2021

Deep Learning for Robotic Grasp Detection

CS39440 – PROJECT OVERVIEW

OLLIE THOMAS (OLT13)

Supervisor: Patricia Shaw (phs)

Degree Scheme: Robotics and Embedded Systems Engineering MEng (132C)

Date: 07th February 2021

Version 1.0 (Release)

Project Description

This project aims to look at the challenging task of robotic grasp detection. Typically, robotic manipulators are programmed with a database of objects that could be picked up, each annotated with physical information and more importantly grasp locations. However, this does not lead to good grasps of unknown objects.

This project will apply a deep learning approach to this problem, much like that of [1, 2, 3]; using a deep convolutional neural network (DCNN) with inputs from a depth camera to 'learn' successful grasps for objects. This trained model will then be applied in simulation – using ROS [4] and a simulator such as Gazebo [5] – to evaluate the efficacy on unseen objects. The project has three main constituent parts: building and training the DCNN model, the simulation of the robotic manipulator, and integrating the model to control the grasps.

The first part of the project; building the model. The DCNN will be built by utilising the Tensorflow [6] and Keras framework. These libraries provide an interface to create efficient neural networks within the Python [7] programming language. The project will involve researching the best way to build the model i.e., the order and number of layers (likely a mixture of 2D convolution and dense layers). This will also involve building prototype networks to learn how to best design and tune the network for this purpose. The DCNN will be trained on a dataset consisting of depth images of objects in random orientations annotated with target grasps. The data will be split into training, validation and testing datasets; meaning the model can be evaluated before using it in the simulation. This can be done by comparing the predictions on the test set to their annotated grasp using a metric such as mean squared error. The current dataset that is being reviewed for the project is the ACRONYM [8, 9] grasp dataset, which uses the ShapeNetSem [10] model database. Another option could be to generate my own grasp dataset using the model database and simulation software such as GraspIt! [11].

The second part of the project; simulating the manipulator. The robotic manipulator being used in this project is the Franka Panda Arm. [12] This arm is used due to its ease, compatibility with the MoveIt [13] framework, and the ACRONYM dataset comes with a reference frame to the Franka Panda Gripper. The project will use ROS and Gazebo to simulate the arm moving to specified grip poses. This can be done initially without the DCNN model, in order to get the other parts of the simulation working first. For example, the setup of the depth camera with a reference transform to the panda arm. The simulation will be programmed with C++ – a common language used in robotics.

Finally, the trained DCNN model will be exported and integrated into the simulation to detect and employ grasps using a real-time image from the depth camera. This will be deployed as a separate node in the ROS network, likely using Python for compatibility with Tensorflow.

The project will follow a modified form of the Feature-Driven Development (FDD) methodology to manage the necessary tasks. This is an iterative plan-driven approach.

Proposed Tasks

The following tasks will be completed for this project.

- **Finding and setting up a suitable dataset.** For this project it is necessary to have a large dataset to train the deep network. There are two ways this can be located:
    - **Using a pre-made dataset.** There are many grasp datasets that are available, for example the ACRONYM, Jacquard or Cornell datasets. These may come with specific license agreements for their use.
    - **Creating a dataset for the project.** If the above datasets are not suitable or are difficult to work with, a new dataset could be made using model database and a simulator such as GraspIt!.
- **Setting up a ROS workspace and Python environment with a version control system.** The project will be backed-up and version-controlled using GitHub [14]. The necessary libraries will need to be installed ROS and Tensorflow to have full functionality.
- **Development.** There are three main tasks for development:
    - **Researching and building the network model.** Learning to use the Tensorflow framework to build a DCNN will include building prototype models.
    - **Simulating an arm to grasp at a certain pose.** Using the MoveIt motion planning framework to move the franka panda arm to a specific grasp pose.
    - **Integrating the trained model into the simulation.** Learning how to export a trained Tensorflow model for integration into the ROS network.
- **Project diary.** The project involves many tasks and meetings; keeping a diary will help me to remember what happens when during the project and keep track of ongoing tasks.
- **Preparation for demonstrations.** I will need to organise and prepare the project work for both the mid-project and final demonstrations.

Project Deliverables

The following are the expected project deliverables.

- **Model Dataset** – A dataset of depth images of models in varying positions, annotated with the best grasp pose for the object. This dataset will be used to train the DCNN model. This dataset may not
- **Neural Network Model** – A trained DCNN model (on the above dataset) that will be able to predict a grasp pose based on an input depth camera image of the object to be grasped.
- **Simulation Software** – A simulation of a robot that will apply the trained neural network model to detect grasp poses of unseen objects. This simulation will be used to evaluate the performance of the model.
- **Documentation** – This document will describe how to interface with the dataset and the neural network. This will be included as part of the appendix in the final report.
- **Final Report** – This document will discuss the completed project work, including associated appendices, and acknowledgement of third-party sources and libraries.
- **Demonstrations** – A mid-project demonstration to demonstrate the advancement of the project and a final demonstration to demonstrate the final product at the end of the project process.

Annotated Bibliography

[1] P. Schmidt, N. Vahrenkamp, M. Wächter, and T. Asfour, ‘Grasping of Unknown Objects Using Deep Convolutional Neural Networks Based on Depth Images’, in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 6831–6838, doi: 10.1109/ICRA.2018.8463204.

[2] J. Watson, J. Hughes, and F. Iida, ‘Real-World, Real-Time Robotic Grasping with Convolutional Neural Networks’, Jul. 2017, pp. 617–626, doi: 10.1007/978-3-319-64107-2_50.

[3] S. Kumra and C. Kanan, ‘Robotic Grasp Detection using Deep Convolutional Neural Networks’, *arXiv:1611.08036 [cs]*, Jul. 2017, Accessed: Feb. 04, 2021. [Online]. Available: http://arxiv.org/abs/1611.08036.

[1, 2, 3] are examples of deep learning being applied to this topic and provide a good understanding of the processes involved in a project such as this.

[4] ‘ROS.org | Powering the world’s robots’. https://www.ros.org/ (accessed Feb. 06, 2021).

[5] ‘Gazebo’. http://gazebosim.org/ (accessed Feb. 07, 2021).

ROS and Gazebo will be used to simulate the manipulator and DCNN system working together to grasp objects.

[6] ‘TensorFlow’, https://www.tensorflow.org/ (accessed Feb. 04, 2021).

[7] ‘Python’, https://www.python.org/ (accessed Feb. 07, 2021).

Python will be used in a conjunction with Tensorflow to build and train the DCNN model.

[8] C. Eppner, A. Mousavian, and D. Fox, ‘ACRONYM: A Large-Scale Grasp Dataset Based on Simulation’, *arXiv:2011.09584 [cs]*, Nov. 2020, Accessed: Feb. 04, 2021. [Online]. Available: http://arxiv.org/abs/2011.09584.

[9] ‘ACRONYM: A Large-Scale Grasp Dataset Based on Simulation’. [Online] Available: https://sites.google.com/nvidia.com/graspdataset (accessed Feb. 04, 2021).

[9, 10] are references to the ACRONYM dataset and accompanying paper; this is the likely dataset that this project will use.

[10] ‘ShapeNet’. https://www.shapenet.org/ (accessed Feb. 07, 2021).

A large model database that is used by the ACRONYM dataset but could also be used for generating a project specific dataset using GraspIt!.

[11] ‘GraspIt!’ https://graspit-simulator.github.io/ (accessed Feb. 04, 2021).

A grasp simulator that could be used to generate the dataset for training.

[12] ‘Franka Emika’. https://www.franka.de/technology (accessed Feb. 07, 2021).

The robot arm and gripper planned for this project.

[13] ‘MoveIt Motion Planning Framework’. https://moveit.ros.org/ (accessed Feb. 07, 2021).

[14] ‘GitHub’, *GitHub*. [Online] Available: https://github.com (accessed Feb. 04, 2021).

A free private repository for version control and backups.