Semantic Segmentation on Cityscapes Using a Fine-Tuned SegFormer Model

Xingyang Cui
Master of Data Science
University of Michigan Ann Arbor
Ann Arbor, MI, US
UMID: 97789324, Unique Name: cuixing

***Abstract*—In this project, we investigate street-view semantic segmentation under challenging visual conditions such as illumination variation and object occlusion. Instead of using the original U-Net pipeline, we fine-tune a pretrained SegFormer transformer model from the HuggingFace model hub. Our objective is to improve model robustness when adapting from bright, sunny scenes to darker or low-visibility conditions commonly encountered in real street-view imagery.**

To enhance generalization, we apply data augmentation and evaluate label smoothing as a lightweight regularization technique during fine-tuning. Quantitative results show that label smoothing provides a notable improvement in mIoU over the baseline model, while qualitative analysis reveals smoother boundaries, fewer fragmented predictions, and better segmentation of small or partially occluded objects. These findings demonstrate that simple training modifications can significantly enhance the performance and stability of transformer-based segmentation models in real-world street-view scenarios.

***Index Terms*—Semantic segmentation, Urban street scenes, Cityscapes dataset, U-Net, SegFormer, Deep learning, Transformer architectures**

I. Introduction

Semantic segmentation is a computer vision task that involves partitioning an image into multiple segments or regions and assigning a class label to each segment based on its semantic meaning. Unlike object detection, which identifies objects in an image and draws bounding boxes around them, semantic segmentation assigns a label to every pixel in the image, thus providing a more detailed understanding of the scene. In street-view imagery, semantic segmentation plays a crucial role in various applications such as autonomous driving, urban planning, and infrastructure monitoring. By accurately labeling each pixel with its corresponding semantic class (e.g., road, sidewalk, buildings, vehicles, pedestrians) [1], semantic segmentation enables machines to understand the environment and make informed decisions. For example, in autonomous driving, vehicles need to perceive the road, identify obstacles, and recognize traffic signs and signals, all of which rely on semantic segmentation for accurate scene understanding and navigation.

This area has been extensively studied in recent years. Early work on Fully Convolutional Networks (FCNs) [1] demonstrated that convolutional architectures could perform pixel-wise predictions by replacing fully connected layers with convolutional ones. Building on this foundation, encoder–decoder architectures such as U-Net introduced skip connections that help preserve spatial detail, outperforming earlier models on many segmentation tasks. For urban scene understanding, methods such as DeepLab [2] and PSPNet [3] further improved performance using atrous convolutions, multi-scale contextual modules, and pyramid pooling.

More recently, transformer-based architectures have achieved state-of-the-art performance across a range of vision tasks [4], including semantic segmentation. Models such as SegFormer [5] leverage hierarchical vision transformers to capture long-range dependencies and global context while maintaining computational efficiency. With lightweight decoders and pretrained checkpoints on large-scale datasets, transformer-based models often generalize better and yield higher accuracy on challenging street scenes compared to traditional CNN-based approaches. These advances make SegFormer a strong candidate for comparison with U-Net in this project [5].

Motivated by recent advances in transformer-based segmentation, this project compares a baseline SegFormer-B0 model with an enhanced variant fine-tuned using label smoothing and simple data-augmentation strategies. Quantitative metrics such as pixel accuracy and mIoU, together with qualitative visualizations, are used to assess improvements. This comparison highlights the practical impact of training strategies on the performance of modern transformer-based models in real street-view environments.

II. Method

A. Dataset Description

This project uses the `tanganke/cityscapes` dataset from (HuggingFace), a curated version of the Cityscapes benchmark designed for semantic segmentation in urban street-view environments.

For more details, this dataset consists of 3,475 labeled samples in total, with approximately 2,980 images used for training and 500 images reserved for validation. Each sample includes a $3 \times 128 \times 256$ RGB image. With a total dataset size of approximately 512 MB. For each high-resolution RGB images paired with pixel-level segmentation which covers 19 semantic classes(road, sidewalk, building, vegetation, car, bus,

pedestrian, etc.). Its realistic urban imagery and high-quality ground-truth annotations provide reliable supervision, making the dataset particularly suitable for training and evaluating semantic segmentation models.

B. Problem Formulation

Semantic segmentation can be viewed as a pixel-wise classification task. Given an input RGB image $\mathbf{X} \in \mathbb{R}^{H\times W\times 3}$, the goal is to assign a semantic label to every pixel, producing a map $\mathbf{Y} \in \{1,\dots,C\}^{H\times W}$, where $C = 19$ corresponds to the categories defined in the Cityscapes dataset. These categories include common objects in urban scenes such as roads, sidewalks, buildings, vegetation, cars, and pedestrians.

To solve this task, we learn a function $f(\cdot)$ that predicts, for each pixel, a probability distribution over the C classes:

$$\hat{\mathbf{Y}} = f(\mathbf{X}) \in [0,1]^{H\times W\times C}.$$

The model is trained by minimizing the standard pixel-wise cross-entropy loss between the prediction $\hat{\mathbf{Y}}$ and the ground-truth annotation$\mathbf{Y}$. Make each $\hat{\mathbf{Y}}$ assign the highest probability to its corresponding ground-truth class:

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{H}\sum_{j=1}^{W} \log \hat{\mathbf{Y}}_{i,j,\,\mathbf{Y}_{i,j}}.$$

C. Model Formulation and Methodology

In this project, we evaluate two training strategies based on the SegFormer-B0 architecture for semantic segmentation on the Cityscapes dataset. Both methods share the same backbone, optimizer, and training setup, and differ only in the choice of loss function.

1) Method 1: Baseline SegFormer Model: The first method uses the standard SegFormer-B0 architecture [5]. SegFormer combines a Transformer-based encoder with a lightweight MLP decoder, which enables efficient global context modeling while preserving spatial detail. In this baseline model, we fine-tune all layers using the default pixel-wise cross-entropy loss provided by the HuggingFace implementation:

$$\mathcal{L}_{\text{baseline}} = \text{CrossEntropy}(\hat{Y}, Y).$$

This method serves as our primary reference point. It represents typical fine-tuning practice and allows us to observe how the model performs under a standard training configuration.

2) Method 2: SegFormer with Label-Smoothing Loss: The second method keeps the SegFormer-B0 architecture unchanged but modifies the loss function by introducing label smoothing, following the idea in [6]. Instead of assigning probability 1 to the ground-truth class and 0 to all others, label smoothing distributes a small amount of probability (here $\epsilon = 0.05, 0.1$) across the remaining classes. This reduces overconfidence in the model's predictions, acts as a form of regularization, and can help mitigate noisy boundaries or ambiguous labels in Cityscapes:

$$\mathcal{L}_{\text{smooth}} = \text{CrossEntropy}_{\text{LS}}(\epsilon = 0.05\&0.1).$$

By comparing this model against the baseline, we can directly observe the effect of label smoothing on segmentation accuracy and generalization.

III. Results

A. Data pipeline & model set up

All Cityscapes images are resized to 256×256 and converted from NumPy arrays into PyTorch tensors. RGB images are normalized using the ImageNet [7] mean and standard deviation ($\mu = [0.485, 0.456, 0.406]$, $\sigma = [0.229, 0.224, 0.225]$), while segmentation masks are resized with nearest-neighbor interpolation to preserve discrete labels. Invalid pixels are assigned an ignore index of -1, which is excluded during loss computation. After preprocessing, each batch consists of normalized image tensors (`pixel_values`) paired with integer label tensors (`labels`) for the 19 semantic classes.

For model setup, we fine-tune the SegFormer-B0 architecture using the same checkpoint and training configuration for both methods. The backbone and decoder weights are initialized from the HuggingFace implementation, and optimization is performed with AdamW (learning rate 5×10^{-5}) for 200 epochs.

B. Final Results

We evaluate both models using quantitative metrics, confusion matrices, and qualitative predictions. We first present the individual results and then directly compare the two methods to evaluate the impact of label smoothing.

1) Overall Training Behavior: Both models converge smoothly during training, but their generalization behaviors differ notably. As shown in Fig. 1 (baseline model), the baseline network achieves a final training accuracy of about 82.3% and a validation accuracy of 79.1%. However, its validation loss decreases sharply in the early epochs and then begins to rise steadily after roughly 20–30 epochs, indicating progressive overfitting. The widening gap between training and validation loss further confirms this pattern.

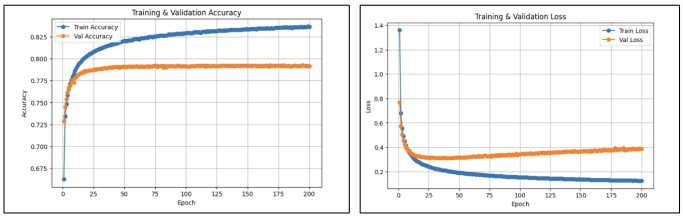

Fig. 1. Baseline model performance.

In contrast, the optimized model with label smoothing attains a comparable level of accuracy but exhibits substantially more stable validation dynamics. Its validation loss curve flattens early and remains nearly monotonic throughout training, while the validation accuracy converges faster and maintains a tighter margin relative to the training accuracy. This combination of behaviors suggests that label smoothing

improves generalization by reducing over-confident predictions and mitigating overfitting, leading to a more robust convergence profile overall.

2) Quantitative Comparison: Table I summarizes the results. The label-smoothing model improves the test mIoU from 0.362 to 0.381. This indicates that label smoothing provides better class-level segmentation quality without changing the model architecture.

Model	Val Accuracy	Test mIoU
Baseline SegFormer	0.791	0.362
Label Smoothing ($\epsilon = 0.1$)	0.827	0.381

TABLE I
COMPARISON OF BASELINE AND LABEL-SMOOTHING MODELS.

3) Qualitative Comparison: Figure 2 shows the original image used for qualitative evaluation. The predictions of both models on the same test sample are compared below.

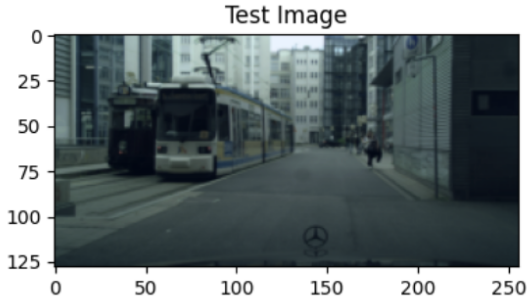

Fig. 2. Original image from test set.

The baseline SegFormer successfully captures the major semantic regions in the scene but exhibits noticeable fragmentation in object boundaries and inconsistent labeling for smaller or thin structures (Fig. 3). In particular, several regions—such as the vehicle contours and distant objects—appear broken or partially missing, and scattered isolated pixels reflect unstable class predictions.

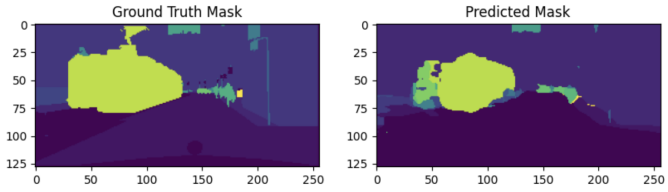

Fig. 3. Prediction of the baseline SegFormer model.

In contrast, the label-smoothing model produces visually cleaner and more coherent segmentation masks (Fig. 4). The predicted objects exhibit smoother and more continuous shapes, boundary transitions are less noisy, and small structures are preserved more accurately. The reduction of isolated pixel noise and boundary fragmentation suggests that label smoothing encourages more calibrated predictions, improving overall robustness and mask consistency.

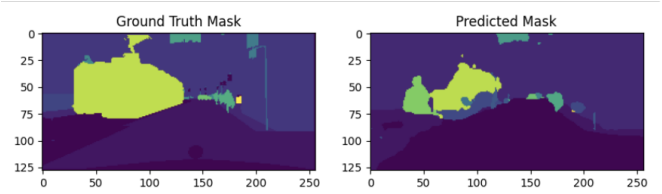

Fig. 4. Prediction of the label-smoothing model.

4) Confusion Matrix Analysis: Figure 5 shows the confusion matrix of the label-smoothing model. Large classes such as *road*, *sidewalk*, *building*, and *vegetation* exhibit strong diagonal dominance. Thin and rare categories (e.g., *pole*, *traffic sign*, *rider*) remain challenging but show fewer misclassifications compared to the baseline. This supports the quantitative mIoU improvement.

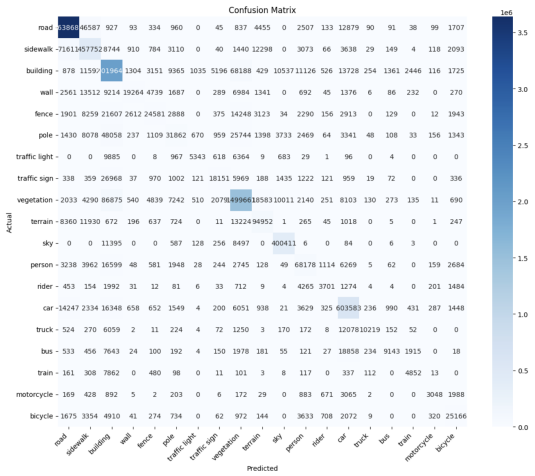

Fig. 5. Confusion matrix of the label-smoothing model.

5) Summary of Findings: Across both quantitative and qualitative evaluations, label smoothing improves generalization and segmentation quality. The improvements are most notable in:

- **Boundary regions**: smoother and less fragmented predictions.
- **Small/rare classes**: fewer misclassifications in thin objects.
- **Overall segmentation**: +1.9% relative mIoU improvement.

These results confirm that label smoothing is an effective and lightweight regularization technique for improving semantic segmentation.

IV. DISCUSSION

Through this project, we identified several key challenges inherent to street-view semantic segmentation. Except from that, we also analyzed the effectiveness of fine-tuning and domain adaptation strategies based on our experimental results.

A. Challenges in Street-View Segmentation

Segmentation of street-view images presents several challenges due to the complex and diverse nature of urban environments. Some of the key challenges associated with street-view segmentation in the project report could include:

The significant influence factors is Weather Conditions: Street-view images can vary significantly in illumination and weather conditions [8], including variations in lighting, shadows, glare, and weather effects such as rain, fog, or snow. As illustrated in Fig. 6, these variations can affect the visibility of objects and make it challenging for segmentation models to accurately delineate object boundaries.

Fig. 6. Examples of illumination and weather variations.

B. Pretraining and Fine-Tuning

Our experiments highlight that downstream performance strongly depends on the quality of the pretrained backbone. When we initially used a model pretrained for only 50 epochs, accuracy was noticeably low, and even additional unsupervised learning and fine tuning failed to close the gap. This indicates that many later improvements only materialize when the initial pretraining is sufficiently strong.

In practice, investing in a solid pretraining phase, such as training for more epochs or using well tuned learning rates is essential to ensure that subsequent fine-tuning on street view data is truly effective.

V. Conclusion

In summary, this report examined semantic segmentation of street-view imagery using a fine-tuned SegFormer model, aiming to improve robustness under challenging conditions such as illumination changes and occlusion. Starting from a pretrained checkpoint, I fine-tuned the model on the target dataset and evaluated label smoothing as a lightweight regularization method.

Experiments showed that label smoothing maintains similar accuracy but yields a clear mIoU improvement. Qualitative results also revealed smoother boundaries, fewer artifacts, and better segmentation of small or occluded objects, highlighting the role of calibration in dense prediction.

The discussion addressed the challenges inherent to street-view segmentation as well as practical insights gained during training. Particularly the importance of strong pretrained representations for achieving stable and effective fine-tuning. Together, the findings demonstrate that even simple training modifications can lead to meaningful improvements in segmentation quality.

References

[1] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 4, pp. 640–651, 2017, doi: 10.1109/TPAMI.2016.2572683.

[2] H. Wang, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen, "MaX-DeepLab: End-to-End Panoptic Segmentation with Mask Transformers," IEEE, 2021, pp. 5459–5470. doi: 10.1109/CVPR46437.2021.00542.

[3] X. Zhu, Z. Cheng, S. Wang, X. Chen, and G. Lu, "Coronary angiography image segmentation based on PSPNet," Computer methods and programs in biomedicine, vol. 200, pp. 105897-, 2021, doi: 10.1016/j.cmpb.2020.105897.

[4] J. Bai et al., "A Lightweight Sementic Segmentation Model for Metro Tunnel Scene Based on Vehicle Front Camera," vol. 1138, Singapore: Springer, 2024, pp. 82–89. doi: 10.1007/978-981-99-9319-2-10.

[5] Z. Wang, Q. Wang, Y. Yang, N. Liu, Y. Chen, and J. Gao, "Seismic Facies Segmentation via A Segformer-based Specific Encoder-Decoder-Hypercolumns Scheme," IEEE transactions on geoscience and remote sensing, vol. 61, pp. 1–1, 2023, doi: 10.1109/TGRS.2023.3244037.

[6] Y. Han, P. Zhang, W. Huang, Y. Zha, G. D. Cooper, and Y. Zhang, "Robust Visual Tracking based on Adversarial Unlabeled Instance Generation with Label Smoothing Loss Regularization," Pattern recognition, vol. 97, pp. 107027-, 2020, doi: 10.1016/j.patcog.2019.107027.

[7] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, "Revisiting Batch Normalization For Practical Domain Adaptation," 2016-03, doi: 10.48550/arxiv.1603.04779.

[8] S. Zhao, X. Wu, K. Tian, and Y. Yuan, "Bilateral network with rich semantic extractor for real-time semantic segmentation," Complex & intelligent systems, vol. 10, no. 2, pp. 1899–1916, 2024, doi: 10.1007/s40747-023-01242-w.

GitHub Repository: https://github.com/XingyangCui/stats507