Traffic Lane Detection using Instance Segmentation

Project by:

Garvit Singh - 2019A7PS0073G

Mohit Dharmik - 2019A7PS0032G

Notebook Link:
https://drive.google.com/file/d/13WtOLQMvhHOp3hY_OOQlig3okMiETurk/view?usp=sharing

Git-hub: https://github.com/Garvit-Singh/Road-Lane-Detection

Introduction

In digital image processing and computer vision , **image segmentation** is the process of partitioning a digital image into multiple **image segments**, also known as **image regions** or **image objects.** Image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image. Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different color respect to the same characteristics.

Image Segmentation consists of two major approaches:

- **Semantic segmentation** is an approach detecting, for every pixel, belonging class of the object. In our case, there are 2 classes pixels, namely the lane class and the background class. Thus the approach need to just classify each pixel in the image as one of the 2 class.
- **Instance segmentation** is an approach that identifies, for every pixel, a belonging instance of the object. It detects each distinct object of interest in the image. In our case, if there are multiple lane boundaries, pixels belonging to one lane boundary

form and instance of the lane class, and different lane boundaries are thus different instances of the same.

Aim

The aim of this project is to create an image segmentation model that is able to detect the distribution of lanes in road images by highlighting the lane boundaries.

The data set used here is “**Lane Detection for Carla Driving Simulator”** which consists of images that were generated with the Carla driving simulator . The *training images* are images captured by a dashcam that is installed in the simulated vehicle. The *label images* are segmentation masks of the lane boundaries in the training image.

Our goal was to create a model that learns to segment the lane images (both seen and unseen) and create results with as high accuracy as possible.

Method

We used Autoencoders to create our model. We chose 2 approaches for this purpose, a UNET and a custom-made autoencoder. UNET is a type of CNN that will be used to perform semantic segmentation. The custom-made autoencoder is a modification of UNET with skip connections and constant channels across the decoders. (This was the model suggested in the research paper). This version was used to perform instance segmentation of the images.

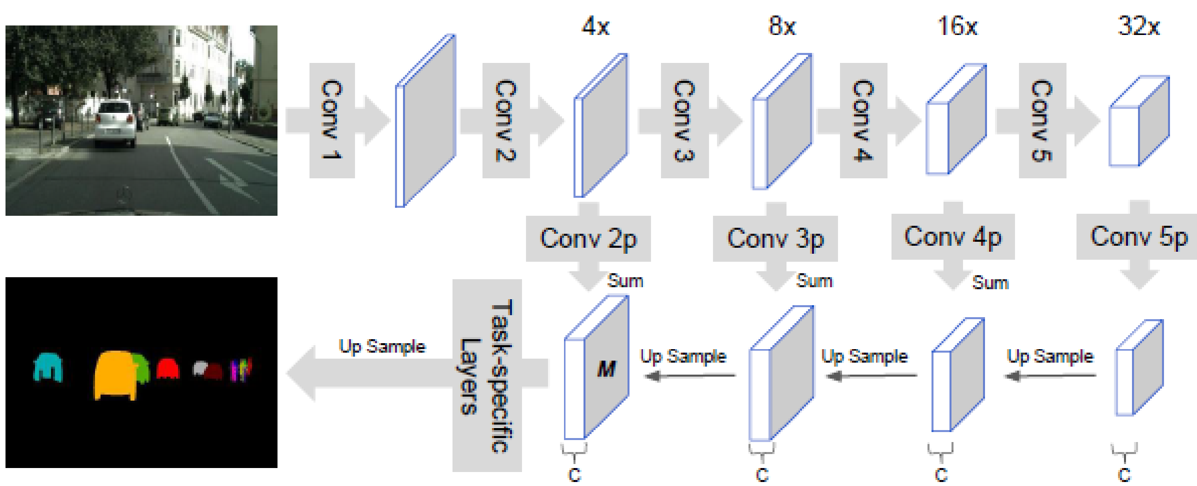


Fig. 4. The network architecture used in this work.

Then the next step was to explore different loss functions and parameters that would give best results for our task. Therefore we tried different loss functions on top of the above 2 approaches and analyzed the respective results.

Results

Time Required for

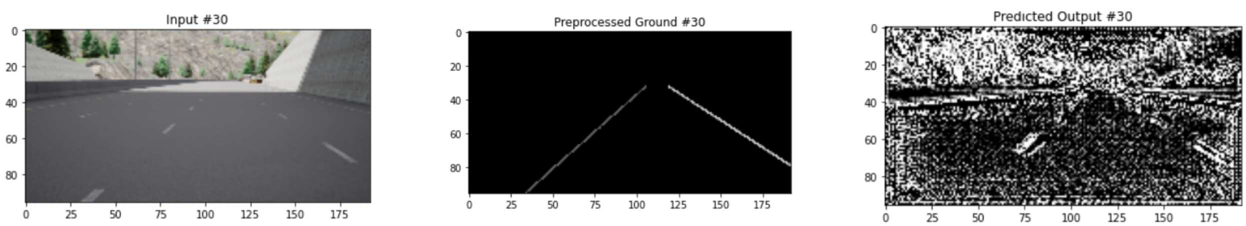
1. Data Loading: 15 - 20 minutes
2. Data Processing: around ~5 seconds for semantic ground truth and ~1.5 minutes for instance ground truth.

The following are the observed results across different loss functions used with the 2 chosen approaches (Semantic and Instance Segmentation models):

1. <u>Mean-Squared Error Loss with Semantic Model</u>

   Mean Validation Accuracy: 47.78 percent

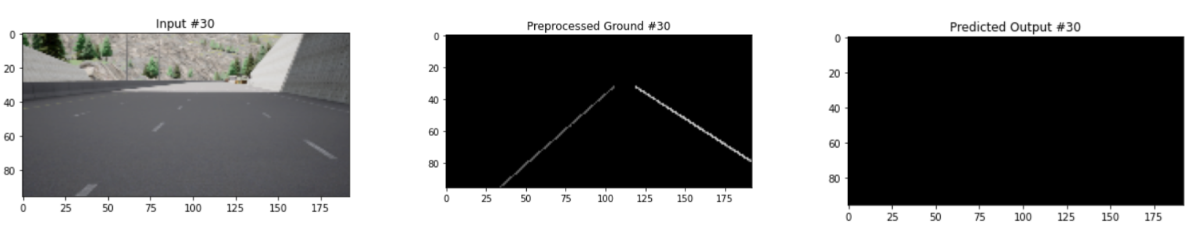
   Predicted Output:



2. <u>Binary Cross Entropy Loss with Semantic Model</u>

   Mean Validation Accuracy: 97.93 percent

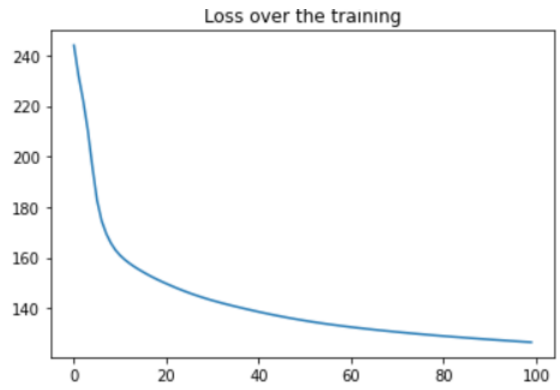
   Predicted Output:



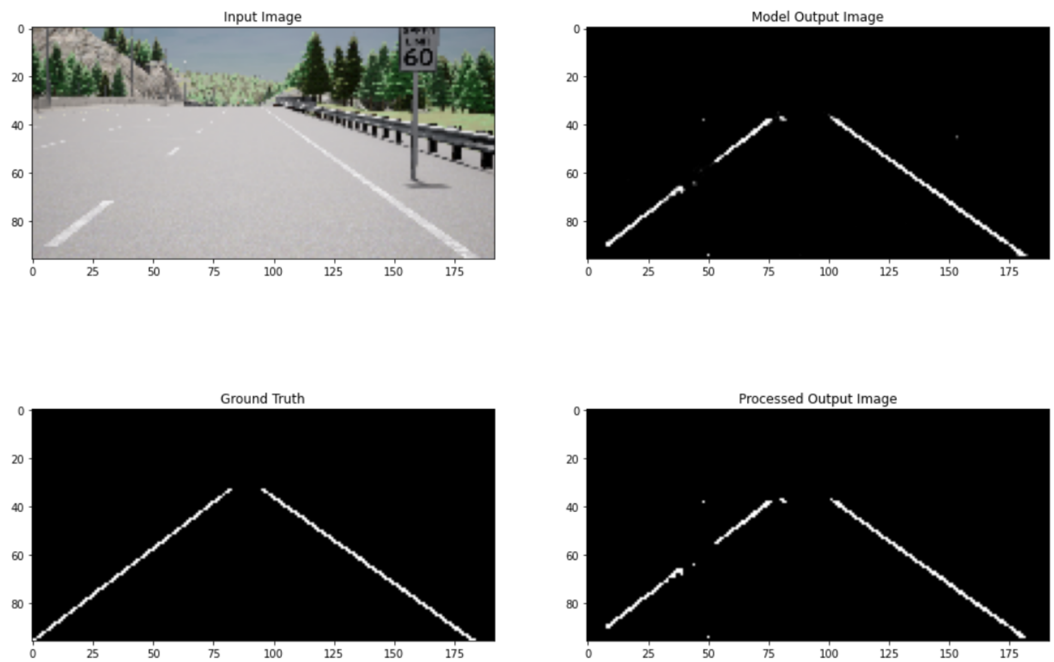
2. <u>Binary Cross Entropy Dice Loss with Semantic Model</u>

Mean Validation Accuracy: 99.14 percent

Loss over training epochs:



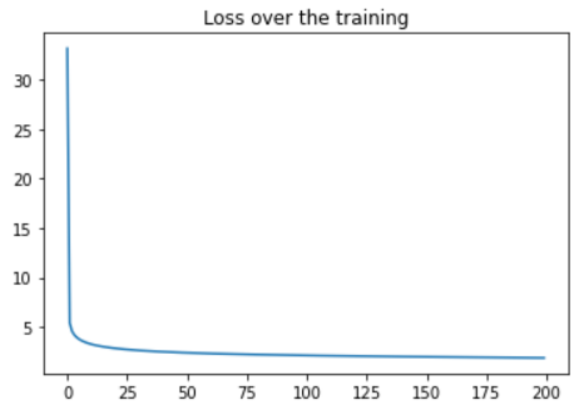
Predicted Output:



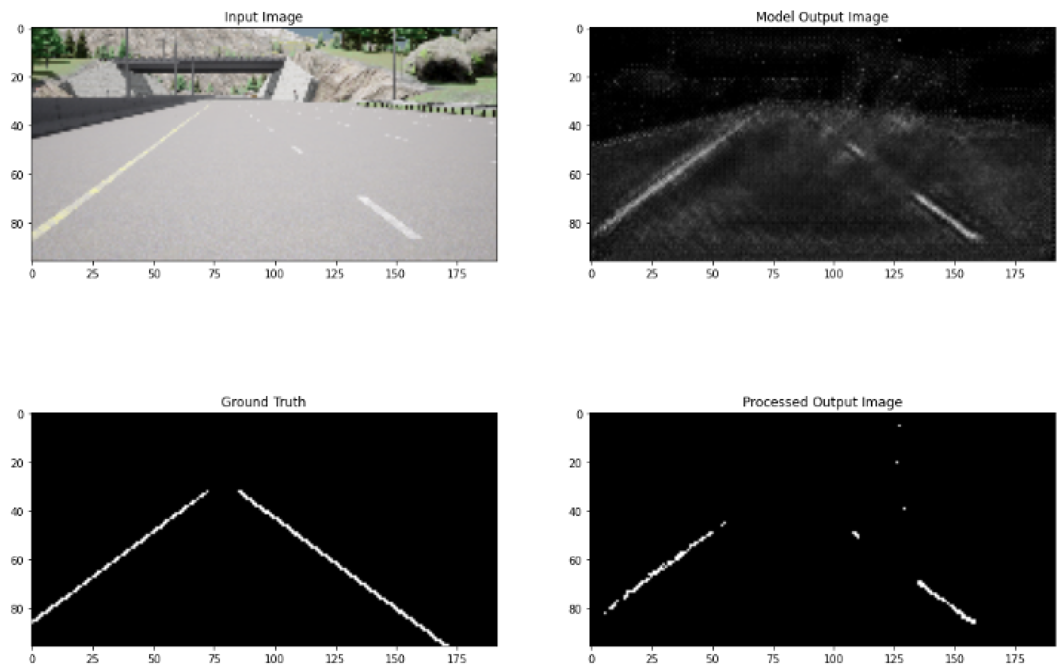
3. <u>Focal Loss with Semantic Model</u>

Mean Validation Accuracy: 98.76 percent

Loss over training epochs:



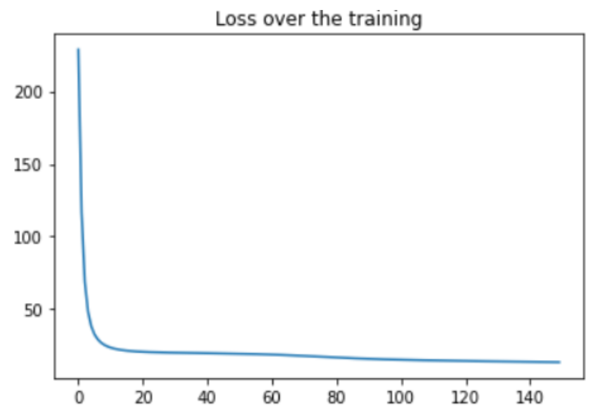
Predicted Output:



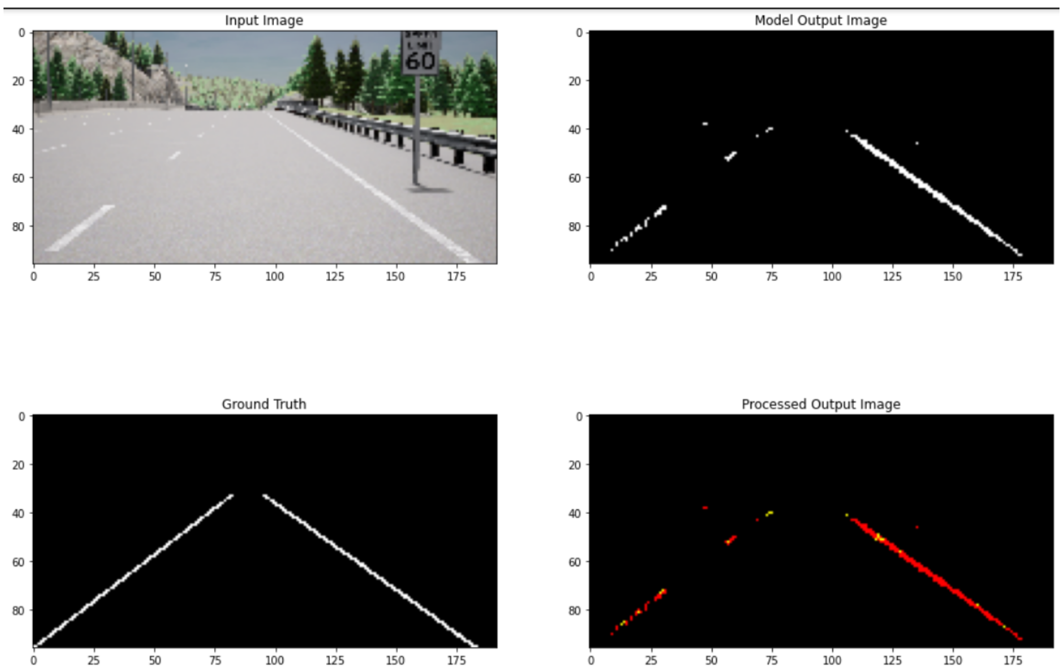
4. <u>Focal Loss with Instance Model</u>

   Mean Validation Accuracy: 98.62 percent

   Loss over training epochs:



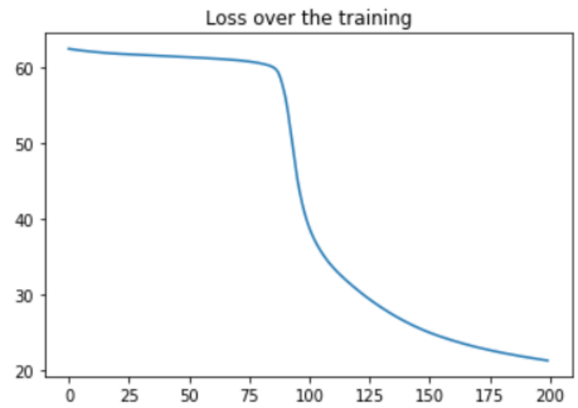
Predicted Output:



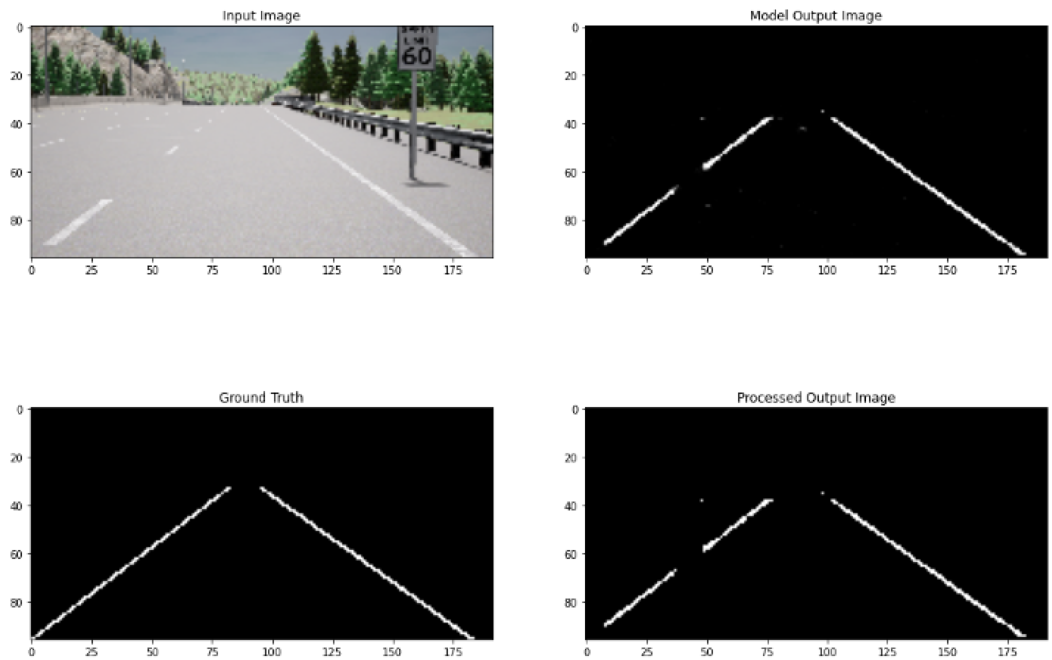
5. <u>Intersection over Union with Semantic Model</u>

   Mean Validation Accuracy: 99.09 percent

   Loss over training epochs:



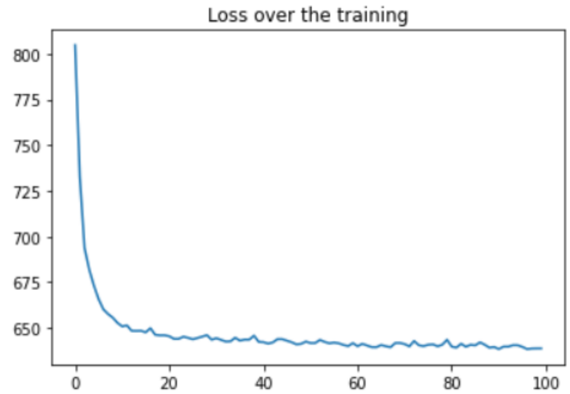
Predicted Output:



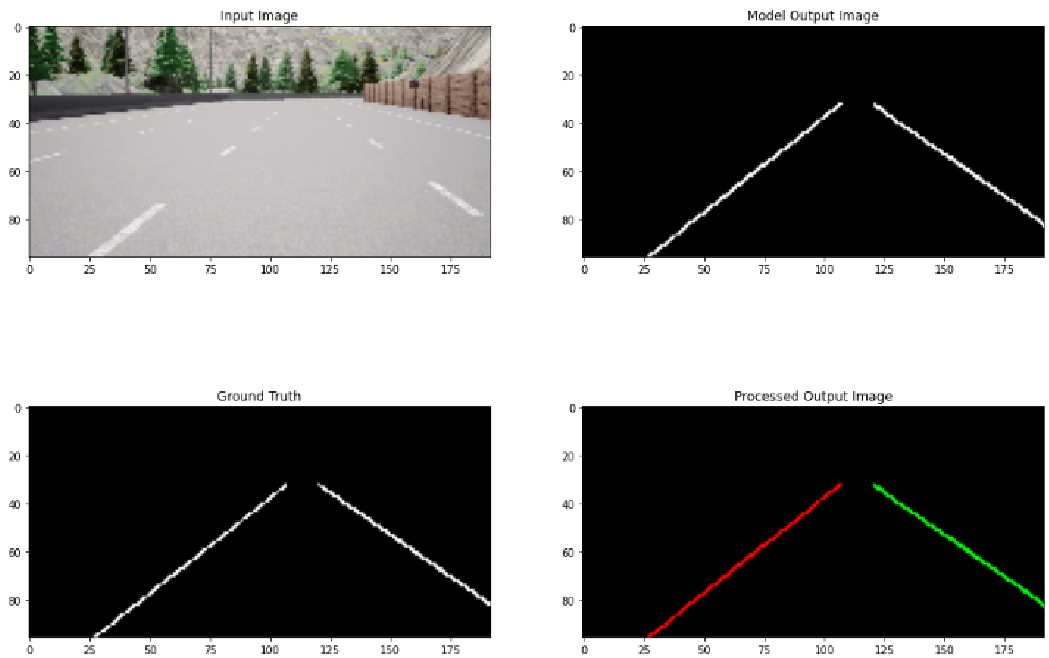
6. <u>Intersection over Union with Instance Model</u>

Mean Validation Accuracy: 99.50 percent

Loss over training epochs:



Predicted Output:



Challenges

The major 2 challenges faced were:

1. <u>Disproportion of class sizes</u> : As the number of white pixels in the label images were very less as compared to the number of black pixels with an approximate ratio of 2:98 percent respectively, most of the loss functions gave completely black image as output since the validation accuracy in this case is around 98 percent due to the dominating matches of the actual black pixels.

2. Research Paper based Loss Functions : The loss functions suggested in the research paper involved a combination of BCE to handle the background class of pixels, KV Divergence and Hinge Loss for instance classification on top of that. The major issue we faced was the large number of pixel pairs present. We tried several approaches to tackle this issue:

    a. Random Sampling : Out of all the pixel pairs, we sampled x number of pixel pairs that should contribute to the overall loss. We tried to change the value of x, but failed to get good results since the number of black-black pixel pairs dominate the pool in the random sampling. Larger values of x took too much time to train (still not giving good results) and smaller x gave all black images since it became similar to BCE.

    b. Max Distance Pairs : We constraint the pixels pairs chosen must be at most k distance apart. Out of these we performed random sampling to get x pairs. This still failed to solve the issue of dominating black-black pixel pairs.

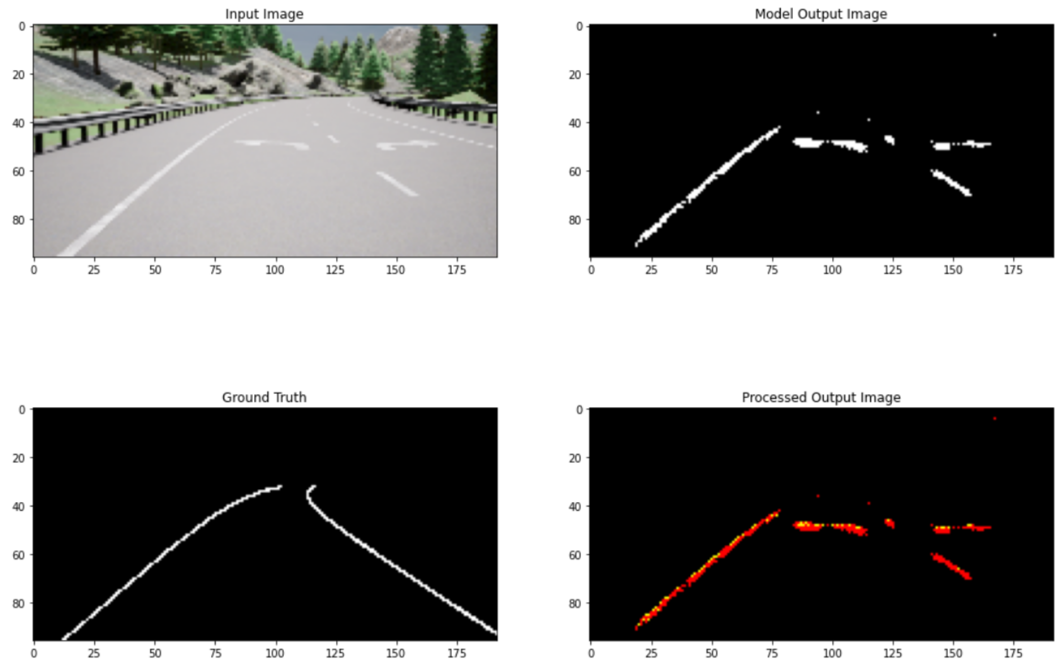
Salient Features

1. MSE Loss : Since this is a very commonly used loss function, this was used to begin with. It failed to get an accuracy above 50 percent thus we conclude it is not a good choice for image segmentation task.

2. Binary Cross Entropy : In order to predict whether the corresponding pixel is a lane or not, i.e., binary classifications using Cross Entropy seemed like a good choice. We noticed that it gave Completely Black Images since this was the local minima which the loss function got stuck to. We concluded this is probably due to the disproportion of class sizes as discussed above.

3. BCE with Dice Loss : We noticed that this gave better results than BCE since it addresses the imbalance problem between foreground and background. We used this along with Focal Loss too, in order to give slight tweak to background channel.

4. Focal Loss : In order to address the disproportionality problem, Focal Loss was chosen. Although it gave very good results in semantic segmentation, it failed to do so in instance segmentation. This was because it, rather than learning lane distribution, it learnt all the white patched on the road, as shown in the below image. Figure a, shows how it learnt white patches on road and Figure b, shows how all instances both green and red got overlapped supporting the same fact.



5. Intersection over Union (IoU) Loss : Since the main goal now was to perform good instance segmentation, performing Intersection over Union to find out each instance separately yielded very good results for instance segmentation. Hence, IoU Loss Function was used.

Contributions

We used the same model as the research paper, but we explored many different loss functions apart from the one suggested in the paper. Additionally we also tried to use these loss functions with a UNET model and analyzed the results. We also tried several approached to make the research paper loss function more efficient but failed to get good results.

References

1. Dataset : https://www.kaggle.com/datasets/thomasfermi/lane-detection-for-carla-driving-simulator
2. Research Paper : https://ieeexplore.ieee.org/abstract/document/8489379
3. Model Architecture :

   https://en.m.wikipedia.org/wiki/U-Net

https://analyticsindiamag.com/my-experiment-with-unet-building-an-image-segmentation-model/