Semantic Segmentation for Self-Driving Cars

Authors
Xinpeng Shan, *shanxinp*
Dechen Han, *handeche*
Wendy Yusi Cheng, *chengw54*
Shi Tang, *tangsh29*

Abstract

Semantic segmentation is essential for self-driving as it provides detailed pixel-level classifications for safe navigation. In this paper, we explore advancements in semantic segmentation using three deep learning models: FCN, U-Net and DeepLab. Our study focuses on adapting and refining these models to enhance their accuracy and efficiency under diverse and challenging conditions, such as varying lighting and complex urban landscapes. We conclude by evaluating the models' performance on both known and unseen datasets to find their strengths and areas for improvement. Our findings indicate that U-Net and DeepLab V3 model can perform a better segmentation compare with the baseline FCN model, but there is room for improvement in their generalization. This research highlights the potential and challenges in applying deep learning techniques to improve the reliability and safety of autonomous vehicles.

1 Introduction

Autonomous driving technology fundamentally transforms how vehicles operate by eliminating the need for human control and decision-making. To enable this, a vehicle must have an advanced understanding of its surroundings, akin to or surpassing human perception. This capability is built upon several layers of technological innovation, among which semantic segmentation plays a crucial foundational role.

Semantic segmentation enables computers to interpret and understand the visual world. In the context of autonomous driving, semantic segmentation involves the detailed and precise classification of every pixel in an image captured by the vehicle's cameras. Each pixel is assigned to a specific category such as road, pedestrians, other vehicles, and traffic signs. This classification is not merely about detecting objects but understanding the role and space each object occupies in the environment. This detailed pixel-level understanding is important for a vehicle to navigate safely through its surroundings.

Challenges in Semantic Segmentation:

1. **Varying Lighting Conditions:** Different times of the day and weather conditions can dramatically change how a camera captures images. Low light, bright sunlight, or reflections can obscure important details and lead to misclassifications.
2. **Diversity of Objects:** The real world presents an almost infinite variety of objects in different shapes, sizes and colours. New object types (such as unusual vehicle designs or uncommon road signs) and complex scenes (like construction zones) add to the challenge.

Improving the accuracy and efficiency of semantic segmentation models is crucial for the development of safer and more reliable autonomous vehicles. Therefore, this project aims to explore and improve the state-of-the-art in semantic segmentation for autonomous driving by leveraging the strengths of FCN, DeepLab, and U-Net. Specifically, we intend to:

RQ1: Adapt and refine FCN, DeepLab, and U-Net for semantic segmentation on autonomous driving cars, and compare the performance across these three model architectures.

RQ2: Compare model performance in unseen dataset including diverse and challenging scenarios like varying weather conditions and urban landscapes.

2 Background

2.1 Applications of Semantic Segmentation

Semantic segmentation is a fundamental but demanding computer vision task, where the goal is to assign each pixel of an image with a category label. Semantic segmentation can be beneficial in many real-world applications, including autonomous driving cars (1), healthcare diagnosis (2), and treatment analysis (3). Semantic information at the pixel level aids intelligent systems in understanding spatial locations and making critical decisions. In this paper, we focus on the evolution and comparison of deep-learning techniques tailored for autonomous driving tasks.

2.2 Deep Learning Methods

Several deep learning architectures have been developed for semantic segmentation. This paper focuses on Fully Convolutional Networks (FCN) (4), DeepLab (5), and U-Net (6). FCNs have pioneered the use of convolutional neural networks for pixel-wise prediction, enabling end-to-end training and inference. As a baseline model, FCN is essential for understanding the foundational elements of convolutional networks in segmentation tasks, providing a comparison point for more state-of-the-art architectures. Building upon the FCN, U-Net modifies and extends this architecture by adding upsampling techniques to work on a few training images and obtaining more precise segmentations. Although originally designed for biomedical segmentation, U-Net adapts well to autonomous driving tasks due to the precise localization it brings, which is indispensable for distinguishing between closely spaced objects on roads. In comparison, DeepLabV3 pays much attention to capturing fine details and contextual information using atrous convolution fully connected Conditional Random Fields. This capability is quite crucial in complex urban environments.

Other alternative segmentation methods like SegNet (7) and Mask R-CNN (8) were considered; however, they were not selected in this paper due to their respective limitations. SegNet, while useful for its encoder-decoder architecture, generally underperforms compared to newer models in terms of efficiency and accuracy. While Mask R-CNN is effective in instance segmentation, a fine-grained extension of traditional semantic segmentation, its application is outside of the scope of this paper, which is on traditional semantic segmentation. Additionally, Mask R-CNN is highly dependent on a specific annotated input dataset, which does not conform to our input dataset.

The choice of FCN, U-Net, and DeepLab V3 (9) for the study is, thus, justified by their previous benchmarked effectiveness and their unique advantages toward the required tasks in autonomous driving: FCN in the baseline performances, U-Net, especially in its ability to create detailed segmentation in the urban setting, and DeepLab in its ability leveraging context information. This comparative approach allows for a comprehensive evaluation of how each model performs under the practical challenges posed by autonomous vehicle environments.

3 Model Architecture

3.1 Datasets

For training our semantic segmentation model, we utilized two datasets derived from the CARLA simulator. The first dataset, sourced from the Lyft-Udacity Challenge on Kaggle, includes diverse scenarios such as sunny, rain, cloudy city landscapes, and more. The data has 5 sets of 1000 images and corresponding labels in 13 classes. We divided this dataset into training (70%), testing (15%), and validation (15%) sets to train, test and optimize the models.(10) The second dataset, used for addressing Research Question 2 (RQ2), specifically designed for testing the model's generalization on unseen test datasets in various scenarios. The dataset can be found at (11).

We employ synthetic datasets for model training as this addresses several limitations inherent to real-world data. Collecting and labelling real data is costly, and often, insufficient annotations are available to train deep neural networks effectively. In contrast, synthetic data can be artificially generated and is at least similar to real-world data to some extent. This type of data also permits the manipulation of variables to enhance diversity in the dataset. Such as weather conditions and the inclusion of various objects like cars and pedestrians. Moreover, synthetic images are automatically annotated, which reduces the need for time-consuming manual labelling. Consequently, synthetic datasets are widely used for training semantic segmentation models within self-driving car systems.

3.2 Input and Compilation

FCN, U-Net and DeepLab handle the same input images of size 256x256x3. This means each image is 256 pixels in height and width with three color channels. For all three models, we use the Adam optimizer to optimize the training and apply Sparse Categorical Crossentropy as the loss function, which is suitable for segmentation tasks to classify each pixel into one of several categories. We use ReLU as the activation function to introduce non-linearity into the models. Additionally, an accuracy metric that measures the percentage of pixels correctly classified is used to track their performance.

3.3 Fully Convolutional Network (FCN)

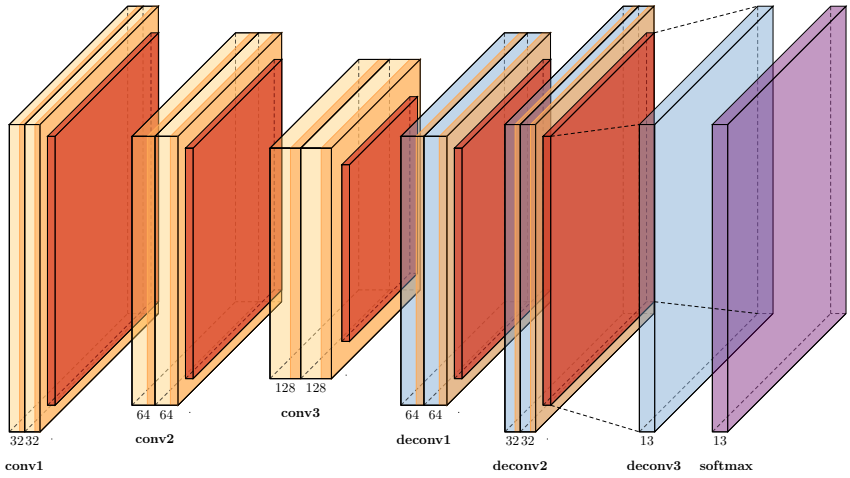

Figure 1: FCN Model built with PlotNeuralNet (12)

FCN's strength lies in its encoder-decoder structure, where the encoder progressively captures hierarchical features and then the decoder localizes and assigns a class to each pixel. The architecture of our FCN model is shown in Figure 1.

The Encoder part has three convolutional blocks. The first block contains two layers that each use 32 small filters (3x3) to capture features from the image. Each layer applies batch normalization and an activation function to keep the calculations simple and efficient. The second block also has two layers, but it uses 64 filters to capture more complex features. The third block further increases the complexity and detail by using 128 filters.

The Decoder begins with a block that upscales the outputs from the last encoder block and combines them with features from the second encoder block. This decoder block includes a convolutional layer with 64 filters, batch normalization and ReLU activation. The second decoder block upscales further and combines these features with those from the previous block, using 32 filters. The output layer is a convolutional layer with 13 filters, one for each class in the segmentation task.

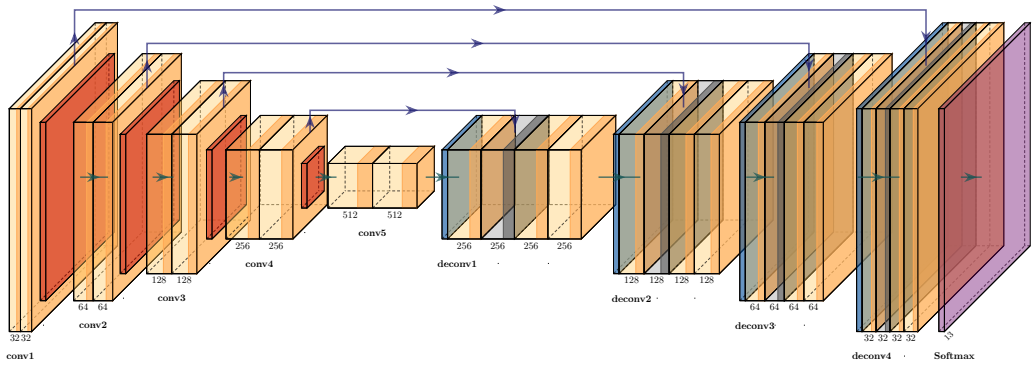

Figure 2: U-Net Model built with PlotNeuralNet (12)

3.4 U-Net

U-Net further enhances the FCN architecture by introducing a symmetric expanding path that complements the contracting path, thereby improving the network's ability to localize fine structures and yield more precise segmentations. Our model architecture is shown in Figure 2.

In the Encoder (Contractive Path), the model applies a series of convolutional blocks that progressively increase in filter size, where the number doubles with each subsequent block, beginning with 32 filters and ending up to 512. Each block applies two convolutions followed by batch normalization and activation to help the model learn non-linear features effectively. Some blocks (conv 1-4) include max pooling to reduce spatial dimensions, and later blocks (conv 4-5) integrate a dropout of 0.3 to prevent the model from overfitting.

The Decoder (Expansive Path) works to reconstruct the feature map to its original image size using a series of upsampling blocks. Each block upsamples the feature map and merges it with corresponding features from the encoder path (skip connections) using two convolutions to preserve important information lost during downsampling. The number of filters decreases by half in each block, inversely following the Encoder's progression. This symmetrical structure helps the model in precisely localizing and classifying different regions in the image. This path ends with the output layer with a set of final convolutions that use 32 filters, followed by a specific convolutional layer (using a kernel size of 1) designed to classify each pixel into one of the potential 13 classes, thus producing a detailed segmentation map.

3.5 DeepLab V3

The DeepLabV3 architecture enhances the traditional convolutional network with an Atrous Spatial Pyramid Pooling (ASPP) module and a pre-trained ResNet-50 backbone. The model architecture is shown in Figure 3. This setup effectively handles the downsampling issue common in convolutional networks, which often results in loss of feature details. DeepLabV3 utilizes a ResNet50 backbone, modified for segmentation tasks by removing the top layer. It uses an Atrous Spatial Pyramid Pooling (ASPP) module with convolutions at various dilation rates to enhance feature extraction at multiple scales. This improves segmentation accuracy by integrating both contextual and detailed information. Low-level features are extracted, refined, and combined with deep features from the ASPP output. This concatenation is further processed through convolutional layers, then upsampled to match the input size. The final layer classifies each pixel into one of 13 classes using sigmoid activation. The model is trained to optimize segmentation performance, using a loss function tailored for multi-class tasks and accuracy as a metric. This design makes DeepLabV3 effective for precise image segmentation.

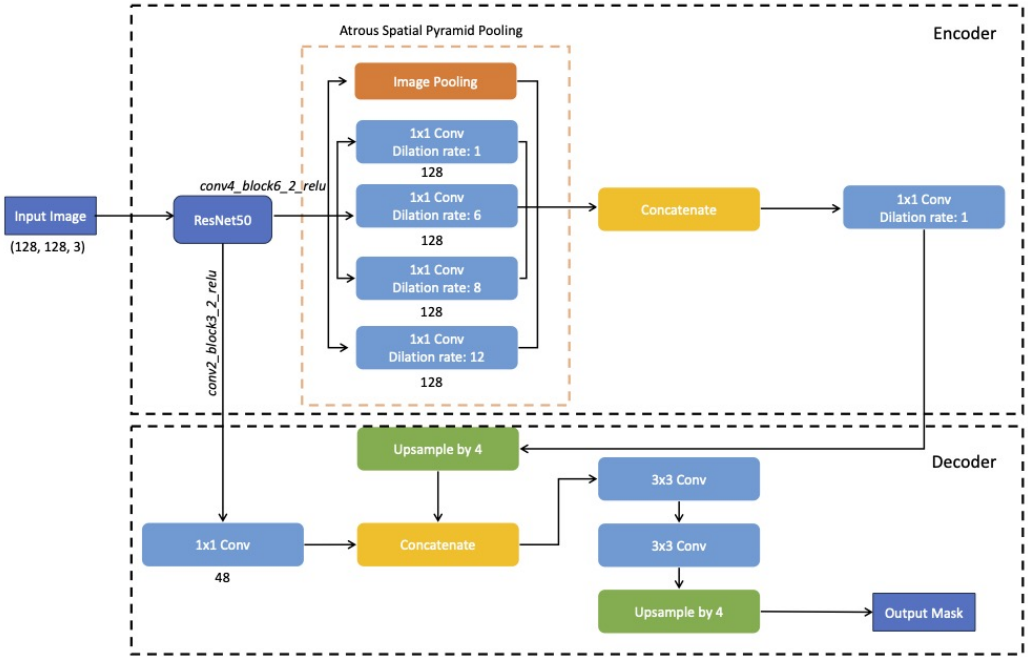

Figure 3: DeepLabV3 with ResNet50 Model Architecture

4 Evaluation and Results

For RQ1, FCN, U-Net, DeepLab V3 models were built from scratch, trained on 70% of the CARLA dataset, fine-toned and optimized the model using the validation dataset (15%), and the performance of three models were tested on the rest of the data. For RQ2, we tested the three models from a new dataset, to see their performance and flexibility on completely unknown senerios.

4.1 Evaluation Metric

The three models were evaluated on pixel accuracy and Mean IoU.

Pixel Accuracy

$$\text{Pixel Accuracy} = \frac{\text{Number of correctly predicted pixels}}{\text{Total number of pixels}}$$

Pixel Accuracy measures the proportion of pixels that are correctly labeled. The range of pixel accuracy is between 0 and 1. Higher accuracy represents better performance.

Mean Intersection Over Union (Mean IoU)

$$\text{Mean IoU} = \frac{1}{\text{Number of Classes}} \sum_{i=1}^{\text{Number of Classes}} \frac{\text{Area of Intersection}}{\text{Area of Union}}_i$$

The range of mean IoU is between 0 and 1. Here, higher IoU represents a better alignment between predicted and true mask. Usually IoU higher than 0.7 represents good performance.

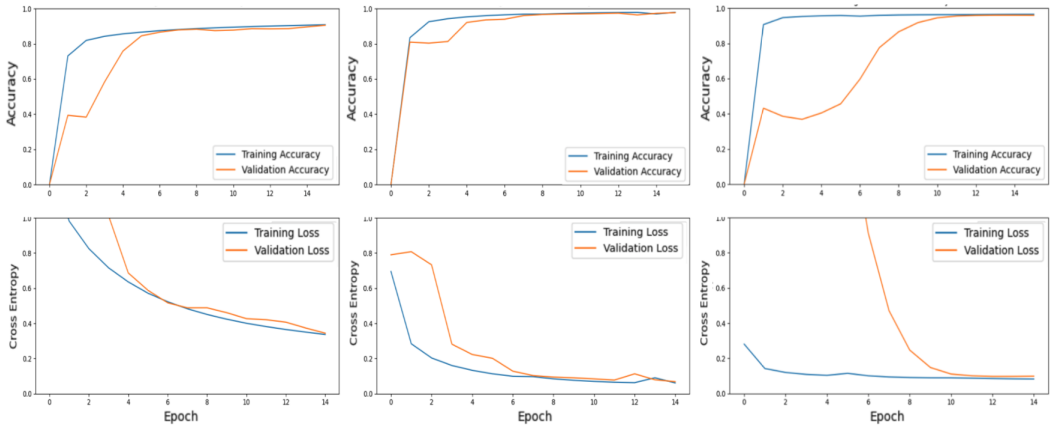

Figure 4: Training and validation Accuracy, Loss for FCN (left), U-Net (middle) and DeepLab V3 (right)

4.2 Results of RQ1: performance of three model architectures

Table 1 summarized the performance of three semantic segmentation models (FCN, U-Net, DeepLab V3) on the test dataset. Four metrics including accuracy, mean Intersection over Union (IoU), precision, and recall were used to compare the performance. Among the models, U-Net shows highest performance in all four metrics, with an accuracy of 0.98, mean IoU of 0.77, precision of 0.86, and recall of 0.81 followed by DeepLab V3 with 0.96 accuracy and 0.73 IoU. The baseline FCN model performed the worse within the three architectures with a 0.91 accuracy and 0.55 IoU.

Table 1: Performance Metrics of Three Models on Test Dataset

Model	Accuracy	Mean IoU	Precision	Recall
FCN	0.91	0.55	0.74	0.61
U-Net	**0.98**	**0.77**	**0.86**	**0.81**
DeepLab V3	0.96	0.73	0.84	**0.81**

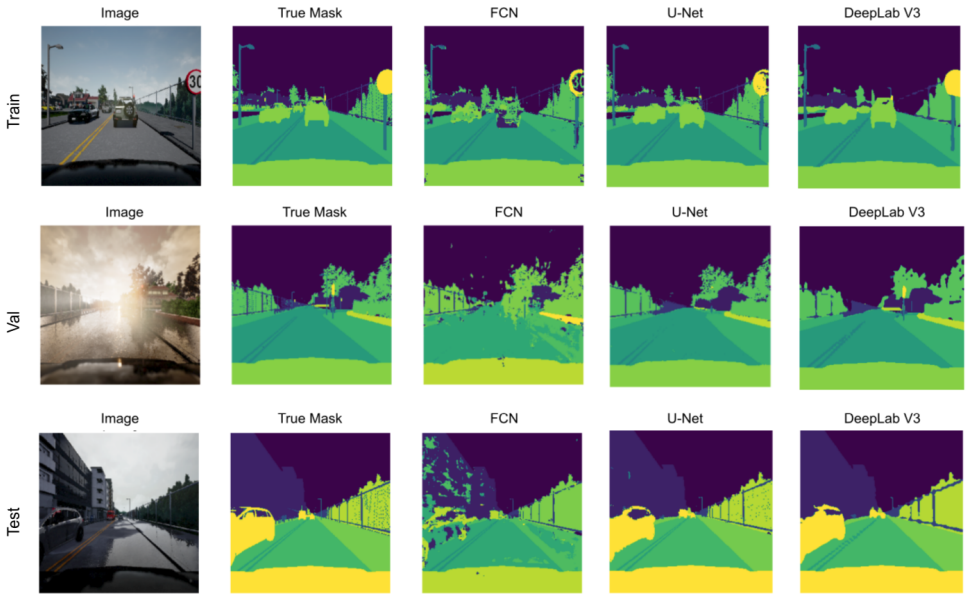

Figure 5: Semantic Segmentation of Three Models on Three Input Images

Figure 4 shows the training and validation accuracy (top line), and cross-entropy loss (bottom line) for each model during training of each epoch. Both FCN and U-Net converged at 6 epoch, while DeepLab model converged at 10 epoch. Early Stopping was used to prevent over-fitting on training data. Training will cease if there is no improvement in the validation accuracy for 5 consecutive epochs. In the figure, it is noticeable that training and validation accuracy are close, which indicates no over-fitting on training data.

To visualize the actual segmentation on different weather conditions, one image from train, validation and test dataset was used to compare the model performance. In Figure 5, the three images are in sunny, extreme bright, and rainy condition. We can notice that, the FCN model failed to identify the car and road sign in the first image, failed to label the solid line, the road and the tree in the second image and failed to identify the car in the third one. On the other hand, DeepLab V3 was able to successfully identify everything including the road signs, solid lines and the car. While U-Net did a better job on identifying clearer boundries between different classes.

4.3 Results of RQ2: performance of three models in unseen dataset

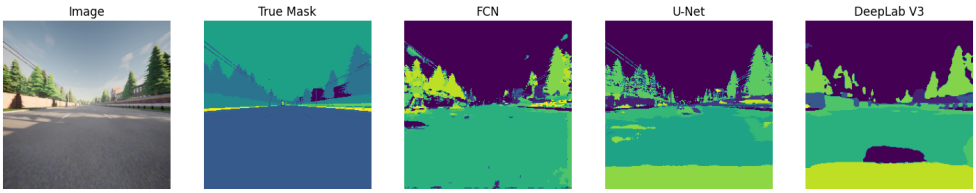

Figure 6: Predicted Masks on New Dataset

In this research question, we aim to investigate the performance of three semantic segmentation models under a dataset with different environmental conditions. These models were initially trained on a dataset with a major representation of complex weather scenarios, such as rain and overcast, which prepared them for adverse driving conditions. However, real-world applications may rely more on normal weather conditions, so we want to evaluate their effectiveness and performance when applied to a new dataset that mainly includes common urban settings under normal weather conditions with gentle afternoon sunlight.

Upon this new dataset, the FCN, U-NET and DeepLab models displayed varied adaptability. Figure 6 shows the predicted results from three models separately. The FCN model shows the best generalization, as it yielded the clearest predicted masks, although it demonstrated some challenges in delineating finer details. The U-Net model generally produced correct segmentation, but its performance on the new dataset revealed some limitations compared to its robust performance on the original dataset. Finally, the DeepLab model also demonstrated a similar ability to maintain segmentation accuracy as the U-Net and it also showed slightly weaker robustness in the new dataset. Comparatively, while all models showed an acceptable level of accuracy, their performance varied. The U-Net and DeepLab models displayed minor struggles, potentially due to the shift in image details and the absence of further training on the new dataset.

The ability of a model to generalize is essential in machine learning. In the context of self-driving cars, the segmentation generated by models should accurately distinguish road conditions and obstacles across diverse environments. Our investigation into the FCN, U-Net and DeepLab models reveals that all excel at segmenting complex scenes, but they exhibited mixed results when introduced to standard conditions, which indicates a need for improvement in their generalization. To enhance generalization, one possible approach is to expand the training datasets to include a broader range of scenarios, such as different times of day, weather conditions and urban settings. By using a more comprehensive training dataset that includes both typical and atypical conditions, we can further improve the models to ensure their application in the real world remains reliable and safe.

5 Computational Resources

The code was run on Kaggle because it supports notebook versioning and it can keep track of all trails of hyper-parameter tuning and error fixing. The Kaggle GPU has more RAM and disk space

Table 2: Summary of GPU experiment settings on Kaggle and Colab.

Property	Kaggle	Colab
Name	Tesla P100	Tesla T4
freq. (MHz)	1328.5	1590.0
memory (MB)	16276	15102
#multiproc.	56	40
RAM (GB)	31	12
disk space (GB)	32673	472

compare with Colab T4 GPU, which is more suitable in our experiment. Once committed the code, Kaggle can run the code in the background until the code finished. The training time of the FCN, U-Net, DeepLab V3 model is 26, 23, 28 minutes respectively for 15 epochs.

6 Conclusion

Throughout the project, we evaluated the performance and adaptability of FCN, U-Net, and DeepLab V3 models in the context of semantic segmentation for autonomous driving. Each model has demonstrated their advantages and certain limitations, highlighting areas for future enhancement.

Although the FCN model has the simplest architecture, it provides a solid baseline for semantic segmentation tasks. It shows good performance in most basic scenarios but struggles with more complex and diverse environments, particularly where fine detail is essential. This suggests a need for further refinement, possibly by exploring deeper network architectures or integrating more advanced regularization techniques to enhance its precision and generalization across varied environments.

We found U-Net, with its symmetric design and effective use of skip connections, excelled in localizing and detailing complex structures within the images. It achieves the highest performance metrics among all models. Its structure effectively captures and reconstructs the spatial hierarchies necessary for precise segmentation. However, the U-Net model exhibits worse performance when applied to unfamiliar datasets. We think Its generalization is limited by the diversity of training dataset, and it could be improved through using extended training data with a broader range of scenarios.

DeepLab utilizes the Atrous Spatial Pyramid Pooling to capture contextual information, which shows strong capability in segmenting fine details and dealing with varied scales of objects. Like U-Net, DeepLab faced challenges in maintaining consistent performance across new and varied datasets. Enhancing its generalization could involve employing regularization methods like Dropoutor or more extensive pre-training on diverse datasets to improve its robustness.

Through the implementation and refinement of these three deep learning models for semantic segmentation, we learned the critical role of model architecture in effectively capturing and classifying detailed image features for autonomous driving. Moreover, this process highlighted the importance of optimizing the training and adaptation strategies to enhance model performance in various scenarios. Consequently, we recognize that while these models are highly effective, they require ongoing adjustments and enhancements to improve their generalization abilities across unseen scenarios, underscoring the need for robust training and evaluation methods.

This project has several limitations that, if addressed, could significantly improve future studies and practical applications of semantic segmentation models in autonomous driving. The main constraints we faced included limited time and computational resources, which restricted the scope of our experiments and the extent of model tuning we could conduct. For future studies, allocating more time and resources towards refining the models, training them with larger and more diverse datasets, exploring additional regularization techniques, and implementing cross-validation methods to assess the models across different datasets could provide greater insights and help develop more sophisticated and reliable systems for autonomous vehicles.

7 Work Allocation

The four of us all took part in coding and writing the paper.

Shi Tang implemented FCN model, Dechen Han implemented U-Net model, Xinpeng Shan implemented Deeplab V3 model and did hyperparameter tonings, Wendy Yusi Cheng implemented RQ2 code. We did data preprocessing together in a meeting.

For the paper, Shi Tang wrote introduction and Background, Wendy Yusi Cheng wrote Model Architecture and Conclusion, Dechen Han and Xinpeng Shan wrote results.

References

[1] B. Li, S. Liu, W. Xu, and W. Qiu, "Real-time object detection and semantic segmentation for autonomous driving," in *MIPPR 2017: Automatic Target Recognition and Navigation*, vol. 10608. SPIE, 2018, pp. 167–174.

[2] X. Zhu, H.-I. Suk, S.-W. Lee, and D. Shen, "Subspace regularized sparse multitask learning for multiclass neurodegenerative disease identification," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 3, pp. 607–618, 2015.

[3] Y. Guo, Y. Gao, and D. Shen, "Deformable mr prostate segmentation via deep feature learning and sparse patch matching," *IEEE transactions on medical imaging*, vol. 35, no. 4, pp. 1077–1089, 2015.

[4] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," 2017.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[7] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," 2016.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," 2018.

[9] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017.

[10] K. Manickavelu, "Lyft-udacity challenge dataset," https://www.kaggle.com/datasets/kumaresanmanickavelu/lyft-udacity-challenge, 2021.

[11] S. Aggarwal, "Semantic segmentation car driving dataset," https://www.kaggle.com/datasets/shivamaggarwal513/semantic-segmentation-car-driving, 2021.

[12] H. Iqbal, "Harisiqbal88/plotneuralnet v1.0.0." Zenodo, 2018.