Deep Learning for Semantic Segmentation of CARLA Simulator Data

ECE 285 course project

Dhruv Talwar
Electrical and Computer Engineering
A59015745
Contribution: DeepLabv3+(Resnet101 & Resnet50)

Surya Lakshmi Subba Rao Pilla
Electrical and Computer Engineering
A59015821
Contribution: U-Net (Vanilla & Attention)

***Abstract*—The development of autonomous driving has gained significant attention in recent years, and it requires the ability to accurately perceive and interpret the surrounding environment. Semantic segmentation and depth estimation are crucial components in autonomous driving, enabling the vehicle to understand the structure and layout of the scene and make informed decisions. In this project, we propose to investigate deep learning techniques for semantic segmentation using data from the CARLA simulator data for self-driving cars. We will explore state-of-the-art deep learning models utilizing encoder-decoder architectures, such as U-NET and DeeplabV3 to develop robust and accurate models. We will also implement and compare different flavors of these models and evaluate the performance of the proposed models using standard metrics, such as dice score and intersection over union (IoU). The results of this project will provide insights into the effectiveness of deep learning techniques for semantic segmentation in CARLA simulator data for self-driving cars. Ultimately, this project aims to contribute to the development of safer and more reliable autonomous driving systems.**

I. Introduction

The rise of autonomous driving in recent years has placed significant emphasis on the need for accurate perception and interpretation of the surrounding environment. To achieve this, there is a growing demand for techniques that are less reliant on hardware and can maintain a high frames-per-second (FPS) rate for real-time processing. Semantic segmentation is critical components in the development of autonomous driving systems, as they allow vehicles to accurately comprehend the scene's layout and structure, enabling informed decision-making.

A. Objective

The main aim of the project is to analyze and compare the segmentation outcomes achieved using both the U-Net and DeepLabV3 models. Additionally, modifications have been made to these conventional networks, and a comparative analysis has been conducted.

II. Dataset

The CARLA Semantic Segmentation Dataset is a valuable resource for developing and evaluating deep learning algorithms in the field of autonomous driving and computer vision. This dataset is specifically designed for semantic segmentation tasks, which involve assigning pixel-level labels to different objects and regions in images.

The dataset is generated using the CARLA simulator, a popular open-source platform for autonomous driving research. CARLA provides a realistic virtual environment that simulates various urban driving scenarios with diverse lighting conditions, weather conditions, and traffic situations. This makes it an excellent tool for training and testing semantic segmentation models.

The CARLA Semantic Segmentation Dataset[1] consists of 5000 of 800x600 resolution images, each accompanied by corresponding pixel-level labels. The labels provide detailed information about different objects and classes present in the scene, such as Vehicles (e.g., cars, trucks, motorcycles), Pedestrians, Cyclists, Traffic signs and signals, Buildings and structures Vegetation and trees, Roads and lanes Sidewalks and curbs, Sky and background This level of annotation enables researchers to develop robust algorithms for scene understanding and perception in autonomous driving systems. We used 4000 images for training and kept 1000 for testing purposes.

A custom dataloader was built for the CARLA dataset. It played a vital role in this project by efficiently handling the CARLA dataset, which comprised RGB images and segmented masks for 23 different classes. The custom dataloader, performed several key functions. Firstly, it loaded the image and mask filenames from the specified directories, making them accessible for further processing. Secondly, it provided a way to retrieve the length of the dataset and individual samples. Each sample was loaded using OpenCV and stored as a dictionary containing the image and its corresponding mask. Additionally, the dataloader applied optional transformations to the images and masks, such as resizing and conversion to tensors.

By utilizing this custom dataloader, the project efficiently handles the CARLA dataset. It enables the application of transformations to the data and facilitates the training and evaluation of deep learning models, such as UNet, UNet with attention, Deeplabv3 with ResNet101, and Deeplabv3 with ResNet50 backbones. The dataloader ensures that the models receive properly preprocessed batches of data, which in turn contributes to the development of accurate and robust segmentation models for the 23 classes in the CARLA dataset.

III. Architecture

A. U-Net

U-Net architecture [2][3] utilizes encoder decoder to segment the image. The architecture is given in Figure 1 and explained below:

Encoder:

The input image is passed through a series of convolutional layers, each followed by a rectified linear unit (ReLU) activation function and max-pooling operation. This downsampling process progressively reduces the spatial dimensions of the input while increasing the number of feature channels. As the encoder path proceeds, the network learns to capture and encode both low-level and high-level features.

Bridge:

At the bottom of the U-Net, there is a bridge that connects the encoder and decoder paths. It consists of additional convolutional layers that help in the transition from the encoding to the decoding stage.

Decoder Path:

The decoder path performs upsampling operations to gradually reconstruct the spatial information lost during downsampling. Each upsampling step is performed by a combination of upsampling (e.g., bilinear interpolation) and convolutional layers. Skip connections are introduced between corresponding encoder and decoder layers to enable the flow of high-resolution features. The skip connections concatenate the feature maps from the encoder path with the upsampled feature maps in the decoder path. By incorporating skip connections, the U-Net architecture allows for fine-grained localization and better preservation of spatial details.

Output:

The final layer of the decoder path typically consists of a 1x1 convolutional layer followed by a suitable activation function, such as the sigmoid function. This produces an output map with the same spatial dimensions as the input image, where each pixel represents the probability of belonging to a particular class or category. During training, the network is optimized using a suitable loss function, such as the Dice loss or binary cross-entropy, to match the predicted segmentation mask with the ground truth. The U-Net architecture's distinctive feature is the combination of the contracting path (encoder) and the expansive path (decoder) with skip connections, enabling it to leverage both local and global context while preserving fine-grained details. This makes it particularly effective for tasks like image segmentation, where precise localization and segmentation accuracy are crucial.

Since its introduction, the U-Net architecture has inspired numerous variants and adaptations to cater to specific domain requirements and challenges, further enhancing its versatility and applicability in a wide range of image analysis tasks.

Figure 1 gives a representation of the U-Net model.

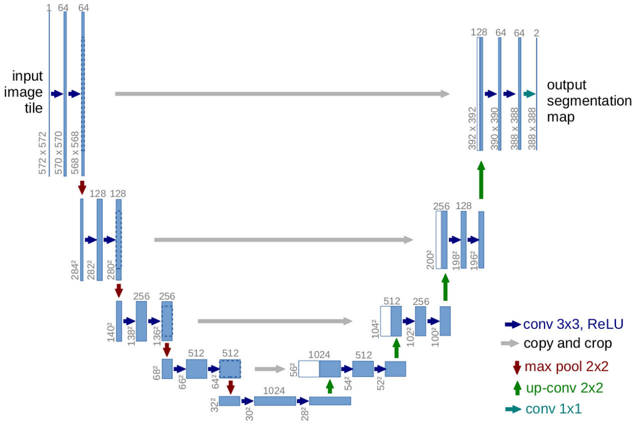

Fig. 1. U-Net Architecture

In addition to its original design, the U-Net architecture has been further improved with the integration of attention mechanisms. Attention U-Net [4] extends the standard U-Net by incorporating attention gates, which dynamically weigh the importance of different spatial locations during the information flow. Attention gates selectively amplify or suppress features based on their relevance to the final segmentation. This mechanism allows the network to focus on the most informative regions, enhancing its ability to capture fine details and intricate structures while reducing the influence of irrelevant or noisy information. By adaptively attending to relevant image regions, the attention U-Net achieves more precise and accurate segmentation results, especially in challenging scenarios with complex backgrounds or class imbalance. The attention mechanism in the U-Net architecture contributes to improved localization and segmentation performance, making it a powerful tool for various image analysis tasks.

B. DeepLabv3+

DeepLabv3+ [5][6] is a network architecture designed for semantic segmentation , which combines an encoder-decoder structure with Atrous Convolution and Atrous Spatial Pyramid Pooling (ASPP) techniques.

Atrous convolution effectively controls the receptive field by introducing a rate parameter. It convolves the input feature map (x) with a filter (w), and each location (i) in the output (y) is computed using a generalized form of atrous convolution described by the following equation:

$$y[i] = \sum_k x[i + r \cdot k]w[k]$$

Unlike regular convolutions, atrous convolution enables information retrieval at multiple scales by employing multiple atrous convolution layers on the image.

ASPP in DeepLabv3+[7] captures multi-scale information effectively. It includes one 1×1 convolution and three 3×3 convolutions with dilation rates of 6, 12, and 18 respectively. ASPP also incorporates image-level features through image pooling. The resulting features from these branches are concatenated and further processed by a 1×1 convolution.

DeepLabv3 serves as an encoder, extracting valuable features at arbitrary resolutions. ASPP enables the exploration of convolutional features at different scales, thanks to the diverse dilation rates. As a result, the output feature map of the encoder networks, which typically has 256 channels and is 32 times smaller than the input image resolution, contains rich semantic information.

The rich encoded features from the encoder networks undergo upsampling by a factor of 4 using bilinear interpolation. They are then combined with lower-level features obtained from a backbone network of the same shape. To prevent the lower-level features from overpowering the encoded features, a 1×1 convolution is applied to reduce their channel dimensions. After concatenation, the combined features undergo a series of 3×3 convolutions and are finally upsampled by a factor of 4 again using bilinear interpolation.

Figure 2 gives a representation of the Deeplabv3 model.

1) Model Backbone: In our project, we employed ResNet-50 and ResNet-101, two state-of-the-art Deep Convolutional Neural Network (DCNN) based backbone network architectures. These backbone networks extract high-level features from images and perform downsampling. They consist of various components such as convolution, pooling, and activation functions, enabling effective feature extraction.

The Residual Network (ResNet) has proven to be effective in training deeper networks by introducing identity shortcut connections that alleviate the vanishing gradient problem. With ResNet, it is possible to create deep network versions, such as ResNet-50 and ResNet-101, which have 50 and 101 layers respectively, allowing for the extraction of advanced features.

In this project, we utilized pre-trained models, specifically ResNet-50 and ResNet-101, with 'Imagenet' weights. Initially, a deep neural network-based model was pre-trained to extract generalized features from different layers. These features were then utilized in the encoder and decoder of the DeepLab architecture, based on their depth, to enhance the method's performance. Finally, our proposed model underwent fine-tuning on the segmentation dataset using augmentation techniques to mitigate overfitting.

In our implementation, to adapt the model for the specific task, a DeepLabHead with an input size of 2048 and the desired output channel size was added as the classifier. Notably, a Tanh activation function was introduced after the last convolution layer. This modified architecture allows the model to generate predictions with 3 output channels.

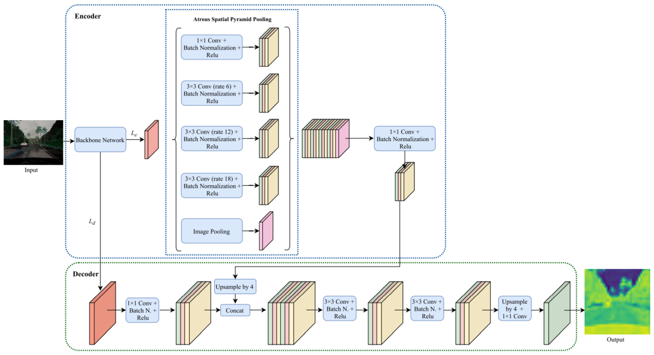
Encoder
Atrous Spatial Pyramid Pooling
1×1 Conv + Batch Normalization + Relu
3×3 Conv (rate 6) + Batch Normalization + Relu
3×3 Conv (rate 12) + Batch Normalization + Relu
3×3 Conv (rate 18) + Batch Normalization + Relu
Image Pooling
Input
Backbone Network
L_e
L_d
1×1 Conv + Batch Normalization + Relu
Decoder
1×1 Conv + Batch N. + Relu
Upsample by 4
Concat
3×3 Conv + Batch N. + Relu
3×3 Conv + Batch N. + Relu
Upsample by 4 + 1×1 Conv
Output

Fig. 2. DeeplabV3+ Architecture

IV. Results

In this section, we present the evaluation metrics used to assess the performance of our segmentation model: accuracy, Intersection over Union (IOU), and Dice score. These metrics provide valuable insights into the quality and effectiveness of the model's predictions.

A. Evaluation Metrics

We use the following metrics to evaluate the performance of our segmentation model:

- **Accuracy**: Accuracy measures the percentage of correctly classified pixels in the predicted masks.
- **Intersection over Union (IOU)**: IOU, also known as the Jaccard index, quantifies the overlap between the predicted mask and the ground truth mask.
- **Dice Score**: The Dice score measures the similarity between two binary masks.

These metrics allow us to assess different aspects of the model's performance and provide a comprehensive evaluation of its segmentation capabilities.

1) Accuracy: Accuracy measures the percentage of correctly classified pixels in the predicted masks. It is calculated by dividing the number of correctly classified pixels by the total number of pixels in the image.

$$\text{Accuracy} = \frac{\text{Number of correctly classified pixels}}{\text{Total number of pixels}} \tag{1}$$

2) Intersection over Union (IOU): IOU, also known as the Jaccard index, quantifies the overlap between the predicted mask and the ground truth mask. It is calculated as the ratio of the intersection to the union of the two masks.

$$\text{IOU} = \frac{\text{Intersection}}{\text{Union}} \tag{2}$$

where the intersection represents the number of pixels that are correctly classified as foreground in both masks, and the union represents the total number of pixels that are classified as foreground in either mask.

3) Dice Score: The Dice score measures the similarity between two binary masks. It is calculated as twice the intersection divided by the sum of pixels in both masks.

$$\text{Dice score} = \frac{2 \times \text{Intersection}}{\text{Sum of pixels in both masks}} \tag{3}$$

The Dice score ranges from 0 to 1, where 1 indicates a perfect match between the predicted and ground truth masks.

B. UNET and Attention

In this section, we present the results obtained from our segmentation models based on the U-Net architecture with and without using Attention layer. We provide a comparison of the models' performance, discuss the hyperparameters used, and present qualitative and quantitative evaluations.

1) Hyperparameters: Table III summarizes the hyperparameters used for training the U-Net model.

TABLE I
HYPERPARAMETERS

Hyperparameter	Value
Learning Rate	0.00001
Batch Size	32
Number of Epochs	10
Optimizer	Adam

The models were trained using the Adam optimizer with a learning rate of 0.00001. The batch size was set to 32, and the models were trained for 10 epochs.

2) Training and Evaluation: The U-Net model with and without attentions were trained on a dataset of 5000 images with corresponding ground truth masks. The dataset was split into a training set and a separate test set. During training, data augmentation techniques such as random flipping and rotation were applied to increase the variability of the training samples.

The models were trained using the cross-entropy loss function and evaluated using three evaluation metrics: accuracy, Intersection over Union (IOU), and Dice score. The accuracy measures the percentage of correctly classified pixels, while the IOU and Dice score quantify the overlap between the predicted and ground truth masks.

3) Results and Comparison: Table II provides a comparison of the performance of the U-Net with and without Attention. The accuracy model with attention model is high beacuse it introduces attention mechanisms that enable the model to selectively focus on informative regions while suppressing irrelevant or noisy information. This attention mechanism allows the network to better understand the contextual relationships between different pixels and capture more detailed information. In contrast, the original UNet architecture lacks this explicit attention mechanism, resulting in a limited ability to effectively capture complex contextual information.

TABLE II
MODEL COMPARISON

Model	Accuracy %	IOU	Dice Score
U-Net	75.73	0.7	0.75
U-Net + Attention	91.2	0.8	0.87

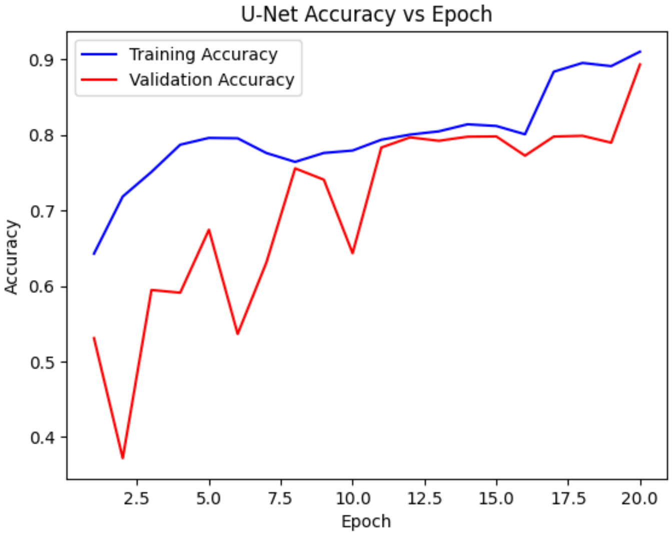

Fig. 3. UNet + Attention

C. Deeplabv3+

In this section, we present the results obtained from our segmentation models based on the Deeplabv3 architecture using ResNet101 and ResNet50 as backbone networks. We provide a comparison of the models' performance, discuss the hyperparameters used, and present qualitative and quantitative evaluations.

1) Hyperparameters: Table III summarizes the hyperparameters used for training the Deeplabv3 models.

TABLE III
HYPERPARAMETERS

Hyperparameter	Value
Backbone Network	ResNet101 / ResNet50
Learning Rate	0.00001
Batch Size	16
Number of Epochs	20
Optimizer	Adam

The models were trained using the Adam optimizer with a learning rate of 0.00001 and weight decay of 0.0001. The batch size was set to 16, and the models were trained for 20 epochs.

2) Training and Evaluation: The Deeplabv3 models with ResNet101 and ResNet50 backbones were trained on a dataset of 5000 images with corresponding ground truth masks. The dataset was split into a training set and a separate test set. During training, data augmentation techniques such as random flipping and rotation were applied to increase the variability of the training samples. Fig 3 shows the plot of

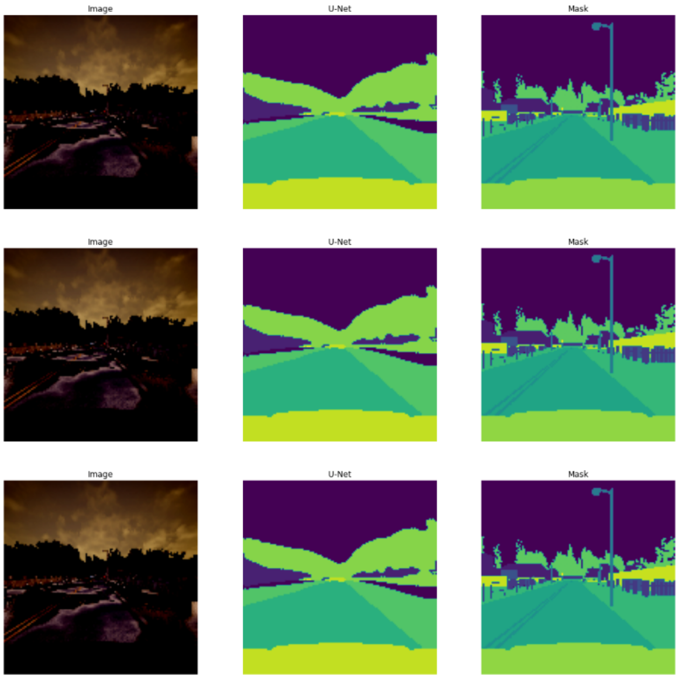

Fig. 4. U-Net

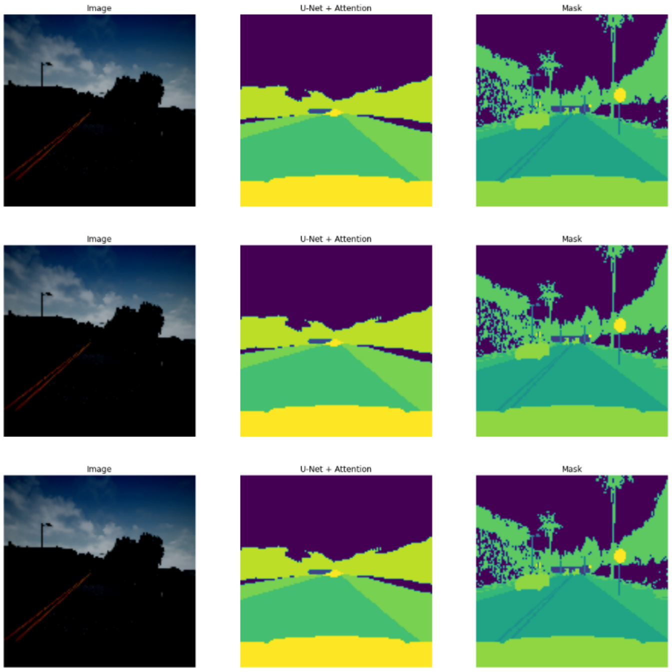

Fig. 5. U-Net + Attention

validation and training accuracy Vs Epoch plot. Fig 4 and 5 are the comparison of unet and unet+attention results.

The models were trained using the cross-entropy loss function and evaluated using three evaluation metrics: accuracy, Intersection over Union (IOU), and Dice score. The accuracy measures the percentage of correctly classified pixels, while the IOU and Dice score quantify the overlap between the predicted and ground truth masks.

3) Results and Comparison: We can observe the training versus validation accuracy graphs with respect to the number of epochs for the ResNet101 model.

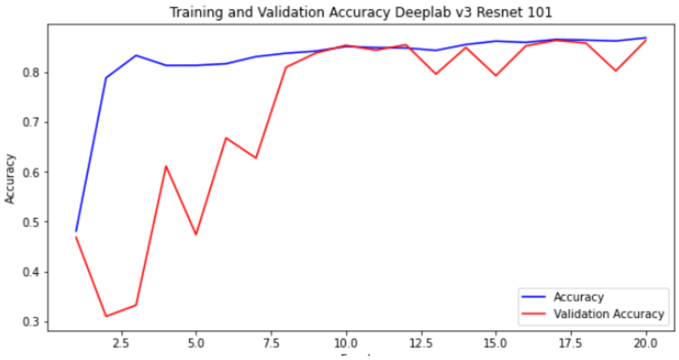

Fig. 6. DeeplabV3+ with ResNet101

In the training versus validation loss graphs, it is observed that the training loss decreases steadily over time, indicating that the model is effectively learning from the training data. However, the validation loss shows intermittent jumps to higher levels before gradually decreasing. These sudden jumps in validation loss may indicate instances of overfitting, where the model becomes too specialized to the training data and performs poorly on unseen data.

By By initializing my model with pre-trained weights from ResNet-101, we can give a good starting point for your training process. The pre-trained weights already capture low-level features such as edges, textures, and basic shapes, allowing the model to focus on learning higher-level features.

Table IV provides a comparison of the performance of the Deeplabv3 models based on ResNet101 and ResNet50.

TABLE IV
MODEL COMPARISON

Model	Accuracy in %	IOU	Dice Score
ResNet101	87.23	0.75	0.85
ResNet50	78.77	0.64	0.78

From the results, we observe that both models achieved high accuracy, with the ResNet101 model achieving an accuracy of 0.87 and the ResNet50 model achieving an accuracy of 0.78. However, the ResNet101 model outperformed the ResNet50 model in terms of IOU and Dice score, indicating a better overlap and similarity between the predicted and ground truth masks.

Qualitative evaluation, Fig 7 and 8 of the models' predictions showed that the ResNet101 model produced smoother and more accurate segmentation masks compared to the ResNet50 model, which exhibited some pixelation and inconsistencies in certain regions.
This the higher performance of the ResNet101 backbone can be attributed to its deeper architecture and increased capacity to capture more complex features. The model with ResNet101 demonstrated improved accuracy, IOU, and Dice scores compared to the model with ResNet50, indicating its ability to better capture the segmentation boundaries and produce more accurate masks.

Our results demonstrate promising performance, as evidenced by high scores in metrics such as accuracy, IOU, and Dice coefficient. However, upon closer inspection of the generated masks, we observed a potential limitation in the qualitative aspect. Specifically, when encountering complex scenes with cars on the road, the predicted masks exhibited a degree of blurriness and failed to define the edges sharply. This issue became particularly noticeable in regions where fine details and intricate structures were present. It is important to note that our experiments were conducted for a relatively short training duration of 20 epochs, which could have limited the model's ability to capture subtle nuances and intricate details. Addressing this challenge may require further experimentation with prolonged training, the exploration of additional data augmentation techniques, or the adoption of more advanced architectural modifications.

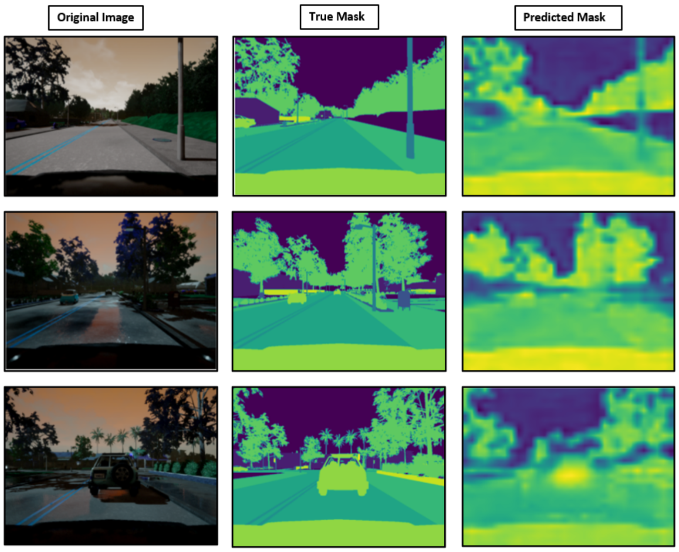

Fig. 7. DeeplabV3+ with ResNet101

D. Inter Model Comparison

The comparison between U-Net and DeeplabV3 provides valuable insights into their respective performance in image segmentation tasks. U-Net with attention, achieving an accuracy of 91.2, IOU (Intersection over Union) of 0.8, and Dice score of 0.82, demonstrates good overall performance.

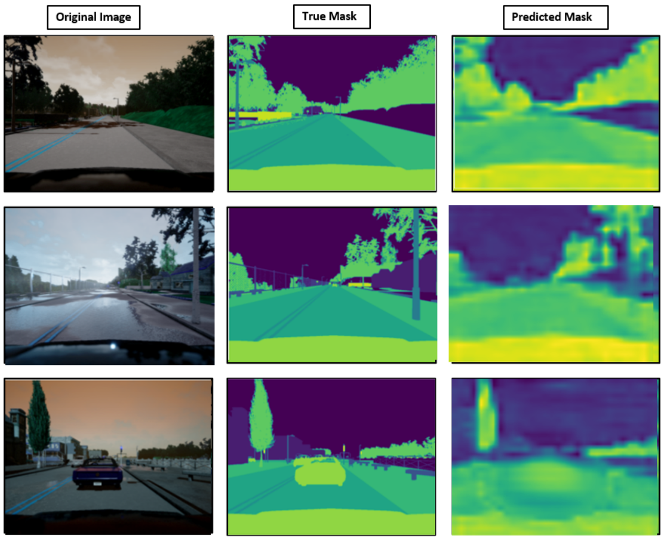

Fig. 8. DeeplabV3+ with ResNet50

The U-Net architecture incorporates skip connections, enabling the fusion of high-resolution features from the encoder with upsampled features from the decoder. This design allows U-Net to capture fine details and produce masks with sharp boundaries.

However, U-Net has its limitations, particularly in detecting small objects and distant details. In our experiments, we observed that U-Net struggled to identify cars and small trees in the predicted masks. The downsampling process in U-Net reduces the spatial dimensions of the input image, potentially leading to the loss of fine-grained details. Although the subsequent upsampling in the decoder path attempts to reconstruct the lost spatial information, it may still face challenges with small objects occupying only a few pixels or distant details that are not well-preserved during downsampling. Furthermore, U-Net's receptive field, representing the effective area that influences each pixel's prediction, may not be large enough to capture small objects or distant details accurately.

On the other hand, DeeplabV3 takes a different approach to image segmentation. With an accuracy of 87.2, IOU of 0.75, and Dice score of 0.85, DeeplabV3 demonstrates competitive performance. It utilizes atrous convolutions and a spatial pyramid pooling module to capture contextual information at multiple scales.
The predicted masks from DeeplabV3 may tend to be blurry, but the model exhibits the ability to detect cars and small, faraway trees to some extent. Despite the lack of sharp boundaries, DeeplabV3 effectively captures important semantic features in the image. By leveraging a broader contextual understanding of the scene, DeeplabV3 excels at detecting objects of interest, even if the boundaries appear less precise.

In summary, U-Net excels in producing masks with precise boundaries and preserving fine details, making it well-suited for applications where accurate localization is crucial, such as medical image segmentation. However, U-Net may face challenges in detecting small objects and distant details. On the other hand, DeeplabV3 sacrifices boundary sharpness but offers a broader contextual understanding of the scene. It can effectively capture important semantic features and detect objects of interest, even if the boundaries appear blurry.

It is important to note that further experimentation, such as hyperparameter tuning and longer training durations, could potentially improve the performance of both models and address some of their limitations. Exploring advanced variants of U-Net, such as U-Net with attention or other state-of-the-art architectures, may also provide promising avenues for enhancing performance in detecting small objects and distant details.

V. Conclusion

In this project, we investigated deep learning techniques for semantic segmentation in the context of autonomous driving using the CARLA simulator data. We explored two state-of-the-art models, U-Net and DeepLabv3+, and evaluated their performance using standard metrics such as accuracy, Intersection over Union (IOU), and Dice score.

The U-Net architecture, with its encoder-decoder structure and skip connections, proved to be effective in capturing both low-level and high-level features for accurate segmentation. We trained the U-Net model on the CARLA dataset and achieved promising results in terms of accuracy and IOU. The model showed potential for scene understanding and perception in autonomous driving systems.

DeepLabv3+ offered an alternative approach with its encoder-decoder structure, Atrous Convolution, and Atrous Spatial Pyramid Pooling (ASPP) techniques. By incorporating multi-scale information and image-level features, DeepLabv3+ demonstrated robust performance in semantic segmentation tasks. We employed two variants of DeepLabv3+, utilizing ResNet-50 and ResNet-101 as backbone networks. Both models achieved competitive results, with ResNet-101 outperforming ResNet-50. Ultimately U-net with attention outperforming all in terms of accuracy, IOU score and Dice score.

VI. Further work

Future work can be conducted to further improve the performance of the models and explore additional enhancements. Some potential avenues for future research include:

Data augmentation techniques: Investigate the effectiveness of additional data augmentation techniques, such as random scaling, cropping, and color transformations, to increase the model's ability to generalize to different scenarios and lighting conditions.

Model ensembling: Explore the use of model ensembling techniques, such as averaging predictions from multiple models or using a weighted combination of their outputs, to improve the overall segmentation performance.

Attention mechanisms: Investigate the integration of attention mechanisms into the U-Net architecture to enhance the model's focus on informative regions and improve segmentation accuracy.

Real-world evaluation: Extend the evaluation of the models to real-world datasets and scenarios, considering factors such as varying weather conditions, diverse road environments, and complex traffic situations. This would provide a more comprehensive assessment of the models' robustness and generalization capabilities.

Depth estimation: Further explore depth estimation techniques in conjunction with semantic segmentation to enable more comprehensive scene understanding. Investigate the fusion of depth information with RGB images to enhance perception capabilities for autonomous driving.

By addressing these areas of future work, we can advance the development of accurate and reliable semantic segmentation and depth estimation models for autonomous driving systems, contributing to the realization of safer and more efficient self-driving vehicles.

References

[1] Carla, "CARLA Simulator," Website, 2020. Available at: http://carla.org/

[2] Subhedar, J., Bachute, M., Koundal, D., and Kotecha, K., 2023. Semantic Segmentation Algorithm for Autonomous Driving using UNET Architectures: A Comparative Study.

[3] Ronneberger, O., Fischer, P. and Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III (pp. 234-241). Springer International Publishing.

[4] Oktay, O., Schlemper, J., Folgoc, L.L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N.Y., Kainz, B., and Glocker, B., 2018. Attention U-Net: Learning where to look for the pancreas. arXiv preprint arXiv:1804.03999.

[5] Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A.L., 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4), pp. 834-848.

[6] Das, S., Fime, A.A., Siddique, N., and Hashem, M.M.A., 2021. Estimation of road boundary for intelligent vehicles based on deepLabV3+ architecture. IEEE Access, 9, pp. 121060-121075.

[7] Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F. and Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 801-818).

Github link of project: git@github.com:suryapilla/ECE285_project.git