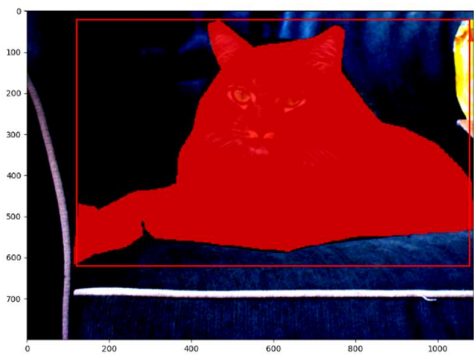
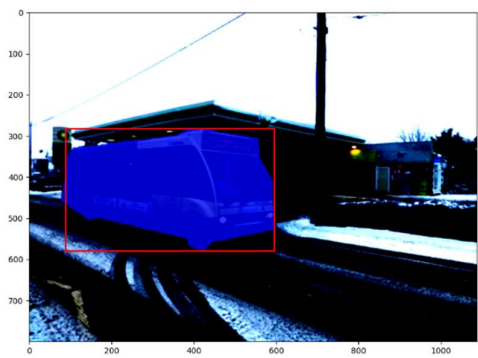


Results of Visualization 5 and 6:

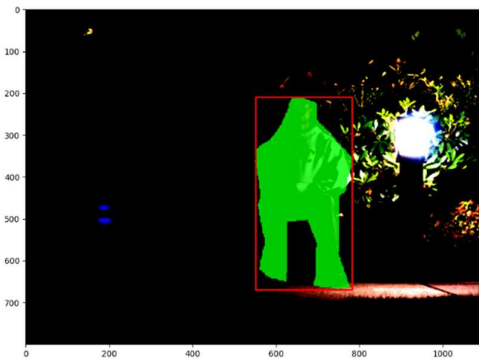
Plots for different classes with masks applied –



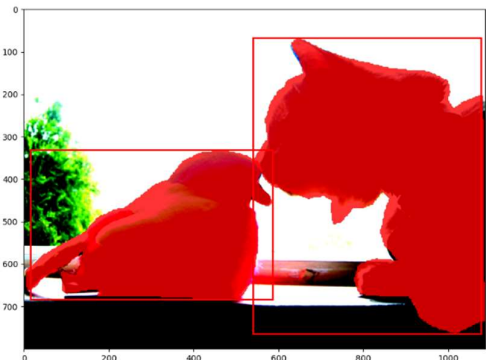
Red class



Blue Class



Green class



Masks are consistent in presence of multiple objects

Plots containing multiple objects of different classes, with masks applied –

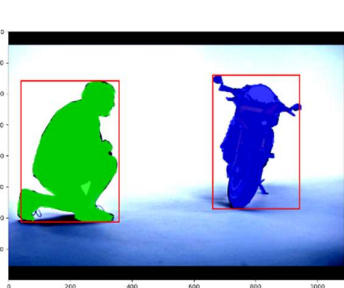
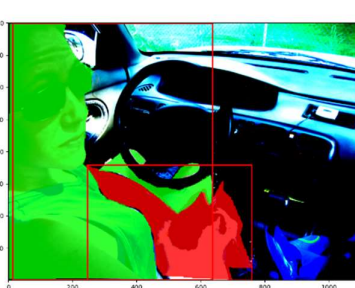
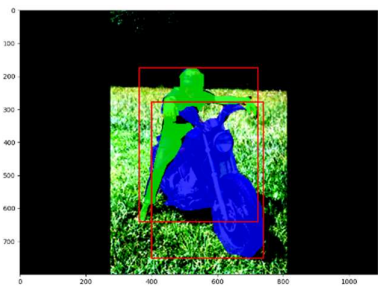
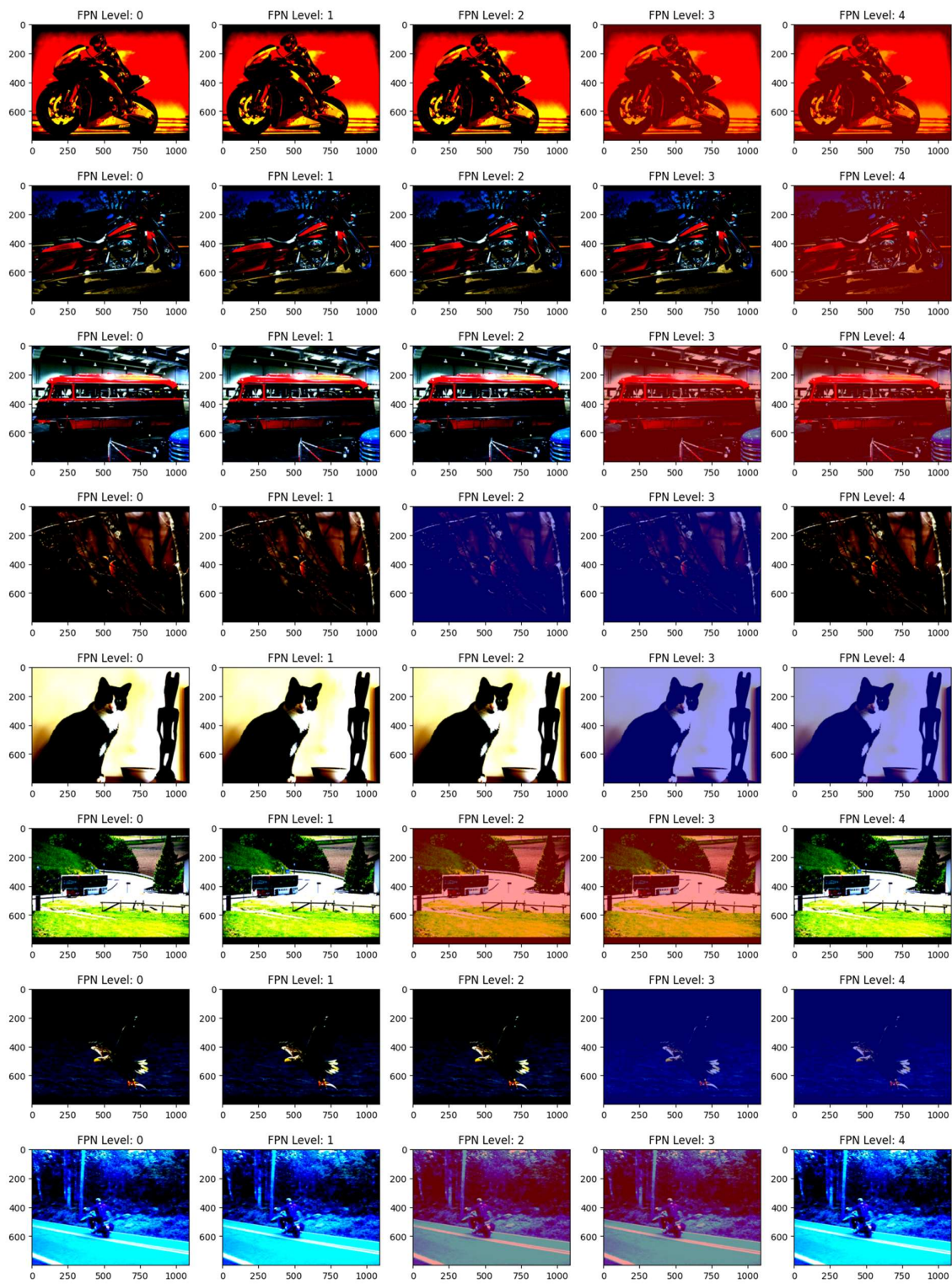
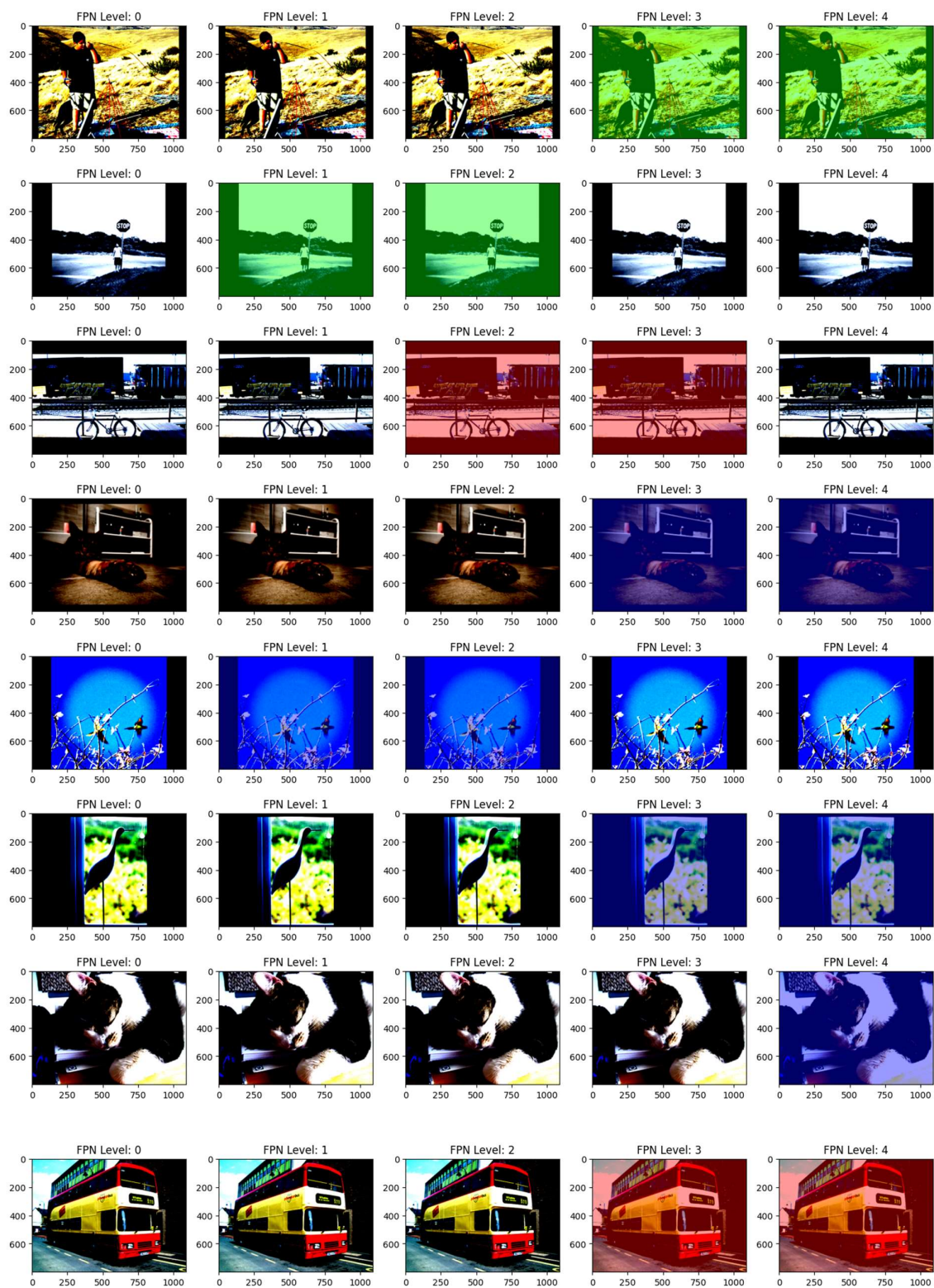


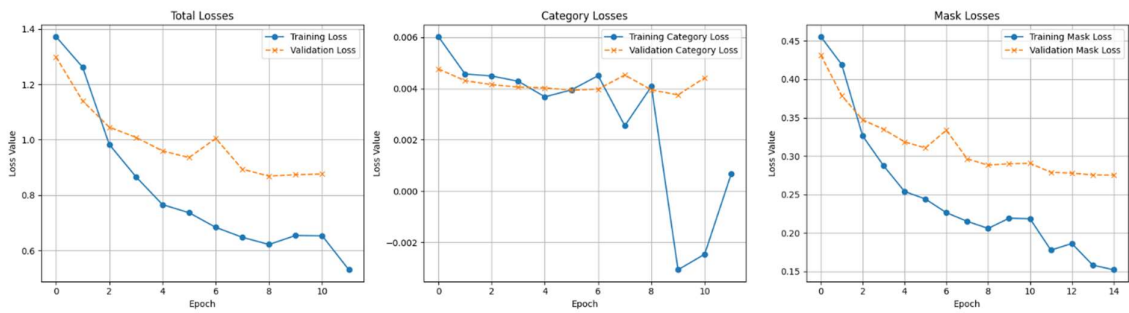
Image containing multiple objects are masked based on their class labels

Plots of target assignments

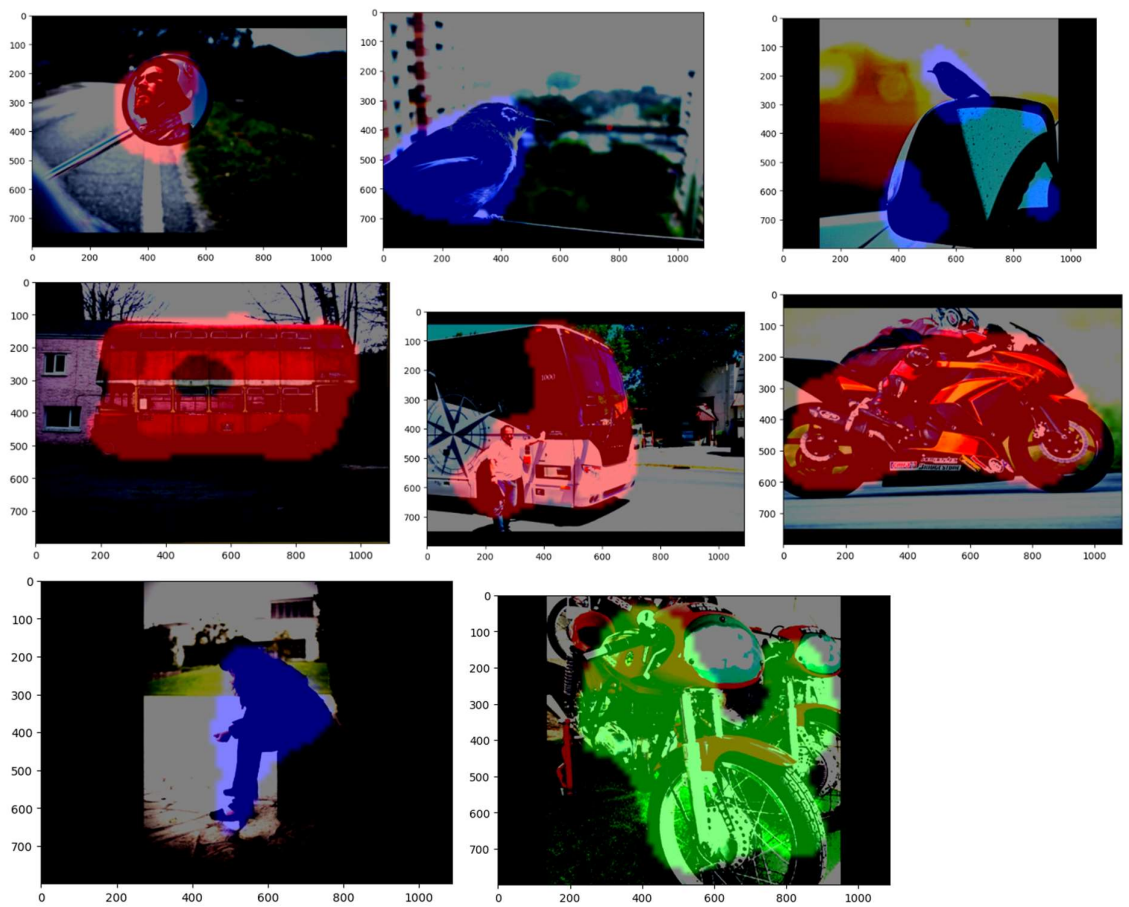




Plots of training and test losses:



Final inference plots



Discussion :

- For this assignment, we limited our training to 15 epochs due to time constraints and challenges encountered using Colab Pro. On two occasions, we exhausted our GPU allocation, interrupting the training and debugging processes. We believe that the model's performance could be enhanced with more training epochs. We were also interested in experimenting with data augmentation, but time constraints prevented us from implementing and testing this potential enhancement. With additional time, we're confident that further refinements could boost the model's performance.
- Another issue we faced was that the debugging was time taking. Since there are many components to this project, it took a lot of time to independently write each of them, test their functionality in accordance with one another. We used Python debugger extensively for this challenge.
- Since Colab Free GPU doesn't have many cores, we built components of our code with batch size 2 and once we had tested on 1-2 epochs, we trained our final model on batch size 16 (Colab Pro) and the optimizer parameters as given to us.
- We also used CSV Loggers and Tensorboard to store our model checkpoints and monitor our losses consistently. Since we used Colab Pro for final training, we also mounted our drive and stored our losses, checkpoints file in drive which ensured that the files were present when compute resources run out.

To execute our project:

- Import dataset.py into the active session. This will present you with visualization 5,6
- For part b, Run the entire notebook in sequence. Please note this includes the training phase, which could be time-intensive. We have left the output code cells for you to see and have plotted the losses, inference results

In the final inference results, we see that the masks are nearly perfect, except or multi object masks. Since we were bounded by resources, we couldn't reach the perfect model. But as you see in the model loss plots, our model has been successful in reducing focal, dice, total loss over the epochs.