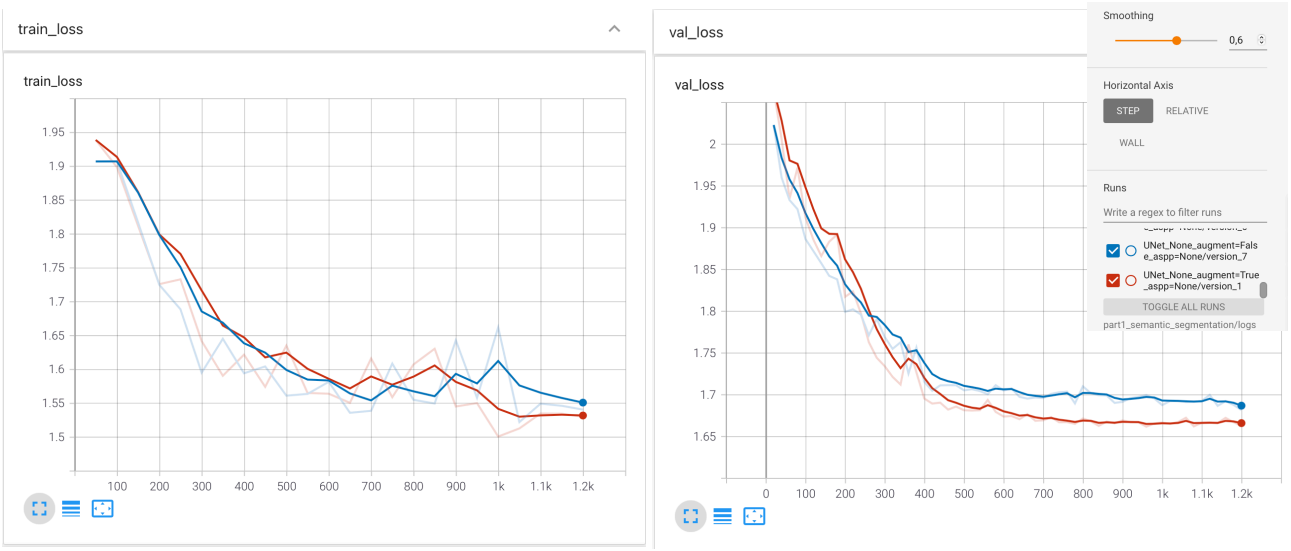
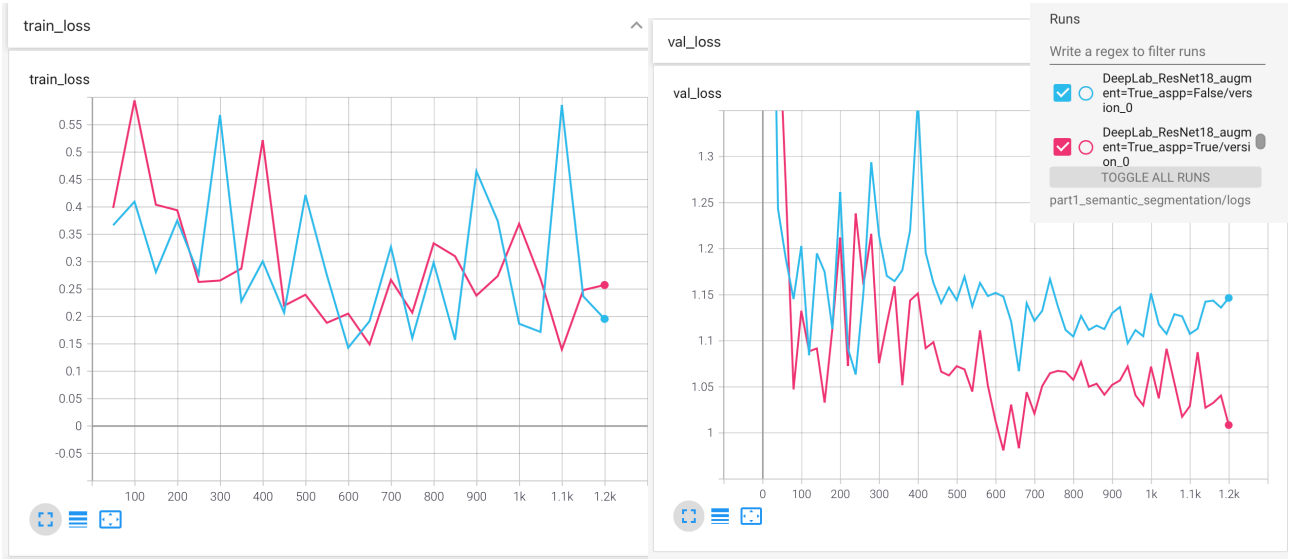
Task 1.
Visualize training loss and validation loss curves for UNet trained with and without data augmentation. What are the differences in the behavior of these curves between these experiments, and what are the reasons?

 # Answer:
UNet with data augumentations (red curves) gives smaller train and val losses than UNet without them (blue curves). By the way, it also gives higher accuracy, recall and IoU. Here n_epochs=60.
With data augumentations we effectively expand the dataset with new (augumented) images, therefore the network learns better.



Task 2.
Visualize training and validation loss curves for ResNet18 trained with and without ASPP. Which model performs better?

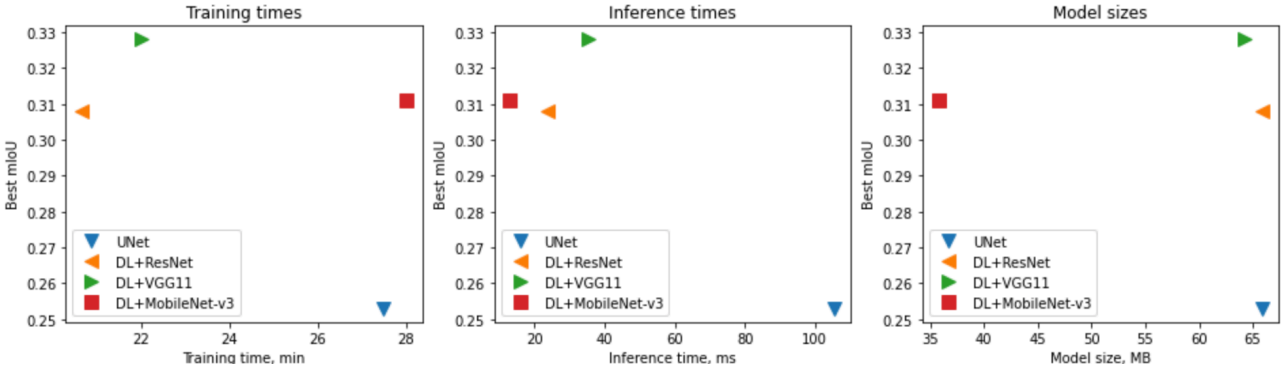
 # Answer:
In terms of validation and training losses, ResNet18 with ASPP (pink curves) performs better. However, all other metrics are higher for ResNet18 without ASPP.



Task 3.
Compare **UNet** with augmentations and **DeepLab** with all backbones (only experiments with **ASPP**). To do that, put these models on three scatter plots. For the first plot, the x-axis is **training time** (in minutes), for the second plot, the x-axis is **inference time** (in milliseconds), and for the third plot, the x-axis is **model size** (in megabytes). The size of each model is printed by PyTorch Lightning. For all plots, the y-axis is the best **mIoU**. To clarify, each of the **4** requested models should be a single dot on each of these plots.

Which models are the most efficient with respect to each metric on the x-axes? For each of the evaluated models, rate its performance using their validation metrics, training and inference time, and model size. Also for each model explain what are its advantages, and how its performance could be improved?

 # Answer:
*** The most efficient in terms of training time: DeepLab with VGG11**
*** The most efficient in terms of inference time: DeepLab with VGG11**
*** The most efficient in terms of model size: DeepLab with MobileNet_v3**

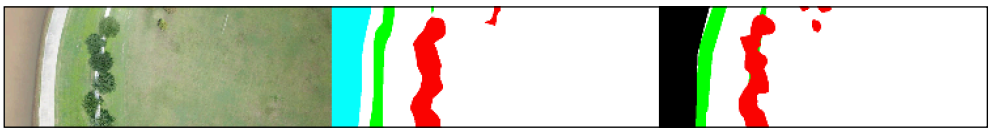


As we can see, the best quality is given by the DL+VGG11 model, since it takes relatively little time to learn and makes predictions quickly, however it is one of the biggest models in terms of parameters. If we want a lightweight yet high-quality model, DL+MobileNet-v3 is what we need. The quality of this model is not much worse than VGG's and it makes quick predictions but takes the longest time to learn. If we want fast learning model, ResNet is perfect. This model has even worse quality and takes 1.5 times longer to make predictions, but it learns quickly.

Task 4.

Pick the best model according to **mIoU** and look at the visualized predictions on the validation set in the TensorBoard. For each segmentation class, find the good examples (if they are available), and the failure cases. Provide the zoomed-in examples and their analysis below. Please do not attach full validation images, only the areas of interest which you should crop manually.

The best model is VGG11.

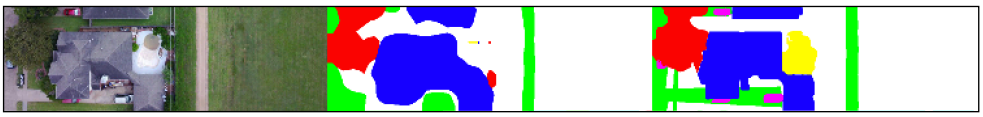


Background (black):
Bad example. The model misclassified the background as water

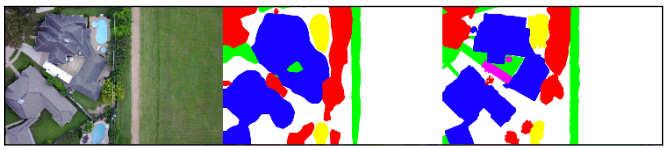
Building (blue):
Bad example. Because of the light roof, the net misclassified it as a road.

Grass (white):
Bad example. Instead of the background, the model predicted grass (but I also rather agree with the model, not with the ground truth)

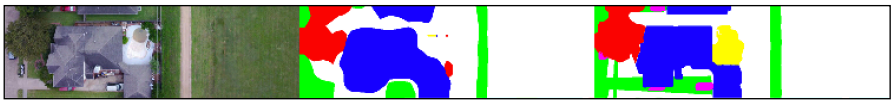
Grass (white):
Good example. Correctly identified grass around buildings and did not even misclassified it as trees.



Pool (yellow):
Bad example. The model did not detect the pool because it is empty



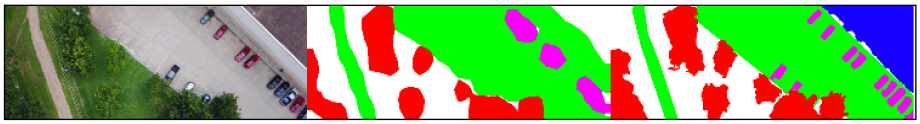
Pool (yellow): Good example. The model very precisely found two small non-empty pools.



Road (green): Good example - the flat, narrow road on the righthand side was identified, but the road around the building was not identified so well (bad example).

Tree (red): Bad example. The model did not separate a tree at the center from the green grass around it.

Tree (red): Good example. The model identified even a tiny bush at the center and did well with big trees on the corners.



Vehicle (pink): Bad example. The model can't separate several cars standing together

Water (light-blue). Bad example. The model predicted water instead of the background, perhaps it recognized it as a muddy river.

Water (light-blue). Good example. Just a good prediction of a wide river.