÓRICA	31
5.1	Segmentação Semântica para Direção Autônoma	31
5.2	Aprendizagem Profunda	32
5.3	Transformadores de visão	32
5.4	Rede Neural Convolucional	33
5.5	Deeper Atrous Spatial Pyramid Pooling (DASPP)	33
5.6	Modelo LiteSeg	34
5.6.1	<i>Codificador</i>	35
5.6.2	<i>Backbone</i>	36
5.6.3	<i>DASPP</i>	37
5.6.4	<i>Decodificador</i>	38
5.7	Treinamento Semi-Supervisionado e não supervisionado	38
5.8	Estratégia de treinamento por consistência	40
5.9	PixMatch: Unsupervised Domain Adaptation via Pixelwise Consistency Training	41
5.9.1	<i>Arquitetura e Treinamento</i>	41
5.9.2	<i>Função de Perturbação</i>	42
5.10	Semi-Supervised Semantic Segmentation with Cross-Consistency Training	44
5.10.1	<i>Arquitetura e treinamento</i>	45

5.10.2	Funções de Perturbação	46
5.10.2.1	Perturbações Baseadas em Predição	46
5.10.2.2	Perturbações Baseadas em Características	47
6	METODOLOGIA	49
6.1	Coleta de dados cenários urbanos brasileiros	49
6.2	Uso dos artigos base: LiteSeg, PixMatch, CCT	50
6.3	Avaliação de resultados	51
6.4	Escolha e preparação dos modelos de redes neurais	53
6.4.1	<i>Considerações iniciais</i>	54
6.5	Arquitetura dos modelos	55
6.6	Parâmetros do experimento	57
7	RESULTADOS	59
7.1	Treinamento e validação dos modelos escolhidos	59
7.1.1	<i>Treinamento das variações CCT</i>	59
7.1.2	<i>Treinamento das variações PixMatch</i>	61
7.1.3	<i>Teste de variação entre os domínios CityScapes e CityScapesBrazil</i>	63
7.1.3.1	<i>Testando a variação dos domínios CityScapes e CityScapesBrazil nas variações CCT</i>	63
7.1.3.2	<i>Testando a variação dos domínios CityScapes e CityScapesBrazil nas variações PixMatch</i>	64
7.1.3.3	<i>Conclusão do Teste de Domínio</i>	64
7.2	Resultados em Conjunto Teste e Validação	65
7.2.1	<i>Conjunto Rotulado</i>	65
7.2.2	<i>Conjunto Não Rotulado</i>	66
7.2.3	<i>Considerações Finais</i>	66
7.3	Performance e Sistema de aplicação	69
8	CONCLUSÃO	71
8.1	Trabalhos Futuros	72
REFERÊNCIAS		73
APÊNDICE A	PESQUISA COMPLEMENTAR SOBRE MODELOS	
	VIT	81
A.1	Modelos baseados em Transformadores de Visão	81
APÊNDICE B	ESTUDOS COMPLEMENTARES	85
B.1	Segmentação Semântica para Direção Autônoma	85
B.2	Rede Neural Convolucional	89

B.3	Deeper Atrous Spatial Pyramid Pooling (DASPP)	92
APÊNDICE C	CARACTERÍSTICAS DA COLETA DO DATASET CITYSCAPESBRAZIL	97
APÊNDICE D	AGRUPAMENTO DE CLASSES CITYSCAPES, CITYSCAPESBRAZIL E GTA5, E COMPARAÇÃO COM O ESTADO DA ARTE	101
D.1	Comparação com o estado da arte	101
APÊNDICE E	RESULTADO GERAL	103
APÊNDICE F	INSPEÇÃO VISUAL	105
APÊNDICE G	EVOLUÇÃO DOS MODELOS	117

Capítulo 1

INTRODUÇÃO

Nos últimos anos, a direção autônoma tem se consolidado como uma área de intensa pesquisa, prometendo transformar o transporte urbano ao oferecer maior **segurança, eficiência e autonomia**. Um dos principais desafios para o desenvolvimento de veículos autônomos é a capacidade de interpretar corretamente o ambiente ao redor em **tempo real**, garantindo tomadas de decisão rápidas e precisas. A segmentação semântica desempenha um papel crucial nesse processo, permitindo que o sistema identifique **áreas navegáveis e obstáculos nas vias**.

Entretanto, desenvolver modelos de segmentação semântica para aplicação no contexto brasileiro apresenta desafios específicos. As vias urbanas e rodovias no Brasil possuem particularidades visuais e estruturais que diferem de outros países, o que dificulta a generalização de modelos treinados em cenários internacionais, como o conjunto **CityScapes**. Além disso, a coleta de dados rotulados é cara e trabalhosa, tornando necessário o uso de estratégias alternativas, como **aprendizado semi-supervisionado e adaptação de domínio não supervisionada**. Estas abordagens exploram dados não rotulados para aprimorar o desempenho dos modelos.

Neste trabalho, buscamos desenvolver um modelo eficiente e leve de segmentação semântica, capaz de operar em **tempo real** e adaptado ao contexto das vias brasileiras. A proposta envolve o uso do modelo **LiteSeg**, que será integrado às arquiteturas **CCT** e **PixMatch**, priorizando eficiência e precisão. Para validar a aplicação, utilizaremos um conjunto de dados específico, o **CityScapesBrazil**, com 21.485 imagens coletadas em cidades como Fortaleza, Curitiba, São Paulo, Rio de Janeiro e Porto Alegre, além de estradas de norte a sul do país.

Os resultados obtidos demonstraram a eficácia da arquitetura proposta, com o modelo adaptado apresentando **95,4% de IoU** (Intersection over Union) na classe “*road*” no conjunto de teste do CityScapes. Além disso, o modelo alcançou **30 FPS (quadros por segundo)** em **imagens de alta resolução (2048x1024 pixels)**, confirmando sua capacidade de operar em tempo real. Inspeções visuais realizadas com o conjunto **CityScapesBrazil** também indicaram uma excelente adaptação às particularidades visuais do contexto brasileiro.

Dessa forma, este trabalho busca contribuir para o avanço da segmentação semântica aplicada à direção autônoma no Brasil, garantindo não apenas um alto desempenho, mas também uma execução eficiente em dispositivos com recursos limitados. Ao final, espera-se entregar uma solução viável para a navegação autônoma segura e prática nas vias brasileiras, alinhada às necessidades de um sistema crítico e ao estado da arte em aprendizado profundo.

Capítulo 2

OBJETIVOS

A segmentação semântica é uma tarefa essencial para a navegação autônoma, pois permite ao sistema identificar e classificar, de forma precisa, áreas navegáveis e não navegáveis. Dado o caráter crítico desse tipo de aplicação, é fundamental que o sistema tome **decisões rápidas** e corretas para garantir a segurança do condutor, dos passageiros e dos demais usuários da via. Além disso, essa segmentação precisa ser realizada em tempo real, o que impõe a necessidade de modelos **leves e eficientes**. O objetivo principal deste trabalho é desenvolver um modelo eficiente para a segmentação semântica de áreas navegáveis e não navegáveis em estradas e rodovias brasileiras, que seja capaz de operar em tempo real e apresentar alto desempenho. Para validar a proposta, utilizaremos um conjunto de dados adaptado para o contexto geográfico brasileiro, avaliando a viabilidade de introduzir um modelo adequado às especificidades desse cenário. Assim, busca-se entregar uma solução que seja ao mesmo tempo eficiente, robusta e adequada ao contexto de sistemas críticos de direção autônoma.

Os objetivos específicos são:

- **Construir uma base não rotulada representativa de cenas urbanas brasileiras.**
- **Treinar um modelo leve utilizando técnicas de semi-supervisão e adaptação de domínio não supervisionada**, garantindo a capacidade de operar em múltiplos domínios e generalizar sobre diferentes tipos de cenários.
- **Avaliar o desempenho do modelo em tempo real**, garantindo uma taxa de execução suficiente para sua aplicação em sistemas críticos de navegação autônoma.
- **Verificar a capacidade do modelo de lidar com dados representativos do território brasileiro**, garantindo sua aplicabilidade em diferentes condições e regiões, de forma robusta e eficiente.

As contribuições desta monografia são:

- Modelo treinado com **estratégia de consistência** baseado em um **modelo leve**, garantindo execução em **tempo real**.
- **Adaptação do modelo para cenários brasileiros**, utilizando dados representativos de ruas, rodovias e estradas do território nacional.

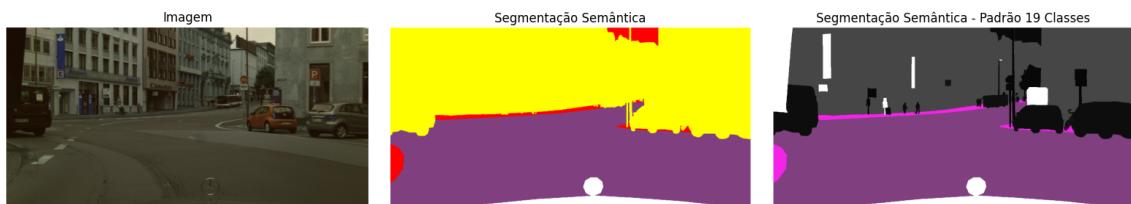
- **Construção de um dataset de baixo custo, abrangendo diversas regiões do Brasil,** para auxiliar no treinamento e validação do modelo. O conjunto é composto por 21.485 imagens, abrangendo cidades como Fortaleza, Porto Alegre, Curitiba, Rio de Janeiro e São Paulo, além de estradas de diversas regiões do Brasil, do Nordeste ao Sul.

As premissas esperadas na avaliação entre as abordagens semi-supervisionada, múltiplos domínio semi-supervisionado e adaptação de domínio não supervisionado são:

- Perturbações em nível de características contribuem para tornar o modelo mais robusto a variações, em comparação a perturbações a nível de entrada, ou estratégias de consistência independentes do nível de perturbação são igualmente robustas a variações.
- Uma estratégia semi-supervisionada é capaz de apresentar um bom desempenho em ambos os subconjuntos, rotulados e não rotulados, em comparação a uma estratégia de adaptação de domínio.
- Uma estratégia híbrida, ou seja, treinamento supervisionado e semi-supervisionado em múltiplos domínios, é capaz de apresentar um desempenho equivalente a uma estratégia semi-supervisionada, garantindo a performance em ambos os domínios e subconjuntos (rotulados e não rotulados).
- Um mesmo conjunto treinado de forma supervisionada no modelo consegue apresentar um desempenho semelhante ao de um treinamento com rótulos ocultos.
- Aspectos regionais entre os datasets CityScapes e CityScapesBrazil são significativos para justificar a adaptação de domínio.

Exemplo de segmentação semântica desejada para este trabalho:

Figura 1 – Exemplo de segmentação semântica CityScapes



Fonte: Elaborada pelo autor.

Capítulo 3

REVISÃO BIBLIOGRÁFICA

Nesta seção são apresentados os principais artigos selecionados ao decorrer da pesquisa bibliográfica. A princípio iniciou-se a pesquisa sobre o potencial uso de modelos baseados em **Transformadores de Visão (ViTs)** ([DOSOVITSKIY, 2020](#)), devido aos modelos de alta generalização publicados pela **Meta/Facebook Research, SAM** ([KIRILLOV *et al.*, 2023](#)) e **DinoV2** ([OQUAB *et al.*, 2023](#)), que oferecem vantagens sobre as **Redes Neurais Convolucionais (CNNs)** como **atenção global**, melhor **escalabilidade, flexibilidade** arquitetural, e tendem a **generalizar** melhor em diferentes tarefas e são eficazes na transferência de conhecimento entre domínios, para esta monografia, é vantajoso para transferência de domínio de cenas urbanas para o contexto brasileiro. Entretanto, os principais modelos ViTs possuem maiores requisitos computacionais e de memória, além de necessitarem de grandes volumes de dados para obter desempenho satisfatório em treinamento zero, o que restringe a aplicação imediata para modelos de segmentação em tempo real e conjunto de dados restrito.

Contudo, a pesquisa afunilou-se para modelos CNNs em tempo real, e uso de estratégias de **aprendizado semi-supervisionado, e adaptação de domínio**, assim viabilizando atender apropriadamente com o objetivo deste trabalho, em proporcionar uma aplicação baseada em aprendizagem profunda para segmentação semântica de **áreas navegáveis e não navegáveis** em **tempo real** para o **cenário brasileiro**.

3.1 Segmentação Semântica: CNNs, ViT, Aprendizado e Adaptação de Domínio

Dois modelos bastante atuais, o **SAM** e o **DINOv2**, ambos da **Meta/Facebook Research**, vem se destacando no cenário atual. Os estudos iniciais deste trabalho abordaram o artigo “*Segment Anything*” ([KIRILLOV *et al.*, 2023](#)) apresentando o projeto **Segment Anything Model (SAM)**, que introduz um novo modelo e conjunto de dados para segmentação de imagens genérica. Ele é um modelo sem necessidade de treinamento adicional, o **desempenho zero-shot** do modelo é impressionante, muitas vezes superando resultados totalmente supervisionados. Já no projeto “*DINOv2: Learning Robust Visual Features without Supervision*” ([OQUAB *et al.*, 2023](#)) a Meta apresenta um método de **aprendizado auto-supervisionado** para criar características visuais robustas sem a necessidade de rótulos ou anotações. Utilizando um grande conjunto de dados curados de **142 milhões de imagens**, o modelo **DINOv2** treina uma arquitetura de **Transformer**

de Visão (ViT) com **1 bilhão de parâmetros** e a destila em modelos menores que superam os melhores recursos visuais disponíveis em benchmarks.

Estes modelos possuem uma performance muito interessante em termos de capacidade de Segmentação Semântica, porém demandam um maior poder de processamento e de recursos. No Apêndice A são detalhados estes modelos, e outros, que são baseados em ViTs, no entanto, neste trabalho, devido aos requisitos de desempenho computacional, vamos focar os estudos e desenvolvimentos em modelos CNNs mais simples e com boas capacidades de aprendizado e adaptação para diferentes domínios (*datasets*).

3.2 Aprendizado semi-supervisionado e adaptação de domínio

No decorrer da pesquisa a necessidade de adaptação da segmentação semântica ao cenário brasileiro ganhou relevância neste trabalho, visto a indisponibilidade de dados rotulados para representar o domínio da aplicação, evidenciou a necessidade de estratégias de treinamento e aprendizagem que façam uso de dados não rotulados, para isso, aprofundou-se as pesquisas para **aprendizagem semi-supervisionado e adaptação de domínio semi-supervisionado e não supervisionado**.

Para o desenvolvimento deste trabalho foram utilizados para nortear as pesquisas os trabalhos “*A Survey on Semi-Supervised Semantic Segmentation*” (PELÁEZ-VEGAS; MESEJO; LUENGO, 2023) e “*Unsupervised Domain Adaptation in Semantic Segmentation: A Review*” (TOLDO *et al.*, 2020), que tratam do uso de dados não rotulados em escalas de domínio distintas, mas ambas exploram o conhecimento extraído de forma não supervisionada para enriquecer os modelos de aprendizagem de máquina, seja sobre o mesmo domínio para aprendizagem semi-supervisionado, ou entre domínio similares para adaptação de domínio não supervisionado.

No artigo “*A Survey on Semi-Supervised Semantic Segmentation*” (PELÁEZ-VEGAS; MESEJO; LUENGO, 2023) é apresentada uma visão abrangente sobre o estado da arte na segmentação semântica semi-supervisionada. Esta abordagem combina imagens rotuladas e não rotuladas para aprimorar a precisão da segmentação, e entre os principais métodos discutidos, destaca-se a **pseudo-rotulagem**, que gera rótulos para dados não rotulados com base em modelos treinados em dados rotulados, outro método importante é a **regularização de consistência**, que assegura que as previsões permaneçam consistentes mesmo com diferentes perturbações nos dados de entrada, características ou saída, além disso, o **aprendizado contrastivo** é utilizado para maximizar a similaridade entre representações de dados semelhantes e minimizar a similaridade entre dados diferentes. Esses métodos usufruem da possibilidade de integrar dados não rotulados para melhorar a performance dos modelos de segmentação semântica.

Já o artigo “*Unsupervised Domain Adaptation in Semantic Segmentation: A Review*”

(TOLDO *et al.*, 2020) revisa as abordagens recentes para adaptação de domínio não supervisionada na segmentação semântica. Neste contexto, modelos treinados em um domínio (fonte) são adaptados para funcionar eficientemente em outro domínio (alvo) sem a necessidade de rótulos no domínio alvo. As estratégias principais incluem o **aprendizado adversário**, que utiliza redes adversariais para alinhar as distribuições de dados dos domínios fonte e alvo, já métodos baseados em **geração** também são destacados, onde imagens no domínio alvo são geradas a partir de imagens no domínio fonte, por fim, também é apresentado a técnica de **auto-aprendizagem**, onde as previsões do próprio modelo servem como rótulos para os dados não rotulados no domínio alvo. Essas estratégias são cruciais para superar a limitação de dados rotulados e reduzir custos, ao mesmo tempo em que melhoram a generalização dos modelos de segmentação semântica.

Capítulo 4

TRABALHOS RELACIONADOS

Esta monografia é fundada sobre três trabalhos que juntos proporcionam uma solução viável para segmentação semântica de áreas navegáveis e não navegáveis com uso de dados não rótulos em tempo real.

4.1 Execução em tempo real

Em relação a execução em tempo real, emprega-se o artigo “*LiteSeg: A Novel Lightweight ConvNet for Semantic Segmentation*” (EMARA; MUNIM; ABBAS, 2019), que apresenta o modelo **LiteSeg** (retratado nesta monografia em detalhes na seção 5.6), um modelo leve para segmentação semântica de imagens. Utilizando uma versão aprimorada do módulo **Atrous Spatial Pyramid Pooling (ASPP)** e convoluções separáveis em profundidade, assim tornando LiteSeg eficiente e rápido.

Outro modelos também buscam aprimorar a performance sem escalar desempenho, são as redes apresentadas nos artigos “*Efficient ConvNet for Real-time Semantic Segmentation*” (ROMERA *et al.*, 2017) e “*ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation*” (MEHTA *et al.*, 2018), que apresentam arquiteturas para segmentação semântica em **tempo real**, focadas na **eficiência computacional**. O primeiro artigo propõe uma rede convolucionar que utiliza **conexões residuais** e **convoluções fatoradas**, apresentando precisão competitiva no conjunto de dados Cityscapes. Já o **ESPNet** introduz o **módulo de pirâmide espacial eficiente (ESP)**, que é 22 vezes mais rápido e 180 vezes menor que o **PSPNet** (ZHAO *et al.*, 2017a) (outra rede leve que utiliza um **módulo de pirâmide de pooling (Pyramid Pooling Module - PPM)** para agregar informações de contexto global em diferentes escalas), com uma redução de apenas 8% na precisão. Ambas as abordagens destacam-se pela combinação de alta eficiência e boa precisão, tornando-as adequadas para aplicações em dispositivos com recursos limitados.

Outro trabalho que emprega estruturas leves para viabilizar tempo de execução em tempo real, é o trabalho “*RTSEG: REAL-TIME SEMANTIC SEGMENTATION COMPARATIVE STUDY*” (SIAM *et al.*, 2018) fornece uma arquitetura de **codificador-decodificador** desacoplado que permite conectar qualquer arquitetura de codificador (ou seja, **VGG16** (SIMONYAN; ZISSERMAN, 2014), **MobileNet** (SANDLER *et al.*, 2018), **ShuffleNet** (ZHANG *et al.*, 2018), **ResNet18** (HE *et al.*, 2016a)) ou decodificador (ou seja, **UNet** (RONNEBERGER; FISCHER;

BROX, 2015), Dilation (YU, 2015), SkipNet (LONG; SHELHAMER; DARRELL, 2015a)) independentemente. Eles descobriram que usar a arquitetura SkipNet junto com MobileNet e ShuffleNet forneceu a melhor relação entre precisão e desempenho.

4.2 Aprendizagem semi-supervisionado e adaptação de domínio não supervisionado

O artigo “*Semi-Supervised Semantic Segmentation with Cross-Consistency Training*” (QUALI; HUDELOT; TAMI, 2020), propõe um método de segmentação semântica semi-supervisionada que utiliza **treinamento de consistência cruzada**, discutido em detalhes na seção 5.10. Esse método reforça a invariância das previsões aplicando diferentes perturbações à saída do codificador. Um codificador compartilhado é treinado com exemplos rotulados, enquanto a consistência é mantida entre as previsões do decodificador principal e dos decodificadores auxiliares. Essa abordagem melhora as representações do codificador e alcança resultados de ponta em vários conjuntos de dados.

O artigo “*PixMatch: Unsupervised Domain Adaptation via Pixelwise Consistency Training*” (MELAS-KYRIAZI; MANRAI, 2021), apresenta uma nova abordagem para adaptação de domínio não supervisionada em segmentação semântica, discutido em detalhes na seção 5.9. A técnica baseia-se no **treinamento de consistência** de pixel, onde a saída do modelo deve ser consistente em relação a pequenas **perturbações** nas imagens do domínio alvo. Um novo termo de perda é introduzido para garantir essa consistência pixel a pixel entre as previsões do modelo em uma imagem alvo e uma versão perturbada da mesma imagem. Comparado a métodos de adaptação adversária, o PixMatch é mais **simples, fácil** de implementar e mais **eficiente** em termos de memória. Experimentos demonstram que essa abordagem alcança resultados fortes em benchmarks desafiadores de adaptação de domínio sintético-para-real, como GTA5-para-Cityscapes e SYNTHIA-para-Cityscapes.

Capítulo 5

FUNDAMENTAÇÃO TEÓRICA

Nesta capítulo é apresentado os estudos para realizar a fundamentação teórica para este trabalho, no qual é tradado a definição e evolução da direção autônoma (seção 5.1), na sequência é introduzido aprendizado profundo (seção 5.2), transformadores de visão (seção 5.3), redes neurais convolucionais (seção 5.4) e *Deeper Atrous Spatial Pyramid Pooling* (DASPP) (seção 5.5), modelo LiteSeg (seção 5.6), treinamento semi-supervisionado e adapatação de domínio (seção 5.7), treinamento por consistência (seção 5.8), modelo PixMatch (seção 5.9), por fim modelo CCT (seção 5.10).

Por fim, no Apêndice B é apresentada uma versão estendida dos tópicos discutidos neste capítulo, "Segmentação Semântica para Direção Autônoma" seção B.1 e "Redes Neurais Convolucionais" seção B.2.

5.1 Segmentação Semântica para Direção Autônoma

A segmentação semântica é uma tarefa essencial em visão computacional, realizada por **Redes Neurais Convolucionais** (i.e., em inglês CNNs), que atribui um rótulo a cada pixel de uma imagem. Essa técnica tem aplicações em áreas como medicina, robótica e direção autônoma. Na condução autônoma, a segmentação é fundamental para interpretar cenários urbanos e identificar **áreas navegáveis, obstáculos e limites**, aspectos essenciais para o funcionamento seguro em diferentes níveis de automação, desde assistência básica (Nível 1) até automação completa (Nível 5), conforme as diretrizes da SAE exposto em (JANAI *et al.*, 2020).

A condução autônoma pode transformar a mobilidade urbana ao **reduzir acidentes**, já que a maioria é causada por erro humano. Veículos autônomos também oferecem benefícios como otimização de rotas, **redução de congestionamentos** e **menor consumo de combustível**, além de ampliar a acessibilidade para pessoas com mobilidade reduzida. No entanto, há desafios significativos para alcançar a robustez necessária em ambientes urbanos complexos, considerando variáveis como clima e condições das vias. Para garantir **segurança e eficiência em tempo real**, os modelos precisam de **alta precisão** e **baixa latência**, o que exige soluções de aprendizado de máquina **robustas e escaláveis**.

Modelos baseados em **Fully Convolutional Networks** (FCNs), como **UNet** e **SegNet**, além de outros mais recentes como **Deeplabv3+** (CHEN *et al.*, 2018) representam marcos na evolução das arquiteturas de segmentação semântica. Recentemente, **transformadores de visão**

têm sido integrados, melhorando a captura de relações contextuais em imagens. No Brasil, a falta de dados rotulados de cenas urbanas representa um obstáculo, tornando fundamentais estratégias como aprendizado semi e não supervisionado. Conjuntos de dados locais, como CaRINA ([SHINZATO *et al.*, 2016](#)) e RTK ([RATEKE; WANGENHEIM, 2021](#)), ajudam a adaptar as soluções para a realidade nacional, mas ainda há a necessidade de desenvolver modelos eficientes que funcionem em tempo real e com **hardware de baixo custo**, tornando a direção autônoma mais acessível e aplicável ao contexto brasileiro.

5.2 Aprendizagem Profunda

O aprendizado profundo (i.e., em inglês *deep learning*), é uma subárea do aprendizado de máquina que utiliza redes neurais artificiais com múltiplas camadas para modelar e resolver problemas complexos. Este campo tem ganhado destaque significativo devido à sua capacidade de lidar com grandes volumes de dados e extrair características, abstraindo tais características a vários domínios de aplicação.

Um dos maiores desafios da atualidade é lidar com dados massivos e a partir destes obter padrões e conhecimento. Quando aplicado a problemas do mundo real modelos robustos a variações ganham relevância para usufruir do volume massivo de dados, como exemplo, lidar com variação de luminosidade, para segmentação semântica, corresponde a pixels tendendo a cor preta ou branca tornando difícil distinguir formas e texturas descaracterizadas pela distorção, tais fatores provocam ruídos e elevam o nível de abstração, dificultando a extração de características.

5.3 Transformadores de visão

Os **Transformadores de Visão** (i.e., em inglês *Vision Transformers*, ViT) adaptam a arquitetura de transformadores, originalmente desenvolvida para processamento de linguagem natural, para tarefas de visão computacional. Introduzidos em 2020 pelo trabalho da equipe de pesquisa do **Google** ([DOSOVITSKIY, 2020](#)), os ***Vision Transformers*** dividem a imagem em pequenos patches, que são convertidos em vetores e tratados como tokens de entrada pelo modelo. A arquitetura inclui patches, camadas de transformador com mecanismos de auto atenção e uma camada convolucional de saída para gerar previsões finais.

A principal vantagem do ViT é sua capacidade de capturar relacionamentos de **longo alcance e contextos globais** dentro da imagem, superando as limitações das **redes neurais convolucionais (CNNs)** que têm campos receptivos mais restritos. No entanto, eles frequentemente exigem grandes volumes de dados para treinamento eficaz e podem ser mais intensivos em termos de memória e tempo de computação.

5.4 Rede Neural Convolucional

Uma versão detalhada deste tópico está presente no Apêndice B.2. As **Redes Neurais Convolucionais (CNNs)** são uma classe específica de redes neurais utilizadas principalmente para o processamento de dados estruturados como matrizes, sendo amplamente aplicadas em imagens. Inspiradas na organização dos neurônios humanos, essas redes são capazes de **identificar padrões e extrair características** complexas por meio de múltiplas camadas. No contexto da segmentação semântica, as CNNs permitem classificar cada pixel de uma imagem em diferentes categorias, facilitando a identificação de objetos e regiões específicas.

A operação fundamental das CNNs é a convolução, onde filtros ou *kernels* deslizam pela imagem para destacar elementos como bordas e texturas. O valor de *stride*, ou passo, define o deslocamento do *kernel* e influencia diretamente o tamanho da imagem resultante. Com diferentes filtros, é possível modificar aspectos visuais, como nitidez, suavidade, bordas e afins. A figura 25 do Apêndice B exemplifica como cada filtro realça características distintas, como contornos ou direções específicas das linhas.

Outro componente essencial das CNNs é a operação de *pooling*, que reduz a dimensionalidade das imagens para simplificar o processamento e evitar o sobreajuste do modelo. As formas mais comuns de *pooling* são o *maxpooling*, que seleciona o maior valor de um grupo de pixels, e o *average pooling*, que calcula a média desses valores. Essas operações diminuem a complexidade computacional e preservam as informações mais relevantes da imagem original.

Para completar a segmentação semântica, arquiteturas como a **UNet** utilizam conexões de salto para combinar diferentes níveis de resolução entre codificador e decodificador, preservando detalhes importantes. Além disso, a convolução transposta é aplicada para restaurar a resolução original da imagem segmentada. Ao final, uma camada **Softmax** converte as ativações em probabilidades, permitindo classificar cada pixel da imagem de forma precisa.

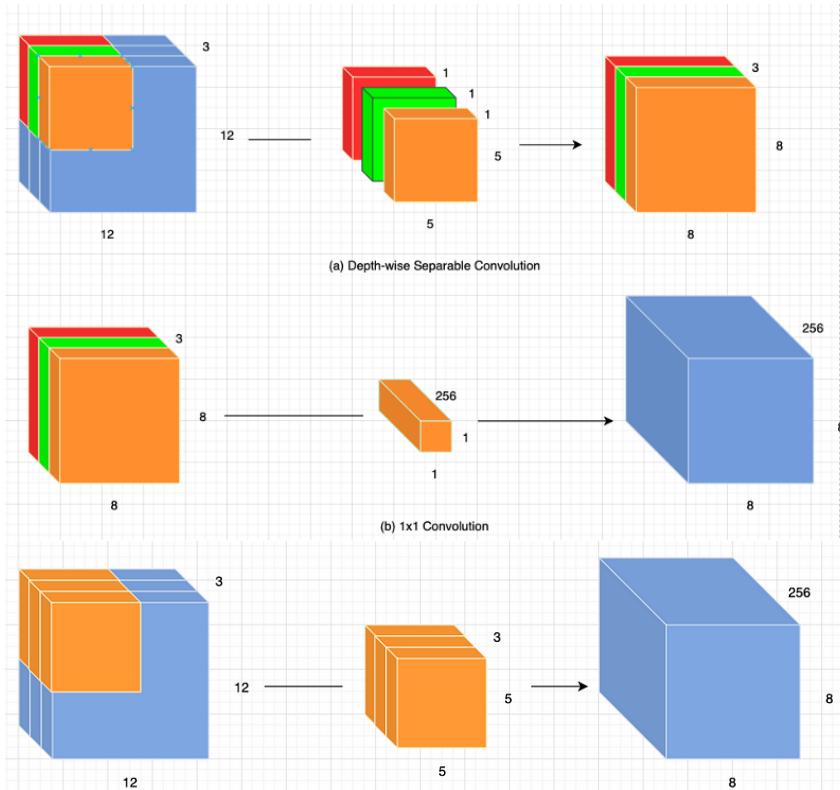
5.5 Deeper Atrous Spatial Pyramid Pooling (DASPP)

O **Deeper Atrous Spatial Pyramid Pooling (DASPP)** é uma evolução do **Atrous Spatial Pyramid Pooling (ASPP)** que aprimora a capacidade das redes neurais convolucionais na extração de características em múltiplas escalas. Utilizando **convoluções dilatadas**, essa técnica amplia o campo receptivo sem reduzir a resolução dos mapas de características ou aumentar os parâmetros treináveis, permitindo que a rede aprenda tanto detalhes finos quanto informações contextuais amplas. O **DASPP** introduz camadas adicionais de convolução sequencial para melhorar a integração de informações e minimizar artefatos na segmentação, resultando em previsões mais precisas. A arquitetura também incorpora *pooling global* e uma camada final de convolução para refinar as características extraídas, otimizando a complexidade visual.

5.6 Modelo LiteSeg

O modelo **LiteSeg**, conforme publicado no artigo "*LiteSeg: A Novel Lightweight ConvNet for Semantic Segmentation*"(EMARA; MUNIM; ABBAS, 2019), é uma arquitetura leve projetada para segmentação semântica de imagens. O **LiteSeg** adota uma abordagem de codificador-decodificador e incorpora algumas técnicas para melhorar a eficiência e a velocidade do modelo. Principais características da arquitetura do modelo **LiteSeg**, é o emprego do **Deeper Atrous Spatial Pyramid Pooling** (**DASPP**, já descrita na seção 5.5) uma versão mais profunda do ASPP, que permite a extração de características em múltiplas escalas, melhorando a capacidade do modelo de capturar informações contextuais de diferentes tamanhos. Também emprega **Conexões Residuais Curtas e Longas**, introduzido no trabalho (HE *et al.*, 2016a), define-se uma estrutura de aprendizagem residual para permitir o treinamento de redes muito profundas, **Residual Networks (ResNets)**, que permite reter mapas de características relevantes na rede, mesmo sobre avanço na profundidade. Por fim, é aplicado também **Convoluçãoções Separáveis em Profundidade** ao invés de convoluções tradicionais (diferença ilustrada na figura 2), que reduz significativamente o número de parâmetros e operações computacionais, mantendo a precisão do modelo como demonstrado no artigo (NEKRASOV; SHEN; REID, 2018).

Figura 2 – Convolução separável em profundidade em comparação com convolução padrão.



Fonte: [Sankar \(2024\)](#).

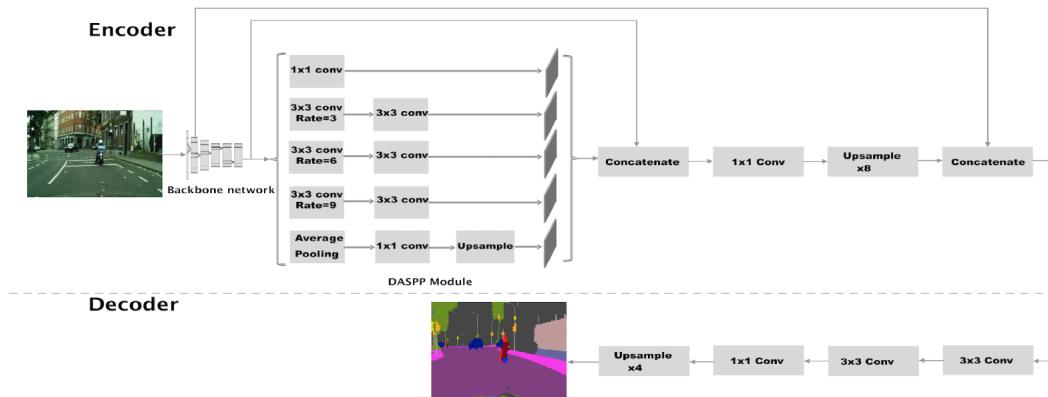
Na figura 2, temos a convolução separável em profundidade 5x5 (a), seguida por convolução padrão 1x1 (b). A última sequência ilustra uma convolução padrão 5x5 equivalente, em

que a primeira requer ordem de grandeza de **54000** multiplicações, enquanto a segunda requer **1200000** multiplicações.

Todas as características implementadas no modelo contribuem para o excelente desempenho alcançado no benchmark CityScapes com emprego do **MobileNetV2** como rede de backbone, atinge uma precisão de **67,81% de intersecção média sobre união a 161 quadros por segundo com resolução de 640 pixels de altura e 360 de largura**.

A arquitetura do modelo **LiteSeg** é do tipo **codificador-decodificador**, composto por módulo codificador com seus componentes de rede de *backbone* e módulo **DASPP**, e o módulo decodificador baseado no **DeepLabV3** (CHEN, 2017). O módulo codificador pega uma imagem de entrada e gera um vetor de característica de alta dimensão. O módulo decodificador restaura as informações espaciais desse vetor de recurso, sua arquitetura é apresentada na figura 3, e seus componentes serão detalhados nas subseções **5.6.1 Codificador**, **5.6.2 Backbone**, **5.6.3 Módulo DASPP** e **5.6.4 Decodificador**.

Figura 3 – Arquitetura do modelo LiteSeg.



Fonte: Emara, Munim e Abbas (2019).

5.6.1 Codificador

O codificador proposto no artigo (EMARA; MUNIM; ABBAS, 2019) contém uma arquitetura de rede de *backbone* que atua como uma arquitetura de classificação de imagem para extração de recursos. No estudo realizado foram avaliadas 3 arquiteturas escolhidas para atender com os requisitos de eficiência desejado, sendo os modelos leves: **MobileNet**, **ShuffleNet** e **Darknet19**. Não apenas o tipo de rede de *backbone* controla o desempenho, mas também o *stride* saída para o módulo **DASPP**, que é definido como 32 para **MobileNet**, depois de revisado o equilíbrio entre performance e desempenho, como realizado na literatura pelos trabalhos (CHEN, 2017) e (MURUGAPPAN *et al.*, 2023).

Na sequência o codificador emprega o módulo **DASPP**, similar ao visto no modelo **DeepLabv3** (CHEN, 2017) que implementa o módulo **ASPP** com diferentes taxas de dilatação

para capturar informações em várias escalas, seguindo a abordagem apresentada em ParseNet ([LIU, 2015](#)). Aqui, uma nova versão mais profunda do módulo **ASPP** é proposta (chamada **Deeper Atrous Spatial Pyramid Pooling (DASPP)**), adicionando convolução padrão 3×3 após convoluções dilatada 3×3 para refinar os recursos e fundindo a entrada e a saída do módulo DASPP por meio de conexão residual curta, que colabora na preservação de características relevantes de baixo nível pela fusão e a reutilização de recursos (que incluem manchas ou bordas coloridas) de camadas inferiores, o uso de conexões entre recursos de baixo nível e alto nível de camadas superiores provaram ser úteis para segmentação de alta resolução ([LIU, 2015](#)) . Além disso, o número de filtros de convolução do ASPP é reduzido pelas convoluções 3×3 adicionais no DASPP de 255 para 96, assim colaborando no ganho de desempenho computacional.

Por fim, no codificador é empregado a abordagem de conexão residual longa pela abordagem de concatenação como uma adição elemento a elemento, que requer que a saída residual e a entrada tenham a mesma dimensão de largura, altura e profundidade em vez da concatenação convencional que requer apenas a mesma dimensão de largura e altura. A incompatibilidade em largura e altura pode ser mantida por upsampling e, opcionalmente, uma convolução 1×1 pode ser usada para reduzir a profundidade dos recursos para eficiência computacional. Foi descoberto que conexões de salto longo ajudam a tornar limites semânticos mais claros e conexões de salto curto com DASPP ajudam no ajuste fino dos segmentos semânticos e, portanto, fornecem informações geométricas mais ricas.

5.6.2 Backbone

MobileNetV2 é uma arquitetura de rede neural convolucional desenvolvida pelo **Google** ([SANDLER et al., 2018](#)), projetada para otimizar a eficiência e a precisão em aplicações de visão computacional em dispositivos móveis e embarcados. Esta arquitetura é uma evolução da **MobileNetV1** e introduz várias melhorias, como **blocos residuais invertidos** e gargalos lineares, que permitem um fluxo de informações mais eficiente e **reduzem a complexidade computacional**. A **MobileNetV2** é especialmente adequada para dispositivos com recursos limitados, onde a eficiência energética e a velocidade de processamento são cruciais, ideal para ser aplicado como backbone de uma arquitetura leve.

A saída da **MobileNetV2** pode ser utilizada de várias maneiras, dependendo da aplicação. Em tarefas de classificação de imagens, a rede pode fornecer probabilidades associadas a diferentes classes de objetos presentes na imagem. Além disso, a **MobileNetV2** pode ser utilizada como backbone em modelos mais complexos, como aplicado neste trabalho para segmentação semântica, nesta tarefa extrai características ricas e detalhadas das imagens, que podem ser utilizadas por outras camadas da rede para realizar tarefas específicas.

As principais características da **MobileNetV2** incluem o uso de **blocos residuais invertidos**, que conectam camadas de diferentes profundidades, e **convoluções separáveis em profundidade**, que dividem a convolução em duas operações distintas para reduzir o número

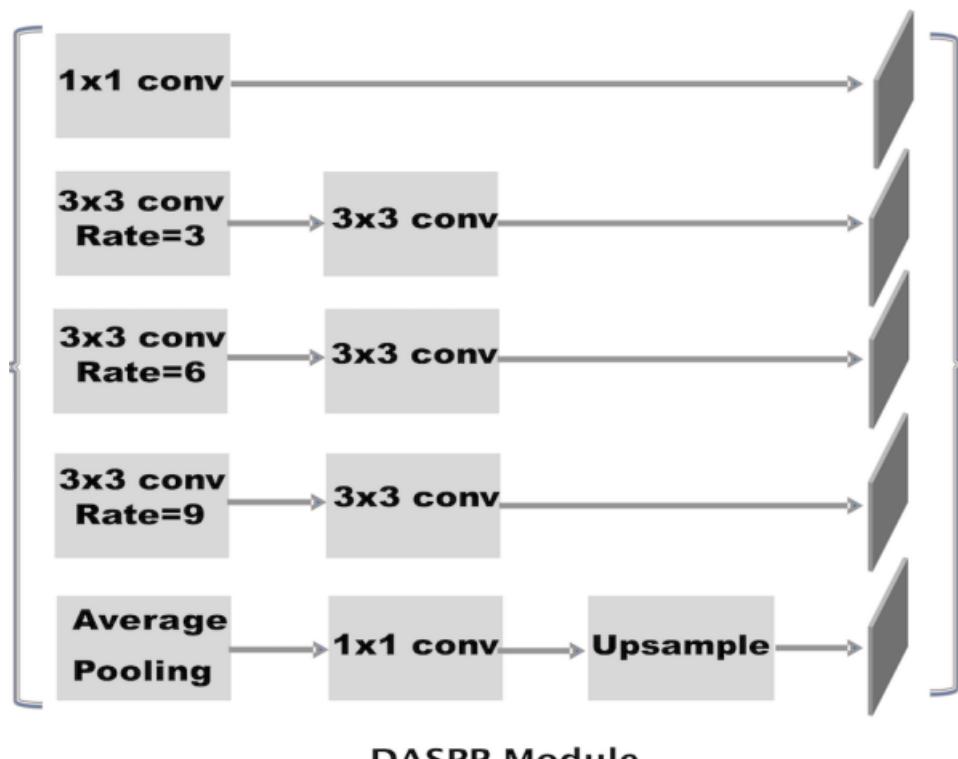
de parâmetros e cálculos necessários, como já mencionado é ilustrado na figura 2. A função de ativação ReLU6 é empregada para limitar a saída da ReLU a 6, prevenindo instabilidades numéricas em cálculos de baixa precisão.

5.6.3 DASPP

Como já apresentado na seção 5.5, o **DASPP** expande essa abordagem **ASPP** ao adicionar mais camadas de **convolução dilatada** e melhorar a integração de informações em múltiplas escalas, por consequência permite que a rede obtenha um campo receptivo maior sem aumentar o tamanho do *kernel*.

No modelo específico do **LiteSeg** o **DASPP** é implementado com 5 ramificações, a primeira é composta por uma convolução 1 x 1, a segunda, terceira e quarta ramificações é composta por uma **convolução dilatada** com taxa de dilatação, respectivamente, 3, 6 e 9, seguido de uma convolução 3 x 3, com a última ramificação é composta por um *global average pooling* (uma operação de pooling projetada para substituir camadas totalmente conectadas em CNNs clássicas) seguido de uma convolução 1x1 e *upsample*, resultando em 5 níveis de escala com o número de filtros de convolução do **ASPP** reduzido de 255 para 96, como já apresentado, possibilita ganho de complexidade computacional, devido ao acréscimo das convoluções 3 x 3, em cada nível do **ASPP**, definindo assim o **DASPP**, como ilustrado na arquitetura do modelo na figura 4.

Figura 4 – Módulo DASPP.



Fonte: [Emara, Munim e Abbas \(2019\)](#).

5.6.4 Decodificador

O decodificador do modelo **LiteSeg** é baseado no modelo da **Meta Deeplabv3+** ([CHEN et al., 2018](#)) que apresenta um decodificador simplificado que é composto de camadas de convolução e *upsampling* 3×3 padrão. A modificação realizada pelos autores do **LiteSeg** e a adição de outra camada de convolução 3×3 e redução do número de filtros em todas as convoluções 3×3 de 256 para 96, assim obtendo ganho de desempenho computacional. Além disso, a saída do codificador é aumentada com recursos de baixo nível de camadas anteriores da rede de backbone por meio de conexão residual longa no codificador.

Esses recursos de baixo nível podem ter muitos mapas de recursos e, para resolver esse problema, uma convolução 1×1 é utilizada para reduzir o número de canais de recursos de baixo nível. Caso contrário, com algumas redes backbone leves, não haverá necessidade de aplicar a convolução 1×1 em recursos de baixo nível devido ao baixo número de canais (por exemplo, 24 no caso de usar MobileNet).

5.7 Treinamento Semi-Supervisionado e não supervisionado

O aprendizado profundo semi-supervisionado é uma abordagem que utiliza tanto dados rotulados quanto não rotulados para treinar modelos de aprendizado de máquina. Em muitos cenários práticos, rotular grandes volumes de dados pode ser caro e demorado, por exemplo, o processo de rotulação do **CityScape** ([CORDTS et al., 2016](#)) demandou aproximadamente 90 minutos por imagem, enquanto dados não rotulados estão frequentemente disponíveis em maior quantidade. O objetivo do aprendizado semi-supervisionado é aproveitar o conhecimento dos dados não rotulados para melhorar o desempenho do modelo, **reduzindo a necessidade de grandes conjuntos de dados rotulados**.

O aprendizado profundo não supervisionado é uma abordagem que busca identificar padrões e estruturas em dados que não têm rótulos ou categorias predefinidas. O objetivo é aprender representações úteis dos dados sem supervisão explícita, que podem ser usadas para várias tarefas, como **redução de dimensionalidade**, **geração de dados** e **clusterização**. Os modelos não supervisionados são essenciais quando os dados rotulados são escassos ou inexistem.

Métodos comuns no aprendizado profundo não supervisionado incluem **autoencoders**, que aprendem a compactar e reconstruir os dados, e **redes adversariais gerativas** (GANs), que geram novos exemplos de dados com base em uma distribuição aprendida. Outra abordagem importante é a **clusterização**, onde técnicas como ***k-means*** e **algoritmos baseados em densidade** são usadas para agrupar dados semelhantes e descobrir estruturas subjacentes.

Para segmentação semântica com redes neurais, existem alguns métodos de treinamento

semi-supervisionado e não supervisionado, como os organizados nos trabalhos “*A Survey on Semi-Supervised Semantic Segmentation*” (PELÁEZ-VEGAS; MESEJO; LUENGO, 2023) e “*Unsupervised Domain Adaptation in Semantic Segmentation: A Review*” (TOLDO *et al.*, 2020), apresenta uma revisão da literatura acerca de alguns métodos e busca pelo uso de dados não rotulados, tanto no aprendizado semi-supervisionado quanto na adaptação de domínio não supervisada. Esse conteúdo será utilizado para os tópicos seguintes, e está resumido na figura 22 no Apêndice B.

O primeiro método é ***Adversarial Learning*** e métodos generativos, como **Redes Adversárias Generativas** (i.e., ***Generative Adversarial Network GANs***) e **Auto Codificadores Variantes** (i.e., ***Variational Autoencoder VAEs***), são técnicas que geram dados a partir de distribuições aprendidas, no ***Adversarial Learning*** é estabelecida uma competição entre duas redes neurais, onde uma gera dados falsos e a outra tenta distingui-los dos dados reais, melhorando a capacidade de geração de dados realistas, suas aplicações são mais populares em tarefas de geração de imagens, síntese de voz e criação de texto, mas também pode ser utilizado para treinamento semi supervisionado e não supervisionado na segmentação semântica.

A análise das **Discrepâncias Entre Classificadores** e a **Auto-Aprendizagem** são métodos que visam melhorar a precisão e a robustez dos modelos. A análise das discrepâncias compara as previsões de múltiplos modelos para identificar e corrigir erros, enquanto a auto-aprendizagem utiliza um modelo treinado para rotular novos dados não rotulados, permitindo um aprendizado contínuo e iterativo. A minimização da entropia é outra técnica que aumenta a confiança das previsões, incentivando o modelo a fazer previsões mais precisas.

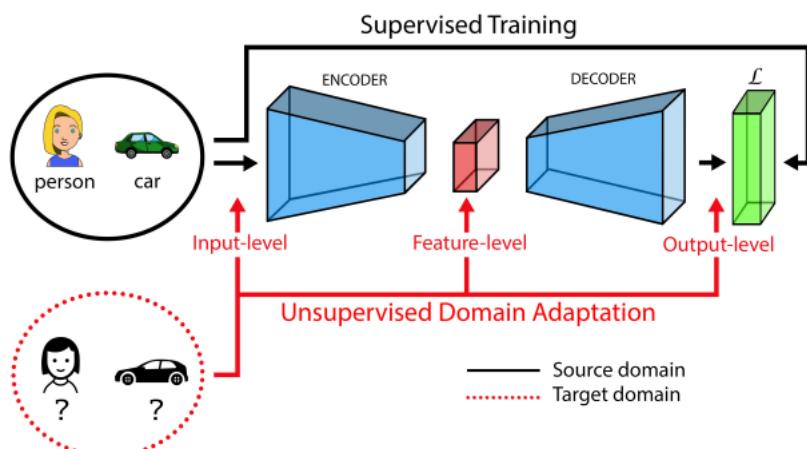
No aprendizado semi-supervisionado e não supervisionado também apresenta o método baseado na **regularização de consistência**, que consiste na suposição de suavidade, essa suposição diz que, para dois pontos próximos no espaço de entrada, seus rótulos devem ser os mesmos. Ou seja, um modelo robusto deve obter previsões semelhantes para um ponto e uma versão localmente modificada dele.

No aprendizado não supervisionado é proposto também o uso do ***Curriculum Learning*** e ***Multi-Task Learning*** que são abordagens que melhoram a eficiência e a capacidade de generalização dos modelos. ***Curriculum Learning*** inspira-se no aprendizado humano, começando com tarefas simples e avançando para tarefas mais complexas, enquanto ***Multi-Task Learning*** treina um modelo em múltiplas tarefas simultaneamente, permitindo o compartilhamento de representações entre tarefas relacionadas. Essas técnicas ajudam os modelos a aprender de forma mais eficaz e robusta, especialmente em cenários com interdependências entre tarefas e amostras com complexidade variante.

5.8 Estratégia de treinamento por consistência

O aprendizado profundo por consistência é uma abordagem no campo do aprendizado semi-supervisionado que visa melhorar o desempenho dos modelos usando tanto dados rotulados quanto não rotulados. Essa técnica se baseia na ideia de que, para dados não rotulados, as previsões do modelo devem ser consistentes mesmo quando esses dados são submetidos a pequenas **perturbações ou transformações**. Essa consistência ajuda o modelo a aprender representações mais **robustas e generalizáveis, reduzindo a necessidade de grandes quantidades de dados rotulados**. Nesse sentido, métodos baseados em regularização de consistência aproveitam dados não rotulados aplicando perturbações, em nível de **entrada, características ou saída do modelo**, ou até mesmo uma combinação de perturbações em múltiplos níveis, como apresentado na figura 5, além da versão perturbada é treinado um modelo que não é afetado por essas perturbações que será comparado com sua versão perturbada, isso proporciona um termo de regularização não supervisionado, que pode ser adicionado à função de perda para medir a distância entre as previsões originais e perturbadas junto a perda supervisionada.

Figura 5 – Representação gráfica do processo de adaptação de domínio não supervisionado.



Fonte: Peláez-Vegas, Mesejo e Luengo (2023).

Na figura 5, uma perda de tarefa L (por exemplo, uma perda de entropia cruzada) é usada para um estágio de treinamento supervisionado no domínio de origem usando as anotações semânticas. A adaptação não supervisionada aos dados de destino sem rótulos pode ser realizada em níveis diferentes (por exemplo, entrada, recursos ou saída) com estratégias diferentes.

A motivação por trás do aprendizado por consistência é que, se um modelo é robusto o suficiente para manter suas previsões consistentes frente a variações nos dados, ele está aprendendo características significativas e úteis. Essa abordagem é particularmente útil em cenários onde é caro ou demorado rotular grandes volumes de dados, permitindo que modelos aproveitem dados não rotulados para melhorar seu desempenho.

Quando observado para treinamento em aprendizagem semi-supervisionado e adaptação

de domínio não supervisionado, verificamos que aplicação de uma estratégia de consistência é similar. Enquanto semi-supervisionado busca aprimorar o modelo sobre o mesmo domínio de dados composto por dados rotulados e não rotulados, também compartilha conhecimento **intra-domínio**, do mesmo modo o **Unsupervised Adaptation Domain (UDA)** busca compartilhar conhecimento entre conjuntos, mas em outra escala, no caso conhecimento entre um domínio rotulado e não rotulado.

Além disso, o **aprendizado por consistência** contribui para a criação de modelos que são **menos suscetíveis a sobreajuste** e que podem generalizar melhor para novos dados. Ao迫使 o modelo a ser consistente em suas previsões frente a variações nos dados, é aprimorado a extração de características que são invariantes e relevantes, melhorando seu desempenho em cenários do mundo real onde os dados podem variar. Apesar dos benefícios, o aprendizado profundo por consistência apresenta alguns desafios. A escolha adequada das perturbações e transformações é crítica, pois perturbações inadequadas podem introduzir ruído excessivo ou desviar o treinamento. Além disso, o ajuste de hiperparâmetros, como a força da regularização, pode ser complexo e exigir experimentação cuidadosa.

5.9 PixMatch: Unsupervised Domain Adaptation via Pixelwise Consistency Training

O **PixMatch** apresentado no trabalho ([MELAS-KYRIAZI; MANRAI, 2021](#)), trata-se de uma arquitetura estruturada para adaptação de domínio não supervisionada com base na estratégia de treinamento de consistência de domínio-alvo. A arquitetura é baseada no viés que para ter um bom desempenho no domínio-alvo, a saída de um modelo deve ser consistente com relação a pequenas **perturbações** de entradas. Especificamente, é um novo termo de perda para impor consistência pixel a pixel entre as previsões do modelo em uma imagem-alvo e uma versão perturbada da mesma imagem. Em comparação com métodos populares de adaptação adversária, o estudo afirma que a abordagem desenvolvida é mais simples, fácil de implementar e mais eficiente em termos de memória durante o treinamento.

O trabalho do **PixMatch** realiza experimentos e estudos extensivos para demonstrar que a abordagem simples atinge resultados notavelmente fortes em dois benchmarks de sintético para real, **GTA5-para-Cityscapes** e **SYNTHIA-para-Cityscapes**, devido a premissa que a segmentação semântica pode fazer valer de treinamento de modelos em imagens anotadas de um domínio simulado (fonte) e implantá-los em domínios reais (alvo).

5.9.1 Arquitetura e Treinamento

Como já mencionado anteriormente o modelo escolhido para o **PixMatch** é o **DeeplabV2** ([CHEN et al., 2017](#)) com backbone **ResNet-101** ([HE et al., 2016b](#)) sem modificações,

onde suas principais contribuições do trabalho é apresentar uma nova arquitetura baseada em **treinamento por consistência** para adaptação de domínio não supervisionado, e exploração de funções de perturbações para estimular a consistência. Para esta monografia, o foco é fazer uso da estrutura **PixMatch** no que diz respeito ao **treinamento por consistência** e empregar as funções de perturbações desenvolvidas no artigo.

A arquitetura do **PixMatch** é desenvolvida para proporcionar o **treinamento de consistência e pseudo rotulagem** para impor consistência no domínio de destino, no qual a função de perda é composta por dois termos de entropia cruzado, como descrito no trabalho ([MELAS-KYRIAZI; MANRAI, 2021](#)). O primeiro termo é a perda cruzada supervisionada padrão para o domínio origem com rótulos, conforme expresso pela fórmula 5.1, no qual $p(i)$ é a distribuição de probabilidade de saída no pixel i para entrada de origem x_s , y é o rótulo semântico verdadeiro e n_s é o número de imagens no conjunto de dados S. O segundo termo é a perda de consistência no domínio alvo, no qual primeiro é predito uma amostra x_t pelo modelo para obter $y_t = \text{argmax}(q_t) = \text{argmax}(p_t(y|x_t))$, em seguida o par (x_t, y_t) é perturbado para produzir o par (x_{tpert}, y_{tpert}) que será utilizado para nova predição de x_{tpert} no modelo resultando y_{t2} , por fim é tomado a entropia cruzada como explicita na fórmula 5.2 com o par y_{tpert} e y_{t2} , logo estabelecendo a consistência entre da predição perturbada e não perturbada, que devem ser próximas. A arquitetura e estratégia de perda da arquitetura estão expostas na figura 6.

$$L_S = -\frac{1}{n_s} \sum_{s \in S} \sum_{i=1}^{H \cdot W} H(y_S^{(i)}, p^{(i)}(y|x_s)) \quad (5.1)$$

$$L_T = -\frac{1}{n_T} \sum_{t \in T} \sum_{i=1}^{H \cdot W} H(y_{tpert}^{(i)}, p^{(i)}(y_{t2}|x_{tpert})) \quad (5.2)$$

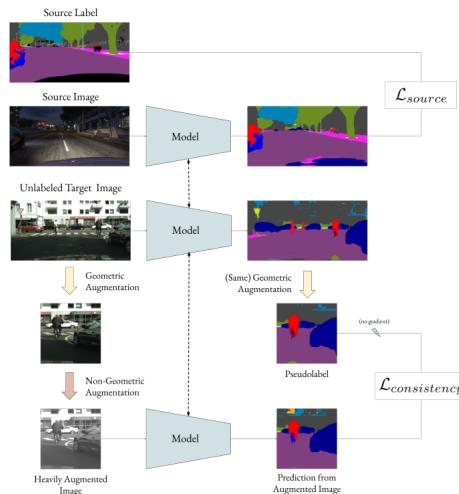
Com tudo, a perda de entropia cruzada total (L) deve ser a combinação linear da perda supervisionada no domínio origem (L_s) e não supervisionada por consistência no domínio alvo (L_t), $L = L_s + p_t \cdot L_t$, onde p_t é um hiperparâmetro da perda não supervisionada que controla a ponderação relativa entre a perda de origem e destino. A representação das perdas calculadas no treinamento pode ser verificada na figura 6.

5.9.2 Função de Perturbação

No decorrer do estudo comparativo entre diferentes combinações de funções de perturbação uma teve melhor resultado nos benchmarks **CityScape** e **GTA5**, **Data Augmentation Consistency**, ou aumento de dados para consistência, que geralmente é utilizado para aumentar o volume do conjunto de dados e estabelecer distorções para tornar modelos mais robustos, neste contexto é utilizado como função de perturbação, exemplo desta perturbação está exposto na figura 7. O aumento de dados realizados foi a composição das seguintes operações:

- **RandomResizedCrop(escala=(0.2, 1))** – A amostra é recortada em uma parte aleatória, em seguida, redimensionar esse recorte aleatoriamente variando de 20% a 100% do tamanho original da imagem.
- **Compose([RandomBrightnessContrast(p=1), HueSaturationValue(p=1)], p=0.8)** – A amostra tem contraste, brilho e saturação ajustados aleatoriamente com probabilidade de 80%.
- **ToGray(p=0.2)** - A amostra é convertida para escala de cinza com probabilidade de 20%.
- **GaussianBlur(Blurlimit=5, p=0.5)** - Aplica um desfoco gaussiano à amostra com um limite de desfoco de 5 e uma probabilidade de 50%.

Figura 6 – Arquitetura e representação das perdas do PixMatch.

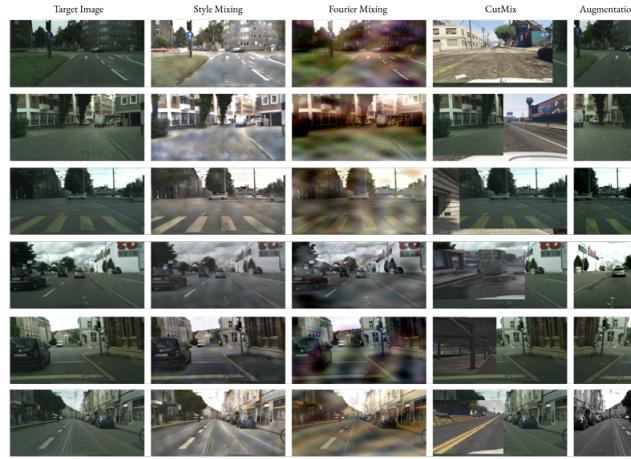


Fonte: [Melas-Kyriazi e Manrai \(2021\)](#).

Verificou-se no trabalho que a arquitetura proposta é receptiva a complementos de outros métodos de adaptação de domínio. Combinou-se mais um termo a função de perda, utilizando a perda não supervisionada baseada na perda quadrática máxima (expresso na formula 5.3), onde p é a matriz de probabilidade das C classes para a dimensão $H \times W$ da imagem predita, como implementado por ([MELAS-KYRIAZI; MANRAI, 2021](#)). Por fim, a melhor configuração de treinamento para o **PixMatch** nos benchmarks **CityScape** e **GTA5**, é a aplicação do aumento de dados como função de perturbação, aplicação do termo de perda quadrática máxima (L_{sm}) a combinação linear da perda com peso 0,1 e 0,05, respectivamente, logo a função perda pode ser expressa como $L = L_s + 0,1L_t + 0,05L_{sm}$.

$$L_{sm}(p|x_t) = -\frac{p^2}{2} \quad (5.3)$$

Figura 7 – Exemplo das perturbações geradas, para este trabalho apenas o Augmentation será aplicado.

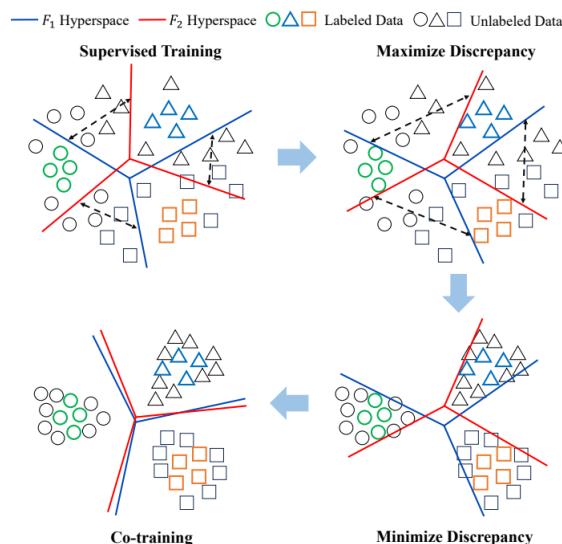


Fonte: [Satya \(2024\)](#).

5.10 Semi-Supervised Semantic Segmentation with Cross-Consistency Training

A arquitetura proposta em “*Semi-Supervised Semantic Segmentation with Cross-Consistency Training*” ([PELÁEZ-VEGAS; MESEJO; LUENGO, 2023](#)), chamado de **CCT**, apresenta uma arquitetura codificador-decodificador com treinamento de consistência cruzada para abordagem de segmentação semântica semi-supervisionado, e treinamento em múltiplos domínio semi-supervisionado. A hipótese do trabalho é que a fronteira de decisão entre as classes (ilustrado na figura 8) está estabelecida em regiões na representação de baixa densidade, ou seja, em nível de características, mais predominante do que na entrada do modelo.

Figura 8 – Ilustração dos hiperespaços para definir a classificação de dados rotulados e não rotulados.



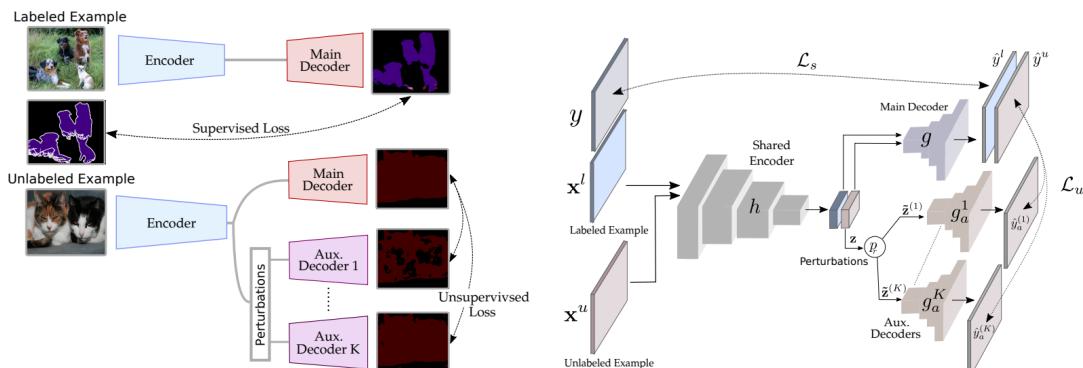
Fonte: [Peláez-Vegas, Mesejo e Luengo \(2023\)](#).

A abordagem da arquitetura baseada em **codificador-decodificador** é composta por um **codificador compartilhado** a um **decodificador principal** treinado de forma supervisionada e um conjunto de decodificadores auxiliares treinados de forma não supervisionada, em que cada **decodificador auxiliar** recebe uma versão diferentemente perturbada da saída do codificador para aproveitar as amostras não rotuladas. Por fim, é imposto a consistência entre o decodificador principal sem perturbações e os decodificadores auxiliares perturbados todos sobre o subconjunto de dados não rotulados, assim estabelecendo a perda não supervisionada do modelo.

5.10.1 Arquitetura e treinamento

Como já apresentado a arquitetura do **CCT**, é composto por codificador compartilhado, aprimorado tanto pelo treinamento supervisionado quanto pelo não supervisionado, um **decodificador principal treinado de forma supervisionada** e **decodificadores auxiliares treinados de forma não supervisionada por consistência**, como exposto na figura 9. As contribuições do trabalho devem-se à proposta de **treinamento de consistência cruzada** para método semi-supervisionado de segmentação semântica, estudo sobre aplicação de funções de perturbação e uso de dados fracamente rotulados.

Figura 9 – Arquitetura e estratégia de treinamento com perda supervisionada e não supervisionada do CCT.



Fonte: [Satya \(2024\)](#).

A perda calculada para o treinamento no modelo é similar ao visto no **PixMatch**, composto por um termo supervisionado (L_s , como expresso pela fórmula 5.4, sobre o domínio D_l supervisionado) e um termo não supervisionado (L_u , como expresso pela fórmula 5.5, sobre o domínio D_u não supervisionado) logo a perda total no treinamento é a combinação linear $L = L_s + uL_u$, em que u é a ponderação do termo não supervisionado para contribuir para a perda total do treinamento.

A perda não supervisionada é obtida através dos dados não rotulados X_u preditos no codificador h , obtendo $z_i = h(X_i^u)$, considerando R funções de perturbação estocásticas, com p_r pertencente a $[1, R]$, temos K versões perturbadas da saída do codificador por g_k decodificadores auxiliares, $z^i = g_k(z_i)$. Por fim, o treinamento tem por objetivo minimizar a perda não supervisionada L_u , no qual é calculado a distância entre a predição do decodificador principal e

os k-ésimos de decodificadores auxiliares, através da medida de distância erro quadrático médio, como expresso na equação 5.6.

$$L_S = \frac{1}{|D_l|} \sum_{X_i^l, y_i \in D_l} H(y_i, f(x_i^l)) \quad (5.4)$$

$$L_u = \frac{1}{|D_u|} \frac{1}{K} \sum_{X_i^u \in D_u} \sum_{k=1}^K d(g(z_i, g_a^k(z_i))) \quad (5.5)$$

$$d_{MSE}(y^u, y_a^k) = \frac{1}{N} \sum_i^N (y^u(i) - y_a^k(i))^2 \quad (5.6)$$

5.10.2 Funções de Perturbação

Nos estudos do CCT foram explorados três tipos de perturbações: baseada em previsão, baseada em característica e perturbações aleatórias. A primeira consiste em adicionar **perturbações baseada na previsão** do codificador sobre os decodificadores auxiliares com transformações de mascaramento e perturbações adversarias; O segundo grupo de perturbações consiste em injetar **ruido** ou **eliminar** algumas ativações de características; por fim, o terceiro é a aplicação de **DropOut** espacial aleatório na saída do decodificador como visto em ([TOMPSON et al., 2015](#)).

5.10.2.1 Perturbações Baseadas em Previsão

Mascaramento Guiado, dada a importância dos relacionamentos de contexto para a compreensão de cenas complexas, a rede pode ser muito dependente desses relacionamentos, logo, para limitá-los, gerou-se duas versões perturbadas da saída do codificador, a primeira máscara os objetos detectados de primeiro plano, e a segunda é o complemento da máscara de objetos, ou seja, o fundo da imagem. Para destacar os objetos do fundo é utilizado a previsão final do decodificador principal.

Recorte guiado, introduzido para reduzir a dependência de partes específicas dos objetos, e inspirado no Recorte ([DELBRACIO; LEZAMA; CARBAJAL, 2019](#)) que mascara aleatoriamente algumas partes da imagem de entrada, no trabalho, primeiro encontra-se a possível extensão espacial (ou seja, caixa delimitadora) de cada objeto detectado usando na saída do decodificador, em seguida, gera-se um corte aleatório dentro da caixa delimitadora (obtida pela previsão do decodificador principal) de cada objeto do mapa de recursos correspondente.

Virtual Adversarial Training Intermediate (I-VAT), empregado para empurrar ainda mais a distribuição de saída para ser isotopicamente suave em torno de cada ponto de dados, investigamos usando VAT ([MIYATO et al., 2018](#)) como uma função de perturbação a ser aplicada a saída do codificador (z) em vez das entradas não rotuladas. Para um determinado decodificador

auxiliar, encontra-se a perturbação adversarial (r_{adv}) que altera sua previsão ao máximo. O ruído é então injetado em z para obter a versão perturbada $z^{adv} = r_{adv} + z$.

5.10.2.2 Perturbações Baseadas em Características

Função ruído, amostrou uniformemente um tensor de ruído $N \simeq U(-0.3, 0.3)$ nas mesmas dimensões da saída do codificador (z), logo o ruído é então injetado na saída z , resultando em $z^r = (z * N) + z$, assim o ruído é injetado proporcionalmente a cada ativação de z .

Função de remoção, novamente amostrou uniformemente o limite $L \simeq U(0.6, 0.9)$, em seguida soma-se a dimensão do canal e normaliza o mapa de características para obter a versão z' da saída do codificador (z), gerando assim uma máscara $M_{drop} = z^{drop} < L$, usada para gerar a versão perturbada de $z^{drop} = z * M_{drop}$, desta forma, é mascarado cerca de 10% a 40% das regiões mais ativas do mapa de características z .

Capítulo 6

METODOLOGIA

Neste capítulo serão apresentados os tópicos de coleta de dados, uso dos trabalhos base, avaliação de resultados, escolha e preparação dos modelos, arquitetura dos modelos e parâmetros do experimento, respectivamente, nas seções [6.1](#), [6.2](#), [6.3](#), [6.4](#), [6.5](#) e [6.6](#)

Ressalta-se que no Apêndice [C](#) estão dispostos mais detalhes sobre a coleta de dados da seção [6.1](#).

6.1 Coleta de dados cenários urbanos brasileiros

Neste trabalho foram empregadas três bases para verificar a precisão e performance da estratégia de treinamento do modelo **LiteSeg** sobre as arquiteturas do **CCT** e **PixMatch**, sendo os conjuntos de imagens: "*The Cityscapes Dataset for Semantic Urban Scene Understanding*"(**CityScapes**) ([CORDTS et al., 2016](#)), "*Playing for data: Ground truth from computer games*"(**GTA5**) ([RICHTER et al., 2016](#)) e o conjunto gerado para esta monografia com cenas urbanas no território brasileiro denominada **CityScapesBrazil**.

CityScapes e **GTA5** são conjuntos estabelecidos para benchmarks no domínio de aplicação para tarefa de segmentação semântica, principalmente, para avaliar adaptação de domínio entre sintético e real, ambas possuem **19 classes** para treino, como listadas no Apêndice [D](#) na tabela [21](#), no qual é discriminado elementos das imagens. CityScape é um conjunto de dados que contém um conjunto diversificado de sequências de vídeo estéreo gravadas em cenas de rua de **50 cidades diferentes da Alemanha** durante o dia, nas estações da primavera, verão, outono, com anotações de nível de pixel de alta qualidade de **5.000 imagens**, além de um conjunto maior de **20.000 imagens** fracamente anotados para um total de **30 classes**. O conjunto extraído do jogo *Grand Theft Auto V*, é um conjunto de dados que consiste em 24966 quadros densamente rotulados, no qual os rótulos de classe são compatíveis com os conjuntos de dados **CamVid** e **CityScapes**.

Para esta monografia, determinou-se o agrupamento das classes em **3 grupos**, **Navegável**, **Inavegável** e **Obstáculos**, no qual tomou-se as classes que representam superfícies, como: estrada, calçada, terreno e por representação em imagens 2D também considerou-se o céu como superfície, e o restante é considerado elementos tridimensionais com possibilidade de se tornar obstáculo. A premissa para escolha desses 3 grupos de classes, é que não importa se o modelo prediz um humano, moto, carro ou poste, pois tudo é considerado objeto tridimensional, e para

tomada de decisão para direção, basta predizer se na trajetória temos objetos tridimensionais a frente, e se a superfície de navegação é apropriada para condução.

O treinamento dos modelos foi implementado considerando apenas a representação dos conjuntos de imagens sobre as 3 classes, Navegável, Inavegável e Obstáculos, infelizmente, isso tornou incompatível a avaliação no benchmark padrão do CityScapes para obter a métrica de média da intersecção da união, possibilitando apenas a métrica sobre a classe Navegável que corresponde exclusivamente a classe "road".

A obtenção de conjunto de dados não rotulados tem o potencial de escalar significativamente, pois não depende de avaliação humana para prepará-la, este potencial pode ser visto no artigo "*Global Streetscapes*" ([HOU et al., 2024](#)), que apresenta um conjunto de dados abrangente de **10 milhões de imagens** de ruas, coletadas de **688 cidades em 210 países** e territórios, extraídas através serviços de crowdsourcing como o **Mapillary** e o **KartaView**, similares ao **Google StreetView**.

Com a indisponibilidade de dados rotulados representativos do território brasileiro, nesta monografia é apresentado o dataset **CityScapesBrazil**, composto com um conjunto de **21.485 imagens** obtidas através da plataforma **StreetView** do Google, como motor de navegação para o site ([MAPCHANNELS, 2024](#)), no qual é possível traçar rotas e navegar automaticamente no **StreetView**. Após definido a trajetória, foi capturada a tela ao decorrer do caminho em **30 frames por segundo na dimensão de 512 por 368 pixels**; por fim, com os vídeos da navegação na trajetória gerados foi utilizado o VLC ([VideoLAN Organization, 2024](#)) para extrair screenshots do vídeo, gerando assim o dataset CityScapeBrazil. No Apêndice C, os trajetos estão expostos na figura 35 e 36, e a volumetria de imagens por trajeto está presente na tabela 20, seguido de exemplos das imagens capturadas presente na figura 34.

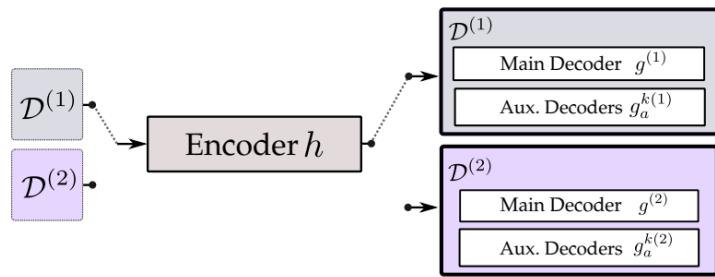
6.2 Uso dos artigos base: LiteSeg, PixMatch, CCT

O modelo **LiteSeg** será o modelo base deste trabalho, que conta com codificador composto pelo backbone **MobileNetV2**, módulo **DASPP**, e decodificador simplificado baseado no **DeepLabV3**, deste modo, espera-se que as arquiteturas que o apliquem adquiram a capacidade de execução em tempo real, visto que as arquiteturas **CCT** e **PixMatch** possuem estruturas adicionais ao modelo apenas para treinamento, como visto nas previsões extras com aumento de dados no **PixMatch**, e os decodificadores auxiliares no **CCT**, assim o tempo de previsão deve ser similar ao do modelo original do **LiteSeg**.

Para esta monografia, o **CCT** será empregado com o foco na estratégia de treinamento semi-supervisionado com o uso da arquitetura de seu codificador, decodificador principal e auxiliares, porém é substituído o modelo base utilizado composto por codificador baseado no **ResNet-50** ([HE et al., 2016b](#)) e **módulos PSP** ([ZHAO et al., 2017a](#)) pelo codificador do **LiteSeg**, assim como para os decodificadores simplificados compostos por convolução 1 x 1 seguido de

três convoluções subpixel com não linearidade ReLU (SHI *et al.*, 2016) para redimensionar a saída ao tamanho original, também pelo decodificador do **LiteSeg**, estabelecendo o **CCTLite**. Para fim de comparação, também será empregado uma versão da arquitetura para treinamento semi-supervisionado para múltiplos domínio, no qual é aplicado um codificador compartilhado, e dois conjuntos (D1 e D2) de decodificador principal e auxiliares, cada um treinando sobre um domínio de dados em específico (D1 e D2), com ilustrado na figura 10.

Figura 10 – CCT aplicado em múltiplos domínios.



Fonte: Elaborada pelo autor.

Já sobre a arquitetura **PixMatch** é empregado devido a sua estratégia de treinamento não supervisionado para adaptação de domínio, visto que não é disponível um conjunto de dados rotulados expressivo para o contexto de cenas urbanas brasileiras. Implementa-se uma versão adaptada do **PixMatch**, chamada **PixMatchLite**, que substitui o modelo base da arquitetura **DeepLabV2** (CHEN *et al.*, 2017) com *backbone* **ResNet-101** (HE *et al.*, 2016b) pelo modelo **LiteSeg** e seu *backbone* **MobileNetV2** e modulo **DASPP**, desta forma é possível atender o contexto de aplicação ao domínio do problema (cenário brasileiro), e ao pré requisito de tempo de execução em **tempo real**.

6.3 Avaliação de resultados

As métricas para avaliar o modelo buscam representar o acerto das regiões delimitadas, ou indiretamente expressar o andamento da aprendizagem do modelo utilizado pelas funções de perda. Neste experimento é empregado a **Média da Interseção sobre União** (i.e., em inglês **MIoU**) sendo uma métrica essencial para avaliar a segmentação semântica, calculando a interseção entre a predição do modelo e a segmentação verdadeira para cada classe, e dividindo pela união dessas áreas, o que proporciona uma visão geral do desempenho em diferentes categorias. Também é aplicado a **Acurácia por Pixel (PA)**, por outro lado, mede a proporção de pixels classificados corretamente, oferecendo uma visão direta da precisão global, embora possa ser influenciada por classes desbalanceadas. Por fim, a função de perda de **entropia cruzada** (i.e., em inglês **Cross-Entropy**) é mensurada no treinamento dos modelos, avaliando a diferença entre a distribuição predita e a verdadeira, e penalizando erros de classificação, o que é aplicado para otimizar as previsões do modelo para alcançar segmentações mais precisas e robustas, pode

também ser usada como métrica indireta para avaliar os modelos. As fórmulas utilizadas para gerar as métricas são expostas em IoU 6.1 e PA 6.2, sobre as máscaras target M_1 e labeled M_2 , e sobre i-nésimo pixel correto p e total de pixel t .

$$IoU = \frac{M_1 \cap M_2}{M_1 \cup M_2} \quad (6.1)$$

$$PA = \frac{\sum_i p_i}{t} \quad (6.2)$$

Ao contrário das métricas anteriores que são utilizadas quando existem rótulos verdadeiros, para o dataset **CityScapesBrazil**, é necessário aplicar métrica que seja capaz de expressar intrinsecamente sobre a predição do modelo para dados não rotulados. Para isso, é aplicado **entropia de Shannon** que trata de uma medida da incerteza e aleatoriedade em um conjunto de dados, e pode ser aplicada na segmentação semântica para dados não rotulados em estratégia de aprendizado semi-supervisionado e não supervisionado para avaliar a informação contida em diferentes regiões da imagem. Ao calcular a entropia, é possível identificar áreas com características semelhantes, ajudando a agrupar pixels em segmentos significativos. A formulação da **entropia de Shannon** está expressa em 6.3, sobre a probabilidade de i-enésima classe c .

$$S(C) = - \sum_i P(c_i) \log_2 P(c_i) \quad (6.3)$$

Serão utilizados 3 subconjuntos dos conjuntos de dados apresentados, ou seja, conjunto de treino, validação e teste. Para essa monografia será aplicado o conjunto de validação para acompanhar o andamento do treinamento dos modelos, e suas melhores épocas serão utilizadas novamente sobre o conjunto de validação para obter as melhores métricas de cada modelo, viabilizando a comparação entre eles, já a base de teste é utilizada para verificar a proximidade dos modelos experimentados com o estado da arte, entretanto para o conjunto de teste do **CityScapes** será utilizado apenas o IoU da classe Navegável, como já mencionado é a única classe compatível com o benchmark do **CityScapes**, demais métricas e classes serão comparadas com a performance no conjunto de validação.

Sobre a classe **Navegável**, é compreendido que trata-se da classe **fundamental** para segmentação semântica para direção autônoma, visto que o acerto desta classe corresponde a acertar o seu respectivo complemento, ou seja, predizendo a classe navegável é possível determinar a área não navegável, que nesta monografia corresponde a classes Inavegável e Obstáculos.

6.4 Escolha e preparação dos modelos de redes neurais

As combinações de bases utilizadas neste trabalho são **GTA5** para **CityScapes** e **CityScapes** para **CityScapesBrazil**, e foram escolhidas para possibilitar tanto análise quantitativa quanto qualitativa. A primeira combinação possibilita a paridade com a literatura por utilizar bases estabelecidas para *benchmarks*, além da obtenção de métricas diretas como **MIoU** e **MPA**, viabilizando uma análise quantitativa sobre os resultados; já a segunda combinação é o alvo desta monografia, no qual será avaliado o resultado qualitativamente por meio de inspeção visual das predições e quantitativamente com a **Entropia de Shannon**, além de sustentado pelo resultado da primeira combinação.

A equipe de desenvolvimento do **LiteSeg** disponibiliza um modelo pré treinado no conjunto de dados do **CityScape** com anotações grosseiras (i.e., **gtCoarse**), que será empregado com objetivo de introduzir algum padrão de extração de característica do domínio de cenas urbanas, assim espera-se que os treinamento posteriores alcancem a convergência mais rápido em comparação a um modelo zero do **LiteSeg**.

Para os experimentos a separação da base de treino, validação e teste, seguiu conforme expresso nas tabelas 1 e 2, no qual vale ressaltar que para o treinamento no **PixMatch** é recomendado pelo artigo o balanceamento de dados entre o domínio supervisionado e não supervisionado, logo para o dataset **CityScapeBrazil**, foi necessário remover o subconjunto de rodovias nordeste sul. Para o CCT foi utilizado o dataset completo.

Tabela 1 – Separação dos subconjuntos de dados de treino, validação e teste, conjuntos CityScapes e GTA5.

Datasets	Treino		Validação		Teste	
	CCT	PIX	CCT	PIX	CCT	PIX
CityScape	2975	2975	500	500	1525	1525
GTA5	12404	2975	6383	6383	6182	6182

Fonte: Elaborada pelo autor.

Tabela 2 – Separação dos subconjuntos de dados de treino, validação e teste, conjunto CityScapesBrazil.

Região	Treino		Validação		Teste	
	CCT	PIX	CCT	PIX	CCT	PIX
Curitiba - SC	675	656	87	87	80	80
Fortaleza - CE	659	640	89	89	79	79
Rio de Janeiro - RJ	1187	1168	133	133	135	135
Porto Alegre - RS	1221	1202	160	160	173	173
São Paulo - SP	1353	1334	157	157	137	137
Rodovias Nordeste-Sul	12093	0	1523	0	1544	0
CityScapeBrazil	~80%		~10%		~10%	

Fonte: Elaborada pelo autor.

Com a pesquisa bibliográfica foram estudadas as arquiteturas com **estratégia baseado em consistência**, sobre dois níveis de perturbações diferentes, **nível de características e entrada**, embora sobre o aspecto de estratégia de treinamento sejam semelhantes ao empregar a perda não supervisionada por consistência, não são equiparáveis sobre o aspecto de perturbações, que possuem características distintas. Entretanto, para esta monografia, serão empregados ambos os

modelos para averiguar o desempenho sobre a aplicação de dois conjuntos de dados rotulados e não rotulados, independente se a abordagem é semi-supervisionada ou adaptação de domínio não supervisionado, ou a perturbação é aplicada em nível de característica ou entrada, logo, busca-se duas arquiteturas baseadas em treinamento por consistência para acomodar o modelo leve **LiteSeg**.

Cada componente nestas arquiteturas merecem estudos posteriores aprofundados, como comparar as estratégias por consistência com outras, por exemplo, redes adversárias, ou, averiguar o desempenho sobre combinações distintas entre perturbações e perturbações em diferentes níveis de aplicação, entrada, características ou saída. Contudo, será abstraído esses estudos para atentar-se ao objetivo de empregar segmentação semântica no **contexto brasileiro** (logo dados não rotulados), e em **tempo real**.

6.4.1 Considerações iniciais

O cenário de aplicação da segmentação semântica neste trabalho busca explorar recursos do estado da arte capazes de atender ao problema de segmentação de **áreas navegáveis** e **não navegáveis** em **tempo real**. Considerando o domínio da aplicação e os requisitos, exploram-se as arquiteturas **CCT** e **PixMatch**, utilizando o modelo leve **LiteSeg** em suas versões **CCTLite** e **PixMatchLite**. Com isso, será avaliado o desempenho das arquiteturas e verificadas algumas premissas.

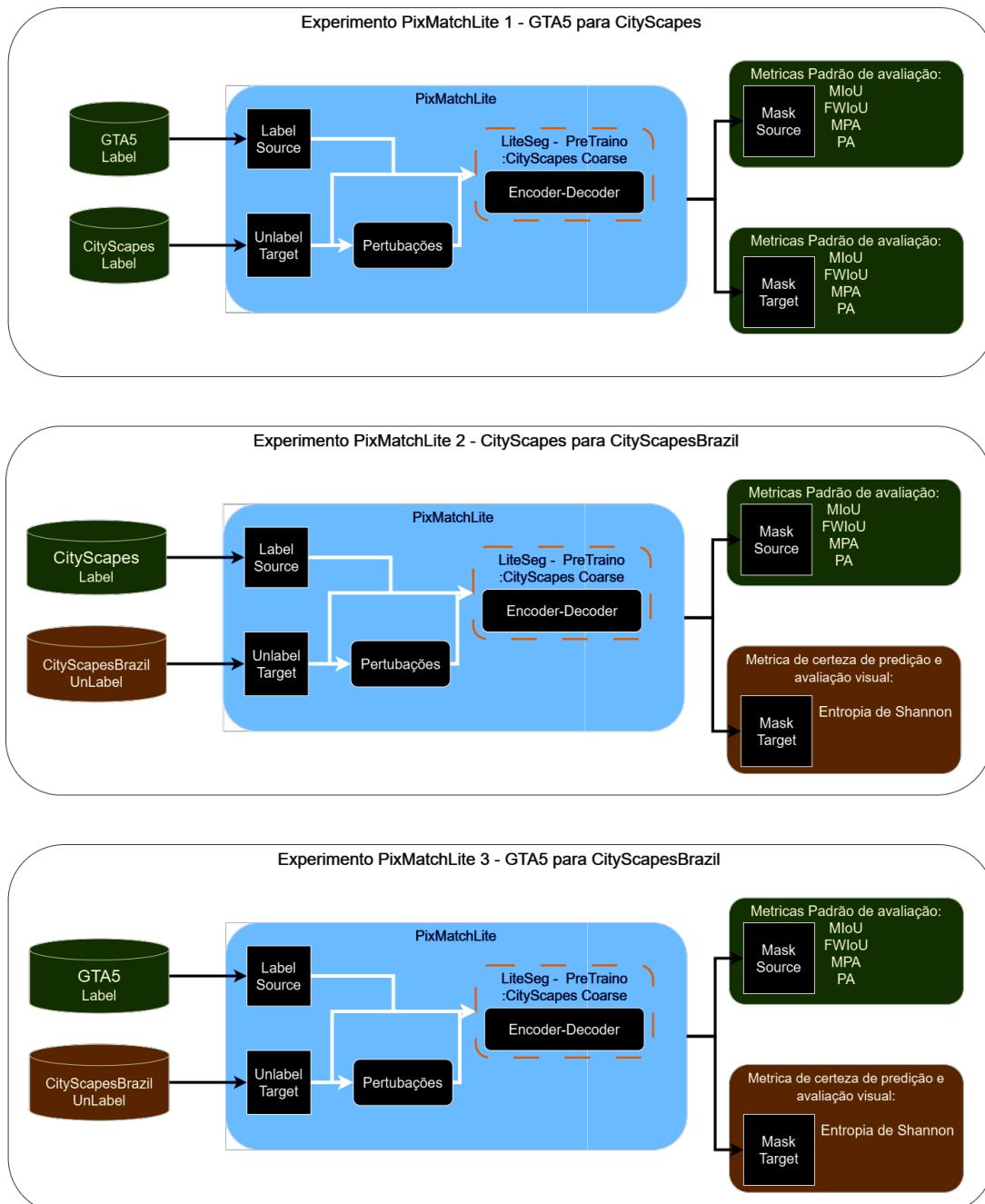
As premissas esperadas na avaliação são:

- Perturbações em nível de características contribuem para tornar o modelo mais robusto a variações, em comparação a perturbações a nível de entrada, ou estratégias de consistência independentes do nível de perturbação são igualmente robustas a variações.
- Uma estratégia semi-supervisionada é capaz de apresentar um bom desempenho em ambos os subconjuntos, rotulados e não rotulados, em comparação a uma estratégia de adaptação de domínio.
- Uma estratégia híbrida, ou seja, treinamento supervisionado e semi-supervisionado em múltiplos domínios, é capaz de apresentar um desempenho equivalente a uma estratégia semi-supervisionada, garantindo a performance em ambos os domínios e subconjuntos (rotulados e não rotulados).
- Um mesmo conjunto treinada de forma supervisionada no modelo consegue apresentar um desempenho semelhante ao de um treinamento com rótulos ocultos.
- Aspectos regionais entre os datasets CityScapes e CityScapesBrazil são significativos para justificar a adaptação de domínio.

6.5 Arquitetura dos modelos

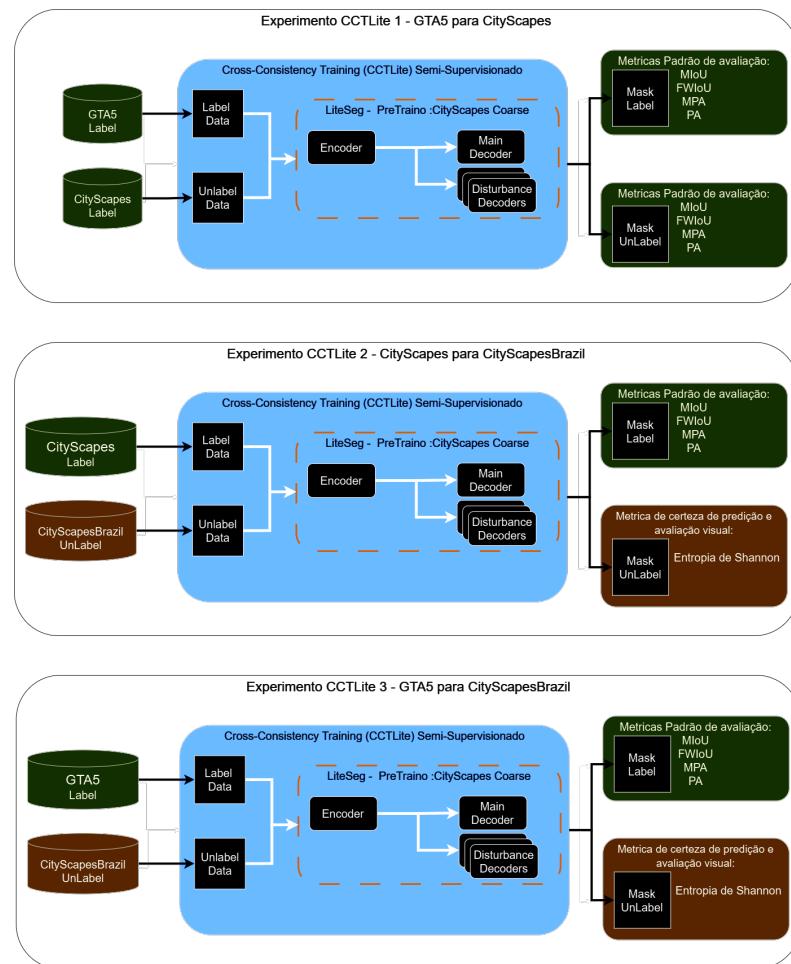
Como já descrito, nas arquiteturas **PixMatch** e **CCT**, serão preservadas com exceção do modelo base das previsões, desta forma, as arquiteturas e as estratégias de treinamento, assim como as combinações dos conjuntos utilizadas, seguem como ilustrado nas figuras 11, 12 e 13.

Figura 11 – Diagrama do modelo PixMatchLite para os treinamento GTA5 para CityScapes, GTA5 para CityScapesBrazil e CityScapes para CityScapesBrazil



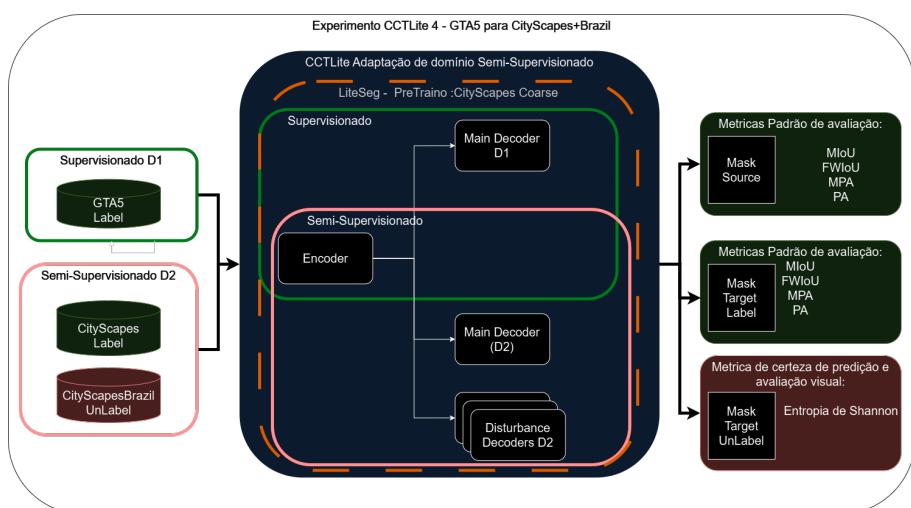
Fonte: Elaborada pelo autor.

Figura 12 – Diagrama do modelo CCTLite para os treinamento GTA5 para CityScapes, GTA5 para CityScapesBrazil e CityScapes para CityScapesBrazil



Fonte: Elaborada pelo autor.

Figura 13 – Diagrama do modelo CCTLite para os treinamento GTA5 e CityScapes para CityScapesBrazil



Fonte: Elaborada pelo autor.

6.6 Parâmetros do experimento

Vale ressaltar que, neste trabalho, não foram reproduzidos os estudos de aderência das funções de perturbação, nem os hiperparâmetros dessas funções ou os hiperparâmetros de treinamento. Aceitamos os resultados obtidos com a melhor configuração de treinamento alcançada nos trabalhos ([MELAS-KYRIAZI; MANRAI, 2021](#)) e ([OUALI; HUDELOT; TAMI, 2020](#)). No entanto, para um novo modelo, seria importante realizar um estudo dos hiperparâmetros de treinamento para verificar a aderência do modelo **LiteSeg** nas arquiteturas **CCT** e **PIX-Match**. Com tudo, serão considerados os hiperparâmetros já verificados no trabalho ([EMARA; MUNIM; ABBAS, 2019](#)). Os hiperparâmetros de treinamento estão apresentados na tabela 3, enquanto os hiperparâmetros das funções de perturbação estão nas tabelas 4 e 5, correspondendo, respectivamente, às arquiteturas PixMatch e CCT.

Tabela 3 – Hiperparâmetros de treinamento para o modelo LiteSeg.

Parâmetro	Valor
Learning rate	$1e^{-7}$
Weight decay learning	$5e^{-4}$
Momentum	0.9
Optimizer	SGD

Fonte: Elaborada pelo autor.

Tabela 4 – Hiperparâmetros peso da perda supervisionada e não supervisionada, além do batch size e época limite.

Parâmetro	Todas as variações de PIX
Loss Supervised and Unsupervised	Cross Entropy
Weight Loss Supervised	1
Weight Loss augmentation data	0.15
Weight Loss square max	0.05
Batch Size	3
Epoch Limit	100

Fonte: Elaborada pelo autor.

Tabela 5 – Hiperparâmetros das funções de perda, assim como a quantidade de decodificadores auxiliares, função de perda, e peso da perda supervisionada e não supervisionada, além do batch size e época limite.

Parâmetro	Todas as variações de CCT
Loss Supervised	Cross Entropy
Loss Unsupervised	Mean Squared Error
Weight Loss Sup	1
Weight Loss Unsup	0-30
Context Masking k decoders	2
Object Masking k decoders	2
Feature Drop k decoders	6
Feature Noise k decodes	6
Feature Noise and Drop Uniform range	0.3
I-VAT k-decoders	2
I-VAT xi	$1e^{-6}$
I-VAT eps	2
Epoch Limit	40
Batch Size	3

Fonte: Elaborada pelo autor.

Capítulo 7

RESULTADOS

Neste capítulo são apresentados os resultados obtidos pelo trabalho sobre as **arquiteturas semi-supervisionada (CCT)** e **adaptação de domínio (PixMatch)**, sobre os conjuntos de **treinamento, validação e teste**, respectivamente, tratados nas seções 7.1 e 7.2, por fim, é apresentado na seção 7.3 a performance dos modelos no sistema de aplicação. Nas subseções 7.1.1 e 7.1.2, trata-se especificamente das discussões sobre o conjunto de treino e validação aplicado respectivamente as variações da arquitetura CCT e PixMatch, por fim, é apresentado um breve estudo sobre a variação dos conjuntos dos domínios **CityScapes** e **CityScapesBrazil** na subseção 7.1.3.

7.1 Treinamento e validação dos modelos escolhidos

No decorrer dos treinamentos foi calculado as métricas para os **7 modelos** implementados ao progresso das épocas, tanto para conjunto de treino quanto para o conjunto de validação, além dos exemplos de predição utilizados para as discussões, todos os gráficos das métricas **MIoU, FWIoU, MPA, PA, Perda e Entropia de Shannon** estão apresentados na seção de apêndices conforme organizado na tabela 6 referente ao Apêndice G.

Tabela 6 – Siglas dos modelos avaliados, descrição e respectivos apêndices.

Sigla	Descrição dos modelos	Apêndice H Imagens
PixGC	PixMatchLite - Adaptação de domínio - GTA5 para CityScapes	48 e 49
PixCB	PixMatchLite - Adaptação de domínio - CityScapes para CityScapesBrazil	52 e 53
PixGB	PixMatchLite - Adaptação de domínio - GTA5 para CityScapesBrazil	54 e 55
CCTGC	CCTLite - Semi-Supervisionado - GTA5 e CityScapes	56, 58 e 57
CCTCB	CCTLite - Semi-Supervisionado - CityScape e CityScapesBrazil	59, 60 e 61
CCTGB	CCTLite - Semi-Supervisionado - GTA5 e CityScapesBrazil	62, 63 e 64
CTTMDGCB	CCTLite - Múltiplos domínios - GTA5 para CityScapes e CityScapesBrazil	65, 66, 67 e 68

Fonte: Elaborada pelo autor.

Sobre os gráficos do conjunto de **validação** obteve a melhor pontuação utilizando a métrica **MIoU** para cada modelo e respectivo conjunto de imagens, que será utilizado para escolher as épocas de cada modelo para compor os resultados da seção 7.2.

7.1.1 Treinamento das variações CCT

Na perspectiva de uma estratégia híbrida, que combina treinamento supervisionado e semi-supervisionado em múltiplos domínios, avaliamos a capacidade do modelo híbrido de

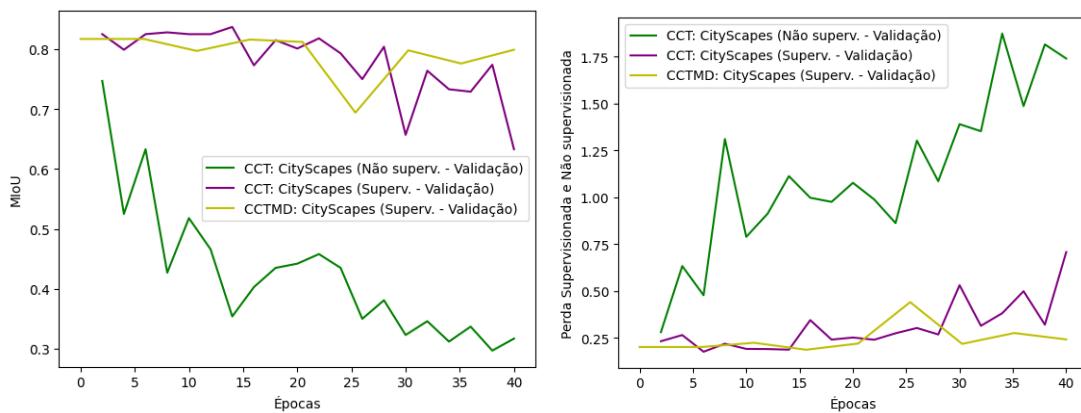
alcançar desempenho semelhante ao de um modelo semi-supervisionado, garantindo eficácia em ambos os domínios e em seus subconjuntos (rotulados e não rotulados). Nas figuras 14, 15 e 16, observamos que, nas bases CityScapes e CityScapesBrazil, ambos os modelos alcançaram desempenho comparável: o modelo CCT (**CCTCB**) e sua variação para múltiplos domínios (**CTTMDGCB**). Por outro lado, na base GTA5, o modelo CTTMDGCB proporcionou estabilidade no treinamento, mas ficou aquém do **CCTGC**, apresentando um máximo de **MIoU de 84%**, comparado aos **87% do CCTGC**. O modelo **CCTGB**, por sua vez, obteve resultados insignificantes.

Adicionalmente, uma abordagem de treinamento em múltiplos domínios permite **flexibilidade** na condução do treinamento, seja por meio do treinamento individual dos modelos ou em conjunto. O codificador compartilhado entre os domínios não demonstrou nem favorecer nem prejudicar o treinamento de nenhuma das bases no modelo **CTTMDGCB**, viabilizando, assim, um modelo único que desempenha o papel de modelos específicos para cada domínio.

Ao comparar os modelos **CCTGB** e **CCTGC**, observamos que a combinação de **GTA5** com **CityScapesBrazil** **prejudicou** o desempenho do treinamento supervisionado, conforme evidenciado na figura 15. O conjunto CityScapesBrazil apresentou o pior valor de entropia de Shannon entre as variações do CCT.

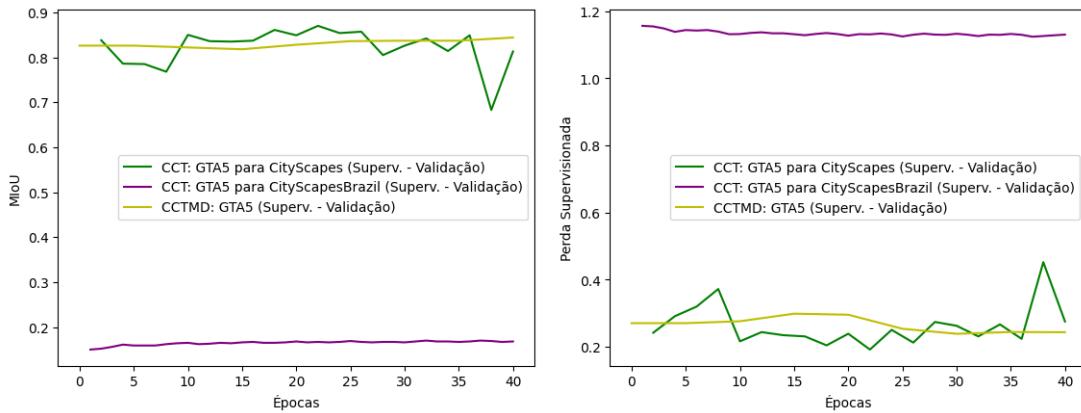
Por fim, ao analisarmos a premissa de que um conjunto treinado de forma supervisionada possa apresentar desempenho similar ao de um conjunto com rótulos ocultos, a figura 14 revela que, apesar da tendência decrescente do **MIoU** para o conjunto **CityScapes** durante o treinamento não supervisionado (**CCTGC**), o desempenho inicial de **75% em MIoU** nas primeiras épocas é significativo em comparação com os demais modelos. Esta curva também indica um sinal de **sobreajuste**, possivelmente decorrente do uso do modelo **LiteSeg pré-treinado com rótulos grosseiros**, que superou o ponto de convergência dos modelos que utilizam **CityScapes** para treinamento.

Figura 14 – Comparando conjunto de **validação** CityScapes como Supervisionado em semi-supervisionado e semi-supervisionado em múltiplos domínios, além da estratégia não supervisionada em adaptação de domínio.



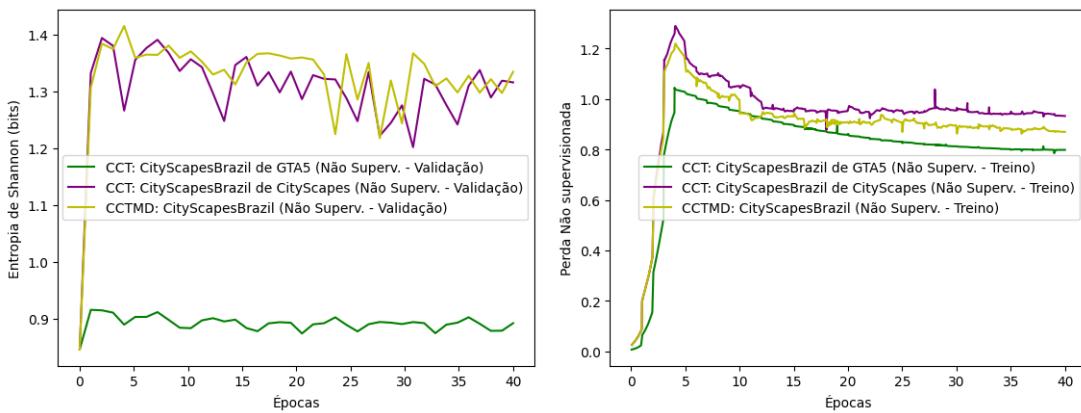
Fonte: Elaborada pelo autor.

Figura 15 – Comparando conjunto de **validação** GTA5 treinado de forma supervisionada para CityScapes e CityScapesBrazil como semi-supervisionado e múltiplos domínios.



Fonte: Elaborada pelo autor.

Figura 16 – Comparando conjunto de **validação** CityScapesBrazil treinado de forma não supervisionado em semi-supervisionado a partir do CityScapes e GTA5, e múltiplos domínios.



Fonte: Elaborada pelo autor.

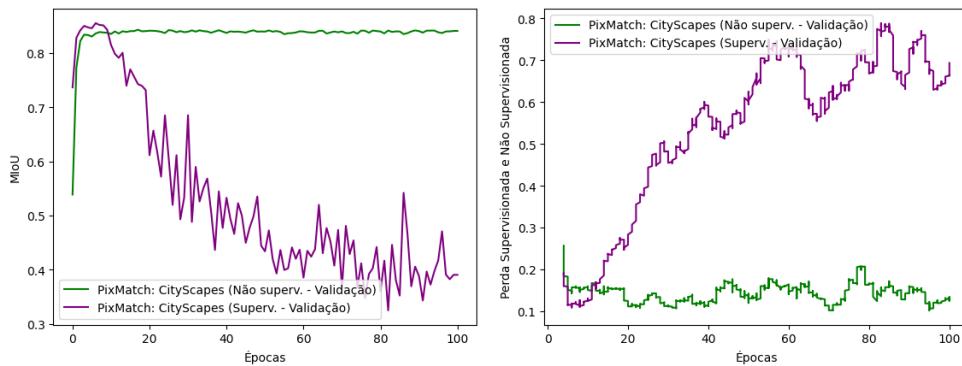
7.1.2 Treinamento das variações PixMatch

Na figura 17, observamos que o conjunto **CityScapes**, tanto em **dados rotulados** quanto **não rotulados**, obteve resultados excelentes em termos da métrica **MIoU**. A abordagem de adaptação de domínio não supervisionado alcançou resultados próximos aos da abordagem supervisionada. É importante destacar que o modelo **LiteSeg**, pré-treinado no conjunto de rotulagem grosseira do CityScapes, pode ter influenciado os altos valores das métricas nas primeiras épocas, antecipando a convergência dos modelos que utilizam o conjunto CityScapes com anotações finas. Esse efeito também é perceptível nas figuras 14, 15 e 16, referentes às variações CCT.

Por outro lado, ao analisar o treinamento supervisionado do conjunto **GTA5**, verificamos que o modelo **PIXGB** é **negativamente** influenciado pelo conjunto **CityScapesBrazil**, não conseguindo adaptar-se adequadamente a esse domínio, conforme evidenciado nos gráficos das figuras 18 e 19. Entretanto, os modelos **PIXCB** e **PIXGC** demonstraram resultados excelentes

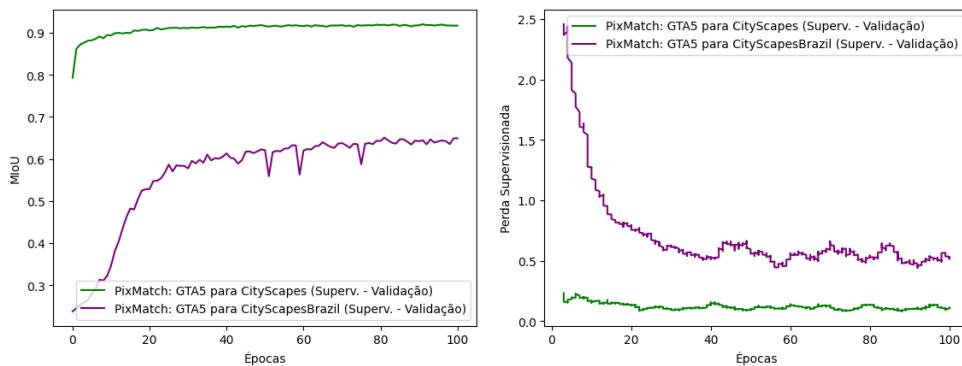
tanto no domínio supervisionado quanto no não supervisionado.

Figura 17 – Comparando conjunto de **validação** CityScapes como Supervisionado (PIXCB) e Não Supervisionado (PIXGC).



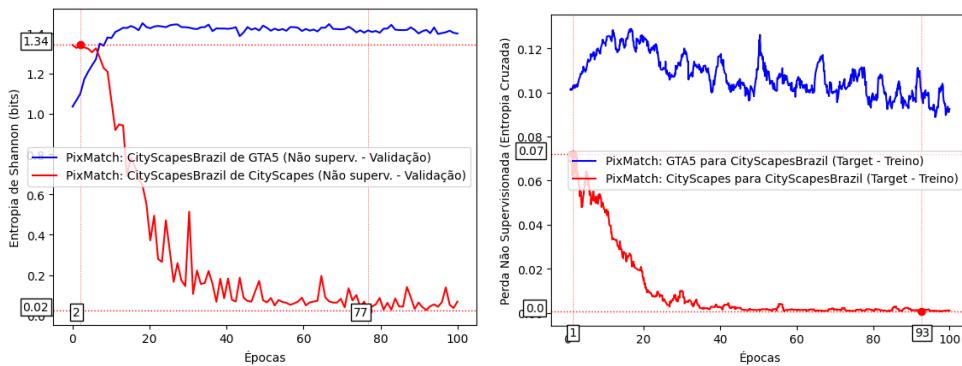
Fonte: Elaborada pelo autor.

Figura 18 – Comparando conjunto de **validação** GTA5 como Supervisionado adaptando os domínios CityScapes e CityScapesBrazil.



Fonte: Elaborada pelo autor.

Figura 19 – Comparando conjunto de **validação** CityScapesBrazil como Não Supervisionado adaptado a partir dos domínios CityScapes e GTA5.



Fonte: Elaborada pelo autor.

7.1.3 Teste de variação entre os domínios **CityScapes** e **CityScapes-Brazil**

Neste tópico, exploramos a premissa de que existem aspectos regionais significativos entre os conjuntos **CityScapes** e **CityScapesBrazil** que justificam a adaptação de domínio. Para isso, realizaremos uma avaliação cruzada entre as bases e os modelos em **predições "zero-shot"**, verificando se o conjunto **CityScapesBrazil** pode ser considerado equivalente ao **CityScapes** ou se, de fato, representa um domínio distinto. Esse entendimento é crucial para determinar a melhor forma de utilizar dados não rotulados, considerando as variações geográficas de captação, especificamente em cenas urbanas da Alemanha e do Brasil.

7.1.3.1 Testando a variação dos domínios **CityScapes** e **CityScapesBrazil** nas variações CCT

Aplicamos o conjunto de dados do **CityScapesBrazil** no modelo **PixGC**, no qual não foi utilizado para treinamento, para verificar se a adaptação de domínio realizada de **GTA5** para **CityScapes** é também suficiente para o conjunto **CityScapesBrazil**. A entropia de Shannon para esse cenário é apresentada na tabela 7. Além disso, realizamos uma avaliação visual de alguns exemplos de predição para ambos os modelos sobre a mesma base (**CityScapesBrazil**), conforme mostrado no Apêndice F figura 44.

O mesmo procedimento foi realizado com o modelo **CCTGB**, utilizando a base **CityScapes**, que não foi usada para o treinamento nessa variação. As predições dos modelos **CCTGB**, **CCTGC** e **CCTCB** foram comparadas em exemplos de imagens, apresentados no Apêndice F figura 45, e suas métricas para a base de validação estão na tabela 8.

Tabela 7 – Entropia de Shannon para **CityScapesBrazil** aplicado aos modelos CCTGC, CCTCB e CTTMDGCB.

Modelo	Shannon Val (bit)	Shannon Test (bit)
CCTGC	1,085	1,095
CCTCB	1,378	1,403
CTTMDGCB	1,367	1,377

Fonte: Elaborada pelo autor.

Tabela 8 – MIoU e IoU sobre as classes Navegável, Inavegável e Obstáculos para **CityScapes** conjunto de validação aplicado aos modelos CCTGB, CCTGC, CCTCB e CTTMDGCB.

Modelo	MIoU	IoU Navegável	IoU Inavegável	IoU Obstáculos
CCTGB	17,0	16,72	0,00	34,40
CCTGC	73,8	75,94	51,67	93,90
CCTCB	83,2	88,15	65,12	96,18
CTTMDGCB	85,2	90,96	68,15	96,62

Fonte: Elaborada pelo autor.

7.1.3.2 Testando a variação dos domínios CityScapes e CityScapesBrazil nas variações Pix-Match

Assim como nas variações do CCT, aplicamos o conjunto de dados do **CityScapesBrazil** no modelo **PIXGC**, sem treinamento, para verificar se a adaptação de domínio de **GTA5** para **CityScapes** é suficiente para o conjunto **CityScapesBrazil**. A entropia de Shannon para esse cenário é apresentada na tabela 9, juntamente com uma avaliação visual de um exemplo de predição para ambos os modelos sobre a mesma base, conforme mostrado no Apêndice F figura 46.

Realizamos o mesmo para a base CityScapes no modelo PixGB, onde comparamos as métricas alcançadas entre as variações PixGB, PixGC e PixCB na tabela 10, e exemplos são apresentados no Apêndice F figura 47.

Tabela 9 – Entropia de Shannon para CityScapesBrazil aplicado aos modelos PixGC e PixCB.

Modelo	Shannon Val (bit)	Shannon Test (bit)
PixGC	1,232	1,242
PixCB	1,371	1,394

Fonte: Elaborada pelo autor.

Tabela 10 – MIoU e IoU sobre as classes Navegável, Inavegável e Obstáculos para CityScapes conjunto de validação aplicado aos modelos PixGB, PixGC e PixCB.

Modelo	MIoU	IoU Navegável	IoU Inavegável	IoU Obstáculos
PixGB	68,7	75,94	44,74	85,55
PixGC	84,2	91,18	65,54	95,88
PixCB	85,4	92,98	67,38	95,77

Fonte: Elaborada pelo autor.

7.1.3.3 Conclusão do Teste de Domínio

Analisando as métricas quantitativas, observamos que para o conjunto **CityScapesBrazil**, a entropia de Shannon apresentada nas tabelas 7 (CCT) e 9 (PIX) revela que as variações treinadas CCTGC e PIXGC ficaram abaixo do modelo adaptado para **CityScapesBrazil**. Esse resultado também é corroborado pela avaliação visual dos exemplos apresentados no Apêndice F figuras 44 e 46. Assim, os modelos treinados com a combinação de **GTA5** para **CityScapes** não conseguem desempenhar adequadamente no domínio **CityScapesBrazil** sem adaptação.

Para o conjunto **CityScapes**, constatamos resultados semelhantes nas métricas de MIoU e IoU por classe, conforme mostrado nas tabelas 8 (CCT) e 10 (PIX). Os modelos com pior desempenho foram **PIXGB** e **CCTGB**, o que também foi evidenciado na análise visual no Apêndice F figuras 45 e 47.

Portanto, a avaliação quantitativa e qualitativa das **predições "zero-shot"** nos conjuntos **CityScapesBrazil** e **CityScapes**, utilizando os modelos **PIXGC/CCTGC** e **PIXGB/CCTGB**,

sugere que esses conjuntos podem ser considerados **domínios distintos**. No entanto, outras variáveis podem impactar a replicabilidade do desempenho nas **previsões "zero-shot"**, como características das câmeras de captura, posicionamento e enquadramento das imagens, e o balanceamento de cenários e classes. Além disso, desconsiderando as diferenças técnicas de captura, as variações são também influenciadas por condições climáticas, arquitetônicas e regionais das localizações das capturas, consolidando assim a distinção entre os dois domínios.

7.2 Resultados em Conjunto Teste e Validação

Todos os resultados para as métricas **MIoU**, **MPA** e **entropia de Shannon**, referentes aos conjuntos de **teste** e **validação**, estão organizados no Apêndice E figura 23. Para avaliar a performance dos modelos, analisamos separadamente os resultados obtidos em teste e validação para as classes Navegável, Inavegável e Obstáculos, focando nas métricas **MIoU** e **IoU por classe**, a fim de identificar as discrepâncias entre os modelos. É importante ressaltar que, no conjunto de dados de teste do **CityScapes**, não é possível reproduzir as classes Inavegável e Obstáculos dentre as 19 classes disponíveis para **submissão ao benchmark** oficial. Portanto, para este conjunto, utilizaremos apenas a classe Navegável. Os recortes do Apêndice E estão apresentados nas tabelas 24 e 25, correspondendo, respectivamente, aos conjuntos de **validação** e **teste**.

7.2.1 Conjunto Rotulado

Verificamos o desempenho dos modelos na base CityScapes, tanto como domínio de **destino (Target)** quanto aplicado como conjunto não rotulado (**Unlabel**) em uma estratégia semi-supervisionada. Para isso, organizamos um recorte da tabela 24 no Apêndice E, e apresenta-se apenas o conjunto de **validação CityScapes** na tabela 11, abrangendo as condições **Target/Unlabel e Source/Label**.

Ao avaliar o desempenho no **domínio de origem (Source)** com **dados rotulados (Label)**, observamos que o modelo **CCT** obteve o melhor desempenho na base **GTA5**, tanto no conjunto de **validação** quanto no de teste (conforme expresso na tabela 12). Para o conjunto CityScapes, as melhores performances foram registradas em ambas as arquiteturas **PIX** e **CCT**, com uma diferença **mínima de 0,2 na métrica MIoU**, conforme mostrado na tabela 11.

Tabela 11 – Conjunto CityScape aplicada como *source (label)* e *target(unlabel)* nos modelos avaliados sobre as métricas MIoU e Shannon no conjunto de **validação**.

Modelo	Aplicação	MIoU Val	Dif MIoU Máx	IoU Navegável Test	Dif IoU Navegável Máx	Shannon (bit)
PixGC	Target/Unlabel	84,2	0,0	95,38	0,00	1,316
CCTGC	Target/Unlabel	73,8	-10,4	87,86	-7,52	1,310
PixCB	Source/Label	85,4	0,0	91,06	0,00	1,291
CCTCB	Source/Label	83,2	-2,2	92,41	1,35	1,325
CTTMDGCB	Source/Label	85,2	-0,2	93,82	2,76	1,317

Fonte: Elaborada pelo autor.

Tabela 12 – Conjunto GTA5 aplicada como source nos modelos avaliados sobre as métricas MIoU e Shannon no conjunto de **validação e teste**.

Modelo	MIoU Validação	Dif Val	MIoU Teste	Dif Teste	Shannon Val (bit)	Shannon Test (bit)
PixGC	88,4	-0,8	88,4	-0,7	1,398	1,413
CCTGC	89,2	0,0	89,1	0,0	1,389	1,435
CTTMDGCB	87,9	-1,3	87,3	-1,8	1,392	1,430

Fonte: Elaborada pelo autor.

7.2.2 Conjunto Não Rotulado

Ao analisar os resultados para dados não rotulados, ou seja, o conjunto aplicado no modelo como **Target**, observamos que o conjunto **CityScapes** foi segmentado de maneira mais eficaz pelo modelo **PIX** em comparação ao modelo **CCT**, conforme apresentado na tabela 11. Em relação ao conjunto **CityScapesBrazil**, a tabela 13 indica que o melhor desempenho também foi alcançado pelo modelo de arquitetura **CCT**, mas a arquitetura **PIX** também apresentou bons resultados, isso é corroborado pela inspeção visual na figura 20. Portanto, considerando especificamente os *datasets* **CityScapes** e **CityScapesBrazil** em conjunto a tarefa de adaptação de domínio demonstrou melhores resultados na manipulação de um conjunto de imagens sem rótulos para múltiplos domínios.

Tabela 13 – Resultado obtido para o conjunto CityScapeBrazil avaliado pela entropia de Shannon, sobre o conjunto de **teste**.

Modelo	Shannon	Diferença Máx
PixCB	1,394	-0,009
CCTCB	1,403	0,000
CTTMDGCB	1,377	-0,026

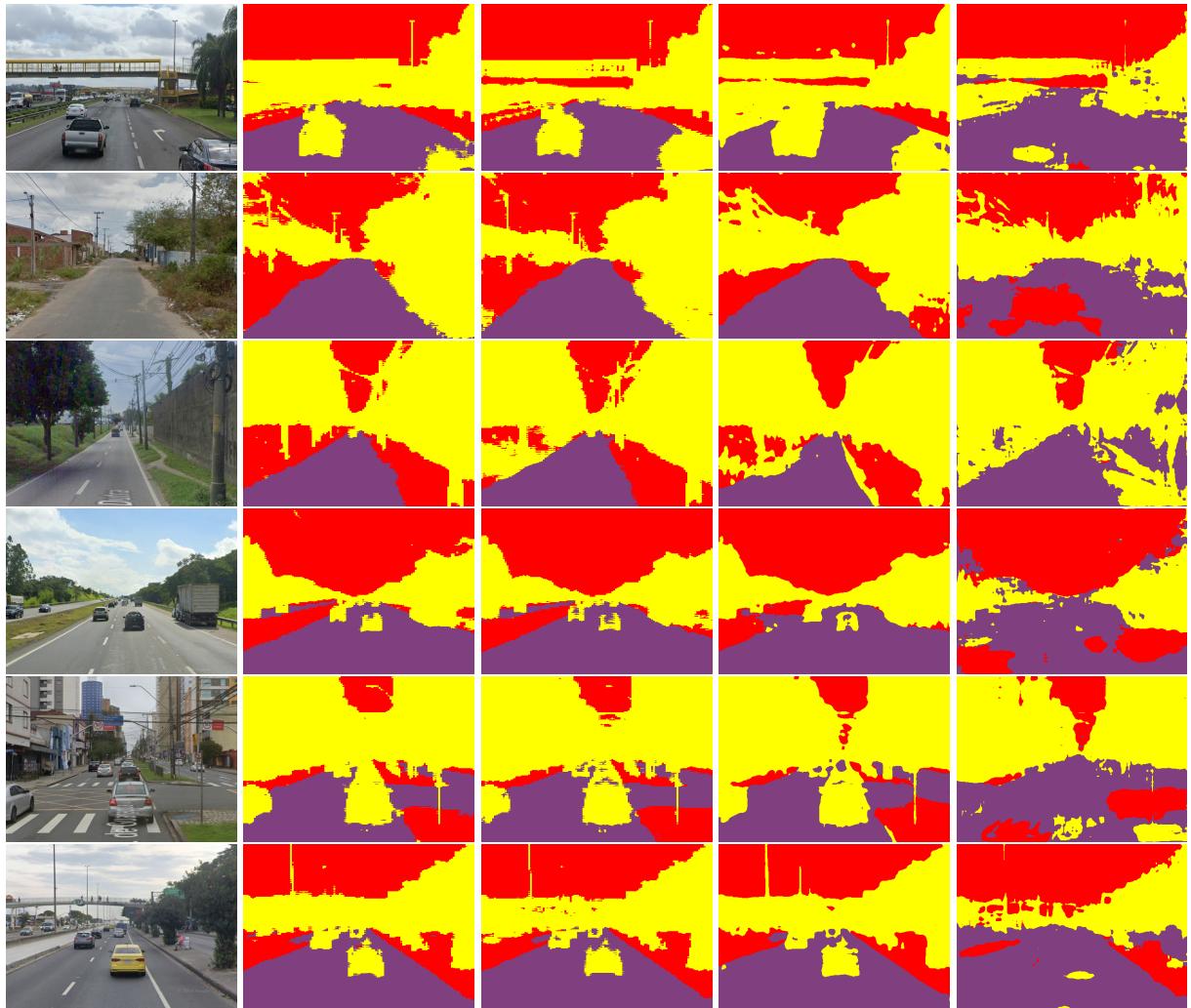
Fonte: Elaborada pelo autor.

7.2.3 Considerações Finais

Os resultados alcançados podem estar relacionados à característica do treinamento semi-supervisionado em favorecer a classe rotulada na predição do modelo. Na arquitetura **CCT**, o decodificador principal é treinado especificamente sobre o subconjunto rotulado, enquanto o conjunto não rotulado apenas reforça o treinamento de características de baixo nível no codificador. Em contrapartida, na arquitetura **PIX**, ambos os subconjuntos são utilizados para treinar o modelo completo (codificador e decodificador). Assim, os resultados estão alinhados com a expectativa de que um conjunto não rotulado é **priorizado** no modelo de **adaptação de domínio não supervisionada**, ao contrário do modelo treinado de forma **semi-supervisionada**.

Os resultados indicam que perturbações a nível de entrada, aplicadas por meio de ***data augmentation***, demonstram **maior eficiência computacional** (corroborado pelo tempo de treinamento apresentado na figura 19) e de desempenho nas variações do modelo **PIX** em comparação às perturbações empregadas no nível de características no modelo **CCT**. Embora as abordagens sejam distintas (semi-supervisionado e adaptação de domínio não supervisionada),

Figura 20 – Exemplos do conjunto de **teste** do CityScapesBrazil na primeira coluna, e suas previsões nos modelos CCTCB, CTTMDGCB, PIXCB e PIXGB nas respectivas colunas seguintes.



Fonte: Elaborada pelo autor.

observamos que, para o conjunto **CityScapes**, o modelo **PIXGC** superou o **CCTGC**, conforme mostrado na tabela 11. Vale ressaltar que não podemos tirar conclusões definitivas sobre as funções de perturbação utilizadas e o nível de aplicação, pois seria necessário realizar um estudo aprofundado a esse respeito. Essa investigação foge ao escopo deste trabalho e à capacidade computacional disponível para treinar mais variações de modelos, o que permitiria um estudo mais abrangente das funções de perturbação e seus níveis de aplicação (entrada, característica ou predição).

O objetivo da combinação **GTA5** para **CityScapes** é viabilizar a comparação dos modelos utilizando métricas quantitativas, estabelecendo assim uma referência em relação aos modelos desenvolvidos nesta monografia e à literatura existente. Em relação ao conjunto de testes, as previsões dos modelos para o conjunto **CityScapes** foram submetidas ao *benchmark* oficial, e os resultados obtidos em comparação a outros métodos estão apresentados na tabela 22 no Apêndice D.1. Ressalta-se que, neste trabalho, alcançamos a excelente marca de **95,4% de IoU** para a

classe Navegável na variação do modelo **PIXGC**, representando a possibilidade do modelo de atingir um alto patamar no estado da arte em segmentação semântica para adaptação de domínio não supervisionada, em comparação com outros métodos organizados na tabela 22 no Apêndice D.1.

Para realizar uma comparação mais justa com a literatura foi treinado uma versão com 19 classes (no padrão **CityScapes** presente na tabela 21) do modelo **PIXGC** (**PIXGC19**) para submissão no **CityScapes benchmark** oficial, seu resultado está organizado nas tabelas 14, 15 e 16, e a evolução no decorrer das épocas exposta nas figuras 50 e 51 no Apêndice G, na qual podemos verificar que ficou abaixo aos artigos originais (MELAS-KYRIAZI; MANRAI, 2021) e (EMARA; MUNIM; ABBAS, 2019), demonstrando que a simplificação de classes afeta positivamente os modelos, priorizando classes relevantes. Nota-se que, mesmo sobre um modelo treinado por múltiplas classes minoritárias, **PIXGC19**, obteve-se a marca de **94,5% de IoU** para classe "road"(Navegável).

Tabela 14 – MIoU e IoU por classe para o modelo PIXGC para 19 classes CityScapes conjunto de teste.

Metric	Average	road	sidewalk	building	wall	fence	pole	trafficlight	trafficsign	vegetation
IoU	47,81	94,53	65,70	85,47	40,35	35,01	26,21	31,51	27,07	88,42
iIoU	21,14	-	-	-	-	-	-	-	-	-
Metric	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle
IoU	58,28	90,80	61,57	9,24	87,16	30,21	48,99	13,58	13,94	0,31
iIoU	-	-	39,91	2,38	76,46	17,23	21,57	7,55	3,84	0,19

Fonte: Elaborada pelo autor.

Tabela 15 – MIoU e IoU por classe para o modelo PIXGC para 7 categorias CityScapes conjunto de teste.

Metric	Average	flat	nature	object	sky	construction	human	vehicle
IoU	76,35	95,64	87,47	28,49	90,80	85,44	63,08	83,52
iIoU	56,33	-	-	-	-	-	41,33	71,33

Fonte: Elaborada pelo autor.

Tabela 16 – MIoU e IoU por classe para o modelo PIXGC para 19 classes agrupamento Navegável, Inavegável e Obstáculos para CityScapes conjunto de teste.

Metrica	Média	Navegável	Inavegável	Obstáculos
IoU	68,69	94,53	71,59	39,94

Fonte: Elaborada pelo autor.

Conclui-se que a abordagem baseada em adaptação de domínio não supervisionada é ideal para lidar com dados não rotulados, principalmente, quando identificado o contexto de adaptação de domínio, devido ao excelente resultado de MIoU verificado na combinação GTA5 para CityScapes, à alta entropia de Shannon na combinação CityScapes para CityScapesBrazil e à análise visual dos exemplos preditos expostos na figura 20.

7.3 Performance e Sistema de aplicação

Nesta seção, estão organizadas as especificações do sistema, o tempo de predição e o tempo de treinamento, apresentados respectivamente nas tabelas 17, 18 e 19. Vale ressaltar que, para a taxa de predição, foram geradas aleatoriamente matrizes com dimensões 1x3x360x640 e 1x3x512x1024, submetendo 1000 amostras; por fim, somou-se o tempo total de predição e dividiu-se pelo número de amostras, obtendo assim frames por segundo.

Tabela 17 – Especificações dos sistema de aplicação dos modelos.

Especificação	Valor
Processador	Ryzen 5 3400G
Placa de Vídeo	RTX 4060 (VRAM 8GB)
Memória RAM	16 GB
Sistema Operacional	Windows 11 - 23H2

Fonte: Elaborada pelo autor.

Tabela 18 – Tempo de predição em Frames por Segundo (FPS), métrica de computação em Flops e número de parâmetros treináveis dos modelos.

Modelo	FPS (360x640)	FPS (1024x2048)	Flops	Nº Parâmetros
Pix	62	30	4.84GMac	4.38M
CCT	66	30	4.85GMac	5.08M
CCTAD	67	30	4.85GMac	5.11M

Fonte: Elaborada pelo autor.

Tabela 19 – Tempo de treinamento aproximado por modelo.

Modelo	Épocas Máxima	Tempo por época	Tempo total*
PixMatch: GTA5 para CityScape	100	20 min	36h30min
PixMatch: GTA5 para CityScape (19 classes)	100	1h30 min	7d23h30min
PixMatch: CityScape para CityScapeBrazil	100	15 min	28h15min
PixMatch: GTA5 para CityScapeBrazil	100	20 min	38h30min
CCT: GTA5 para CityScape	40	1h	53h15min
CCT: CityScape para CityScapeBrazil	40	50min	46h30min
CCT: GTA5 para CityScapeBrazil	40	45min	42h30min
CCTAD: GTA5 para CityScape e CityScapeBrazil	40	1h20min	66h30min

Fonte: Elaborada pelo autor.

* Tempo total inclui tempo de treinamento de todas as épocas, e tempo de validação.

Capítulo 8

CONCLUSÃO

Neste trabalho, buscamos aplicar modelos e arquiteturas avançadas para solucionar o problema da segmentação semântica em tempo real em rodovias e estradas brasileiras. Para isso, exploramos técnicas de treinamento semi-supervisionado e adaptação de domínio não supervisionada, integradas a um modelo leve e eficiente o suficiente para execução em tempo real. Também utilizamos estratégias de treinamento por consistência, aplicando funções de perturbação para introduzir regularização não supervisionada.

Durante o desenvolvimento, utilizamos os conjuntos de dados CityScapes, GTA5 e CityScapesBrazil, optando por agrupar as classes em três categorias principais: Navegável, Inavigável e Obstáculos, facilitando tanto a análise quanto o treinamento dos modelos. Adaptamos o modelo LiteSeg para as arquiteturas CCT e PixMatch, priorizando eficiência e treinamento semi-supervisionado e não supervisionado. A avaliação dos modelos foi feita com base em métricas como Média da Intersecção da União e Acurácia por Pixel, além da aplicação de entropia de Shannon para medir a incerteza em dados não rotulados. As decisões sobre divisão de dados (treino, validação e teste) e configurações de hiperparâmetros foram baseadas em referências da literatura, evitando grid search e otimizando a performance.

Com base nos resultados, confirmamos as premissas levantadas, como as estratégias combinadas de treinamento semi-supervisionado em múltiplos domínios podem apresentar desempenho semelhante ao treinamento isolado por domínio. Além disso, verificamos que a adaptação de domínio pode alcançar resultados no domínio destino próximos aos do treinamento supervisionado como domínio de origem. Observamos também que os domínios CityScapes e CityScapesBrazil são distintos, reforçando a importância de priorizar o domínio de destino durante a adaptação. Nesse sentido, a adaptação de domínio mostrou-se vantajosa ao dar mais relevância a dados não rotulados. Por fim, constatamos que, neste trabalho, as perturbações aplicadas no nível da entrada (para PixMatch, data augmentation) foram mais eficientes, tanto em termos computacionais quanto em desempenho, em comparação às aplicadas no nível de características internas.

Com base nas análises quantitativas e qualitativas, concluímos que a arquitetura PixMatch, integrada ao modelo base LiteSeg, se mostrou eficiente para adaptação de domínio não supervisionada com treinamento por consistência. Esse conjunto apresentou desempenho robusto para a segmentação semântica de cenas urbanas, atingindo 95,4% de IoU para a classe "road" no conjunto de teste do CityScapes, com uma taxa de 30 FPS em imagens de alta resolução.

lução (2048x1024 pixels). Além dos bons resultados quantitativos, inspeções visuais também confirmaram a eficácia do modelo ao lidar com o conjunto CityScapesBrazil.

Assim, esta monografia entrega um modelo eficiente de segmentação semântica em tempo real, com foco na distinção entre áreas navegáveis e não navegáveis em cenas urbanas brasileiras. Além disso, consolidamos a relevância do CityScapesBrazil, um conjunto diversificado com 21.485 imagens que abrangem diferentes regiões do Brasil, do Nordeste ao Sul.

8.1 Trabalhos Futuros

Durante o treinamento, identificamos que a aplicação de *grid search* para ajuste de hiperparâmetros poderia melhorar significativamente o desempenho dos modelos, especialmente ao combinar as arquiteturas PixMatch e CCT com o modelo base LiteSeg. Como essas combinações ainda não foram exploradas pelos autores das arquiteturas originais, estudos futuros que otimizem essa integração podem oferecer novos avanços em desempenho.

Neste trabalho, restringimos a aplicação de funções de perturbação a abordagens básicas. No entanto, para um treinamento não supervisionado por consistência mais robusto, é fundamental investigar mais a fundo tipos, intensidade e contexto de aplicação das perturbações. A escolha inadequada dessas transformações pode introduzir ruído excessivo ou prejudicar o aprendizado, desviando o modelo de seu objetivo. Futuras pesquisas podem explorar perturbações específicas para diferentes cenários e avaliar seu impacto no desempenho.

Os resultados mostraram que as características visuais dos diferentes conjuntos de dados (como CityScapes e CityScapesBrazil) afetam a generalização dos modelos. Isso evidencia a necessidade de capturas mais consistentes e adequadas ao contexto de navegação autônoma, superando a simplicidade das imagens obtidas por sistemas como o Google Street View. Trabalhos futuros podem focar em capturas otimizadas para simular a perspectiva de câmeras específicas para veículos autônomos, considerando o posicionamento, enquadramento e equilíbrio de classes e cenários no momento da coleta, além de acrescentar outros sensores como radar.

Nesta pesquisa, agrupamos as classes em três categorias: Navegável, Inavegável e Obstáculos, suficientes para garantir uma navegação segura. No entanto, um sistema completo de navegação autônoma exige maior complexidade na segmentação semântica. Por exemplo, diferenciar pedestres de veículos é essencial, já que a prioridade de reação pode variar, ou seja, desviar de um pedestre é mais crítico do que de outro veículo. Pesquisas futuras podem ampliar a granularidade da classificação, criando categorias como pedestres, veículos e ciclistas. Além disso, a integração de módulos de tomada de decisão mais sofisticados pode enriquecer a percepção e a resposta do veículo em diferentes contextos urbanos, tornando a navegação ainda mais segura e eficiente.

REFERÊNCIAS

- BARHOUMI, Y.; RASOOL, G. Scopeformer: N-cnn-vit hybrid model for intracranial hemorrhage classification. **arXiv preprint arXiv:2107.04575**, 2021. Citado na página 82.
- BOTTESINI, G. Influência de medidas de segurança de trânsito no comportamento dos motoristas. 2010. Citado na página 86.
- CHEN, L.-C. Rethinking atrous convolution for semantic image segmentation. **arXiv preprint arXiv:1706.05587**, 2017. Citado na página 35.
- CHEN, L.-C.; PAPANDREOU, G.; KOKKINOS, I.; MURPHY, K.; YUILLE, A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 40, n. 4, p. 834–848, 2017. Citado 4 vezes nas páginas 41, 51, 87 e 93.
- CHEN, L.-C.; ZHU, Y.; PAPANDREOU, G.; SCHROFF, F.; ADAM, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. **Proceedings of the European conference on computer vision (ECCV)**, p. 801–818, 2018. Citado 3 vezes nas páginas 31, 38 e 102.
- CHEN, M.; XUE, H.; CAI, D. Domain adaptation for semantic segmentation with maximum squares loss. In: **Proceedings of the IEEE/CVF international conference on computer vision**. [S.l.: s.n.], 2019. p. 2090–2099. Citado na página 102.
- CHEN, Y.-C.; LIN, Y.-Y.; YANG, M.-H.; HUANG, J.-B. Crdoco: Pixel-level domain transfer with cross-domain consistency. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2019. p. 1791–1800. Citado na página 102.
- COCCOMINI, D. A.; MESSINA, N.; GENNARO, C.; FALCHI, F. Combining efficientnet and vision transformers for video deepfake detection. In: SPRINGER. **International conference on image analysis and processing**. [S.l.], 2022. p. 219–229. Citado na página 82.
- CORDTS, M.; OMRAN, M.; RAMOS, S.; REHFELD, T.; ENZWEILER, M.; BENENSON, R.; FRANKE, U.; ROTH, S.; SCHIELE, B. The cityscapes dataset for semantic urban scene understanding. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 3213–3223. Citado 2 vezes nas páginas 38 e 49.
- DAI, Z.; LIU, H.; LE, Q. V.; TAN, M. Coatnet: Marrying convolution and attention for all data sizes. **Advances in neural information processing systems**, v. 34, p. 3965–3977, 2021. Citado na página 81.
- DELBRACIO, M.; LEZAMA, J.; CARBAJAL, G. Aprendizaje profundo para visión artificial. línea] Available: <https://iie. fing. edu. uy/~mdebra/DL2017/slides/c11. pdf>. Último acceso, v. 3, 2019. Citado na página 46.
- DOLHANSKY, B.; BITTON, J.; PFLAUM, B.; LU, J.; HOWES, R.; WANG, M.; FERRER, C. C. The deepfake detection challenge (dfdc) dataset. **arXiv preprint arXiv:2006.07397**, 2020. Citado na página 82.

DOSOVITSKIY, A. An image is worth 16x16 words: Transformers for image recognition at scale. **arXiv preprint arXiv:2010.11929**, 2020. Citado 2 vezes nas páginas [25](#) e [32](#).

EMARA, T.; MUNIM, H. E. A. E.; ABBAS, H. M. Liteseg: A novel lightweight convnet for semantic segmentation. In: **IEEE. 2019 digital image computing: Techniques and applications (DICTA)**. [S.l.], 2019. p. 1–7. Citado 8 vezes nas páginas [29](#), [34](#), [35](#), [37](#), [57](#), [68](#), [93](#) e [95](#).

FERREIRA, E. E.; ANDRADE, J. O.; KOMATI, K. S. Cross-database in deepfake detection based on a convolutional neural network and vision transformer. In: **SBC. Anais do XVIII Workshop de Visão Computacional**. [S.l.], 2023. p. 60–65. Citado na página [82](#).

GAO, S.; LI, Z.-Y.; YANG, M.-H.; CHENG, M.-M.; HAN, J.; TORR, P. Large-scale unsupervised semantic segmentation. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 45, n. 6, p. 7457–7476, 2022. Citado na página [88](#).

HAN, K.; WANG, Y.; CHEN, H.; CHEN, X.; GUO, J.; LIU, Z.; TANG, Y.; XIAO, A.; XU, C.; XU, Y. *et al.* A survey on vision transformer. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 45, n. 1, p. 87–110, 2022. Citado na página [87](#).

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 770–778. Citado 2 vezes nas páginas [29](#) e [34](#).

_____. _____. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 770–778. Citado 3 vezes nas páginas [41](#), [50](#) e [51](#).

HOU, Y.; QUINTANA, M.; KHOMIAKOV, M.; YAP, W.; OUYANG, J.; ITO, K.; WANG, Z.; ZHAO, T.; BILJECKI, F. Global streetscapes—a comprehensive dataset of 10 million street-level images across 688 cities for urban science and analytics. **ISPRS Journal of Photogrammetry and Remote Sensing**, Elsevier, v. 215, p. 216–238, 2024. Citado na página [50](#).

HOYER, L.; DAI, D.; GOOL, L. V. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2022. p. 9924–9935. Citado na página [83](#).

_____. _____. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2022. p. 9924–9935. Citado na página [102](#).

HUANG, J.; LU, S.; GUAN, D.; ZHANG, X. Contextual-relation consistent domain adaptation for semantic segmentation. In: **SPRINGER. European conference on computer vision**. [S.l.], 2020. p. 705–722. Citado na página [102](#).

IQBAL, J.; ALI, M. Mlsl: Multi-level self-supervised learning for domain adaptation with spatially independent and semantically consistent labeling. In: **Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision**. [S.l.: s.n.], 2020. p. 1864–1873. Citado na página [102](#).

JANAI, J.; GÜNEY, F.; BEHL, A.; GEIGER, A. *et al.* Computer vision for autonomous vehicles: Problems, datasets and state of the art. **Foundations and Trends® in Computer Graphics and Vision**, Now Publishers, Inc., v. 12, n. 1–3, p. 1–308, 2020. Citado 2 vezes nas páginas [31](#) e [85](#).

- KANG, G.; WEI, Y.; YANG, Y.; ZHUANG, Y.; HAUPTMANN, A. Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation. **Advances in neural information processing systems**, v. 33, p. 3569–3580, 2020. Citado na página 102.
- KIM, M.; BYUN, H. Learning texture invariant representation for domain adaptation of semantic segmentation. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2020. p. 12975–12984. Citado na página 102.
- KIRILLOV, A.; MINTUN, E.; RAVI, N.; MAO, H.; ROLLAND, C.; GUSTAFSON, L.; XIAO, T.; WHITEHEAD, S.; BERG, A. C.; LO, W.-Y. *et al.* Segment anything. In: **Proceedings of the IEEE/CVF International Conference on Computer Vision**. [S.l.: s.n.], 2023. p. 4015–4026. Citado 2 vezes nas páginas 25 e 81.
- LEE, C.-Y.; BATRA, T.; BAIG, M. H.; ULBRICHT, D. Sliced wasserstein discrepancy for unsupervised domain adaptation. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2019. p. 10285–10295. Citado na página 102.
- LI, F.; ZHANG, H.; SUN, P.; ZOU, X.; LIU, S.; YANG, J.; LI, C.; ZHANG, L.; GAO, J. Semantic-sam: Segment and recognize anything at any granularity. **arXiv preprint arXiv:2307.04767**, 2023. Citado na página 81.
- LI, G.; KANG, G.; LIU, W.; WEI, Y.; YANG, Y. Content-consistent matching for domain adaptive semantic segmentation. In: SPRINGER. **European conference on computer vision**. [S.l.], 2020. p. 440–456. Citado na página 102.
- LI, X.; ZHANG, L.; YOU, A.; YANG, M.; YANG, K.; TONG, Y. Global aggregation then local distribution in fully convolutional networks. **arXiv preprint arXiv:1909.07229**, 2019. Citado na página 102.
- LI, Y.; YUAN, L.; VASCONCELOS, N. Bidirectional learning for domain adaptation of semantic segmentation. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2019. p. 6936–6945. Citado na página 102.
- LIAN, Q.; LV, F.; DUAN, L.; GONG, B. Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach. In: **Proceedings of the IEEE/CVF International Conference on Computer Vision**. [S.l.: s.n.], 2019. p. 6758–6767. Citado na página 102.
- LIRA, C. Piloto automático da Tesla falha e dono precisa agir para evitar acidente. 2024. Disponível em: <<https://autoesporte.globo.com/setor-automotivo/inovacao-e-tecnologia-em-automoveis/noticia/2024/05/piloto-automatico-da-tesla-falha-e-dono-precisa-agir-para-evitar-acidente.ghtml>>. Citedo na página 86.
- LIU, W. Parsenet: Looking wider to see better. **arXiv preprint arXiv:1506.04579**, 2015. Citado na página 36.
- LIU, X.; HU, Y.; CHEN, J. Hybrid cnn-transformer model for medical image segmentation with pyramid convolution and multi-layer perceptron. **Biomedical Signal Processing and Control**, Elsevier, v. 86, p. 105331, 2023. Citado na página 82.
- LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 3431–3440. Citado na página 30.

_____. _____. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 3431–3440. Citado 2 vezes nas páginas 86 e 102.

MACIEL, L. R.; BRIGADÃO, R. F.; MONTEIRO, T. A.; GIULIANI, A. C. Mobilidade urbana e suas teorias e operacionalidades: Um ensaio teórico sobre os desafios na cidade de campinas/sp. **Revista de Administração IMED**, Faculdade Meridional-IMED, v. 7, n. 2, p. 166–182, 2017. Citado na página 86.

MAPCHANNELS. **MapChannels**. 2024. Disponível em: <<https://www.mapchannels.com/>>. Citado 3 vezes nas páginas 50, 98 e 99.

MEHTA, S.; RASTEGARI, M.; CASPI, A.; SHAPIRO, L.; HAJISHIRZI, H. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In: **Proceedings of the european conference on computer vision (ECCV)**. [S.l.: s.n.], 2018. p. 552–568. Citado na página 29.

MELAS-KYRIAHI, L.; MANRAI, A. K. Pixmatch: Unsupervised domain adaptation via pixelwise consistency training. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2021. p. 12435–12445. Citado 7 vezes nas páginas 30, 41, 42, 43, 57, 68 e 102.

MIYATO, T.; MAEDA, S.-i.; KOYAMA, M.; ISHII, S. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 41, n. 8, p. 1979–1993, 2018. Citado na página 46.

MOHAMMED, A. A.; AMBAK, K.; MOSA, A. M.; SYAMSUNUR, D. A review of traffic accidents and related practices worldwide. **The Open Transportation Journal**, v. 13, n. 1, 2019. Citado na página 86.

MURUGAPPAN, M.; BOURISLY, A. K.; PRAKASH, N.; SUMITHRA, M.; ACHARYA, U. R. Automated semantic lung segmentation in chest ct images using deep neural network. **Neural Computing and Applications**, Springer, v. 35, n. 21, p. 15343–15364, 2023. Citado na página 35.

NEKRASOV, V.; SHEN, C.; REID, I. Light-weight refinenet for real-time semantic segmentation. **arXiv preprint arXiv:1810.03272**, 2018. Citado na página 34.

NGO, B. H.; DO-TRAN, N.-T.; NGUYEN, T.-N.; JEON, H.-G.; CHOI, T. J. Learning cnn on vit: A hybrid model to explicitly class-specific boundaries for domain adaptation. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2024. p. 28545–28554. Citado na página 83.

OQUAB, M.; DARDET, T.; MOUTAKANNI, T.; VO, H.; SZAFRANIEC, M.; KHALIDOV, V.; FERNANDEZ, P.; HAZIZA, D.; MASSA, F.; EL-NOUBY, A. *et al.* Dinov2: Learning robust visual features without supervision. **arXiv preprint arXiv:2304.07193**, 2023. Citado 2 vezes nas páginas 25 e 81.

OUALI, Y.; HUDELOT, C.; TAMI, M. Semi-supervised semantic segmentation with cross-consistency training. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2020. p. 12674–12684. Citado 2 vezes nas páginas 30 e 57.

PAN, F.; SHIN, I.; RAMEAU, F.; LEE, S.; KWEON, I. S. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2020. p. 3764–3773. Citado na página 102.

PELÁEZ-VEGAS, A.; MESEJO, P.; LUENGO, J. A survey on semi-supervised semantic segmentation. **arXiv preprint arXiv:2302.09899**, 2023. Citado 5 vezes nas páginas 26, 39, 40, 44 e 88.

PROSDOSKIMIS, N. A. P. T. R. **MOBILIDADE AUTÔNOMA PARA PCDs**. 2020. Acesso em: 15 out. 2024. Disponível em: <<https://wwwaea.org.br/premio/downloads/2021/trabalhos/TB00045.pdf>>. Citado na página 86.

RATEKE, T.; WANGENHEIM, A. V. Road surface detection and differentiation considering surface damages. **Autonomous Robots**, Springer, v. 45, n. 2, p. 299–312, 2021. Citado 2 vezes nas páginas 32 e 88.

RICHTER, S. R.; VINEET, V.; ROTH, S.; KOLTUN, V. Playing for data: Ground truth from computer games. In: SPRINGER. **Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14**. [S.l.], 2016. p. 102–118. Citado na página 49.

ROMERA, E.; ALVAREZ, J. M.; BERGASA, L. M.; ARROYO, R. Efficient convnet for real-time semantic segmentation. In: IEEE. **2017 IEEE Intelligent Vehicles Symposium (IV)**. [S.l.], 2017. p. 1789–1794. Citado na página 29.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. **Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III 18**. [S.l.], 2015. p. 234–241. Citado na página 30.

SANDLER, M.; HOWARD, A.; ZHU, M.; ZHMOGINOV, A.; CHEN, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2018. p. 4510–4520. Citado 2 vezes nas páginas 29 e 36.

SANKAR, A. **A Primer on Atrous Convolutions and Depth-wise Separable Convolutions**. 2024. <<https://towardsdatascience.com/a-primer-on-atrous-convolutions-and-depth-wise-separable-convolutions-443b106919f5>>. Acesso em: 15 out. 2024. Citado 2 vezes nas páginas 34 e 93.

SANTOS, V. A. dos; LYRA, R.; PEREIRA, T. F. A study on semantic segmentation for autonomous vehicles. **Anais do Computer on the Beach**, v. 11, p. 059–061, 2020. Citado na página 86.

SATYA. **Real-time Semantic Segmentation with Deep Learning**. 2024. Disponível em: <https://medium.com/@satya15july_11937/real-time-semantic-segmentation-with-deep-learning-3e29e5d3c193>. Citado 3 vezes nas páginas 44, 45 e 87.

SHI, W.; CABALLERO, J.; HUSZÁR, F.; TOTZ, J.;AITKEN, A. P.; BISHOP, R.; RUECKERT, D.; WANG, Z. Real-time single image and video super-resolution using an efficient sub-pixel

convolutional neural network. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 1874–1883. Citado na página 51.

SHINZATO, P. Y.; SANTOS, T. C. dos; ROSERO, L. A.; RIDEL, D. A.; MASSERA, C. M.; ALENCAR, F.; BATISTA, M. P.; HATA, A. Y.; OSÓRIO, F. S.; WOLF, D. F. Carina dataset: An emerging-country urban scenario benchmark for road detection systems. In: **2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)**. [S.l.: s.n.], 2016. Citado 2 vezes nas páginas 32 e 88.

SIAM, M.; GAMAL, M.; ABDEL-RAZEK, M.; YOGAMANI, S.; JAGERSAND, M. Rtseg: Real-time semantic segmentation comparative study. In: **IEEE. 2018 25th IEEE International Conference on Image Processing (ICIP)**. [S.l.], 2018. p. 1603–1607. Citado 2 vezes nas páginas 29 e 86.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014. Citado na página 29.

TOLDO, M.; MARACANI, A.; MICHELI, U.; ZANUTTIGH, P. Unsupervised domain adaptation in semantic segmentation: a review. **Technologies**, MDPI, v. 8, n. 2, p. 35, 2020. Citado 4 vezes nas páginas 26, 27, 39 e 88.

TOMPSON, J.; GOROSHIN, R.; JAIN, A.; LECUN, Y.; BREGLER, C. Efficient object localization using convolutional networks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 648–656. Citado na página 46.

TRANHEDEN, W.; OLSSON, V.; PINTO, J.; SVENSSON, L. Dacs: Domain adaptation via cross-domain mixed sampling. In: **Proceedings of the IEEE/CVF winter conference on applications of computer vision**. [S.l.: s.n.], 2021. p. 1379–1389. Citado na página 102.

TREVISAN, V. **Como Funcionam as Redes Neurais Convolucionais (CNNs)**. 2024. Disponível em: <<https://medium.com/data-hackers/como-funcionam-as-redes-neurais-convolucionais-cnns-71978185c1>>. Citado 7 vezes nas páginas 89, 90, 91, 92, 93, 94 e 95.

TSAI, Y.-H.; HUNG, W.-C.; SCHULTER, S.; SOHN, K.; YANG, M.-H.; CHANDRAKER, M. Learning to adapt structured output space for semantic segmentation. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2018. p. 7472–7481. Citado na página 102.

TSAI, Y.-H.; SOHN, K.; SCHULTER, S.; CHANDRAKER, M. Domain adaptation for structured output via discriminative patch representations. In: **Proceedings of the IEEE/CVF international conference on computer vision**. [S.l.: s.n.], 2019. p. 1456–1465. Citado na página 102.

VideoLAN Organization. **VLC Media Player**. 2024. Accessed: 18 October 2024. Disponível em: <<https://www.videolan.org/vlc/>>. Citado na página 50.

VU, T.-H.; JAIN, H.; BUCHER, M.; CORD, M.; PÉREZ, P. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2019. p. 2517–2526. Citado na página 102.

WANG, H.; SHEN, T.; ZHANG, W.; DUAN, L.-Y.; MEI, T. Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. In: SPRINGER. **European conference on computer vision**. [S.l.], 2020. p. 642–659. Citado na página 102.

- WANG, Q.; DAI, D.; HOYER, L.; GOOL, L. V.; FINK, O. Domain adaptive semantic segmentation with self-supervised depth estimation. In: **Proceedings of the IEEE/CVF International Conference on Computer Vision**. [S.l.: s.n.], 2021. p. 8515–8525. Citado na página 102.
- WEISS, K.; KHOSHGOFTAAR, T. M.; WANG, D. A survey of transfer learning. **Journal of Big data**, Springer, v. 3, p. 1–40, 2016. Citado na página 88.
- WU, H.; XIAO, B.; CODELLA, N.; LIU, M.; DAI, X.; YUAN, L.; ZHANG, L. Cvt: Introducing convolutions to vision transformers. In: **Proceedings of the IEEE/CVF international conference on computer vision**. [S.l.: s.n.], 2021. p. 22–31. Citado na página 81.
- XIE, E.; WANG, W.; YU, Z.; ANANDKUMAR, A.; ALVAREZ, J. M.; LUO, P. Segformer: Simple and efficient design for semantic segmentation with transformers. **Advances in neural information processing systems**, v. 34, p. 12077–12090, 2021. Citado na página 87.
- YANG, Y.; SOATTO, S. Fda: Fourier domain adaptation for semantic segmentation. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2020. p. 4085–4095. Citado na página 102.
- YU, F. Multi-scale context aggregation by dilated convolutions. **arXiv preprint arXiv:1511.07122**, 2015. Citado 2 vezes nas páginas 30 e 93.
- ZHANG, P.; ZHANG, B.; ZHANG, T.; CHEN, D.; WANG, Y.; WEN, F. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2021. p. 12414–12424. Citado na página 102.
- ZHANG, Q.; ZHANG, J.; LIU, W.; TAO, D. Category anchor-guided unsupervised domain adaptation for semantic segmentation. **Advances in neural information processing systems**, v. 32, 2019. Citado na página 102.
- ZHANG, X.; ZHOU, X.; LIN, M.; SUN, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2018. p. 6848–6856. Citado na página 29.
- ZHAO, H.; SHI, J.; QI, X.; WANG, X.; JIA, J. Pyramid scene parsing network. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 2881–2890. Citado 2 vezes nas páginas 29 e 50.
- _____. _____. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 2881–2890. Citado na página 102.
- ZHU, J.; LUO, Y.; ZHENG, X.; WANG, H.; WANG, L. A good student is cooperative and reliable: Cnn-transformer collaborative learning for semantic segmentation. In: **Proceedings of the IEEE/CVF International Conference on Computer Vision**. [S.l.: s.n.], 2023. p. 11720–11730. Citado na página 82.
- ZHUANG, Y.; YANG, F.; TAO, L.; MA, C.; ZHANG, Z.; LI, Y.; JIA, H.; XIE, X.; GAO, W. Dense relation network: Learning consistent and context-aware representation for semantic image segmentation. In: **IEEE. 2018 25th IEEE international conference on image processing (ICIP)**. [S.I.], 2018. p. 3698–3702. Citado na página 102.

ZOU, Y.; YU, Z.; KUMAR, B.; WANG, J. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: **Proceedings of the European conference on computer vision (ECCV)**. [S.l.: s.n.], 2018. p. 289–305. Citado na página [102](#).

ZOU, Y.; YU, Z.; LIU, X.; KUMAR, B.; WANG, J. Confidence regularized self-training. In: **Proceedings of the IEEE/CVF international conference on computer vision**. [S.l.: s.n.], 2019. p. 5982–5991. Citado na página [102](#).

APÊNDICE A

PESQUISA COMPLEMENTAR SOBRE MODELOS VIT

A.1 Modelos baseados em Transformadores de Visão

Como início deste trabalho, foi estudado o artigo “*Segment Anything*” ([KIRILLOV et al., 2023](#)) apresentando o projeto ***Segment Anything Model*** (SAM), que introduz um novo modelo e conjunto de dados para segmentação de imagens genérica. Ele é projetado para ser “*promptable*”, permitindo transferência para novas distribuições de imagens e tarefas sem necessidade de treinamento adicional, o desempenho *zero-shot* do modelo é impressionante, muitas vezes superando resultados totalmente supervisionados anteriores, o modelo e o conjunto de dados estão disponíveis para fomentar a pesquisa em visão computacional. Já no projeto “*DINOv2: Learning Robust Visual Features without Supervision*” ([OQUAB et al., 2023](#)) a META/Facebook apresenta um método de aprendizado auto-supervisionado para criar características visuais robustas sem a necessidade de rótulos ou anotações. Utilizando um grande conjunto de dados curados de 142 milhões de imagens, o modelo DINOv2 treina uma arquitetura de Transformer de Visão (ViT) com 1 bilhão de parâmetros e a destila em modelos menores que superam os melhores recursos visuais disponíveis em *benchmarks*. Ressalta-se a capacidade do modelo DinoV2 em acelerar e estabilizar o treinamento em larga escala, resultando em características visuais que são robustas e performam bem em diversas tarefas de visão computacional sem necessidade de ajuste fino, com uso do DinoV2 como backbone e acoplados a pretrained heads para estimação de profundidade, segmentação semântica e classificação de imagens.

O SAM apesar de seu alto desempenho em *zero-shot* não apresenta a capacidade de semântica de forma nativa, o que limita seu uso fora do prompt, com isso o artigo “*Semantic-SAM: Segment and Recognize Anything at Any Granularity*” ([LI et al., 2023](#)) apresenta o modelo Semantic-SAM, um modelo universal de segmentação de imagens que permite segmentar e reconhecer qualquer coisa em qualquer granularidade desejada, porém sem a possibilidade de estabelecer classes de interesse.

Os transformadores de visão também ganharam espaço com modelos híbridos com CNNs tradicionais, para fazer uso da simplicidade de CNNs e compreensão contextual viabilizado pelos ViTs, como visto nos artigos “*CoAtNet: Marrying Convolution and Attention for All Data Sizes*” ([DAI et al., 2021](#)) e “*CvT: Introducing Convolutions to Vision Transformers*” ([WU et al., 2021](#)),

que exploram a integração de convoluções e Transformadores de Visão (ViTs) para melhorar o desempenho em tarefas de visão computacional. O CoAtNet combina CNNs e ViTs de forma eficiente, empilhando camadas de convolução e atenção para alcançar uma precisão de 90.88% no ImageNet. Já o CvT introduz convoluções nos ViTs, criando uma hierarquia de Transformadores com tokens convolucionais e blocos de Transformador convolucional, resultando em melhor desempenho e eficiência em comparação com outros ViTs e redes residuais no ImageNet-1k. Ambos os estudos demonstram que a combinação de CNNs e ViTs pode levar a avanços significativos na área.

Outro exemplo de modelos híbridos entre CNNs e ViTs são utilizados na detecção de deepfakes, classificando conteúdos como falsos ou não. O artigo “*Combining EfficientNet and Vision Transformers for Video Deepfake Detection*” ([COCCOMINI et al., 2022](#)) propõe uma abordagem que combina EfficientNet como extrator de características e ViTs, alcançando uma AUC de 0.951 e uma pontuação F1 de 88.0% no DeepFake Detection Challenge (DFDC) ([DOLHANSKY et al., 2020](#)), sem utilizar métodos de destilação ou ensemble. Já o artigo “*Cross-Database in Deepfake Detection Based on a Convolutional Neural Network and Vision Transformer*” ([FERREIRA; ANDRADE; KOMATI, 2023](#)) explora a detecção de deepfakes com uma combinação de CNNs e ViTs, focando na generalização dos modelos para diferentes bancos de dados e destacando desafios de overfitting. Ambos os estudos mostram que a integração de CNNs e ViTs pode melhorar significativamente a detecção de deepfakes, embora ainda existam desafios a serem superados.

Outra exemplo de área de aplicação de redes híbridas CNNs e ViTs é para imagens médicas, como visto nos artigos “*Scopeformer: n-CNN-ViT Hybrid Model for Intracranial Hemorrhage Classification*” ([BARHOUMI; RASOOL, 2021](#)) e “*Hybrid CNN-Transformer Model for Medical Image Segmentation with Pyramid Convolution and Multi-Layer Perceptron*” ([LIU; HU; CHEN, 2023](#)) propõem modelos híbridos para melhorar a análise de imagens médicas. O primeiro artigo foca na classificação de hemorragias intracranianas em tomografias, utilizando CNNs Xception para extrair características e ViTs para analisar essas características em múltiplos níveis, alcançando uma precisão de 98.04%. Já o segundo artigo propõe um modelo de segmentação de imagens médicas em formato de U, combinando convoluções de *kernel* grande com ViT e perceptrons multicamadas para capturar informações detalhadas em várias escalas, melhorando a precisão da segmentação. Ambos os modelos se destacam pela integração das vantagens das CNNs e dos ViTs, visando maior precisão e eficiência na análise de imagens médicas.

Além de modelos híbridos CNNs e ViTs, também podem ser explorados para colaboração entre redes para aprimorar tarefas de aprendizado de máquina, sobre estratégia de redes colaborativas, exemplos desta abordagem podem ser visto nos artigos “*A Good Student is Cooperative and Reliable: CNN-Transformer Collaborative Learning for Semantic Segmentation*” ([ZHU et al., 2023](#)) e “*Learning CNN on ViT: A Hybrid Model to Explicitly Class-specific Boundaries for*

Domain Adaptation” (NGO *et al.*, 2024). O primeiro artigo propõe um framework colaborativo para segmentação semântica, utilizando técnicas de destilação para melhorar a consistência e a transferência de conhecimento entre as redes. O segundo artigo apresenta uma arquitetura híbrida para adaptação de domínio, combinando as propriedades dos ViTs para identificar fronteiras de decisão específicas e as CNNs para agrupar características, melhorando a qualidade dos pseudo rótulos e reduzindo discrepâncias de conhecimento.

Por fim, o artigo “*DAFormer: Improving Network Architectures and Training Strategies for Domain-Adaptive Semantic Segmentation*” (HOYER; DAI; GOOL, 2022a) propõe uma nova abordagem para adaptação de domínio em segmentação semântica, resolvendo problemas de desempenho ao transferir modelos entre domínios distintos (como de ambientes sintéticos para reais). A arquitetura DAFormer combina camadas de atenção com convoluções, utilizando uma estrutura híbrida eficiente para capturar tanto informações locais quanto contextuais amplas. Além disso, o artigo introduz uma nova estratégia de treinamento, baseada na regularização de consistência e no uso de pseudo-rótulos filtrados, o que melhora a robustez e a estabilidade do modelo. Experimentos demonstram que o DAFormer supera outras abordagens, alcançando resultados de ponta em *benchmarks* desafiadores como GTA5-to-Cityscapes e SYNTHIA-to-Cityscapes, provando sua eficácia na adaptação de domínio.

APÊNDICE B

ESTUDOS COMPLEMENTARES

B.1 Segmentação Semântica para Direção Autônoma

Segmentação Semântica é uma tarefa de visão computacional por Redes Neurais Convolucionais, ou *Convolutional Neural Network* (CNN), definido como o processo de classificação de cada pixel presente na imagem em um rótulo de classe, sendo a próxima granularidade de classificação em comparação a tarefa de detecção de objetos em caixas delimitadoras mínima (i.e. *bounding box*). Esta subárea da visão computacional é empregado em diversos problemas da área de imagens médicas, robótica, direção autônoma, imagens aéreas, entre outros domínios de aplicação. Para direção autônoma, a compreensão semântica das cenas urbanas é um dos principais componentes para a tomada de decisão dentre os vários níveis de direção autônoma definido pela *Society of Automotive Engineers* – SAE como exposto em ([JANAI et al., 2020](#)), sendo necessário reconhecer, principalmente, a área navegável, não navegável e obstáculos.

Os níveis de direção autônoma são:

- Nível 0 (Sem Automação): O motorista controla totalmente o veículo. Não há assistência automatizada.
- Nível 1 (Assistência ao Motorista): Sistemas como controle de cruzeiro adaptativo ajudam, mas o motorista ainda é responsável pela direção e monitoramento do ambiente.
- Nível 2 (Automação Parcial): O veículo pode controlar a direção e a aceleração/desaceleração, mas o motorista deve estar atento e pronto para intervir.
- Nível 3 (Automação Condicional): O veículo pode gerenciar a maioria das tarefas de direção em certas condições, mas o motorista deve estar pronto para assumir o controle quando solicitado.
- Nível 4 (Alta Automação): O veículo pode operar de forma autônoma em muitas situações, mas pode haver restrições geográficas ou ambientais.
- Nível 5 (Automação Completa): O veículo é totalmente autônomo e não requer intervenção humana em nenhuma circunstância.

A direção autônoma representa um avanço significativo na tecnologia de transporte, com potencial para transformar a mobilidade urbana e reduzir acidentes de trânsito. A automação na condução de veículos tem o potência de estabelecer parcialmente ou totalmente salva guarda na tomada de decisão durante a condução, podendo reduzir o fator humano que contribui na estimativa em mais de 90% da ocorrência em acidentes, em que grande parte desta contribuição advém do comportamento do condutor, como exposto em (BOTTESEINI, 2010), (MOHAMMED *et al.*, 2019) e (SANTOS; LYRA; PEREIRA, 2020). Além disso, a eficiência dos veículos autônomos em otimizar rotas e reduzir congestionamentos, pode tornar os centros urbanos mais eficientes na locomoção dos cidadãos e mercadorias, e consequentemente, pode levar a uma diminuição no consumo de combustível e nas emissões de poluentes, como teorizado no ensaio (MACIEL *et al.*, 2017). A acessibilidade também é um tema impactado, pois essa tecnologia pode beneficiar pessoas com mobilidade reduzida, proporcionando-lhes maior independência (PROSDOSKIMIS, 2020). A relevância econômica é notável, com a criação de novos empregos e oportunidades de negócios, especialmente nos setores de logística e entrega de mercadorias, visto que a automação de veículos impacta na manufatura, frota, e disponibilidade de serviço no que diz respeito ao delivery de objetos. Assim, a direção autônoma não apenas promete melhorar a segurança e a eficiência do transporte, mas também tem o potencial de gerar impactos positivos na economia e na sociedade como um todo.

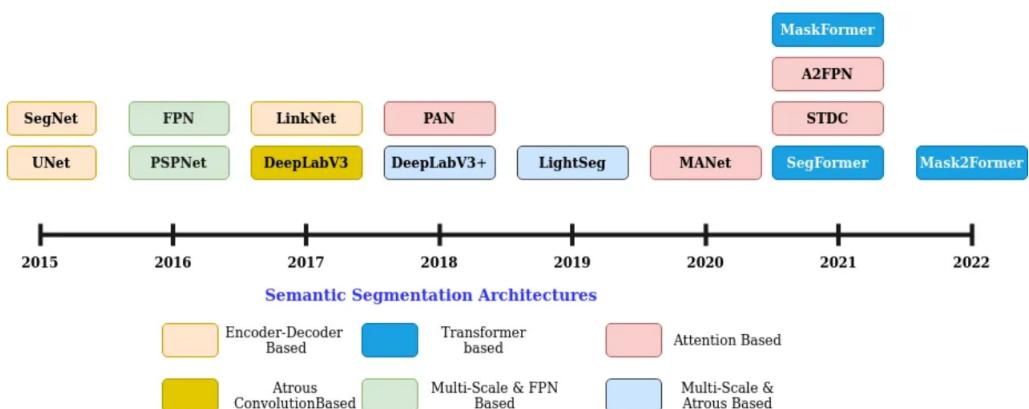
Na aplicação de segmentação semântica neste domínio de aplicação, direção autônoma, encontramos desafios acerca da precisão, eficiência, robustez e versatilidade dos modelos de aprendizado de máquina empregados, isto se deve, principalmente, ao caráter crítico da tomada de decisão na condução de veículos. Os modelos de segmentação semântica necessitam alcançar alta precisão para cumprir corretamente com condução do veículo, e eficiente o suficiente para toda de decisão em tempo real, ou seja, possuir baixa latência para iniciar e finalizar a predição, com tempo de predição na ordem de milissegundos, logo, a complexidade computacional não pode ser negligenciada em prol do mais alto patamar de acurácia dos modelos. Sobre cenas urbanas temos o fator de ruídos advindos de condições climáticas, iluminação e condições da via, o que requer do modelo robustez a variações mantendo a performance constante, e nem sempre a adaptação de domínio intrínseco ao modelo é capaz de lidar com diferentes cenários distintos do conjunto dados de treino. Mesmo com avanços de modelos para segmentação semântica em precisão e eficiência no estado da arte, ainda é necessário atenção sobre situações adversas, para reduzir casos como visto na reportagem (LIRA, 2024), e aumentar a disponibilidade de modelos em tempo real para tornar a tecnologia de direção autônoma praticável, como exposto no trabalho (SIAM *et al.*, 2018).

Em 2014, a Fully Convolutional Network (FCN), introduzida em “Fully Convolutional Networks for Semantic Segmentation” (LONG; SHELHAMER; DARRELL, 2015b). As redes FCNs foram pioneiras ao adaptar CNNs para tarefas de segmentação semântica, permitindo a classificação de cada pixel em uma imagem. A arquitetura de primeira linha para segmentação semântica foi do tipo codificador-decodificador. No codificador a entrada é reduzida gradualmente

em uma representação de espaço latente comprimido formando mapas de características através da aplicação de filtros em convoluções consecutivas com amostragem descendente (i.e *pooling* como *down-sampling*), na sequência o decodificador reconstrói a dimensionalidade da imagem realizando operações de amostragem ascendente (i.e *up-sampling*), sendo populares a redes UNet e SegNet como arquiteturas do tipo codificador-decodificador. A arquitetura de segunda linha baseadas em convoluções dilatadas (i.e *Atrous*), aumentando o campo receptivo da camada anterior sem diminuição das dimensões espaciais da camada, o que reduz o custo computacional para *down-sampling*, redes populares em empregar essa convolução são as versões do Deeplab (CHEN *et al.*, 2017) a evolução desta arquitetura proporcionou a arquitetura baseado em *Atrous Spatial Pyramid Pooling* (ASPP) (CHEN *et al.*, 2017) ,no qual aplica-se uma única convolução *atrous*, ASPP aplica múltiplas convoluções dilatadas em paralelo, cada uma com uma taxa de amostragem dilatada diferente, assim, constitui uma “pirâmide” de diferentes campos de visão, permitindo que a rede capture tanto detalhes finos quanto contextos mais amplos da imagem.

Por fim, o estado da arte a segmentação semântica está cada vez mais complexo com novas arquiteturas emergentes, mas recentemente utilizando Transformadores de Visão originalmente aplicado a problemas de Processamento de Linguagem Natural, agora compartilha espaço com as técnicas tradicionais de segmentação semântica envolvendo Redes Neurais Convulsionais (XIE *et al.*, 2021), mais eficientes pelo módulos baseadas em atenção, que melhoram a capacidade do modelo de capturar relações contextuais entre diferentes partes da imagem, assim não se limitando a pares locais pré-definidos para determinar o campo receptivo a características, como aplicado por CNN (HAN *et al.*, 2022). Um resumo dos principais modelos e suas abordagens ao longo do tempo pode ser visualizado na figura 21.

Figura 21 – Arquiteturas ao longo do tempo.

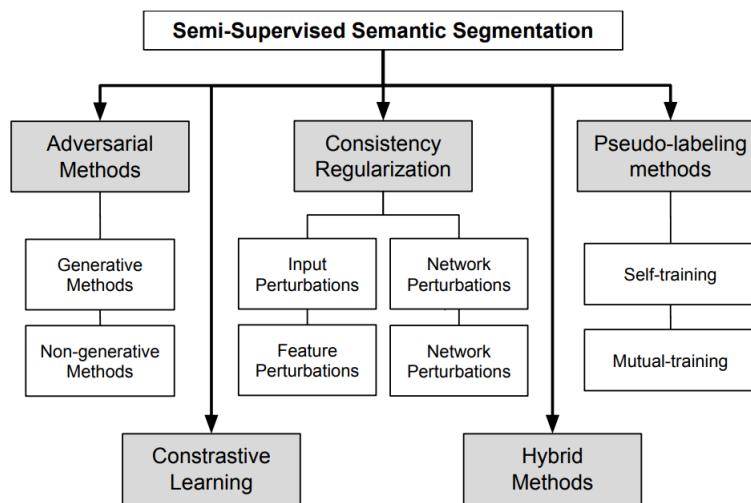


Fonte: Satya (2024).

Não apenas por arquitetura é marcado os avanços em segmentação semântica, também é explorado técnicas para melhorar a robustez a variação do domínio, através de métodos de adaptação de domínio, e estratégias de treinamento. Também utilização de técnicas de

aprendizado por transferência de modelos fundamentais para adaptar onde modelos pré-treinados em grandes conjuntos de dados são adaptados para tarefas específicas de segmentação (WEISS; KHOSHGOFTAAR; WANG, 2016). Estratégias de treinamento também ganham relevância para explorar tarefas variantes do treinamento supervisionado, como semi-supervisionado e não supervisionado, e suas aplicações a tarefa de adaptação de domínio, pelas estratégias de métodos adversárias (generativo, e não generativo), aprendizado contrastivo, regularização por consistência (perturbações na entrada, características ou redes), pseudo rótulos (auto treinamento, co-treinamento) como organizado na figura 22, e acurado nos trabalhos (PELÁEZ-VEGAS; MESEJO; LUENGO, 2023) e (TOLDO *et al.*, 2020).

Figura 22 – Taxonomia das estratégias de treinamento semi-supervisionado e não supervisionado.



Fonte: Peláez-Vegas, Mesejo e Luengo (2023).

Sobre o contexto de cenas urbanas brasileiro é carente de conjunto de imagens representativas e rotuladas o suficiente para representar toda a variedade das vias do território brasileiro. De forma geral a obtenção de rótulos é cara e demorada, logo estratégias de treinamento que não exijam rótulos podem viabilizar modelos de aprendizado de máquina capazes de explorar conjuntos de dados maiores e disponíveis sem rótulos, além de viabilizar segmentação semântica em larga escala (GAO *et al.*, 2022). Existem alguns conjuntos de imagens para o contexto brasileiro encontrados ao decorrer da pesquisa bibliográfica desta monografia, sendo o CaRINA Dataset (SHINZATO *et al.*, 2016) que contém conjuntos de dados de cenários urbanos brasileiros selecionados e benchmarks de detecção de estradas consistindo em dados anotados de RADAR, LIDAR e câmera, e Road Traversing Knowledge (RTK) Dataset (RATEKE; WANGENHEIM, 2021) contém imagens capturadas por uma câmera de baixo custo (HP Webcam HD-4110) de estradas com diferentes tipos de superfície: variações de asfalto, de outros tipos de pavimento e inclusive estradas não pavimentadas, contém também situações com danos na estrada, como por exemplo: buracos.

Com tudo, a tarefa de segmentação semântica está aquecida como solução para múltiplos

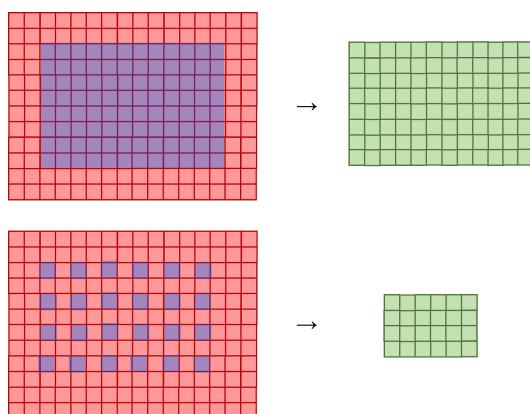
domínios de aplicação, imagens médicas, robótica, direção autônoma, entre outros, e o estado da arte está repleto de modelos das mais variadas arquiteturas, treinados sobre diversas estratégias, mas quando observado o problema em particular, “segmentação semântica de áreas navegáveis e não navegáveis para cenas urbanas brasileiras em tempo real”, ainda é um tema que carece de estudos para viabilizar o treinamento de modelos admitindo a variedade de vias no território brasileiro (majoritariamente sem rótulos), e praticável sua execução em hardware de baixo custo em tempo real.

B.2 Rede Neural Convolucional

Modelos de aprendizado profundo, especificamente, redes neurais artificiais são implementadas utilizando a representação bioinspirada do funcionamento de sinapses de neurônios de um cérebro humano, com o objetivo de reconhecer padrões e extrair características para diversos propósitos, devido a sua composição de camadas, funções de ativação e filtros, possibilita soluções complexas e não lineares a diversos domínios de aplicação. As Redes Neurais Convolucionais, ou *Convolutional Neural Network* (CNN), é uma classe de redes neurais amplamente utilizada para o processamento de dados em estrutura matricial, como imagens. Para a segmentação semântica, as CNNs são usadas para classificar cada pixel de uma imagem em uma classe específica, permitindo a identificação e a demarcação de diferentes objetos ou regiões contidas na imagem.

São realizadas várias operações nas CNNs para a segmentação semântica. A operação de convolução aplica filtros (i.e., *kernels*) à imagem de entrada, no qual esses filtros “deslizam” pela imagem por um valor de salto (i.e., *stride*), como exposto nas figuras 23 e 24, produzindo mapas de características que destacam diferentes aspectos da imagem, permitindo a extração características importantes, como bordas, texturas e demais padrões, como ilustrado na figura 25.

Figura 23 – Efeito do stride na convolução.

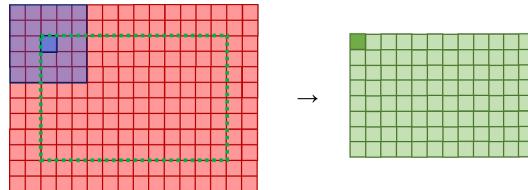


Fonte: [Trevisan \(2024\)](#).

Na figura 23, a parte superior da imagem mostra um caso em que o *stride* é 1, ou seja, o *kernel* anda um pixel na horizontal para cada iteração e no final da linha anda um pixel na

vertical. A metade inferior ilustra um caso com valor de *stride* igual a 2, então o *kernel* itera dois pixels na horizontal e quando a linha termina, anda dois pixels na vertical. Em ambas as imagens os pixels azuis correspondem ao centro do *kernel* de tamanho 5x5 em cada iteração, e pode-se verificar que a convolução com maior *stride* gera uma saída de menor dimensão.

Figura 24 – Convolução bidimensional em imagens.



Fonte: [Trevisan \(2024\)](#).

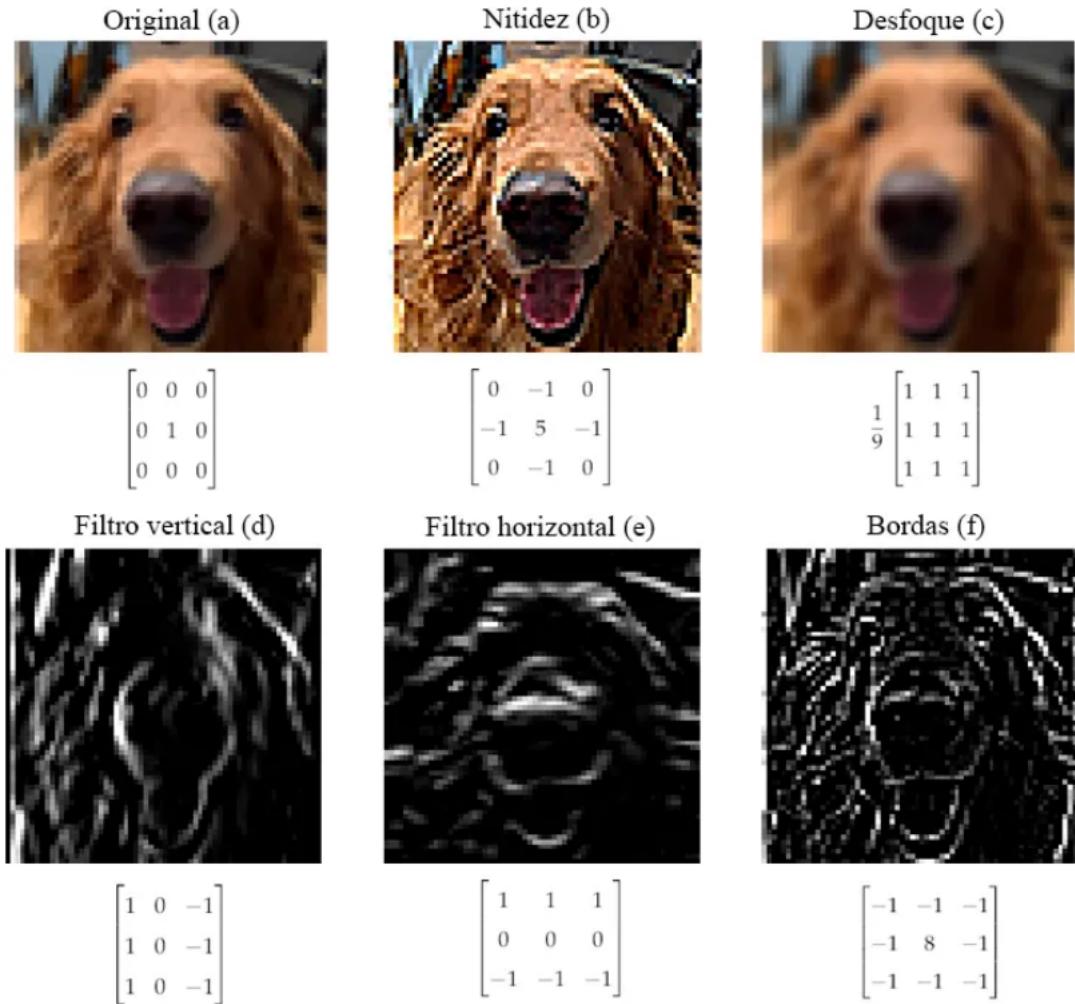
Na Figura 24, o *kernel* (azul) é convolucionado sobre a imagem (vermelha), ou seja, operação é feita nos pixels da imagem que correspondem à posição em que o *kernel* está. O resultado da operação de convolução determina o valor do pixel central dessa região, e o conjunto dos resultados da aplicação do *kernel* em cada pixel forma a imagem convolucionada (verde), que é menor do que a imagem original.

Na figura 25, é apresentada a imagem original em (a), juntamente com o *kernel* identidade, que não faz modificações. Em (b) a imagem passou por um *kernel* que aumenta a nitidez, reforçando o pixel central e aumentando a diferença dele em relação à sua vizinhança. Em (c) foi usado um *kernel* de desfoque, que substitui um pixel pela média dele com seus vizinhos. A imagem em (d) é o resultado de aplicar na imagem em escala de cinza um *kernel* que reforça suas linhas verticais, enquanto (e) reforça suas linhas horizontais. O *kernel* em (f) reforça os contornos (bordas) da imagem.

Principalmente para arquiteturas do tipo codificador-decodificador, a operação de amostragem descendente (i.e., *pooling*), reduz a dimensionalidade dos mapas de características, mantendo as informações mais importantes, como mostrado na figura 26. As operações de *pooling* mais comuns são o *max pooling* (entre os pixels de cada iteração é escolhido o de maior valor) e o *average pooling* (entre os pixels de cada iteração é obtido a média), que implica na redução da complexidade computacional e a controlar o sobre ajuste do modelo (i.e. *overfitting*), seus efeitos práticos na imagem estão expostos na figura 27. Entre as camadas de convolução e amostragem é aplicado camadas de ativação que introduzem não-linearidade no modelo, permitindo que aprenda representações mais complexas do domínio do problema.

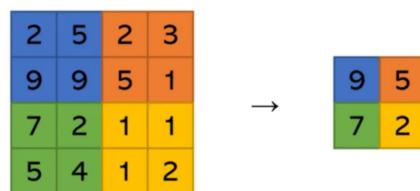
Na figura 26, cada grupo de 2x2 pixels é representado pelo valor de maior intensidade do grupo. Para o *AvgPooling* a cada grupo de 2x2 pixels é representado pela média. Já na figura 27, a imagem de bordas detectadas por um *kernel* adequado em (a). Em (b) e em (c) a aplicação do *MaxPool* e do *AvgPool* respectivamente, ambos com tamanho 2. As imagens resultantes têm metade do tamanho da original.

Figura 25 – Aplicações de diferentes kernels em uma imagem.



Fonte: Trevisan (2024).

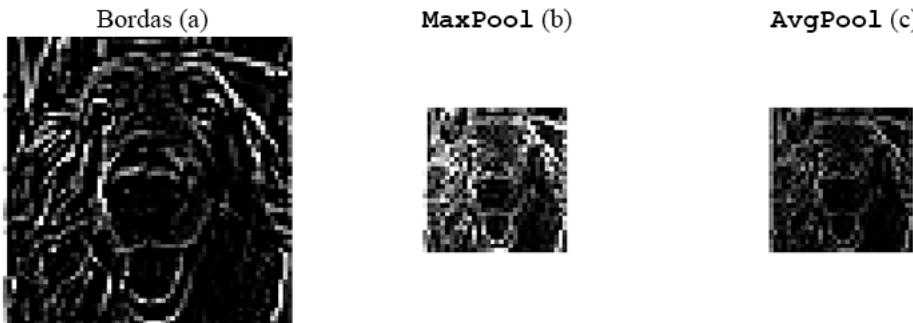
Figura 26 – Funcionamento do MaxPooling.



Fonte: Trevisan (2024).

Para a segmentação semântica do tipo codificador-decodificador, é necessário recuperar a resolução original da imagem no decodificador, logo as camadas convolucionais transpostas, aumentam a resolução dos mapas de características, permitindo a reconstrução detalhada da imagem segmentada, como ilustrado na figura 28. Em algumas arquiteturas, como a UNet, também utilizam conexões de salto (i.e., *skip connections*) para combinar informações de diferentes níveis de resolução entre o codificador e decodificador, o que preserva detalhes finos e melhora a precisão da segmentação.

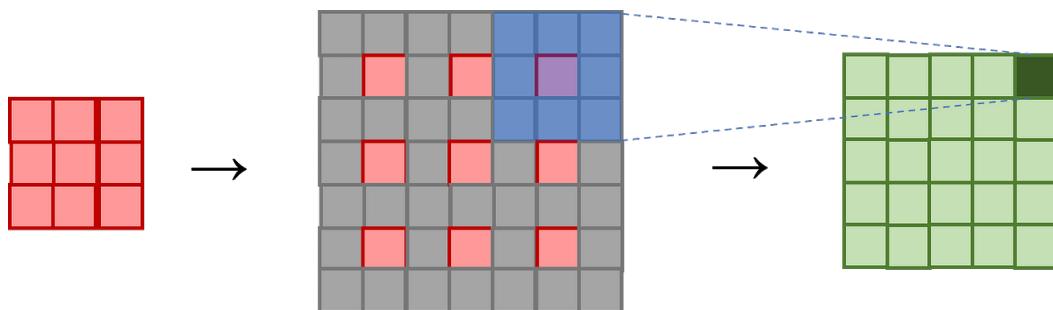
Figura 27 – Aplicação de pooling em uma imagem de bordas.



Fonte: [Trevisan \(2024\)](#).

Por fim, uma camada *Softmax* é frequentemente usada para converter as ativações em probabilidades de classe para cada pixel, permitindo a classificação precisa de cada pixel na imagem.

Figura 28 – Convolução transposta.



Fonte: [Trevisan \(2024\)](#).

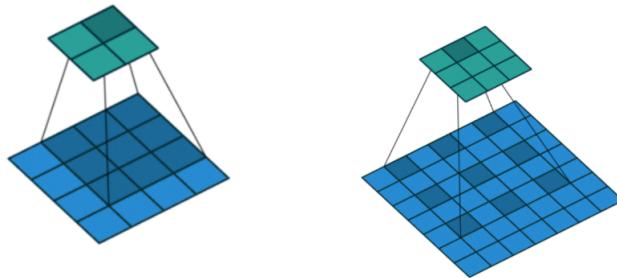
Na figura 28, uma convolução de *kernel* 3x3 com *stride* 1 em uma imagem 5x5 geraria uma imagem de dimensões 3x3. A convolução transposta aplicada na imagem 3x3 (vermelha) para se obter uma imagem 5x5 (verde) é feita em duas etapas. Primeiramente (centro) se insere pixels de valor 0, representados pela cor cinza, entre os pixels da imagem original, juntamente com pixels de borda. Em seguida se faz uma convolução nessa imagem alterada também usando um *kernel* 3x3 e *stride* 1.

B.3 Deeper Atrous Spatial Pyramid Pooling (DASPP)

Nas arquiteturas com uso de convolução e agrupamento de *strides* tradicionais é notório a diminuição do tamanho do campo receptivo em uma perda de informação espacial. Para superar essa restrição a convolução dilatada é empregada para aumentar o campo receptivo sem qualquer redução na resolução do mapa de características nem aumento nos parâmetros treináveis, possibilitando que a rede aprenda características de contexto global em toda a imagem para refinar previsões de resolução máxima, como exposto em ([EMARA; MUNIM; ABBAS](#),

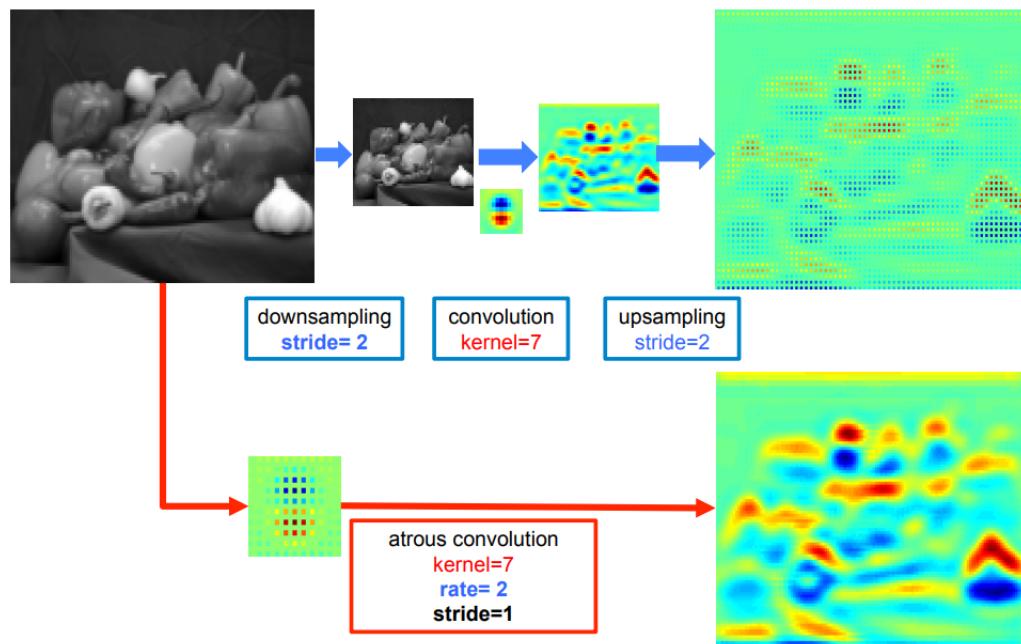
2019) e proposto em (CHEN *et al.*, 2017) e (YU, 2015). A comparação entre a extração de características pela convolução padrão e convolução dilatada pode ser observado nas figuras 29 e 30.

Figura 29 – Projeção das convoluções padrão e dilatada lado a lado.



Fonte: Sankar (2024).

Figura 30 – Ilustração da convolução dilatada bidimensional.



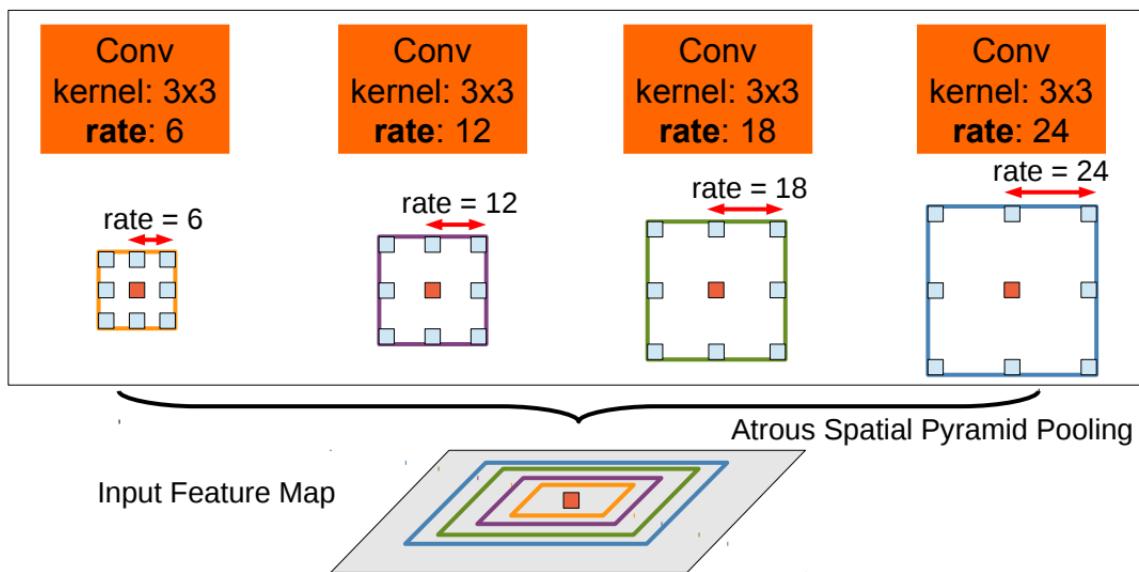
Fonte: Trevisan (2024).

Na figura 29, a ilustração da esquerda representa uma convolução padrão 3×3 , já a figura da direita uma convolução dilatada 3×3 com taxa de dilatação igual a 2. É evidente a diferença na dimensão final das convoluções. Já na figura 30, a linha superior: extração de recursos esparsos com convolução padrão em um mapa de recursos de entrada de baixa resolução, na linha inferior: extração de características densas com convolução dilatada com taxa igual 2, aplicado em um mapa de recursos de entrada de alta resolução.

O Atrous Spatial Pyramid Pooling (ASPP) é uma técnica utilizada para aprimorar redes neurais convolucionais na extração de características. Baseado no conceito de convolução

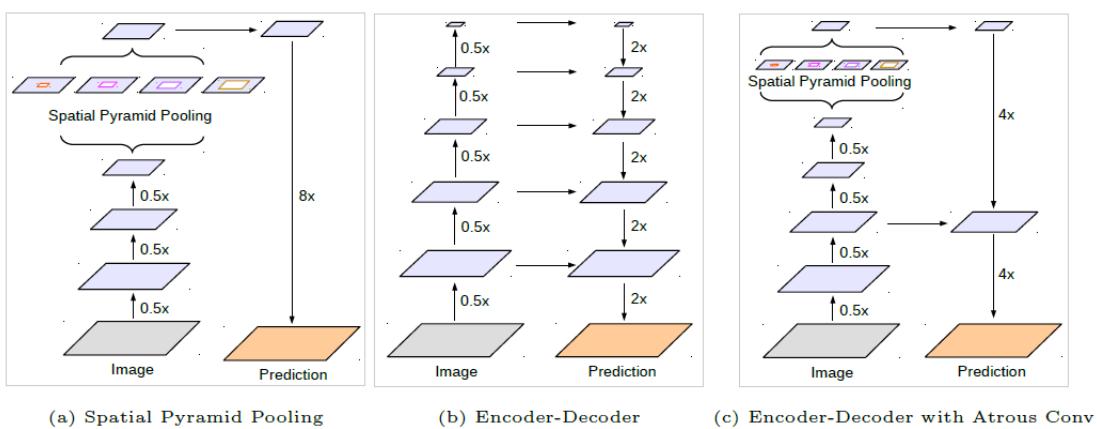
dilatadas, o ASPP captura informações contextuais em múltiplas escalas ao aplicar filtros com diferentes taxas de dilatação, como ilustrado na figura 31, por consequência permite que a rede obtenha um campo receptivo maior sem aumentar o tamanho do kernel, facilitando a análise de imagens em diversas resoluções, essa abordagem permite que a rede analise simultaneamente detalhes finos e características mais amplas da imagem, proporcionando uma compreensão mais rica e detalhada do conteúdo visual, e simplificando a amostragem ascendente como visto na figura 32. A arquitetura do DeepLabV3+ que apresenta o ASPP pode ser verificada na figura 33.

Figura 31 – Agrupamento de pirâmide espacial dilatada (ASPP).



Fonte: [Trevisan \(2024\)](#).

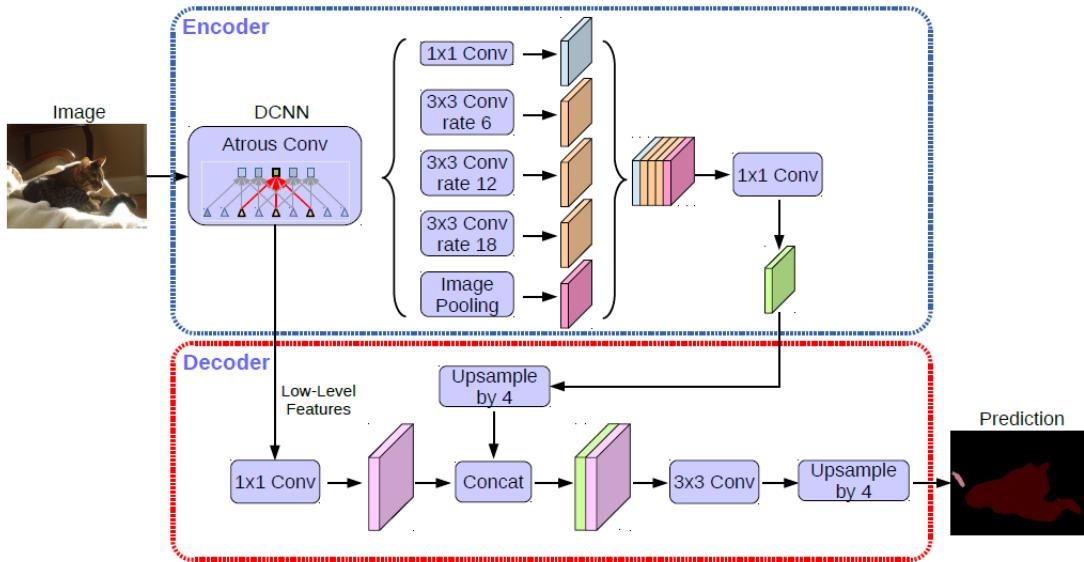
Figura 32 – Ilustra diferentes topologias com e sem ASPP.



Fonte: [Trevisan \(2024\)](#).

Na figura 31, para classificar o pixel central (laranja), o ASPP explora recursos em múltiplas escalas empregando vários filtros paralelos com taxas diferentes. Os campos de visão efetivos são mostrados em cores diferentes (azul, verde, roxo e laranja). Já na figura 32, na ilustração (a) : Com *Atrous Spatial Pyramid Pooling* (ASPP) , capaz de codificar informações

Figura 33 – Arquitetura DeepLabV3+ estendendo DeepLabV3.



Fonte: Trevisan (2024).

contextuais em várias escalas. ilustração (b): Com a Arquitetura *Encoder-Decoder*, a informação espacial é recuperada. A Arquitetura *Encoder-Decoder* provou ser útil na literatura como SegNet e U-Net para diferentes tipos de propósitos. ilustração (c): O DeepLabv3+ faz uso das topologias (a) e (b), simplificando o decodificador.

A versão profunda do ASPP, DASPP (*Deeper Atrous Spatial Pyramid Pooling*) expande essa abordagem ao adicionar mais camadas de convolução padrão em sequência as convoluções dilatadas e melhorar a integração de informações em múltiplas escalas. Essa profundidade adicional melhora a capacidade da rede de capturar características complexas e reduz artefatos na segmentação, resultando em previsões mais precisas e robustas. O modelo também inclui uma camada de *pooling global* para agregar informações espaciais e uma camada final de convolução para combinar e refinar as características extraídas em múltiplas escalas, como publicado no artigo ([EMARA; MUNIM; ABBAS, 2019](#)).

APÊNDICE C

CARACTERÍSTICAS DA COLETA DO DATASET CITYSCAPESBRAZIL

Tabela 20 – Volumetria de amostras por trajeto.

Região	Nº Amostras
Curitiba - SC	842
Fortaleza - CE	827
Rio de Janeiro - RJ	1455
Porto Alegre - RS	1554
São Paulo - SP	1647
Rodovias Nordeste-Sul	15160
CityScapesBrazil	21485

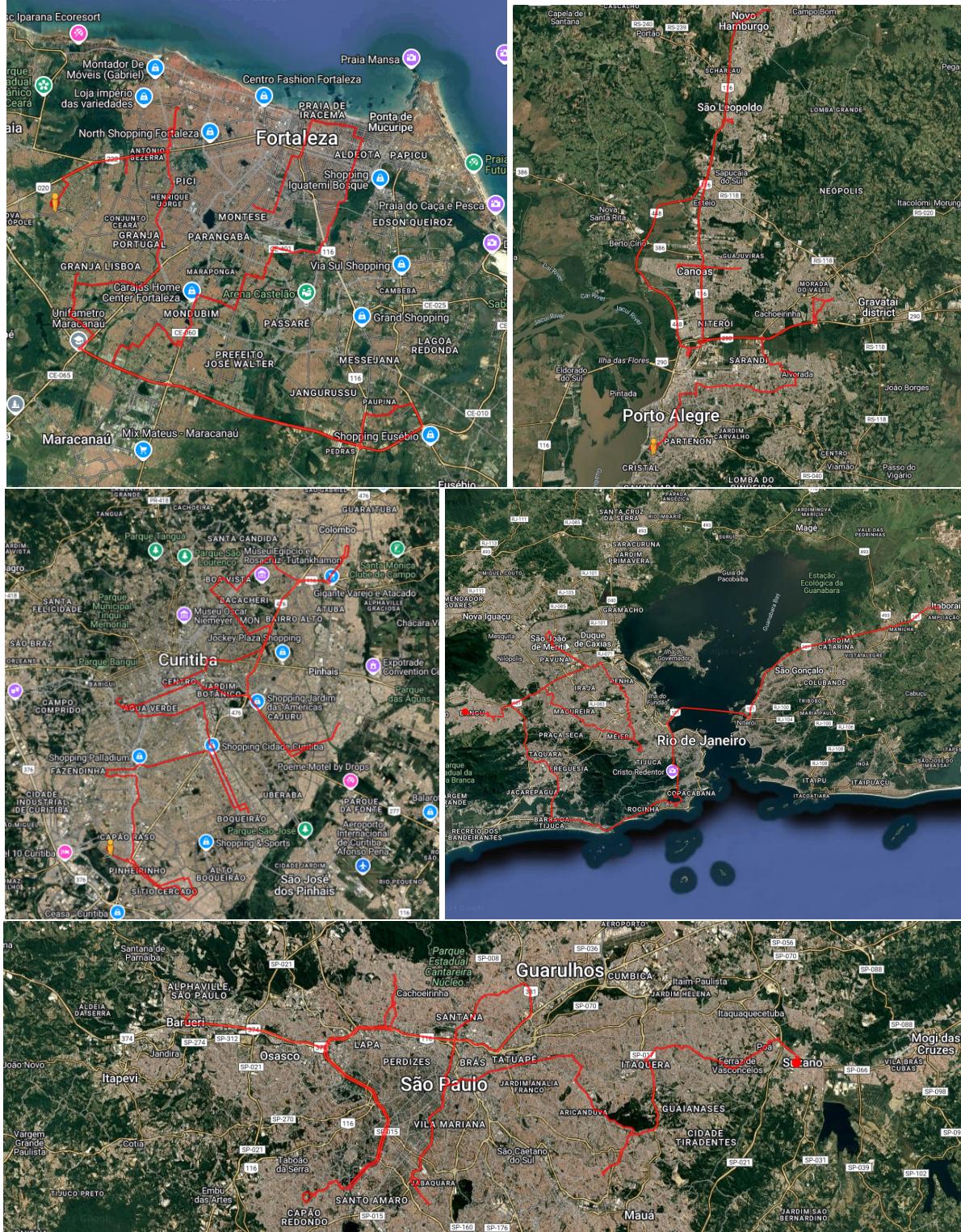
Fonte: Elaborada pelo autor.

Figura 34 – Exemplos do conjunto de dados obtido.



Fonte: Elaborada pelo autor.

Figura 35 – Trajetos percorridos no streetview, nas cidades de Fortaleza, Porto Alegre, Curitiba, Rio de Janeiro e São Paulo.



Fonte: MapChannels (2024).

Figura 36 – Trajetória por rodovias de nordeste a sul do Brasil.



Fonte: [MapChannels \(2024\)](#).

APÊNDICE D

AGRUPAMENTO DE CLASSES CITYSCAPES, CITYSCAPESBRAZIL E GTA5, E COMPARAÇÃO COM O ESTADO DA ARTE

Tabela 21 – Nome e codificação das classes padrão e para esta monografia

Nome da Classe	ID Classe	ID Nesta Monografia
Road	7	Navegável (0)
Sidewall	8	Inavegável (1)
Building	11	Obstáculos (2)
Wall	12	Obstáculos (2)
Fence	13	Obstáculos (2)
Pole	17	Obstáculos (2)
Traffic Light	19	Obstáculos (2)
Traffic Sign	20	Obstáculos (2)
Vegetation	21	Obstáculos (2)
Terrain	22	Inavegável (1)
Sky	23	Inavegável (1)
Person	24	Obstáculos (2)
Rider	25	Obstáculos (2)
Car	26	Obstáculos (2)
Truck	27	Obstáculos (2)
Bus	28	Obstáculos (2)
Train	31	Obstáculos (2)
Motorcycle	32	Obstáculos (2)
Bicycle	33	Obstáculos (2)

Fonte: Elaborada pelo autor.

D.1 Comparação com o estado da arte

Tabela 22 – Comparação do estado da arte de segmentação semântica para o conjunto CityScape para teste, sobre MIoU e classe Navegável correspondente a classe padrão “road”, id = 7.

Ano/Referência	Modelo	BackBone	MIoU	IoU Navegável (classe road)
Consistência Semi-Supervisionado				
2024 (LI <i>et al.</i> , 2019) (ZHUANG <i>et al.</i> , 2018)	CCTLiteGC GALD-Net DRN-CRL	MobileNet RN-50 RN-38	- 32,9 32,9	87,86 98,80 98,80
Adaptação de Domínio Consistência				
2024 2024 2024 (MELAS-KYRIAZI; MANRAI, 2021) (PAN <i>et al.</i> , 2020) (KANG <i>et al.</i> , 2020) (LI <i>et al.</i> , 2020) (LEE <i>et al.</i> , 2019)	PixLiteGC PixLiteGC19 CTTLiteMDGCB PixMatch IntraDA PLCA CCM SWD	MobileNet MobileNet MobileNet RN-101 RN-101 RN-101 RN-101 RN-101	- 47,81 - 65,8 61,5 59,3 66,6 61,4	95,38 94,53 93,82 91,60 90,60 84,00 93,50 92,00
Auto-Treinamento				
(HOYER; DAI; GOOL, 2022b) (TRANHEDEN <i>et al.</i> , 2021) (WANG <i>et al.</i> , 2021) (ZHANG <i>et al.</i> , 2021) (IQBAL; ALI, 2020) (IQBAL; ALI, 2020) (LIAN <i>et al.</i> , 2019) (ZOU <i>et al.</i> , 2019) (ZOU <i>et al.</i> , 2019) (ZOU <i>et al.</i> , 2019) (CHEN; XUE; CAI, 2019) (CHEN; XUE; CAI, 2019) (VU <i>et al.</i> , 2019) (ZOU <i>et al.</i> , 2018)	DAFormer DACS CorDA ProDA MLSL(+PWL)ST MLSL(SISC)ST PyCDA CRST(LRENT) CRST(MRKLD) CRST(MRENT) MS+IW+Multi MaxSquare(MS) MinEnt CBST-SP	MiT-B5 RN-101 RN-101 RN-101 RN-101 RN-101 RN-101 RN-101 RN-101 RN-101 RN-101 RN-101 WRN-38	31,9 30,0 31,6 29,3 59,4 60,5 61,7 62,9 63,4 63,0 61,7 58,4 55,1 59,2	95,70 89,90 94,70 87,80 89,00 91,00 90,50 91,80 91,00 91,80 89,40 88,10 86,20 88,00
Transferência de Estilo / Imagem para Imagem				
(KIM; BYUN, 2020) (YANG; SOATTO, 2020) (YANG; SOATTO, 2020) (CHEN <i>et al.</i> , 2019) (LI; YUAN; VASCONCELOS, 2019)	LTIR FDA(Ensemble) FDA CrDoCo BDL(M2-F2)	RN-101 RN-101 RN-101 DRN-26 RN-101	66,3 65,3 59,4 62,7 64,0	92,90 92,50 88,80 95,10 91,00
Adversárias				
(HUANG <i>et al.</i> , 2020) (WANG <i>et al.</i> , 2020) (WANG <i>et al.</i> , 2020) (ZHANG <i>et al.</i> , 2019) (TSAI <i>et al.</i> , 2019) (VU <i>et al.</i> , 2019) (VU <i>et al.</i> , 2019) (TSAI <i>et al.</i> , 2018) (TSAI <i>et al.</i> , 2018)	CrCDA FADA-MST FADA CAG Patch-Disc AdvEnt+MinEnt AdvEnt AdaptSegNet-LS AdaptSegNet	RN-101 RN-101 RN-101 RN-101 RN-101 RN-101 RN-101 RN-101 RN-101	64,7 65,1 64,5 64,2 63,2 60,2 59,6 61,2 55,8	92,40 91,00 92,50 90,40 92,30 89,40 89,90 91,40 86,50
Supervisionados				
(CHEN <i>et al.</i> , 2018) (CHEN <i>et al.</i> , 2018) (ZHAO <i>et al.</i> , 2017b) (LONG; SHELHAMER; DARRELL, 2015b) (LONG; SHELHAMER; DARRELL, 2015b)	DeepLabV3+ DeepLabV3+ PSPnet FCN-8s FCN-8s	RN-50 RN-101 RN-101 RN-50 RN-101	67,8 65,2 70,8 98,2 98,4	98,20 98,40 98,60 97,90 98,20
Média			Pendente	Pendente
Desvio Padrão			Pendente	Pendente

Fonte: Elaborada pelo autor.

APÊNDICE E

RESULTADO GERAL

Tabela 23 – Resultados obtidos para os conjuntos CityScapes, GTA5 e CityScapesBrazil, sobre o conjunto de teste e validação considerando as melhores épocas verificadas em validação ao decorrer do treinamento.

Modelo	Conjunto	MIOU		MPA		Shannon (bit)		Época
		Val	Test	Val	Test	Val	Test	
PixGC19	CityScape	46,9	0	55,4	0	2,092	2,149	96
PixGC	CityScape	84,2	-	90,1	-	1,316	1,306	17
PixGC	GTA5	88,4	88,4	93,6	93,4	1,398	1,413	91
PixGC	CityScapeBrazil	-	-	-	-	1,232	1,242	17
PixCB	CityScape	85,4	-	89,8	-	1,291	1,278	6
PixCB	CityScapeBrazil	-	-	-	-	1,371	1,394	12
PixGB	CityScape	68,7	-	79,5	-	1,311	1,278	100
PixGB	GTA5	65,1	65,6	78,4	77,9	1,442	1,430	81
PixGB	CityScapeBrazil	-	-	-	-	1,425	1,437	100
CCTGC	CityScape	73,8	-	86,4	-	1,310	1,324	1
CCTGC	GTA5	89,2	89,1	94,3	94,1	1,389	1,435	22
CCTGC	CityScapeBrazil	-	-	-	-	1,085	1,095	1
CCTCB	CityScape	83,2	-	91,9	-	1,325	1,333	14
CCTCB	CityScapeBrazil	-	-	-	-	1,378	1,403	28
CCTGB	CityScape	18,8	-	30,8	-	0,890	0,899	40
CCTGB	GTA5	17	17,6	32,5	32,7	0,855	0,860	32
CCTGB	CityScapeBrazil	-	-	-	-	0,897	0,898	40
CTTMDGCB	GTA5	87,9	87,3	93,4	92,8	1,392	1,430	39
CTTMDGCB	CityScape	85,2	-	91,2	-	1,317	1,322	2
CTTMDGCB	CityScapeBrazil	-	-	-	-	1,367	1,377	35

Fonte: Elaborada pelo autor.

Tabela 24 – Média da intersecção da união, e IoU para cada uma das classes Navegável, Inavegável e Obstáculos, sobre o conjunto de validação para as melhores épocas respectivas a cada modelo e base.

Modelo	Conjunto	MIoU	IoU Navegável	IoU Inavegável	IoU Obstáculos
PixGC	CityScape	84,2	91,18	65,54	95,88
PixGC	GTA5	88,4	88,89	81,73	94,59
PixCB	CityScape	85,4	92,98	67,38	95,77
PixGB	CityScape	68,7	75,94	44,74	85,55
PixGB	GTA5	65,1	66,03	50,20	79,19
CCTGC	CityScape	73,8	75,94	51,67	93,90
CCTGC	GTA5	89,2	89,96	83,90	93,63
CCTCB	CityScape	83,2	88,15	65,12	96,18
CCTGB*	CityScape	18,8	16,39	0,00	39,90
CCTGB*	GTA5	17,0	16,72	0,00	34,40
CTTMDGCB	CityScape	85,2	90,96	68,15	96,62
CTTMDGCB	GTA5	87,9	88,18	81,41	94,03
Média		81,1	84,8	66,0	92,5
Desvio Padrão		8,7	8,9	13,8	5,7

Fonte: Elaborada pelo autor.

* Considerado como outlier para o cálculo da média e desvio padrão

Tabela 25 – Média da intersecção da união, e IoU para cada uma das classes Navegável, Inavegável e Obstáculos, sobre o conjunto de teste para as melhores épocas respectivo a cada modelo e base.

Modelo	Conjunto	MIoU	IoU Navegável	IoU Inavegável	IoU Obstáculos
PixGC	CityScape	-	95,38	-	-
PixGC	GTA5	88,4	90,25	80,57	94,58
PixCB	CityScape	-	91,06	-	-
PixGB	CityScape	-	70,20	-	-
PixGB	GTA5	65,6	70,08	49,37	77,42
CCTGC	CityScape	-	87,86	-	-
CCTGC	GTA5	89,1	91,17	82,21	93,84
CCTCB	CityScape	-	92,41	-	-
CCTGB*	CityScape	-	10,20	-	-
CCTGB*	GTA5	17,6	18,32	0,00	34,53
CTTMDGCB	CityScape	-	93,82	-	-
CTTMDGCB	GTA5	87,3	89,05	79,08	93,70
Média		82,6	87,1	72,8	89,9
Desvio Padrão		11,4	9,2	15,7	8,3

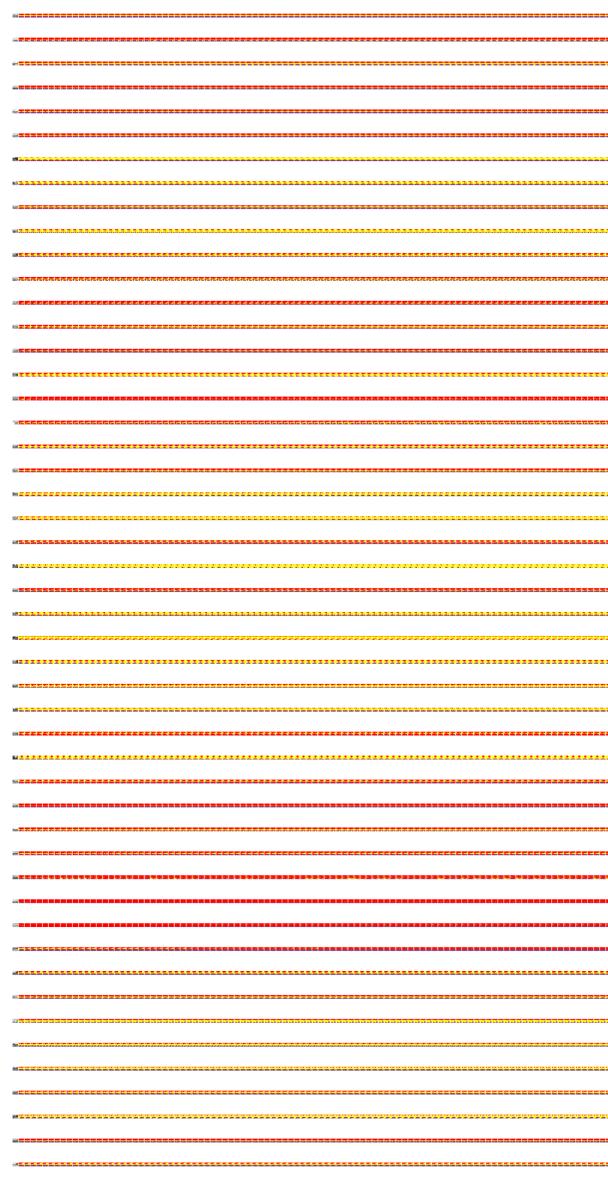
Fonte: Elaborada pelo autor.

* Considerado como outlier para o cálculo da média e desvio padrão

APÊNDICE F

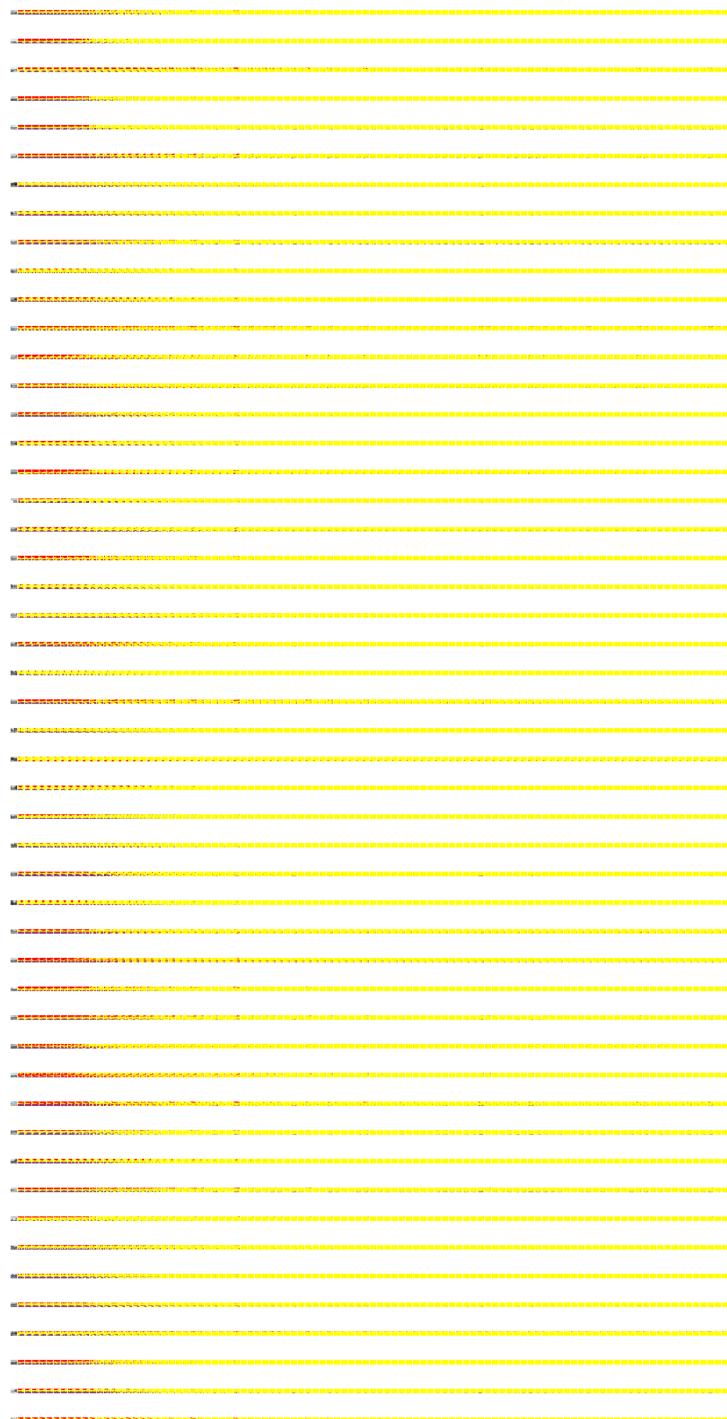
INSPEÇÃO VISUAL

Figura 37 – Amostra com 50 imagens do conjunto de validação do CityScapesBrazil para 100 épocas no modelo PixMatch: GTA5 para CityScapes



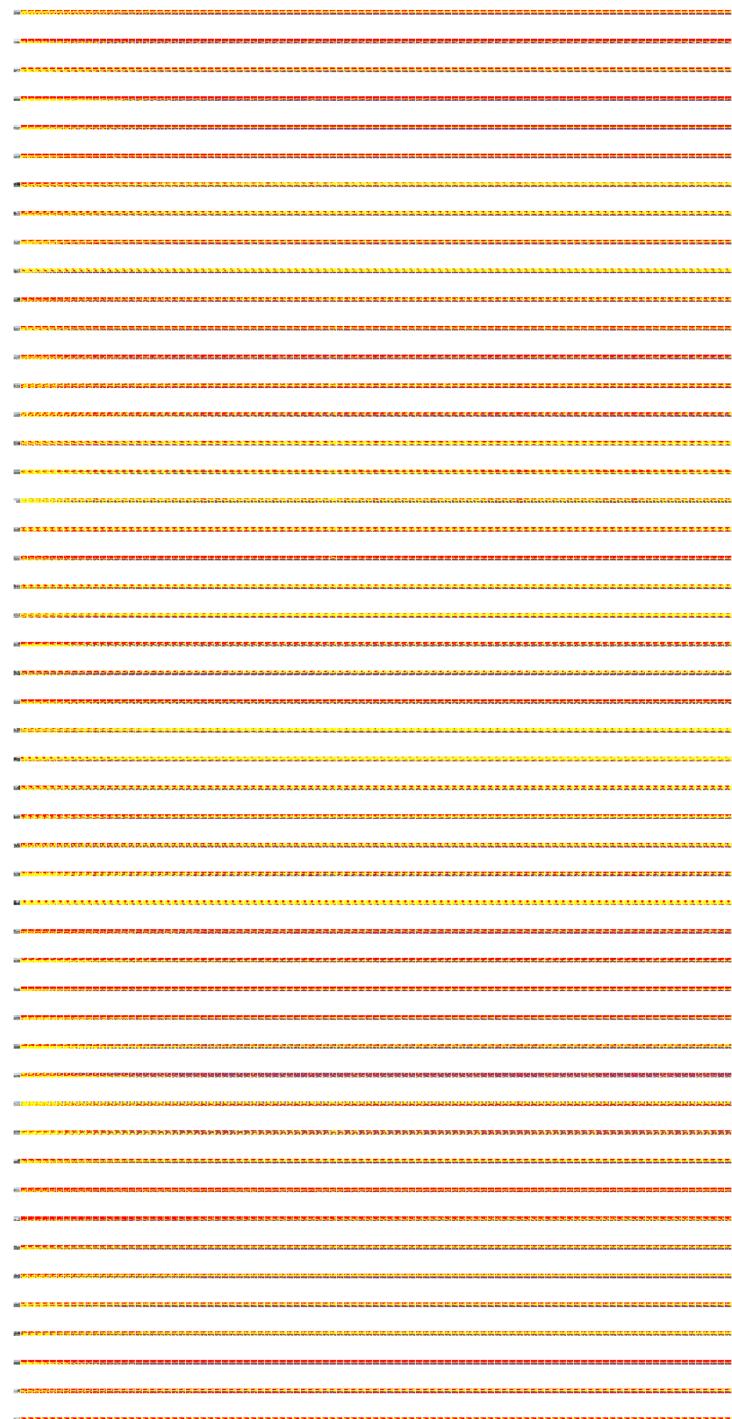
Fonte: Elaborada pelo autor.

Figura 38 – Amostra com 50 imagens do conjunto de validação do CityScapesBrazil para 100 épocas no modelo PixMatch: CityScapes para CityScapesBrazil



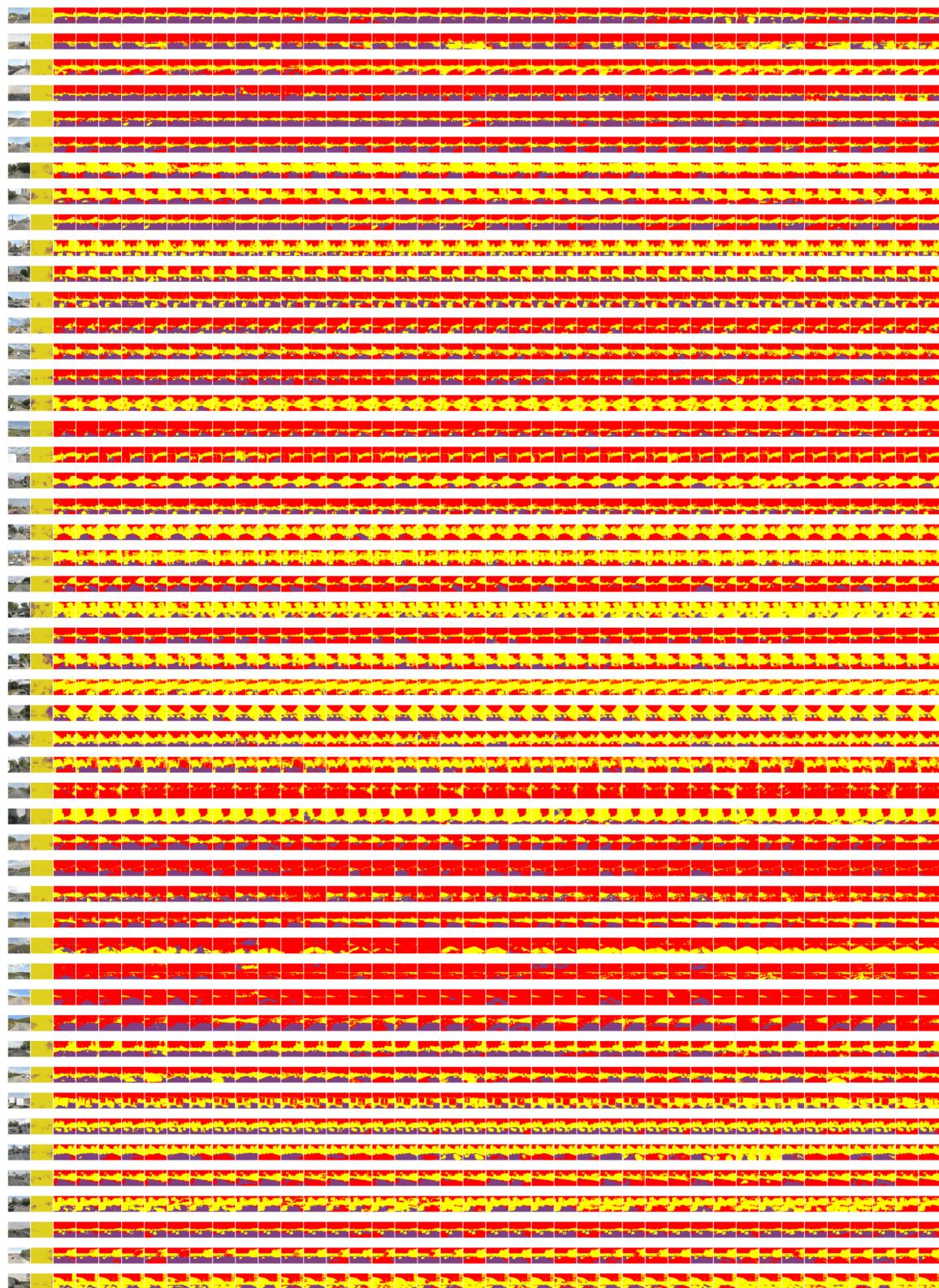
Fonte: Elaborada pelo autor.

Figura 39 – Amostra com 50 imagens do conjunto de validação do CityScapesBrazil para 100 épocas no modelo PixMatch: GTA5 para CityScapesBrazil



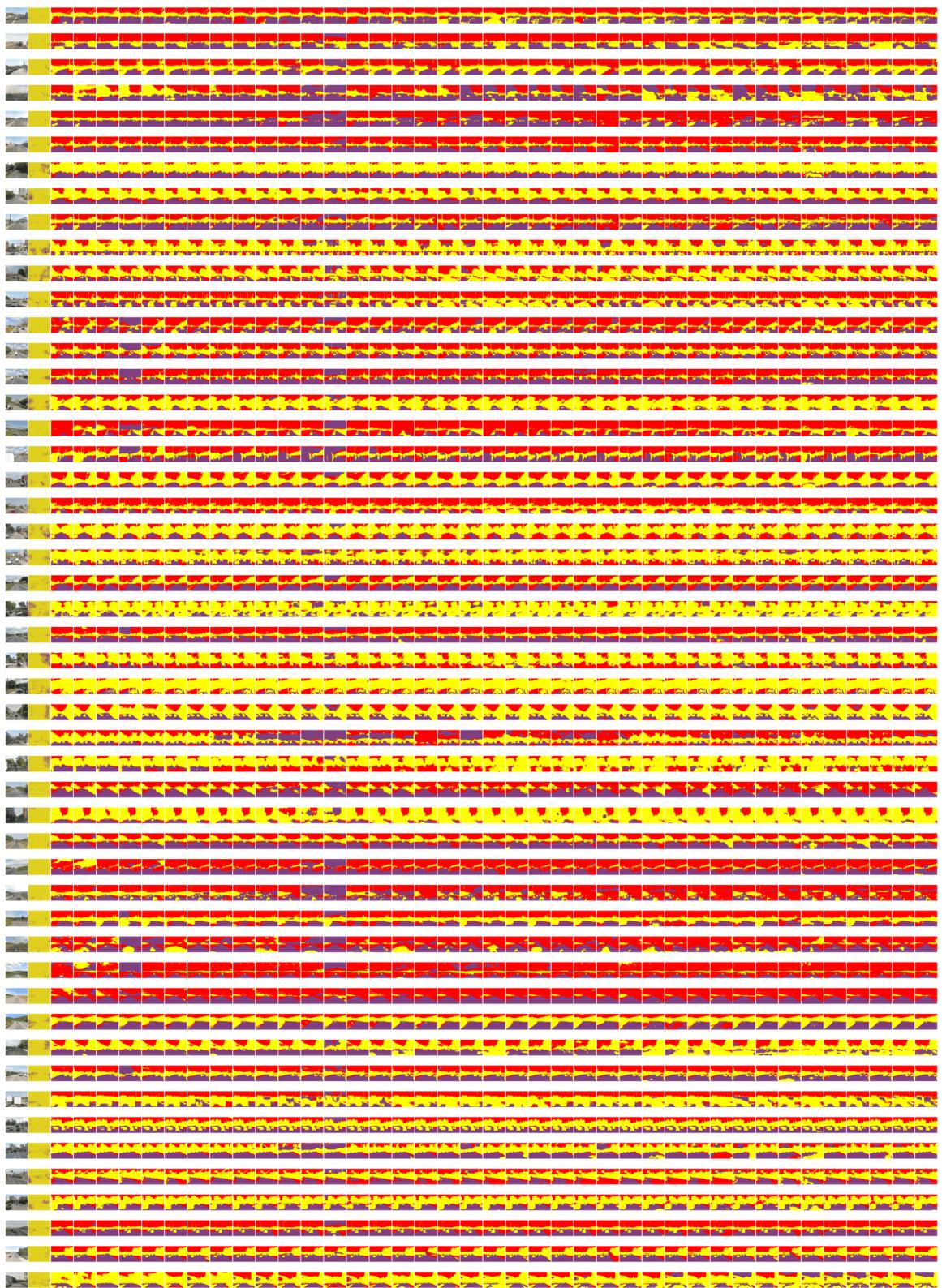
Fonte: Elaborada pelo autor.

Figura 40 – Amostra com 50 imagens do conjunto de validação do CityScapesBrazil para 40 épocas no modelo CCT: GTA5 para CityScapes



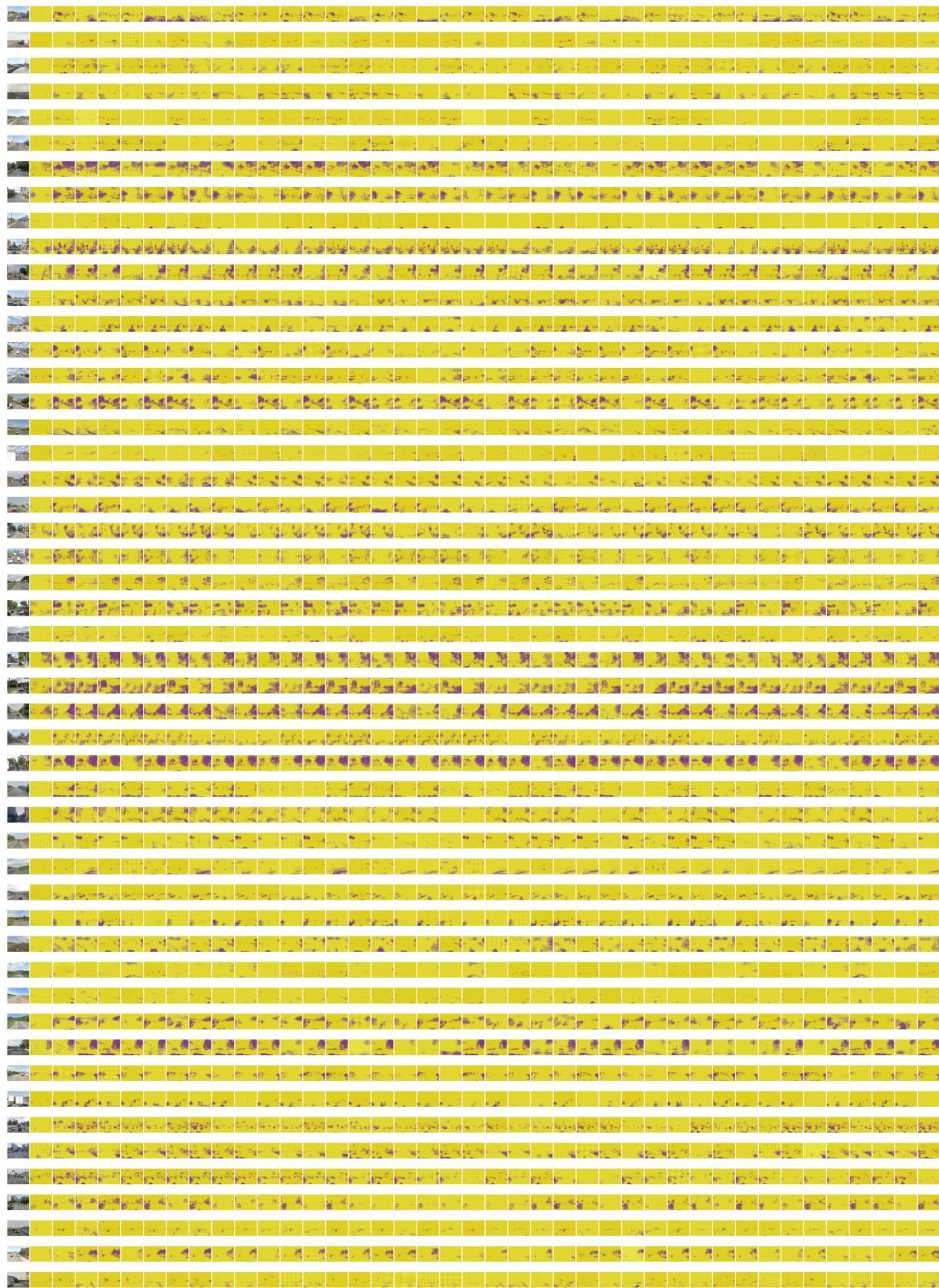
Fonte: Elaborada pelo autor.

Figura 41 – Amostra com 50 imagens do conjunto de validação do CityScapesBrazil para 40 épocas no modelo CCT: CityScapes para CityScapesBrazil



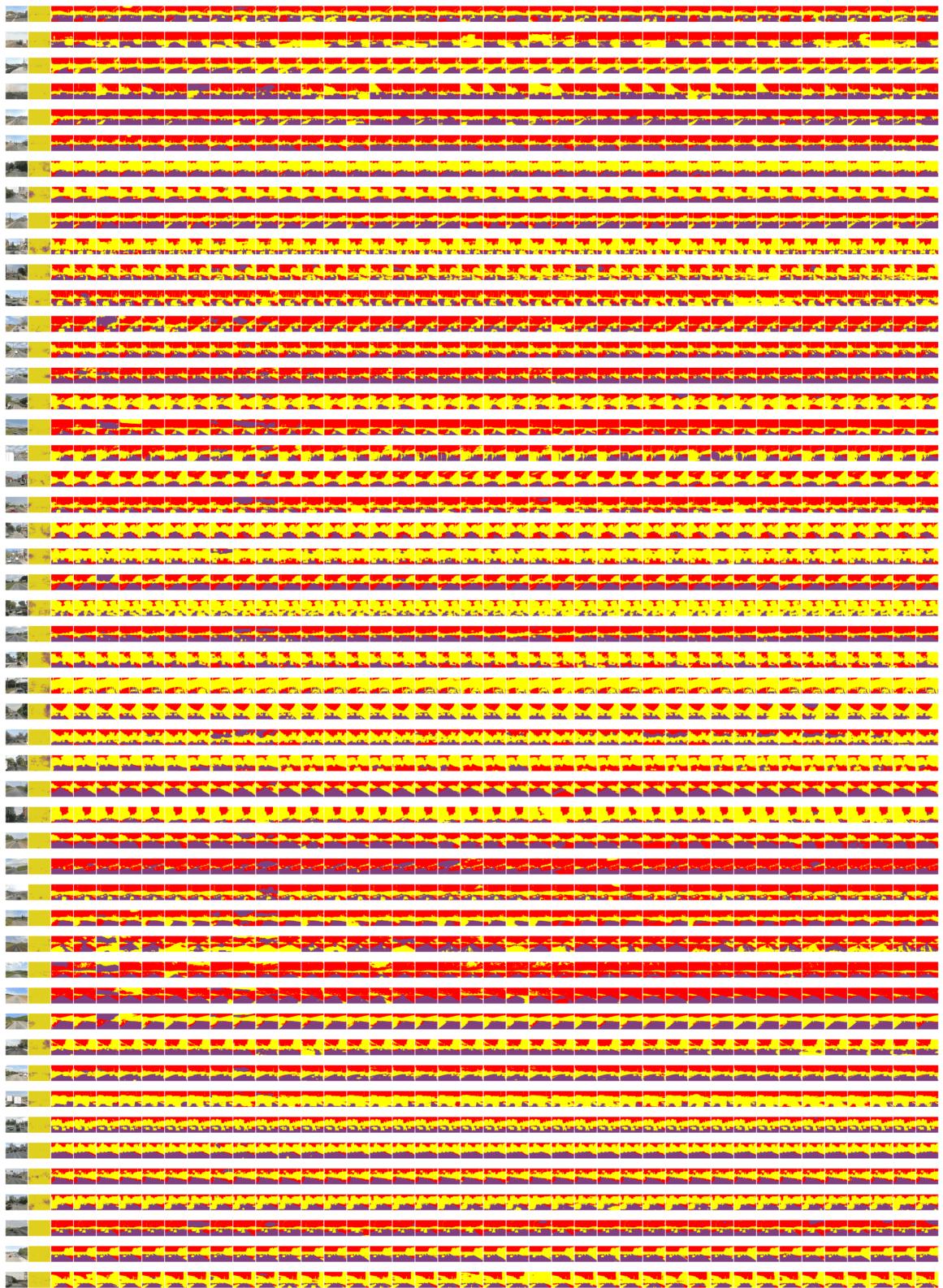
Fonte: Elaborada pelo autor.

Figura 42 – Amostra com 50 imagens do conjunto de validação do CityScapesBrazil para 40 épocas no modelo CCT: GTA5 para CityScapesBrazil



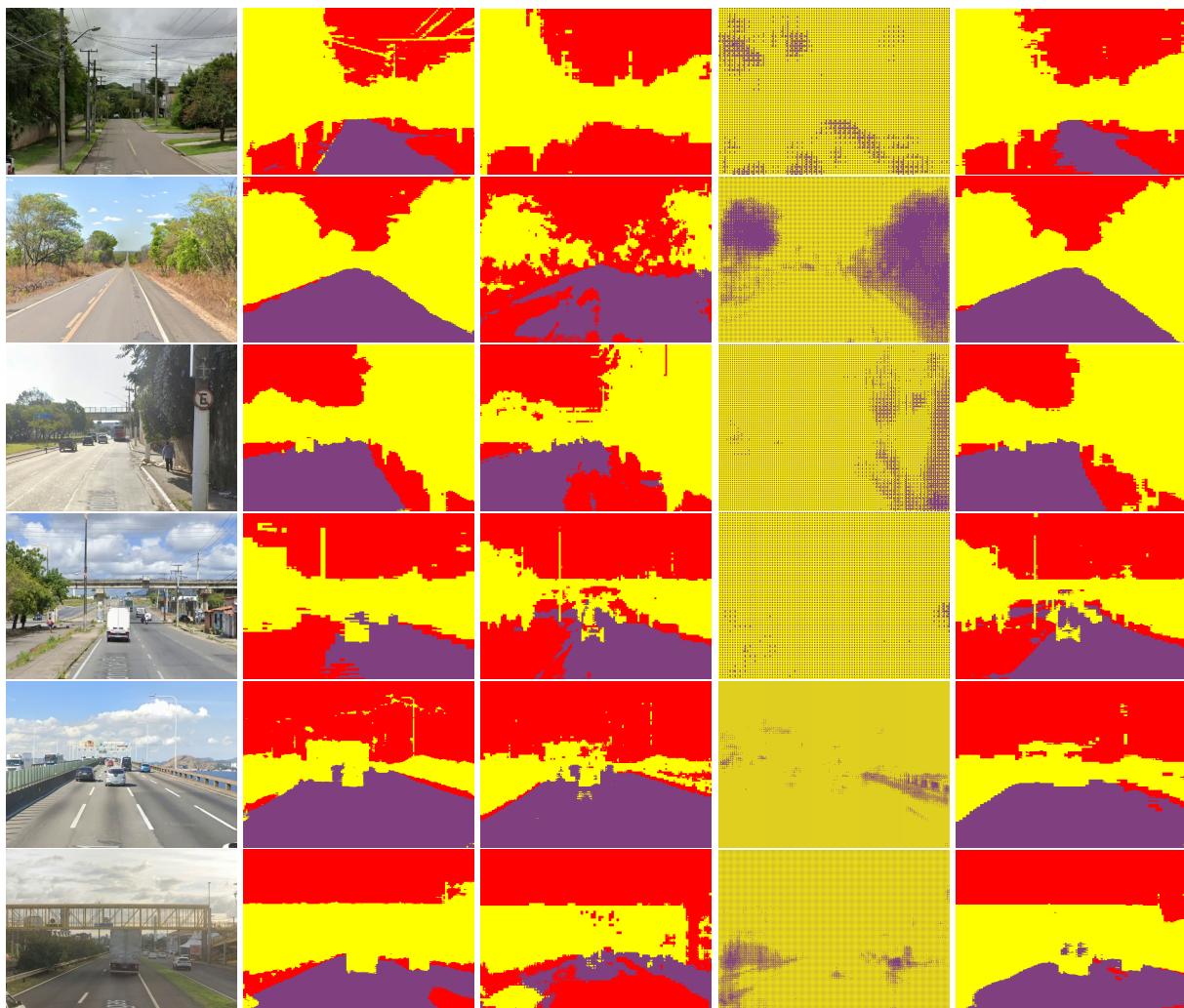
Fonte: Elaborada pelo autor.

Figura 43 – Amostra com 50 imagens do conjunto de validação do CityScapesBrazil para 40 épocas no modelo CCT: GTA5 para CityScapes e CityScapesBrazil



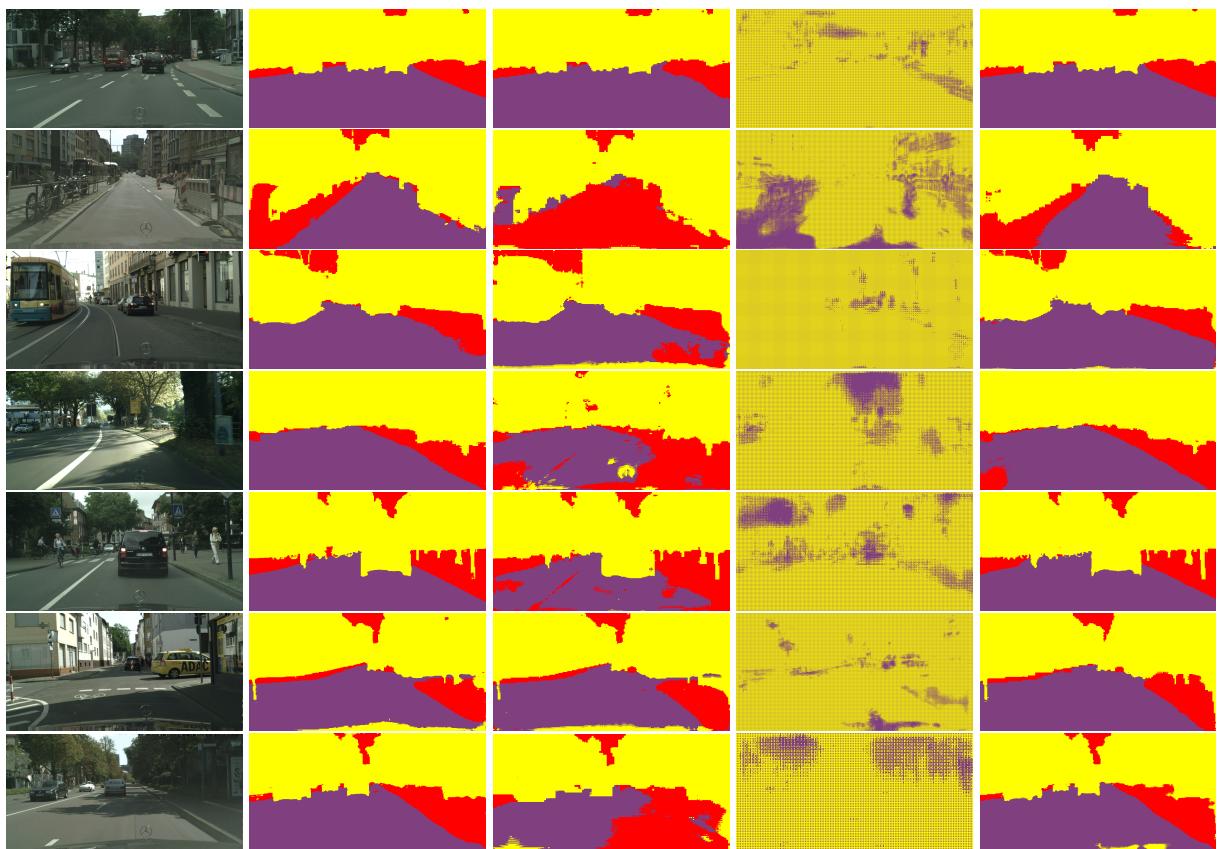
Fonte: Elaborada pelo autor.

Figura 44 – Exemplos do conjunto de validação do CityScapesBrazil na primeira coluna, e suas previsões dos modelos CCTCB, CCTGC, CCTGB e CTTMDGCB, nas respectivas colunas seguintes.



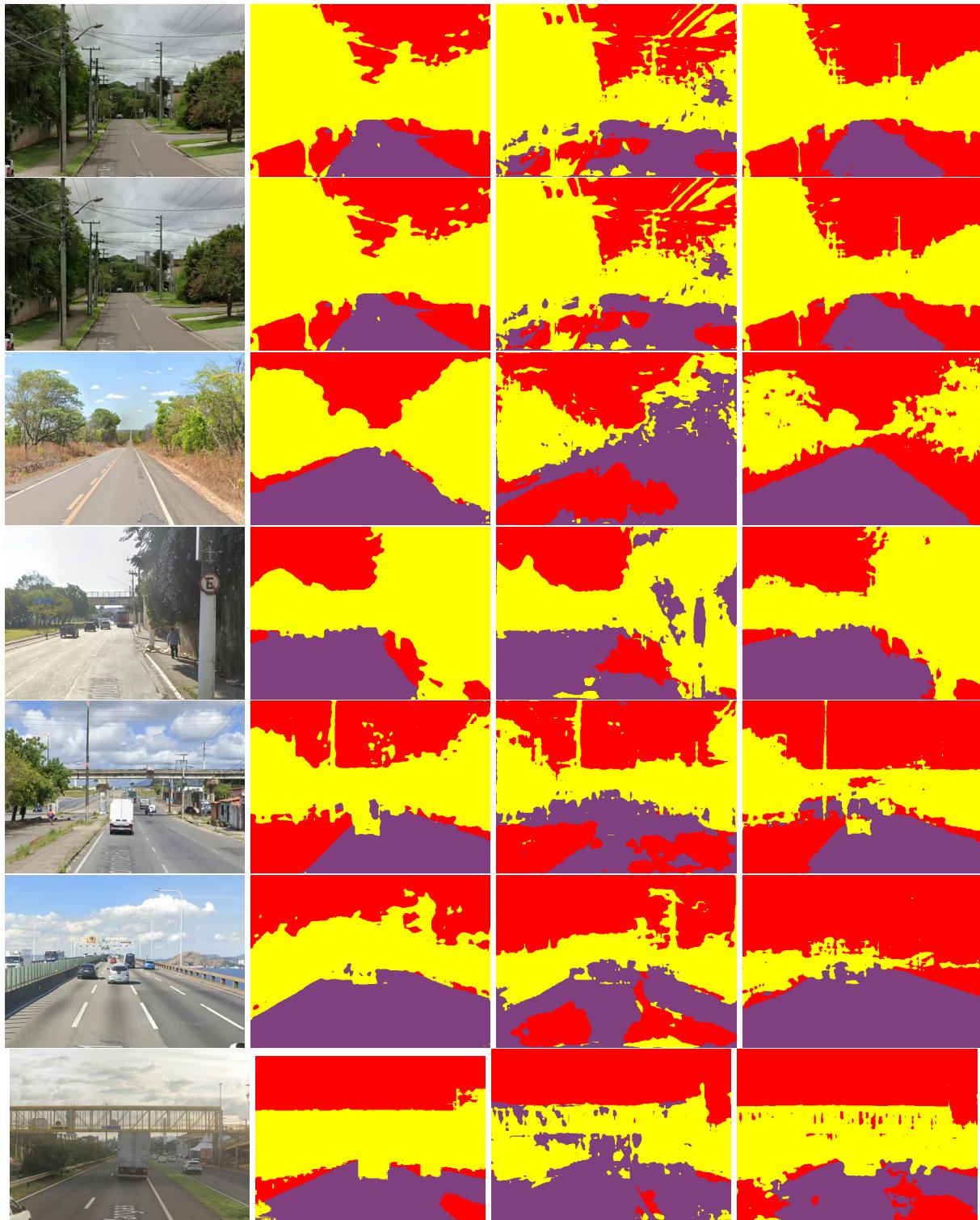
Fonte: Elaborada pelo autor.

Figura 45 – Exemplos do conjunto de validação do CityScapes na primeira coluna, e suas previsões nos modelos CCTCB, CCTGC, CCTGB e CTTMDGCB, nas respectivas colunas seguintes.



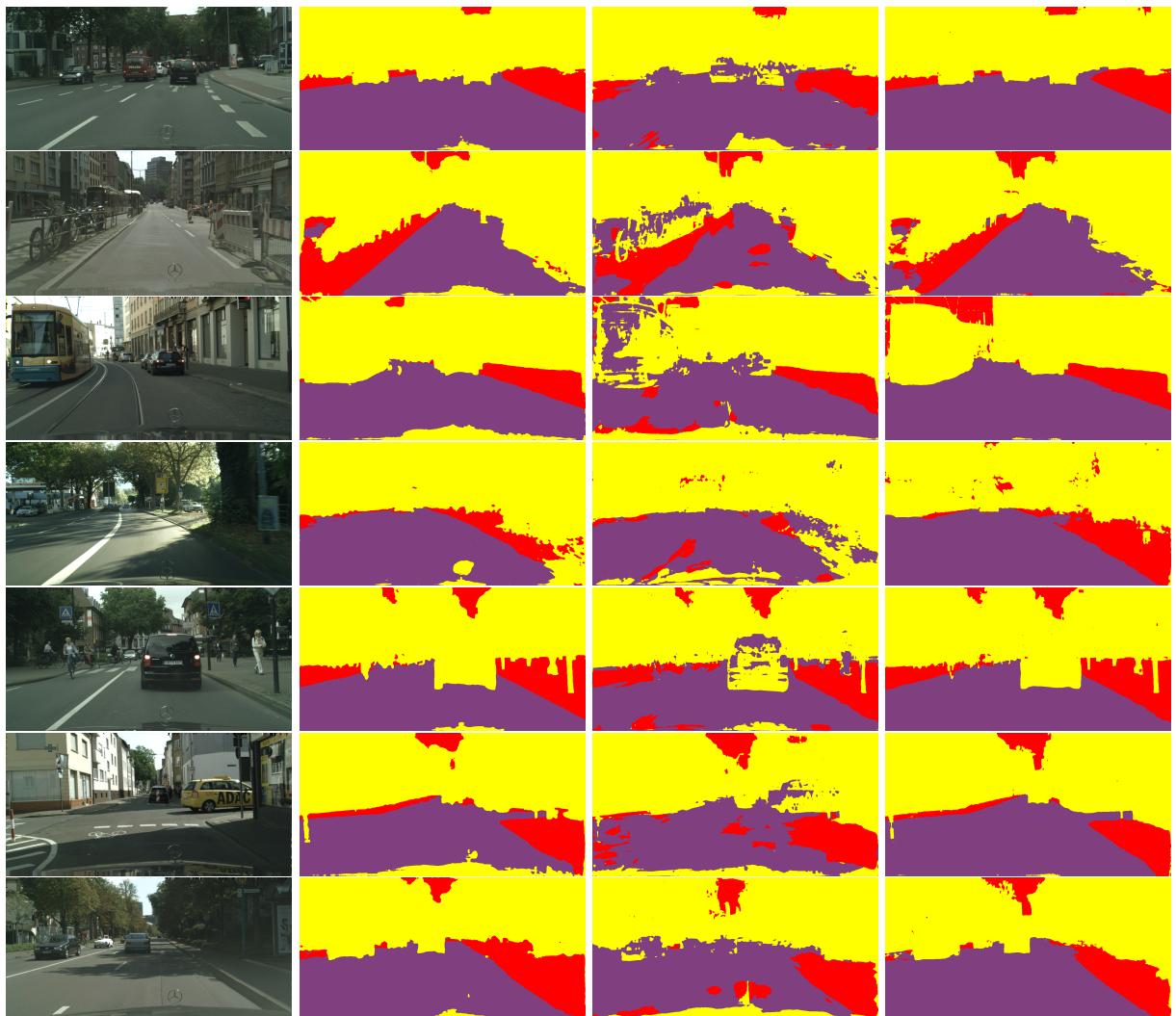
Fonte: Elaborada pelo autor.

Figura 46 – Exemplos do conjunto de validação do CityScapesBrazil na primeira coluna, e suas previsões nos modelos PIXCB, PIXGB, e PIXGC, nas respectivas colunas seguintes.



Fonte: Elaborada pelo autor.

Figura 47 – Exemplos do conjunto de validação do CityScapes na primeira coluna, e suas previsões nos modelos PIXCB, PIXGB, e PIXGC, nas respectivas colunas seguintes.

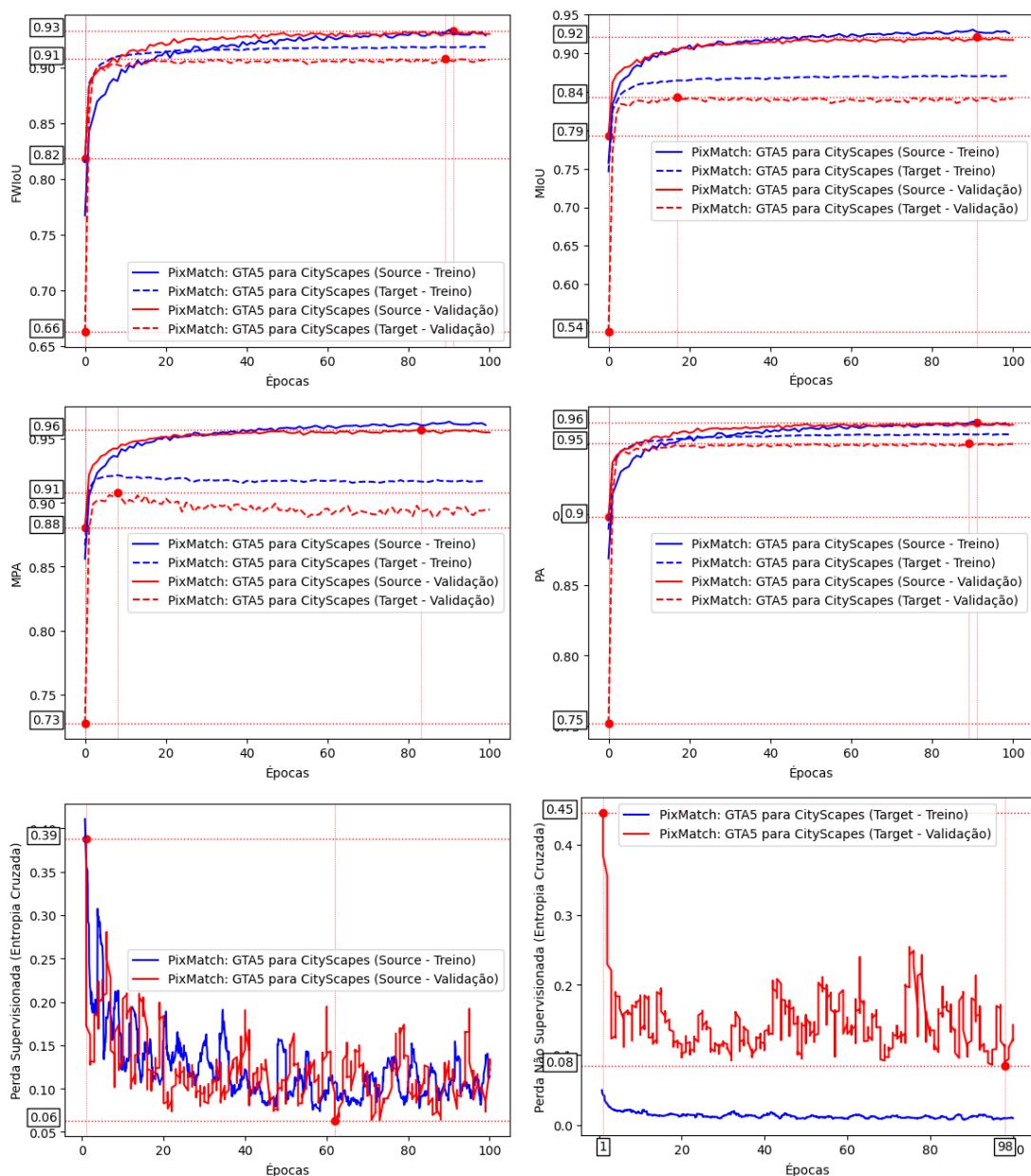


Fonte: Elaborada pelo autor.

APÊNDICE G

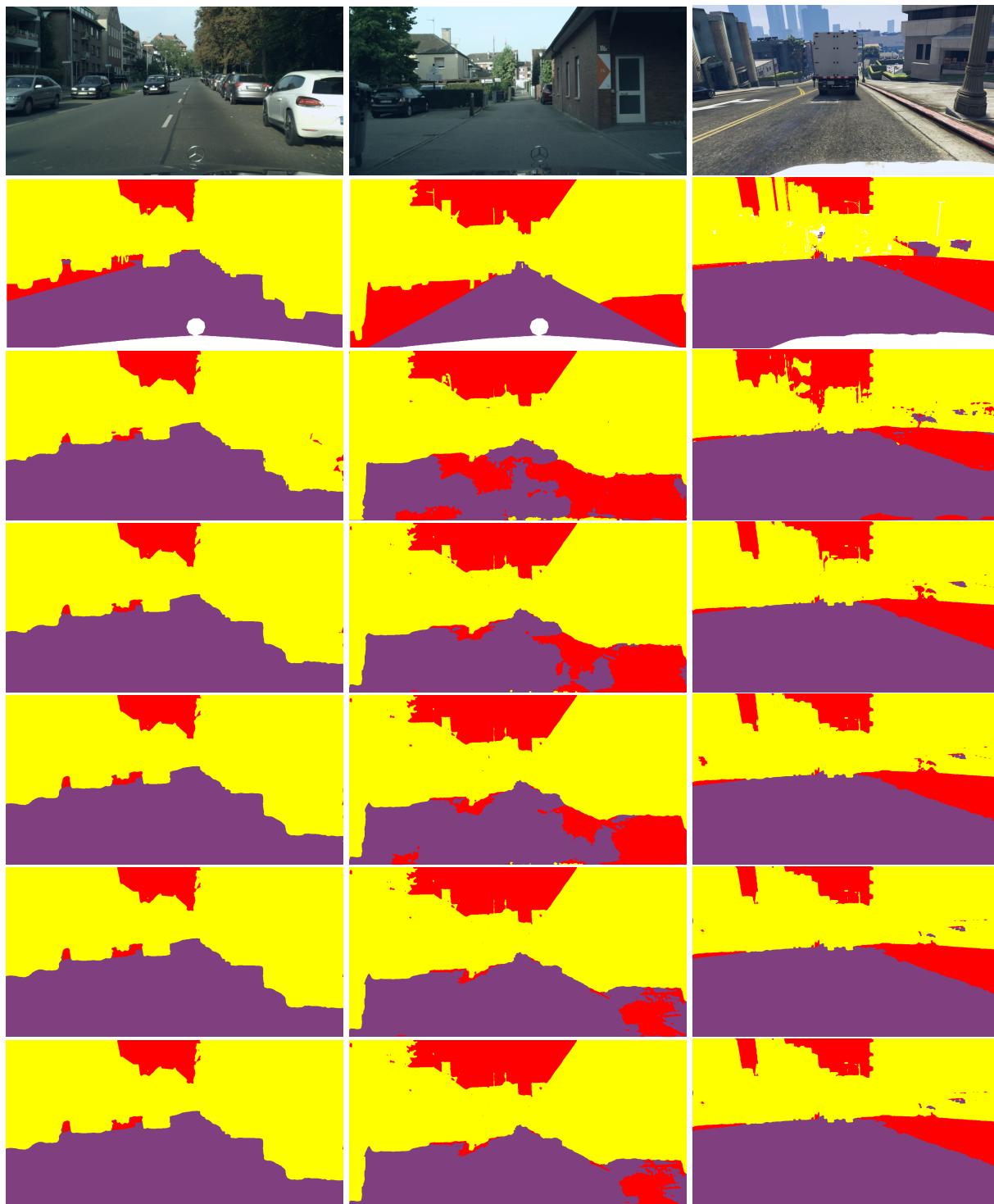
EVOLUÇÃO DOS MODELOS

Figura 48 – Métricas Padrão - PixMatch: GTA5 para CityScapes



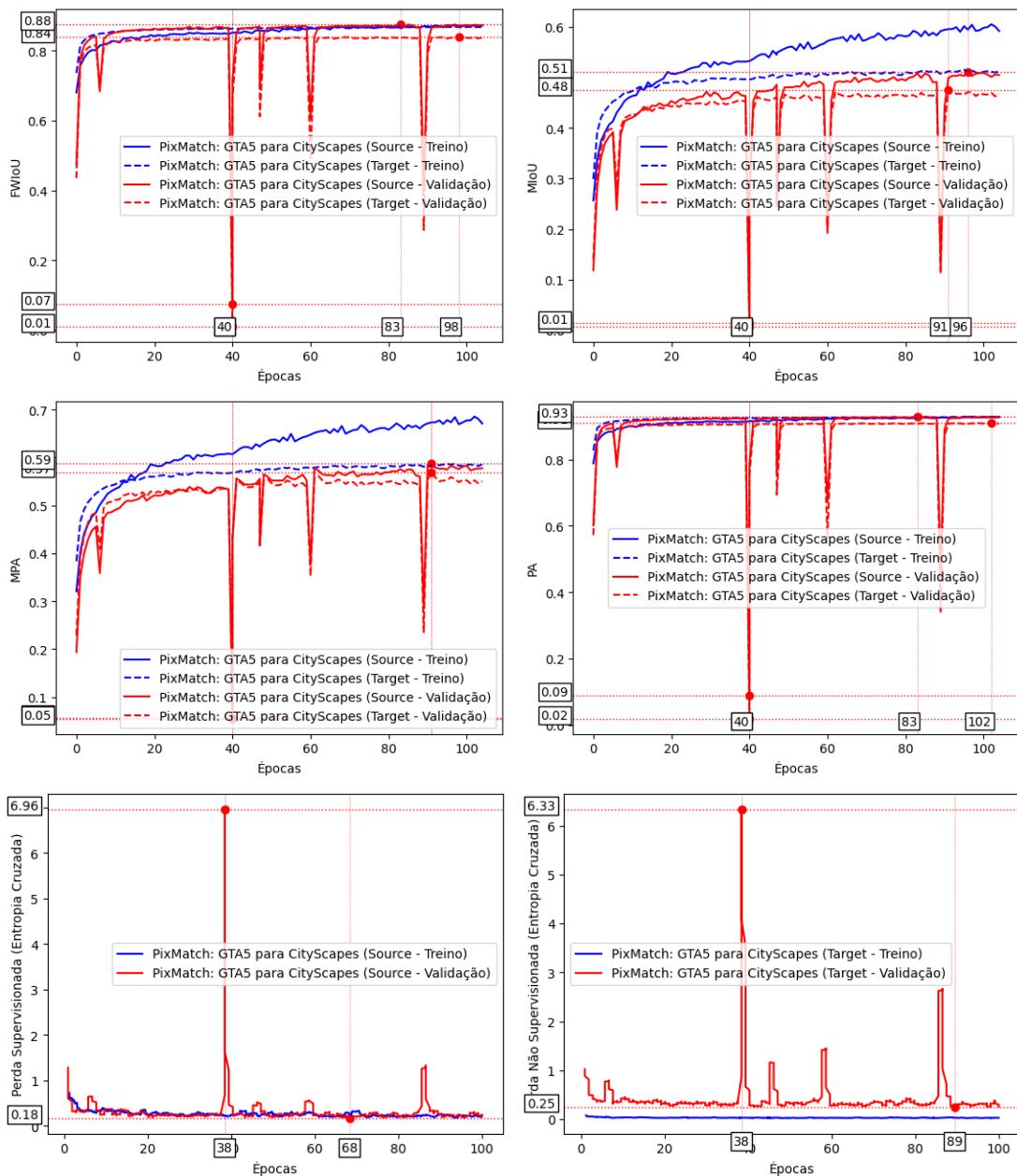
Fonte: Elaborada pelo autor.

Figura 49 – Exemplos de predição ao longo das épocas para o modelo PixMatch: GTA5 para CityScapes, com amostras a cada 20 épocas, primeira linha imagem original, segunda linha rótulo verdadeira, demais linhas são as previsões



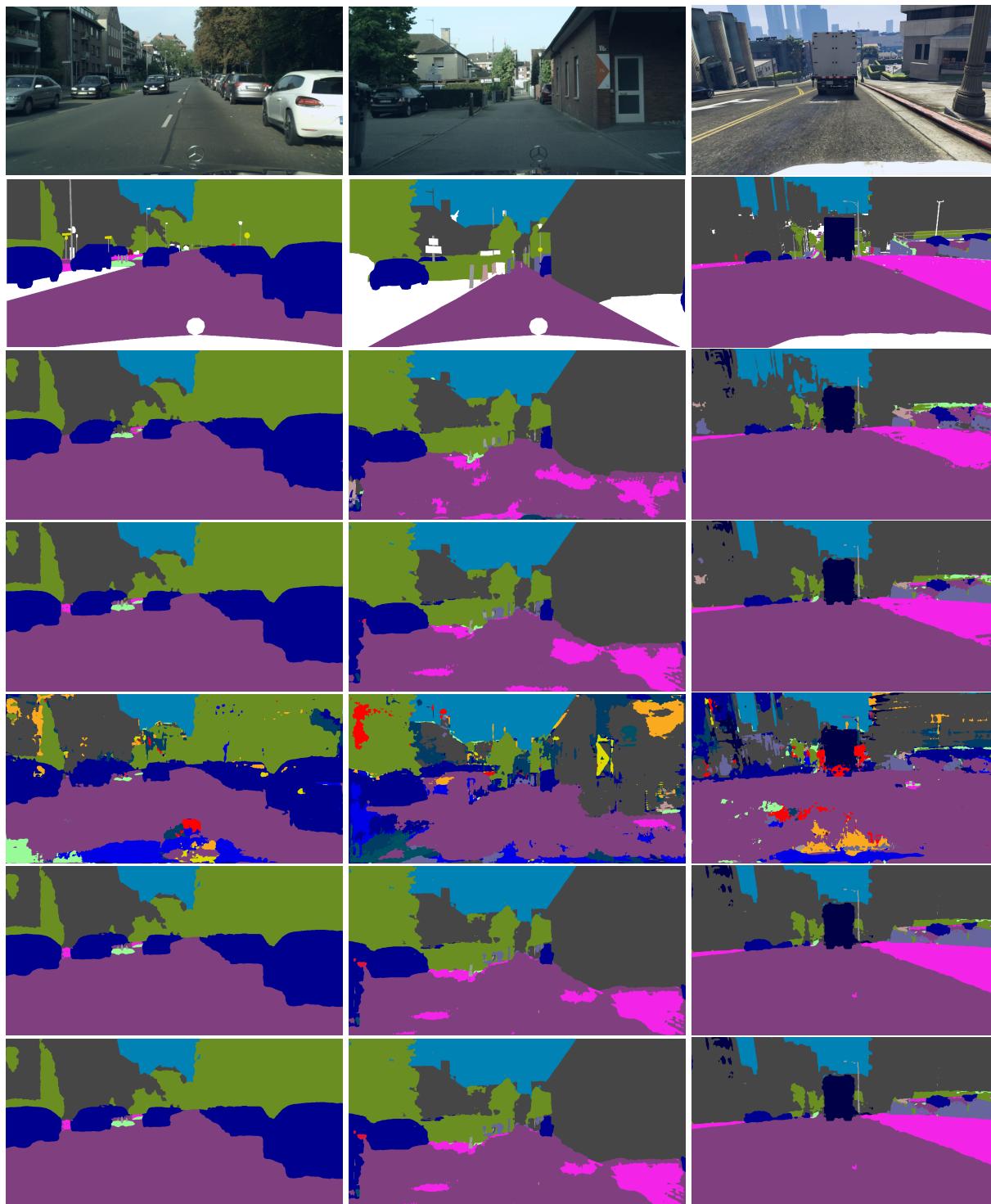
Fonte: Elaborada pelo autor.

Figura 50 – Métricas Padrão - PixMatch: GTA5 para CityScapes (19 classes)



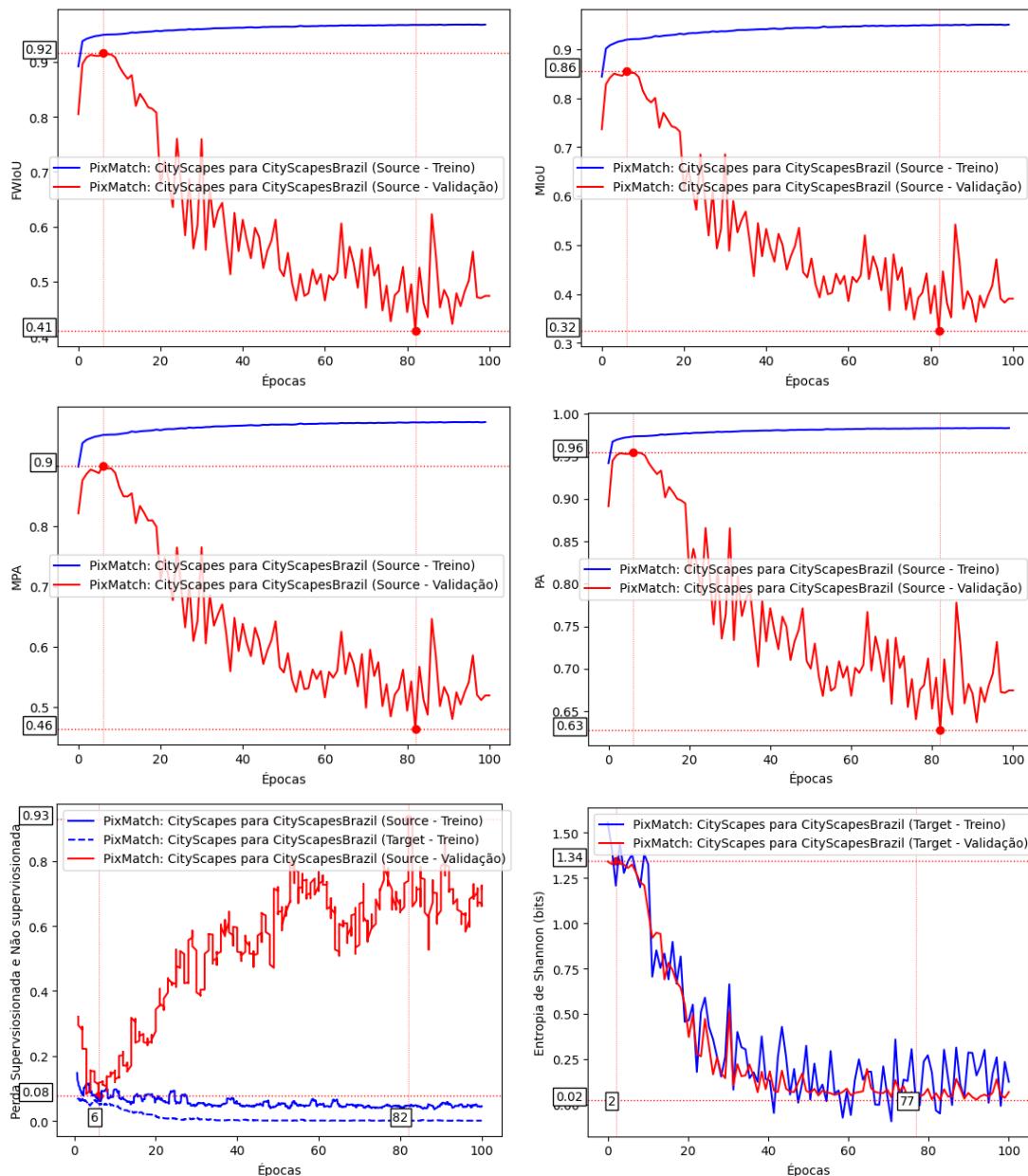
Fonte: Elaborada pelo autor.

Figura 51 – Exemplos de predição ao longo das épocas para o modelo PixMatch: GTA5 para CityScapes (19 classes), com amostras a cada 20 épocas, primeira linha imagem original, segunda linha rótulo verdadeira, demais linhas são as previsões



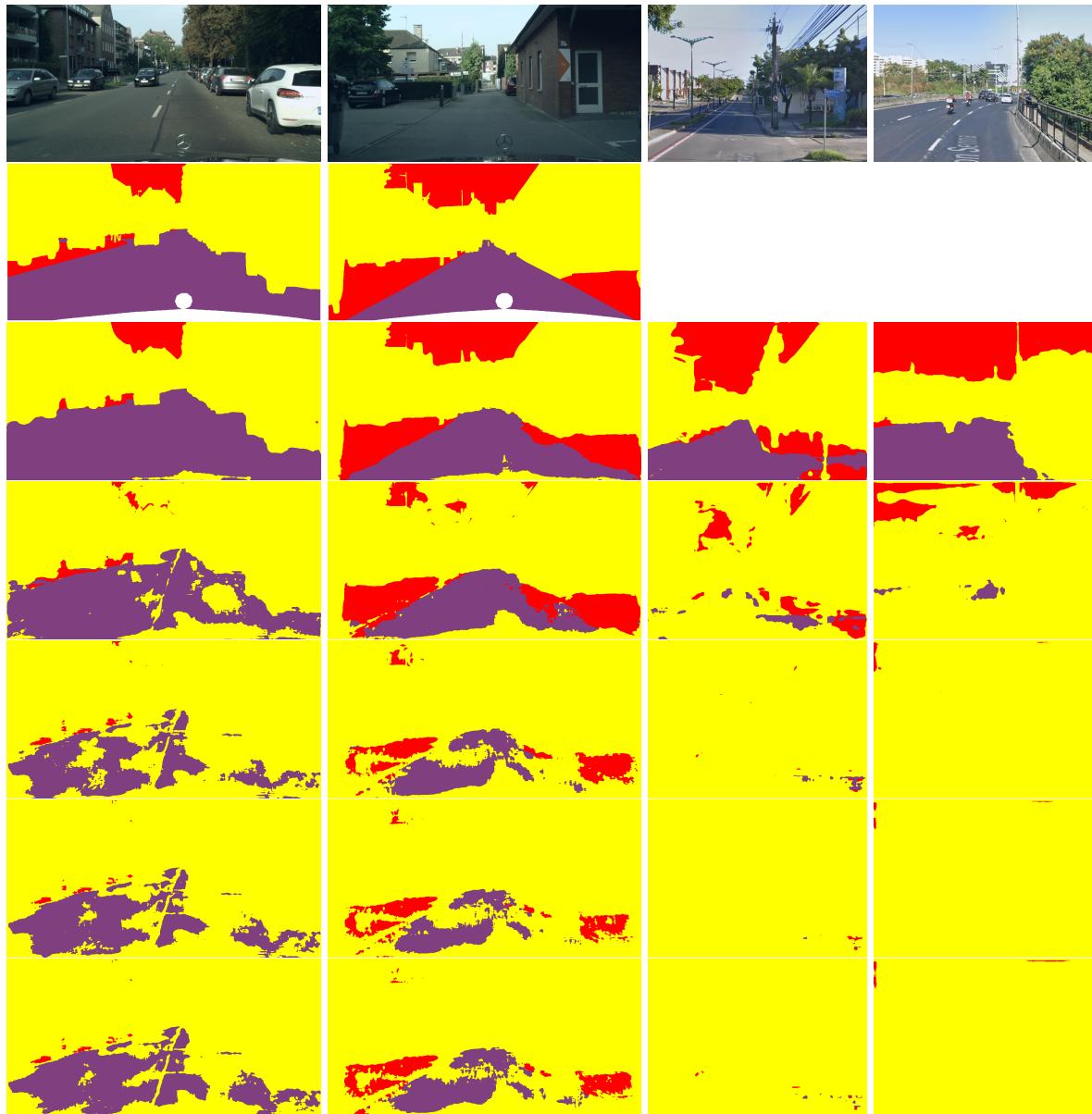
Fonte: Elaborada pelo autor.

Figura 52 – Métricas Padrão - PixMatch: CityScape para CityScapesBrazil



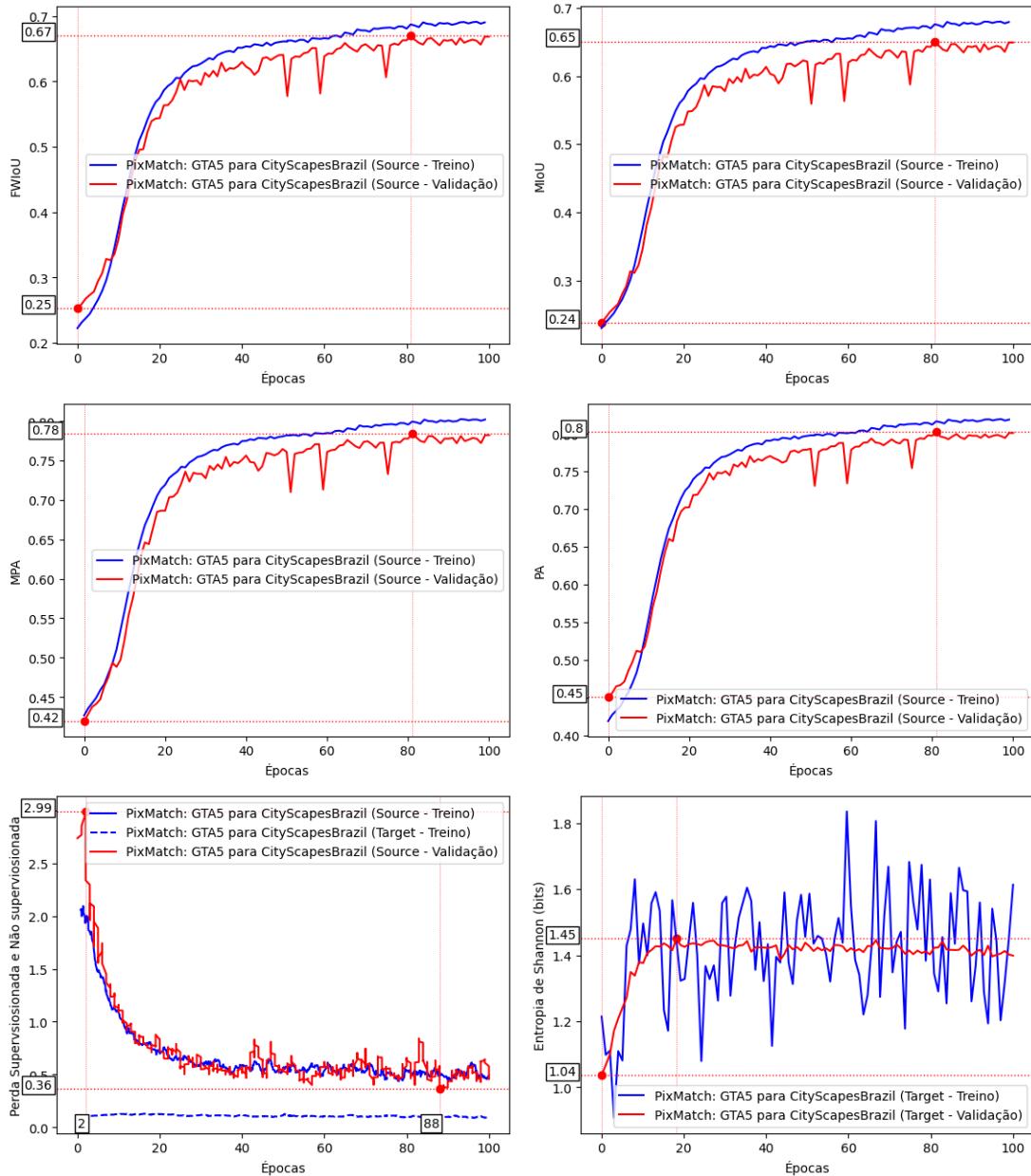
Fonte: Elaborada pelo autor.

Figura 53 – Exemplos de predição ao longo das épocas para o modelo PixMatch: CityScape para CityScapesBrazil, com amostras a cada 20 épocas, primeira linha imagem original, segunda linha rótulo verdadeira, demais linhas são as previsões



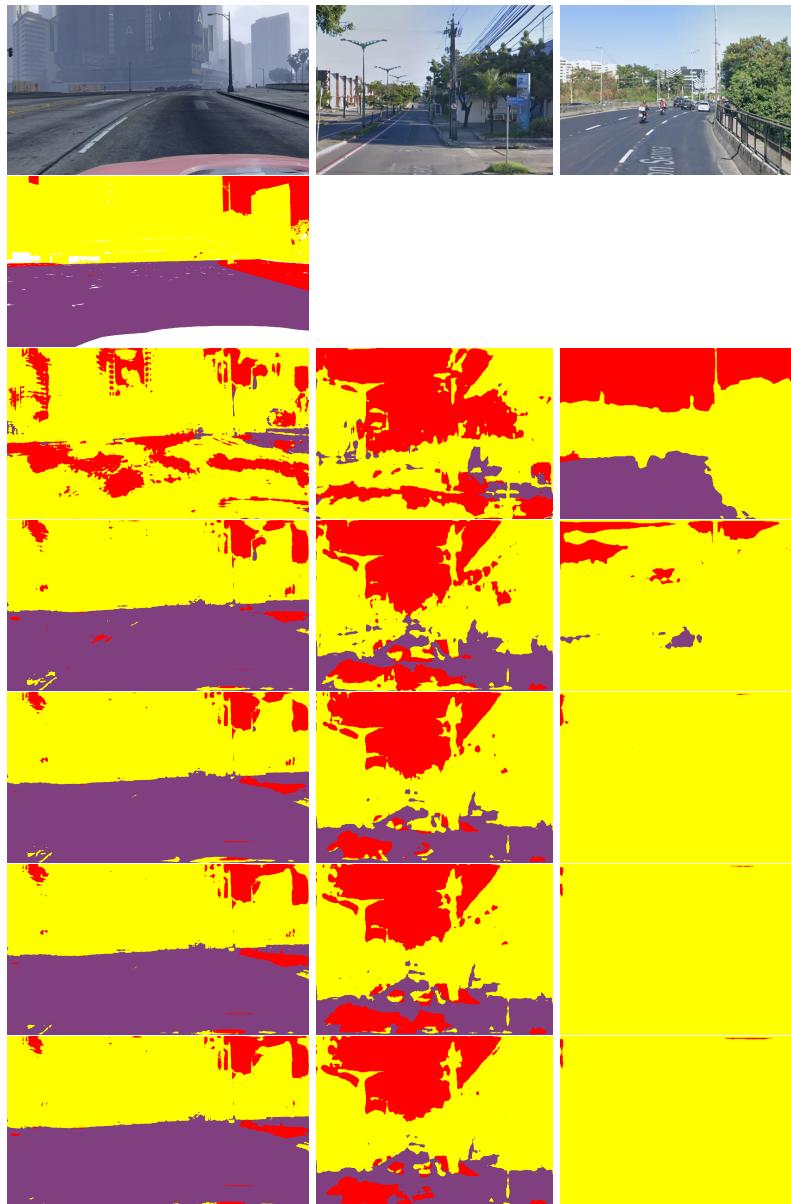
Fonte: Elaborada pelo autor.

Figura 54 – Métricas Padrão - PixMatch: GTA5 para CityScapesBrazil



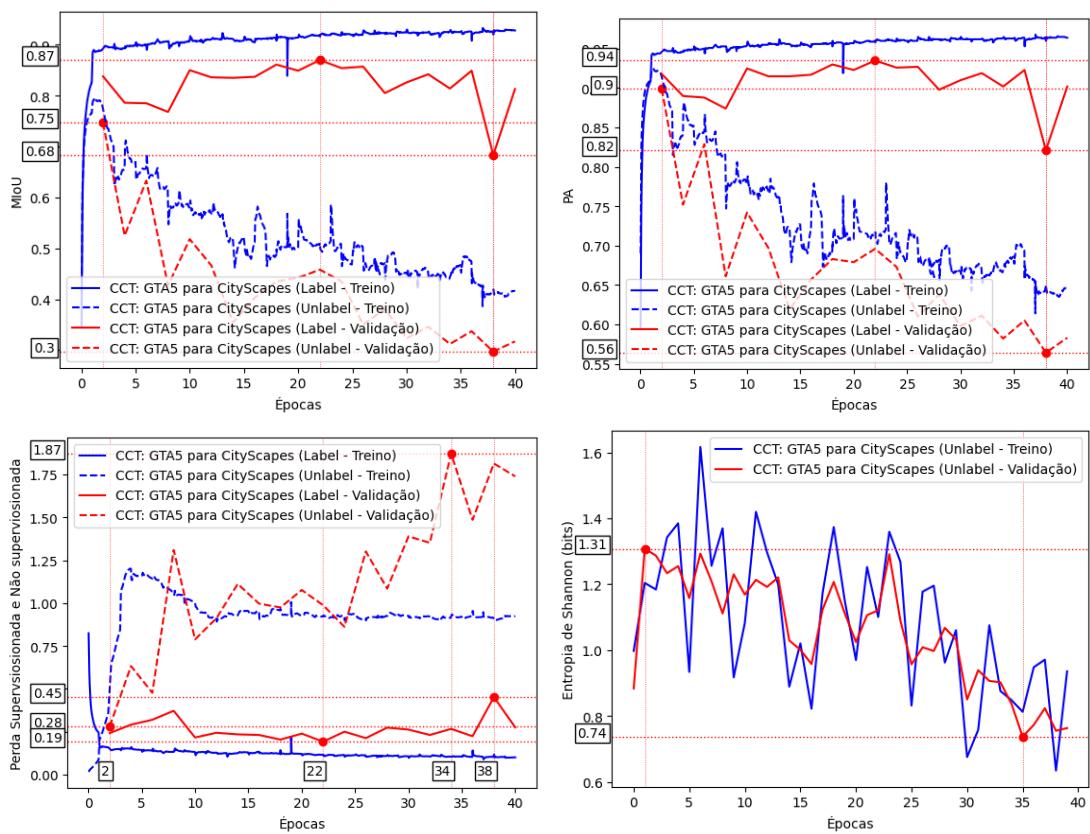
Fonte: Elaborada pelo autor.

Figura 55 – Exemplos de predição ao longo das épocas para o modelo PixMatch: GTA5 para CityScapesBrazil, com amostras a cada 20 épocas, primeira linha imagem original, segunda linha rótulo verdadeira, demais linhas são as previsões



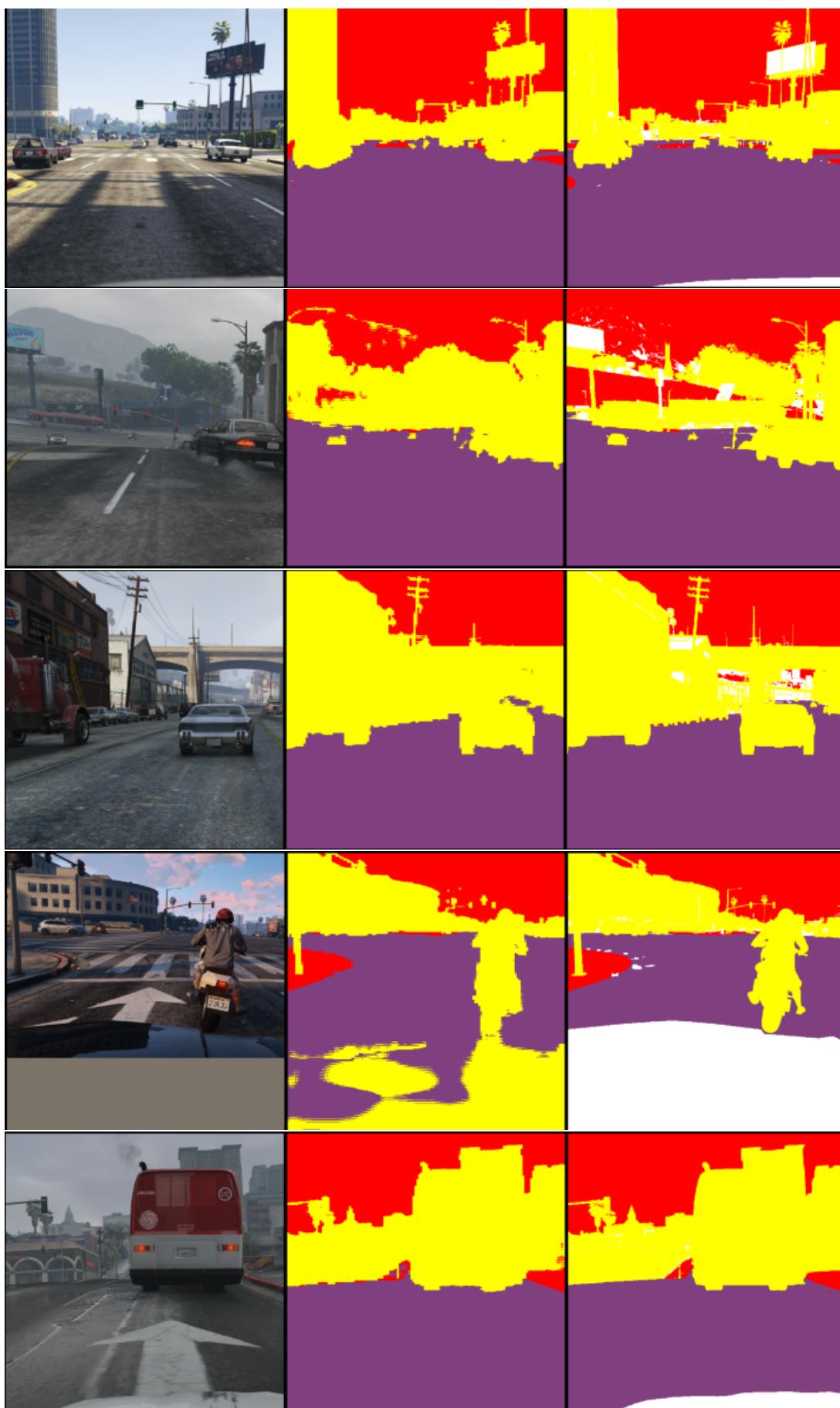
Fonte: Elaborada pelo autor.

Figura 56 – Métricas Padrão - CCT: GTA5 para CityScapes



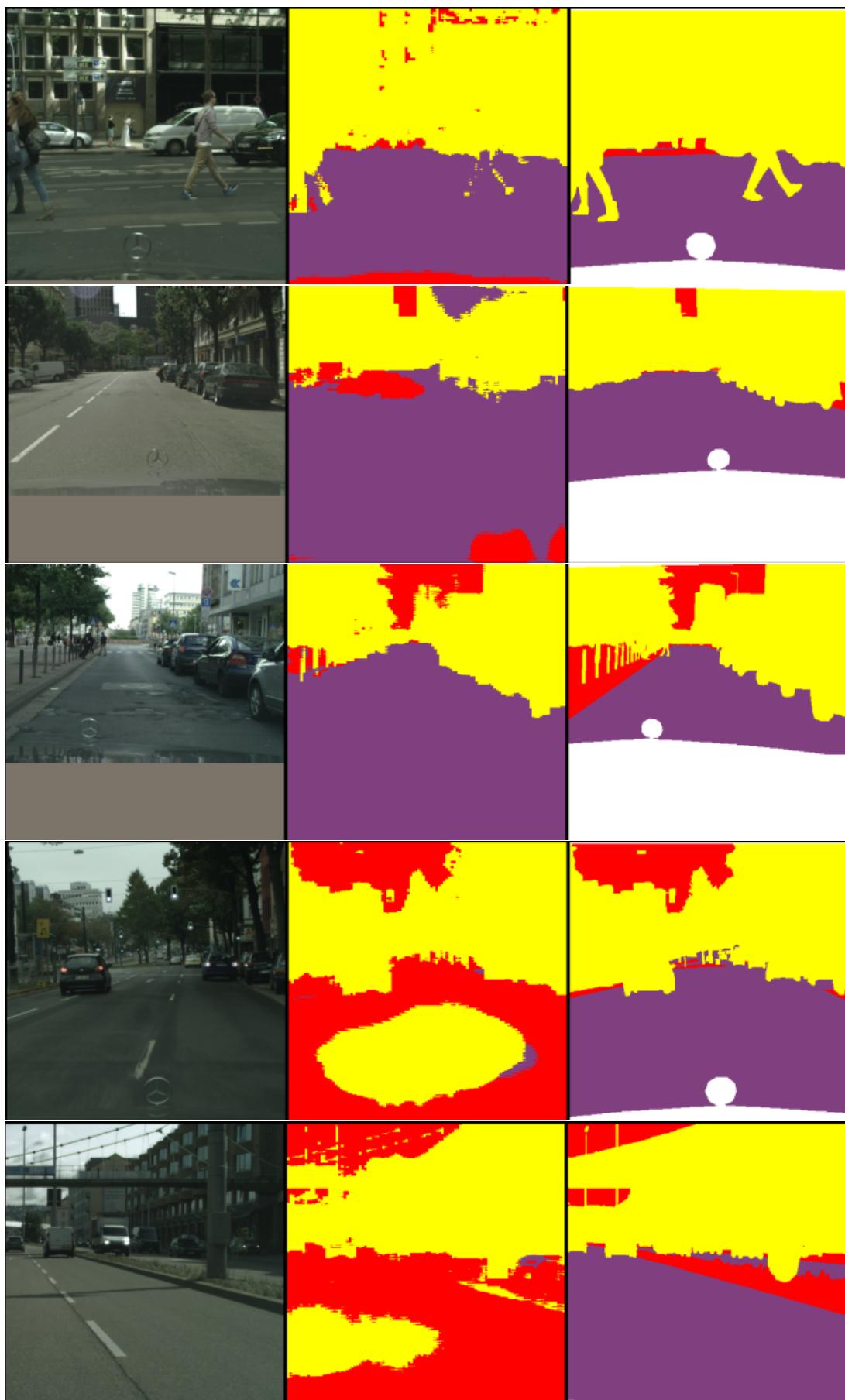
Fonte: Elaborada pelo autor.

Figura 57 – Exemplos de predição ao longo das épocas para o modelo CCT: GTA5 para CityScapes, conjunto GTA5, com amostras a cada 10 épocas, primeira coluna imagem, segunda coluna predição e terceira coluna rótulo verdadeiro



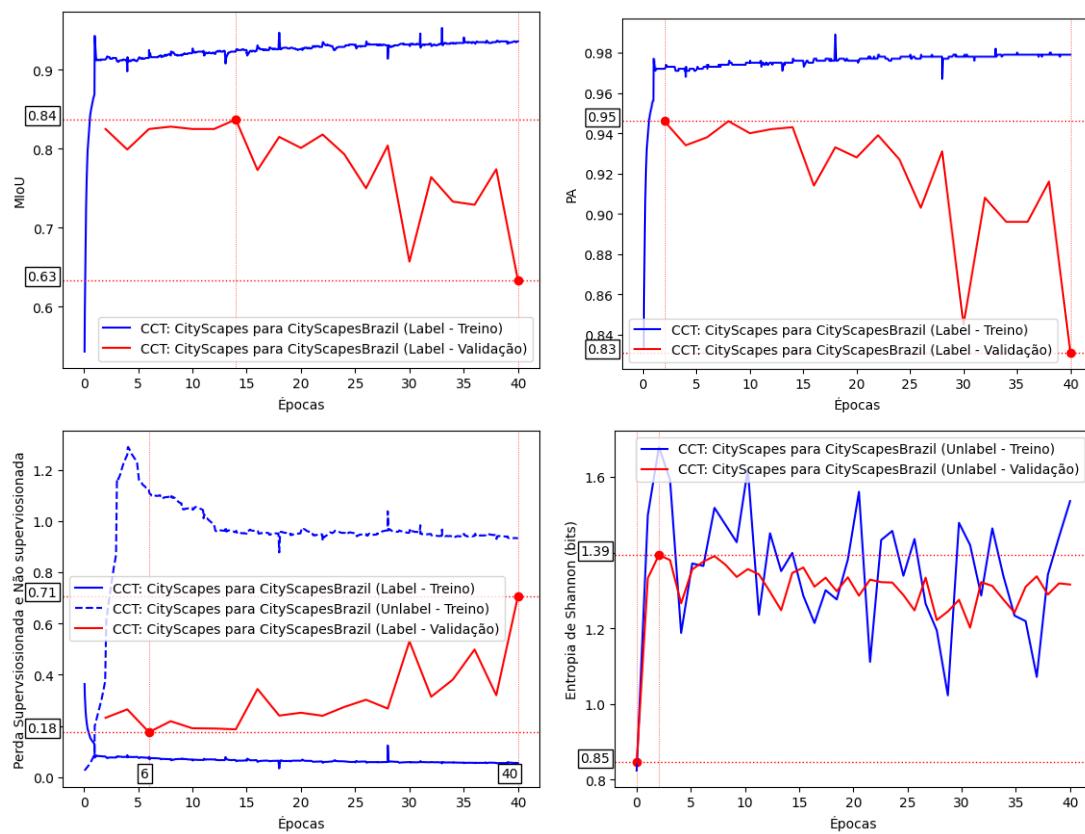
Fonte: Elaborada pelo autor.

Figura 58 – Exemplos de predição ao longo das épocas para o modelo CCT: GTA5 para CityScapes, conjunto CityScapes, com amostras a cada 10 épocas, primeira coluna imagem, segunda coluna predição e terceira coluna rótulo verdadeiro



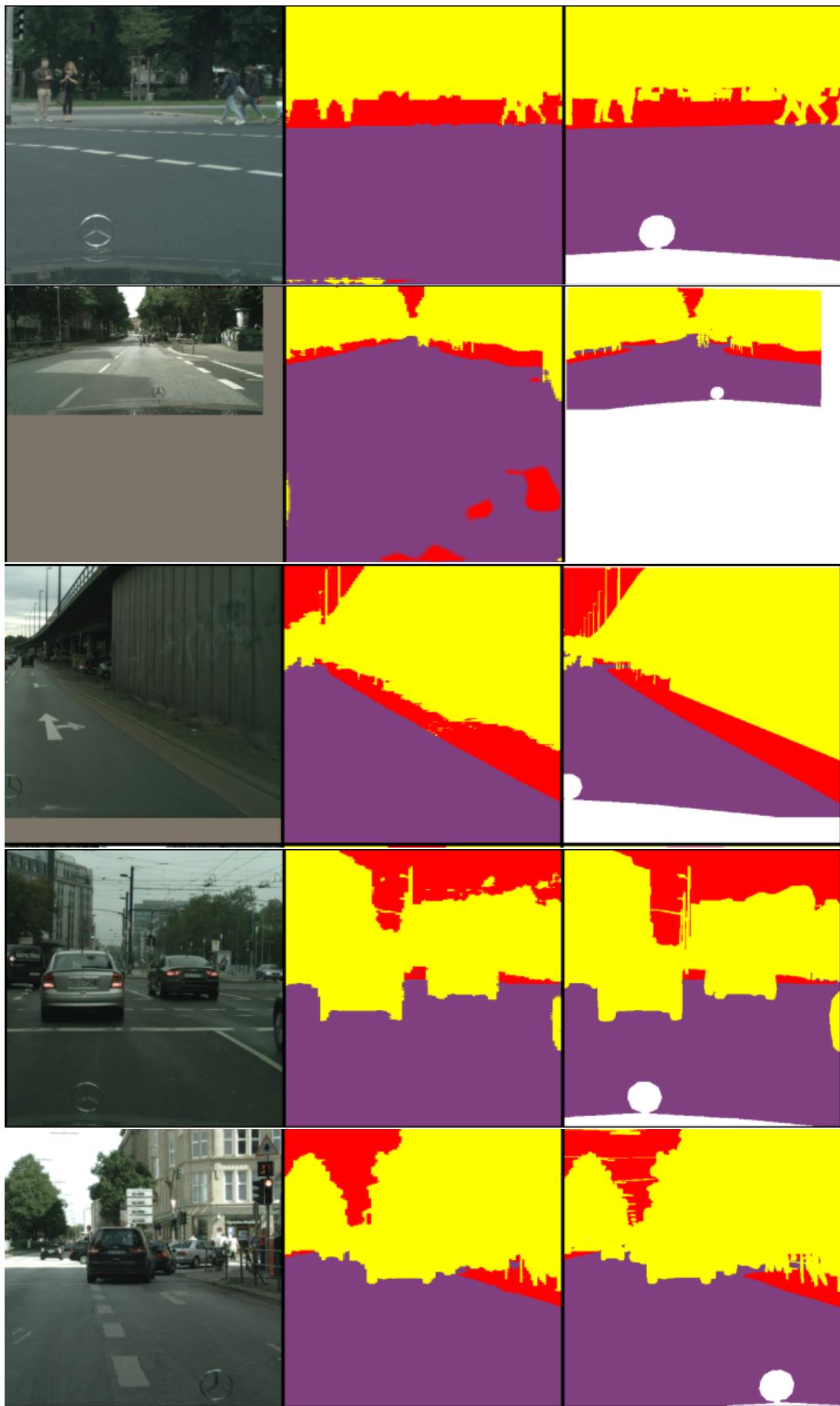
Fonte: Elaborada pelo autor.

Figura 59 – Métricas Padrão - CCT: CityScapes para CityScapesBrazil



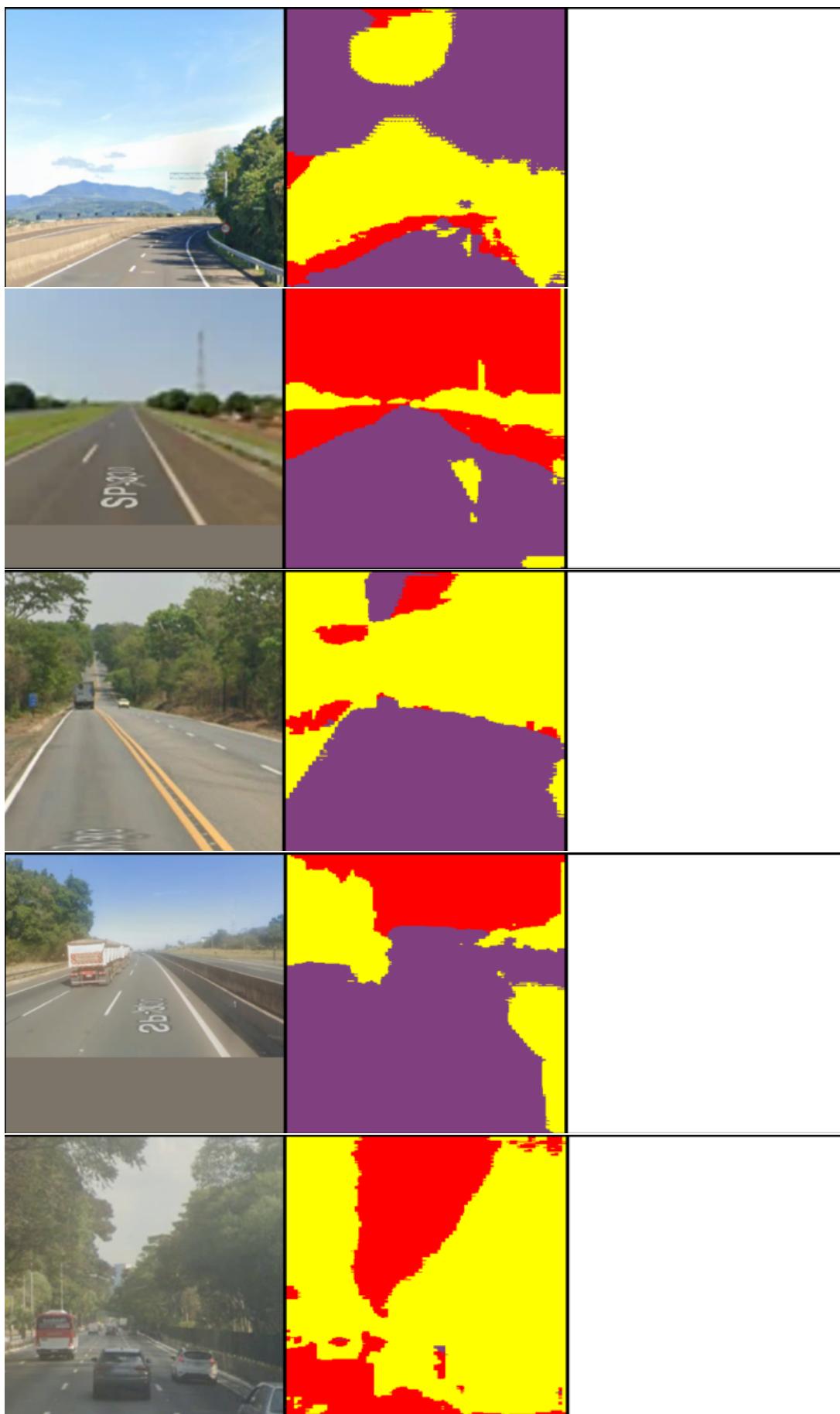
Fonte: Elaborada pelo autor.

Figura 60 – Exemplos de predição ao longo das épocas para o modelo CCT: CityScapes para CityScapesBrazil, conjunto CityScapes, com amostras a cada 10 épocas, primeira coluna imagem, segunda coluna predição e terceira coluna rótulo verdadeiro



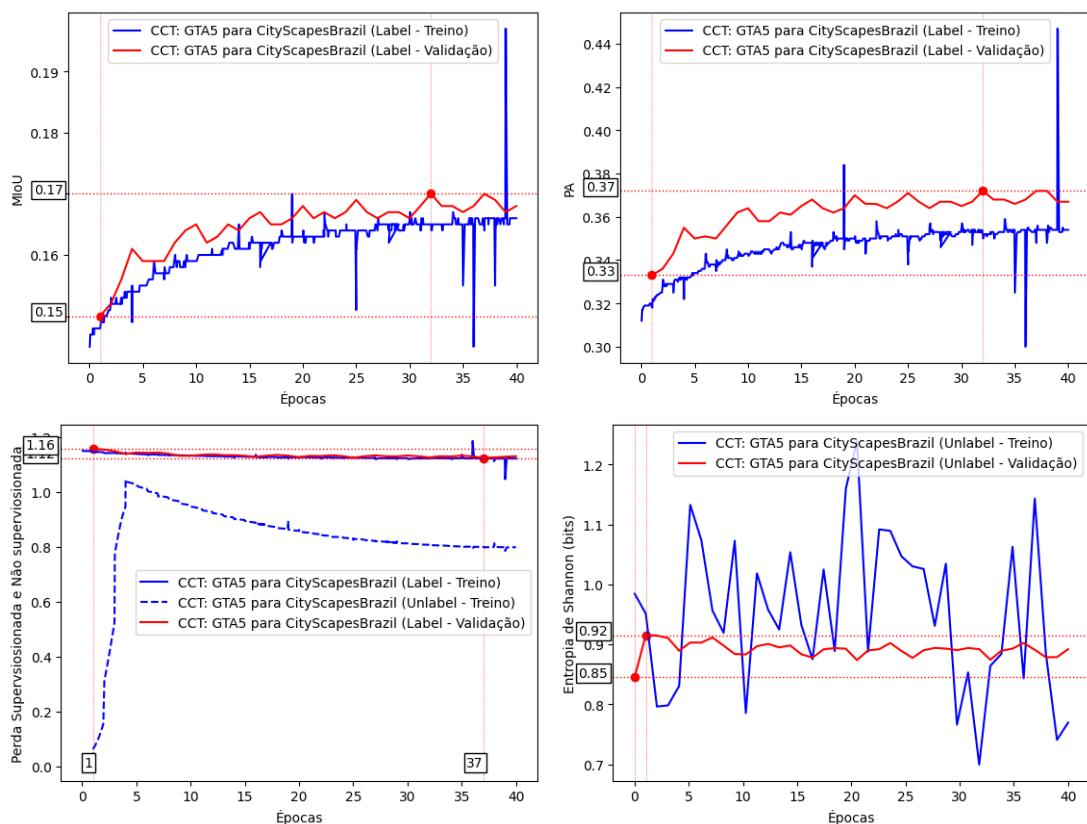
Fonte: Elaborada pelo autor.

Figura 61 – Exemplos de predição ao longo das épocas para o modelo CCT: CityScapes para CityScapesBrazil, conjunto CityScapesBrazil, com amostras a cada 10 épocas, primeira coluna imagem, segunda coluna predição e terceira coluna rótulo verdadeiro



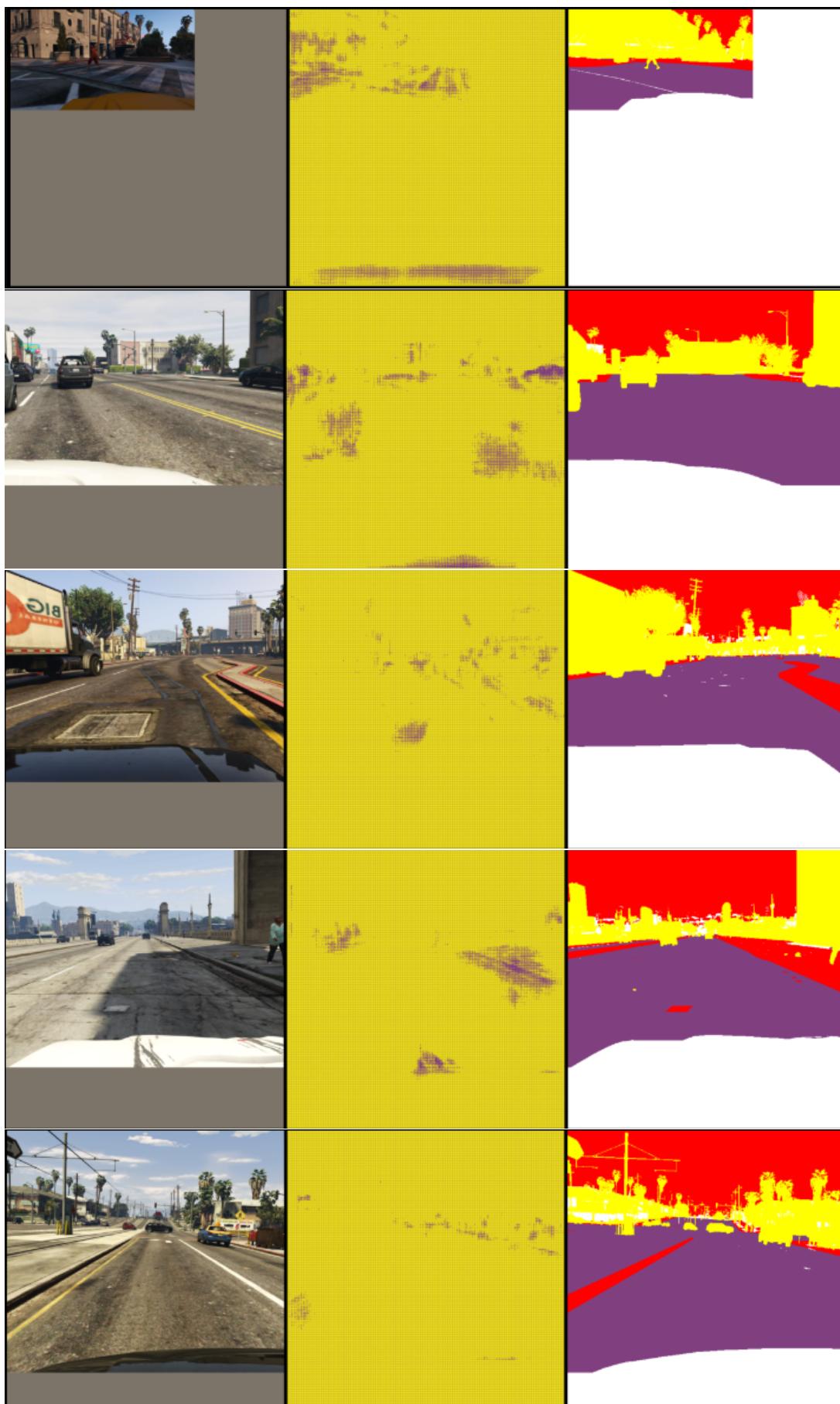
Fonte: Elaborada pelo autor.

Figura 62 – Métricas Padrão - CCT: GTA5 para CityScapesBrazil



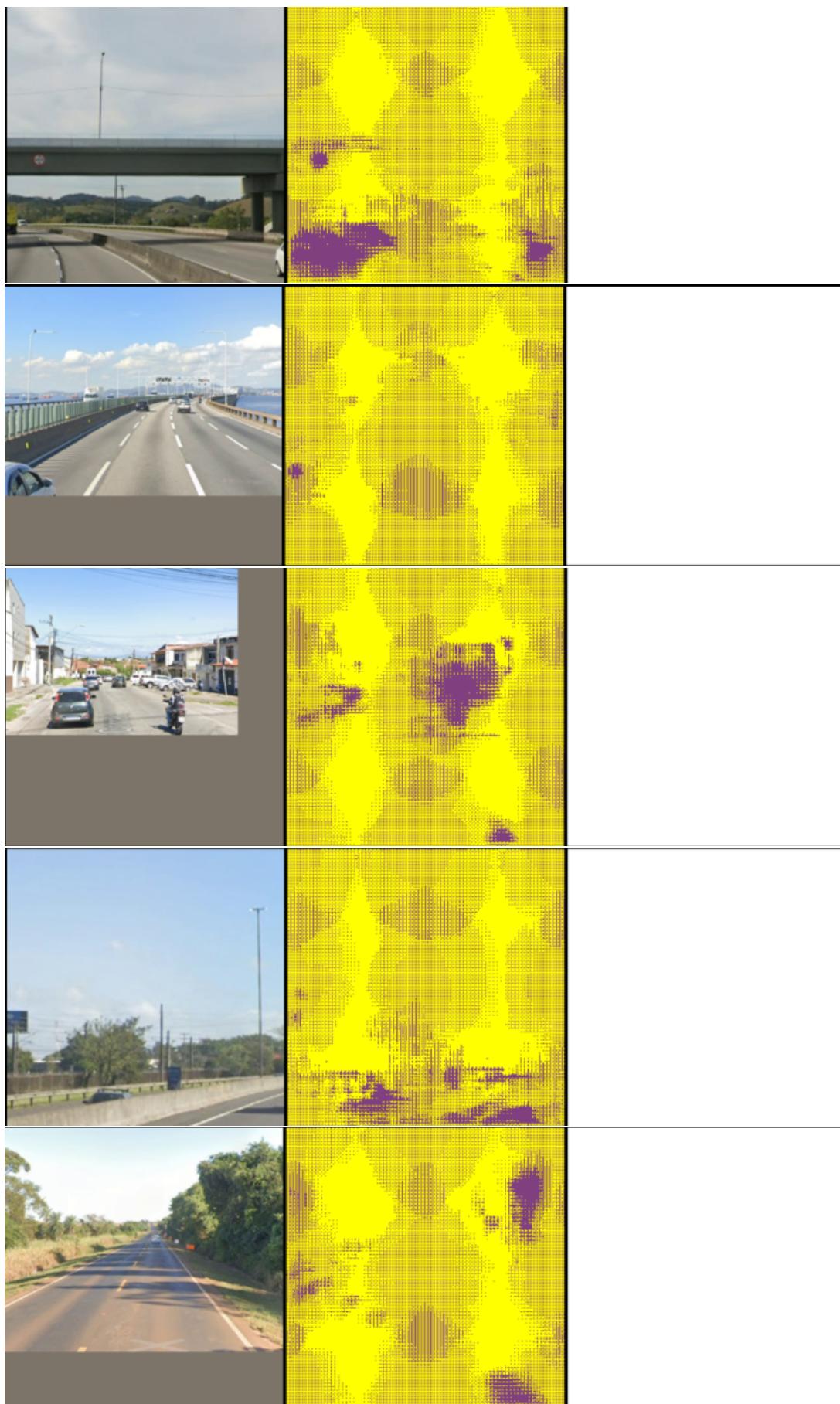
Fonte: Elaborada pelo autor.

Figura 63 – Exemplos de predição ao longo das épocas para o modelo CCT: GTA5 para CityScapesBrazil, conjunto GTA5, com amostras a cada 10 épocas, primeira coluna imagem, segunda coluna predição e terceira coluna rótulo verdadeiro



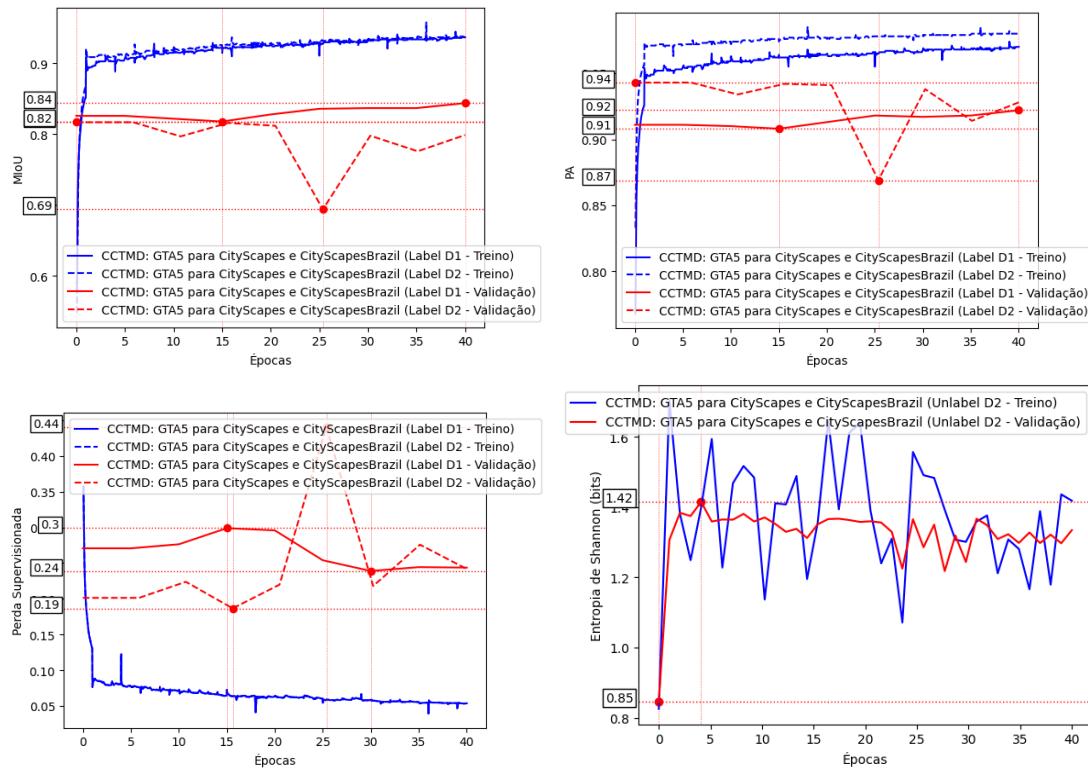
Fonte: Elaborada pelo autor.

Figura 64 – Exemplos de predição ao longo das épocas para o modelo CCT: GTA5 para CityScapesBrazil, conjunto CityScapesBrazil, com amostras a cada 10 épocas, primeira coluna imagem, segunda coluna predição e terceira coluna rótulo verdadeiro



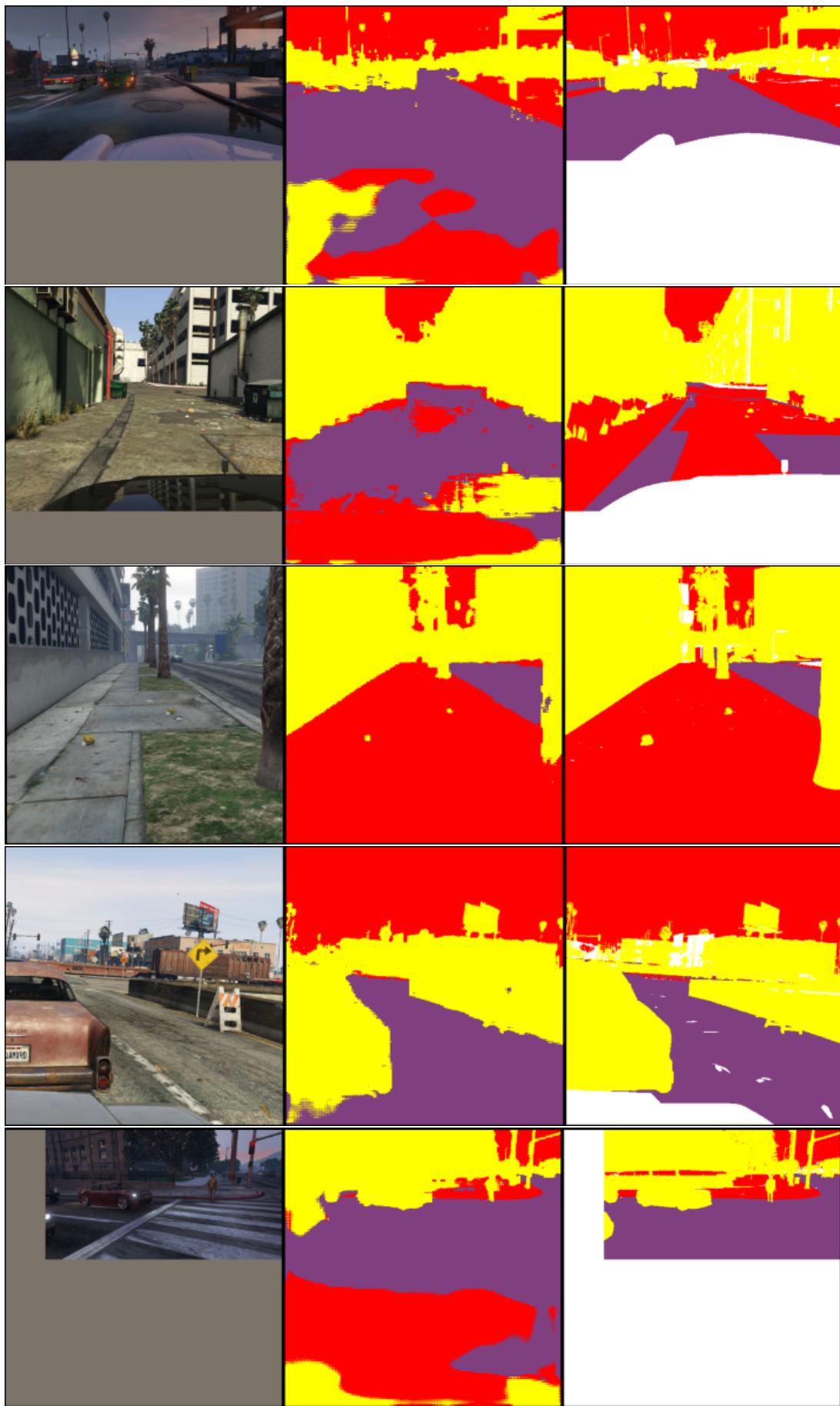
Fonte: Elaborada pelo autor.

Figura 65 – Métricas Padrão - CCTMD: GTA5 para CityScapes e CityScapesBrazil



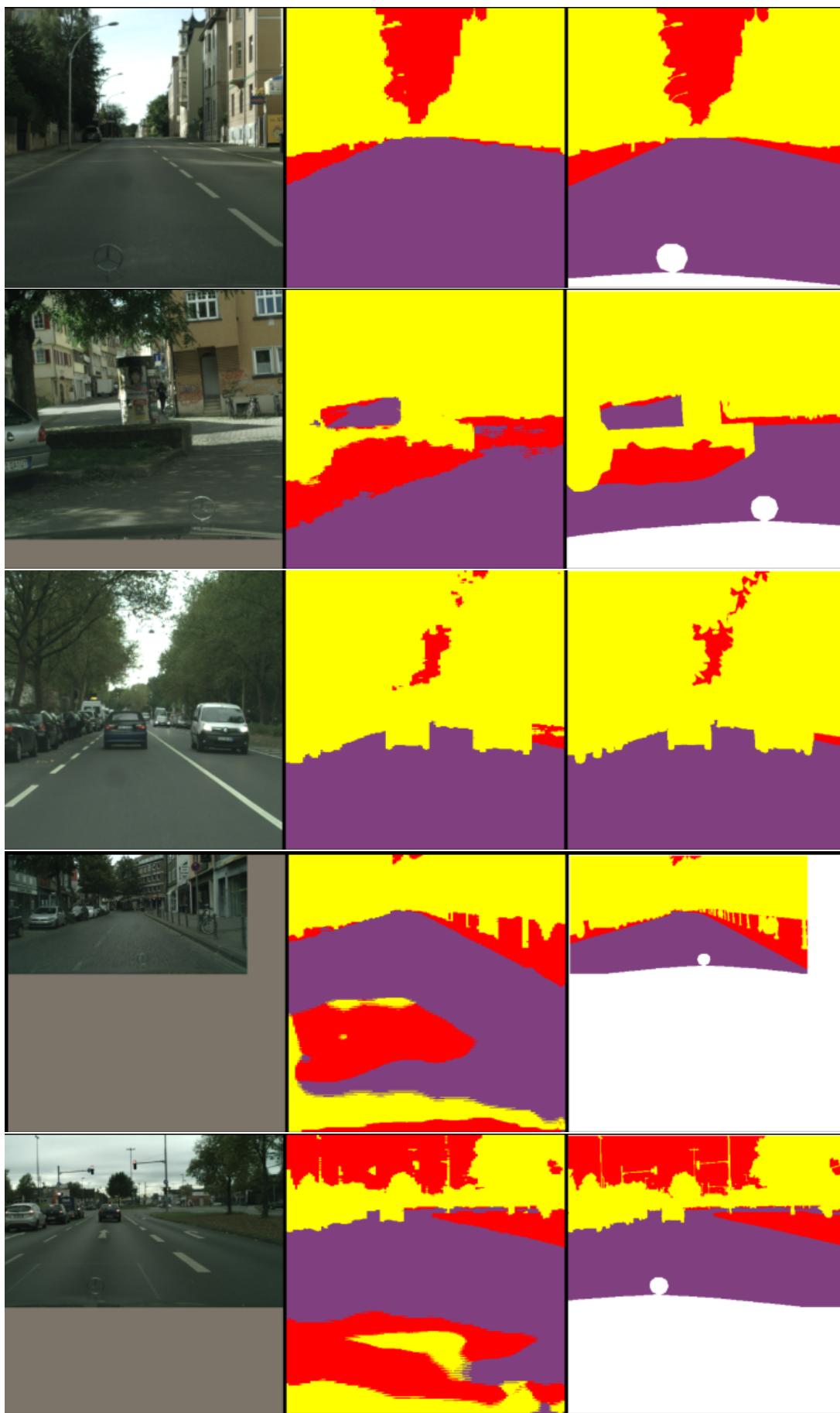
Fonte: Elaborada pelo autor.

Figura 66 – Exemplos de predição ao longo das épocas para o modelo CCTMD: GTA5 para CityScapes e CityScapesBrazil, conjunto GTA5, com amostras a cada 10 épocas, primeira coluna imagem, segunda coluna predição e terceira coluna rótulo verdadeiro



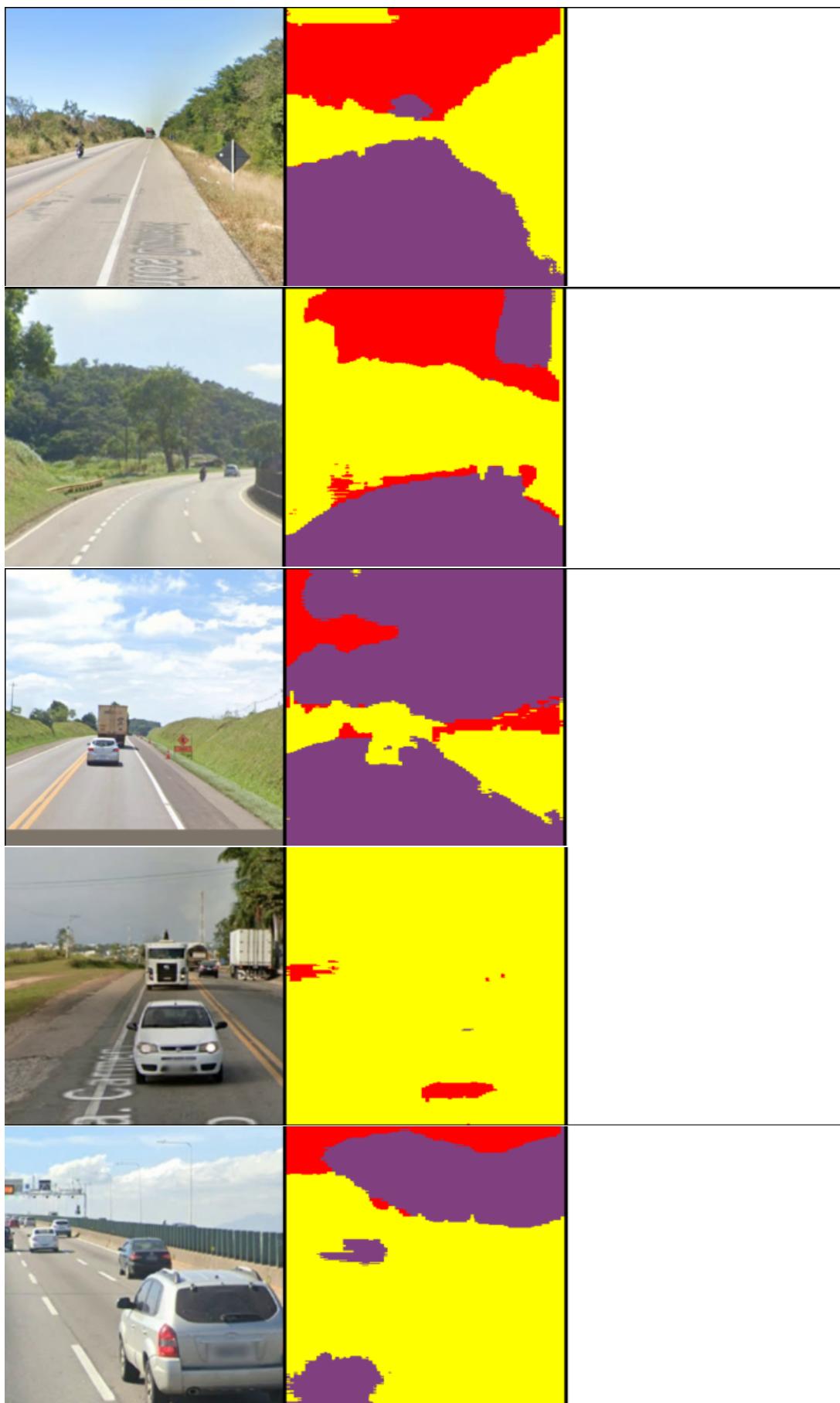
Fonte: Elaborada pelo autor.

Figura 67 – Exemplos de predição ao longo das épocas para o modelo CCTMD: GTA5 para CityScapes e CityScapesBrazil, conjunto CityScapes, com amostras a cada 10 épocas, primeira coluna imagem, segunda coluna predição e terceira coluna rótulo verdadeiro



Fonte: Elaborada pelo autor.

Figura 68 – Exemplos de predição ao longo das épocas para o modelo CCTMD: GTA5 para CityScapes e CityScapesBrazil, conjunto CityScapesBrazil, com amostras a cada 10 épocas, primeira coluna imagem, segunda coluna predição e terceira coluna rótulo verdadeiro



Fonte: Elaborada pelo autor.