Launch file parameter descriptions:

* segmentation\_node
  + neural\_net\_graph – must specify path to the weights file (.hdf5 file)
  + skip\_input\_img\_freq – set to a value high enough to make the predictions around 1 Hz if you are using octomap cropping, otherwise high frequency input images will not follow the robot position. For example, if you set this parameter to 1, the robot will fall beyond the cropped octomap, if you set this parameter to 25-30, the robot will stay in the center of the cropped cube
  + enable\_colored\_labeling – will publish RGB predictions to a separate topic. NOTE: when set to True, slows down considerably
* labeled\_pc\_map\_node
  + labels\_to\_ignore\_in\_map – change this value according to the label numbers given in “label\_color.png” or “label\_color.xlsx” (this parameter ignores selected labels in the octomap)
  + target\_labels\_in\_map – same logic
  + bounding\_box\_size – single value for setting the bounding cube’s dimensions
  + octomap\_cropping\_enabled – if true, octomap is cropped according to the bounding\_box\_size variable, otherwise it’s an ever growing octomap

For running the new network, you need to install the required packages. The easiest way is to do the following:

conda create --name tf tensorflow-gpu

conda activate tf

pip install tqdm opencv-python matplotlib scikit-learn segmentation-models

And then provide the path to the python interpreter (can be found with the “which python” command) in the “$(find SegmentationMapping)/scripts/segmentation\_node” file.