Point Cloud Grasp Classification for Robot Grasping

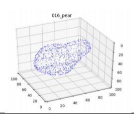
Navjot Singh, Zachary Blum, Neethu Renjith
navjot@stanford.edu, zblum25@stanford.edu, neethur@stanford.edu

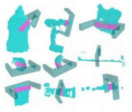
Executive Summary

This work aims to improve the grasping capabilities of robotic arms by using a neural network that classifies a potential grasp of an arbitrary object as a good grasp or not, given a direct point cloud of the grasp. Inspired by the recent successes of new architectures [1], [2], [3] that are able to classify objects directly through their point clouds, this project aims to leverage the PointNetGPD [4] pipeline and compare the performance of these new architectures to classify point clouds of grasps.

Data and Features

- YCB Dataset of 59 everyday objects represented through 3D scans and RGBD pictures
- Point cloud of object created through RGBD scans
- Given a mesh file from 3D scans, random antipodal grasps are sampled for each object and a numerical score is provided based on two metrics and combined through: $Q(\mathbf{s}, \mathbf{g}) = \alpha Q_{fc}(\mathbf{s}, \mathbf{g}) + \beta Q_{gws}(\mathbf{s}, \mathbf{g})$
- Given projection area of a robot gripper, each grasp is projected onto the object point cloud to create the grasp point cloud
- Grasps with scores above 0.6 are labeled 1 and 0 otherwise
- **Data Augmentation:** Each grasp point cloud was randomly rotated in addition to applying Gaussian noise



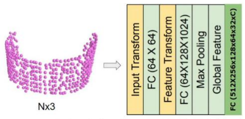


References

[1] https://arxiv.org/abs/1706.02413
[2] https://arxiv.org/abs/1704.01222
[3] https://arxiv.org/abs/1801.07829
[4] https://arxiv.org/abs/1809.06267

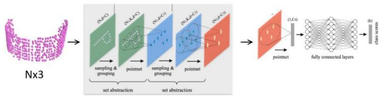
Models

PointNet Deeper



Additional FC layers were added to the original PointNet Architecture in order to reduce bias.

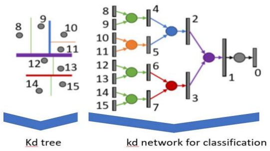
PointNet++



PointNet++ applies the PointNet architecture recursively at different scales to bring in localized features.

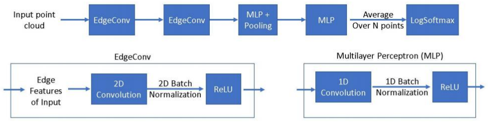
KD-Networks

Uses kd-trees constructed by recursively picking the axis with largest range of point coordinates and splitting the set of points into equally-sized subsets.



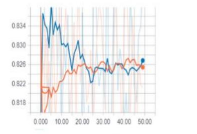
DGCNN

Edge convolution layers within the network dynamically analyze the neighbors of a point in the point cloud and performs convolution on the edges of the associated local graph.

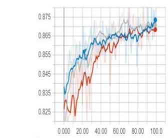


Results

Example Test Accuracy and Loss Graphs (for KD-Networks):



Test accuracy with epoch
(Orange: depth=11, Blue: depth=15)



Test accuracies for PointNet-based architectures against epochs
(red: PointNet++, blue: PointNet, gray: PointNetDeeper)

Results for All Models:

Model	Test Accuracy (%)	Training Accuracy (%)
PointNet	87.35	88.02
PointNetDeeper	87.09	87.43
PointNet++	86.85	86.87
KD Net (depth = 11)	82.13	83.27
KD Net (depth = 15)	84.30	82.58
DGCNN (LR = 0.001)	85.42	85.06
DGCNN (LR = 0.01)	82.42	82.23
DGCNN Deeper (LR = 0.001)	85.87	86.09

Discussion & Future Work

- Baseline PointNet performs the best among all the architectures
- The accuracies of the different models are comparable and could be improved through hyperparameter tuning
- A more end-to-end generator framework can be produced that selects optimal grasp regions of an object's point cloud rather than passively classifying sampled grasps