Assignment 5

Report

By Utkarsh Kansal (3683-9858-81)

Task 1: A brief description of the programs you write, including the source listing.

In the given assignment, 2 models were developed for semantic segmentation task. The first model was a modified FCN32 architecture and the second model was a modified FCN16 architecture. Both models employed ResNet18 as its backbone.

A custom Dataset of Kitti was used in this segmentation task. The images were normalized using ImageNet mean and standard deviation, resized into shape (1200, 360), padding of 100 was added on each side, and converted to tensor. Dataset was divided into 3 parts train (70%), validation (15%), and test (15%). DataLoader was instantiated for each of them with batch size of 8.

In FCN32 architecture, last 2 layers of ResNet18 were removed and 3 other layers were added instead namely AvgPool2d, Conv2d, and ConvTranspose2d to upscale the image 32 times.

In FCN16 architecture, the 7^{th} layer of ResNet18 was upscaled by 2 and added to the 6th layer. Their sum was later upscaled by 16 times.

Cross Entropy Loss function along with Adam optimizer was used to train the models for 40 epochs.

After every 5 epochs train and validation loss and mean IoU were displayed.

Functions to plot Loss and Mean IoU vs Epoch were also written. Pixel level IoU for validation and test sets was calculated and shown. Finally, using a colormap, predictions of the model were converted to RGB values and compared with ground truth.

Sources:

1. Lecture Notes
2. https://github.com/pytorch/vision/blob/main/torchvision/models/_utils.py
3. https://github.com/wkentaro/pytorch-fcn/blob/main/torchfcn/models/fcn16s.py
4. https://pytorch.org/docs/stable/nn.html
5. https://discuss.pytorch.org/t/how-to-extract-features-from-intermediate-layers-of-vgg16/76571
6. https://www.youtube.com/watch?v=IHq1t7NxS8k&t=1399s
7. https://d2l.ai/chapter_computer-vision/fcn.html

Task 2: Evolution of loss function with multiple steps.

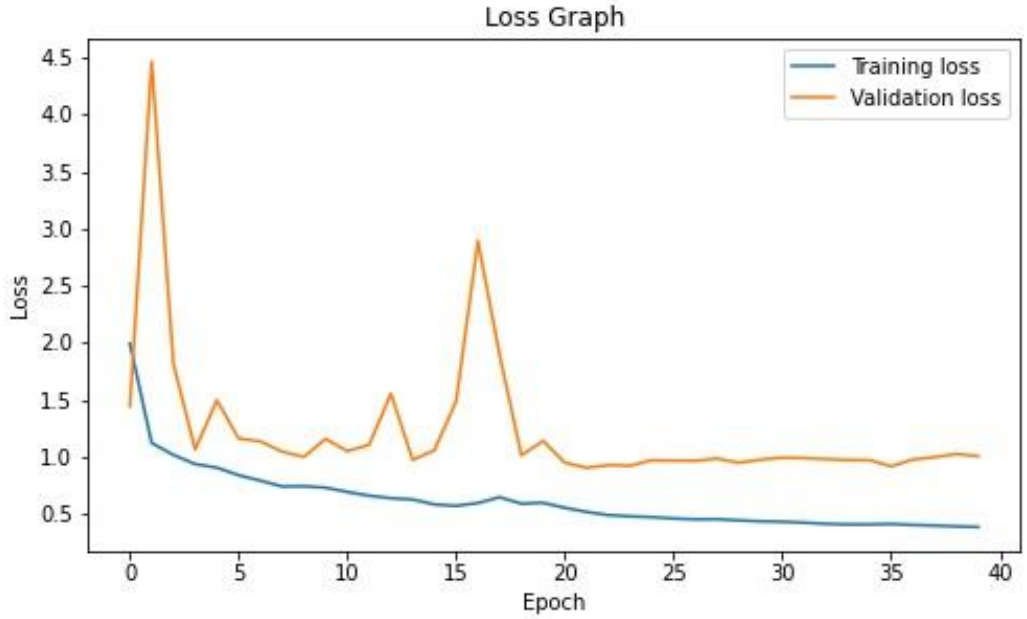

Fig1: Loss Graph for FCN32

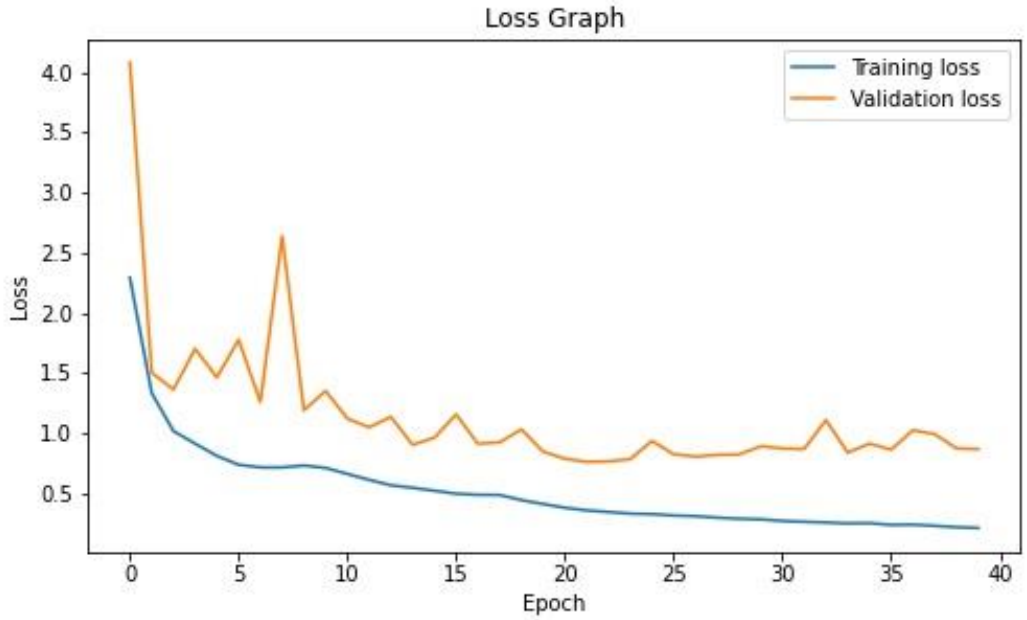

Fig2: Loss Graph for FCN16

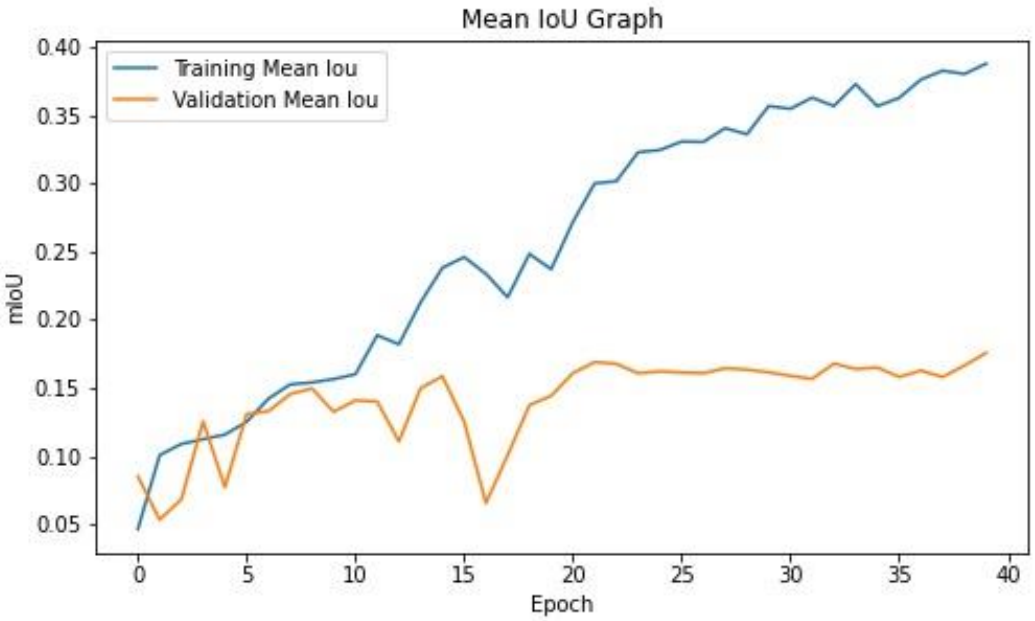

Fig3: Mean IoU Graph for FCN32

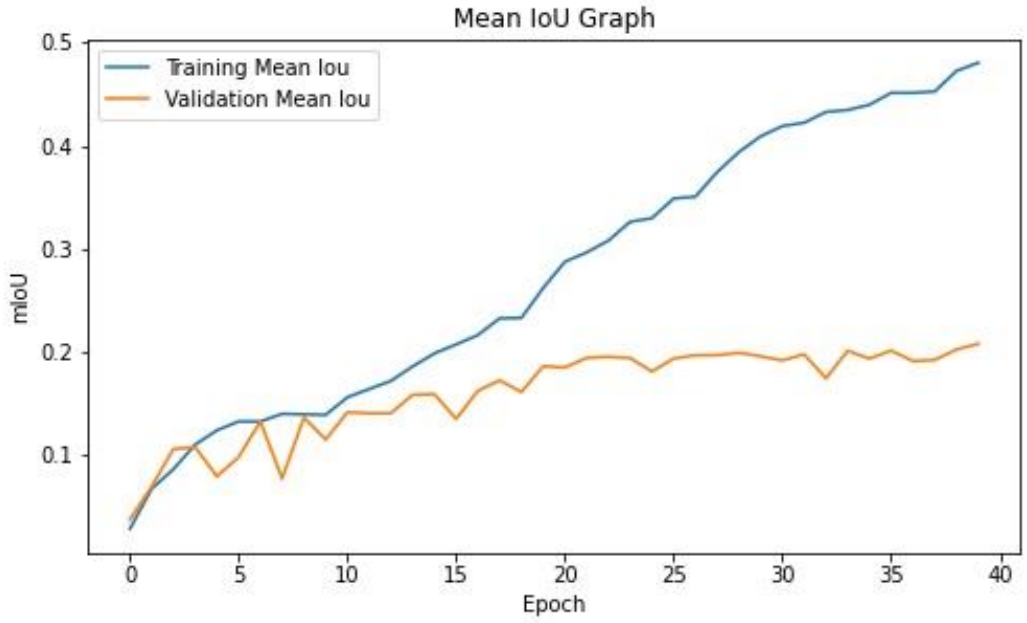

Fig4: Mean IoU Graph for FCN16

Task 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.

- It can be seen from fig5 and Table1 that FCN16 performs much better than FCN32. However, training and prediction time of FCN32 is less than FCN16.
- Various experiments were performed during this assignment. When the images were resized to very small size, it was observed that the performance of both the models depreciated.
- While fiddling with the learning rate, it was noticed that when LR was 1e-2, the models won't converge while training.
- FCN16 gives better IoU than FCN32 in 13 classes while FCN32 has an edge over FCN16 in only 4 classes.

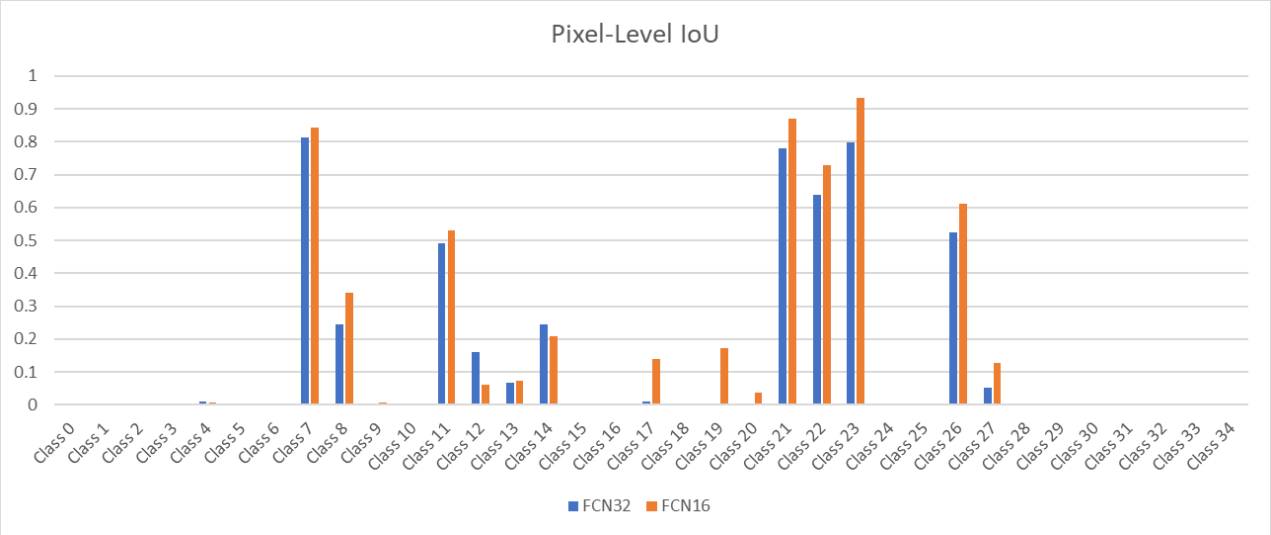

Fig5: Comparison between the Models on Test Set

Table 1: Per-Class IoU for both Models on Test Set

Classes	FCN32	FCN16
Mean IoU	0.1861	0.2108
Class 0	0	0
Class 1	nan	nan
Class 2	nan	nan
Class 3	nan	nan
Class 4	0.0097	0.007
Class 5	0	0
Class 6	0.0003	0
Class 7	0.8128	0.8432
Class 8	0.246	0.3404

Class 9	0	0.0082
Class 10	0	0
Class 11	0.4928	0.5317
Class 12	0.16	0.0604
Class 13	0.0689	0.0737
Class 14	0.2436	0.2093
Class 15	0	0
Class 16	nan	0
Class 17	0.01	0.1384
Class 18	0	0
Class 19	0	0.1713
Class 20	0	0.0369
Class 21	0.7813	0.8687
Class 22	0.6374	0.7299
Class 23	0.7985	0.933
Class 24	0	0
Class 25	0	0
Class 26	0.5234	0.6107
Class 27	0.0538	0.1286
Class 28	0	0
Class 29	nan	nan
Class 30	nan	nan
Class 31	nan	nan
Class 32	nan	nan
Class 33	0	0
Class 34	nan	nan

- Fig6 and Fig7 shows the segmentation result of the 2 models. We can see that both models are able to capture broader aspect of the images.
- FCN16 is able to capture much finer details that that FCN32. However, it fails when an object is occupying very little area in an image.
- If we consider predicted image 1, FCN16 was able to locate the street lamp while FCN32 couldn't do it.
- Both models couldn't identify the traffic light in image 1.
- FCN16 achieved Mean IoU of 0.2108 and FCN32 had 0.1861 on the test data.
- In conclusion, FCN16 performs better than FCN32 but the results can still be improved by various techniques such as data augmentation, tweaking leaning rate, etc.

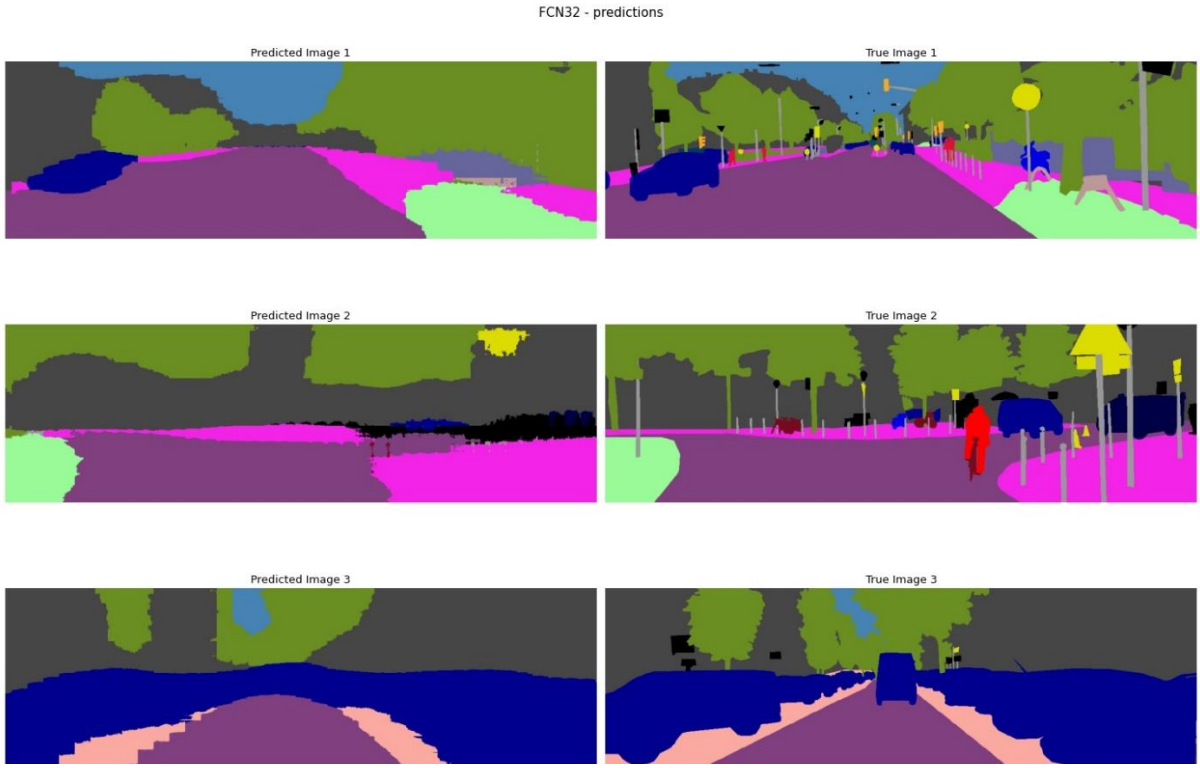

Fig6: FCN32 predictions and comparision with ground truth

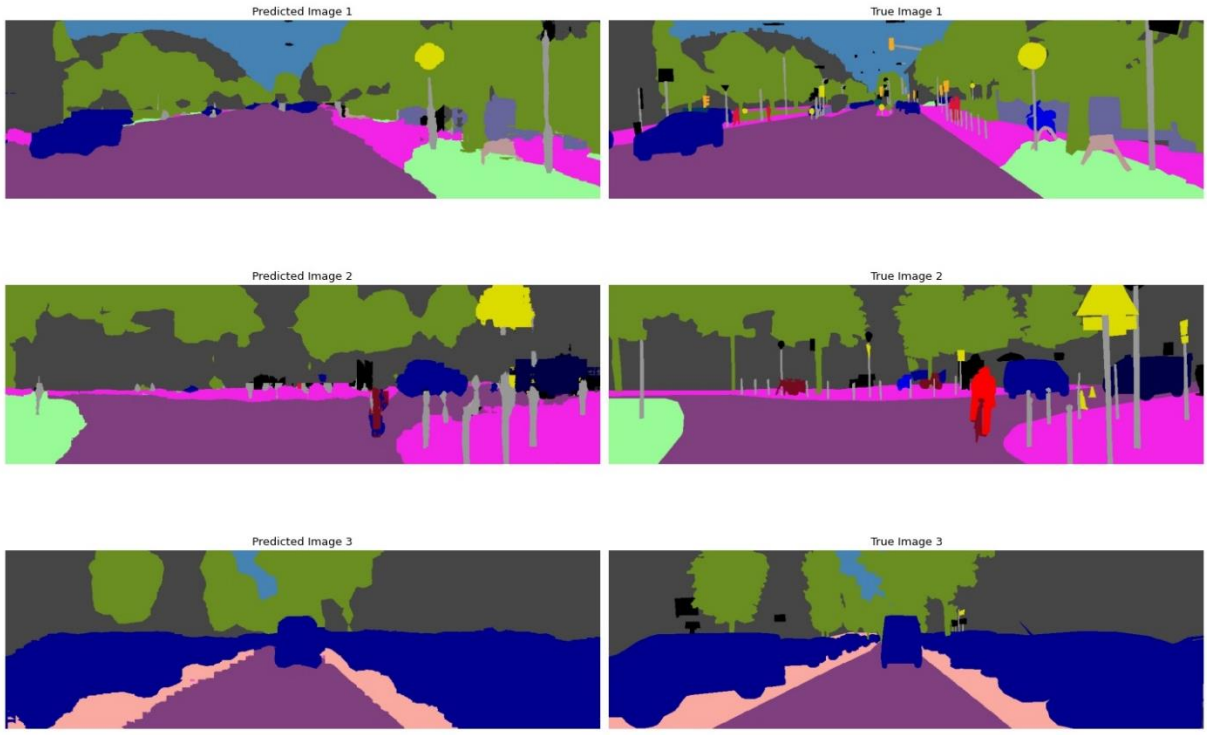

Fig7: FCN16 predictions and comparision with ground truth