TrimBot2020 Deliverable D4.2

Recognition and localization of Garden Proto-Objects

Work in progress

Principal Author: University of Amsterdam (UvA)
Contributors:
Dissemination: RE

Abstract. The aim of Deliverable 4.2 is to identify the objects of interest being seen by the robot's cameras. This includes semantic segmentation of 2D images captured by the robot and, based on the estimated depth, reconstructing the semantic 3D point cloud.

Deliverable due: Month 36

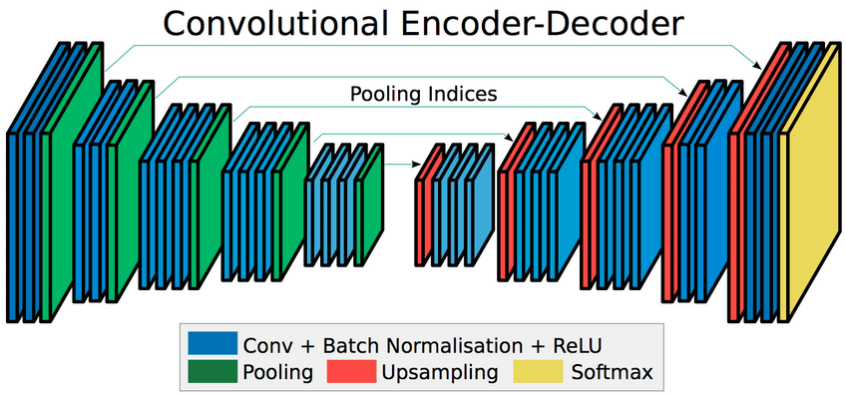

Figure 1: SegNet architecture

1 Semantic Segmentation

Semantic segmentation partitions a given image into different semantically meaningful regions. This involves labelling each pixel in the image with the name of the object depicted by that pixel. This is challenging because algorithms need not only to recognize the objects in the scene but also localize them at a very precise level.

Several approaches have been proposed to tackle the problem, but fall short to generalize especially for challenging conditions that the project features: outdoor lighting conditions and confusing object textures. We tackle the problem with a data-driven approach by training a deep convolutional neural network to recognize different garden objects and identify them in an image.

We use the SegNet architecture [1] as shown in Figure 1 that receives an RGB image as input and return the label image of the same size as output. The module is built upon the Caffe deep learning framework [2]. The package is provided in the TrimBot2020 gitlab repository, split into multiple ROS nodes:

- `uva_segnet`, `uva_segnet_view`: SegNet module for 2D image segmentation

SegNet module takes images from the left camera as input and outputs semantic segmentation. The segmentation definition follows the convention provided by TrimBot2020 workshop challenge 3DRMS, in particular

0 Unknown

1 Grass

2 Ground

3 Pavement

Figure 2: Sample images from synthetic dataset (top row) and real images by TrimBot2020 (bottom row)

4 Hedge

5 Topiary

6 Rose

7 Obstacle

8 Tree

9 Background

Deep learning networks require a large amount of training data to generalize well. Due to the high cost of semantic segmentation labelling, we design several synthetic garden models with similar semantic labels with one required in the project to accommodate the network with preliminary training data. Figure 2 shows a few sample images from the synthetic and real dataset. Notice the difference not only in the appearance of the bushes and plants but also in the whole image color. We explore the impact of such difference to the performance of the network by training on synthetic image and testing on real images with and without fine-tuning.

It is clearly shown in Figure 3 that fine-tuning the network on real data helps it cope better with the changes in contexts and thus perform better in real images. The quantitative results are shown in Figure 4. In general, the network achieves 90% accuracy with test garden images reported in TrimBot2020 workshop challenge at ECCV 2018.

2 Semantic point cloud construction

The label output from the semantic segmentation module can be used to reconstructed semantic point clouds of the scene using depth image data from FPGA stereo module. The result point

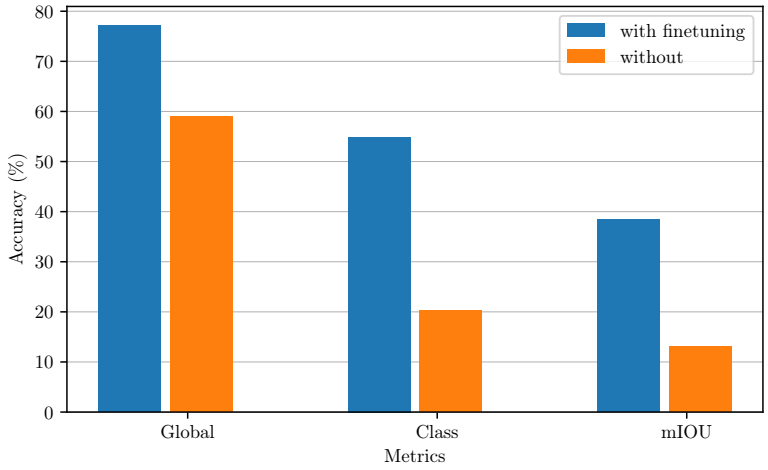

Figure 3: Validation results from network trained on synthetic data with and without fine-tuning.

clouds of the sample images are shown in the first 2 columns of Figure 5. Despite a few mistake between topiary and hedge (depicted in cyan and yellow), the results show good delineation of objects, namely less flying pixels on the ground, which appear sometimes in the ground truth point clouds.

The package includes the following ROS nodes:

- `uva_pc`: point cloud reconstruction

3 Object localization

The garden objects are extracted from the semantic point cloud by cluster analysis and extraction. We use an octree data structure to subdivide the 3D grid Euclidean space to separate different objects in each semantic cluster resulted from the previous step.

Clusters of different classes are treated separately before being merged and having non-max suppression based on their location and overlapping regions. To this state, we provide object localization in the form of bounding boxes that enclose each object's point set. Each bounding box composes of a `centroid`, a point in 3D coordinates, with `size` describing the 3 dimensions of the box, namely `height`, `width`, and `depth` along 3 coordinate axes, and a `label` that indicates the type of enclosed object. A few examples of the real garden are shown in Figure 5. The results shown here are preliminary and a full evaluation is in progress.

The package includes the following ROS nodes:

- `uva_loc`: object localization

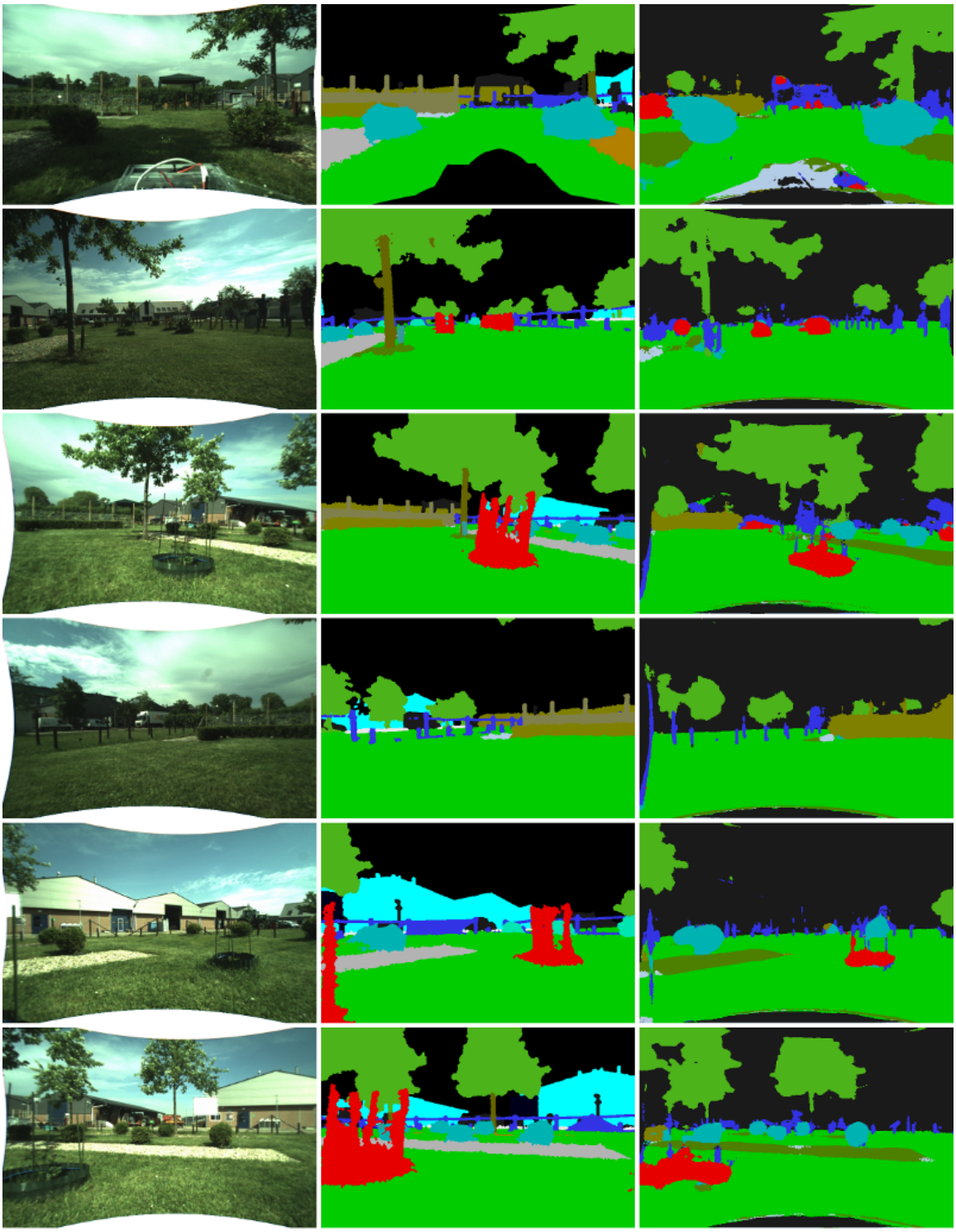

Figure 4: Quantitative segmentation results on TrimBot2020 test images. From left to right: input RGB image, ground truth, predicted segmentation.

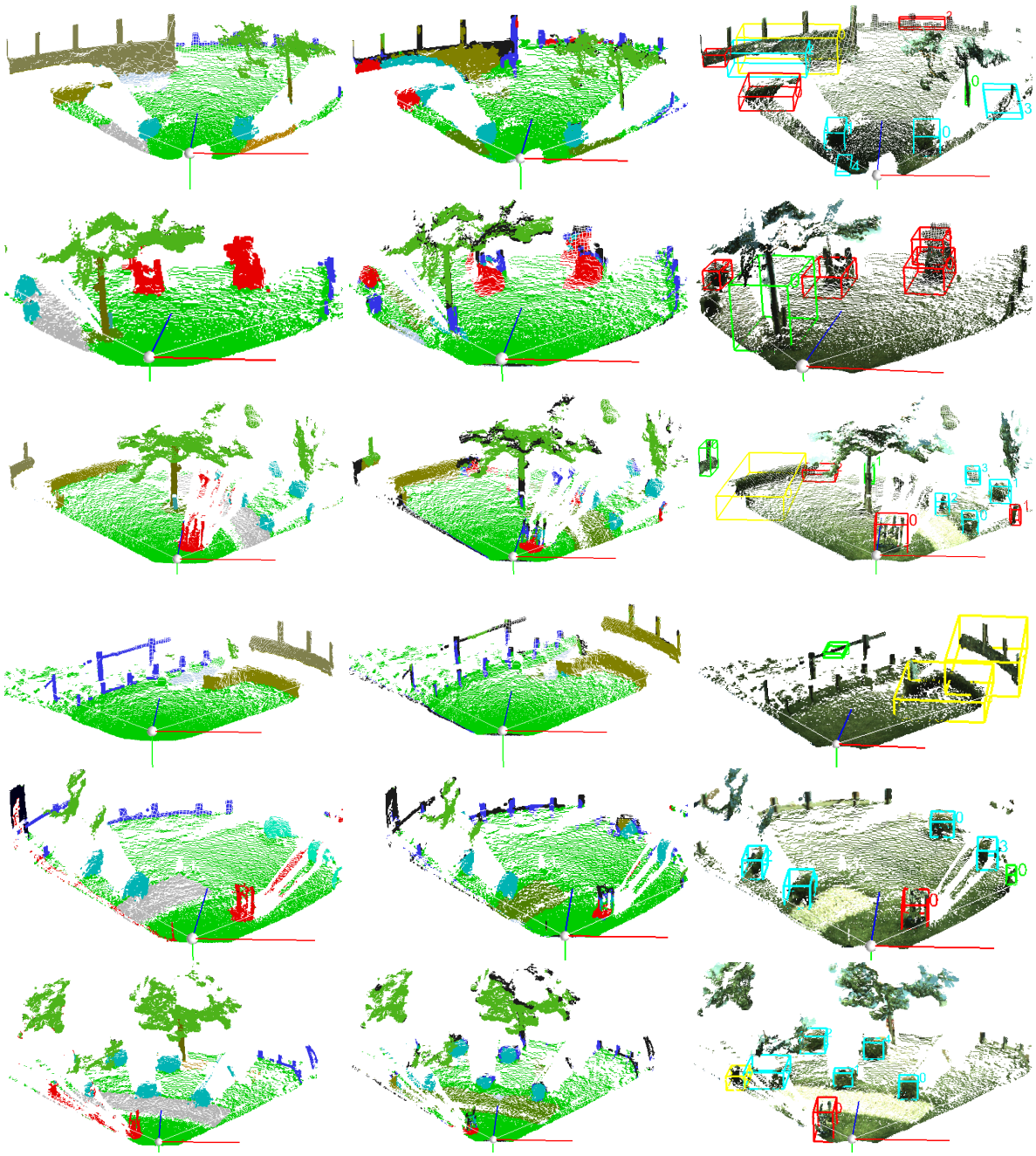

Figure 5: Semantic point cloud reconstruction and garden objects localization. From left to right: point cloud with ground truth label, predicted label, and object localization.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[2] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

