Architectural Trade-offs in Semantic Segmentation Under Environmental Stress and Embedded Constraints

Jonathan Wang, Raymond Chu

Code and Data Availability: https://github.com/raychu23/Driving-Scene-Segmentation

1. Introduction

Semantic segmentation is a core perception task in autonomous driving systems, providing pixel-level scene understanding for downstream components such as path planning and obstacle avoidance. Unlike image classification, segmentation requires preserving fine spatial detail across the entire image, making it computationally demanding. In deployed systems, segmentation models must operate under fixed latency, memory, and power constraints on embedded platforms, which limits both model size and allowable inference cost.

Separately, real-world driving environments introduce significant variability in segmentation due to weather and lighting conditions such as rain, fog, and nighttime illumination. These conditions degrade image quality and increase visual ambiguity, leading to higher error rates in perception systems (Ulku & Akagunduz, 2019).. Because many failures in autonomous driving occur under such conditions, understanding how segmentation performance changes as visual quality deteriorates is critical for system reliability.

From an engineering perspective, addressing these challenges is often constrained by deployment realities. Once a perception stack is integrated onto a target platform, designers typically cannot rely on retraining, data augmentation, or computationally expensive robustness techniques. Instead, robustness is commonly addressed during model selection, where different segmentation architectures are evaluated based on their accuracy-efficiency tradeoffs and their behavior under non-ideal conditions (Shaik et al., 2023). In contrast, many research evaluations emphasize peak accuracy or architectural complexity without fully accounting for the size, latency, and deployment constraints that shape real embedded systems.

Motivated by this workflow, we focus on comparing segmentation architectures rather than proposing new training methods. In this work, we evaluate lightweight and high-capacity segmentation models under a shared training setup and analyze their accuracy, latency, and inference efficiency. Using driving datasets with environmental annotations, we examine how different architectures respond to challenging conditions, emphasizing comparative trends that inform practical model selection for autonomous driving systems. This investigation aims to answer the following research question: How do different semantic road segmentation models remain usable under challenging weather conditions. We hypothesize that lightweight architectures, while achieving lower peak accuracy, will exhibit more stable performance degradation under adverse conditions compared to higher-capacity models.

2. Background

Semantic segmentation is a dense prediction task in which each pixel of an input image is assigned a semantic class label. Given an RGB image, a segmentation model outputs a spatially aligned label map that identifies objects such as roads, vehicles, pedestrians, and buildings at the pixel level. Ground-truth annotations are provided as segmentation masks, where each pixel stores its corresponding class index. Model performance is evaluated by comparing predicted and ground-truth masks using mean Intersection-over-Union (mIoU), which measures the overlap between predicted and true regions for each class and averages across

classes. This metric emphasizes spatial accuracy and boundary alignment, making it well-suited for evaluating segmentation quality.

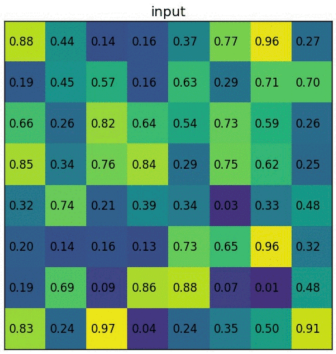


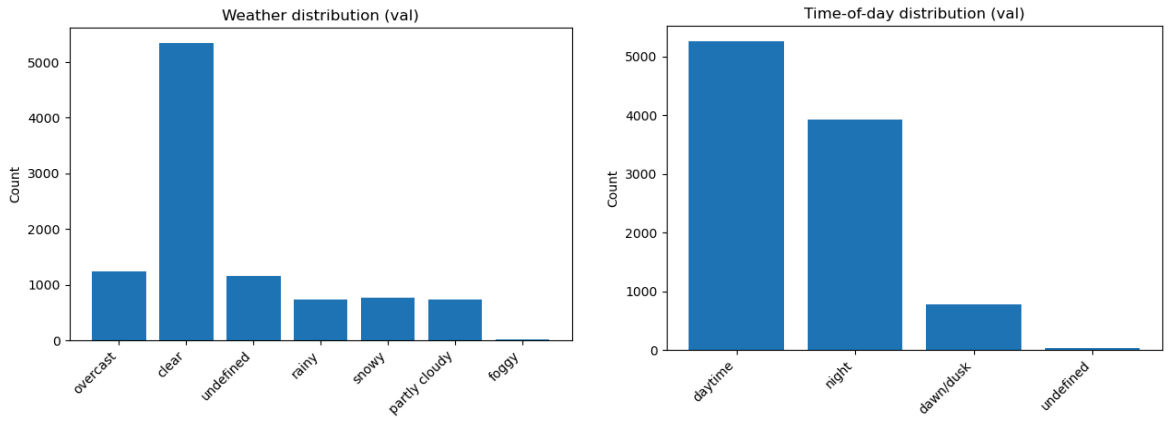
Figure 1: Illustration of a semantic segmentation operation at the pixel level with each cell containing a real-valued scalar. Subsequent figures display a single sample of an input image paired with its annotated and predicted segmentation masks.

3. Methods

3.1 Dataset and Exploratory Data Analysis

We use the semantic segmentation subset of the BDD100K dataset, which contains pixel-level annotations for urban driving scenes captured from dashcam video. From the full dataset, approximately 10,000 labeled images are available with corresponding segmentation masks. Each image has an original resolution of 1280 × 720 and includes annotations for 19 semantic classes.

As part of exploratory data analysis, we examined the distribution of available metadata across images through the included JSON file. As expected for real-world driving data, clear weather and daytime scenes dominate the dataset, while adverse conditions such as rain, fog, and nighttime driving are underrepresented. This skew reflects natural data collection processes and is preserved during training to avoid introducing artificial distributions or biasing the learning process toward rare conditions. And we visually verify image-mask alignment and annotation consistency by inspecting random image-mask pairs across the dataset prior to training.



Figures 2 & 3: Distribution of environmental conditions in the BDD100K validation split. Figure 2 shows the distribution of annotated weather conditions, while Figure 3 displays the distribution of time-day labels.

3.2 Data Splits and Evaluation Protocol

We follow the official training and validation splits provided with the BDD100K semantic segmentation dataset at a 70% to 30% split, respectively. To evaluate robustness under adverse environmental conditions, we additionally perform evaluation on the ACDC (The Adverse Conditions Dataset dataset) (Sakaridis et al. 2021). ACDC serves as an external test set and contains 2006 pixel-level annotations equally distributed under four challenging conditions of fog, rain, snow, and nighttime conditions. This dataset is not used during training or validation and is employed exclusively to assess performance degradation under environmental stress. This design isolates robustness evaluation from training-time distribution manipulation and reflects realistic deployment constraints.

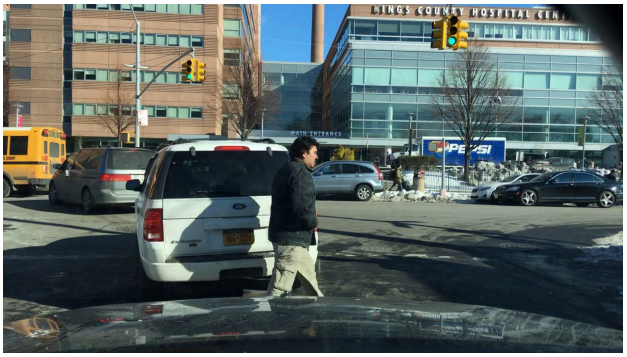
3.3 Feature Engineering Pipeline

After completing the initial data extraction, our feature engineering pipeline consists of three core processes: file alignment, image transformations, and mask encoding. To ensure correct pairing between inputs and labels, RGB images and their corresponding segmentation masks are stored in separate directory trees. Image-mask pairs are constructed by matching filenames and verifying the existence of the corresponding annotation mask for each RGB frame. Samples with missing annotation masks are excluded from the dataset, preventing misaligned supervision during training.

To ensure the training process remains stable, we resize all images from their original resolution (1280 × 720) to 512 × 512 and normalize pixel values using the ImageNet mean and standard deviation, ensuring compatibility with pretrained backbones such as ResNet and MobileNet. These transformations are applied consistently across training and validation data. Segmentation masks are resized using nearest-neighbor interpolation to preserve discrete class labels. Masks are represented as two-dimensional matrices in which each entry corresponds to an integer class index. Pixels labeled with the dataset's ignore value are mapped to an ignore index during training to ensure they do not contribute to the loss function.

3.4 Preprocessing Steps

Our preprocessing pipeline ensures consistent and reproducible inputs across all segmentation models to the BDD100K dataset. This pipeline operates on the individual sample level, beginning with paired raw RGB dashcam images and their corresponding semantic segmentation annotations. Using a single image–mask pair as a conceptual illustration, we demonstrate how pixel-level correspondence between the input image and its annotation is preserved throughout the preprocessing pipeline. Such an example serves as a visual reference for all subsequent preprocessing steps applied at scale.



Figures 4 & 5: *Original RGB dashcam image* (left), *Corresponding semantic segmentation annotation mask for the same scene* (right). Figure 4 captures an image of a complex urban driving scene of vehicles, pedestrians, traffic signals, and buildings under natural lighting conditions. In Figure 5, each pixel is assigned a class label for each object represented by distinct colors, providing ground-truth supervision for segmentation training.

The next step involves an application of the same preprocessing procedure to all 10,000 labeled images selected from the segmentation annotations. We perform image extractions by loading RGB dashcam frames captured under varied conditions at their original resolution, then apply the necessary transformations (resize to 512 x 512) to match uniform input dimensions. This ensures that we preserve spatial correspondence between image pixels and mask labels. This yields the final outputs of shape (3, 512, 512) for our image tensor, and (512, 512) for our mask tensor. During the final step of constructing a dataset object from our preprocessed image-mask pairings, we apply shuffling to randomize the sample order of each epoch during training to prevent learning sequence-dependent patterns. Our batching groups samples into sizes of 4 with an image batch size of (4, 3, 512, 512) and a mask batch of size (4, 512, 512). These batches are sequentially streamed to the GPU until all samples are processed, constituting one training epoch. Throughout the entire training process schedule, we repeat our batching and process for a total of 5 epochs. We use the cross-entropy loss with an ignore index of −100 to exclude unlabeled pixels from optimization. Optimization is performed using the Adam optimizer with a learning rate of 1×10^{-4}. The batch size of 4 reflects realistic memory constraints and aligns with deployment-oriented experimentation.

3.5 Deep Learning-Based Segmentation Models

To evaluate how model capacity affects robustness to adverse weather conditions, we compared three deep learning segmentation architectures spanning a range of complexity and

deployment targets. In this process, two fundamental challenges of semantic segmentation must be addressed.

First, semantic segmentation requires preserving spatial detail. Most convolutional neural networks reduce image resolution through pooling or strided convolutions to improve efficiency and capture high-level features. While this is effective for recognizing what objects are present, it removes fine-grained spatial information, making it harder to precisely determine where object boundaries lie. This loss of spatial resolution can lead to blurred edges and inaccurate predictions, especially for thin or irregular structures.

Second, segmentation models must handle objects at multiple scales within the same image. Large objects such as roads or buildings benefit from wide contextual understanding, while smaller objects like pedestrians or traffic signs require high-resolution features. As networks become deeper, they gain larger receptive fields that improve global context but may suppress small-scale details. Effective segmentation architectures must balance global semantic understanding with local spatial precision.

Baseline CNN

The baseline convolutional network applies sequential convolution and downsampling layers without explicit mechanisms to recover lost spatial detail. While it can learn high-level semantic features, it often produces coarse segmentation outputs with poorly defined boundaries, highlighting the limitations of naive architectures for dense prediction tasks (Long et al. 2015).

MobileNetV3

MobileNet-based models prioritize efficiency by using depthwise separable convolutions to reduce computation and memory usage. This makes them well-suited for resource-constrained environments, but aggressive downsampling can lead to reduced spatial precision and smoother, less detailed segmentation masks (Howard et al., 2019).

DeepLabV3

DeepLabV3 addresses both spatial resolution loss and multi-scale object challenges by using dilated (atrous) convolutions, which expand the receptive field without reducing feature map resolution. Its Atrous Spatial Pyramid Pooling module aggregates contextual information at multiple scales, enabling accurate segmentation at the cost of higher computational complexity (Chen et al. 2018).

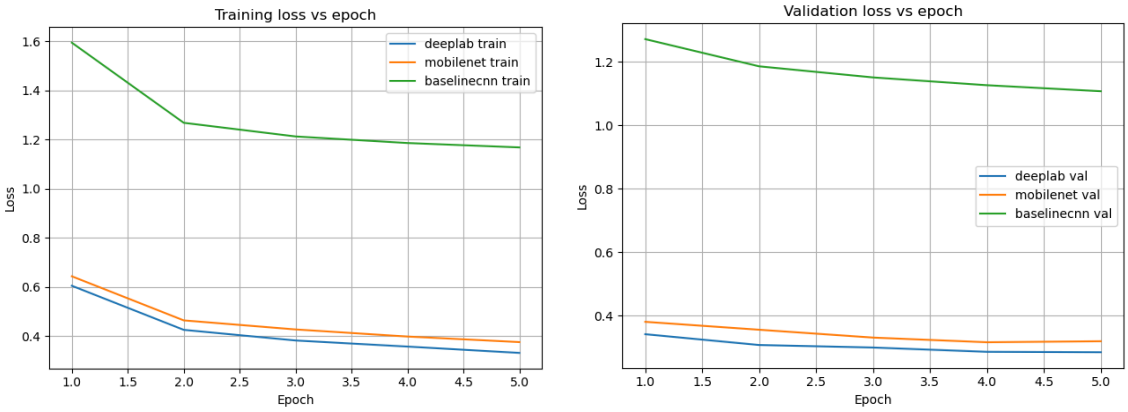
4. Results

4.1 Training and Validation Behavior

Figures 6 and 7 (below) show the training and validation loss curves, respectively, for all evaluated models over five epochs. Across architectures, training loss decreases steadily, while validation loss follows a similar trend without significant divergence. This behavior suggests stable optimization and no strong evidence of overfitting within the limited training schedule.

Higher-capacity models such as DeepLabV3 converge to lower absolute loss values, while lightweight architectures exhibit faster initial loss reduction but plateau at higher final loss

values. Despite these differences, all models demonstrate consistent convergence behavior under identical preprocessing and optimization settings.



Figures 6 & 7: A training loss curve (left) and a validation loss curve (right) across five epochs for all evaluated segmentation models

4.2 Overall Segmentation Performance

Table 1 summarizes segmentation performance across all models on the BDD100K validation set using mean Intersection-over-Union (mIoU). DeepLabV3 achieves the highest overall accuracy, followed by MobileNet-based segmentation, while the baseline CNN substantially underperforms both modern architectures. The observed performance gap generally aligned with the architectural expectations. DeepLabV3 achieves substantially higher segmentation accuracy, and is able to perform segmentation at an acceptable level. MobileNet had a surprisingly competitive value compared to expectations despite its efficiency-oriented architecture. In contrast, the baseline CNN underperforms, which is expected given its lack of explicit mechanisms to preserve or recover spatial detail after downsampling.

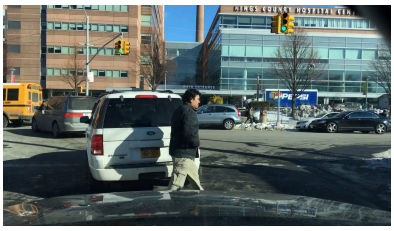
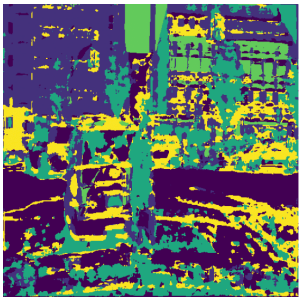
Model	mIoU	Images
Raw Image	N/A	
Ground Truth	1.000	
DeepLabV3	0.423	
MobileNetV3	0.388	
Baseline CNN	0.210	

Table 1: Overall segmentation performance on the validation set measured by mIoU

4.3 Accuracy–Latency Tradeoff

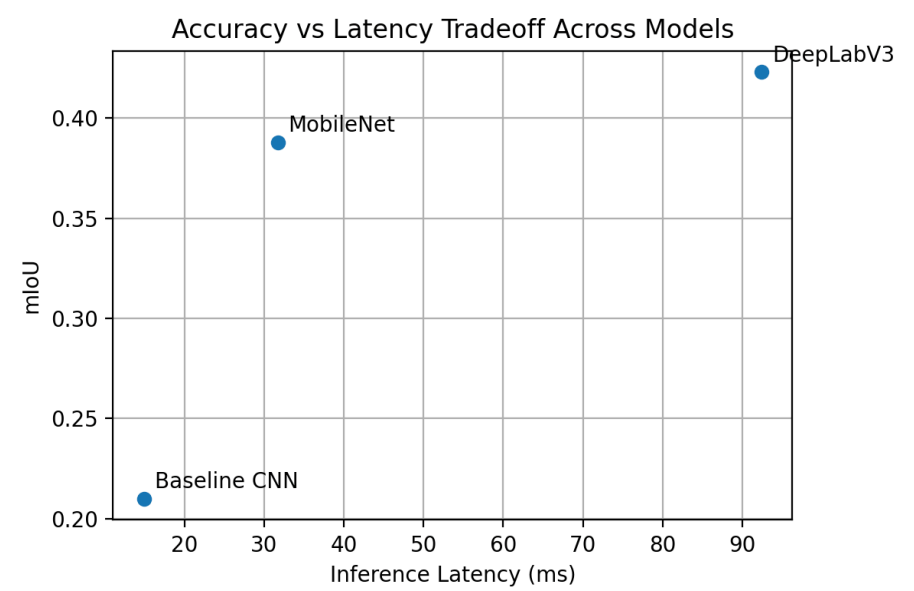


Figure 8: Comparison of mIoU relative to inference latency (ms) across the three evaluated architectures.

This visualization illustrates the tradeoff between predictive performance and computational efficiency. Although DeepLabV3 achieves the highest accuracy, its latency places it at a severe disadvantage for time-constrained systems. MobileNet occupies a favorable middle ground, delivering strong segmentation accuracy (0.388 mIoU) while maintaining significantly lower latency. The baseline CNN achieves the fastest inference but at the expense of substantial accuracy degradation.

These results confirm the expectation that higher-capacity architectures trade latency for accuracy, while efficient models offer improved responsiveness with modest performance loss.

5. Discussion

Our work emphasizes the importance of assessing semantic segmentation models for autonomous driving through architecture-level analysis under environmental stress. By separating training from condition-specific evaluation, our methodology avoids conflating robustness with data augmentation or retraining strategies, instead isolating how architectural design choices influence generalization under realistic deployment constraints.

Our results show that performance degradation under adverse conditions exposes architectural differences that are not apparent from standard validation metrics alone. High-capacity models such as DeepLabV3 achieve strong peak accuracy under nominal conditions, but their increased computational complexity is substantial relative to the observed accuracy gains. In contrast, lightweight architectures offer a more favorable balance between accuracy and efficiency than expected, likely because their simpler designs reduce sensitivity to environmental noise and avoid over-reliance on fine-grained features that degrade under poor visibility

5.1 Limitations and Extensions

We note several limitations of our study. First of all, our training schedule was made intentionally limited to ideally reflect realistic deployment experiments, and extended training may improve the absolute performance across all models. Secondly, inference latency is a useful proxy for efficiency, but it does not fully capture the deployment cost. Future works would need to allocate larger amounts of memory and computational power to produce a more comprehensive system level analysis. Finally, while our study focuses on the selection of different architectures, complementary studies involving approaches such as data-pruning, condition-aware adaption, or uncertainty quantification (Miandashti et al. 2024) could contribute to further robustness within the same evaluation framework.

We also note certain limitations in our evaluation, as it primarily reports mean performance for each weather condition of rain, fog, snow, and nighttime. Future work could extend this by reporting class-wise mIoU by condition and relative degradation plots that normalize performance under adverse conditions against nominal settings. Such analyses would further clarify how different architectures degrade and which failure modes dominate under environmental stress. Additionally, inference efficiency in this work is measured using PyTorch runtime on a general-purpose device. While sufficient for relative comparison, this does not fully capture deployment behavior on embedded platforms. Future evaluation could incorporate inference frameworks and toolchains designed for embedded systems to further strengthen the deployment relevance of the results.

Overall, our study conveys that architecture choice plays a central role in determining both performance stability and deployability in semantic segmentation systems. By considering the tradeoffs of accuracy, latency, and robustness across an adverse set of environmental conditions, we provide a practical evaluation that aligns a conceptual model selection to simulate operationally realistic conditions of safety-critical autonomous vehicle systems.

References

Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV).

Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., & Girdhar, R. (2021). Per-pixel classification is not all you need for semantic segmentation. In Advances in Neural Information Processing Systems (NeurIPS).

Goan, E., & Fookes, C. (2023). Uncertainty in real-time semantic segmentation on embedded systems. arXiv. https://doi.org/10.48550/arXiv.2301.01201

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., & Adam, H. (2019). Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).

Kou, W.-B., Zhu, G., Ye, R., Lin, Q., Ren, Z., Tang, M., & Wu, Y.-C. (2024). Generalizable autonomous driving system across diverse adverse weather conditions. arXiv. https://doi.org/10.48550/arXiv.2409.14737

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Miandashti, H. S., Zou, Q., & Brenner, C. (2024). Calibrated and efficient sampling-free confidence estimation for LiDAR scene semantic segmentation. arXiv. https://doi.org/10.48550/arXiv.2411.11935

Sakaridis, C., Dai, D., & Van Gool, L. (2021). ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (pp. 10765–10775). https://doi.org/10.1109/ICCV48922.2021.01057

Shaik, F. A., Malreddy, A., Billa, N. R., Chaudhary, K., Manchanda, S., & Varma, G. (2023). IDD-AW: A Benchmark for Safe and Robust Segmentation of Drive Scenes in Unstructured Traffic and Adverse Weather. arXiv. https://doi.org/10.48550/arXiv.2311.14459

Ulku, I., & Akagunduz, E. (2019).
A survey on deep learning-based architectures for semantic segmentation on 2D images. arXiv.
https://arxiv.org/abs/1912.10230

Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F.,
Madhavan, V., & Darrell, T. (2018, May 30).
BDD100K: A large-scale diverse driving video database.
Berkeley AI Research Blog. https://bair.berkeley.edu/blog/2018/05/30/bdd/