FreiCAR
Practical Autonomous Driving

Team Name: **DiveDeep**

Exercise 3: **Semantic Segmentation**

Repository:

Ajesh Krishnan Kizhakke Menakath(5053372)
Kiran Kumaraswamy(4964003)
Madhu Basavanna(5053269)
Prasad Ramachandra Lakshmi(4971677)

2 Tasks

2.1 Train Semantic Segmentation Network

Semantic segmentation was trained for 50 epoch using SGD optimizer and with dropout of 0.1 for LearningToDownSample and Feature Fusion layer. Best results were obtained for 30 epoch and stayed constant afterwards.
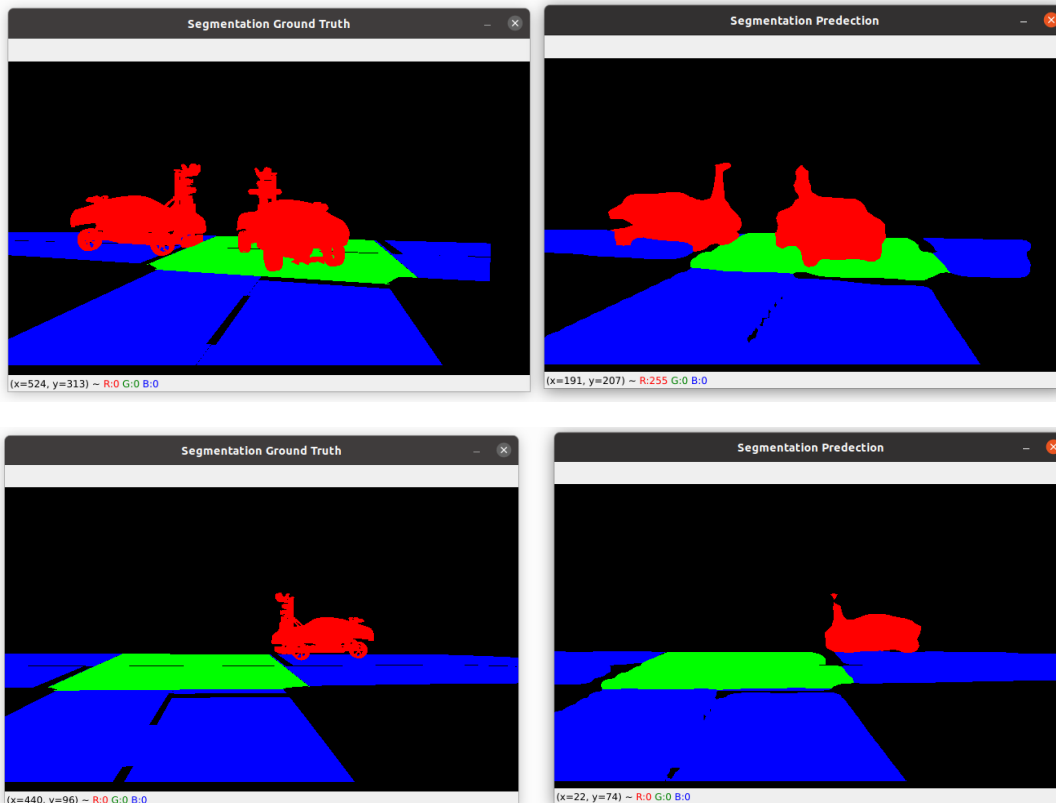
2.2 Semantic Segmentation Evaluation

IoU Analysis

Iou values were captured after each epoch for both training and validation. The results are as below in the table. The least iou accuracy was obtained for the Car class.

	0: Background	1: Road	2: Junction	3: Car	Mean IOU
Training Data	0.9909	0.9726	0.9629	0.8848	0.9528
Validation Data	0.9747	0.9313	0.7880	0.5802	0.8186

Semantic segmentation results:

Semantic segmentation loss and miou for training and validation
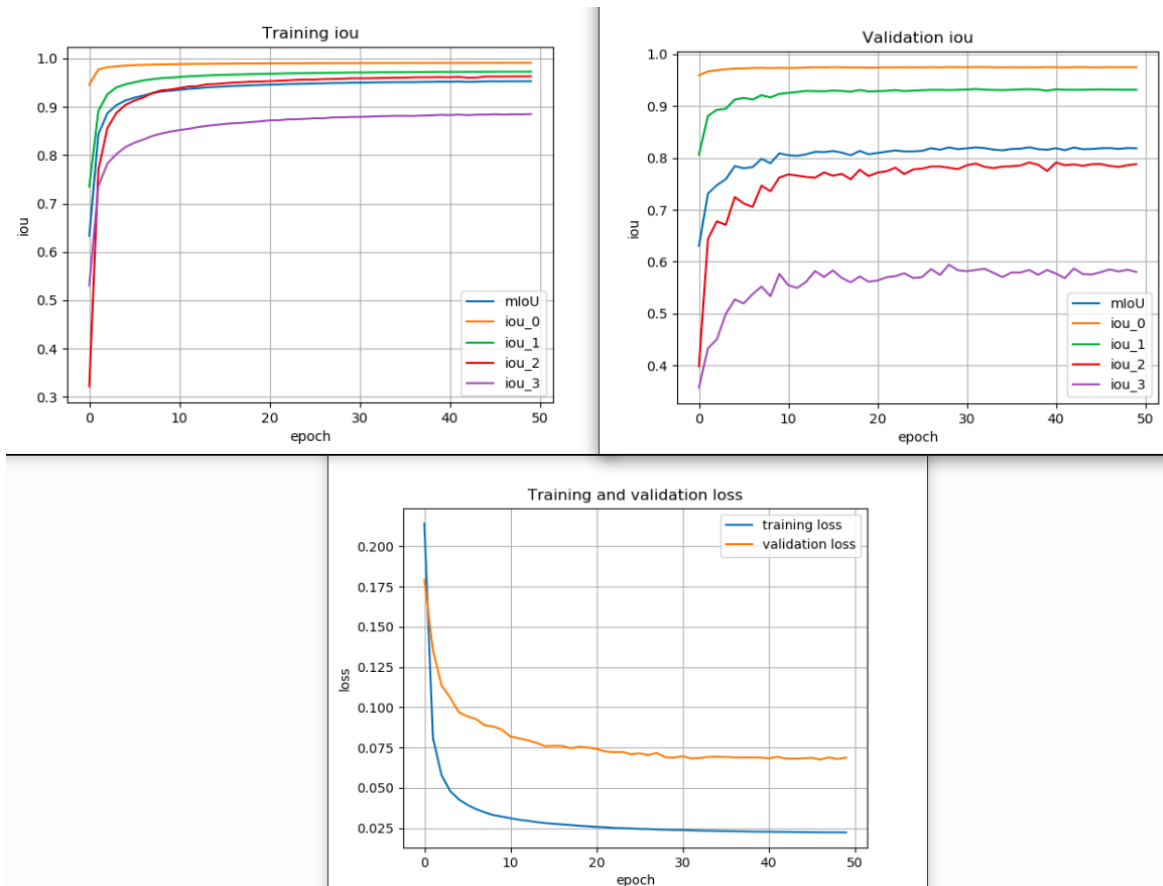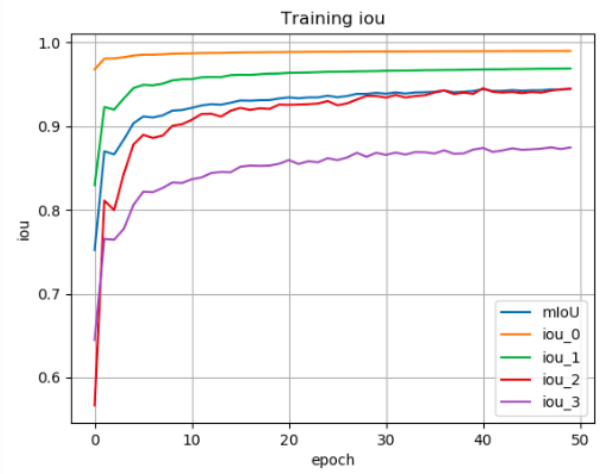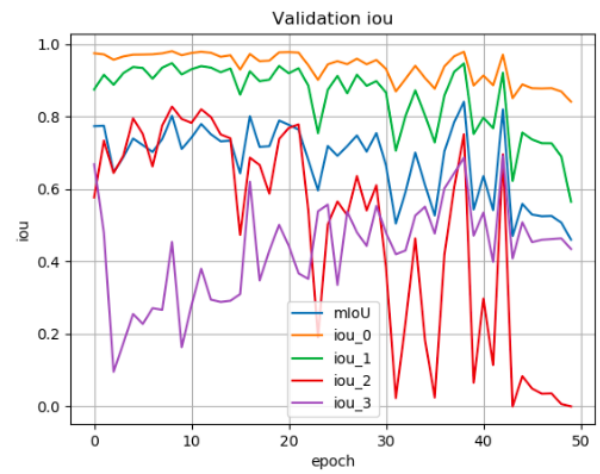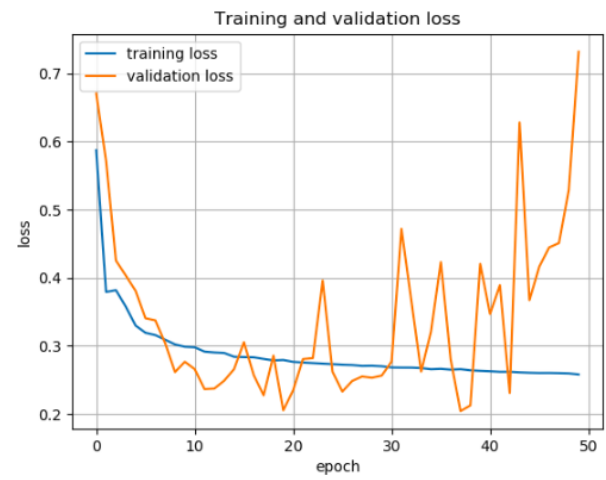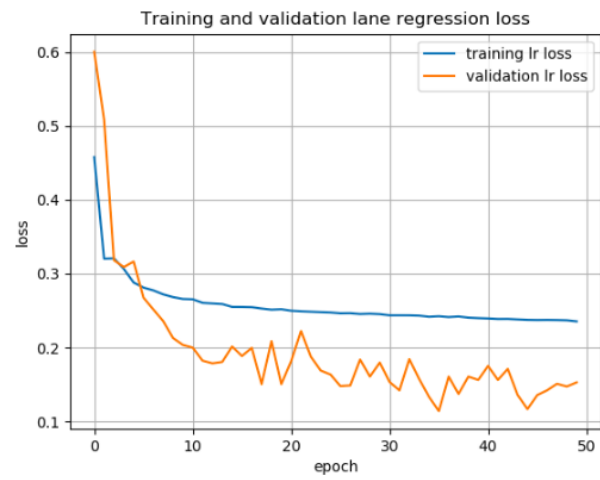

2.3 Add Lane Regression Branch

Semantic segmentation and Lane regression were trained together with separate loss functions with Lane regression loss(L1) loss divided by 16 to compensate for the loss magnitudes. We trained the model for 50 epochs but we got the best model at 38th epoch. We trained the model using Adam optimizer.

2.4 Semantic Segmentation & Lane Regression Evaluation
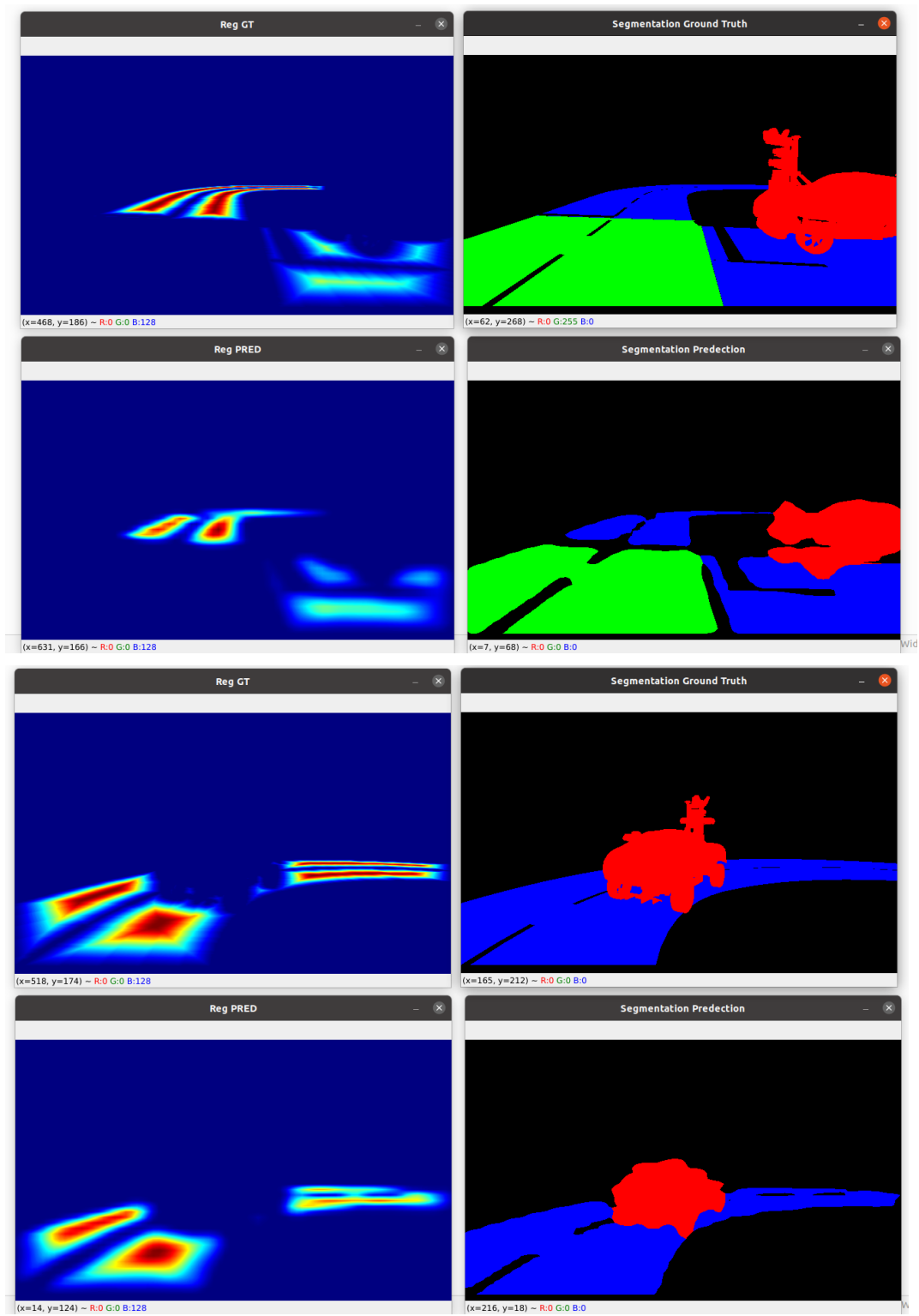
IoU Analysis

	0: Background	1: Road	2: Junction	3: Car	Mean IOU
Training Data	0.9895	0.9674	0.9402	0.8675	0.9411
Validation Data	0.9782	0.9459	0.7507	0.6859	0.8402

Semantic segmentation + lane regression loss and miou for training and validation

Training and validation lane regression loss

Legend: training lr loss, validation lr loss

Training and validation loss

Legend: training loss, validation loss

Validation iou

Legend: mIoU, iou_0, iou_1, iou_2, iou_3

Training iou

Legend: mIoU, iou_0, iou_1, iou_2, iou_3

Semantic segmentation and lane regression results:

2.5 Write ROS node

Ros node has been written and delivered as a separate directory. We have provided the model to run the ROS node in the github in home directory.

How to run the code

- Take a clone of the 02-02-semantic-segmentation-exercise repository from the link 'https://github.com/madhu-basavanna/Freicar_DiveDeep/edit/master/02-02-semantic -segmentation-exercise'

- Run catkin build from the terminal at file location ~/freicar_ws you will find the build of 'ros_pkg' package under ROS folder

- Run 'roslaunch freicar_launch local_comp_launch.launch' from the terminal to launch the simulator

- Run 'roslaunch freicar_agent sim_agent.launch name:=freicar_anyname tf_name:=freicar_anyname spawn/x:=0 spawn/y:=0 spawn/z:=0 spawn/heading:=20 use_yaml_spawn:=true sync_topic:=!' to spawn the car

- Run 'rosrun freicar_executables freicar_carla_agent_node 3'

- Change the directory in the terminal to ~/02-02-semantic-segmentation-exercise/ROS/ros_pkg

- Activate freicar python environment

- Run 'python lane_detector.py --load_model recent_model_19.pth.tar' from the terminal

- The 'lane_detector' node is subscribed to '/freicar_anyname/sim/camera/rgb/front/image', which receives the input image from camera and passes to the model to perform the inference.

- The Semantic Segmentation and Birds Eye View of Lane Regression images are published from node 'lane_detector' to topics 'semantic_segmentation' and 'lane_detection' respectively.