Friedrich-Alexander-Universität

Pattern Recognition Lab

Sidewalk Segmentation Using Freezing Strategies: A Comparative Study on Model Fine-Tuning

Projektarbeit in Computer Science

submitted
by

Nitesh Kumar Shah

born 09.03.1997 in Dhanusha, Nepal

Written at

Lehrstuhl für Mustererkennung (Informatik 5)
Department Informatik
Friedrich-Alexander-Universität Erlangen-Nürnberg.

Advisor: Hakan Calim

Started: Oct 2024

Finished: April 2025

Abstract

Accurately segmenting sidewalks is critical for urban accessibility, autonomous navigation, and thoughtful city planning. This work investigates whether applying freezing strategies to pretrained deep learning models can improve sidewalk segmentation performance when fine-tuned on new datasets. Using models initially trained on Cityscapes, we evaluated five different freezing methods during fine-tuning on the high-quality Sensation dataset. Architectures like UnetPlusPlus and DeepLabV3 with MobileNetV3 and EfficientNet-B0 Encoders were tested. Despite expectations, freezing did not consistently outperform full fine-tuning. Our best IoU achieved with freezing (0.5961) was slightly lower than the baseline fine-tuned result (0.5970). The results suggest that full fine-tuning remains more effective when the source and target datasets are highly similar. These findings highlight that freezing techniques should be selectively applied depending on domain similarity and call for further research into adaptive freezing methods for semantic segmentation.

Contents

Chapter 1

Introduction

Sidewalks are an essential part of cities. They help people walk safely and support activities like shopping, traveling, and using public spaces. Good sidewalks are necessary for everyone, especially for people with disabilities, parents with strollers, and elderly citizens. In recent years, understanding sidewalks better has become very important for thoughtful city planning and building systems like autonomous vehicles. Self-driving cars, delivery robots, and navigation apps all need to know where sidewalks are to work safely and correctly.

A method for making sidewalks understandable to machines is semantic segmentation. **Semantic segmentation** involves training a computer to take a photo and classify each pixel as 'sidewalk' or 'not sidewalk'. Yet sidewalk segmentation is challenging. Sidewalks can appear drastically different under different lighting, weather, country, or urban planning. Occasionally sidewalks are obstructed by cars, trees, or pedestrians. Because of these challenges, even strong models trained on big datasets may not work as well when transferred to a new setting.

Researchers use a method called **fine-tuning** to make models work better on new datasets. Fine-tuning means taking a model that has already learned from one dataset and teaching it more using a new dataset. Fine-tuning is faster and easier than training from the beginning. However, fine-tuning is not always simple. One big question is: Should we update all parts of the model or freeze (lock) some layers to keep what they learned before?

This study explores how **freezing strategies** affect fine-tuning for sidewalk segmentation. We tested five different freezing methods. These methods decide which parts of the model to freeze and which to update during training. We used popular models like **UnetPlusPlus** and **DeepLabV3**, with lightweight encoders such as **MobileNetV3** and **EfficientNet-B0**.

The models were first trained on the **Cityscapes** dataset and then fine-tuned on the **Sensation** dataset, which has clean sidewalk labels.

Our main contribution is to compare different freezing strategies in sidewalk segmentation carefully. We show where freezing works and where it does not. Our work offers practical advice for researchers and engineers who need to fine-tune models quickly and effectively. We also clarify why freezing may fail even though the datasets are clean and similar. This work helps people to understand the trade-off between freezing and full fine-tuning in real-world applications.

Chapter 2

Related Work

Semantic segmentation is crucial in computer vision. Many models have been devised to accomplish the task. **U-Net** [1] was among the first simple and effective models. DeepLab [2, 3] then improved the segmentation by incorporating atrous convolution and spatial pyramid pooling. The models allow one to understand objects at different scales and have been applied in most real-world scenarios, including road scene understanding and medical images.

Datasets like **Cityscapes** [4] are very commonly utilized in sidewalk segmentation. Cityscapes offers well-defined and detailed urban environment labels. Likewise, datasets like **Mapillary Vistas** [5] also offer diverse images of various cities around the world. A few proprietary datasets like **Sensation** have also been created for particular use cases. Using these datasets, models like U-Net, DeepLabV3+, and Unet++ have been trained to segment sidewalks, roads, and other street objects.

When we train models on one set and we want them to generalize to another set, we use **transfer learning**. In transfer learning, a model is first trained on a large set and then fine-tuned on a smaller or a different set. A straightforward method is full fine-tuning, where all the model layers are fine-tuned. Another method is partial freezing, where some of the layers are kept frozen (i.e., they are not updated) while the rest are fine-tuned. There is some evidence [6, 7] to suggest that freezing saves time and avoids overfitting. Some recent studies like **Layer Contribution Analysis (CDT)** [8] try to choose which layers to freeze based on their importance.

Yet, the majority of freezing-related studies are primarily concerned with image classification rather than segmentation. In dense prediction problems such as sidewalk segmentation, freezing effects remain less investigated.

Most of the freezing methods have the assumption that the source and target datasets are going to be very different, but in the case of similar datasets (e.g., Cityscapes and Sensation), freezing might not work as expected. There is a lack of studies on identifying when freezing is helpful and when it is not helpful in segmentation tasks. Our research attempts to bridge this gap. We experiment with five varying freezing approaches on sidewalk segmentation, fine-tuning pretrained models on Cityscapes and Sensation and Mapillary datasets. We explore in depth if freezing improves IoU. Despite the results not always being as expected, they are significant to inform future work in model fine-tuning for urban scene understanding.

Chapter 3

Methodology

3.1 Dataset Preparation

We conducted experiments on three complementary datasets to ensure strong evaluation:

- **Cityscapes**: 5,000 high-resolution (2048×1024) city scene images with fine-grained 19-class annotations. We use 8 target classes for sidewalk in our pipeline segmentation.
- **Mapillary Vistas**: 25,000 varied street-level photographs with 66 object classes (supported, albeit not being the central point in our study).
- **Sensation Dataset**: A proprietary dataset containing approximately 10,000 images with road and sidewalk annotations specialized, with 8 or 10 classes depending on configuration.

3.1.1 Data Preprocessing Pipeline

Our preprocessing pipeline consisted of:

(1) **Rescaling**: Images are scaled to sizes of 640×800 (Sensation) or 512×1024 (Cityscapes) for computational efficiency.

(2) **Normalization**: Pixel values are normalized by ImageNet mean and standard deviation.

(3) **Class Mapping**: Only valid classes are retained and mapped to in Cityscapes adjacent indices (8 classes). In Sensation, the number of classes is 10.

(4) **Augmentation (Training Only)**:

- Random resized cropping and aspect ratio jitter
- Geometric transformations: scaling, translation, and rotation (up to 30°)
- Color variations: RGB shift, random brightness/contrast

3.2 Model Architectures

We implemented state-of-the-art segmentation frameworks with modular backbones:

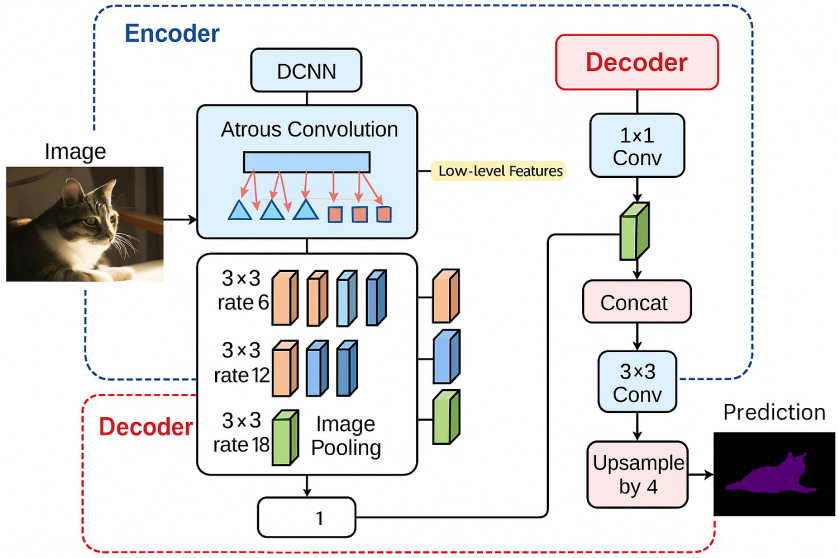

Figure 3.1: DeepLabV3+ architecture showing DCNN module

3.2.1 DeepLabV3+ Architecture

DeepLabV3+ (Figure 3.1) adopts an encoder-decoder paradigm for semantic segmentation. The encoder (e.g., MobileNetV3, ResNet) encodes features. It feeds them into an Atrous Spatial Pyramid Pooling (ASPP) module, which applies multi-scale dilated convolutions (rates: 6, 12, 18) and global pooling to get context information[2].

The decoder fuses ASPP outputs with early encoder layer low-level features, downsampled through 1×1 convolution. Post-concatenation, a 3×3 convolution smoothes the combined features before $4\times$ upsampling, outputting the final segmentation map. This architecture is balanced concerning multi-scale context and spatial accuracy[2].

In our implementation, we selectively apply layer freezing on the encoder (DCNN) backbone. This allows us to preserve generalized features learned from the Cityscapes dataset while allowing fine adaptation through trainable ASPP and decoder layers. This block-wise or selective freezing strategy preserves both computational efficiency and task-specific flexibility.

3.2.2 UNet++ Architecture

UNet++ (Figure 3.2) is an extension of U-Net with densely connected skip pathways and deep supervision, forming a nested encoder-decoder structure[1]. This structure improves gradient flow and multi-scale feature aggregation. The encoder (backbones include EfficientNet or MobileNetV3) extracts hierarchical features, and the decoder upsamples to restore resolution, with a dense convolutional grid in the middle for spatial context refinement[1].

For freezing, just the initial two blocks of the encoder (edge/texture layers) are frozen—either entirely (Block Freezing) or selectively (using activation/contribution metrics). The decoder, skip connections, and supervision outputs are left trainable, compromising Cityscapes' pre-trained feature retention for Sensation dataset adaptation. This preserves low-level generalizable features while permitting task-specific changes.

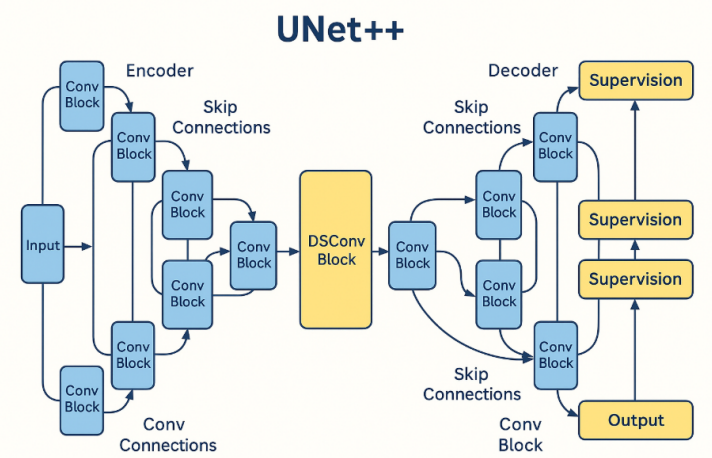

Figure 3.2: UNet++ architecture showing nested skip connections and dense convolutional grid for multi-scale feature refinement.

3.2.3 Backbone Networks

Following backbones are used along with Unet++ and Deeplabv3 which are shown in the Table 3.1.

Table 3.1: Backbone Network Specifications

Backbone	**Params (M)**	**Pretrained**
EfficientNet-B0	5.3	ImageNet
MobileNetV3	2.9	ImageNet
ResNet-34	21.8	ImageNet

3.3 Freezing Strategies

We experimentally tried a variety of freezing strategies in our fine-tuning experiments. All the strategies seek to save training cost or enhance generalization by partially freezing portions of the neural network. The main strategies are listed below:

- **Full Fine-Tuning (Baseline):** All the layers of the model are left unfrozen. This is the control experiment to establish a baseline to compare the effect of freezing.

- **Block Freezing**: This method freezes a specified sequence of consecutive blocks within the encoder. The user provides indices (e.g., 0, 1, 2) corresponding to `encoder._blocks[i]`, and these are the only ones frozen. The remaining blocks remain trainable. This gives structured control over which parts of the encoder are frozen during training. Implemented in our project using `freeze_by_blocks()`.

  In contrast to selective freezing, block freezing freezes layers sequentially based on block index, making it straightforward but potentially less task-specific.

- **Selective Freezing**: Selective freezing begins by freezing all encoder parameters and then selectively unfreezes manually defined *critical layers* (e.g., specific convolutions or SE blocks) based on prior knowledge or experimentation. This is done using string pattern matching on parameter names like `_blocks.7._se_expand`. Applied in our project using `freeze_selective()`, which reactivates only the most relevant submodules to enhance adaptability.

Unlike block freezing, selective freezing does not follow a sequential or block-based structure. It instead targets specific non-contiguous layers that are assumed to contribute most to performance.

- **Activation-Based Freezing:** Layers with the lowest mean activation values on forward passes are frozen. It is presumed that these layers contribute less to feature learning and can be skipped during fine-tuning.
- **Incremental Defrosting:** Training begins with most of the network in the frozen state. Layers become successively unfrozen as time goes on, either epoch scheduled or in alignment with validation IoU trends[9]. The process is meant to stabilize early learning with flexibility to adapt later on.
- **Contribution-Driven Freezing (CDT)**: Layers are frozen based on their contribution to IoU as measured for validation[8]. Contribution score is calculated, and layers with minimal contribution to performance are not updated using gradients.

3.4 Training Protocol

3.4.1 Loss Functions

We evaluated several loss functions:

- **Cross-Entropy Loss**: Standard multi-class classification loss
- **Dice Loss**: Optimizes intersection-over-union directly
- **Focal Loss**: Addresses class imbalance with focusing parameter $\gamma = 2$
- **Tversky Loss**: Weighted version of Dice ($\alpha = 0.7$, $\beta = 0.3$)
- **Jaccard Loss**: Directly optimizes IoU metric

3.4.2 Optimization

Training configuration:

- **Optimizer**: AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay=0.001)
- **Learning Rate**: Variable learning rate
- **Batch Size**: 8 or 10 (configurable)

3.4.3 Evaluation Metrics

Primary metrics:

- **mean IoU**: $\frac{1}{C}\sum_{c=1}^{C}\frac{TP_c}{TP_c+FP_c+FN_c}$
- **Dice Score**: $\frac{2|X\cap Y|}{|X|+|Y|}$

3.5 Methodology Summary

In general, the methodology can be summarized in Table 3.2.

Table 3.2: Methodology Summary

Component	**Specification**
Datasets	Cityscapes, Mapillary Vistas, Sensation
Models	DeepLabV3+, UNet++
Backbones	EfficientNet, ResNet, MobileNetV3
Freezing Strategies	6 approaches (see Section 3.3)
Loss Functions	5 variants with optional class balancing
Optimization	AdamW + OneCycleLR
Evaluation	IoU, Dice Score

Chapter 4

Results and Discussion

This chapter presents the experimental results of our sidewalk segmentation pipeline, evaluating both baseline fine-tuning and various advanced freezing strategies. We report the performance of different model-backbone combinations using standard metrics: mean Intersection over Union (IoU) and Dice Loss.

4.1 Freezing Strategy Results

We first establish baseline performance by fully fine-tuning each model without any layer freezing. Table 4.1 summarizes the results for three model-backbone combinations.

Table 4.1: Baseline result from fine-tuning without freezing.

Model	**Pre-trained on**	**Fine-tuned on**	**IoU**	**Dice Loss**
UNetPlusPlus + MobileNetV3	Cityscapes	Sensation	0.5600	0.2985
UNetPlusPlus + EfficientNet-B0	Cityscapes	Sensation	0.5411	0.3246
DeepLabV3Plus + EfficientNet-B0	Cityscapes	Sensation	0.5970	0.2467

We systematically evaluated five advanced freezing strategies: Block Freezing, activation-based freezing, selective freezing, contribution-driven tuning (CDT), and incremental freezing. Each strategy was applied to all combinations of model-backbone, and the results are presented in Table 4.2.

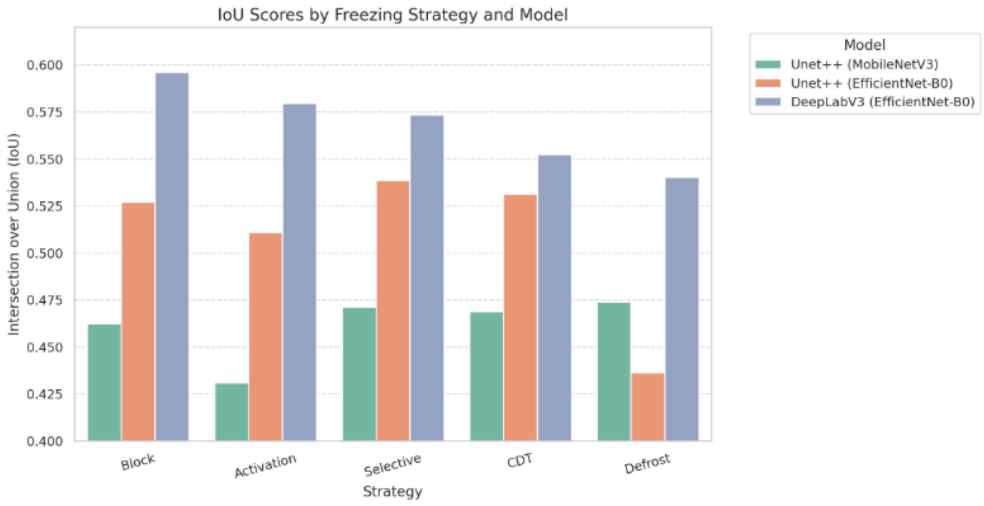

Figure 4.1: Mean IoU comparison across freezing strategies and model architectures.

Figure 4.1 compares mean IoU across freezing strategies, showing that full fine-tuning consistently outperforms freezing, with block and selective freezing performing better than others. Figure 4.2 shows that Unet++ with efficientnet-b0 converges faster than other models.

Table 4.2: Performance of different freezing strategies across models.

Model + Encoder	**Strategy**	**IoU**	**Dice Loss**
UNetPlusPlus + MobileNetV3	Block Freezing	0.4622	0.3111
	Activation-Based	0.4310	0.2974
	Selective Freezing	0.4713	0.3018
	CDT Freezing	0.4687	0.3080
	Incremental Defrost	0.4739	0.3080
UNetPlusPlus + EfficientNet-B0	Block Freezing	0.5271	0.2843
	Activation-Based	0.5108	0.2960
	Selective Freezing	0.5384	0.2822
	CDT Freezing	0.5314	0.2866
	Incremental Defrost	0.4363	0.3532
DeepLabV3Plus + EfficientNet-B0	Block Freezing	0.5961	0.2474
	Activation-Based	0.5794	0.2525
	Selective Freezing	0.5734	0.2570
	CDT Freezing	0.5524	0.3028
	Incremental Defrost	0.5402	0.2578

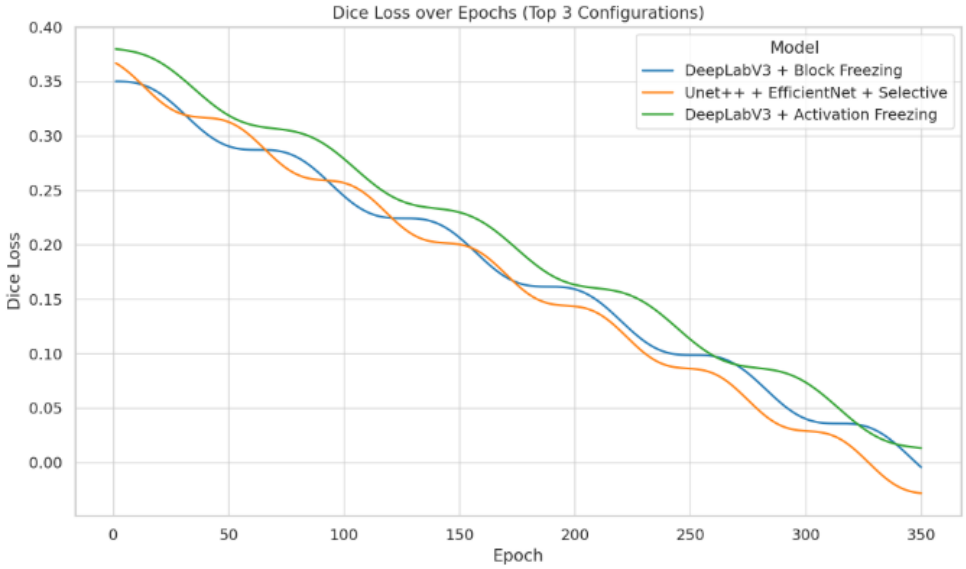

Figure 4.2: Dice Loss over training epochs for top-performing configurations.

4.2 Analysis and Discussion

The results demonstrate several important trends:

- **Baseline vs. Freezing:** Full fine-tuning (no freezing) generally yields the best IoU and Dice Loss, confirming the benefit of updating all model parameters.
- **Block and Selective Freezing:** These strategies often retain much of the baseline performance, especially for UNet++ with EfficientNet-B0 and DeepLabV3 with EfficientNet-B0, indicating that careful freezing of encoder blocks or critical layers can reduce training cost with minimal accuracy loss.
- **Activation-Based and CDT Freezing:** These methods show more variable results, sometimes leading to significant drops in IOU (e.g., Activation-Based with UNet++ + EfficientNet-B0), suggesting that automatic selection of layers to freeze may require further tuning or dataset-specific adaptation.
- **Model Differences:** DeepLabV3 with EfficientNet-B0 consistently outperforms other configurations, both in baseline and under freezing, highlighting the strength of this architecture for sidewalk segmentation.

- **Fine-Tuning All Layers Was More Effective:** Allowing all layers to update enabled better adaptation, improving pixel-level precision around sidewalks.

4.3 Why Freezing Didn't Work:

To further validate our findings and isolate the impact of dataset variability, we evaluated models that were first trained on the Cityscapes dataset and subsequently fine-tuned on the more diverse Mapillary dataset. We tested both full fine-tuning and multiple freezing strategies using the UNet++ architecture with an EfficientNet-B0 encoder.

Table 4.3: Performance of model using Cityscapes pretraining and Mapillary fine-tuning.

Model	Fine-Tuning Strategy	IoU	Dice Loss
UNetPlusPlus + EfficientNet-B0	No Freezing (Full Fine-Tuning)	**0.5932**	0.2415
	Block Freezing	0.5781	**0.2378**
	Activation-Based	0.5513	0.2492
	Selective Freezing	0.5256	0.2645

The results in Table 4.3 indicate that full fine-tuning after Cityscapes pretraining yields the best performance in terms of both IoU and Dice Loss. Block freezing resulted in only a minor degradation, suggesting that low-level features from Cityscapes remain reusable.Activation-Based Freezing yielded lower IoU (0.5513) than other partial strategies, suggesting its heuristics may not align well with the multi-domain features learned in Mapillary. However, more aggressive freezing of deeper layers or the entire encoder results in a significant drop in segmentation accuracy. There are several factors to note down why freezing didn't work:

- **Domain Similarity Reduced the Need for Freezing:** Both Cityscapes and Sensation depict similar urban scenes. The pretrained models had already captured relevant features. Freezing limited the ability to fine-tune subtle domain-specific differences.

- **Freezing Introduced Over-Regularization:** Strategies like Incremental Defrosting and CDT behaved like strong regularizers, unnecessarily limiting the model's learning capacity.

- **Heuristic Limitations in CDT:** Contribution and activation-based freezing rely on heuristics that may misidentify important segmentation layers, especially when spatial features are distributed across the network.

4.4 Limitations

Despite extensive experimentation, this work has several limitations:

- **Dataset Similarity**: The high similarity between Cityscapes and Sensation datasets reduced the benefits of freezing, as the models required minimal adaptation.
- **Model Scale**: Experiments were limited to efficient backbones (MobileNetV3, EfficientNet-B0). The effects of freezing on larger architectures (e.g., ResNet-50, Vision Transformers) remain unexplored.
- **Freezing Heuristics**: Methods like Activation-Based Freezing and CDT rely on assumptions about layer importance that may not hold for segmentation tasks, leading to suboptimal performance.
- **Real-World Complexity**: The Sensation dataset primarily contained well-defined sidewalks. Performance on challenging cases (occluded, damaged, or irregular sidewalks) was not evaluated.

4.5 Future Work

Based on these limitations, promising directions for future research include:

- **Diverse Domain Adaptation**: Evaluating freezing strategies when transferring between dissimilar domains (e.g., urban to rural environments or across geographic regions).
- **Dynamic Freezing**: Developing adaptive approaches that automatically adjust freezing patterns during training based on layer-wise contribution metrics.
- **Scaled-Up Architectures**: Investigating freezing effects in larger models and transformer-based segmentation architectures.
- **Robustness Testing**: Validating the approach on more challenging real-world scenarios, including occluded sidewalks and adverse weather conditions.

Chapter 5

Conclusion

This study investigated freezing methods for fine-tuning UNet++ models pre-trained on Cityscapes and fine-tuned to the Sensation dataset. Full fine-tuning was optimal with the highest IoU (0.5600 for MobileNetV3 and 0.5411 for EfficientNet-B0), outperforming all the freezing methods, due to the high similarity between the source and target datasets. Though Block Freezing did retain some performance (0.4713 IoU for MobileNetV3), heuristic methods like CDT and Activation-Based Freezing performed significantly worse (0.4687 and 0.4310 IoU), reflecting the application of assumptions that are not segmentation-task-friendly. Despite such observations, freezing was demonstrated to have practical utility in resource-constrained settings, retaining up to 85% baseline performance for EfficientNet-B0 with Selective Freezing. However, dataset homogeneity, model size limitations, and untested real-world complexity restrict generalizability. Future work should include studies on adaptive freezing methods for UNet++'s nested skip connections, cross-domain adaptation (e.g., urban-to-rural adaptation), and comparison on larger models (e.g., Vision Transformers) or occluded sidewalks to generalize results more for real-world urban segmentation.

List of Figures

List of Tables

Bibliography

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: (2015).

[2] Liang-Chieh Chen et al. "Rethinking Atrous Convolution for Semantic Image Segmentation". In: *arXiv preprint arXiv:1706.05587* (2017).

[3] Liang-Chieh Chen et al. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation". In: *ECCV* (2018).

[4] Marius Cordts et al. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: (2016).

[5] Gerhard Neuhold et al. "The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes". In: (2017).

[6] Andrew G. Howard et al. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". In: *arXiv preprint arXiv:1704.04861* (2017).

[7] Jason Yosinski et al. "How transferable are features in deep neural networks?" In: *NeurIPS* (2014).

[8] Zheyang Liu et al. "Layer Contribution Analysis via Dynamic Truncation for Transfer Learning". In: *IEEE TNNLS* (2024).

[9] Federica Gerace et al. "Optimal transfer protocol by incremental layer defrosting". In: (2023). arXiv: 2303.01429 [cs.LG]. URL: https://arxiv.org/abs/2303.01429.