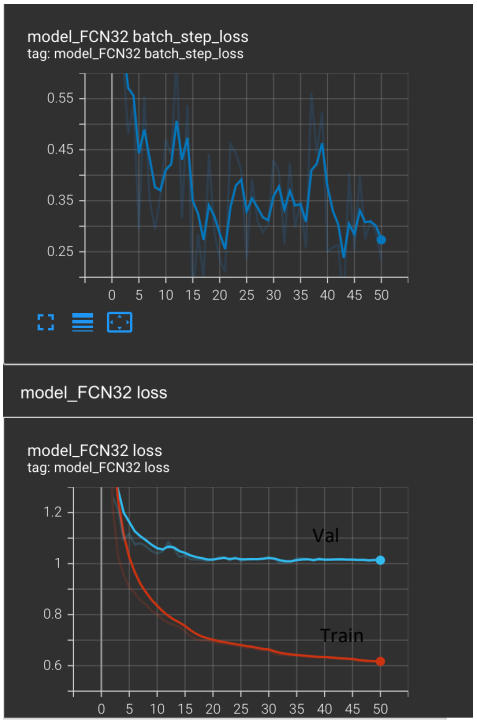
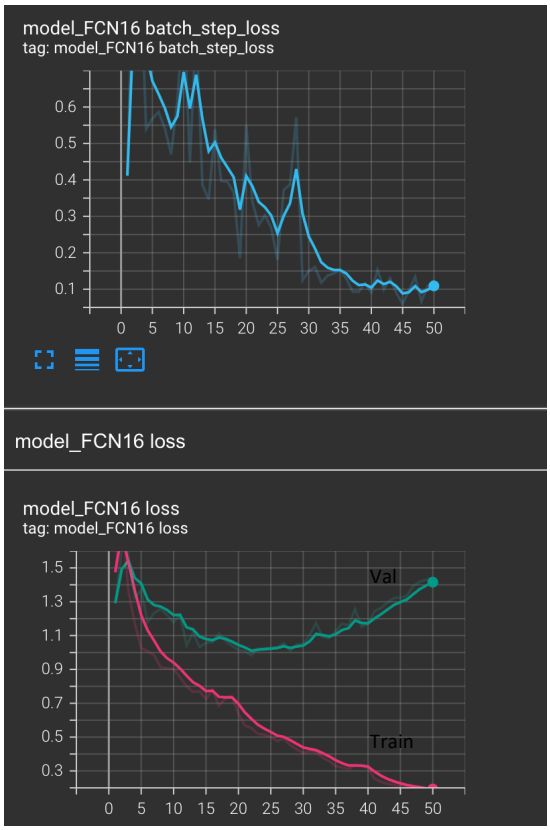
You should turn in a PDF report along with your code in a compressed file with the following components:

1. A brief description of the programs you write, including the source listing.
    a. Split the dataset after sorting the filenames
    b. Using dataloader and batchsize 1 load the dataset into test train val loaders
    c. Train for 50 epochs and log the stepwise, val and train loss on tensorboard for both the networks save the best model with least val loss.
    d. Plot confusion matrix using sklearn libraries confusion matrix function capture values and calculate pixelwise IoU and mIoU
    e. Use the labels.py code to map label to rgb color code
    f. Using the rgb_color_map plot the predictions and ground truth side by side for both the networks
    g. Sources referred:
        1. https://www.kaggle.com/lightfeather/semantic-segmentation-is-easy-with-pytorch : for reading datasets
        2. https://discuss.pytorch.org/t/how-to-extract-features-from-intermediate-layers-of-vgg16/76571 : for getting intermediate layers for fcn16
        3. https://github.com/wkentaro/pytorch-fcn/blob/main/torchfcn/models/fcn16s.py : for understanding fcn networks and crop function
        4. https://discuss.pytorch.org/t/confusion-matrix-for-semantic-segmentation/36238 : for confusion matrix

2. Evolution of loss function with multiple steps.

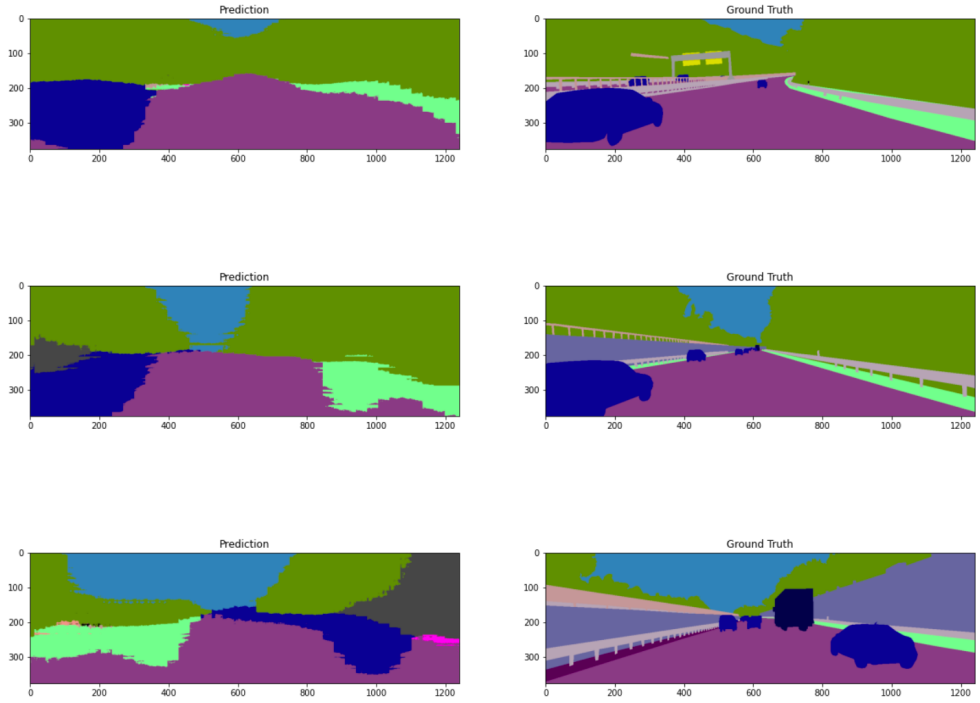


Train

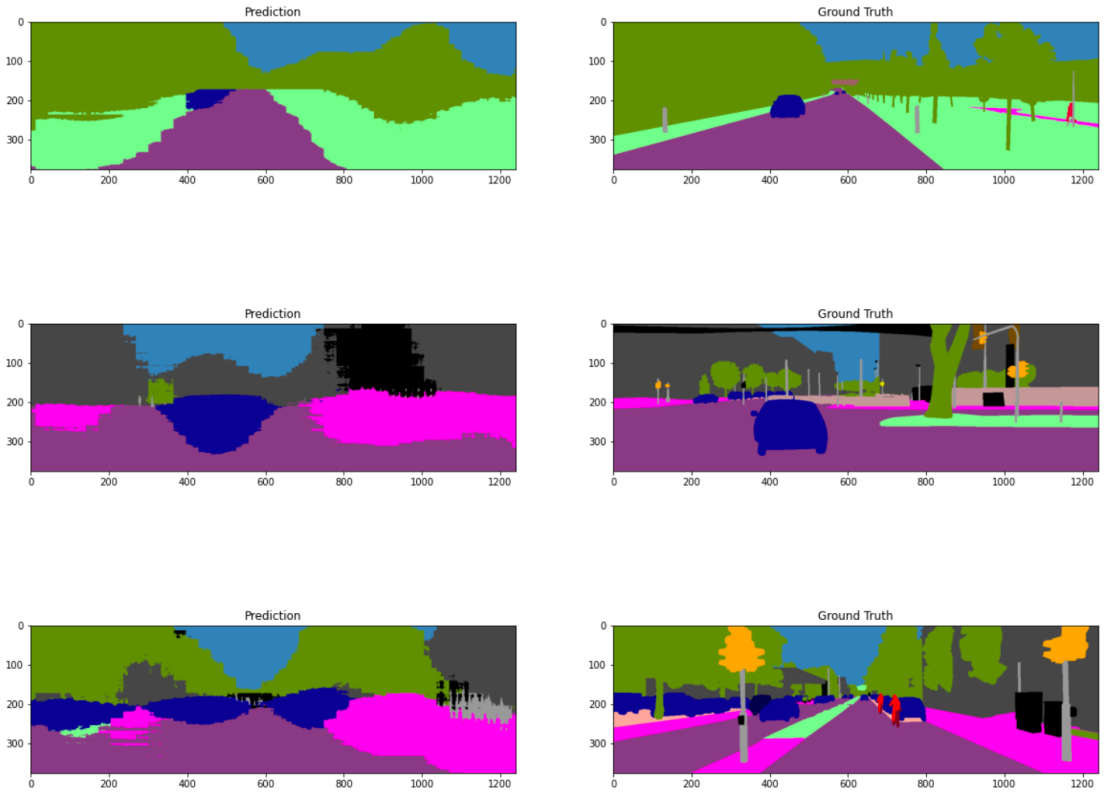


3. A summary and discussion of the results, including the effects of parameter choices. Compare the 2 versions of modified FCN (32s and 16s). Include the visualization of results; show some examples of successful and some failure examples.

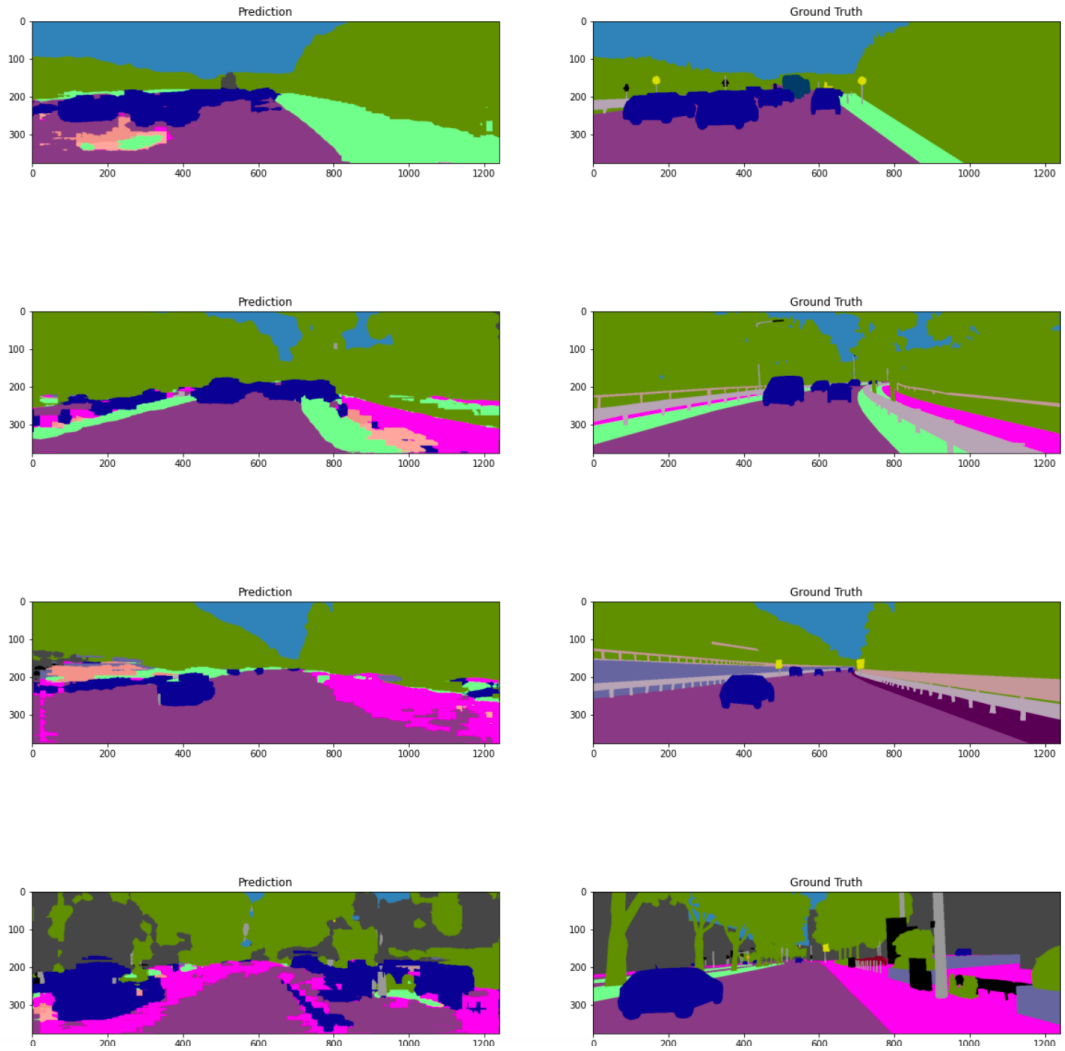
a. The FCN16 clearly gives more finer detailed segmentations, this should be due to the additional spatial information from the previous convolutional layer.
b. L2 Regularization and step size of 15 gave better results for FCN32 whereas for FCN16 step size of 20 and no regularization gave better results.
c. Reducing the number of classes from 35 to 34 where unlabelled class handles the background class instead, improved the model accuracies
d. Using batch size of 1 gave the best results as it did not involve interpolation



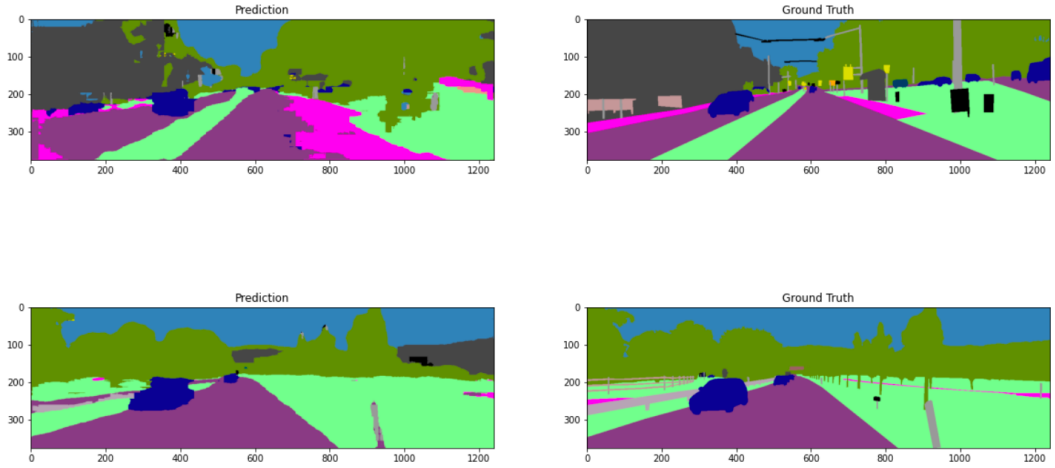
Above are some of the result of the FCN32 network :
The first prediction looks comparatively better than the rest two, we can observe that road class and vegetation has been learn very well by the model but small objects like pole fence etc have not been learnt properly. In general for bigger objects or objects closer to the camera the predictions are more accurate.



Above are few more examples from the FCN32 network, we can see that in general position of cars are learnt well but fine pixel level information is lost and instead of predicting the car's pixels there is a huge blob of car predictions. Same are the first set of pictures the first prediction look very accurate because the vegetation and road class are the predominant classes in the images and they have been learnt well by the network



Above are the results from FCN16 network. We can observe that the network is trying to get more finer detailed predictions and it is also able to predict more number of classes in comparison to FCN32. In the first prediction we can see that the group of cars have been neatly categorized by the network since it is well distinguished from the other classes like greenery and road that it has learnt to segregate well. However in the last image we can see that it is trying to predict car class for a lot of objects it has not come across instead of learning the classes which occur less number of times.



Above are some of the best predictions by the FCN16 network, which closely maps to the ground truth. I.e the classes present in these images have been learnt well with minor false negatives and false positives in comparison to FCN32.