

# Lane Detection System For ADAS

Krishna Kanth Vuppala Narasimha, Beenaa Motiram Salian, Srimoghavarsha Beerangi Srinivasa

Fall 2023 ECE 4554/5554 Computer Vision: Course Project

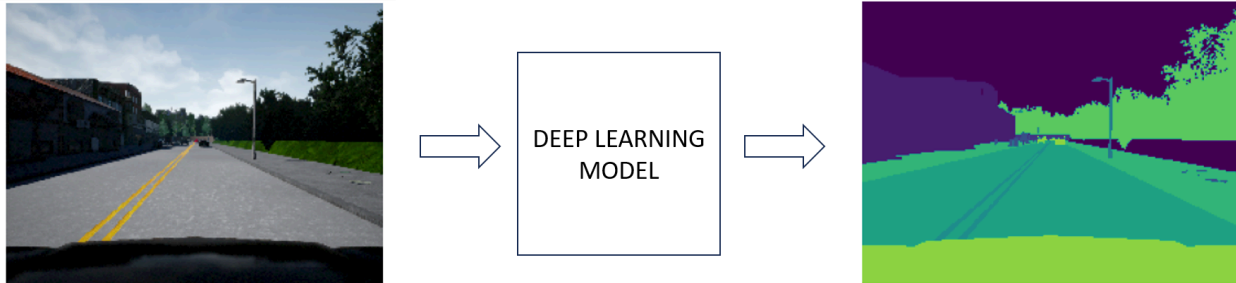
Virginia Tech

---

## Abstract

With the increasing demand for safer and more efficient autonomous driving, our project tackles the crucial task of enhancing lane detection systems. Accurate lane detection is pivotal for safe navigation and has far-reaching implications for the future of self-driving vehicles. We adopted a comprehensive approach by implementing and comparing two advanced deep learning models, U-Net (LaneNet) and SegNet, both trained on a diverse dataset of synthetically generated highway images. Our evaluation, conducted on a video generated by CARLA Simulator, provides insights into the models' performance in real-world driving scenarios.

## Teaser figure



# Introduction

In the rapidly evolving landscape of autonomous driving and advanced driving assistance systems, the accuracy of lane detection is crucial. Ensuring reliable lane identification is vital for vehicle safety, supporting essential functionalities like lane-keeping and path planning, particularly in complex urban environments. Traditional computer vision methods, while foundational, often struggle with real-world variables such as inconsistent lane markings, variable lighting, and adverse weather conditions.

This project explores the capabilities of deep learning in lane detection by harnessing U-Net and SegNet, two distinct models renowned for their segmentation prowess. U-Net, known for its precise segmentation abilities, is juxtaposed against SegNet, which excels in efficient semantic segmentation derived from the VGG16 architecture. Utilizing a synthetic dataset generated by the CARLA simulator, an open-source platform for autonomous driving research, this study trains and evaluates these models on a specially curated evaluation set. This comparative analysis aims to discern their performance and generalization capabilities in simulated real-world scenarios, contributing to the advancement of lane detection technologies in the realm of autonomous vehicles.

## Approach

- **Dataset Generation using CARLA**

The first critical step in our approach involved the generation of a comprehensive dataset using the CARLA simulator. To create a dataset that is both extensive and diverse, we focused on seven distinct towns available within CARLA, each offering unique urban layouts and environmental conditions. From each town, we generated a total of 1500 images, accompanied by their respective segmentation masks, amounting to a primary dataset of 10500 images.

This dataset was designed to capture a wide spectrum of driving scenarios, including variations in weather, lighting, and traffic conditions. The segmentation masks were meticulously crafted to accurately represent lane markings, ensuring high-quality ground truth data for model training.

For the evaluation phase, we extended our dataset generation to include a eighth town, not previously seen by the models during training. This new town contributed an additional 1500 images with corresponding masks, specifically reserved for testing the models' performance and generalization capabilities.

- **Model Selection**

For this project, we focused on two advanced deep learning models, each with distinct architectural strengths suited for the task of lane detection:

1. **U-Net (LaneNet)**

In our quest to advance lane detection technology, we chose to delve into the capabilities of LaneNet, particularly focusing on the U-Net architecture, which forms a significant part of LaneNet's design. LaneNet is known for its dual approach to lane detection, combining both semantic segmentation and instance segmentation. However, for our project's specific focus on semantic segmentation - the task of classifying each pixel into a lane or non-lane category - we selected U-Net, a core component of LaneNet's architecture responsible for its binary segmentation capability.

The architecture of U-Net consists of the following features:

- **Symmetrical Design:** U-Net's architecture is characterized by its symmetrical design, consisting of two main parts: a contracting (downsampling) path and an expansive (upsampling) path, connected by a bottleneck.
- **Contracting Path:** The contracting path follows the typical architecture of a convolutional network. It comprises repeated application of two 3x3 convolutions (each followed by a rectified linear unit or ReLU), and a 2x2 max pooling operation with stride 2 for downsampling. At each downsampling step, U-Net doubles the number of feature channels.
- **Bottleneck:** This is the transition layer between the contracting and expansive paths. It usually involves two 3x3 convolutions with ReLU and sometimes includes dropout for regularization.
- **Expansive Path:** Each step in the expansive path consists of an up-sampling of the feature map followed by a 2x2 convolution that halves the number of feature channels, a concatenation with the corresponding cropped feature map from the contracting path, and two 3x3 convolutions, each followed by ReLU. The expansive path recovers the spatial dimensions lost during downsampling.
- **Skip Connections:** A key feature of U-Net is the skip connections that link the layers of the contracting path to the corresponding layers in the expansive path. These connections help recover the spatial context lost in the downsampling process, which is critical for accurate pixel-wise classification.
- **Output Layer:** The final layer of the network is a 1x1 convolution that maps each feature vector to the desired number of classes.

## 2. SegNet

After exploring the U-Net architecture for its precise segmentation capabilities, we shift our focus to another pivotal model in our study - SegNet. The selection of SegNet is grounded in its architectural innovations and effectiveness in semantic segmentation tasks, particularly for applications in autonomous driving like lane detection.

The architecture of SegNet consists of the following features:

- **Basic Structure:** SegNet's architecture is a deep convolutional encoder-decoder structure that draws its foundational design from the VGG16 model. It consists of a series of encoder layers followed by a corresponding set of decoder layers and a final pixel-wise classification layer.
- **Encoder:** The encoder in SegNet is essentially similar to the VGG16 network, comprising 13 convolutional layers. These layers are grouped into five sets, each followed by a max-pooling operation. The purpose of the encoder is to capture high-level contextual information from the input images. The key aspect here is that the max-pooling indices (the positions of the maximum feature value in each pooling window) are stored for each layer.
- **Decoder:** The decoder part of SegNet uses the stored max-pooling indices from the corresponding encoder to perform non-linear upsampling of the feature maps. This unique approach helps in better reconstruction of the input image resolution in the segmented output. The decoder also consists of 13 convolutional layers grouped similarly to the encoder, but in reverse order. Each set of decoder layers upsamples its input feature map and then applies convolution to generate denser feature maps.
- **Output Layer:** The final layer of the network is a 1x1 convolution that maps each feature vector to the desired number of classes.

# Experimentation

## 1. Dataset Curation

The dataset, vital for training and evaluating our models, was curated using images generated from the CARLA simulator. The initial step in dataset curation involved splitting the data to create distinct sets for training, validation, and testing. We ensured a balanced representation of various driving scenarios across these sets.

- Training Set: Comprised 60% of the total images, serving as the primary source for model training.
- Validation Set: Accounted for 20% of the images, used to fine-tune model parameters and prevent overfitting.
- Testing Set: The remaining 20% formed the testing set, reserved for evaluating the model performance post-training.

Generated Dataset - 7 towns: 1500 Images

Dataset				
Set	Train	Val	Test	Total
Images	6300	2100	2100	10500
Masks	6300	2100	2100	10500
Proportion	60%	20%	20%	100%

Evaluation Dataset - 1 towns: 1500 Images (Exclusive to model train/test data)

## 2. Preprocessing

Preprocessing is a crucial step to ensure that the data fed into our models is clean and standardized. This phase involved:

- Image Normalization: Scaling the RGB values of the images to a [0, 1] range to facilitate smoother training.
- Mask Processing: Converting the segmentation masks to a format suitable for semantic segmentation. This involved resizing the masks to match the image dimensions and extracting relevant channels that represent lane markings.
- Resizing: Downscaling images and masks to reduce dimensionality of data to make it easier for deep learning networks to learn.

## 3. Model training

The training of U-Net and SegNet models was a crucial phase of our experiment. Each model underwent a rigorous training process using the TensorFlow and Keras frameworks, adhering to specific architectural guidelines and hyperparameters. Our approach to training each model was methodical, ensuring that both models were optimized for the best possible performance on lane detection tasks.

### U-Net Training Process

- **Architecture Setup:** The U-Net model was constructed with a sequence of encoder and decoder blocks, each consisting of convolutional layers with ReLU activation and batch normalization. The encoder blocks used max pooling for downsampling, while the decoder blocks utilized transposed convolutions for upsampling.
- **Compilation:** The model was compiled with the Adam optimizer and binary cross-entropy loss function, suitable for the binary classification nature of our lane detection task. Performance metrics included accuracy, Dice coefficient, and Intersection over Union (IoU).
- **Callbacks:** To enhance training efficiency, callbacks such as Early Stopping and ReduceLROnPlateau were used. Early Stopping prevented overfitting by halting the training if the validation loss did not improve for five epochs, while ReduceLROnPlateau reduced the learning rate when the validation loss plateaued, ensuring finer adjustments to the model weights.
- **Training Execution:** The model was trained for 20 epochs with a batch size of 32, using the prepared training and validation datasets. Data was fed into the model in batches for efficient memory utilization.

### **SegNet Training Process**

- **Architecture Construction:** SegNet's architecture, built on an encoder-decoder structure, involved custom layers for max pooling with argmax indices and corresponding unpooling in the decoder. This unique setup was crucial for capturing and utilizing spatial information efficiently.
- **Compilation Details:** Similar to U-Net, SegNet was compiled using the Adam optimizer and binary cross-entropy loss. Accuracy, Dice coefficient, and IoU were also used as metrics.
- **Callbacks Implementation:** The same set of callbacks used for U-Net, including Early Stopping and ReduceLROnPlateau, were employed for SegNet to ensure optimal training performance.
- **Training Process:** SegNet's training was conducted for 20 epochs with a batch size of 32, considering the model's architectural complexity and memory requirements.

Both models' training was monitored for performance metrics on the validation set, ensuring that the models were learning to generalize beyond the training data. The ModelCheckpoint callback was employed to save the best version of each model based on the validation IoU, ensuring that we retained the model iteration with the highest segmentation accuracy.

## **4. Evaluation**

The final stage of our experiment was the evaluation of the trained models. This involved:

- **Testing on Unseen Data:** Both models were tested on the reserved testing set and an additional 'surprise' set obtained from a new CARLA environment, to assess their generalization capabilities.
- **Performance Metrics:** We evaluated the models based on loss, accuracy, Dice coefficient, and IoU. These metrics provided insights into the models' effectiveness in segmenting lane markings accurately.
- **Visual Inspection:** Besides quantitative metrics, we also conducted a visual inspection of the predicted segmentation masks against the ground truth. This qualitative evaluation helped in understanding the practical applicability of the models in real-world scenarios.

# **Results**

## 1. Quantitative Results

- U-Net

Model 1: U-Net (20 epochs - early stopping and checkpointing)

DataSet	Set	dice	iou
Original Dataset	Train	84.51	93.40
	Val	84.52	93.33
	Test	84.63	93.39
Evaluation Dataset	full	66.10	88.77

- SegNet

Model 2: SegNet (20 epochs - early stopping and checkpointing)

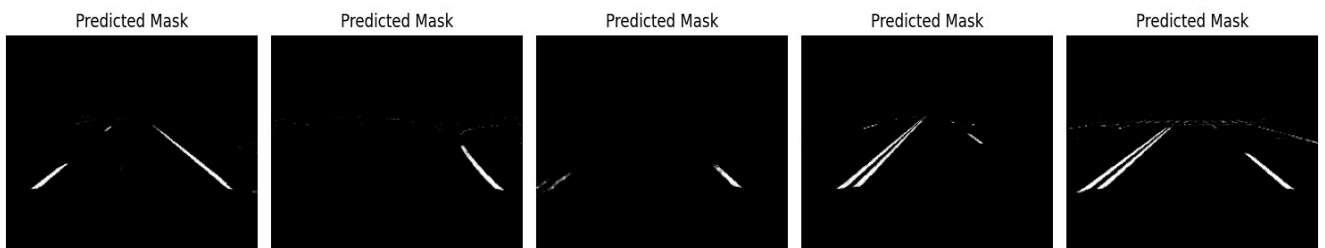
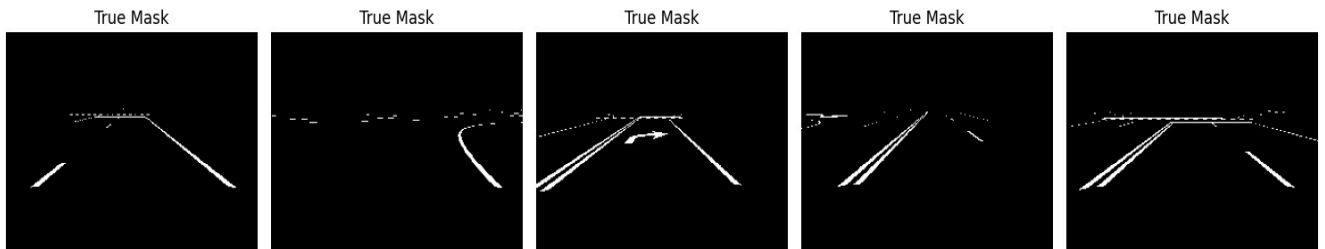
DataSet	Set	dice	iou
Original Dataset	Train	73.75	89.17
	Val	73.93	89.26
	Test	73.55	89.28
Evaluation Dataset	full	43.57	79.18

**Architecture efficiency:** U-Net's architecture is more efficient in learning relevant features from the given dataset due to its symmetric structure and the use of skip connections. These connections help combine low-level feature details with high-level context, which is critical in semantic segmentation for capturing the precise shape and location of lanes. U-Net achieved Dice coefficients of 84.51%, 84.52%, and 84.63% on the train, validation, and test sets respectively, and an IoU of 93.4%, 93.33%, and 93.39% on the same sets, indicating its proficiency in capturing lane details. In contrast, SegNet's decoder does not benefit from skip connections, potentially leading to less accurate reconstruction of the segmentation maps from the compressed feature representation through relaying pooling indices only. This is reflected in its lower Dice scores of 73.75%, 73.93%, and 73.55% and IoU scores of 89.17%, 89.26%, and 89.28% on the train, validation, and test sets respectively.

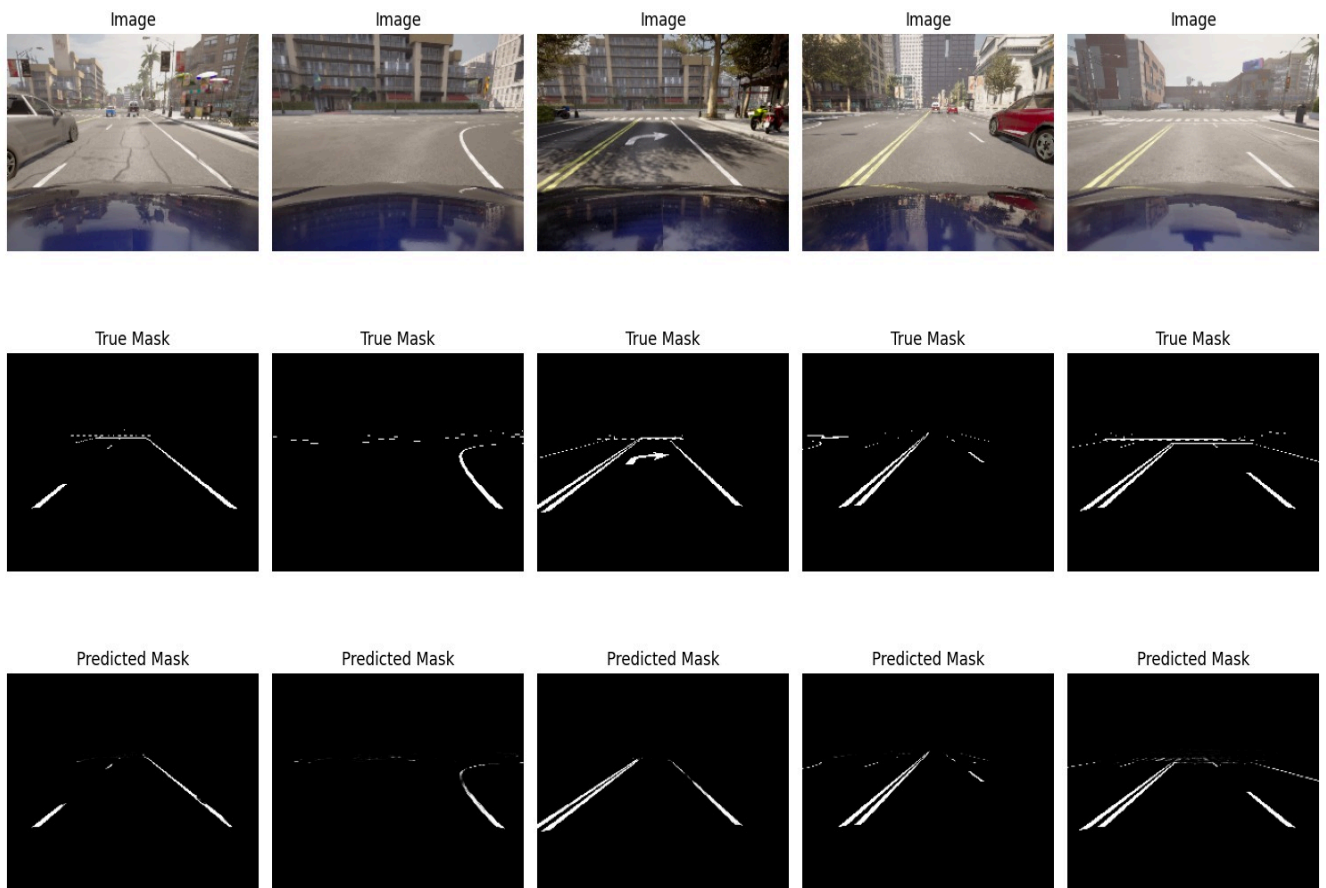
**Generalization Ability:** The drop in performance on the full evaluation dataset suggests that while both models generalize beyond the training data, U-Net does so more effectively, with a Dice score of 66.1% and an IoU of 88.77%, compared to SegNet's considerable decline. U-Net's ability to preserve and utilize fine-grained spatial information is crucial for generalizing to images of unseen towns or varied environmental conditions produced by the CARLA simulator.

## 2. Qualitative Results

- U-Net



- SegNet



Upon observing the visual predictions of U-Net and SegNet in comparison to the true masks, we can see that U-Net is capable of capturing fine grained details of the lane marking much better than SegNet. This can again be explained by the architecture differences between the two networks that help U-Net in creating a more precise and accurate segmentation map.

## Conclusion

The exploration into lane detection using U-Net and SegNet conducted within the simulated environments of CARLA has yielded significant insights into the efficacy of these deep learning models. Lane detection is a critical component in the advancement of autonomous driving technologies, where the accurate identification and segmentation of lane markings are crucial for navigation and safety. In such a context, the choice of the right model for semantic segmentation becomes pivotal.

CARLA, as a versatile and sophisticated simulator, played a crucial role in this study by providing a diverse and realistic dataset for training and evaluating our models. The ability to simulate various driving conditions, from urban landscapes to different weather scenarios, allowed us to test the robustness and adaptability of U-Net and SegNet in environments that closely mimic real-world conditions. This comprehensive approach to dataset generation ensured that our findings would be relevant and applicable to actual autonomous driving systems.

In our comparative analysis, U-Net distinguished itself as the more capable model for high-fidelity semantic segmentation in the context of lane detection. Its architecture, featuring a symmetric design with skip connections, excelled in maintaining essential feature details throughout the network. This capability translated into superior performance in both quantitative and qualitative



assessments, particularly in preserving the intricacy and accuracy required for lane detection. On the other hand, SegNet, while demonstrating commendable segmentation capabilities, fell short in aspects of feature propagation and generalization. These findings highlight U-Net's potential as a more suitable and reliable choice for lane detection tasks in autonomous driving, underscoring the importance of architectural considerations in the development of deep learning models for such critical applications.

## References

- [1] LaneNet: Real-Time Lane Detection Networks for Autonomous Driving (<https://arxiv.org/abs/1807.01726>)
- [2] SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation (<https://arxiv.org/abs/1511.00561>)
- [3] CARLA: Open-source simulator for autonomous driving research (<https://carla.org/>)

---

© Krishna Kanth Vuppala Narasimha, Beenaa Motiram Salian, Sriamoghavarsha Beerangi Srinivasa

---