Handle Grasp Improvement using Weighted-Maximum Likelihood Estimation

Anirudh Narasimamurthy Jayasimha Alex Mitrevski Paul G. Ploger

Abstract—**Grasping and manipulating handles is an important task for robots to explore new areas of its environment. We propose a novel method of finding the best grasp position on the handle using perception and use this to robustly grasp and manipulate the handle. We use Weighted-Maximum Likelihood Estimation to model the underlying normal distribution that improves the chances of as successful grasp.**

This algorithm was tested for opening vertical and horizontal fixed handles in a real world environment using Toyota HSR robot. Our initial experimental evaluation shows that this method can help the robot to autonomously grasp the handle robustly and perform manipulation sequence on the handle.

***Index Terms*—Maximum Likelihood Estimation, Handle Manipulation**

I. Introduction

The task of manipulating articulated handles involves a complex pipeline of understanding the object structure and use it for applying forces on it. Handle is a common type of articulated object that is commonly found in both domestic and industrial environments. The handles are usually attached to an underlying object like a door, cabinet drawer etc to manipulate it.

In unstructured environments like domestic environments, hardcoded priors and maps are not enough for manipulation tasks. For the robot to work in a wide variety of domestic environments, a tight coupling of perception and manipulation must be present for the robot to perform autonomous manipulation robustly. Perception tasks that work as a prior for manipulation are more complex than general object detection tasks. Perception algorithms in these cases must also be able to find the various physical properties of the object and the environment like the object shape, size, pose and position of the object with respect to the robot.

This paper provides a novel method of coupling the perception and manipulation subtasks for the particular task of manipulating handles. The main contribution of this paper are as following a) A general pipeline coupling perception and manipulation subtasks for grasping and manipulating different types of handles. b) A learning based method for finding the best possible grasping position for successfully manipulating the handle. c) Developing and testing the pipeline on a Toyota HSR robot in real world test environment.

Our successful experiments show that learning based methods will be very helpful in teaching the robots generalizable skills and improve the robustness of the skills over time.

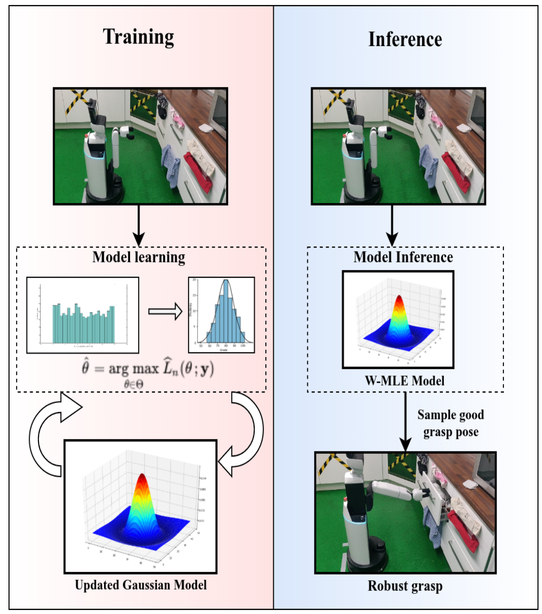

Fig. 1. Pipeline showing the training and inference steps for a robust grasp pose generation using W-MLE model.

II. Related Work

Most of the previous methods have tried to solve each part of the pipeline by using a combination of various algorithms. Although these methods have tried to solve the problem with various degree of success, there is room for a more robust, generalizable and safer method. In this paper we have confined ourselves to methods that were specifically used for some subtask of the handle manipulation task.

A. Perception

Modern robots have a large number of cameras to capture the visual data of its environment and learn important features about it. Reliable perception is very important in robotics as further operations in the pipeline will be dependent on the success and the quality of the subtask. For the task of manipulation of handles, the perception subtask involves detection of handles in the robot camera frame and the handle pose along with its position with respect to the robot.

Object detection is one of the most important tasks in computer vision. The study of object detection methods can be broadly divided into two time periods : *"traditional object detection period (before 2014)" and "deep learning based* *detection period (after 2014)"* [1] Classical methods like HoG and Haar features were extensively used by [2] [3] for detecting handles in the camera frame. To improve the detection accuracy of these methods spatial [4] and textural cues like the handle position on the door, color difference between the door and the handle were used. [5] use modern CNN methods to obtain highly generalizable state of the art results for detecting various classes of handles and doors in the camera by using a large corpus of annotated 2D images. [6] [7] [8] [9] used the pointcloud data obtained by the RGB-D camera or Lidar to detect doors planes. The handles are then assumed to be the point-cloud cluster that lies beyond the door plane. Point-cloud techniques are very sensitive to environment lighting and object properties and don't generalize as well as algorithms that work on RGB cameras. Some methods like [3] use a multimodal approach by using the bounding box of the handle detected by the 2D camera as a ROI for the pointcloud. This helps in mitigating the problems associated each mode of the data.

B. Manipulation

Based on the taxonomy of the manipulation tasks given by [10], handle manipulation can be considered as a can be considered as contact based, prehensile task of medium precision and dexterity. Handle manipulation involves grasping the handle tightly such that it doesn't slip from the robot end effector and then perform some action like pulling for fixed handle and twisting for lever type handles. Manipulation algorithms can broadly be classified into various methods based on the characteristics like the type of parameters it improves and how it encodes them etc. Previous literature for handle manipulation task can be classified as Control based, reinforcement learning, graph optimization, fuzzy logic and probabilistic based approaches.

Non-compliant control based approaches like [11] [12] [13] involves hardcoded trajectories that are used to manipulate different classes of handles without taking into consideration external forces on the robot joints due to unexpected external forces or model inaccuracies. *Compliance* based methods are designed to keep the internal forces within a reasonable limit during manipulation. Learning based methods like *Reinforcement Learning* rely on finding the best possible trajectory for manipulation without being explicitly designed for it. Policy search methods use reinforcement learning for determining the best set of parameters for an underlying policy (DMP,RBFs etc) to complete the task. Deep reinforcement learning methods like [14] use artificial neural networks as the underlying policy for learning complex task policies. But these learning based methods require very high number of rollouts to converge to a usable policy and are highly sample inefficient. To make collection of rollouts for RL tasks practical many simulation environments like DoorGym [15] are used. These simulated environments can be used to train a model which can be then transferred to a real robot [1]. To reduce the state and action space when using RL, [16] use a structured search approach by using an the underlying physical constraints of the robot controller. *Probabilistic methods* includes methods in which the parameters are encoded in the form of a probability distribution. Few novel methods like *Graph based methods* [17] involves encoding the states and actions of the robot as a graph and use graph optimization to find the optimal path to imitate a trajectory. *Fuzzy logic* method like [18] use a fuzzifier to convert the binary decision of twisting the handle into a fuzzy decision based on the force sensor values and the rotation angle of the handle.

[1]Transfer of model from simulated environment to a real robot is called as Sim to Reality transfer

III. Pipeline Development

The general pipeline for manipulating handles can be logically divided into two subtasks 1) Perception 2) Manipulation. Perception subtask deals with classifying the type of handle in the robot camera frame and finding the 3D-bounding box around it. Manipulation subtask involves using the bounding-box from the perception for finding the best grasping position on the handle and perform a pre-planned trajectory based on the kinematic model of the classified handle.

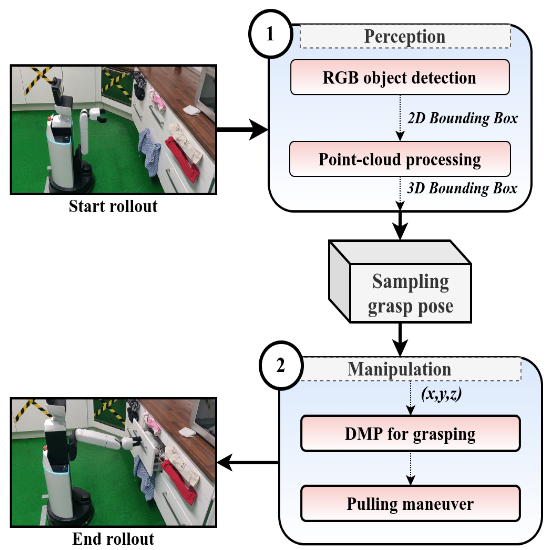

Fig. 2. The complete pipeline for solving the task of manipulation is divided into logical two subtasks 1) Perception and 2) Manipulation. The grasp pose sampling acts as a coupling between the two subtasks.

Few constraints are set when performing the experiment on a real world robot. This paper will consider only the common fixed type handles found in domestic environments when performing experiments. We assume that the handle will always be visible in the robot camera frame and the handle is also within the grasping distance of the robot end effector. We also don't consider any locking mechanism on the drawer or the door. For each rollout during training the robot starts off at a fixed initial position that is directly in-front of the fridge or handle door. While performing inference on the trained model we can place the robot nearly parallel to the door but making sure that the handle is visible within the robot camera frame.

A. Perception

To solve the task of finding the handle in the camera frame we will use a CNN based object detector. Compared to classical object detection methods like template matching, modern learning based methods based on neural networks are very robust to different environment conditions and can generalize better on objects of same category. For finding the handles in the camera frame of the robot, a FR-CNN [19] based pretrained model is used by us. Since a high quality bounding box is required for further processing and grasping, a two stage detector is preferred compared to single stage and faster detectors like SSD [20] or YOLO. To improve the detection on the handles, a custom dataset of different classes of handles (fixed , lever and round handles) found in the robot environment are collected using the robot camera. A 0.8:0.2 testing/training dataset split is used for performing transfer learning on the model using Tensorflow object detection API [2]. On the test split of dataset we got a mAP of 74%.

Fig. 3. Results of the CNN based object detector tested on different classes of handles.

The 2D bounding box detected by the handle detector is used as a region of interest on the RGB-D data obtained by the Xtion camera of the Toyota HSR for finding the 3D bounding box $\mathcal{B}$ around the handle. This bounding box will be used in the further steps for sampling grasp position for training the MLE model and subsequent inference on the model.

B. Manipulation

For training the MLE model and learning the best grasp position on the handle we collect 50 rollouts each for the drawer fixed horizontal handle and a fridge door fixed vertical handle. The position to grasp is sampled from a uniform distribution within the limits of the 3D bounding box around the handle found using the perception subtask. We manually mark the success or failure of each of the performed grasp for each of the rollout. The trained model is then used to perform similar actions again for 50 rollouts and the success rates are compared between the learnt and the untrained models.

[2]https://github.com/tensorflow/models/tree/master/research/object$_{d}etection$

a) Weighted - Maximum Likelihood Estimation: We try to model the underlying distribution of the successful grasps on the handle as a gaussian distribution. We use a method referred to as Weighted - Maximum Likelihood Estimation for finding the best distribution parameters that fits our observed successful data points. Importance weights are given to each of the data-points in our training dataset based on the success of grasp and handle manipulation action. The importance weights are distributed such that the grasp positions during the training which lead to a successful handle grasp and pulling action on the drawer or door are the highest and if only grasp is successful the weightage is lower. A simple extension of the ”frequency weighting” is used, where we add duplicates of the data point to the dataset during training to increase weightage of the data points.

Algorithm 1 Weighted Maximum Likelihood Estimation

Data: List of grasp position coordinates, $y_i \in \mathcal{Y}$ where $y_i = \{x, y, z\}$
Data: Successful samples in $\mathcal{Y}$ given by $\mathcal{S}^+$
Data: Weighted list of samples $\overline{\mathcal{Y}}$
Result: The parameters of gaussian distribution ($\mathcal{N}$), μ and Σ
for *Each sample* $y_i \in \mathcal{Y}$ **do**
  if y_i *leads to a successful grasp or successful pull* **then**
    if $y_i in \mathcal{Y}^+$ *leads to only a successful grasp* **then**
      Add 3 instances of y_i to $\overline{\mathcal{Y}}$;
    else
      Add 5 instances of y_i to $\overline{\mathcal{Y}}$;
    end
  else
    Add 1 instance of y_i to $\overline{\mathcal{Y}}$.
  end
end
Use the $\overline{\mathcal{Y}}$ to perform MLE and find parameters of the normal distribution $\mathcal{N}$

Let y_n be a single sample in the collected training set $\mathcal{Y}$. A single grasp sample is denoted by its 3D coordinate (x,y,z) wrt. the robot base.

$$y_n = \{x, y, z\} \; \forall \; y_n \in \mathcal{Y} \tag{1}$$

Let $\mathcal{Y}^+$ denote all the grasp positions that are successful and $\mathcal{Y}^-$ denote the unsuccessful grasps.

As per the problem statement our goal is to find the set of parameters $\hat{\theta} \in \Theta$ such that we increase the likelihood of sampling successful grasp poses.

$$\hat{\theta} = arg\,max_{\theta \in \Theta} \mathcal{L}(\theta, \mathcal{Y}^+) \tag{2}$$

Since we assume the underlying distribution leading to successful grasp position is a normal distribution the parameters θ learnt by the MLE are the mean and covariance of the 3D gaussian (σ, Σ).

We have given the weightage for a successful grasp along with a successful pull action for the fixed drawer handle is given as 5 and a grasp point that leads to only a successful grasp is given a weightage of 3. The selection of weights are relative such that the most important data point is given the maximum weightage while less important data points are given less weights.

IV. Experiments

We perform two experiments to test and verify the trained W-MLE model for manipulating a) Horizontal fixed handles of a drawer b) Vertical fixed handle of a fridge, using a Toyota HSR robot.

The two major steps for using this algorithm on the real robot are the training step, where we collect rollouts and manually mark the success or failure and use that for training the model. Next step is the inference step where we use the trained model for sampling grasp poses and use them for validating our model.

The steps for collecting the rollouts for subsequent training of the MLE model involves the following:

1) The robot is places in a fixed initial position for every new iteration. The initial position should be within the grasping reach of the handle.
2) We initialize the distribution as a uniform distribution(prior) from which we sample the goal positions for training the model. The limits of the 3D position are fixed within the limits of a bounding box $\mathcal{B}$ such that $x \in \{\alpha_0, \alpha_1\}$, $y \in \{\beta_0, \beta_1\}$ and $z \in \{\theta_0, \theta_1\}$. This 3D cuboid behaves as a restricted region for exploration of the goal positions.
3) For each goal position sampled from the initial distribution, the DMP implementation of the robot is used to accomplish the rollout. For each rollout the success of grasping the handle is manually noted.
4) After performing 50 such rollouts, the initial uniform distribution is updated using the new success and failure data such that the new distribution will lead to a higher chances of success during the subsequent sampling during inference.
5) The training and the update step for the distribution is performed multiple times till we obtain a reasonable Gaussian distribution for sampling the goal position with very high rates of success.

After the model is trained offline on the data collected during training, we repeat a similar set steps to perform the inference manipulation using the new model. This time the robot can be fixed in any position such that the handle is within the view of the camera but parallel to the door/cabinet plane. The trained gaussian distribution is then used for sampling good grasping positions.

V. Results

	Sampling	Successful grasp	Successful pull	Total Success (50 rollouts)	Total Success (percentage)
Drawer handle	Uniform	14	7	14	28%
	Gaussian	30	11	30	60%
Fridge handle	Uniform	13	6	13	26%
	Gaussian	20	7	20	40%

TABLE I
SUCCESS OF GRASP AND PULL TASK BY SAMPLING GOAL POSITION USING A UNIFORM DISTRIBUTION AND A LEARNT GAUSSIAN DISTRIBUTION. THE TOTAL NUMBER OF COLLECTED ROLLOUTS ARE 50 FOR EACH EXPERIMENT.

Since the main goal of the trained model is to robustly grasp the handle, we count a successful grasp as complete success of the rollout. Sampling from a uniform distribution for drawer handle gives only a success rate of 28% while for fridge vertical fixed handle the success rate was a similar 26%. Using the model trained on the data collected by the uniform distribution rollouts, the new gaussian distribution improved the success of grasping. For a horizontal drawer handle the grasp success rate increased to 60% and the for the fridge vertical handle the success rate increased to a modest 40%.

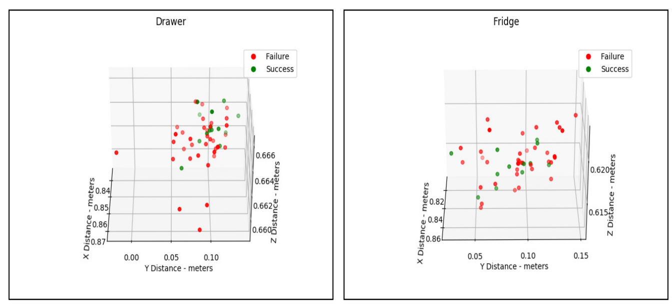

Fig. 5. Sampled grasp poses collected during training which lead to a successful grasp(red) and unsuccessful grasp(green).

It was observed that when sampling poses from uniform distributions the failures in grasping were mainly due to the sampling region $\mathcal{B}^{\mathcal{U}}$ being bigger that the handle dimensions leading to the robot completely missing the handle in many cases.

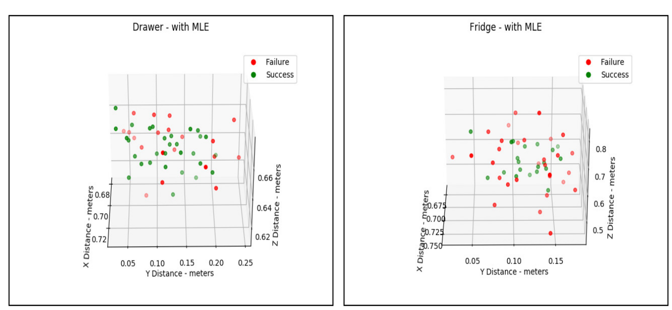

Fig. 6. Grasp poses sampled from the learnt W-MLE model after the first iteration.

The new sampling region $\mathcal{B}^{\mathcal{N}}$ obtained after training the model leads to an increase in the grasp probability within the

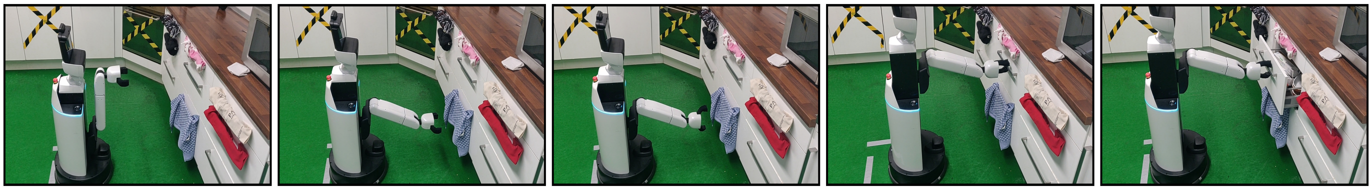

Fig. 4. Complete handle manipulation pipeline running on a Toyota HSR. The robot starts of from detecting the handle and using the appropriate distribution sampling the grasp pose. The DMP implementation on the robot is then used to grasp the handle and pull it.

handle dimensions. The failures in pulling the handle even after grasping the handle correctly were mainly attributed the hardware limitations of the robot and the handles in our test environment. It was observed that the handles were very thin and slippery to for the Toyota HSR grippers.

VI. Limitations

While the task of autonomously detecting handles and manipulating them are highly complex, our paper has tried to provide a new method of solving this task by formulating it as a learning problem. Despite the encouraging results obtained on running the experiment on a real robot, we have got a few areas to potentially further improve the algorithm.

- The perception algorithm detects all the handles that are found in the camera frame of the robot. But the manipulation algorithm is designed to grasp and perform the manipulation action on only one handle during a rollout. If multiple handles are perceived, the manipulation action will be performed on one of the handles randomly. This problem can be corrected by using region of interest based object detection within which we can guarantee that only one handle is visible. The selection of the handle to grasp and manipulate can also be improved by having the user in the loop by creating a GUI based application.
- Different types of forces must be applied based on the type of handle and the underlying kinematic model of door. The pipeline has only been tested for fixed handles and can be further modified and tested for round and lever type handles.

VII. Conclusion

In this paper we describe a new learning based method to teach the robot the best grasping position on the handle to improve the possibility of a successful grasp. The pipeline is designed such that the robot is able to autonomously find the handle in the camera frame and use the trained Weighted-MLE model for performing robust handle grasp and perform the pulling manoeuvre for opening fixed handles.

Our initial experimentation shows that the learned model is able to compensate for small disturbances in the initial position of the robot and improves the overall handle grasping capability when tested on different horizontal and vertical fixed handles.

A promising direction for future work on this is the extension of this pipeline to work as a lifelong learning framework for the robot where the environment can be modified to help the robot detect the success or failure status of the manipulation task. Exploring such research will help in a more robust manipulation of handles and other objects for the next generation of robots.

Acknowledgment

I would like to thank Alex Mitrevski and Ahmed Abdelrahman for giving various insights and ideas. I would also like to thank Prof. Dr. Paul G. Ploger for providing the required hardware for performing the experiments.

References

[1] Z. Zou, Z. Shi, Y. Guo, and J. Ye, “Object detection in 20 years: A survey,” *ArXiv*, vol. abs/1905.05055, 2019.

[2] E. Klingbeil, A. Saxena, and A. Y. Ng, “Learning to open new doors,” *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2751–2757, 2010.

[3] W. Meeussen, M. Wise, S. Glaser, S. Chitta, C. McGann, P. Mihelich, E. Marder-Eppstein, M. Muja, V. Eruhimov, T. Foote, J. M. Hsu, R. B. Rusu, B. Marthi, G. R. Bradski, K. Konolige, B. P. Gerkey, and E. Berger, “Autonomous door opening and plugging in with a personal robot,” *2010 IEEE International Conference on Robotics and Automation*, pp. 729–736, 2010.

[4] T. Rühr, J. Sturm, D. Pangercic, M. Beetz, and D. Cremers, “A generalized framework for opening doors and drawers in kitchen environments,” *2012 IEEE International Conference on Robotics and Automation*, pp. 3852–3858, 2012.

[5] M. Arduengo, C. Torras, and L. Sentis, “A versatile framework for robust and adaptive door operation with a mobile manipulator robot,” *ArXiv*, vol. abs/1902.09051, 2019.

[6] H. Chen, Y. Kakiuchi, M. Saito, K. Okada, and M. Inaba, “View-based multi-touch gesture interface for furniture manipulation robots,” *Advanced Robotics and its Social Impacts*, pp. 39–42, 2011.

[7] J. Stückler, D. Droeschel, K. Gräve, D. Holz, J. Kläß, M. Schreiber, R. Steffens, and S. Behnke, “Towards robust mobility, flexible object manipulation, and intuitive multimodal interaction for domestic service robots,” in *RoboCup*, 2011.

[8] K. Nagahama, K. Takeshita, H. Yaguchi, K. Yamazaki, T. Yamamoto, and M. Inaba, “Estimating door shape and manipulation model for daily assistive robots based on the integration of visual and touch information,” *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7660–7666, 2018.

[9] Z. Li, Z. Xia, G. Chen, Y. Gan, Y. Hu, and J. Zhang, “Kinect-based robotic manipulation for door opening,” *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 656–660, 2013.

[10] I. M. Bullock, R. R. Ma, and A. M. Dollar, “A hand-centric classification of human and robot dexterous manipulation,” *IEEE Transactions on Haptics*, vol. 6, pp. 129–144, 2013.

[11] B. Axelrod and W. H. Huang, “Autonomous door opening and traversal,” *2015 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, pp. 1–6, 2015.

[12] N. Banerjee, X. Long, R. Du, F. Polido, S. Feng, C. G. Atkeson, M. A. Gennert, and T. Padir, “Human-supervised control of the atlas humanoid robot for traversing doors,” *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 722–729, 2015.

[13] J. Li, J. Tao, L. Ding, H. Gao, Z. Deng, and K. rui Xia, "Twisting door handles and pulling open doors with a mobile manipulator," *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 686–691, 2015.

[14] S. Gu, E. Holly, T. P. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3389–3396, 2017.

[15] Y. Urakami, A. Hodgkinson, C. Carlin, R. Leu, L. Rigazio, and P. Abbeel, "Doorgym: A scalable door opening environment and baseline agent," *ArXiv*, vol. abs/1908.01887, 2019.

[16] B. Nemec, L. Zlajpah, and A. Ude, "Door opening by joining reinforcement learning and intelligent control," *2017 18th International Conference on Advanced Robotics (ICAR)*, pp. 222–228, 2017.

[17] T. Welschehold, C. Dornhege, and W. Burgard, "Learning manipulation actions from human demonstrations," *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3772–3777, 2016.

[18] H.-R. Su and K.-Y. Chen, "Design and implementation of a mobile robot with autonomous door opening ability," *2017 International Conference on Fuzzy Theory and Its Applications (iFUZZY)*, pp. 1–6, 2017.

[19] R. B. Girshick, "Fast r-cnn," *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, 2015.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *ECCV*, 2016.