6-DoF Pose Estimation

Srinidhi Kalgundi Srinivas
skalgundisrinivas@ucsd.edu
Username: ***sks***

***Abstract*—This report presents the approach taken to obtain poses of objects in the given test set. It includes the detailed explanation of methods used to obtain the segmentation masks and utilization of ICP for pose estimation.**

Submission to leader board is made under the username: sks

I. Methods

The training set and validation set contains RGB images, depth images and the corresponding segmentation labels. Given test images do not contain any segmentation masks which requires us to calculate the segmentation masks before running the ICP. The approach taken to find the pose of the object was similar to K-Nearest Neighbor approach where every image in the test set was compared with all the images in the training set.

A. Data

Table I shows the split that was used for training, validating and testing the segmentation model.

TABLE I: Dataset split

Train	Validation	Test
3964	236	400

B. Semantic Segmentation

Given images only consisted of single instances of the object. This enabled us to use semantic segmentation instead of instance segmentation. U-Net [1] was utilized for the segmentation task. U-Net architecture is as shown in Figure 1.

1) Modifications to U-Net: To train the U-Net model described in the above section, the number of filters had to be reduced. This was due to the fact that the input image size was 720x1280px which was quite large to fit into the GPU. Experiments with smaller image size yielded sub-optimal results as discussed in the experiments section. Filters of size [32, 64, 128, 256] was used to encode the image to a latent space and the decoder was used to bring back the image to its original size. Results of semantic segmentation for a full sized validation image is as shown in Figure 2.

It was observed that the pose accuracy was highly dependent on the accuracy of the semantic segmentation model. This correlation is highlighted in the experiments section.

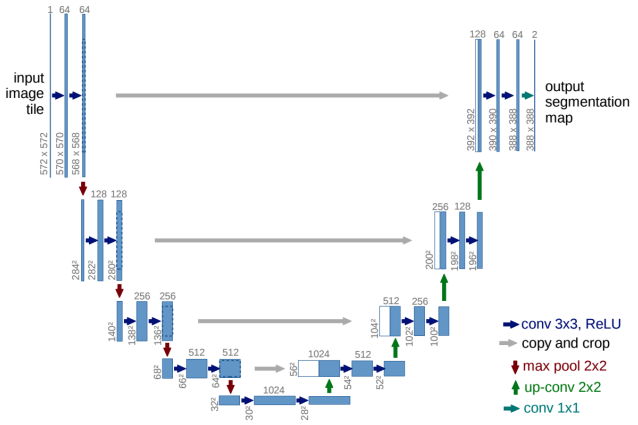


Fig. 1: U-Net Architecture

C. Iterative Closest Point

1) Data Formatting for Pose Estimation: Training data was parsed initially and stored in a .pkl file to allow easy access and to save time. For every image in the training data, point cloud of the image was obtained using camera intrinsics. Using the object mask provided in the training set, objects were segmented out. Point cloud was then brought into the world frame using camera extrinsics and then to canonical frame using **poses_world** information provided in the "meta" file. This information along with **poses_world** was then saved to a pickle file.

2) Initial Pose: During experiments, it was found that the initial pose value for ICP plays a crucial role in obtaining good results. A bad pose initialization causes ICP to reach local minima. To overcome this challenge, the **poses_world** information stored in the pickle was used as the initial pose of the source point cloud for the ICP.

3) Test set: For every image in the test set, the point cloud was first obtained using depth image and camera intrinsics. For all the objects in the image, point cloud was cropped based on the segmentation mask obtained using u-net model. This cropped point cloud was then moved to the world frame using camera extrinsics. This was then compared to all the point cloud of the object in the training set. To achieve good results, initial pose was set to **poses_world** from training set. Point clouds from training set were used as source point clouds and object point cloud was used as target point cloud.

4) ICP Parameters: Open3D library's ICP method was used. Threshold was set **7** to obtain good correspondences

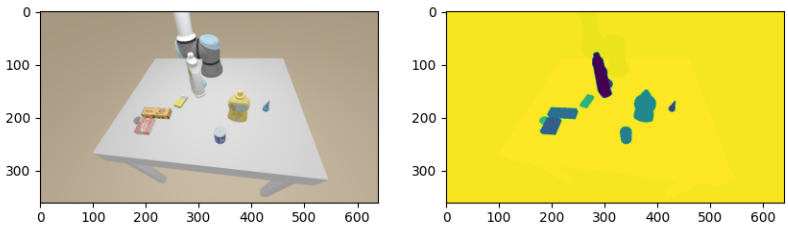


Fig. 2: Output of Semantic Segmentation using U-Net

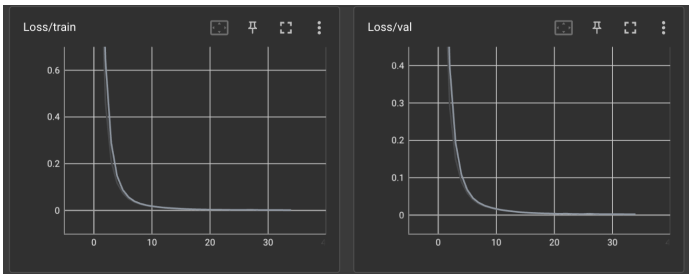


Fig. 3: Training and Validation Loss

using nearest neighborhood. Maximum iterations parameter was set to **50**.

II. Experiments

1) Data Preparation: The semantic segmentation model was trained on different sized images to understand the effect of segmentation on pose accuracy. During training, the image was downsized to 360x640px, 160x320px and masks were generated for the same size. During testing, the masks were resized to 720x1280px using **nearest neighbor interpolation** method which resulted in lack of finer features as shown in Figure 4.

2) Training of Semantic Segmentation Model: U-Net was trained and verified with data split as mentioned in I. The network was trained from scratch for 30 epochs. **Cross entropy** loss function used for comparison of predicted data with ground truth data. To get the best results, **Adam** optimizer was used with an initial learning rate of **1e-3**. Learning rate was reduced based on the the condition of plateau with a patience of 3 epochs conditioned on validation loss.

Training and validation loss is as shown in 3

3) Ablation: To achieve good accuracy with segmentation, different segmentation model such as SegNet[2] was utilized. It was observed that the segmentation results with SegNet was sub par and the final estimated pose was worse than it was for U-Net as shown in Table II.

4) Results: Table III shows the pose accuracy obtained based on the size of the image input to the segmentation model.

TABLE II: Segmentation using U-Net and SegNet

Model	5 degree_1cm
U-Net	**0.918**
SegNet	0.206

TABLE III: Test Set Results - Pose Accuracy

Image Size (px)	5 degree_1cm
720x1280	**0.918**
360x640	0.67
160x320	0.406

TABLE IV: Test Set Results - Pose Error

Image Size (px)	rre	rre_symmetry	rte
720x1280	**1.089**	**0.45**	**0.00044**
360x640	87.48	2.29	0.0028
160x320	97.72	6.05	0.0045

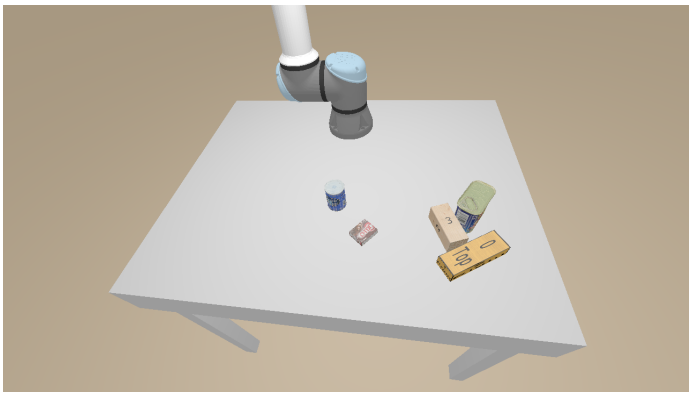
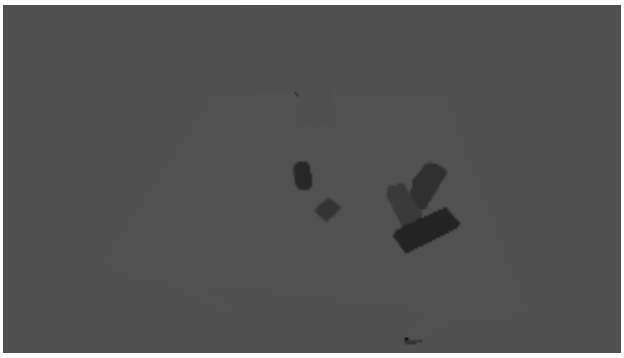


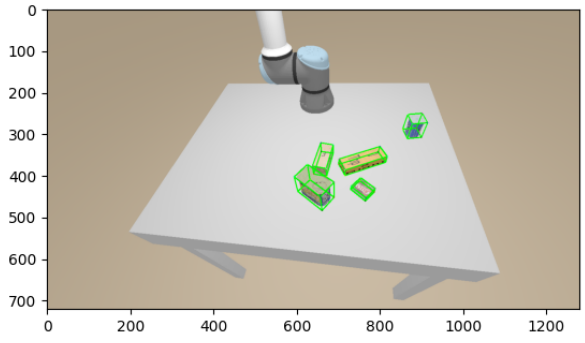
Fig. 4: Input Image

(a) Segmentation mask for input size 360x640px

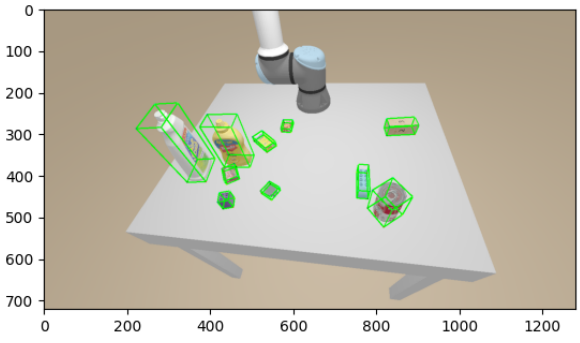


(b) Segmentation mask for input size 160x320px

Fig. 5: Segmentation Masks for different input sizes

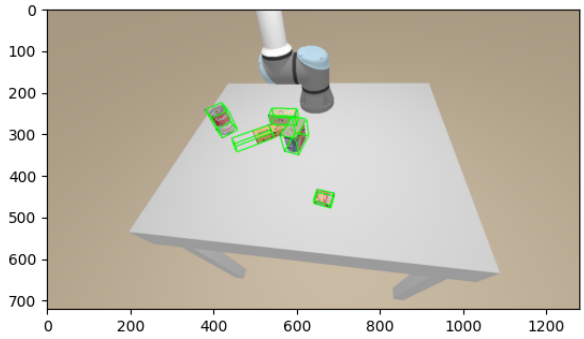


(a) Final bounding box for Test Image 1-2-11

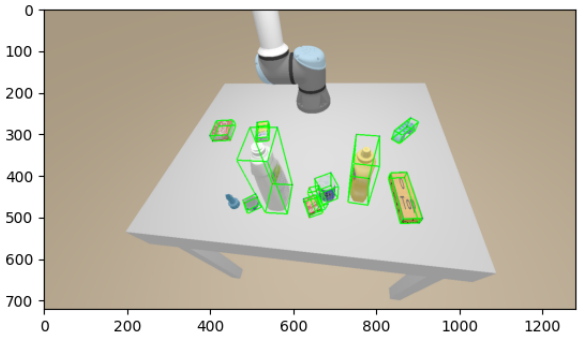


(b) Final bounding box for Test Image 2-89-11

Fig. 6: Bounding Box Output

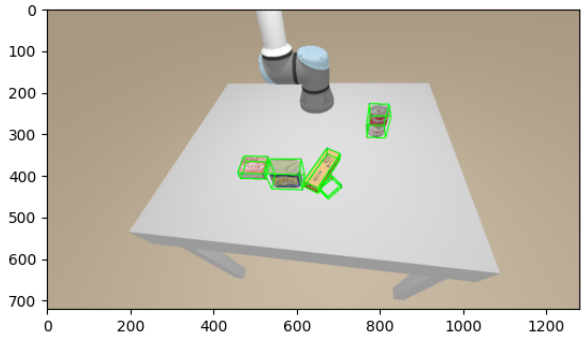


(a) Final bounding box for Test Image 1-97-10

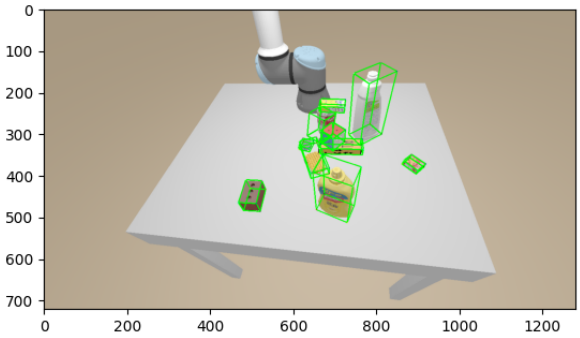


(b) Final bounding box for Test Image 2-5-10

Fig. 7: Bounding Box Output - Failure



(a) Problems with Occlusion - 1-57-11



(b) Problems with Occlusion - 2-60-11

Fig. 8: Bounding Box Output - Failure

REFERENCES

[1] Ronneberger, O., Fischer, P. and Brox, T., 2015, October. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.

[2] Badrinarayanan, V., Kendall, A. and Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence, 39(12), pp.2481-2495.