

# **Comparative Study of Semantic Segmentation Architectures for ADAS**

Evaluating U-Net, DeepLabV3+, and SegFormer on the Cityscapes Dataset

**Presented by Vittorio Albertin and Lorenzo Bacchini**

20 February 2026

# Introduction & Problem Statement

## > Objective

**Pixel-level classification** of urban street scenes to distinguish drivable surfaces from obstacles and other road users

## > Dataset

**Cityscapes** benchmark featuring 19 semantic classes and high-resolution (1024x2048) images.

## > Constraints

Scaling high-resolution segmentation research down to consumer-grade hardware (Single 8GB GPU).



# Dataset

## Diverse Urban Data

Captured across 50 cities in Germany and neighboring countries, providing varied weather and architectural contexts.

## High-Volume Annotation

Contains **5,000 fine pixel-level annotations** and 20,000 coarse annotations for robust training.

## 19 Semantic Classes

Evaluated on critical classes including Road, Sidewalk, Pedestrian, Traffic Light, Signage, and Vehicle.

GROUP	SPECIFIC CLASSES
 FLAT	road, sidewalk, parking <sup>+</sup> , rail track <sup>+</sup>
 HUMAN	person, rider
 VEHICLE	car, truck, bus, on rails, motorcycle, bicycle, caravan <sup>+</sup> , trailer <sup>+</sup>
 CONSTRUCTION	building, wall, fence, guard rail <sup>+</sup> , bridge <sup>+</sup> , tunnel <sup>+</sup>
 OBJECT	pole, pole group <sup>+</sup> , traffic sign, traffic light
 NATURE	vegetation, terrain
 SKY	sky
 VOID	ground <sup>+</sup> , dynamic <sup>+</sup> , static <sup>+</sup>

The + denotes the classes excluded from the metric evaluation

# Motivation



## Architectural Evolution

Moving from classic CNNs (**U-Net**) to more advanced CNNs (**DeepLabV3+**) and modern Transformers (**SegFormer**).



## Class Imbalance

Addressing the challenge where **safety-critical classes** like *Traffic Lights* and *Pedestrians* are **rare** compared to *Road* or *Building*.



## Resource Engineering

Proving methodological rigor is possible without massive computational capabilities, focusing on **efficiency**.

# Class Imbalance

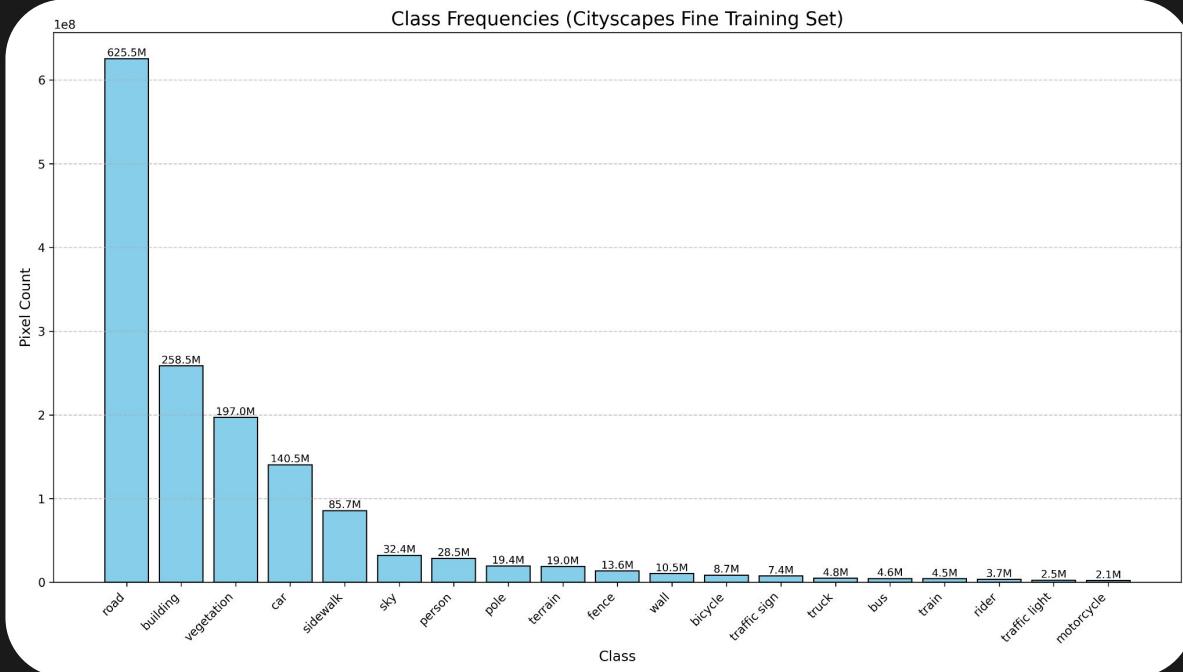


Image 1: Class frequencies of Cityscapes Dataset  
(per pixel count)



Image 2: Class weights used for **Cross Entropy** Loss function as

$$w_{class} = \frac{1}{\ln(1.02 + p_{class})}$$

# Class Imbalance

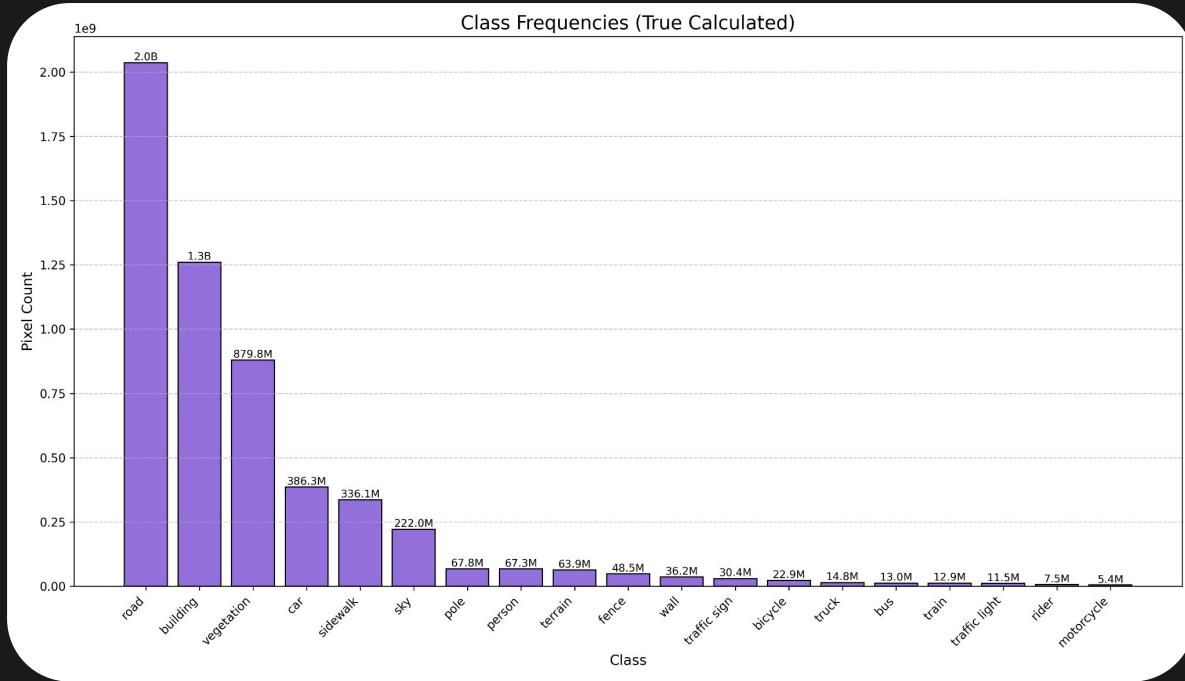


Image 1: True value of class frequencies

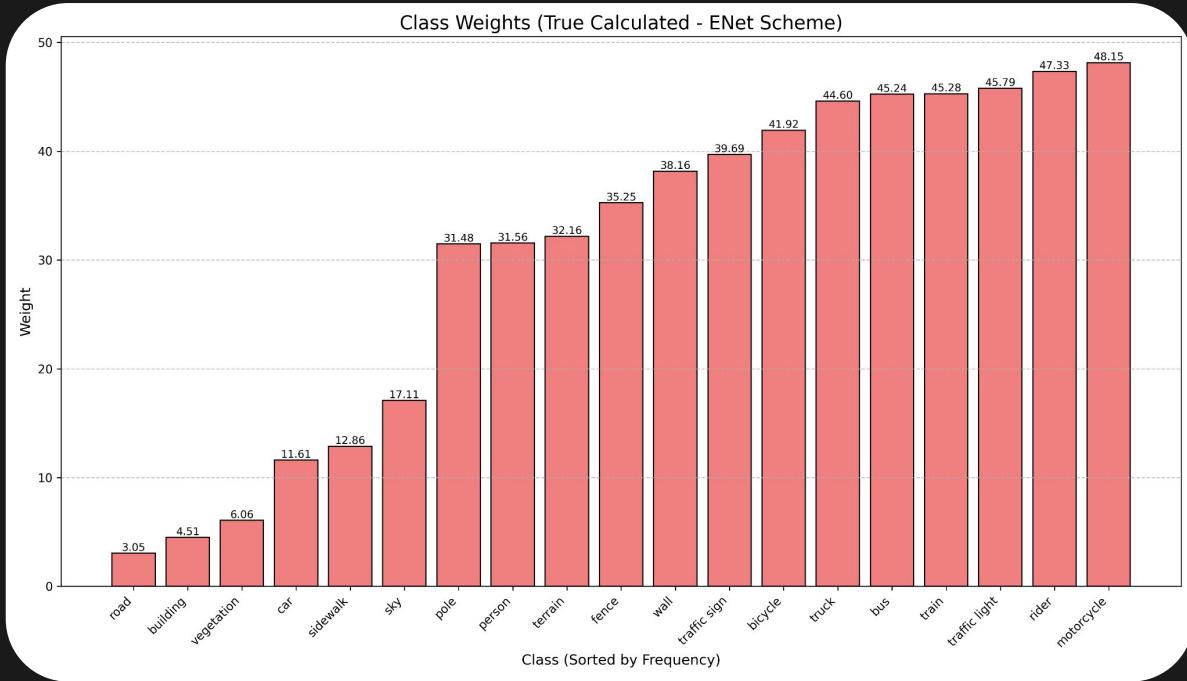


Image 2: True values of class weights CE loss function

# Resource Engineering for 8GB VRAM



## Mixed Precision

Used **Automatic Mixed Precision (AMP)** with float16/float32, reducing VRAM usage considerably.



## Gradient Accumulation

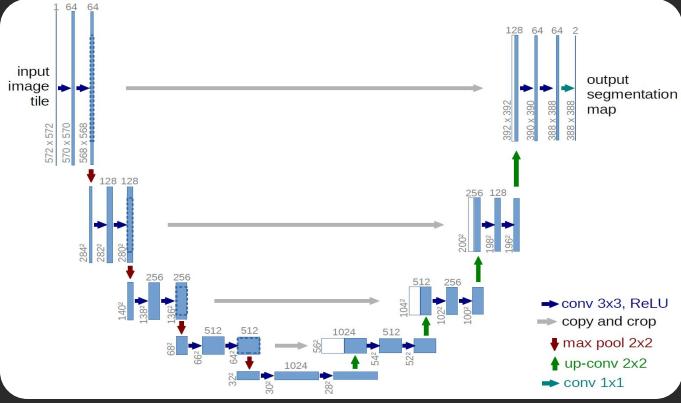
Accumulated gradients over **4 steps** to simulate an **Effective Batch Size of 4**, bypassing hardware limits.



## Resolution strategy

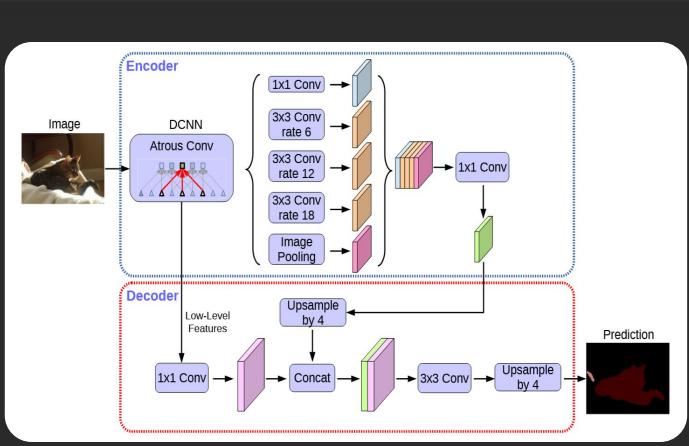
Trained on random **(512 x 1024) crops** to fit memory while maintaining high-resolution feature learning.

# Methodology: Architecture Overview



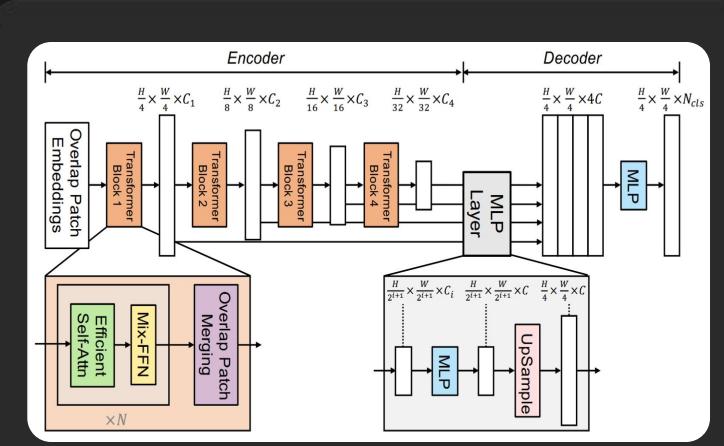
## U-Net

Baseline **CNN Encoder-Decoder**  
utilizing skip connections for feature  
localization.



## DeepLabV3+

CNN Encoder-Decoder using  
**ResNet-50** and **ASPP** (Atrous Spatial  
Pyramid Pooling).



## SegFormer

Transformer-based approach with  
hierarchical encoder (**MiT-B0**) and  
**MLP decoder**.

# U-Net

Encoder-Decoder

Skip Connections

Symmetry

- > **Modification:** Replaced Batch Normalization with **Instance Normalization** to avoid variance calculation failure.
- > **Structural Update:** Enforced **padding=1** to preserve spatial dimensions, removing cropping needs.
- > **Upsampling:** Used deterministic **bilinear upsampling** instead of transposed convolutions to prevent checkerboard artifacts and reduce computational load.

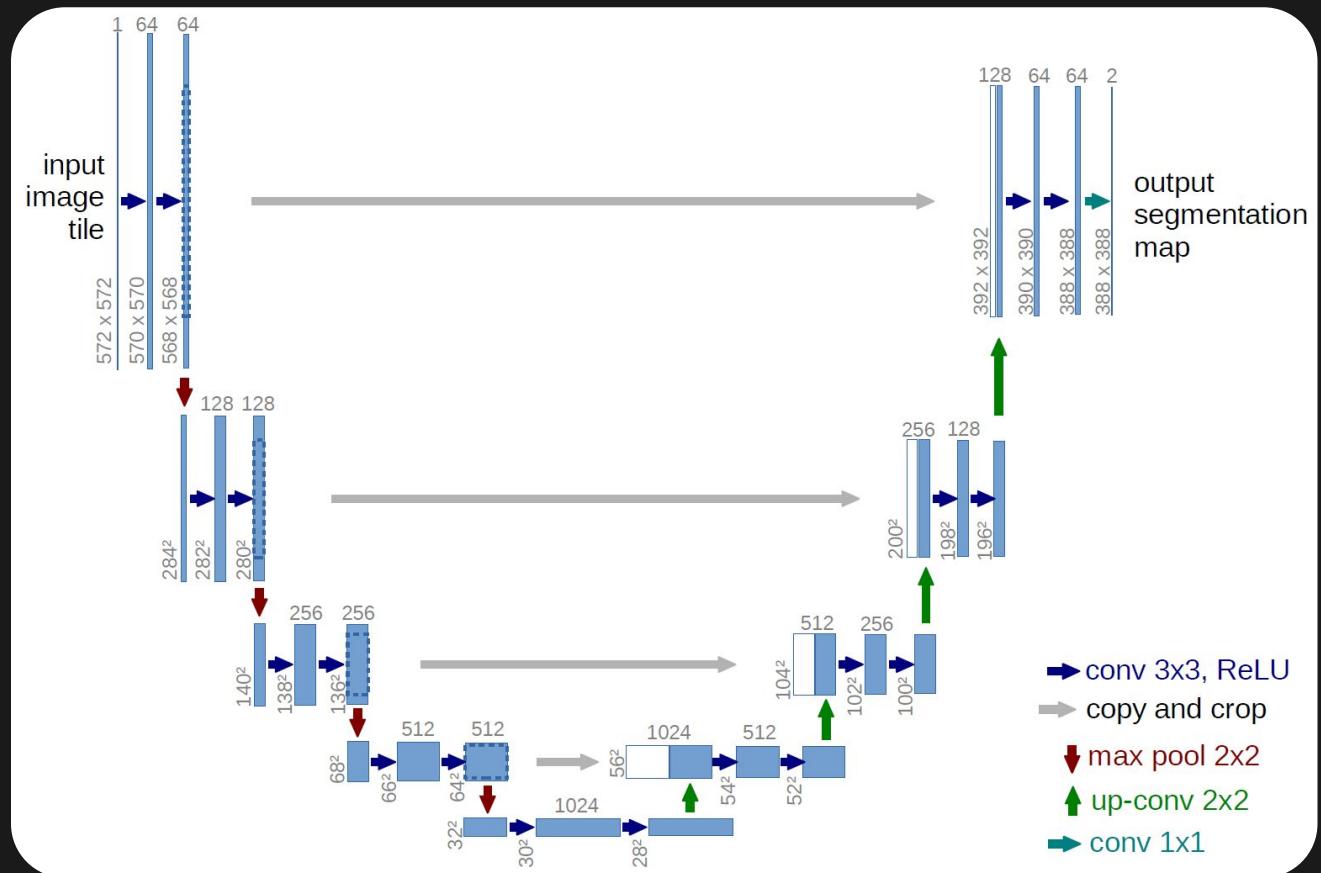
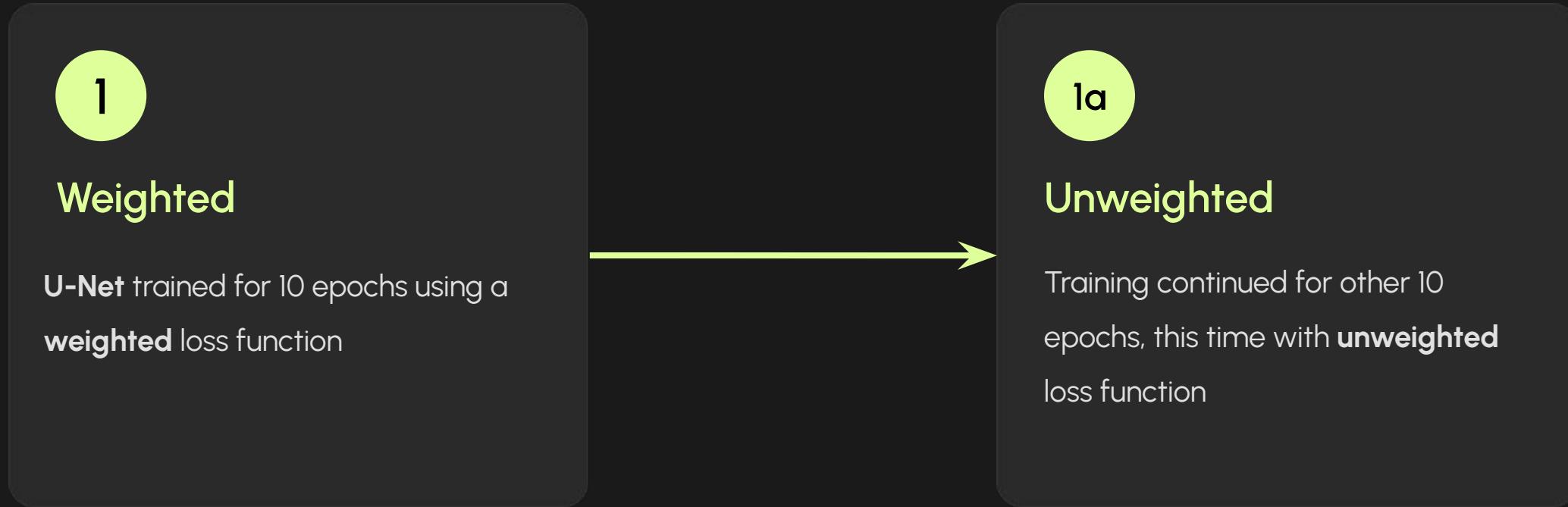


Image: <https://doi.org/10.48550/arXiv.1505.04597>

# U-Net: Training



20 epoch training

# U-Net: Training Results

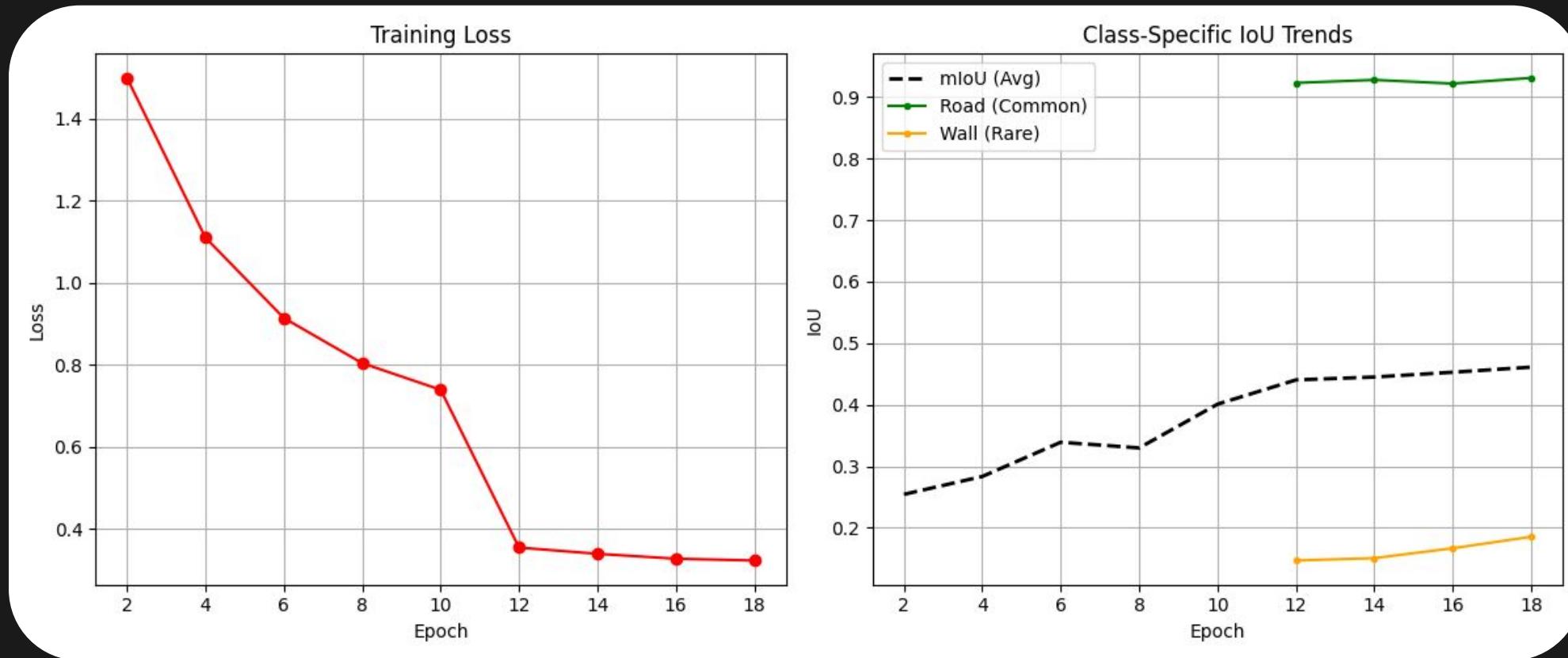


Image: Training plot of model U-Net 1a

# U-Net: Evaluation Results

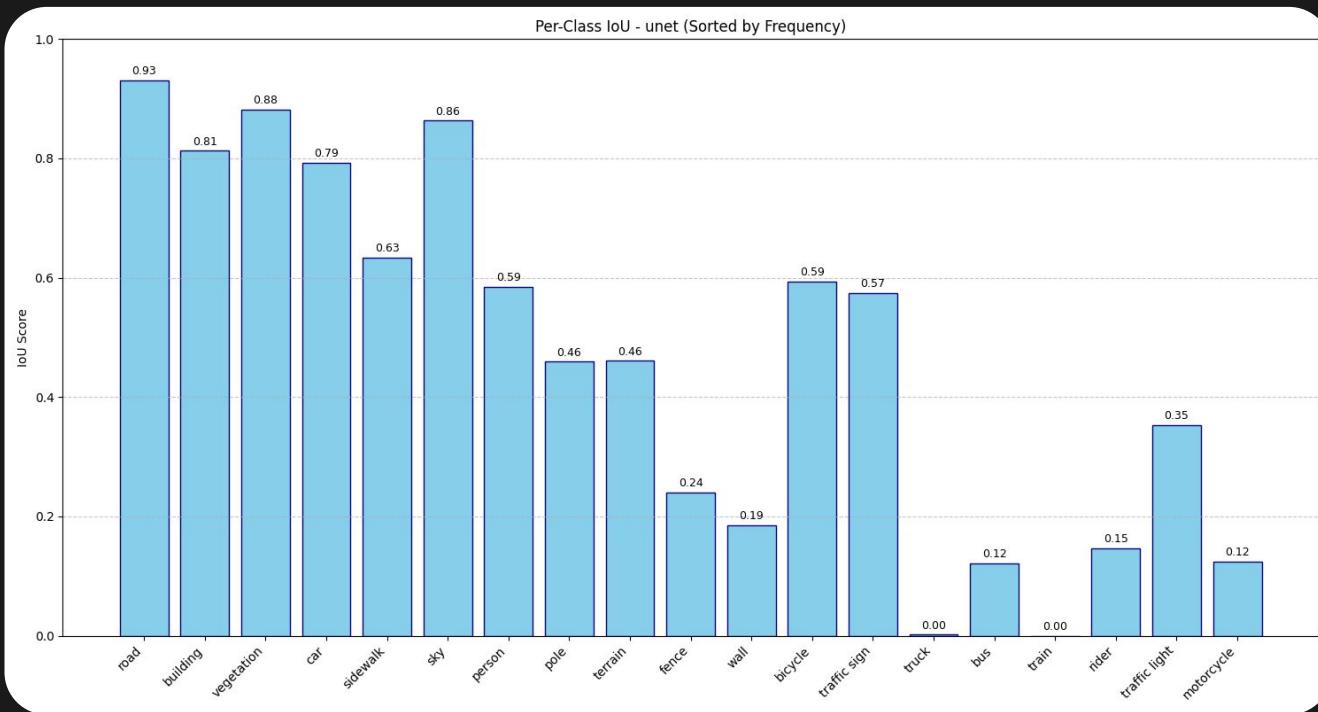


Image 1: Per-Class IoU plot of model U-Net 1a

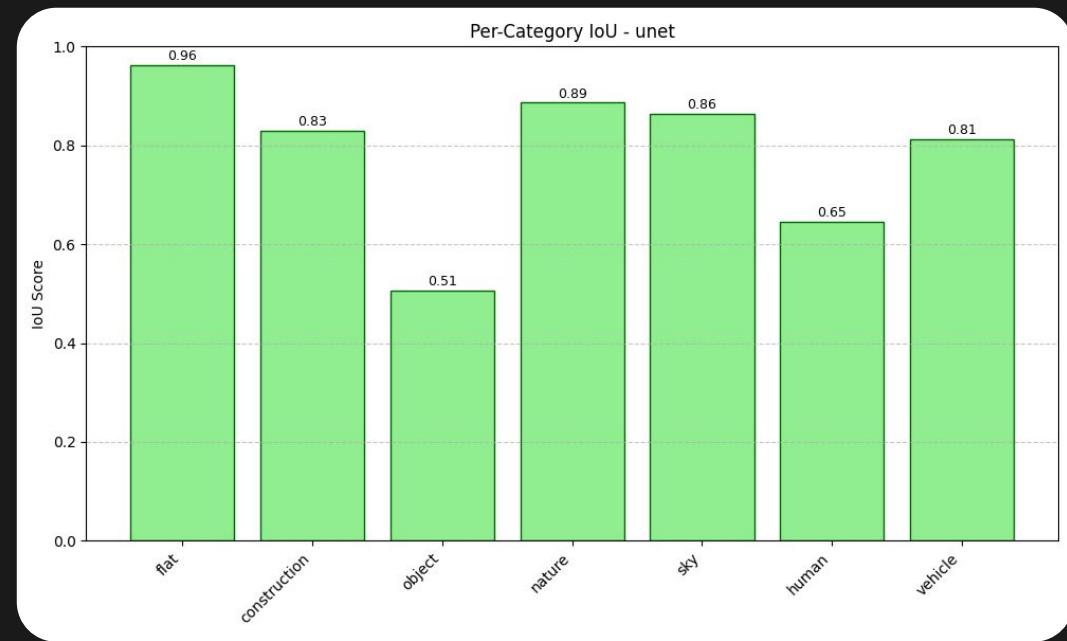


Image 2: Per-Category IoU plot of model U-Net 1a

# U-Net: Evaluation Results



Reference image: frankfurt\_000000\_001236\_leftImg8bit.png

# DeepLabV3+

Encoder-Decoder

ResNet-50

Atrous Spatial Pyramid Pooling (ASPP)

High spatial resolution

- > **Multi grid:** Used `replace_stride_with_dilation` in ResNet-50 to replace stride with atrous convolution which led us to **not** implement the multi grid logic of the original paper.
- > **Operation Optimization:** Bypassed depthwise separable convolutions in favor of **standard convolutions** for the 8GB budget.

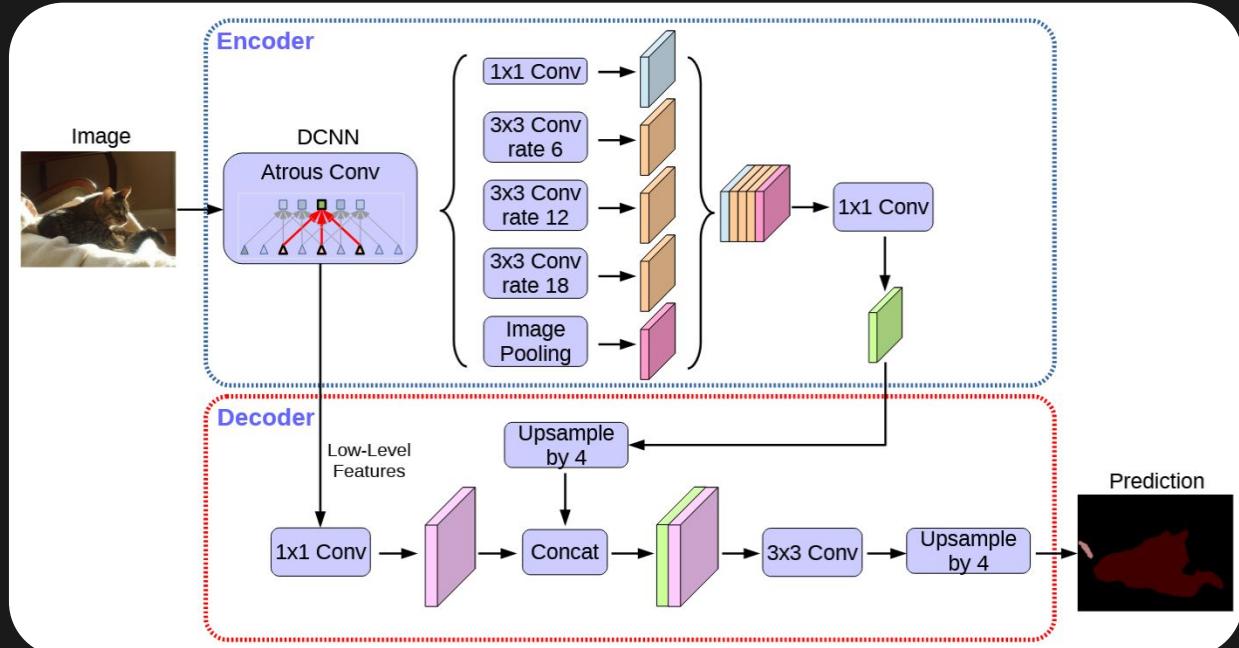


Image: <https://doi.org/10.48550/arXiv.1802.02611>

# DeepLabV3+: Training 1

1

## Unfrozen Weighted

DeepLabV3+ trained for 20 epochs  
with **unfrozen** backbone and  
**weighted** loss function

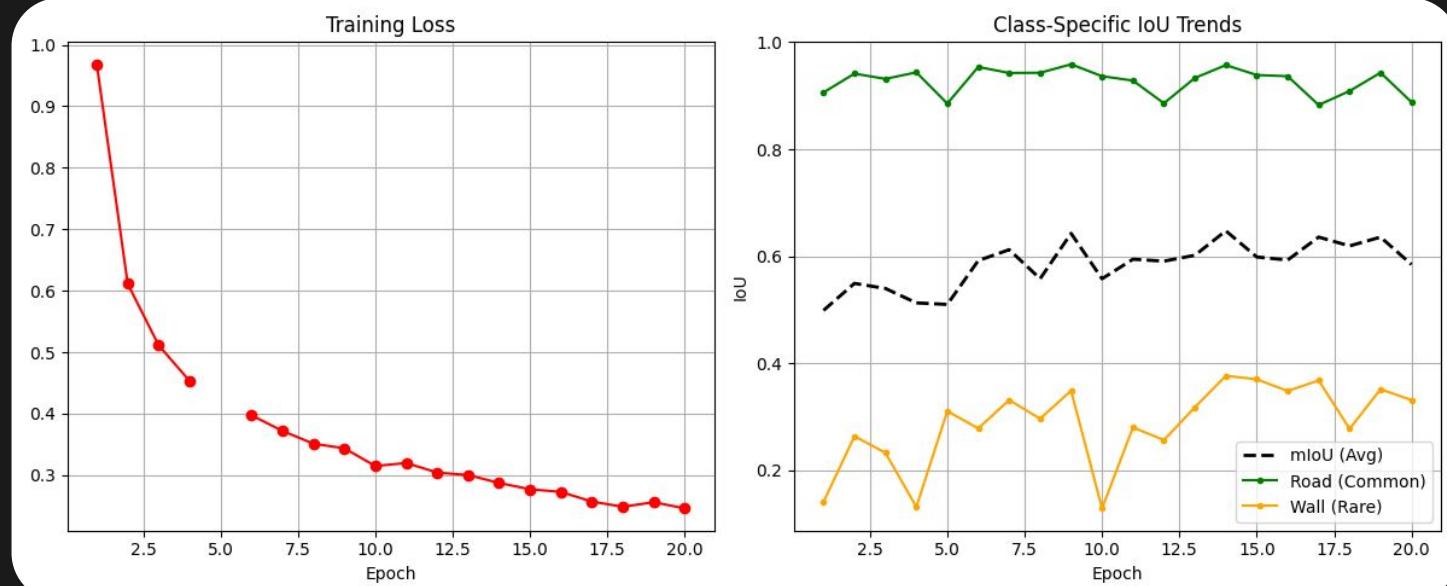
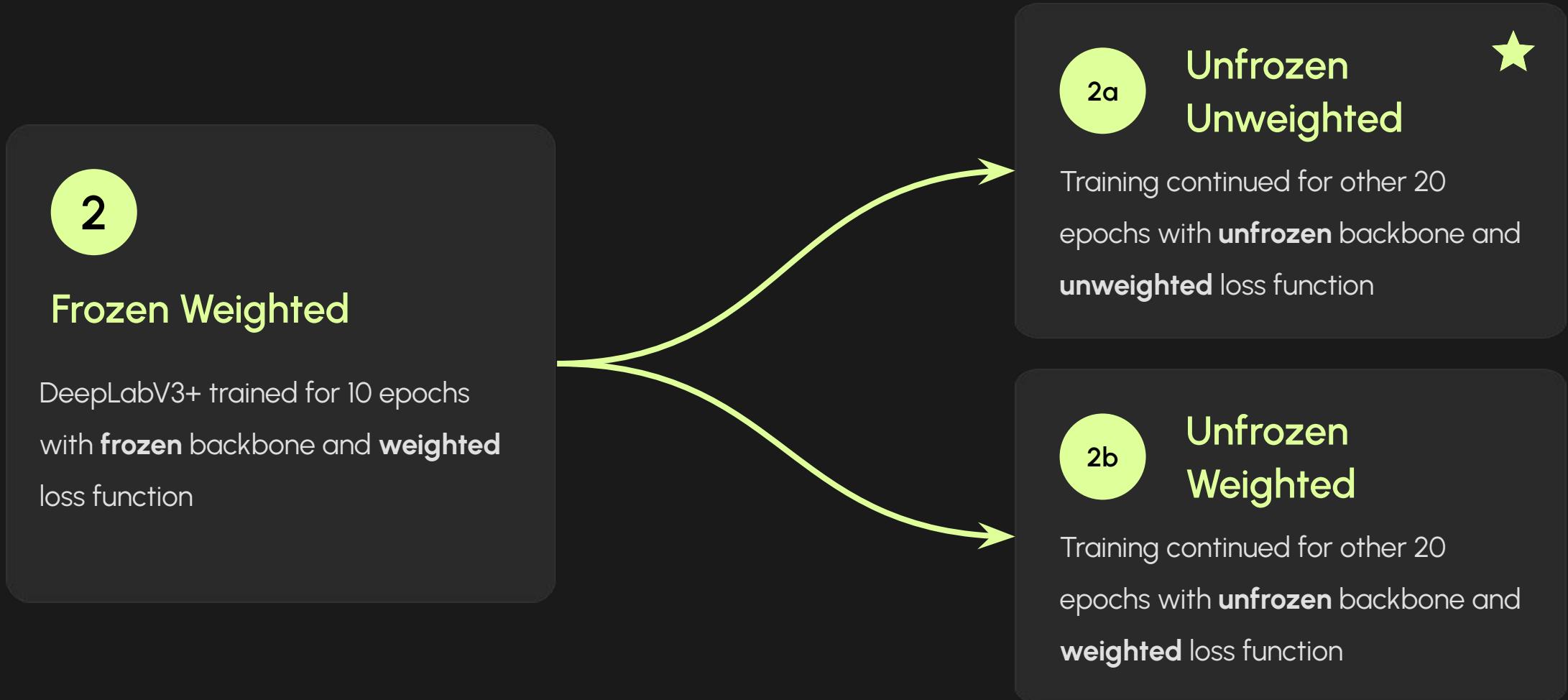


Image: Training plot of model DeepLab 1

20 epoch training

# DeepLabV3+: Training 2



# DeepLabV3+: Training 2 Results

Image 1: Training plot of model DeepLab 2a

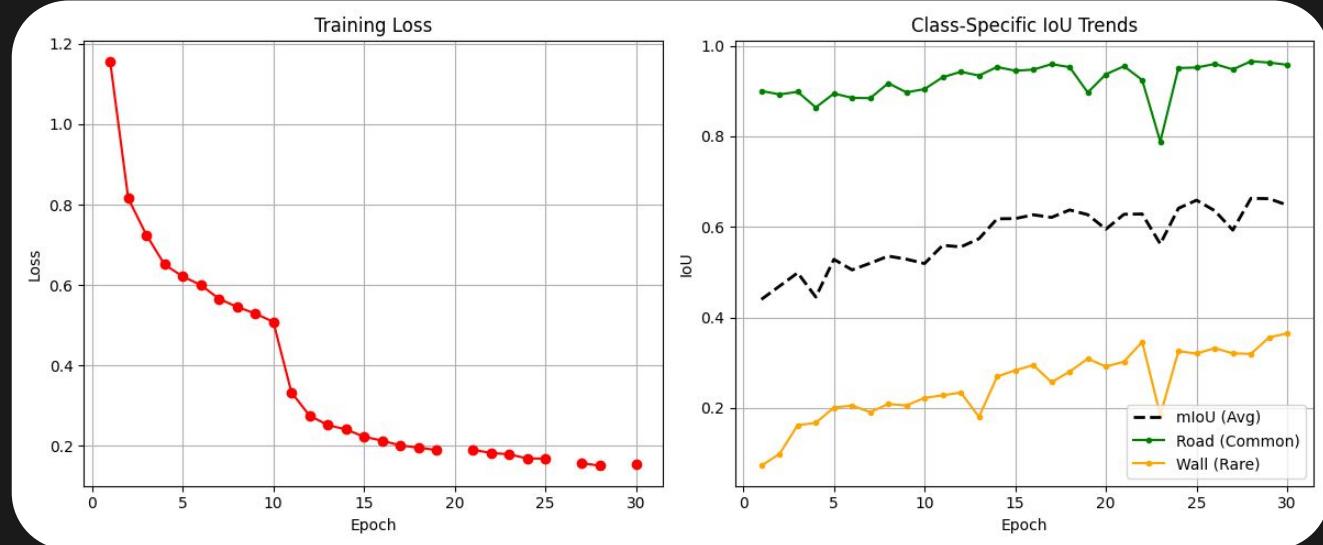
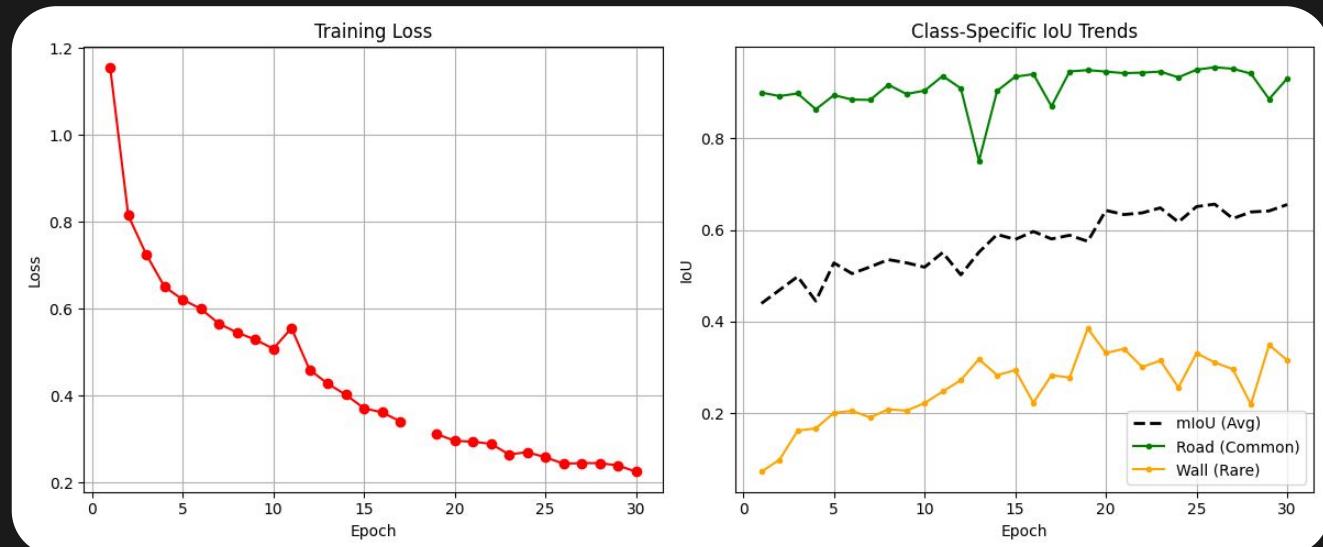


Image 2: Training plot of model DeepLab 2b



# DeepLabV3+: Evaluation Results

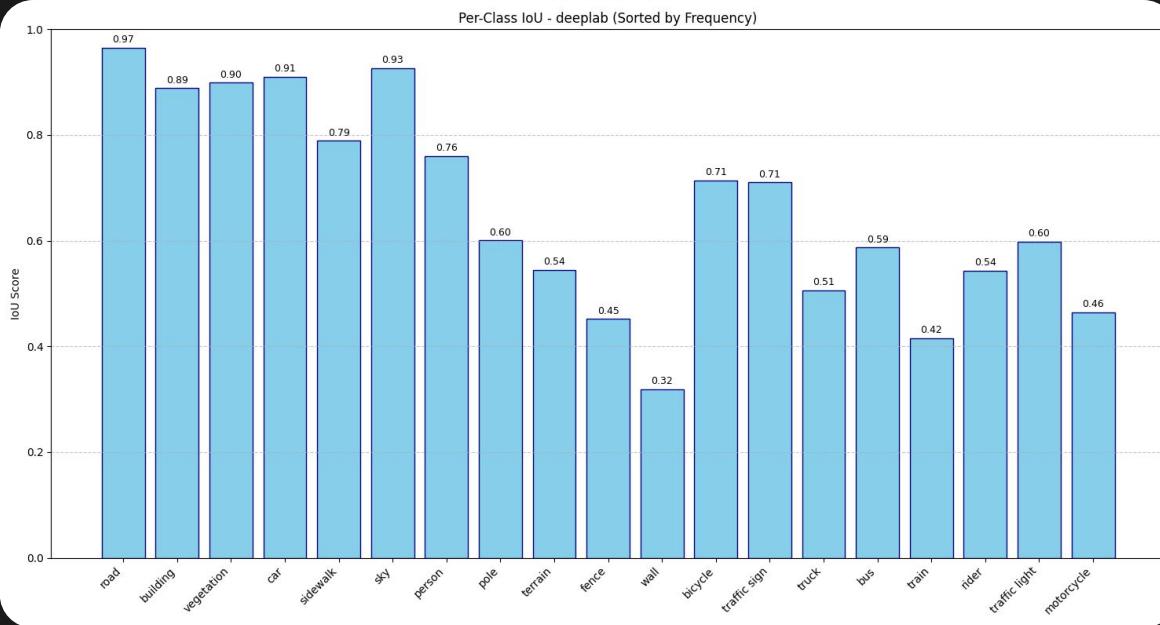


Image 1: Per-Class IoU plot of model DeepLab 2a

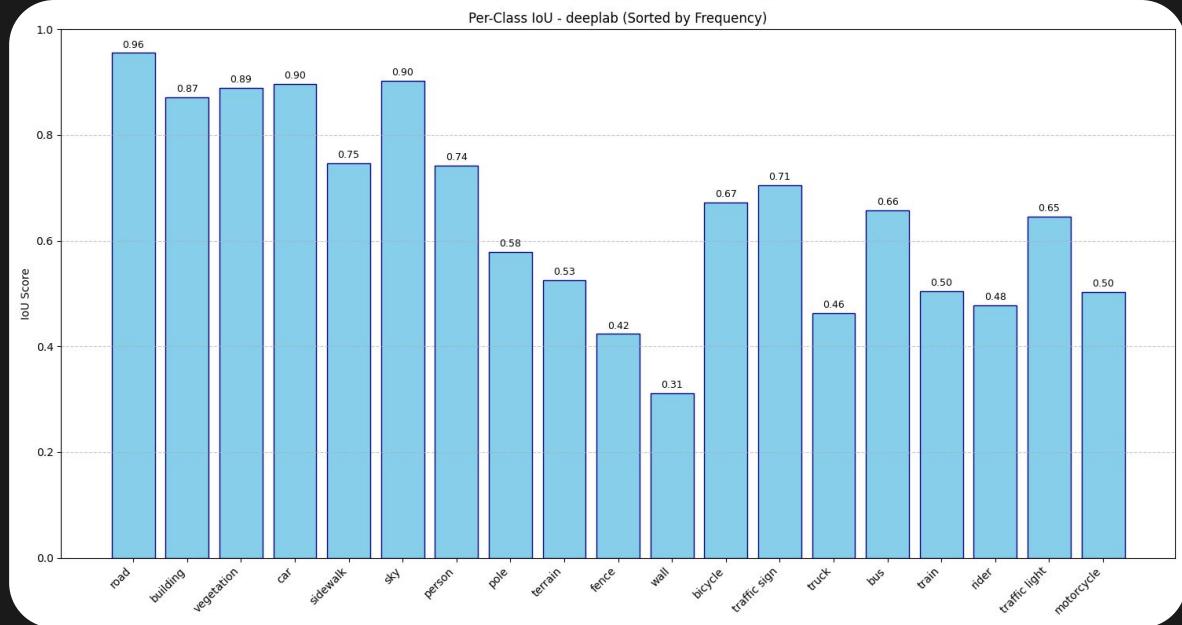


Image 2: Per-Class IoU plot of model DeepLab 2b

# DeepLabV3+: Evaluation Results

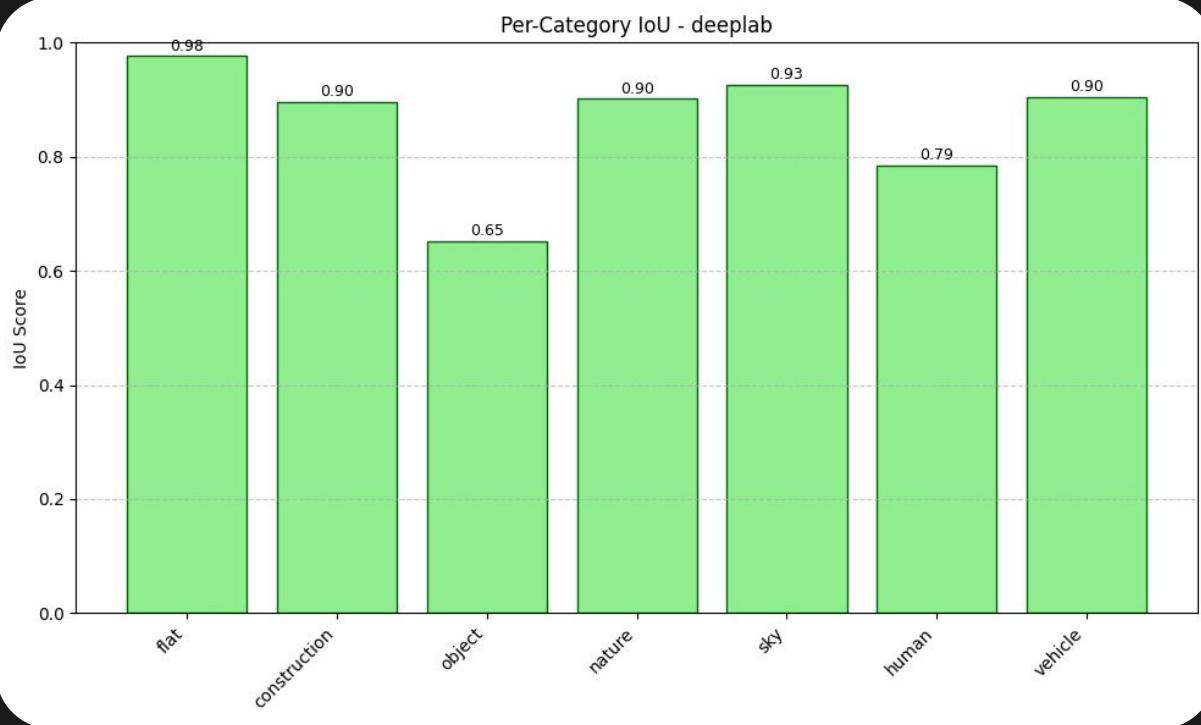


Image 1: Per-Category IoU plot of model DeepLab 2a

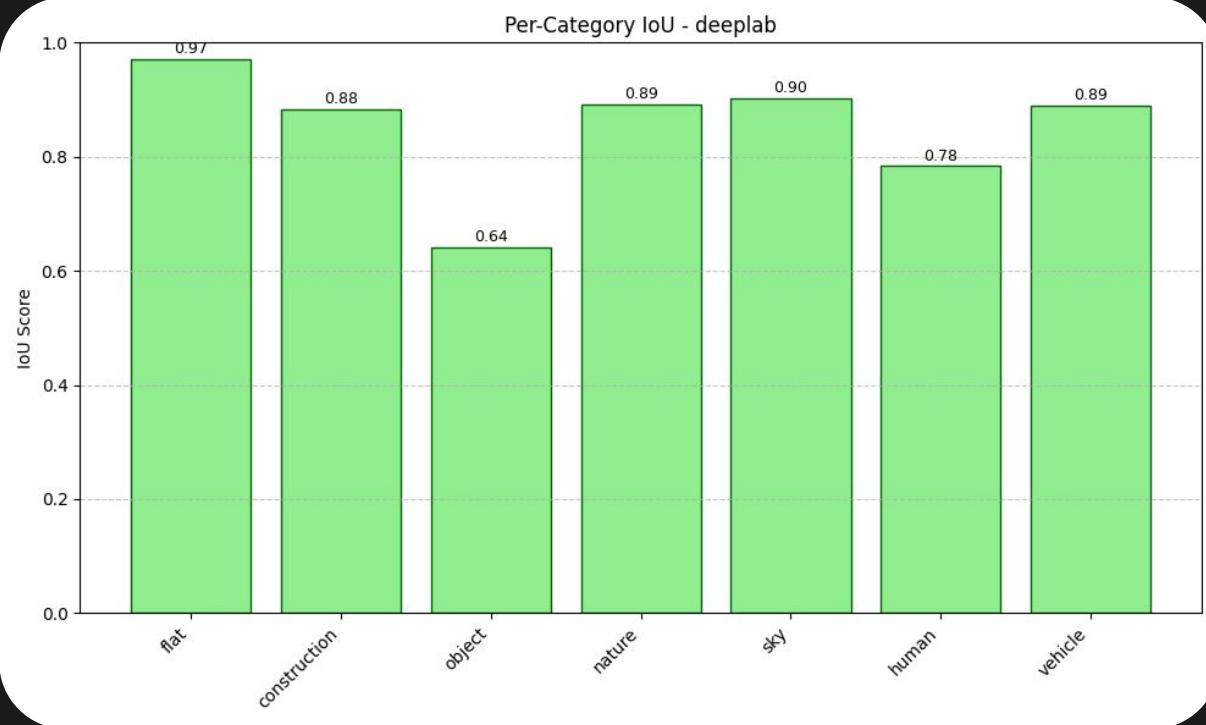
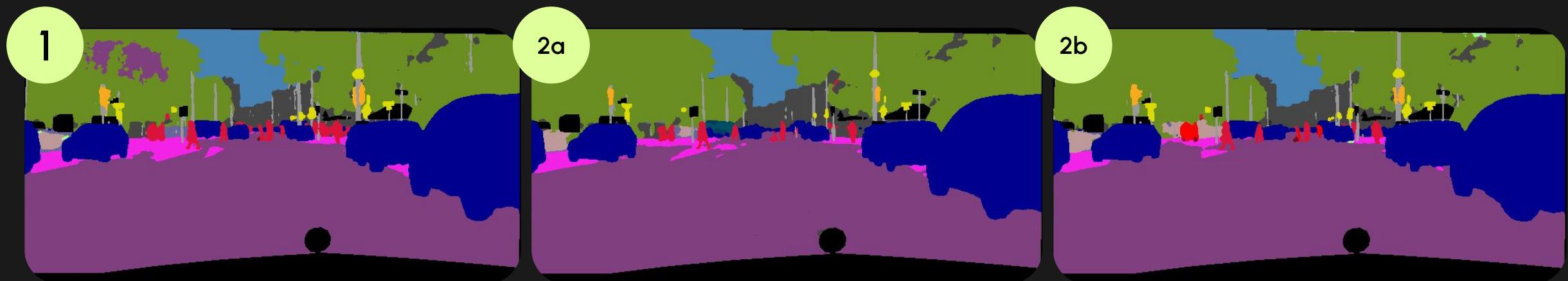


Image 2: Per-Category IoU plot of model DeepLab 2b

# DeepLabV3+: Evaluation Results



Reference image: frankfurt\_000000\_001236\_leftImg8bit.png

# SegFormer

Encoder-Decoder

Hierarchical Transformer

Simple MLP

Positional-Encoding-Free

MiT - BO

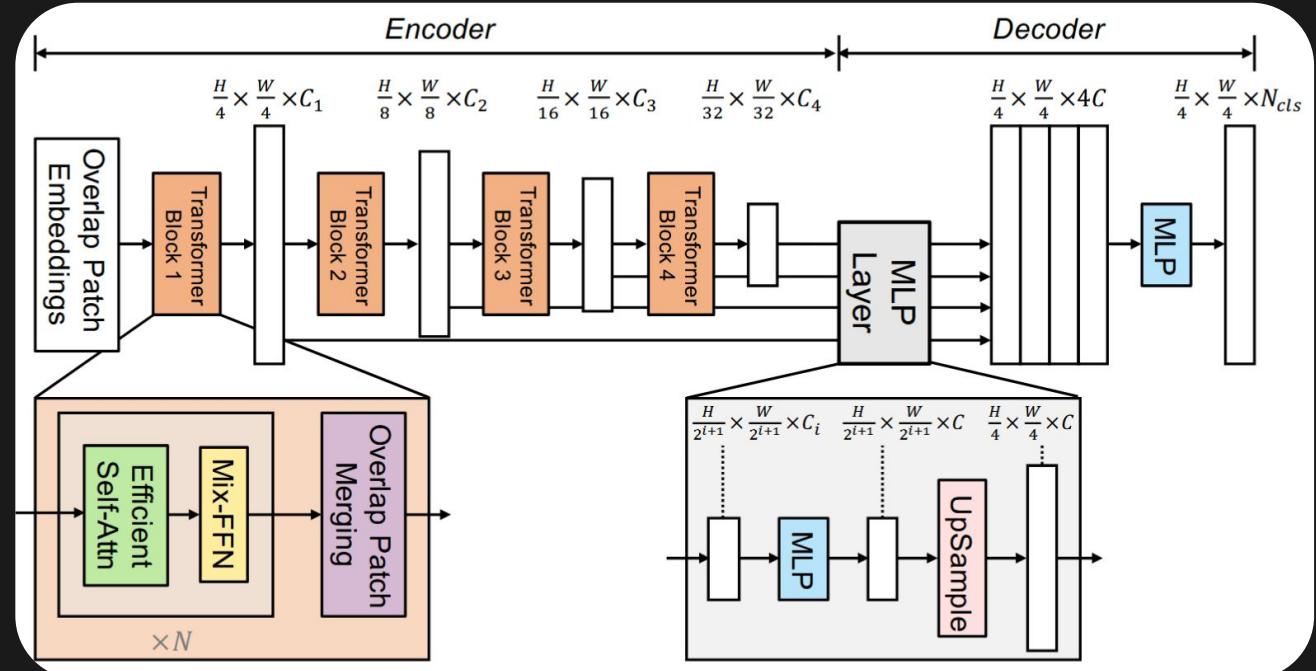


Image: <https://doi.org/10.48550/arXiv.2105.15203>

# SegFormer: Training 1

1

## Unfrozen Weighted

SegFormer trained for 20 epoch with  
**unfrozen** backbone and **weighted**  
loss function

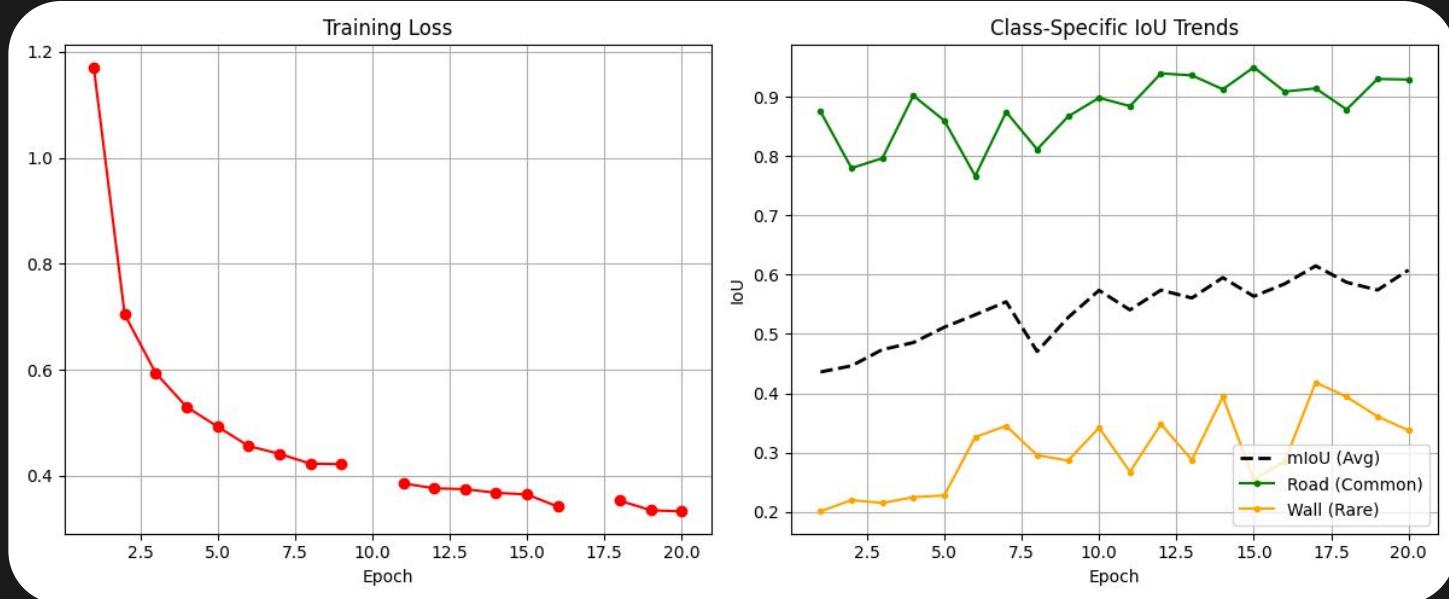
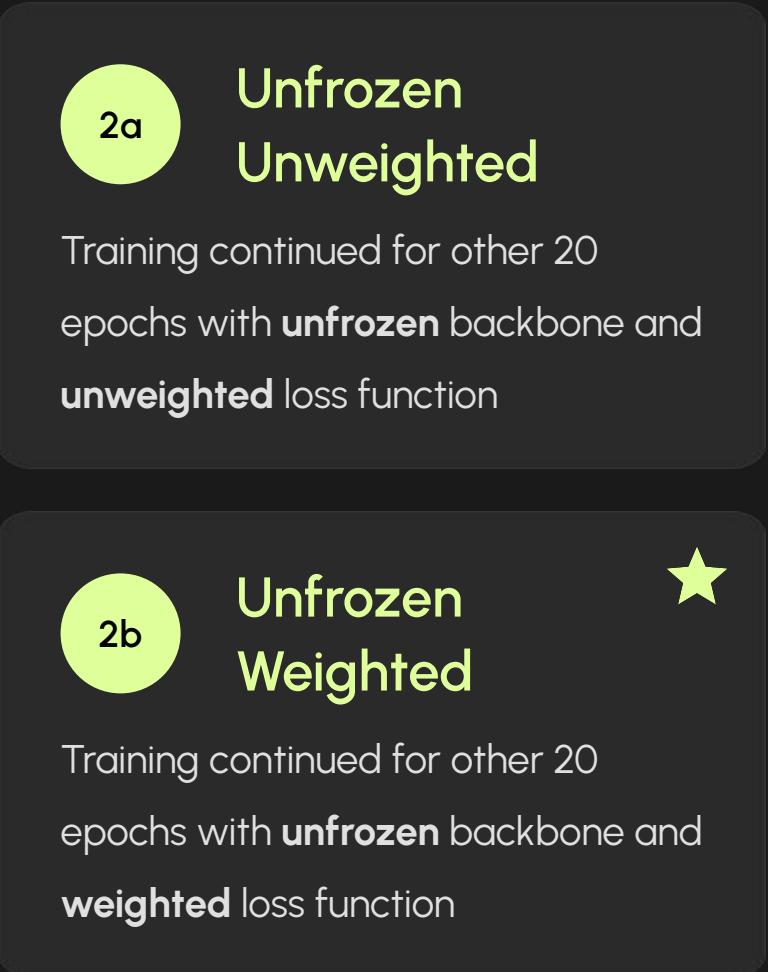
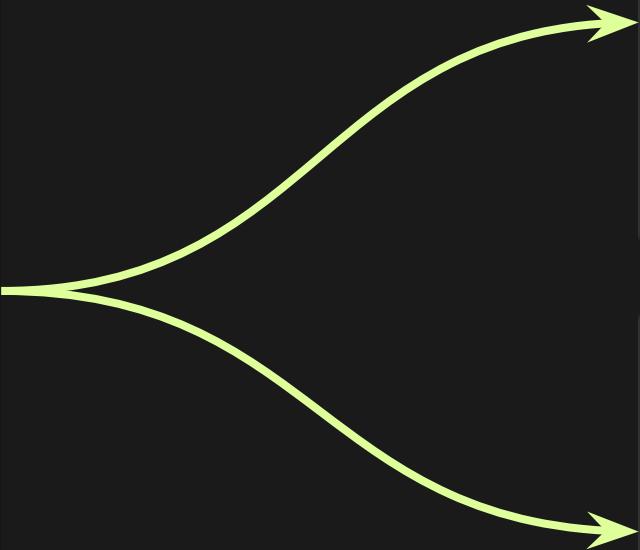
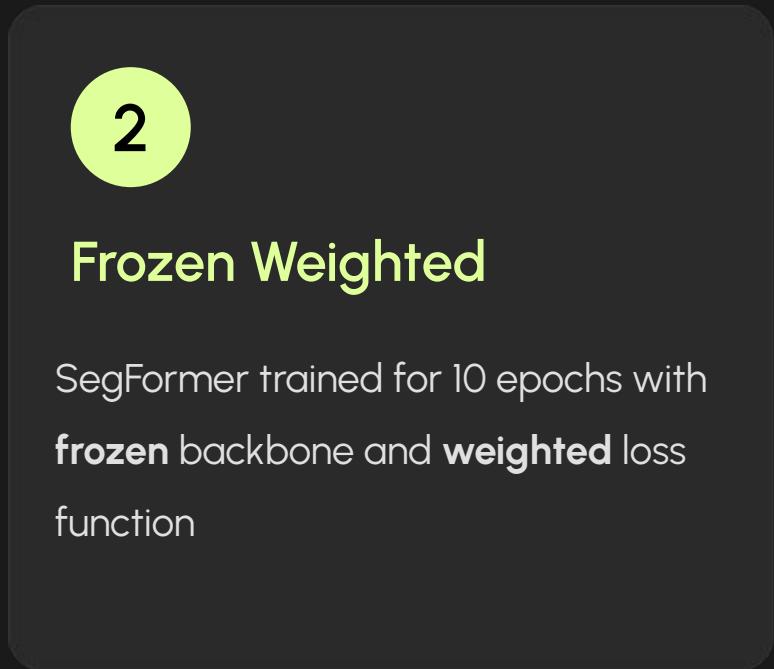


Image: Training plot of model SegFormer 1

20 epoch training

# SegFormer: Training 2



# SegFormer: Training Results

Image 1: Training plot of model SegFormer 2a

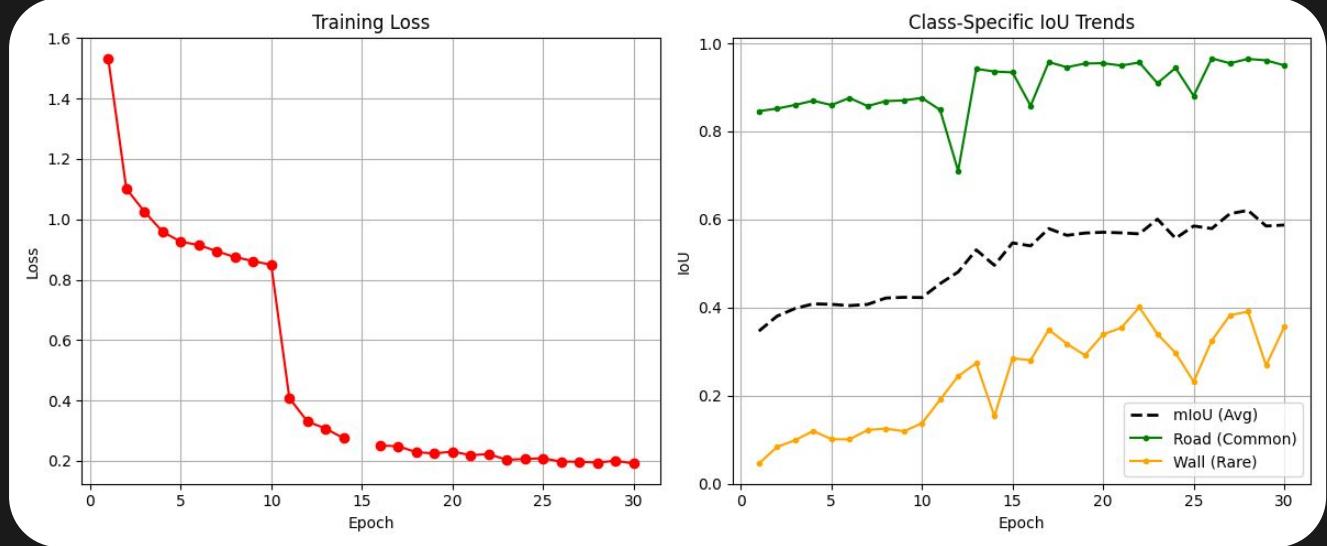
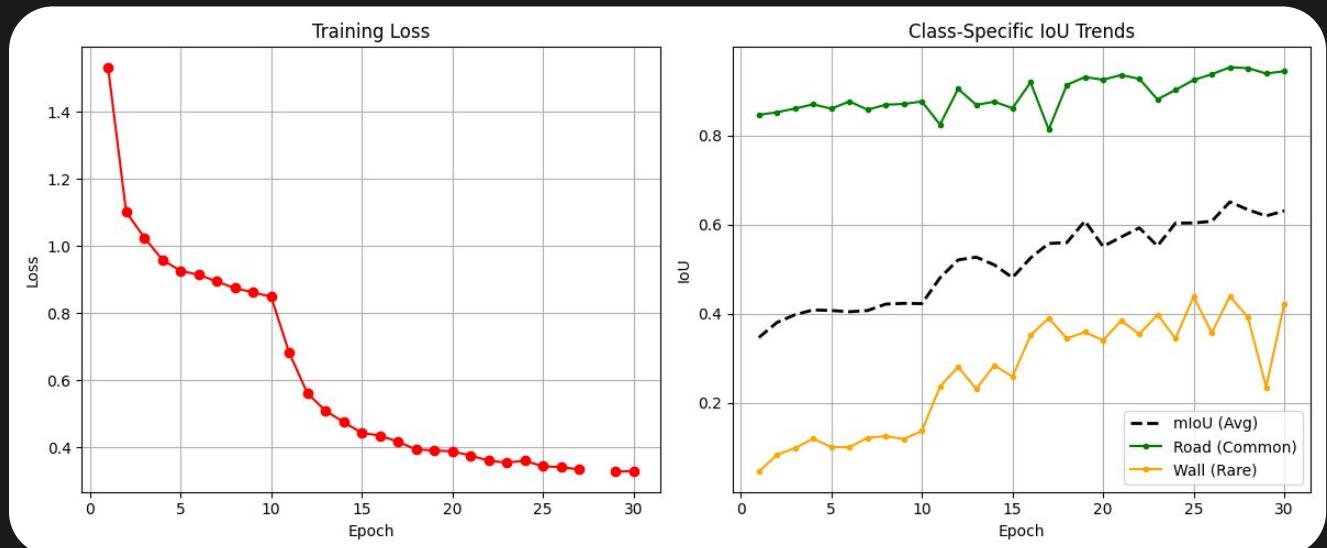


Image 2: Training plot of model SegFormer 2b



# SegFormer: Evaluation Results

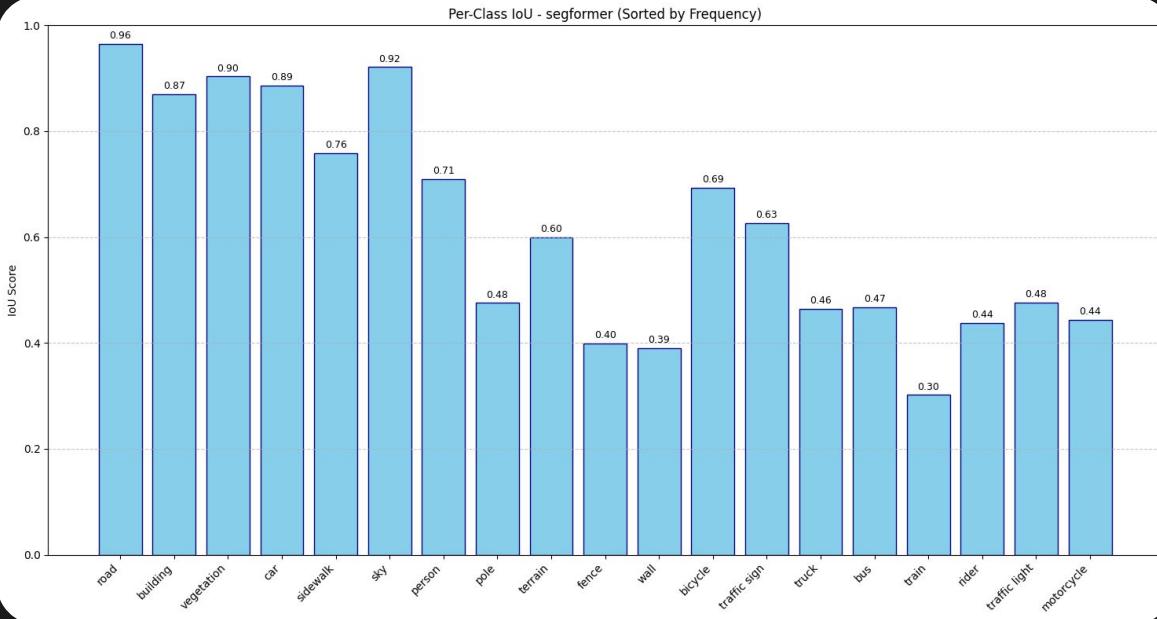


Image 1: Per-Class IoU plot of model SegFormer 2a

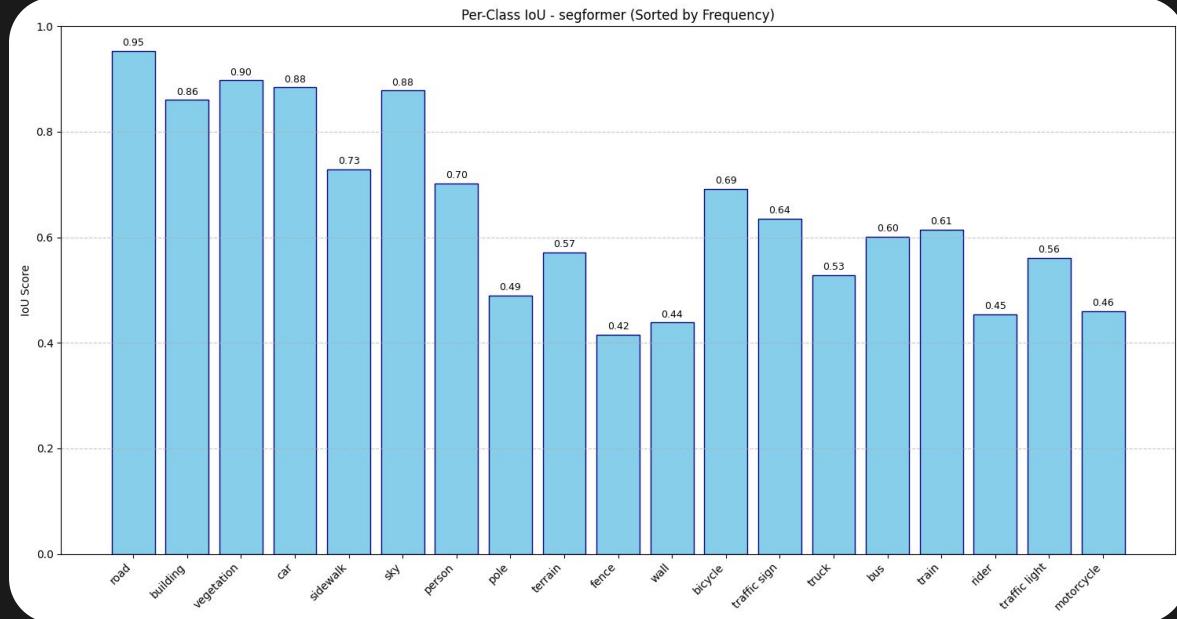


Image 2: Per-Class IoU plot of model SegFormer 2b

# SegFormer: Evaluation Results

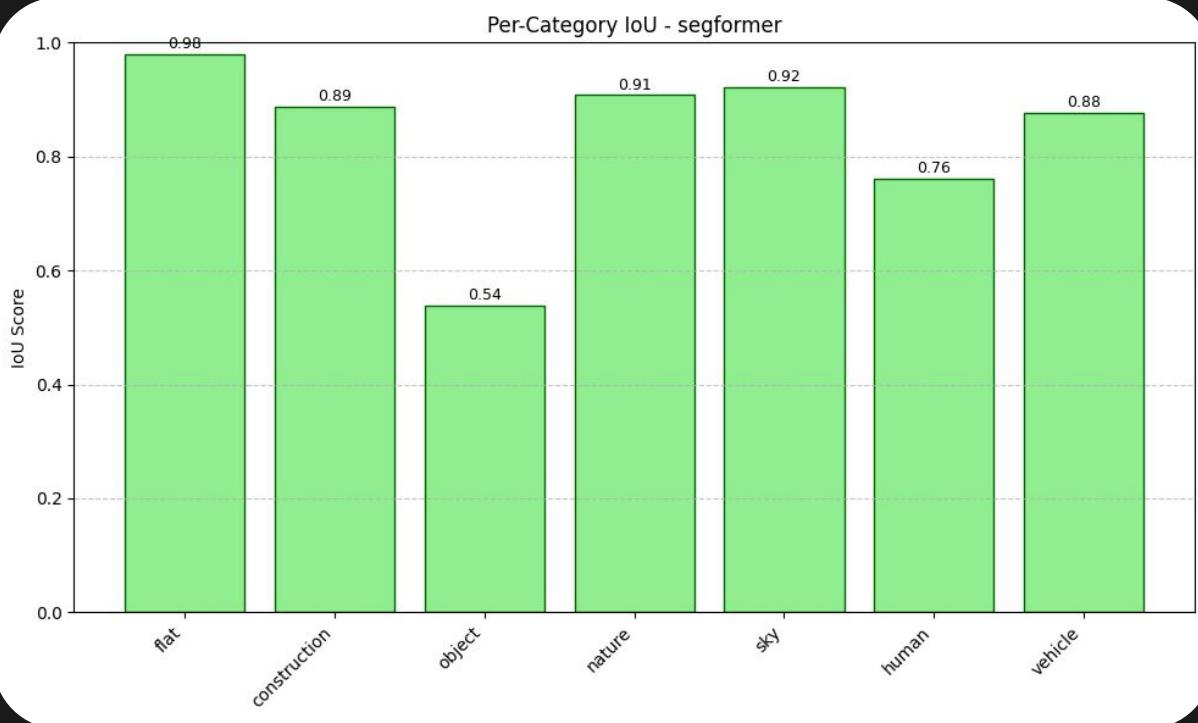


Image 1: Per-Category IoU plot of model SegFormer 2a

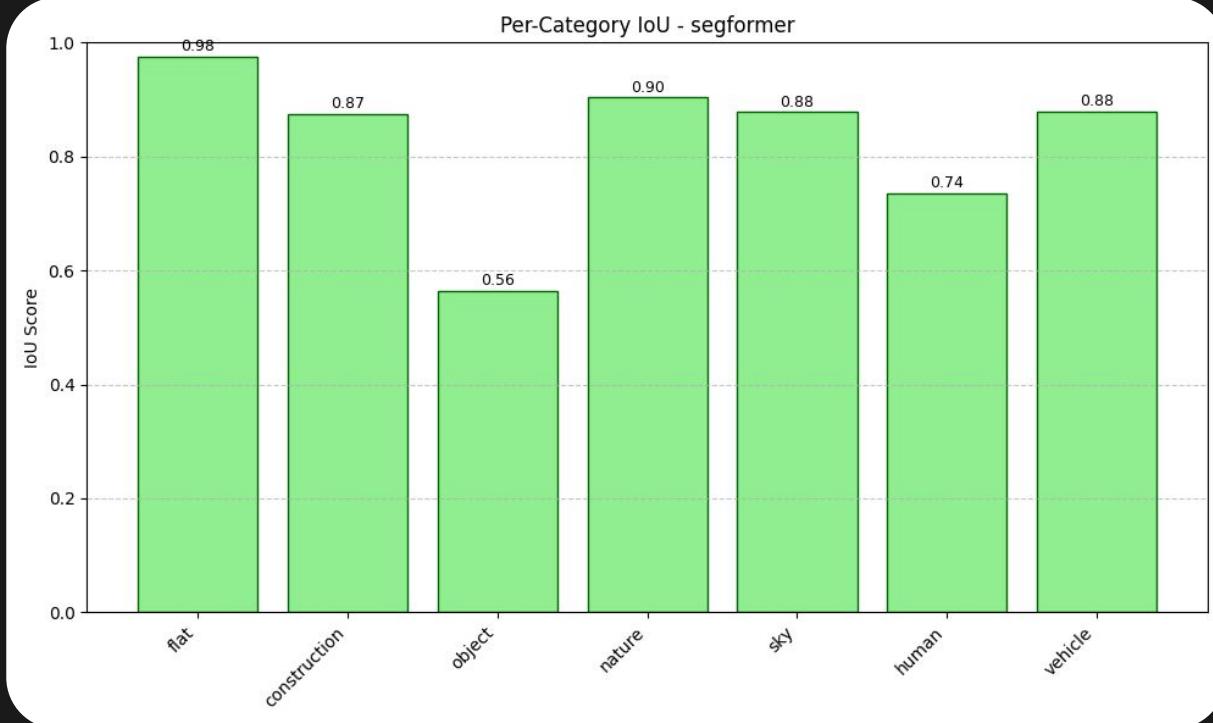
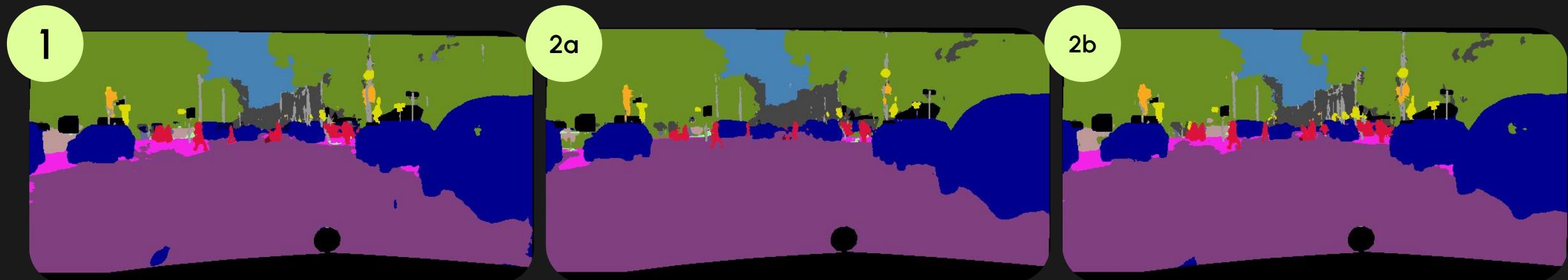


Image 2: Per-Category IoU plot of model SegFormer 2b

# SegFormer: Evaluation Results



Reference image: frankfurt\_000000\_001236\_leftImg8bit.png

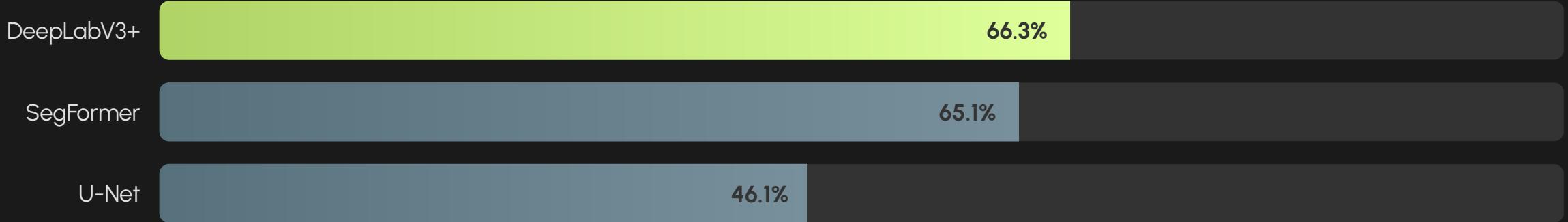
# Model Parameters

Model	Scenario	Total Params	Trainable Params	% Trainable
U-Net	No backbone	31.38M	31.38M	100.0%
	Unfrozen	42.40M	<b>42.40M</b>	100.0%
DeepLabV3+	Frozen	42.40M	16.84M	39.7%
	Unfrozen	3.72M	3.72M	100.0%
SegFormer	Frozen	3.72M	<b>0.40M</b>	10.7%

# Comparative Analysis

Model	Pixel Acc.	Mean Per-Class Acc.	mIoU Class	mIoU Category
U-Net 1a	0.8998	0.5491	<b>0.4612</b>	0.7869
DeepLab 1	0.9290	<b>0.8069</b>	0.6477	0.8429
DeepLab 2a	<b>0.9408</b>	0.7531	<b>0.6629</b>	<b>0.8632</b>
DeepLab 2b	0.9303	0.7998	0.6565	0.8522
SegFormer 1	0.9073	0.8000	0.6149	0.8177
SegFormer 2a	0.9334	0.7335	0.6206	0.8388
SegFormer 2b	0.9264	<b>0.8069</b>	<b>0.6508</b>	0.8300

# Results Deep Dive: mIoU Comparison



**DeepLabV3+:** Superior performer, benefiting from multi-scale ASPP context and large resolution

**SegFormer:** Similar performance to DeepLabV3+ with a lighter backbone (4M parameters)

**U-Net:** Least performing model with the heaviest computational load.

## CNN Insight

U-Net and DeepLab benefit significantly from a "**Mixed**" loss strategy (Weighted → Unweighted).

## Transformer Insight

SegFormer requires **persistent class weighting**. Removing weights caused a performance drop on less frequent classes.

# Training Time: cropped (512x1024) Input

Model	Per-Epoch	Per-Validation	Total Time
<b>U-Net 1a</b>	10m	20m	7h
<b>DeepLab 1</b>	5m30s	2m30s	2h45m
<b>DeepLab 2a</b>	5m30s	2m30s	4h
<b>DeepLab 2b</b>	5m30s	2m30s	4h
<b>SegFormer 1</b>	2m20s	1m30s	1h20m
<b>SegFormer 2a</b>	2m20s	1m30s	2h
<b>SegFormer 2b</b>	2m20s	1m30s	2h

# Inference Speed: (1024x2048) Input

SegFormer

6.49 FPS

FP32

7.62 FPS

AMP

DeepLabV3+

3.68 FPS

FP32

7.18 FPS

AMP

U-Net

0.24 FPS

FP32

0.51 FPS

AMP

\*not to scale

## ⚠️ Memory Bottleneck

U-Net exceeds 8GB VRAM, forcing data swapping to system RAM.

Result: Latency spikes to ~4222ms without AMP

Model (AMP)	Latency	FPS
SegFormer	131.16 ms	<b>7.62</b>
DeepLabV3+	139.29 ms	<b>7.18</b>
U-Net	1974.98 ms	0.51

# Inference Speed: 512x1024 Input

SegFormer

61.59 FPS

FP32

69.18 FPS

AMP

DeepLabV3+

15.89 FPS

FP32

31.72 FPS

AMP

U-Net

10.12 FPS

FP32

14.44 FPS

AMP

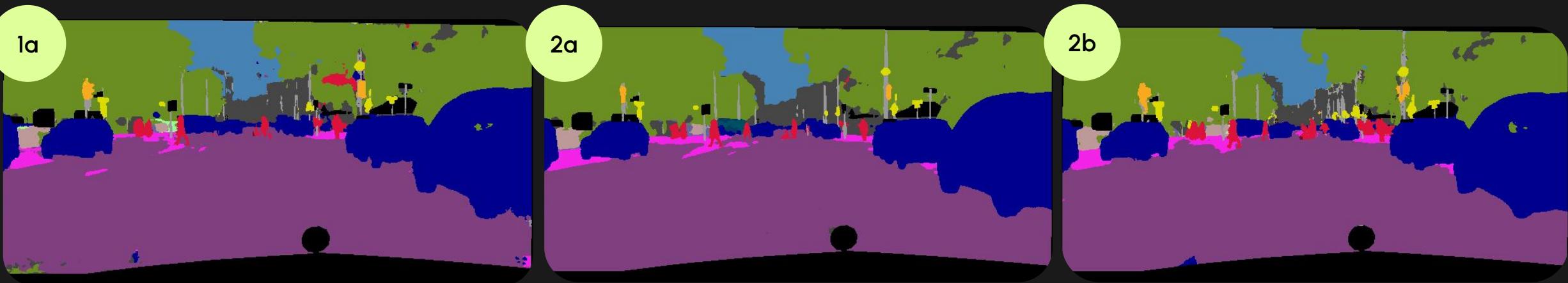
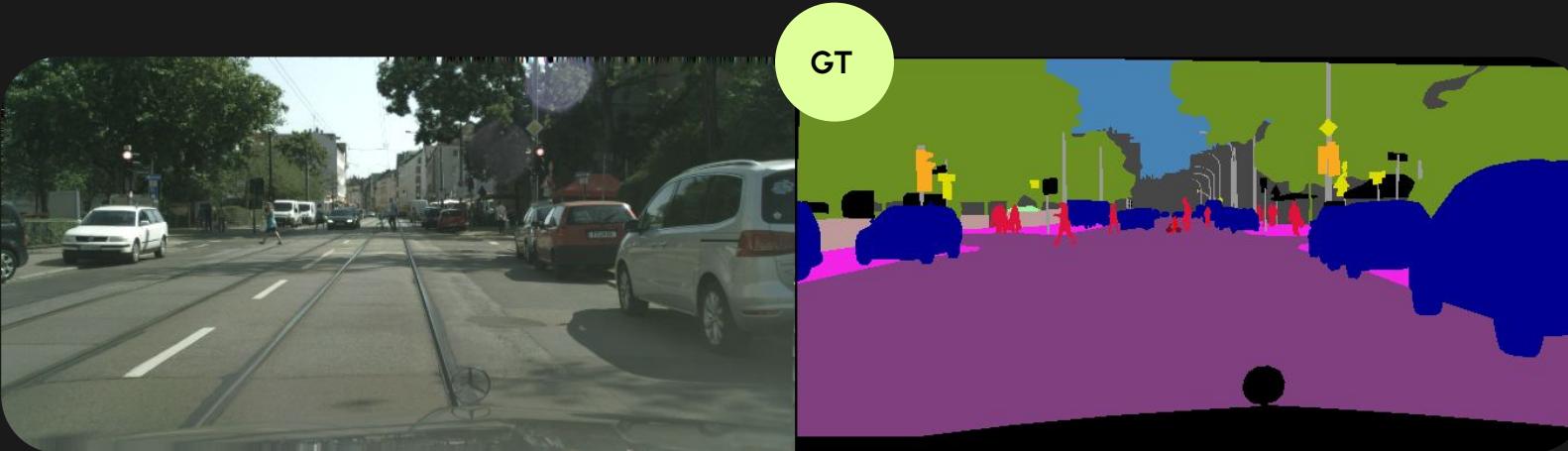
## ✓ VRAM Constraint Resolved

At **512x1024**, U-Net fits in 8GB VRAM, eliminating system RAM swapping.

Note: SegFormer achieves ~69 FPS, being the most effective approach for real time systems.

Model (AMP)	Latency	FPS
SegFormer	14.45 ms	69.18
DeepLabV3+	31.52 ms	31.72
U-Net	69.26 ms	14.44

# Qualitative analysis



U-Net 1a

DeepLab 2a

SegFormer 2b

# Conclusion & Recommendations

## › Optimal Architecture

**DeepLabV3+** obtained the best mIoU results on the validation dataset, followed by **SegFormer** which achieves similar results while being drastically more efficient.

## › Key Learnings

1. **Gradient Accumulation & AMP** are non-negotiable for high-res images on constrained hardware.
2. **Adaptations** (Instance/Group normalization) are mandatory to bypass small-batch instability.
3. **Class Weighting** is fundamental to address the "long-tail" distribution.

# Thank you

Any question?

# Bibliography

## > U-Net

Source: <https://doi.org/10.48550/arXiv.1505.04597>

---

## > DeepLabV2

Source: <https://doi.org/10.48550/arXiv.1606.00915>

---

## > DeepLabV3

Source: <https://doi.org/10.48550/arXiv.1706.05587>

---

## > DeepLabV3+

Source: <https://doi.org/10.48550/arXiv.1802.02611>

---

## > SegFormer

Source: <https://doi.org/10.48550/arXiv.2105.15203>

---

## > Demo semantic segmentation ADAS

Source: <https://github.com/MarcelloCeresini/DemoSemanticSegmentationADAS.git>

---

## > Cityscapes

Source: <https://doi.org/10.48550/arXiv.1604.01685>

---

## > ENet

Source: <https://doi.org/10.48550/arXiv.1606.02147>

---