CHALMERS
UNIVERSITY OF TECHNOLOGY

Experimental Analysis for Building Robust Grasp Planning Strategies for Industrial Objects

Master's thesis in System, Control and Mechatronics

AISHWARYA KANCHAN and YIYUN XIA

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023
www.chalmers.se

MASTER'S THESIS 2023

Experimental Analysis for Building Robust Grasp Planning Strategies for Industrial Objects

AISHWARYA KANCHAN

YIYUN XIA

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023

Experimental Analysis for Building Robust Grasp Planning Strategies for Industrial Objects
AISHWARYA KANCHAN and YIYUN XIA

Industrial supervisor: Atieh Hanna, Volvo GTO
Academic supervisor: Yasemin Bekiroglu Department of Electrical Engineering, Chalmers University of Technology
Academic co-supervisor: Ahmet Ercan Tekden, Department of Electrical Engineering, Chalmers University of Technology
Examiner: Yasemin Bekiroglu, Department of Electrical Engineering, Chalmers University of Technology

Master's Thesis 2023
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: A successful grasp of a singulated Volvo object

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2023 caption

Experimental Analysis for Building Robust Grasp Planning Strategies for Industrial Objects
Aishwarya Kanchan and Yiyun Xia
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Collaborative robots, designed to enhance efficiency and ensure safety in industrial settings, have become increasingly prevalent. In this domain of automation, bin picking plays a crucial role as an automated process involving the extraction of objects from containers and their transfer to designated locations. This technology stands as a key enabler across various industries. This thesis focuses on industrial bin-picking applications and presents analysis of grasping strategies given real industrial objects.

Bin-picking applications face challenges such as accurate prediction and execution of stable grasps, which involve main steps such as perception, grasp and motion planning and execution. In this context, we use an MLP model trained on singulated industrial objects, incorporating relative UR10 hand position and orientation as input features. This choice leads to results in an MLP model that accurately classifies grasps.

The UR10 robot is employed for the experimental execution of industrial objects from Volvo. The grasp pose detector is utilized for this purpose, and it is observed that its performance is suboptimal for the generated data. In this thesis, multiple multi-layer perceptron (MLP) models are constructed, incorporating variations in weights and layers to predict the quality of the grasp. Furthermore, simulation experiments are conducted to assess the selected MLPs, and their performance is compared with that of the UR10 in the laboratory setting.

Keywords: Robot Operating System (ROS), Bin-picking, Universal Robots 10 (UR10), Grasp Synthesis for Industrial Objects, Multi-layer Perceptron (MLP).

Acknowledgements

We would like to express our sincere gratitude and appreciation to the individuals who have contributed to the completion of this master thesis.

We would like to extend our thanks to our academic supervisor, Dr. Yasemin Bekiroglu, for her guidance and expertise. Her insightful feedback and dedication have been instrumental in shaping the direction and quality of this work. We would also like to express our gratitude to Ahmet Tekden, who co-supervised and supported our work, for his insights and discussions.

Additionally, we would like to express our deepest gratitude to Atieh Hanna, our supervisor from Volvo GTO. Her expertise, encouragement, guidance, support, and collaboration have been truly appreciated and allowed us to push boundaries.

We are also thankful to the faculty members, researchers, and staff at Chalmers University of Technology for providing an intellectually stimulating environment and the necessary resources for conducting this research. Their support and dedication to academic excellence have greatly contributed to our learning and growth.

Lastly, we would like to express our gratitude to our family, friends, and loved ones for their unwavering support, encouragement, and understanding throughout this journey. Their love, belief in us, and continuous motivation have been the pillars of our success. We are sincerely grateful to all those who have played a part in this thesis. Their contributions have made this research endeavor both rewarding and fulfilling.

Aishwarya Kanchan, Gothenburg, June, 2023
Yiyun Xia, Gothenburg, June, 2023

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

AGV	Automated Guided Vehicle
DOF	Degree of Freedom
GPD	Grasp Pose Detector
GPG	Grasp Pose Generator
MLP	Multi Layer Perceptron
ROS	Robot Operating System
UR10	Universal Robot 10

Contents

List of Acronyms **ix**

List of Figures **xiii**

List of Tables **xv**

1 Introduction **1**
1.1 Industrial background 1
1.2 Challenges 2
1.3 Literature review/Related work 3
1.4 Contribution and limitations 5

2 Background **7**
2.1 Grippers and UR10 robot 7
2.1.1 Universal Robot 10 7
2.1.1.1 Trajectory planning using Inverse Kinematics (IK) solver 8
2.1.2 Two finger gripper 9
2.2 RealSense RGB-D camera 9
2.2.1 Calibration of camera on UR10 and point cloud Generation . 10
2.3 Grasps 12
2.3.1 Visualization of the datatset 12
2.3.2 Preprocessing Point clouds 13
2.4 Robot Operating System (ROS) 14
2.5 Industrial objects by Volvo Group 15
2.6 PyBullet - simulation tool for grasp executions 17
2.7 RVIZ visualization 17

3 Grasp Pose Detector **19**
3.1 Grasp Pose Detection(GPD) baseline 19
3.1.1 Execution of baseline grasps in PyBullet simulation and real UR10 robot 20

4 Proposed method and model for Grasp Classification **23**
4.1 System overview 23
4.2 Model architecture - MLP 23
4.3 Dataset generation 25

4.3.1 Representation of grasps in the MLP dataset 25
4.3.2 Plotting grasps on Point clouds 25
4.4 Training of multilayer perceptron 26
4.5 Testing and inference of the MLP models 26
4.5.1 Collection of grasp poses for testing dataset 27

5 Results 29
5.1 Execution of GPD grasps in PyBullet and using a real UR10 robot . 29
5.2 Training and testing accuracy of the MLP models 30
5.2.1 Accuracy of MLP models during training 30
5.2.2 Testing the classification accuracy of MLP models 31

6 Discussion 35
6.1 Implementation of GPD on industrial objects 35
6.2 Analysis of the grasping points of the industrial objects 35

7 Conclusion and Future scope 37
7.1 Future scope 37

A Appendix 1 I

B Appendix 2 III
B.1 Replacement of the RG2 gripper with RG6 III
B.1.1 Mathematical verification III
B.1.2 Physical verification IV

List of Figures

1.1 Demo Station at Volvo's Lab (image provided by Volvo) 2

2.1 UR10 robot. Image source: Universal Robots. 8
2.2 Left to right: vacuum grippers [1], magnetic [2] and multi-finger grippers [3] [4] 9
2.3 The physical dimensions and the area enclosed by the fingers of the RG2-FT Gripper as provided by the datasheet. 10
2.4 RG6 gripper from the datasheet [5] 11
2.5 Visualization of an industrial object on Rviz. 11
2.6 An illustration of an antipodal grasp. It is inspired from the explanations provided in [6] 12
2.7 Illustration of the training data - a subset of the recorded grasp poses for each object. The grasps are depicted in red and green which signify unsuccessful and successful respectively. 13
2.8 Illustration of the test grasps (grasps produced for MLP test dataset) 14
2.9 Point cloud generated from PyBullet simulation. The figure displays the scene in the form of point clouds and the Volvo industrial object is at the center, inside the red circle. 15
2.10 The figure depicts the result of applying RANSAC to remove the table from the pointcloud shown in Figure 2.9 16
2.11 Objects provided by Volvo for grasp learning 17
2.12 The grasps by GPD being executed in the PyBullet environment. . . 18

3.1 A successful grasp of the Volvo industrial object (c) PRT _1_0 (from Figure 2.11) by GPD on PyBullet 19
3.2 Co-ordinate frame of GPD [7] 20
3.3 Left: Grasp proposed by GPD. The arrow indicates the direction of the arm while trying to grip the object; Right : Execution of GPD grasp in PyBullet simulation 21
3.4 The GPD grasps that are scored. Red color indicates grasps classified by the model as unsuccessful while green signifies successful grasps. . 21

4.1 Overview of the system 23
4.2 MLP layers where HL stands for Hidden Layer 24
4.3 GPD would classify this grasp as successful but it fails in execution because of physical constraints. 26

5.1 A successful grasp by GPD . . . 30
5.2 A grasp generated by GPD. This grasp is predicted to be successful. . 31
5.3 Grasps proposed by the baseline GPD . . . 32
5.4 The accuracy score and standard deviation values for MLP trained only using position and orientation values of a grasp. The training datasets include an average of 12 grasps (both successful and unsuccessful) per object. . . . 32
5.5 The distribution of the grasp positions of training and testing data for each object . . . 33

6.1 Examples of stable grasps in the training dataset . . . 36

A.1 Grasps before removing the table . . . I
A.2 Part 1_0 grasps from the manual dataset. From left to right, good grasp, good grasp and bad grasp . . . I

B.1 Left: RG2 gripper range. Right: RG6 gripper range . . . III
B.2 A translation of 0.04 in z direction yields the same grasp pose and outcome as that of RG2 . . . V
B.3 A translation of 0.04 in z direction yields the same grasp pose and outcome as that of RG2 . . . VI
B.4 A translation of 0.04 in z direction yields the same grasp pose and outcome as that of RG2 . . . VII
B.5 A translation of 0.04 in z direction yields the same grasp pose and outcome as that of RG2 . . . VIII
B.6 A translation of 0.04 in y direction yields the same grasp pose and outcome as that of RG2 . . . VIII
B.7 A translation of 0.04 in x direction yields the same grasp pose and outcome as that of RG2 . . . IX

List of Tables

5.1 5 successful grasps from GPD are executed on the real UR10. The table records how many out of the 5 are executed successfully. 29
5.2 Verification of our MLP model accuracy on the real UR10 robot . . . 32

B.1 Abbreviations used in the formula IV

1
Introduction

In the past decade, the field of automation and robotics in industrial settings has experienced significant advancements. Among these applications, the field of bin picking stands out as an area that holds immense potential but requires further enhancements to fully unlock its value. While identifying and picking up objects may be a relatively simple task for humans, it poses challenges when performed by robots.

In contrast, robots need to perceive their environments [8], namely acquire a representation of objects to construct robust grasps. This involves tasks such as perceiving the object's shape, estimating its pose, and planning appropriate grasping strategies [9] [10]. These tasks require sophisticated algorithms and sensor technologies to enable robots to perceive and interact with objects in a manner that is both effective and efficient [11], [12], [13], [14].

1.1 Industrial background

The Volvo Group is a prominent global manufacturer of sustainable transport solutions, renowned for its commitment to automation and continuous improvement in product development and production methods. With a strong focus on innovation and sustainability, the Volvo Group has established itself as a leader in the automotive and transport industry. The company's dedication to automation is evident in its adoption of cutting-edge technologies and robotics to enhance efficiency and precision across its manufacturing processes. By leveraging automation, the Volvo Group aims to optimize production workflows, minimize errors, and ensure consistent quality in its products.

The construction process of Volvo Trucks encompasses several essential steps, and one crucial phase involves the assembly of industrial parts for the trucks. These parts are made available at a kitting station, where they need to be carefully selected and placed onto an Automated Guided Vehicle (AGV). An AGV is a collaborative robot that plays a vital role in material handling and movement within warehouses and industrial settings.

Currently, the responsibility of picking out the required materials from the kitting station and placing them onto the AGV lies with the human operator. This assembly task can be repetitive and monotonous, which presents an excellent opportunity for introducing automation to enhance overall efficiency and productivity.

By implementing automation in the assembly process, Volvo Trucks aims to streamline operations and reduce the reliance on manual labor for mundane tasks. Introducing robotic automation, such as robotic arm systems or other smart robotic solutions, can significantly improve the speed and accuracy of material selection and placement onto the AGV. This not only frees up human operators from repetitive tasks but also allows them to focus on more complex and value-adding aspects of the assembly process. As seen in Figure 1.1, the Volvo lab has a demo station that is equipped with a robot arm and gripper along with a high-power camera. The blue boxes in the image are containers that hold the required industrial objects.

Incorporating automation in the construction of Volvo Trucks not only increases efficiency but also enhances workplace safety by reducing the risk of physical strain and repetitive motion injuries for the operators. Additionally, automation can lead to more consistent and standardized assembly results, contributing to higher product quality and customer satisfaction.

Figure 1.1 Demo Station at Volvo's Lab (image provided by Volvo)

1.2 Challenges

In the preceding subsection, it is highlighted that the assigned worker is burdened with monotonous and repetitive tasks, such as fastening bolts as well as inserting pipes and industrial components onto the table. These operations can be time-consuming and laborious, occasionally requiring collaboration between workers from different stations to meet time constraints. To enhance overall efficiency, automation of the assembly process emerges as a promising solution. By automating the assembly of parts, the system aims to alleviate the worker's workload and expedite the overall task completion, thereby streamlining the entire operation.

Robotic bin-picking directly addresses industrial challenges by automating monotonous tasks, ensuring precise control, reliable object identification, and efficient grasp planning. This technology streamlines assembly processes, reduces costs, and promotes industrial automation adoption. These challenges encompass precise control of robotic motion to ensure accurate manipulation of objects, accurate and reliable identification of various objects within cluttered and unstructured environments, and the development of robust grasp planning algorithms for efficient and successful pick-and-place operations. Addressing these challenges is of paramount importance to further enhance the capabilities and efficiency of robotic bin-picking systems and foster their widespread adoption in industrial and manufacturing settings.

A challenge faced by bin-picking applications pertains to the picking environment's complexity and diversity. Objects to be grasped may be situated in an open space or confined within a bin or table. Additionally, they could either be isolated or stacked in the bin, necessitating distinct approaches for successful grasping. Devising strategies to adapt the robotic system to diverse and dynamic picking scenarios is essential for robust and reliable performance. In this thesis, the Volvo object is placed at the center of the table, devoid of clutter.

1.3 Literature review/Related work

Within the domain of bin-picking applications, research efforts have primarily focused on household objects [15, 16, 17] and industrial objects [18, 19, 20, 21]. Notably, previous studies [22, 23] have predominantly leveraged datasets such as the Dex-Net Dataset [24] and Cornell Dataset [25] to address grasp prediction for regular household objects. While these models show promise in that context, applying them to industrial objects presents challenges due to distinct variations in object characteristics and environmental constraints [19]. Feng et al. [26] contributed by creating an instance segmentation dataset tailored for industrial objects, which is crucial for training models capable of handling industrial complexities.

Recently, researchers like Xu et al. [27] have directed their attention towards instance segmentation in industrial bin-picking using 3D point cloud data. This approach aims to accurately locate individual objects amid clutter, enabling effective grasping strategies. Novel research has emerged to address the distinct challenges posed by bin picking for industrial objects [28, 18]. For instance, in the work presented in [28], the spotlight is placed specifically on industrial objects. This research contributes by offering accurate 6D pose estimations, thereby enabling the application of machine learning techniques in industrial bin-picking environments.

Within the domain of robot manipulation, a crucial factor is the ability to predict the best possible grasp for a robotic gripper to securely hold an object. Grasp prediction involves configuring robotic grippers optimally for the action of picking and lifting objects. This accuracy challenge is influenced by factors spanning the information fed to grasp planning models to the algorithms employed. While certain works address this, bin picking applications confront further challenges in cluttered or dynamic environments [29, 30, 31, 32, 33]. Additionally, studies like Ponce and

Latombe [34] and Nguyen et al. [35] explore force-closure grasps, generating suitable grasp poses for two-finger grippers using a top-down approach to enhance grasp reliability. It is important to note that the scenarios in this thesis do not involve objects placed in cluttered settings.

Grasp representations in 3D space can be broadly categorized into two types: 3 DOF [14], [36] and 6 DOF grasp [37], [38], [39],[40]. In 3 DOF grasps, only the x and y position along with the yaw angle of the gripper in 3D space is considered, while in 6 DOF grasps, both the position and orientation of the gripper are taken into account. The choice of grasp representation impacts the complexity and versatility of the grasping strategy. In thisr thesis, we adopt a 6 DOF grasp representation to define the grasp pose. This representation encompasses the essential parameters of the grasping configuration, which consist of the x, y, and z position coordinates, along with the x, y, z, and w components of the quaternion orientation.

In the context of robotic grasp planning, several prevalent works [41] have employed various types of camera inputs, such as point clouds [42], [43], RGBD Images [44] [45], and Voxel Grids [39]. Gupta et al. [46] proposed geocentric embedding as a means of feature extraction and learning feature representations from depth images. point clouds [42], [47], [15], [43] have emerged as a data type in the field of grasp prediction, primarily due to their capacity to capture extensive spatial and geometric information. This representation of objects enables improved grasp predictions and higher success rates. Additionally, the fusion of point cloud data from multiple perspectives enhances the accuracy and richness of the object representation, further enhancing the effectiveness of grasp prediction algorithms. Processing point cloud data can be achieved through different approaches, including employing specialized networks such as PointNet [48] and Pointnet++ [49] [50]. In summary, the choice of camera input and the methods used for data processing impact the development of effective grasp planning models. In this thesis, the point cloud of Volvo's industrial objects is collected and utilized in the baseline testing, employing the pretrained model of Grasp Pose Detection (GPD) [51].

The success and generalization ability of grasp prediction algorithms are also influenced by various factors such as diversity in data in terms of shapes, sizes, poses, grasp pose representation, etc. In the subsequent chapters, tests are carried out on the industrial objects using pretrained model of GPD. GPD relies on the Big Bird dataset [52] for its training. This dataset comprises 125 objects, each with 600 RGB-D point clouds, and provides pose information for each image and point cloud. The success rates may be suboptimal while using pretrained models on novel objects and it highlights the importance and influence of datasets. Therefore, this thesis focuses on building a new dataset for the industrial models provided by Volvo. A grasp prediction MLP model is then trained on the created dataset.

Researchers have also explored various other approaches for dataset generation, including the use of synthetic data in addition to traditional open-source datasets. Synthetic data generation using PyBullet simulations allows researchers to define physical constraints for objects and generate randomized object poses. For instance, the CATGrasp [53] framework leverages a database of synthetic data to predict grasps on a novel, densely cluttered objects. Similarly, REGNET [54] trains on a large-scale dataset with simulated object poses, using YCB objects loaded into MuJuCo [55] to record various positions and orientations. Furthermore, the PointNetGPD approach created a substantial dataset comprising 350k parallel jaw grasps with associated point clouds and grasp quality scores, enabling robustness assessment for each grasp prediction. These strategies of dataset creation, both traditional and synthetic, play a vital role in evaluating the efficiency and accuracy of grasp predictions on novel objects, ensuring the algorithm's adaptability to diverse real-world scenarios.

The work presented in [28] focuses on the estimation of 6D pose for picking industrial objects. The researchers compiled a diverse dataset, encompassing both synthetic and real-world scenarios, to train and evaluate their grasp prediction model. This work contributes to the development of grasp planning algorithms tailored for handling industrial objects. It also highlights the importance of incorporating diverse and annotated datasets to advance the field of robotic bin-picking.

The Rutgers APC dataset [56] and the T-Less dataset [57] have been utilized for grasp prediction tasks; however, they exhibit limitations in terms of homogeneity and pose variability. The Rutgers APC dataset comprises 24 objects set in an unstructured warehouse environment. While it offers a diverse set of objects, the dataset lacks sufficient pose variability, which can restrict the model's ability to generalize well to various real-world scenarios.

Similarly, the T-Less dataset includes 30 industrial objects, which could potentially cater to the needs of grasp prediction research. However, the dataset suffers from data redundancy, potentially limiting its ability to provide a wide range of object poses and configurations. This lack of variability may hinder the robustness and effectiveness of grasp planning models trained on this dataset. More recent algorithms, such as CGPN [58] [59], have focused on utilizing depth images exclusively to predict 6 DOF grasps. These approaches leverage image encoders to extract relevant scene information, enabling the generation of collision-free and robust grasps.

1.4 Contribution and limitations

In this thesis, the aim is to develop a bin-picking method suitable for real industrial objects. A new dataset is created using the industrial objects provided by Volvo, and a multilayer perceptron (MLP) model is devised and trained using this newly generated dataset for grasp prediction. Additionally, the industrial objects are also tested on the pretrained model of Grasp Pose Detector (GPD) [51] as a baseline analysis. For each round of the baseline experiment, an object is placed at the center of the table, and point clouds of the object are recorded. The grasps predicted by the model are then executed. It is observed that the success rate of the grasps on the novel industrial objects is not high. The details on the process and results are discussed in Chapter 3 and 5 respectively.

Building upon the results obtained from this baseline evaluation, we proceed to construct dedicated datasets for both training and testing multilayer perceptron (MLP) models. This thesis analyzes and documents the accuracy of grasp classification for singulated Volvo objects. We focus on utilizing small datasets as large datasets may not always be available in industrial settings.

The analysis in this work aims to add understanding that can benefit the future development of a generalized bin-picking algorithm suitable for industrial objects. The grasp prediction models are trained on each provided industrial object for experimental analysis, but they are not transferable across different objects, which would require feature extraction from visual input such as point clouds. Additionally, the analysis conducted in this thesis revolves around a specific set of singulated industrial objects, which may not comprehensively represent the wide diversity of objects encountered in complex and unstructured real-world environments (such as cluttered settings). The findings obtained from this limited set of objects may require further validation on a broader range of items and settings to ensure the generalization ability of the results.

2
Background

This chapter provides an in-depth analysis of the software, hardware, and theoretical foundations that form the basis of the implementation of this thesis. In this study, we work with a UR10 robot [60], a two-finger RG6 gripper [61], and the ROS (Robot Operating System) communication interface [62].

To achieve the desired outcomes, the software aspect plays a crucial role. Various software tools and algorithms are employed to facilitate efficient data processing [8], [20], robot control [9] [10], and decision-making. On the hardware front, the UR10 robot is versatile, reliable and capable of handling various tasks in industrial environments. The two-finger gripper is selected for its ability to grasp and manipulate objects of different shapes and sizes. The gripper's design and functionality align with the requirements of the bin-picking task, ensuring efficient object manipulation and transfer.

The integration of the ROS communication interface serves as a vital component in this project. ROS provides a flexible and modular framework for the development of robotic systems. It facilitates seamless communication between different software modules, simplifies the implementation of complex control algorithms, and enables the integration of diverse hardware components. The utilization of ROS enhances the overall efficiency, modularity, and interoperability of the robotic bin-picking system.

2.1 Grippers and UR10 robot

2.1.1 Universal Robot 10

The experiments in this thesis are carried out using a specific collaborative robot model called UR10. The UR10 robots can handle physically heavy and repetitive tasks with ease. They have a maximum payload of 10 kg and are often deployed in the automotive and food manufacturing industries. The combination of collaborative features, ease of use, flexibility, and versatility make it a great choice (Figure 2.1).

This collaborative robot has 6 rotating joints (Degrees of Freedom) and a 12-inch touchscreen with a PolyScope graphical user interface. Additionally, the working

Figure 2.1: UR10 robot. Image source: Universal Robots.

range of the axis movement of the base, shoulder, elbow and the three wrists are all 360 degrees. The movement of UR10 can be controlled using the interactive tablet or ROS communication.

2.1.1.1 Trajectory planning using Inverse Kinematics (IK) solver

The successful accomplishment of a robotic task greatly relies on the chosen motion trajectory undertaken by the robot. A trajectory refers to the path or sequence of positions that a moving object, such as a robot, follows over a specific period. It represents the planned sequence of spatial configurations or poses that the robot needs to traverse to accomplish a particular task or movement. It is desirable for the robot to select a trajectory that not only minimizes the execution time but also ensures safety. Safety considerations encompass collision avoidance and compliance with joint limits throughout the trajectory execution. To address these requirements, trajectory planning utilizing an Inverse Kinematics (IK) solver offers a systematic and efficient approach for generating feasible and smooth robot motion.

For trajectory planning in this project, the TRAC-IK Kinematics Solver (IK_solver) is used within the robot environment. This solver works with the `MoveIt!` framework [63], which operates within the Robot Operating System (ROS). The MoveIt! framework is a widely used motion planning and manipulation software package in the field of robotics. It is an open-source software framework that provides a comprehensive set of tools, libraries, and APIs for motion planning, control, and manipulation tasks. The TRAC-IK solver offers improvements over the conventional KDL's convergence algorithms by addressing the challenges associated with joint limits.

2.1.2 Two finger gripper

The selection of an appropriate gripper for the UR10 robot is important as it significantly impacts the outcomes of the project. The fundamental objective of a gripper is to mimic or improve the dexterity and functionality of a human hand, enabling the robot to grasp objects smoothly and efficiently. There exists a wide range of grippers available, each offering distinct features and capabilities. Examples include vacuum grippers [1], magnetic grippers [2], two-finger grippers, and multi-finger grippers [3] [4], as illustrated in Figure 2.2. Each type of gripper possesses its own unique functionalities, and some are equipped with sensors that further enhance their performance in the task of bin-picking objects.

In the context of this project, the focus lies on the versatility and effectiveness of two-finger grippers, which have been extensively discussed in the literature [64]. The two-finger grippers demonstrate the ability to successfully grasp objects and exhibit a wide range of functionalities that contribute to their effectiveness in the bin-picking task. In this thesis, the objects provided by Volvo exhibit distinctive features, and the gripper chosen for this application is the RG2-FT gripper. This gripper is equipped with sensors at the fingertips, enabling it to measure the force and torque applied to the object between its fingers. During the course of the study, technical issues were encountered, leading to the replacement of the initially intended RG2 gripper with the RG6 gripper. The physical dimensions of the RG 2 gripper is in Figure 2.3 and the RG6 gripper are illustrated in Figure 2.4. However, despite this change, the overall experimental setup and methodology remained unaffected. A comprehensive explanation of the modifications made to the experiments using the RG6 gripper is provided in the Appendix.

Figure 2.2 Left to right: vacuum grippers [1], magnetic [2] and multi-finger grippers [3] [4]

2.2 RealSense RGB-D camera

The Realsense RGBD camera [45] is a highly versatile sensor that offers both color and depth information, making it well-suited for a wide range of robotic applications, including bin picking [65, 66, 67] and fast-moving scenarios. Its ability to capture detailed 3D information and generate dense point clouds enables accurate perception, grasp planning, and manipulation of objects within complex and dynamic environments. These point clouds contain information about the shape, size,

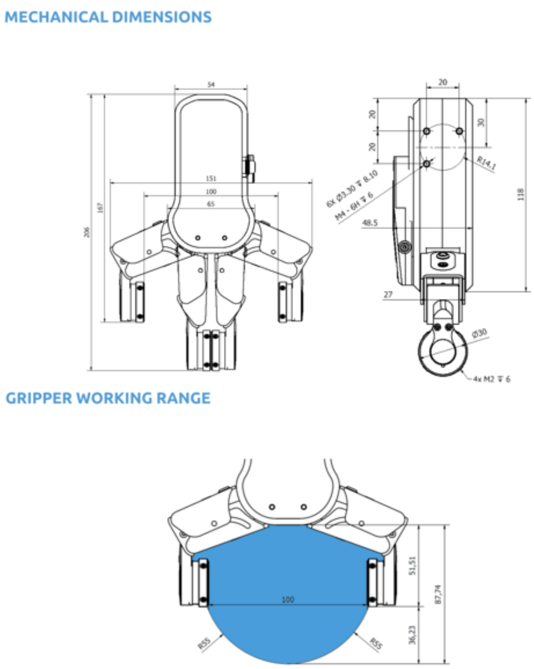


Figure 2.3 The physical dimensions and the area enclosed by the fingers of the RG2-FT Gripper as provided by the datasheet.

and position of objects.

Point clouds provide a comprehensive representation of the object, even in occluded settings where parts of the object may be hidden from view. They offer spatial information about the scene and enable the prediction and evaluation of grasp positions and orientations. By leveraging the use of point clouds, the process of object identification and manipulation within a given scene becomes more efficient and straightforward. The rich spatial information contained in point clouds enables a better understanding of the contours and shapes of the objects present in the scene.

2.2.1 Calibration of camera on UR10 and point cloud Generation

Camera calibration is a crucial step in computer vision and 3D reconstruction tasks as it establishes an accurate relationship between 3D points in the real world and their projections onto a 2D image plane. By estimating the intrinsic and extrinsic camera parameters, camera calibration enables precise mapping of image coordinates to real-world coordinates.

Camera calibration is typically performed by capturing images of a known calibration pattern, such as a checkerboard, from different viewpoints. By analyzing the image correspondences between the calibration pattern and its 3D reference points, the camera parameters can be estimated using mathematical techniques such as geometric or optimization-based methods. To ensure reliable and accurate point cloud

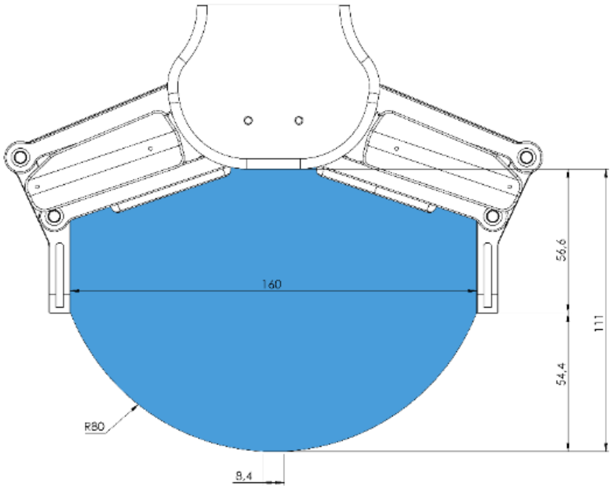


Figure 2.4 RG6 gripper from the datasheet [5]

generation, it is crucial to calibrate the camera mounted on the UR10 robot. This is done using [68]. The RealSense camera is positioned between the robot's wrist and gripper to capture the scene. In Rviz, the camera output is observed in real-time by selecting the camera tool, as shown in Figure 2.5 and includes a panel displaying the point cloud scene generated by the calibrated camera.

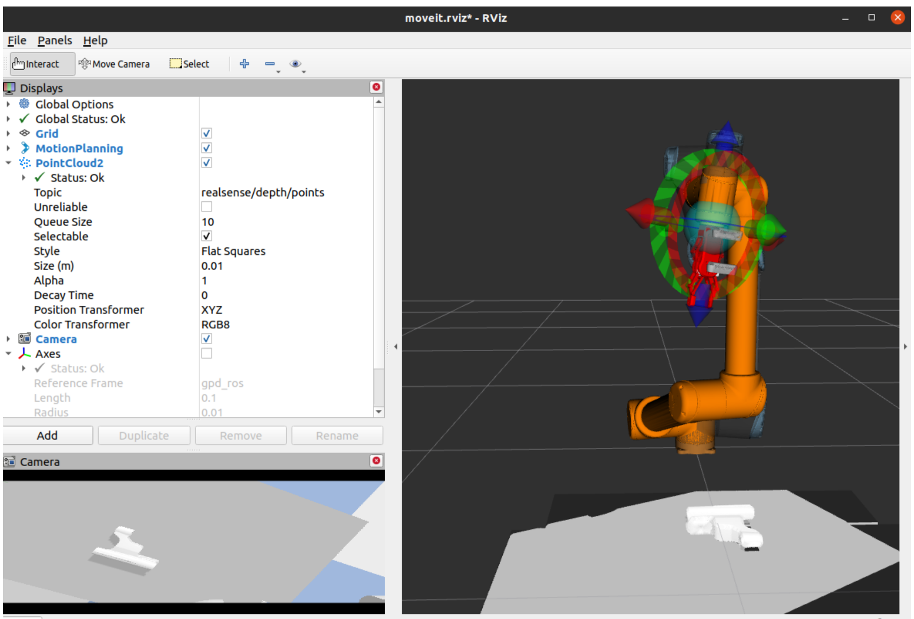


Figure 2.5: Visualization of an industrial object on Rviz.

2.3 Grasps

Bin-picking applications [69], [70], [71], [72], [73] usually consist of two main types of grasps - SE(2) and SE(3). A two-finger grasp with a top-down approach is referred to as SE(2) and the grasp poses are obtained in the form of x, y, θ [74] where x and y is the position and θ is the orientation.

In the thesis, the investigation involves using SE(3) grasps, a pose with six degrees of freedom, and offers a more comprehensive solution for applications involving bin picking. While it overcomes the limitations of the SE(2) grasps, the implementation is significantly more challenging.

In the context of grasp evaluation, antipodality refers to a geometric property of a grasp where the fingers of a gripper make contact with an object at a pair of points with opposite surface normals. These pairs of points are known as antipodal points (Figure 2.6). Antipodal grasps are particularly important because they ensure force closure, which means that the fingers exert forces on the object that create a stable and secure grip. Achieving force closure is crucial for successful manipulation tasks, as it provides stability and control over the grasped object. When the fingers make contact with the object at antipodal points, they generate opposing forces that result in a tight grasp, preventing the object from slipping or being easily dislodged. The concept of antipodality is commonly used as a metric for evaluating the quality of grasps. Grasps with antipodal contact points are generally considered to be more robust and stable, as they provide better force closure and a higher likelihood of successful manipulation.

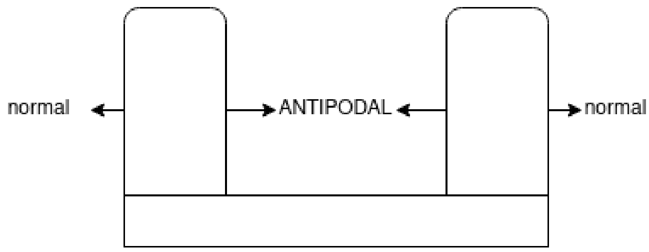


Figure 2.6: An illustration of an antipodal grasp. It is inspired from the explanations provided in [6]

2.3.1 Visualization of the datatset

The training and testing datasets are plotted on the point cloud images of the Volvo objects to enhance the visualization and understanding of the different types of successful and unsuccessful grasps. This approach provides a visual representation of the grasp locations and helps to identify patterns or characteristics associated with successful or unsuccessful grasps. The grasps from the training dataset are visualized in Figure 2.7 and the ones from the testing dataset are visualized in Figure 2.8

Including these visualizations in the thesis allows for a more intuitive understanding of the grasp quality and facilitates the analysis of the dataset's effectiveness in capturing the diversity of grasp configurations. Additionally, it enables the comparison of grasping patterns between the training and testing datasets.

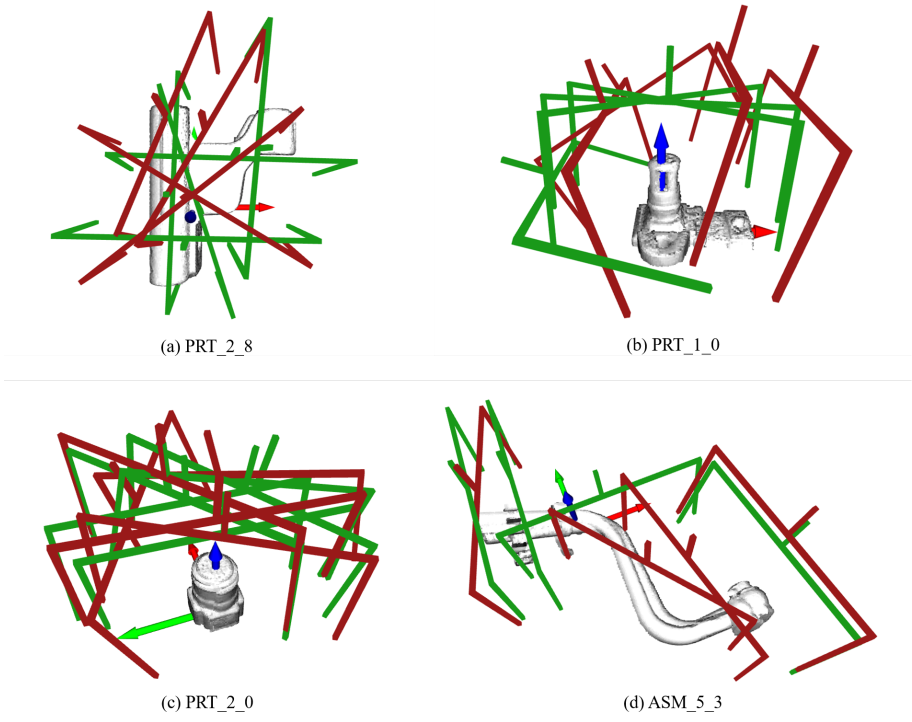


Figure 2.7: Illustration of the training data - a subset of the recorded grasp poses for each object. The grasps are depicted in red and green which signify unsuccessful and successful respectively.

2.3.2 Preprocessing Point clouds

In the process of generating the point cloud of the scene, whether in simulation or real world using the UR10, the resulting point cloud includes the table on which the object is placed (as shown in Figure 2.9).

Cropping out the table provides several advantages in the context of the research study. It simplifies the perception task by reducing the complexity of the scene. By removing the table, the focus is solely on the object of interest, which facilitates easier and more accurate analysis of the point cloud data. Furthermore, table removal has the benefit of reducing noise and false detections. From a visualization standpoint, the removal of the table improves the clarity and interpretability of the generated grasps.

In the thesis, the process of table removal is accomplished using two different approaches: RANSAC and a user-guided region selection. The choice between these

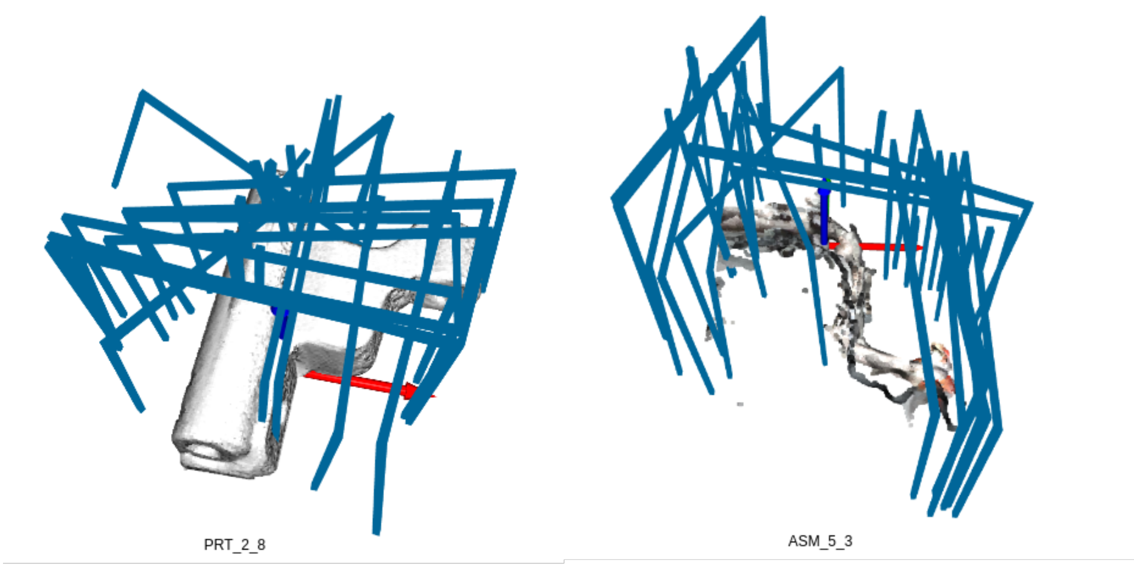


Figure 2.8: Illustration of the test grasps (grasps produced for MLP test dataset)

approaches depends on several factors, including the specific requirements of the application, the quality of the point cloud data, and the available resources for manual intervention.

RANSAC (Random Sample Consensus) is a widely used algorithm for robust plane fitting and can automatically identify and remove the table surface based on an analysis of the point cloud [75]. On the other hand, manual cropping involves manually selecting and removing the table region from the point cloud data. The decision to employ either RANSAC or manual cropping depends on the specific circumstances of the research, including the complexity of the table geometry, the accuracy requirements, and the feasibility of the region-selection intervention.

Both methods have their advantages and limitations. RANSAC provides an automated way to estimate the table plane, but it may be sensitive to noise or variations in the point cloud data. Manual cropping, on the other hand, offers more control and can be tailored to the specific scene, but it requires manual effort and may not be applicable in all cases. Figure 2.10 shows the result of applying RANSAC to one of the industrial objects provided by Volvo.

2.4 Robot Operating System (ROS)

ROS (Robot Operating System) is a widely adopted communication framework extensively used in robotics for seamless communication between different components and workspaces. It serves as an open-source tool that enables effective communication and provides visualization capabilities within a robotic project. By leveraging ROS, researchers and developers can establish efficient communication channels, exchange data, and coordinate the activities of different modules within a robotic system.

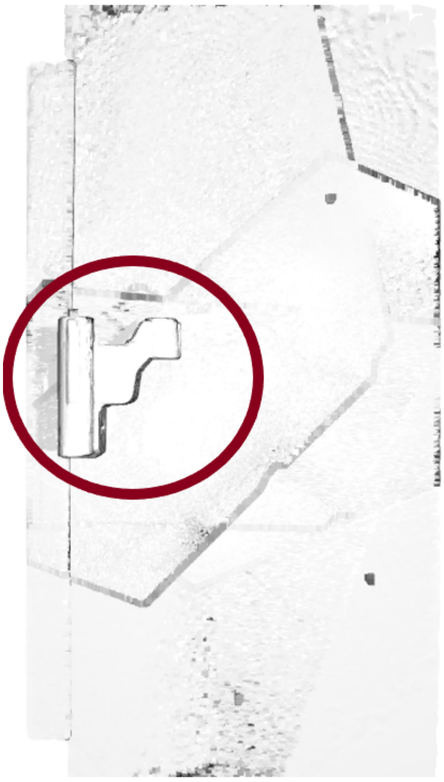


Figure 2.9: Point cloud generated from PyBullet simulation. The figure displays the scene in the form of point clouds and the Volvo industrial object is at the center, inside the red circle.

In ROS, communication between components is facilitated through nodes, which act as endpoints for message publication and subscription. Nodes enable the transfer of information by creating topics and declaring the corresponding message types. This allows different parts of the system to exchange data and synchronize their operations. For instance, in the context of integrating the GPD (Grasp Pose Detection) and UR10 components, the ROS framework played a crucial role in establishing communication between distinct workspaces. Point clouds are passed to GPD for predicting grasp poses.

A detailed explanation of this process, including the setup and utilization of ROS for communication is presented in Chapter 3, Section 3. By employing ROS as a communication tool, the integration and coordination of different modules within the robotic system are achieved effectively, facilitating seamless data transfer and collaborative operation.

2.5 Industrial objects by Volvo Group

The objects used for bin picking in this project are provided by Volvo, as depicted in Figure 2.11. These objects exhibit variations in their geometrical features, weights,

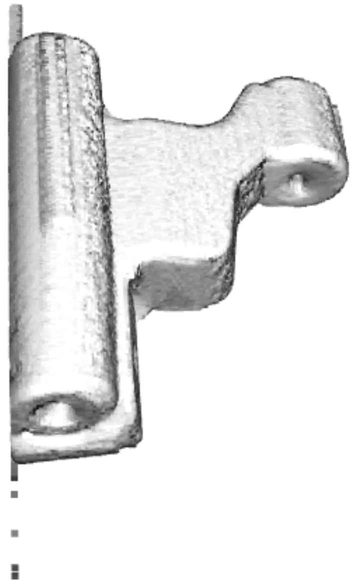
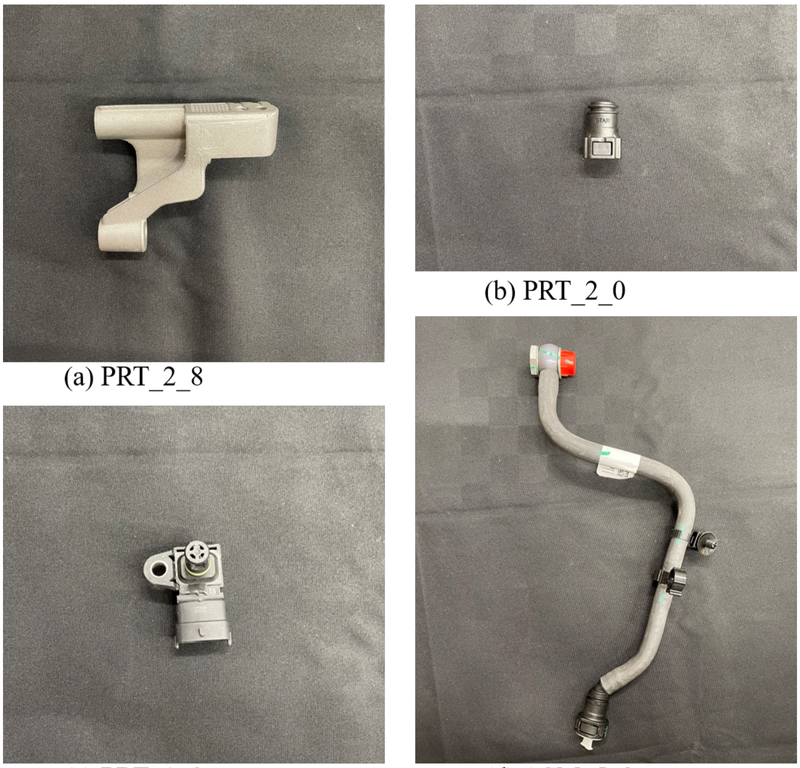


Figure 2.10: The figure depicts the result of applying RANSAC to remove the table from the pointcloud shown in Figure 2.9

and surface conditions, contributing to the complexity of grasp-learning problems. During two-finger grasping tests, it is observed that viable grasps for objects (a) and (d) are more dispersed around the objects. Furthermore, object (a) requires more adjustments in the gripper's opening width, while object (b) necessitates a greater change in the grasp position. Despite both object (b) and (c) being small in size, it is evident that object (b) is easier to pick up due to local detailed features on the geometry of object (c) adversely affecting the success of grasping.

To ensure better alignment between the simulation and the real robot setup, CAD models of the objects are employed. These CAD models serve as the foundation for generating point clouds and models, which are then utilized to create scenes in the simulation environment.

To achieve accurate alignment, the CAD models undergo translational and rotational adjustments using a software tool called MeshLab [76]. The purpose of these adjustments is twofold. The first purpose is to set the origin of the object model at a distinguishable surface point on the object. This enables establishing correspondences between the physical object and its virtual representation in the simulation, facilitating precise perception and grasp planning in both simulation and real-world scenarios. The second purpose is to ensure that the frame orientation of the object model is aligned with the table frame. This ensures consistency between the reference coordinate systems of the objects in the simulation and the physical environment, enabling seamless coordination between the two.



(a) PRT_2_8 (b) PRT_2_0 (c) PRT_1_0 (d) ASM_5_3

Figure 2.11: Objects provided by Volvo for grasp learning

2.6 PyBullet - simulation tool for grasp executions

PyBullet [77] is a popular open-source simulation tool that is an intuitive, interactive tool with multiple features such as collision detection and rigid body dynamics. It is a powerful academic tool due to its high efficiency and performance integration with other frameworks and applications.

In context to the thesis, the PyBullet simulation environment mimics the working environment of the UR10 robot and RG2-FT gripper. The positioning of the Volvo objects on the table, its frames and the trajectory undertaken by the IK solver play a crucial task in the execution of the simulations thereby affecting the outcomes. In context of this thesis, PyBullet is adopted as the tool for building the digital twin of the real robot and facilitates the simulation as well as testing.

2.7 RVIZ visualization

Rviz is a powerful and versatile 3D visualization tool that is a part of the ROS package and is commonly used to visualize sensor data, camera output, frames, overall robot system; etc. The tool provides users with a 3D environment to navigate and manipulate objects and camera view for deeper analysis. It is particularly useful as it allows users to observe the trajectory executed by the UR10 robot within the simulation. Furthermore, Rviz provides motion control features that enable users to manually move the UR10 arm to specific locations, both within the simulation and in the physical UR10 robot. The following image (Figure 2.12) shows an example of the visualization.

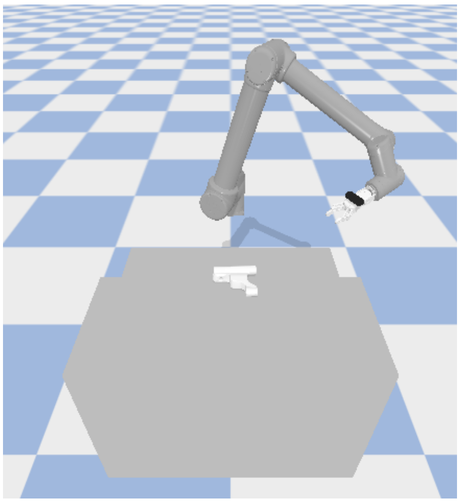


Figure 2.12: The grasps by GPD being executed in the PyBullet environment.

3
Grasp Pose Detector

3.1 Grasp Pose Detection(GPD) baseline

The Grasp Pose Detector (GPD) [51] is a widely adopted framework for predicting grasps on point clouds of Objects. Its selection as a baseline in this thesis is motivated by its proven accuracy and implementation in the field of robotic grasping. The GPD system has been trained using a diverse range of regular household items and has demonstrated a remarkable success rate of 93 percent when applied to bin-picking scenarios, even in the presence of dense clutter. To evaluate the effectiveness of GPD, we conducted tests on Volvo objects using PyBullet simulations and its performance is discussed in Chapter 5. A grasp is considered successful if the object can be held in the air for a duration of ten seconds or more, as shown in Figure 3.1.

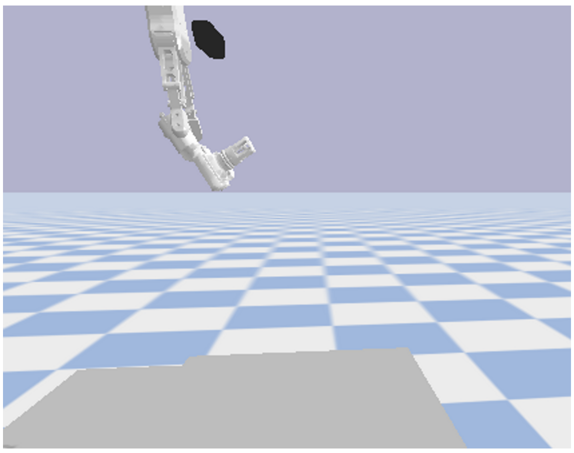


Figure 3.1: A successful grasp of the Volvo industrial object (c) PRT _1_0 (from Figure 2.11) by GPD on PyBullet

The research conducted on the GPD system provides insights into the determination and scoring of successful grasps. The grasp candidates are classified using a binary classification and a Convolutional Neural Network(CNN). This network has four layers and its output predicts if the proposed grasp candidate is successful or not. The CNN used in GPD is similar to LeNet [78]. In this thesis, grasps are classified as successful or unsuccessful using a multilayer perceptron (MLP). Chapter 4 discusses the design and implementation of MLP models for each of the industrial objects provided by Volvo. The Convolutional Neural Network (CNN) is trained using a dataset of labeled grasps for 55 out of the 125 objects present in the Big Bird Dataset [79]. Chapter 4 discusses the methodology used to create the training and testing datasets for the multilayer perceptrons (MLPs).

In the Grasp Pose detector (GPD) framework, where input is in the form of point clouds, the grasps are represented by three vectors as depicted in Figure 3.2. These vectors play a significant role in determining the orientation and alignment of a robot hand or gripper with respect to an object. The first vector, known as the approach vector, defines the direction in which the gripper approaches the object during the grasping motion. This vector not only guides the movement of the gripper but also determines the direction of the force exerted by the gripper onto the object during the grasp. The second vector, referred to as the axis vector and depicted in blue, specifies the rotation axis of the gripper. It influences the rotational motion of the gripper as it closes around the object during the grasp. The third vector, called the binormal vector, is orthogonal to both the approach vector and the axis vector. This vector plays a crucial role in defining the orientation of the gripper. Typically, it is oriented perpendicular to the grasping surface, providing additional information about the desired hand orientation during the grasp. By considering these three vectors, the GPD system is able to generate grasps that incorporate both the approach direction and the rotational motion of the gripper, while also ensuring an appropriate gripper orientation for successful grasping.

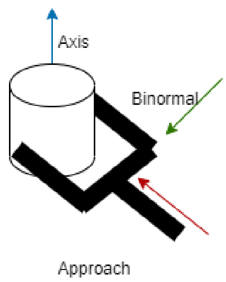


Figure 3.2: Co-ordinate frame of GPD [7]

3.1.1 Execution of baseline grasps in PyBullet simulation and real UR10 robot

Using these point clouds as input, GPD generates a set of grasps; along with their scores; around these objects as illustrated in Figure 3.4. These grasps are separated into binary classes, and the successful grasps are ranked in descending order of score; with a higher score indicating a higher probability of successful execution. The scoring of the final grasps obtained from the Grasp Pose Detection (GPD) system is based on a cost function that incorporates multiple factors. Defined based on qualitative experience, the cost function prioritizes the grasps with high grasp position and top orientation, while penalizing long travel distance for the robot arm required by a grasp candidate. The purpose of this cost function is to evaluate and select the grasp that exhibits the highest likelihood of success by considering these influential parameters as discussed above [80].

During the grasp generation process, it is possible to define a region of interest (ROI) within the input point cloud by adjusting the parameters within the provided config file. By specifying the desired area of the workspace and the number of grasps to be generated, the Grasp Pose Detector (GPD) system can focus on extracting potential grasps within that region. Once the grasps are generated, they are scored, and the grasp with the highest score is selected as the best candidate.

In the PyBullet simulation environment, the grasps generated by GPD are executed as seen in Figure 3.3. To ensure precise execution, a novel reference frame, denoted as the "GPD frame", is employed. While GPD provides a comprehensive and intuitive approach to grasping objects, it is worth noting that the model displays a better performance in grasping household items. Its performance may vary when applied to objects from industrial settings or other specific domains.

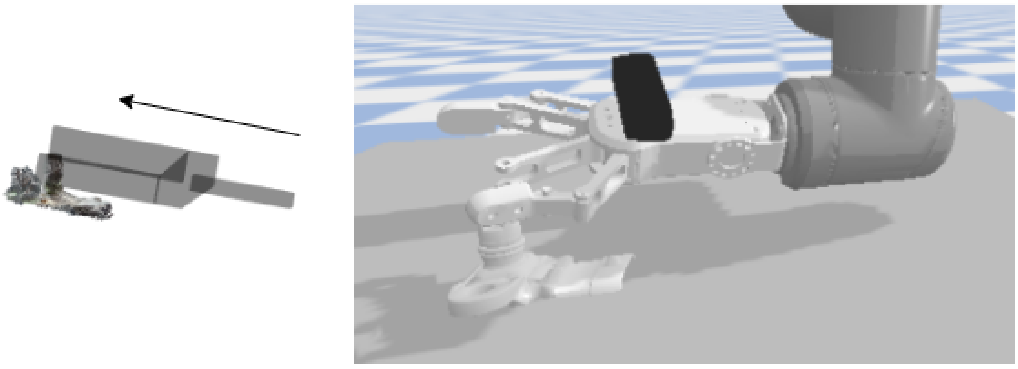


Figure 3.3: Left: Grasp proposed by GPD. The arrow indicates the direction of the arm while trying to grip the object; Right : Execution of GPD grasp in PyBullet simulation

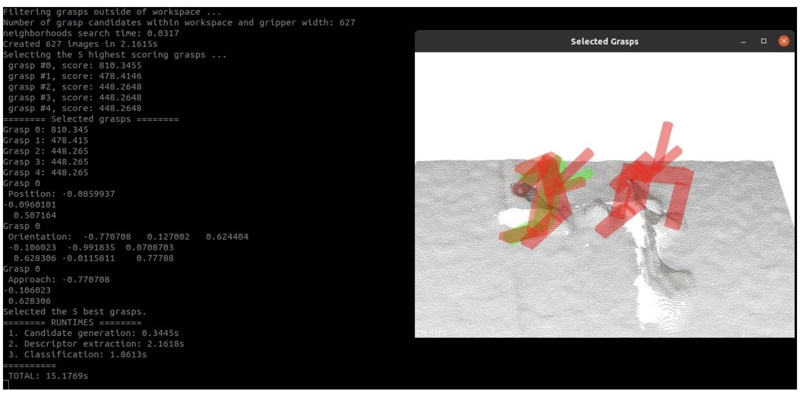


Figure 3.4: The GPD grasps that are scored. Red color indicates grasps classified by the model as unsuccessful while green signifies successful grasps.

4

Proposed method and model for Grasp Classification

4.1 System overview

In this thesis, 6DOF quaternion representation of grasps are given as input to a multilayer perceptron (MLP) for training. The process of generating the input grasp candidates is discussed in Section 4.5.1. The successful grasp poses are subsequently evaluated in two distinct manners: through a UR10 robot implementation and via PyBullet simulation. Figure 4.1 shows an overview of the proposed method for grasp classification.

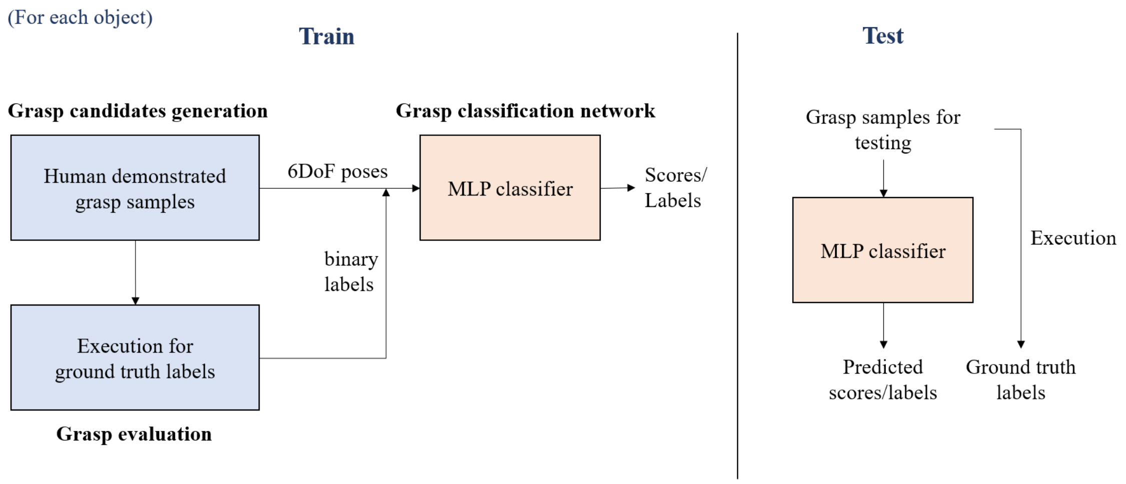


Figure 4.1: Overview of the system

4.2 Model architecture - MLP

Multilayer perceptrons (MLPs) are artificial neural networks that consist of multiple layers of interconnected neurons. MLPs are effective in capturing and learning non-linear relationships within data, making them suitable for a wide range of tasks such as classification and regression.

The MLP is composed of an input layer and an output layer, with each layer comprising nodes or neurons to represent the inputs and outputs, respectively. Additionally, intermediate layers, known as the hidden layers, are positioned between the input and output layers. The hidden layer can consist of any number of nodes. Figure 4.2 displays the schematic diagram of the MLP. The diagram showcases two input nodes, aligning with the inputs in the input layer, along with a solitary output node within the output layer. The initial hidden layer consists of 5 nodes, obtaining input from the preceding layer (the input layer). This processed information is then conveyed to the second hidden layer, which incorporates 2 nodes. In a feed-forward manner, each layer's computation serves as the input for the subsequent layer. Consequently, this sequential computation continues until it culminates at the output layer, yielding the final computation outcome.

MLPs offer several adjustable parameters that can be optimized to improve their accuracy and performance. The activation functions parameter in MLPs employs activation functions to introduce non-linearity into the network. Common activation functions include sigmoid, tanh, and ReLU. MLPs can have multiple hidden layers, allowing them to learn hierarchical representations of the data [81]. The depth of the network, determined by the number of hidden layers, can affect its ability to capture and represent complex relationships in the data. The weights and biases play a crucial role in determining the model's performance and its ability to learn from the data.

By adjusting these parameters and optimizing them through techniques like gradient descent, the MLP can minimize the error in predictions and improve its overall performance. However, finding the optimal configuration for an MLP often requires experimentation and fine-tuning based on the specific task and dataset at hand. Section 4.4 elaborates on the process of configuring the MLP.

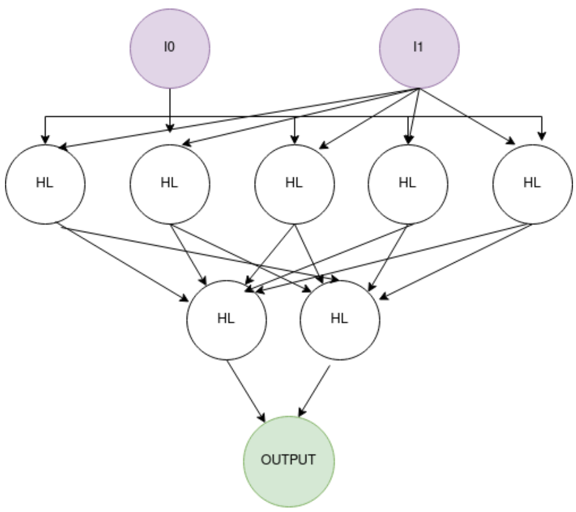


Figure 4.2: MLP layers where HL stands for Hidden Layer

However, it is important to note that MLPs can be sensitive to the scaling of input features. Feature scaling refers to the process of normalizing the range of input variables to a standard scale. To address this sensitivity to feature scaling, it is common practice to pre-process the input data by normalizing or standardizing the features. This ensures that all features have a similar scale and reduces the impact of uneven scaling on the MLP's learning process. By applying appropriate feature scaling techniques, the MLP can achieve more stable and effective learning results.

4.3 Dataset generation

The quality and variability of a dataset have a significant impact on the training of MLP models and ultimately influence their performance. It results in models that are more accurate, robust, and capable of generalizing well to unseen data.

As discussed in Chapter 1, the recent work in bin-picking projects often uses using deep learning to execute 6 Degree of Freedom (6DOF) grasps and rely on large and heavy datasets [82]. We have taken the approach of generating custom datasets by recording various grasping scenarios. The datasets contain essential information such as the position (x, y, z) in 3D space and quaternion orientation (x, y, z, w). By incorporating these features, the MLP models can learn the relationships between different variables and make accurate predictions during grasp planning and execution.

Generating our own datasets allows us to tailor the data collection process to the specific requirements of our research. This ensures that the datasets are well-suited to the objectives of our work and capture the relevant information needed to train and evaluate our MLP models effectively.

4.3.1 Representation of grasps in the MLP dataset

The grasp representations mentioned in the thesis include 6 DOF values of grasp pose (3D position and quaternion orientation). During the grasping process, the gripper's arm is manipulated to maneuver around the object positioned on the table. Subsequently, the gripper is closed around the object, allowing it to be lifted. Throughout this process, the corresponding position (x, y, z) and orientation (x, y, z, w) values are recorded. These recorded values are then associated with labels indicating whether the grasps are classified as successful or unsuccessful. A total of 72 grasps have been captured, each with its corresponding position and orientation information. A grasp is termed successful if the object can be picked up and it stays in the air for a minimum of 15 seconds.

The pre-trained models from existing dataset when tested with the Volvo objects did not yield successful results. Hence, we experiment with curating a new dataset to create a precise and accurate network for grasp classification.

4.3.2 Plotting grasps on Point clouds

A simplified hand skeleton, as seen in Figure 4.3, is plotted alongside the object's point cloud to visualize the grasps collected during the training data collection process. This visualization provided an intuitive representation of grasp poses. In order to align the grasp plot with the manually collected grasps, both the grasp poses and the point clouds are transformed into the table frame of reference.

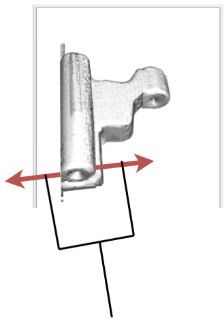


Figure 4.3: GPD would classify this grasp as successful but it fails in execution because of physical constraints.

4.4 Training of multilayer perceptron

Within the scope of the thesis, four distinct multilayer perceptrons (MLPs) are constructed, each with a unique architecture tailored specifically for classifying input grasps as successful or unsuccessful. The MLPs are trained on the dataset using the Adam [83] optimization algorithm. Although alternative optimizers such as Adagrad [84] and SGD [85] are evaluated, the Adam optimizer demonstrated better performance in this context.

The performance evaluation of each MLP model involved calculating the average validation accuracy over 10 iterations, where each iteration consisted of 50 epochs of training. The standard deviation of the validation accuracy is also computed to assess the consistency of the model's performance. The models are designed to achieve a test accuracy of approximately 80 percent. These results have been further validated using the UR10 robot in practical experiments. An analysis of the obtained results are presented in Chapter 5 of the thesis.

4.5 Testing and inference of the MLP models

During the course of this thesis, two types of finger grippers have been employed for data collection and model verification purposes. Initially, the RG2 Force-Torque gripper was utilized during the data collection phase. However, technical issues arose with the RG2 gripper, leading to its replacement with the RG6 gripper for the subsequent verification of the MLP model. The RG6 gripper is used in the verification process of the MLP model. Further clarification of the gripper switch and its implications are discussed in detail in Appendix. The grasp candidates for testing purpose are sampled uniformly and then scored by the MLP. Figure 2.8 illustrates the test grasps.

4.5.1 Collection of grasp poses for testing dataset

To assess the accuracy and robustness of the MLP models, grasps are sampled across the object models. These grasps are not specifically optimized for success or of the highest quality. This approach helps identify any potential limitations or areas for improvement in the system's ability to execute different grasp poses.

There are two possible ways to perform this task. The first alternative would be modifying existing frameworks like Grasp Pose Generator (GPG) [86] to generate grasps. GPG focuses on sampling a large amount of random grasp poses distributed around the point cloud of the object. These grasps are not assigned any specific score or predefined classification as successful or unsuccessful. The grasp candidates are later pruned by their feasibility and evaluated by the grasp classifier for further selection and execution.

The second alternative to generate grasps for testing is to sample directly around the object with an even distribution. It can be more time-consuming compared to the automatic candidate generator, and involves the exploration and manipulation of objects through direct interaction, while we can gain insights into which types of grasps are successful and how the object's shape and material properties impact the grasping process. This method involves demonstrating various grasp poses and analyzing their success rates.

For the thesis, the testing dataset for each object is generated by sampling grasps across the object. The first option of using frameworks such as GPG is more robust, however, we used the second option due to time constraints. This dataset includes various grasp positions and their quaternion orientations. 40 grasps for each object is generated, amounting to a total of 160 grasps. This served as an input to the MLP. Amongst these grasps, the ones labeled by the MLP as successful are executed using the UR10 Robot and RG6 gripper. The results of these executions are summarized in Table 5.2 in Chapter 5.

5

Results

In this chapter, we present the results obtained from our evaluation of the pre-trained Grasp Pose Detector (GPD) [51] model on Volvo's industrial objects, Figure 2.11. Additionally, the accuracy and effectiveness of the multilayer perceptron (MLP) model on the uniformly sampled grasp candidates are also evaluated in this chapter.

5.1 Execution of GPD grasps in PyBullet and using a real UR10 robot

Using the ROS framework, the grasps suggested by the GPD are evaluated in the PyBullet simulation environment. Among the Volvo objects, three of them exhibited favorable graspable positions. However, it is observed that in the simulation, a successful grasp is achieved around 1 out of 5 times. On Part 1_0, Part 2_0, and Part 2_8 successful grasps are observed as they objects remained stable in the air for at least 10 seconds. These results are visualized in Figures 5.1 and 5.2.

To further evaluate the performance of the GPD, the identified successful grasps are also executed on the UR10 robot. The grasps are visualized in Figure 5.3. Moreover, the GPD provided the position and quaternion coordinates of the successful grasps it selected. The evaluation here aims to investigate how well GPD generalizes to industrial objects, particularly the Volvo objects in this case.

Object	No.of successful executions by GPD	Percentage
Part2_8	0/5	0 %
Part1_0	2/5	40 %
Part2_0	2/5	40 %
Asm5_4	3/5	60 %

Table 5.1: 5 successful grasps from GPD are executed on the real UR10. The table records how many out of the 5 are executed successfully.

An observation drawn from the evaluation highlights a limitation of GPD when

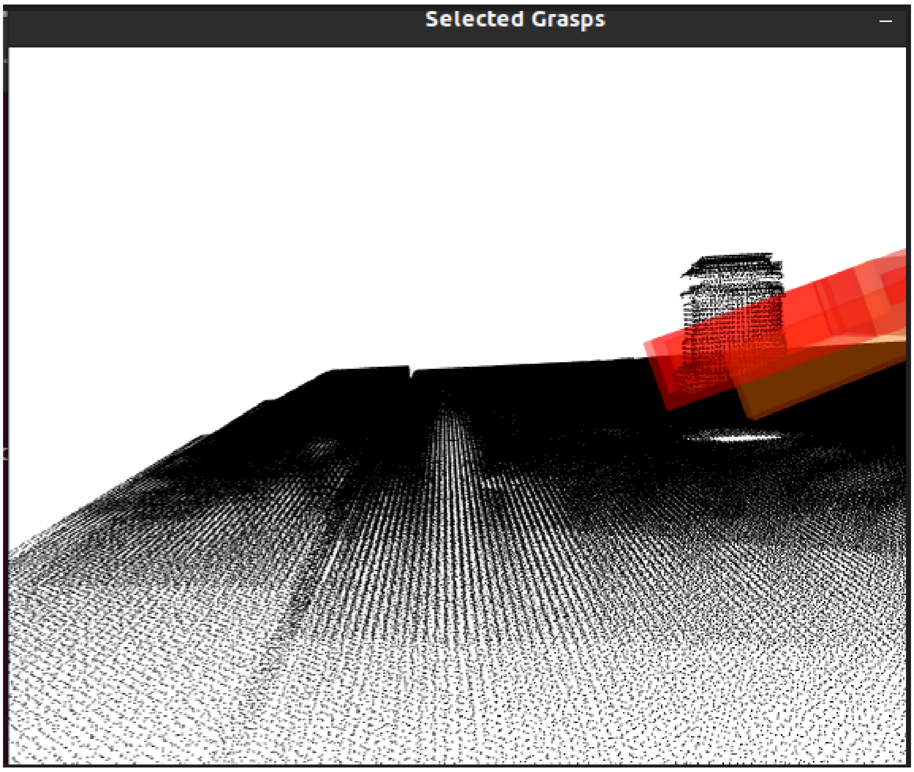


Figure 5.1: A successful grasp by GPD

applied to industrial objects. GPD may classify some grasps as successful, but the visual inspection of these grasps often reveals that they may not be of the best quality (further discussion on the possible reasons for failure are discussed in the following chapter). This observation is further supported by the conclusive evidence obtained from the testing on the UR10 robot. It has been found that while the pre-trained GPD model may work well for regular objects, its success rates for industrial objects used in this thesis, such as the Volvo objects in this case, are relatively low.

5.2 Training and testing accuracy of the MLP models

Datasets are created for the four unique MLPs, one for each of the four Volvo objects. This dataset is then split for training and validation in the ratio of 60:40. The final output of the MLP is binary and describes if the grasp is successful or unsuccessful. The accuracy results for validation and testing are described in the sections below.

5.2.1 Accuracy of MLP models during training

After training the MLP model, it is evaluated on the validation test set to assess its accuracy. The validation accuracy is recorded over 10 iterations to account for variability, and the results are presented in Figure 5.4. Additionally, the standard deviation of the validation accuracy values is calculated to measure the consistency

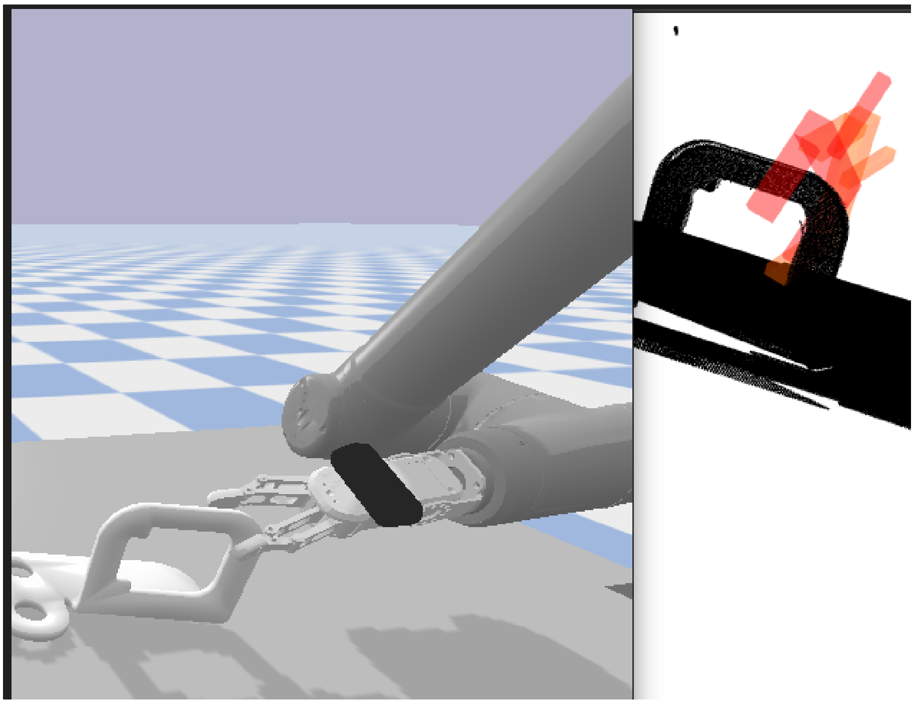


Figure 5.2: A grasp generated by GPD. This grasp is predicted to be successful.

of the model's performance across the iterations.

By analyzing the validation accuracy and its standard deviation, it is possible to assess the effectiveness and reliability of the MLP model for the task of predicting the position and orientation of the Volvo objects. Higher validation accuracy values indicate better performance and more accurate predictions, while a lower standard deviation suggests consistent and reliable results.

5.2.2 Testing the classification accuracy of MLP models

During the training and validation process, the MLP models achieved an accuracy of around 80 percent, indicating that they are able to make accurate predictions for the position and orientation of the Volvo objects. To further assess the accuracy and effectiveness of these predictions, the grasps classified as successful by the models are executed using the UR10 robot and the RG6 gripper.

To verify the accuracy of the model predictions, a carefully curated test dataset is used. The grasps in these datasets are visualized in Figure 2.8. With the testing dataset as input to the MLP models, the models classified certain grasps as successful, indicating that they are predicted to be successful grasps. These grasps are then executed by the UR10 robot using the RG6 gripper to verify if the prediction is accurate. The number of successful grasps for each object is recorded and summarized in Table 5.2.
It is important to clarify that the results presented in the table (Table 5.2) represent

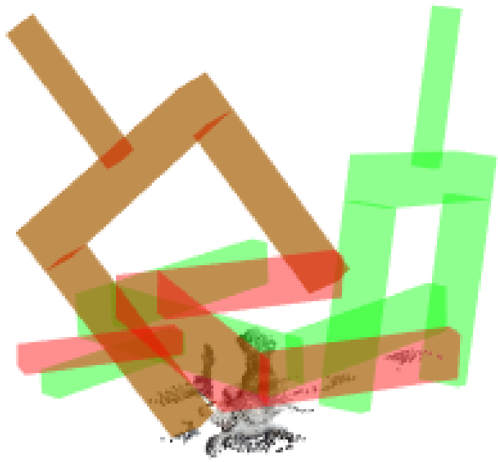


Figure 5.3: Grasps proposed by the baseline GPD

Object	Score	Standard deviation of validation accuracy
1. part2_8	Average validation accuracy: 0.9	0.10897247358851683
2. asm 5_3	Average validation accuracy: 0.9333333313465119	0.13333332538604736
3. part1_0	Average validation accuracy: 0.8166666626930237	0.11666665758405506
4. part2_0	Average validation accuracy: 0.8142857253551483	0.1115749887197071

Figure 5.4: The accuracy score and standard deviation values for MLP trained only using position and orientation values of a grasp. The training datasets include an average of 12 grasps (both successful and unsuccessful) per object.

Object	No.of successful executions	No. of grasps labelled as successful by MLP
Part2_8	20/21	21
Part1_0	17/19	19
Part2_0	26/27	27
Asm5_3	21/24	24

Table 5.2: Verification of our MLP model accuracy on the real UR10 robot

only the grasps that are classified as successful by the MLP models, indicating that they are predicted to be successful grasps. These results demonstrate the accuracy and success of the grasps that are identified as potentially successful by the models. The test grasps are collected by sampling the 6 DOF position of the gripper over the body of the object. This leads to a possibility of a slight bias being introduced in the system. The presence of bias is mitigated by attempting to uniformly/evenly sample the grasps across the object as in Figure 2.8.

Figure 5.5 also shows the distribution of the hand positions for both training and testing samples. The plots indicate that the evaluation of the models for PRT_1_0 and ASM_5_3 may have better reliability due to higher similarity between the training and testing grasp distributions. Another rough observation from the figure is that compared to the testing data, the distribution of the training samples is relatively narrow compared to other objects, e.g. for object PRT_2_0, which may need further quantitative analysis to evaluate its influence on the model performance. Having a diverse distribution of the training data allows the model to learn from examples of a wide variety, thus enhancing its ability to generalize and avoid over-fitting. Therefore, improvement is possible by increasing the size and diversity of the training datasets.

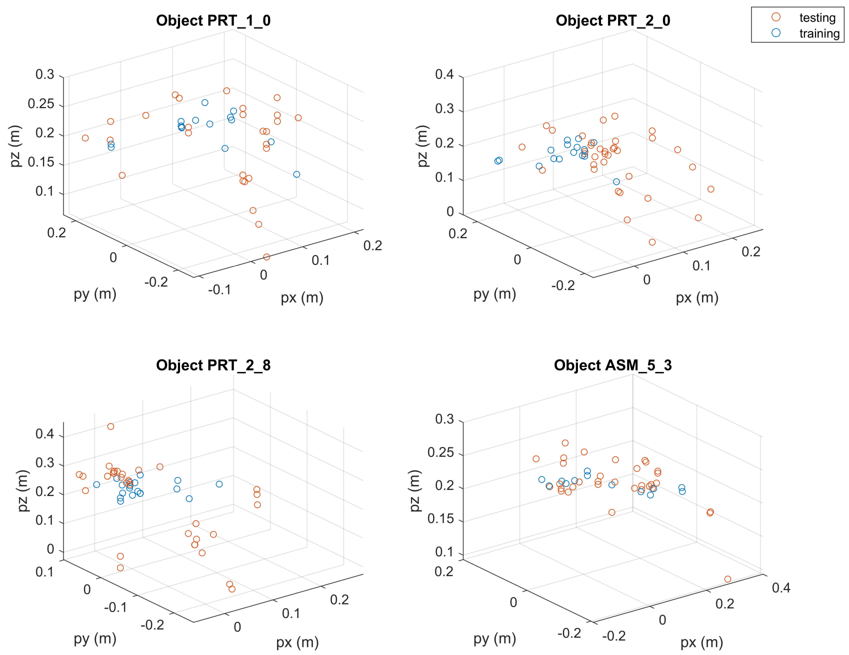


Figure 5.5: The distribution of the grasp positions of training and testing data for each object

6
Discussion

In this chapter, we delve into the factors and intuitions that led to the sub optimal accuracy when executing GPD grasps on Volvo's industrial objects. Moreover, a preliminary analysis of the grasping points on the provided industrial objects is also presented.

6.1 Implementation of GPD on industrial objects

During the execution of PyBullet simulations and real-world tests on the UR10, we encounter instances where grasps classified as successful by the Grasp Pose Detector (GPD) system fail to successfully pick up Volvo industrial objects. An analysis of these failed grasps reveals that the critical factor contributing to their failure is the actual grasping points on the object.

GPD relies on classifying grasps as successful based on their antipodal nature. While this criterion is generally effective, we observe that relying solely on antipodality might be insufficient in certain scenarios. Figure 4.3 illustrates an example where the points indicated by the brown arrows are considered antipodal and deemed successful by GPD. However, during the execution phase, these grasps fail due to the specific shape of the gripper fingertips and the insufficient space for a robust grip. Possible improvements in grasp classification are discussed in the Future Work section of the following chapter.

6.2 Analysis of the grasping points of the industrial objects

During the process of creating the dataset, we observe that the selection of the gripper and the grasping region significantly impacts the success of the grasp. These observations are made by manually maneuvering the gripper around the object and lifting it into the air (as depicted in Figure 6.1), followed by the classification of the grasps as either successful or unsuccessful. This is done to analyze and identify the regions of the object that would lead to stable grasps. The aim is to learn from the minimal data and partially rely on human experience and knowledge in finding good grasps.

A consistent observation across all four objects in the dataset reveals that the success of the grasp is also contingent on the object's positioning on the table. To illustrate, consider the case of the pipe (Asm 5_4), where the grasp quality is influenced by the pipe's orientation. The lower portion of the pipe rests flat against the table, while the upper half is slightly elevated above the table surface by a few millimeters. Furthermore, the curved segments along the pipe's lower half closely conform to the table surface. These factors, encompassing object pose and shape, distinctly impact the grasp quality.

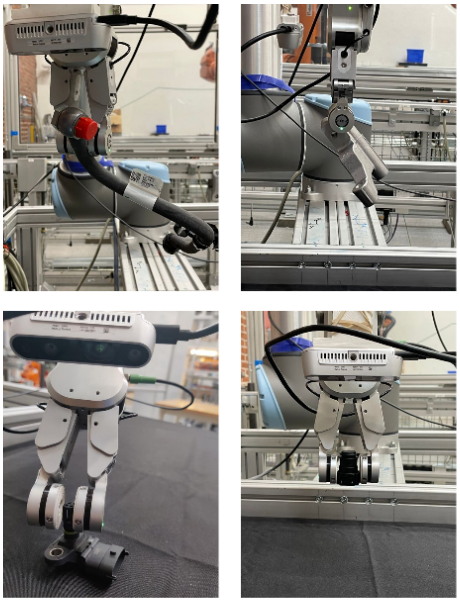


Figure 6.1: Examples of stable grasps in the training dataset

It is worth noting that the behavior of the RG6 gripper and the RG2-FT gripper exhibits a near-identical performance in terms of grasping. This observation is substantiated by conducting successful and unsuccessful grasps from the training dataset using both types of grippers (further elaborated in Appendix B). In order to bolster the verification process and ensure robustness, images of the grasps around the objects, accompanied by their corresponding labels, are additionally incorporated into the dataset.

7
Conclusion and Future scope

In this thesis, we have explored the application of the Grasp Pose Detector (GPD) system [51] on Volvo's industrial objects and developed a multilayer perceptron (MLP) model to classify grasps as successful or unsuccessful. The evaluation of GPD on Volvo's industrial objects highlights the potential for improving the classification performance by considering additional criteria alongside antipodal points. Moreover, it is plausible that the failure in grasping could be attributed to the type of dataset used during GPD's training, which does not encompass industrial objects.

The training and testing datasets are generated by attempting to achieve a balanced distribution of grasps from the industrial objects across the four MLP models. Our findings demonstrate that the MLP model differentiates between successful and unsuccessful grasps and achieves an average validation accuracy of around 80 percent. However, we acknowledge that the performance of the MLP model may vary when confronted with novel objects not included in the training dataset. Further research could be undertaken to enhance its generalization capabilities for unknown objects. The MLP model demonstrates the potential for accurate grasp classification and provides an opportunity to further improve the robustness of the model by experimenting with different grasp generation techniques.

7.1 Future scope

Increasing the size of the dataset size could potentially improve the quality of results and prediction capabilities of the model. Additionally, recent advances in technologies have developed grippers with extra sensors that could complement the research by providing additional information on the object. For example, tactile sensor readings could also provide a deeper insight into the quality of grasps and the relationship between the grasp pose with the overall geometry. Given that the RG2 gripper is outfitted with force-torque sensors on both fingers, exploring the integration of tactile sensor data into the system could be valuable. Although it is difficult to use it in the grasp prediction as the tactile sensor data is not available before grasping, it can be utilized to provide insight into the stability of the grasp.

Obtaining information on the geometric features either from point clouds or the image of the object itself could potentially improve the quality of predicted grasps. Lack of shape awareness [87] leads to grasps that are less accurate and unable to generalize to different objects. Improvement in this area by encoding or extracting geometric information from the point cloud could potentially improve the success rates of successful grasps. Popular works such as [67] include the use of PointNet and PointNet++ which could also potentially be included in future work. Additionally, an increase in the size of the dataset with synthetic data could also be a potential solution towards improving the generalization ability of models as well as grasp predictions. The evaluation of the model can be further improved by carrying out an analysis of the false labels and the testing bias.

Bibliography

[1] Muhammad Usman Khalid, Felix Spenrath, Manuel Mönnig, Marius Moosmann, Richard Bormann, Holger Kunz, and Marco F Huber. Automatic grasp generation for vacuum grippers for random bin picking. In *Advances in Automotive Production Technology–Theory and Application: Stuttgart Conference on Automotive Production (SCAP2020)*, pages 247–255. Springer, 2021.

[2] Adrian Schyja and Bernd Kuhlenkötter. Realistic simulation of industrial bin-picking systems. In *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*, pages 137–142. IEEE, 2015.

[3] Dirk Holz and Sven Behnke. Fast edge-based detection and localization of transport boxes and pallets in rgb-d images for mobile robot bin picking. In *Proceedings of ISR 2016: 47st International Symposium on Robotics*, pages 1–8. VDE, 2016.

[4] OnRobot. 3fg15 three-finger gripper. Retrieved from `https://onrobot.com/en/products/3fg15-three-finger-gripper`.

[5] OnRobot. Rg6 datasheet. `https://onrobot.com/sites/default/files/documents/Datasheet_RG6_v1.1_EN.pdf`, 2021.

[6] Marcus Gualtieri, Andreas Ten Pas, Kate Saenko, and Robert Platt. High precision grasp pose detection in dense clutter. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 598–605. IEEE, 2016.

[7] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *International Conference on Advanced Robotics (ICAR)*, 2015.

[8] Le Duc Hanh and Khuong Thanh Gia Hieu. 3d matching by combining cad model and computer vision for autonomous bin picking. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 15:239–247, 2021.

[9] Lars-Peter Ellekilde and Henrik Gordon Petersen. Motion planning efficient trajectories for industrial bin-picking. *The International Journal of Robotics Research*, 32(9-10):991–1004, 2013.

[10] Adrian Schyja, Alfred Hypki, and Bernd Kuhlenkötter. A modular and extensible framework for real and virtual bin-picking environments. In *2012 IEEE International Conference on Robotics and Automation*, pages 5246–5251. IEEE, 2012.

[11] Ander Iriondo, Elena Lazkano, and Ander Ansuategi. Affordance-based grasping point detection using graph convolutional networks for industrial bin-picking applications. *Sensors*, 21(3):816, 2021.

[12] Matthias Nieuwenhuisen, David Droeschel, Dirk Holz, Jörg Stückler, Alexander Berner, Jun Li, Reinhard Klein, and Sven Behnke. Mobile bin picking with an anthropomorphic service robot. In *2013 IEEE International Conference on Robotics and Automation*, pages 2327–2334. IEEE, 2013.

[13] Gonçalo Leão, Carlos M Costa, Armando Sousa, and Germano Veiga. Detecting and solving tube entanglement in bin picking operations. *Applied Sciences*, 10(7):2264, 2020.

[14] Yukiyasu Domae, Haruhisa Okuda, Yuichi Taguchi, Kazuhiko Sumi, and Takashi Hirai. Fast graspability evaluation on single depth maps for bin picking with general grippers. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1997–2004. IEEE, 2014.

[15] Hao-Yuan Kuo, Hong-Ren Su, Shang-Hong Lai, and Chin-Chia Wu. 3d object detection and pose estimation from depth image for robotic bin picking. In *2014 IEEE international conference on automation science and engineering (CASE)*, pages 1264–1269. IEEE, 2014.

[16] Zhekai Tong, Yu Hin Ng, Chung Hee Kim, Tierui He, and Jungwon Seo. Dig-grasping via direct quasistatic interaction using asymmetric fingers: An approach to effective bin picking. *IEEE Robotics and Automation Letters*, 6(2):3033–3040, 2021.

[17] Ming-Yu Liu, Oncel Tuzel, Ashok Veeraraghavan, Yuichi Taguchi, Tim K Marks, and Rama Chellappa. Fast object localization and pose estimation in heavy clutter for robotic bin picking. *The International Journal of Robotics Research*, 31(8):951–973, 2012.

[18] Dirk Buchholz, Simon Winkelbach, and Friedrich M Wahl. Ransam for industrial bin-picking. In *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, pages 1–6. VDE, 2010.

[19] Carlos Martinez, Remus Boca, Biao Zhang, Heping Chen, and Srinivas Nidamarthi. Automated bin picking system for randomly located industrial parts. In *2015 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, pages 1–6. IEEE, 2015.

[20] Kilian Kleeberger, Christian Landgraf, and Marco F. Huber. Large-scale 6d object pose estimation dataset for industrial bin-picking. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2573–2578, 2019.

[21] Marco Ojer, Xiao Lin, Antonio Tammaro, and Jairo R Sanchez. Pickingdk: A framework for industrial bin-picking applications. *Applied Sciences*, 12(18):9200, 2022.

[22] Yusuke Inagaki, Ryosuke Araki, Takayoshi Yamashita, and Hironobu Fujiyoshi. Detecting layered structures of partially occluded objects for bin picking. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5786–5791. IEEE, 2019.

[23] Ruinian Xu, Fu-Jen Chu, and Patricio A Vela. Gknet: grasp keypoint network for grasp candidates detection. *The International Journal of Robotics Research*, 41(4):361–389, 2022.

[24] Berkeley gq-cnn training datasets. `https://berkeley.app.box.com/s/6mnb2bzi5zfa7qpwyn7uq5atb7vbztng`.

[25] Cornell University Robot Learning Lab. Learning to grasp. `https://www.kaggle.com/oneoneliu/cornell-grasp`.

[26] Yidan Feng, Biqi Yang, Xianzhi Li, Chi-Wing Fu, Rui Cao, Kai Chen, Qi Dou, Mingqiang Wei, Yun-Hui Liu, and Pheng-Ann Heng. Towards robust part-aware instance segmentation for industrial bin picking. *arXiv preprint arXiv:2104.02195*, 2021.

[27] Yajun Xu, Shogo Arai, Diyi Liu, Fangzhou Lin, and Kazuhiro Kosuge. FPCC: Fast point cloud clustering-based instance segmentation for industrial bin-picking. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[28] Kilian Kleeberger, Christian Landgraf, and Marco F. Huber. Large-scale 6d object pose estimation dataset for industrial bin-picking. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2573–2578, 2019.

[29] Andy Zeng, Shuran Song, Kuan-Ting Yu, Elliott Donlon, Francois R. Hogan, Maria Bauza, Daolin Ma, Orion Taylor, Melody Liu, Eudald Romo, Nima Fazeli, Ferran Alet, Nikhil ChavanDafle, Rachel Holladay, Isabella Morona, Prem Qu Nair, Druck Green, Ian Taylor, Weber Liu, Thomas Funkhouser, and Alberto Rodriguez. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. *The International Journal of Robotics Research*, 2019.

[30] Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Tossingbot: Learning to throw arbitrary objects with residual physics. `http://tossingbot.cs.princeton.edu`, Year.

[31] Zhixuan Liu, Zibo Chen, Shangjin Xie, and Wei–Shi Zheng. Transgrasp: A multi-scale hierarchical point transformer for 7-dof grasp detection. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1533–1539, 2022.

[32] Chaozheng Wu, Jian Chen, Qiaoyu Cao, Jianchi Zhang, Yunxin Tai, Lin Sun, and Kui Jia. Grasp proposal networks: An end-to-end solution for visual learning of robotic grasps. *Advances in Neural Information Processing Systems*, 33:13174–13184, 2020.

[33] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, 2015.

[34] Jean Ponce, Daria Stam, and Bertrand Faverjon. On computing two-finger force-closure grasps of curved 2d objects. *The International Journal of Robotics Research*, 12(3):263, 1993.

[35] Viet Nguyen. Constructing stable force-closure grasps. In *ACM Fall Joint Computer Conference*, 1986.

[36] Marcos Alonso, Alberto Izaguirre, and Manuel Graña. Current research trends in robot grasping and bin picking. In *International Joint Conference SOCO'18-CISIS'18-ICEUTE'18: San Sebastián, Spain, June 6-8, 2018 Proceedings 13*, pages 367–376. Springer, 2019.

[37] Jane Shi and Gurdayal S Koonjul. Real-time grasping planning for robotic bin-picking and kitting applications. *IEEE Transactions on Automation Science*

and Engineering, 14(2):809–819, 2017.

[38] Ping Jiang, Yoshiyuki Ishihara, Nobukatsu Sugiyama, Junji Oaki, Seiji Tokura, Atsushi Sugahara, and Akihito Ogawa. Depth image–based deep learning of grasp planning for textureless planar-faced objects in vision-guided robotic bin-picking. *Sensors*, 20(3):706, 2020.

[39] Kai-Tai Song, Cheng-Hei Wu, and Sin-Yi Jiang. Cad-based pose estimation design for random bin picking using a rgb-d camera. *Journal of Intelligent & Robotic Systems*, 87:455–470, 2017.

[40] Cheng-Hei Wu, Sin-Yi Jiang, and Kai-Tai Song. Cad-based pose estimation for random bin-picking of multiple objects using a rgb-d camera. In *2015 15th International Conference on Control, Automation and Systems (ICCAS)*, pages 1645–1649. IEEE, 2015.

[41] Minghao Gou, Hao-Shu Fang, Zhanda Zhu, Sheng Xu, Chenxi Wang, and Cewu Lu. Rgbmatters: Learning 7-dof grasp poses on monocular rgbd images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[42] Jeffrey Mahler and Ken Goldberg. Learning deep policies for robot bin picking by simulating robust grasping sequences. In *Conference on robot learning*, pages 515–524. PMLR, 2017.

[43] Jinho Park, Martin BG Jun, and Huitaek Yun. Development of robotic bin picking platform with cluttered objects using human guidance and convolutional neural network (cnn). *Journal of Manufacturing Systems*, 63:539–549, 2022.

[44] Chungang Zhuang, Zhe Wang, Heng Zhao, and Han Ding. Semantic part segmentation method based 3d object pose estimation with rgb-d images for bin-picking. 2022.

[45] Jun Yang, Yizhou Gao, Dong Li, and Steven L Waslander. Robi: A multi-view dataset for reflective objects in robotic bin-picking. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9788–9795. IEEE, 2021.

[46] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014.

[47] Diyi Liu, Shogo Arai, Jiaqi Miao, Jun Kinugawa, Zhao Wang, and Kazuhiro Kosuge. Point pair feature-based pose estimation with multiple edge appearance models (ppf-meam) for robotic bin picking. *Sensors*, 18(8):2719, 2018.

[48] Lu Wang, Yu Xiang, and Dieter Fox. Goal-auxiliary actor-critic for 6d robotic grasping with point clouds. *arXiv preprint arXiv:2106.03821*, 2021.

[49] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5099–5108. Curran Associates Inc., 2017.

[50] Zhikai Dong, Sicheng Liu, Tao Zhou, Hui Cheng, Long Zeng, Xingyao Yu, and Houde Liu. Ppr-net: point-wise pose regression network for instance segmentation and 6d pose estimation in bin-picking scenarios. In *2019 IEEE/RSJ In-*

ternational Conference on Intelligent Robots and Systems (IROS), pages 1773–1780. IEEE, 2019.

[51] Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. Grasp pose detection in point clouds. *The International Journal of Robotics Research*, 36(13-14):1455–1473, October 2017.

[52] Berkeley Robot Learning Lab. BigBird: Large-Scale 3D Object Shape Understanding. `https://rll.berkeley.edu/bigbird/`.

[53] Bowen Wen, Wenzhao Lian, Kostas Bekris, and Stefan Schaal. Catgrasp: Learning category-level task-relevant grasping in clutter from simulation. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[54] Binglei Zhao, Hanbo Zhang, Xuguang Lan, Haoyu Wang, Zhiqiang Tian, and Nanning Zheng. Regnet: Region-based grasp network for end-to-end grasp detection in point clouds. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.

[55] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5026–5033. IEEE, 2012.

[56] C. Rennie, R. Shome, K. E. Bekris, and A. F. De Souza. A dataset for improved RGBD-based object detection and pose estimation for warehouse pick-and-place. *IEEE Robotics and Automation Letters (RA-L)*, 1(2), 2016.

[57] T. Hodaň, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, and X. Zabulis. T-LESS: An rgb-d dataset for 6d pose estimation of texture-less objects. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.

[58] Xinghao Zhu, Lingfeng Sun, Yongxiang Fan, and Masayoshi Tomizuka. 6-dof contrastive grasp proposal network. *arXiv preprint arXiv:2103.15995*, 2021.

[59] Adithyavairavan Murali, Arsalan Mousavian, Clemens Eppner, Chris Paxton, and Dieter Fox. 6-dof grasping for target-driven object manipulation in clutter. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[60] Universal Robots. Ur10 robot. Retrieved from `https://www.universal-robots.com/products/ur10-robot/`.

[61] OnRobot. Rg6 gripper. Retrieved from `https://onrobot.com/en/products/rg6-gripper`.

[62] ROS Wiki. Ros noetic ninjemys/installation. Retrieved from `http://wiki.ros.org/noetic/Installation`.

[63] MoveIt! Moveit! - the motion planning framework. Retrieved from `https://moveit.ros.org/`.

[64] Patrik Fager, Stefano Rossi, Robin Hanson, Lars Medbo, Omkar Salunkhe, Mats I. Johansson, and Åsa Fast-Berglund. Gripper types and components in robotic bin picking. In *IFIP Advances in Information and Communication Technology*, volume 591, 2020.

[65] Cheng-Hei Wu, Sin-Yi Jiang, and Kai-Tai Song. Cad-based pose estimation for random bin-picking of multiple objects using a rgb-d camera. In *2015 15th International Conference on Control, Automation and Systems (ICCAS)*, pages 1645–1649, 2015.

[66] Jongwook Kim, Hyungmin Kim, and Jong-Il Park. An analysis of factors affecting point cloud registration for bin picking. In *2020 International Conference on Electronics, Information, and Communication (ICEIC)*, pages 1–4, 2020.

[67] Hongzhuo Liang, Xiaojian Ma, Shuang Li, Michael Görner, Song Tang, Bin Fang, Fuchun Sun, and Jianwei Zhang. Pointnetgpd: Detecting grasp configurations from point sets. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[68] ChalmersDextrousRobotics. Ur10robotenv. Retrieved from `https://github.com/ros-planning/moveit_calibration`, Year.

[69] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1710.01330*, 2017.

[70] Zhenhua Zhou, Md Saad Rahman, Maria Aparício, and Eric P Xing. Multi-view knowledge distillation for multi-classification. *arXiv preprint arXiv:1903.11239*, 2019.

[71] Wei-Cheng Chang, Chi-Jung Lu, Han Chen, and Cheng-Yuan Chiang. Deep-learning-based fruit detection and recognition using an autonomous unmanned aerial vehicle. *Sensors*, 21(3):816, 2021.

[72] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. 2015.

[73] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. 2019.

[74] Siddarth Gupta, Jaemin Lee, and Siddhartha Srinivasa. Dexterous manipulation of objects in clutter using multi-contact affordance estimation. *arXiv preprint arXiv:2211.04895*, 2022.

[75] S. N. Mohd Isa, S. A. Abdul Shukor, N. A. Rahim, I. Maarof, Z. R. Yahya, A. Zakaria, A. H. Abdullah, and R. Wong. Point cloud data segmentation using ransac and localization. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2022.

[76] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. Meshlab. Retrieved from `https://www.meshlab.net/`, 2008.

[77] PyBullet Developers. Pybullet - open source physics engine, 2023.

[78] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[79] Arjun Singh, James Sha, Karthik S Narayan, Tudor Achim, and Pieter Abbeel. Bigbird: A large-scale 3d database of object instances. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 509–516. IEEE, 2014.

[80] Jianjun Yu, Miaoqiang Zhou, Daoxiong Gong, Naigong Yu, and Ruihua Zhu. Pointnet++gpd: 6-dof grasping pose detection method based on object point cloud. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1234–1245, 2021.

[81] How to configure the number of layers and nodes in a neural network.

[82] Berk Calli, Arjun Singh, Avinash Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. Ycb object and model set: Towards common benchmarks for manipulation research. Retrieved from `https://www.ycbbenchmarks.com/`, 2015.

[83] Pytorch documentation: torch.optim.adam. `https://pytorch.org/docs/stable/generated/torch.optim.Adam.html`.

[84] Pytorch documentation: torch.optim.adagrad. `https://pytorch.org/docs/stable/generated/torch.optim.Adagrad.html`.

[85] Pytorch documentation: torch.optim.sgd. `https://pytorch.org/docs/stable/generated/torch.optim.SGD.html`.

[86] Andreas ten Pas, Colin Keil, and Robert Platt. Efficient and accurate candidate generation for grasp pose detection in se (3). In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5725–5732. IEEE, 2021.

[87] Binglei Zhao, Hanbo Zhang, Xuguang Lan, Haoyu Wang, Zhiqiang Tian, and Nanning Zheng. Regnet: Region-based grasp network for end-to-end grasp detection in point clouds. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

A
Appendix 1

During the investigation of the quality of the grasps generated on point clouds, care is taken to set the workspace to focus only on the object and avoid the table. Additionally, the table in the background can also be edited out by manually cropping it out or by segmentation using RANSAC.

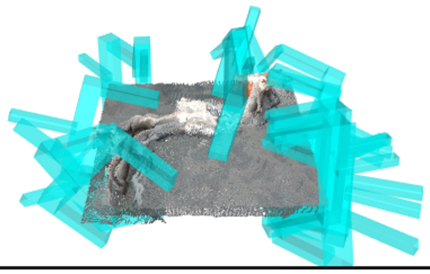


Figure A.1: Grasps before removing the table

Figure A.2: Part 1_0 grasps from the manual dataset.
From left to right, good grasp, good grasp and bad grasp

B

Appendix 2

B.1 Replacement of the RG2 gripper with RG6

The working range of the RG2 and RG6 grippers is visualized in Figure B.1. During the course of the thesis, a switch from the RG2 gripper to the RG6 gripper was necessitated due to technical issues. While the training phase utilized the RG2 gripper, the testing phase of the MLP model was conducted using the RG6 gripper.

To ensure the validity and accuracy of the results obtained from MLP testing with the RG6 gripper, we have thoroughly validated our decision to employ a translation of 0.04. In the following section, we describe our methodology for determining the value of 0.04 translation. Additionally, in the physical experimentation section, we provide a comprehensive explanation of the translation process, supported by relevant figures that enhance the understanding of the concept.

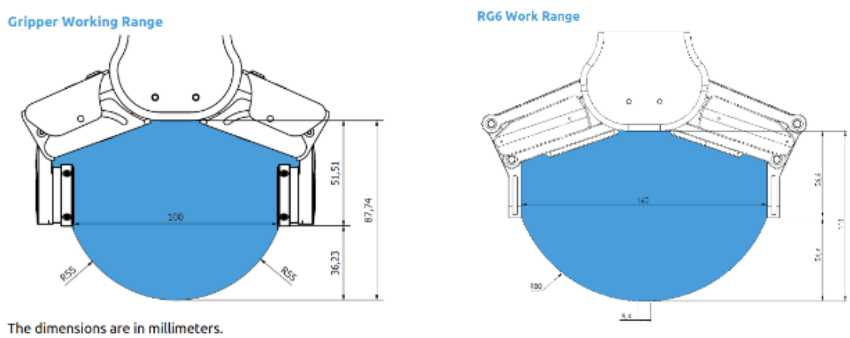


Figure B.1: Left: RG2 gripper range. Right: RG6 gripper range

B.1.1 Mathematical verification

To determine the difference in translation required for the RG6 gripper to behave similarly to the RG2 gripper, the quantity of the translation in the z-direction in the hand frame has been investigated.

To yield the variation range of translation difference, a two-step procedure was followed. the In the first step, the fingers of both the RG2 and RG6 grippers were completely closed, resulting in a width between the fingers (L) of 0mm. In the second step, the gripper width of the RG6 gripper was set to 100mm using the OnRobot tablet. This value was chosen because it corresponds to the maximum width of the RG2 gripper. For both steps, the complete length (Z) was measured, which refers to the distance from the wrist of the robot (beginning of the gripper) to the fingertips. The difference in Z between the two steps represents the translation required for the RG6 gripper to align with the behavior of the RG2 gripper.

$$\begin{aligned}
\delta Z &= Z_{rg6} - Z_{rg2} \\
\delta Z_{min} &= Z_{rg6} - Z_{rg2} \\
\delta Z_{min} &= 262 - 219 \\
&= 43mm \\
\delta Z_{max} &= Z_{rg6} - Z_{rg2} \\
&= 250 - 182 \\
&= 68mm
\end{aligned} \tag{B.1}$$

$$43mm <= \delta Z_{min} <= 68mm \tag{B.2}$$

Table B.1: Abbreviations used in the formula

Abbr.	Meaning
Z	distance from the gripper base to the top of the object at closing point
δ	difference in the distance in the Z from RG2 and RG6

The selection of the translation value for the RG6 gripper execution was based on a range of values tested, ranging from 0.04 to 0.06 As seen in equation B.2. Among these values, 0.04 was chosen as it proved to be the most suitable for all the objects considered in the thesis. Considering that the objects being grasped are small in size, they are more sensitive to variations in the grasp execution. Therefore, a larger variation of 0.06 would have been too significant for small objects such as Part 2_0.

To verify the appropriateness of the chosen translation value, a thorough verification process has been conducted, which is described in the following section.

B.1.2 Physical verification

In the training dataset, we gathered the position and orientation information of each grasp pose, along with attached pictures showcasing the outcome of the grasp. As part of the verification process, we conducted a re-execution of the grasp poses using the RG6 gripper with a translation of 0.04 in the z-direction. The objective was to validate whether a translation of 0.04 was an appropriate choice. During the verification process, the translations were applied to the grasps in the x, y, and z directions, and the resulting grasps were compared to the original ones. The purpose was to demonstrate that despite the translation, the grasps remained highly similar in terms of their quality and effectiveness.

The translation in the x, y, and z directions for the grasps was set to be 0.04, depending on the grasp pose. Specifically, if the grasp approach was top-down along the z-axis, a translation in the z-direction was applied. Similarly, if the grasp approach was from the right side (as shown in Figure B.6), a translation along the y-axis was performed. Likewise, if the grasp approach was from the front or back, a translation along the x-axis was made. Since the majority of the grasps were performed along the top surface of the objects or parallel to the table, the translation process was relatively straightforward. The value of 0.04 was chosen to account for the variation in the closing region of the gripper fingers as they extended to grasp around the object.

Test (a): Figure B.6, B.3, B.4, and B.5 illustrate the results. The top row represents the grasp executed using the RG2 gripper during the manual data collection process. The second row shows the execution of the same grasp pose using the RG6 gripper, but with a translation of 0.04 in the z-direction. This translation in the z-direction is typically applied to prevent the gripper fingers from hitting the table due to its slightly larger dimensions.

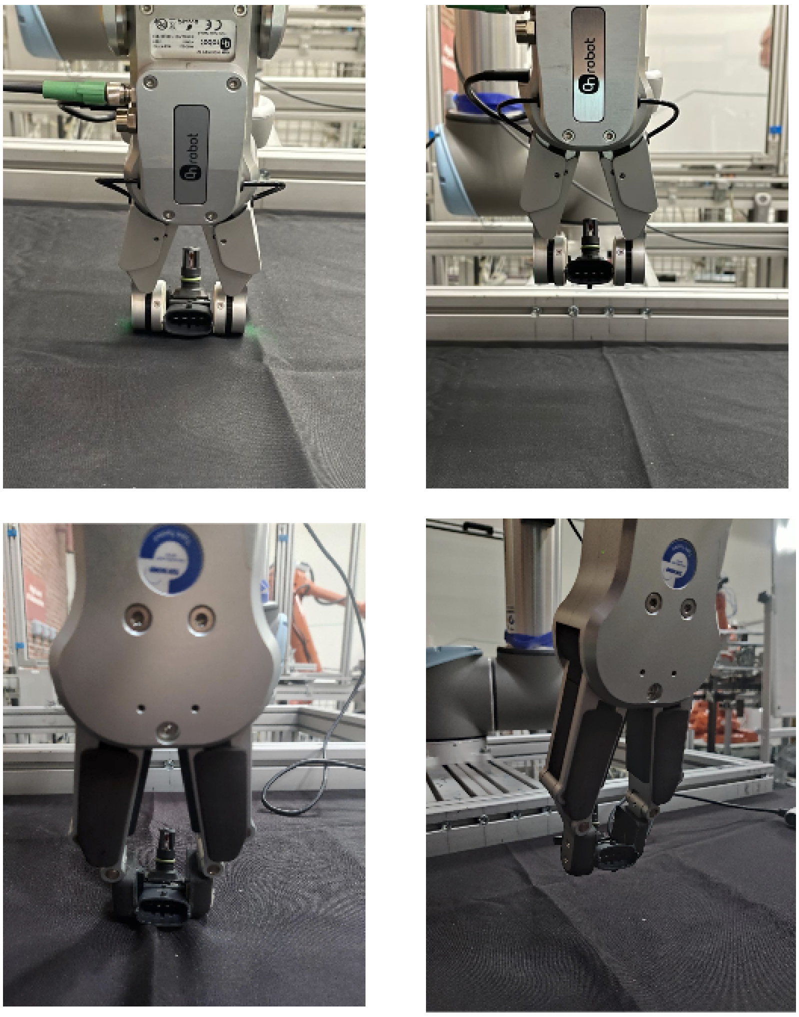


Figure B.2: A translation of 0.04 in z direction yields the same grasp pose and outcome as that of RG2

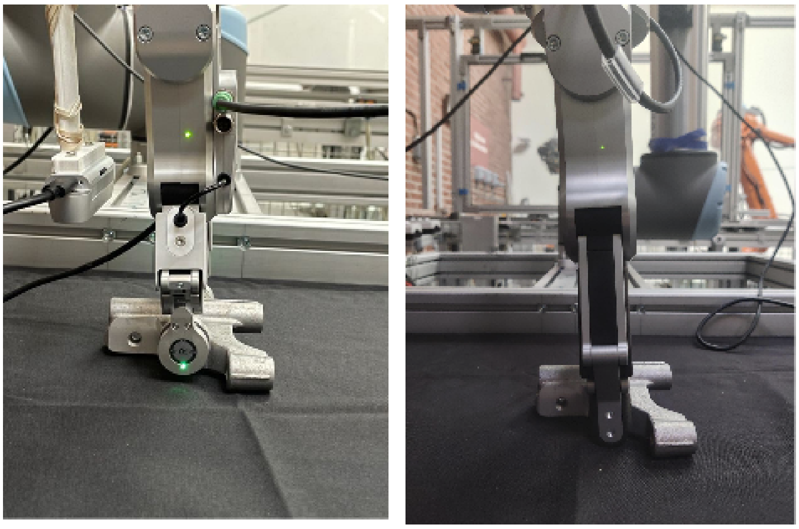


Figure B.3: A translation of 0.04 in z direction yields the same grasp pose and outcome as that of RG2

Figure B.4: A translation of 0.04 in z direction yields the same grasp pose and outcome as that of RG2

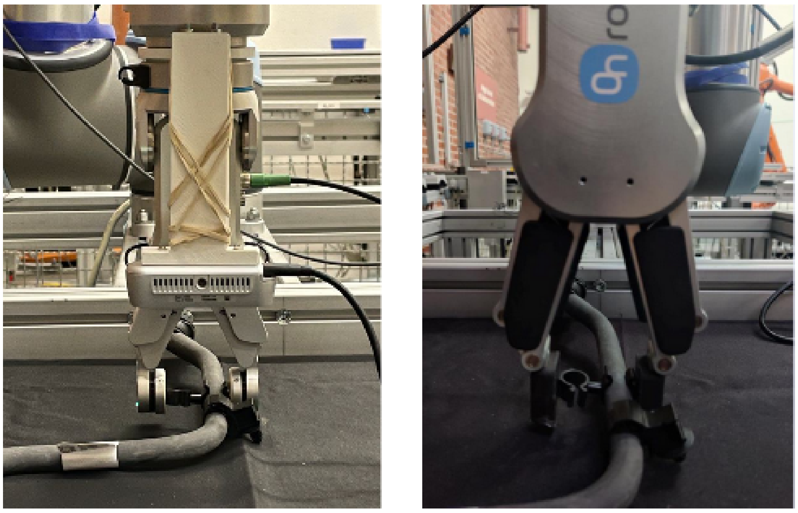


Figure B.5: A translation of 0.04 in z direction yields the same grasp pose and outcome as that of RG2

Test (b) involved examining a grasp pose from the training dataset that approached the object from the y direction of the world frame. We tested whether applying a translation of 0.04 in the y direction would result in a similar outcome. The results of this test are depicted in Figure B.6. By comparing the top row (grasp executed by the RG2 gripper) with the second row (grasp executed by the RG6 gripper with a translation of 0.04 in the y direction), we can observe that the outcomes are consistent and effective.

Figure B.6: A translation of 0.04 in y direction yields the same grasp pose and outcome as that of RG2

Test (c): Due to limitations in the robot trajectory planning, there are fewer grasps from the front or back sides of the object, along the x axis. The IK solver and collision avoidance constraints restrict the robot's movement in those directions. However, we managed to test a grasp pose that, although not perfectly parallel to the table, closely approximates the desired orientation. The outcomes of this test can be observed in Figure B.7.

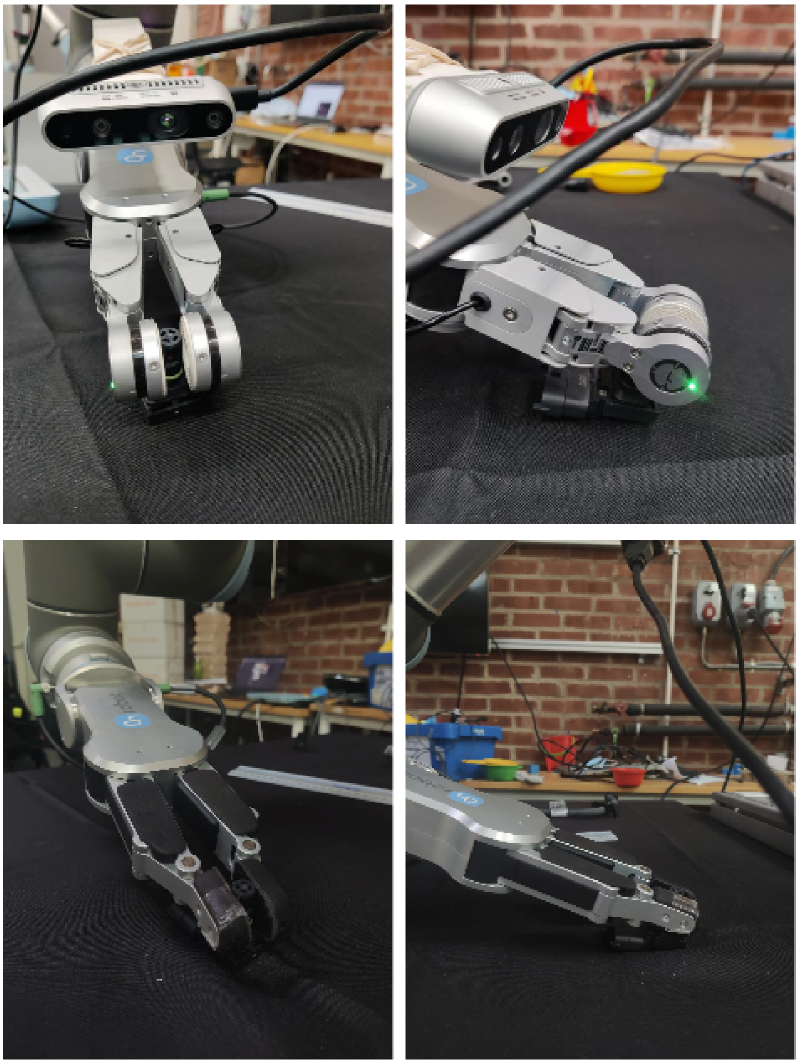


Figure B.7: A translation of 0.04 in x direction yields the same grasp pose and outcome as that of RG2

By performing these tests, it was confirmed that the chosen translation of 0.04 adequately accounted for the variations in the closing region of the gripper fingers and ensured successful grasping. The grasp outcomes were consistent and reliable in all three cases, validating the effectiveness of the chosen translation value in capturing the necessary adjustments for successful grasping and verifying the accuracy of the MLP during testing.

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY