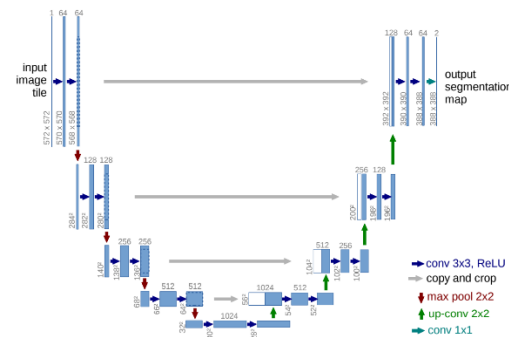
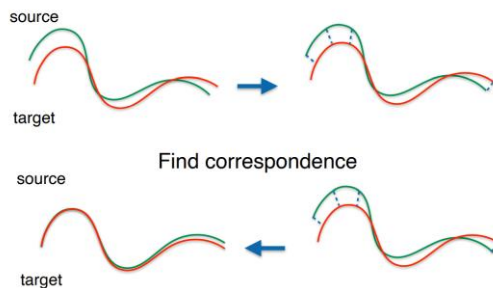


Method Description

- Overview: My method to make pose estimation is an indirection approach, which is a segmentation network + ICP approach. I first started with pointnet architecture for segmentation task. However, it didn't work so well. The object in the image is small which results in very few points sampled from the object even though I center cropped the image and applied preprocessing and normalization techniques. Due to the time constraint for submitting homework, I turned to Unet for segmentation network. I would later figure out why my pointnet is not working well as it should have a relatively good performance.
- Network Architecture: For Unet Segmentation model, the backbone is the encoding process through downsampling and Conv2d layers which extracted image features representation through different stages and at multiple different levels. The segmentation network is Unet. For the second part, it includes upsampling and concatenation followed by regular convolution operations. The upsampling stage can be bilinear interpolation or transposed convolution. Here I applied is the bilinear interpolation.



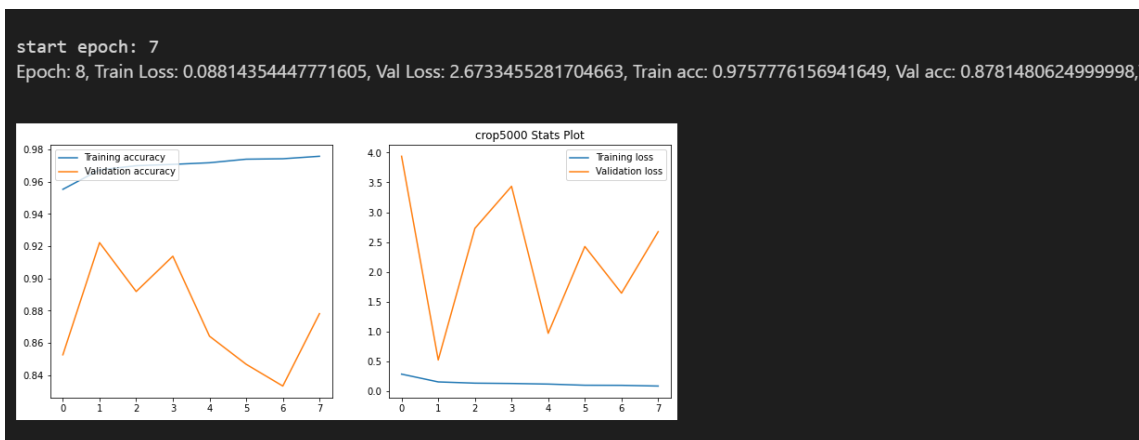
The pose estimation network I applied is Iterative Closest Point approach. Given the assumption that the closest points have a better correspondence, we can iteratively find the correspondence, give an estimation, then find the closest point, and find correspondence again.



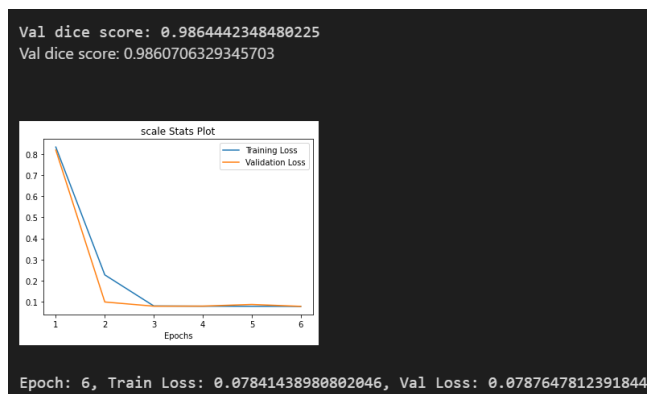
- Losses: For Unet segmentation, the losses I used is cross entropy loss and dice loss which is calculating the intersection of predicted mask and true mask to deal with the unbalanced class object in our case. Instead of using 2 as dice coefficient, I cited the way Unet github officials used to calculate the dice coefficient which considered the number of classes and the shape of input. For icp, I used Umeyama's method to optimize the least-square estimation of rigid transformation through a closed form solution.

Experiments

- Data: The data I used is all the 31795 rgb images in our training set. I preprocessed the image and label by a scale of 0.8 to fit it into my gpu memory. The resize method for image is bicubic and for mask is nearest neighbor interpolation. Then I converted image from 0-255 to 0-1 for neural network training purposes. The postprocessing I did is to resample the predicted label back to its original size 1280 x 720 to match the depth image when I lift to point cloud when doing ICP.
- Training details:
 - learning rate: 1e-5.
 - batch size: 2 (also tried 4 but image scale is low)
 - optimization: RMSprop with weight decay=1e-8 and momentum=0.9
 - number of epochs: 10 (but 5 is actually sufficient)
 - Learning rate scheduler: ReduceLROnPlateau with mode = max and patience=2
- Ablation Study: As I mentioned in the overview, I started with a pointnet segmentation network which didn't work so well. The validation loss has a very big oscillation which indicates the network is generalizing poorly, very sensitive to noise and overfitting badly. In my point of view, this could be the results that sampled object points is low in number. Thus the shape of object is incomplete and inconsistent, making the network hard to distinguish which is the right one.

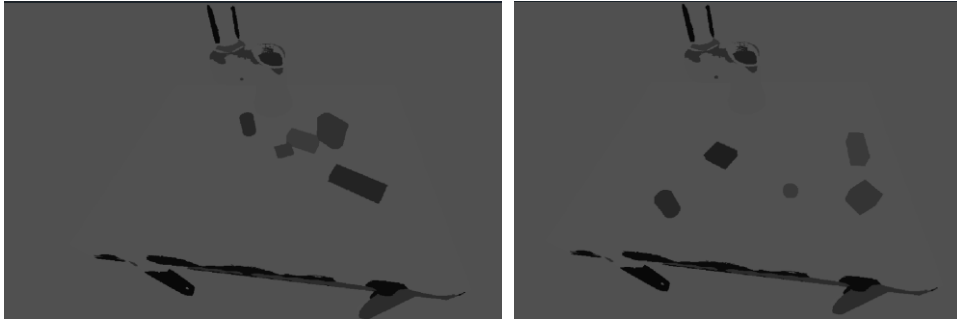


The Unet Segmentation is working well. With parameters described in the training details, the network converged very smoothly.

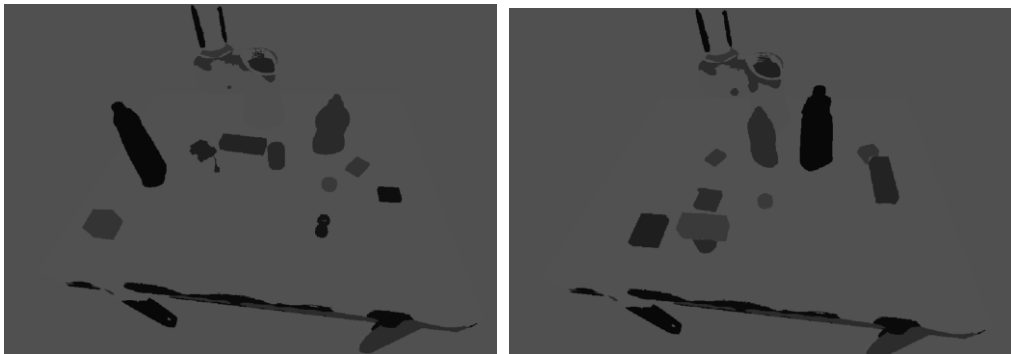


For the segmentation network with a basic ICP approach, the pose_acc_5deg_1cm is 0.505, after I used 20 initial transformation matrix and a two layered ICP, I achieved 0.6316 for pose_acc_5deg_1cm. I later tried using 3 layered ICP and 100 initial transformation matrix which is time computing consuming. However, it only gives 0.629 which is not an improvement. The reason might be that 3 layered is too much and using a very low threshold leading to mismatch between point clouds.

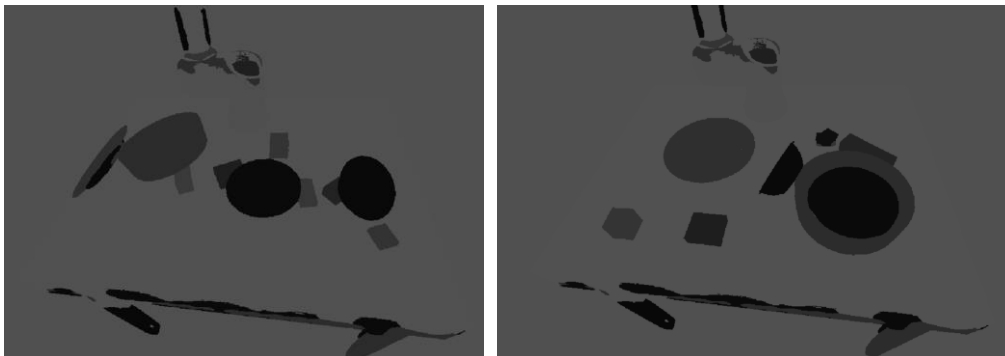
- Visualization:
 - Two Level1 segmentation output examples:



- Two Level2 segmentation output examples:



- Two Level3 segmentation output examples:



- ICP result example (green car):



- Performance:
 - Username: gordonhu
 - Pose_acc_5deg_1cm: 0.6316