Autonomous Book Shelf Organiser

Annanya
Robotics Institute
Carnegie Mellon University
annanyaa@andrew.cmu.edu

Gangadhar
Robotics Institute
Carnegie Mellon University
vnageswa@andrew.cmu.edu

Sivvani
Robotics Institute
Carnegie Mellon University
smuthusa@andrew.cmu.edu

***Abstract*—TAbstract—This paper introduces an autonomous book sorting system utilizing a robotic manipulator equipped with a depth camera. The system is designed to categorize books from a mixed collection and place them onto designated shelves based on their genres. Leveraging computer vision techniques, the system employs object detection and classification to identify the category each book belongs to. A multi-step strategy is employed for accurate pose estimation and refinement, ensuring precise placement on the shelf. MoveIt planner is utilized to enable collision-free movement of the robotic arm during the sorting process. The system's operation is governed by a high-level finite state machine, orchestrating the sequence of actions including picking up, placing, and resetting the books. Evaluation on a physical setup demonstrates the system's capability to autonomously sort books with a high success rate of 82% while maintaining efficiency and accuracy.We open-source our dataset, models, and code on GitHub for further development. Github-https://github.com/gangadhar-nageswar/robot_autonomy_team8b**

I. Introduction

The task of autonomously manipulating objects has garnered significant attention due to its broad applicability, ranging from industrial manufacturing to warehouse management and enhancing human-robot collaboration. A particular focus in recent research has been on addressing the bin-picking problem, wherein robots are tasked with perceiving and identifying objects within cluttered environments, devising grasp strategies tailored to diverse object shapes and sizes, and executing precise maneuvers to retrieve objects without causing damage [1]. This trend aligns with the industry's growing interest in deploying autonomous solutions for tasks such as kitting and sortation.

Drawing parallels, the problem of sorting a collection of books into different categories presents similar challenges. This task involves identifying the genre or category of each book and accurately placing it onto shelves marked with corresponding categories. Our system comprises a 6 DOF Franka Emika robotic arm with a gripper and a camera mounted on the hand to capture the books postions and book's name. The robotic arm is tasked with identifying each book's category, retrieving it from the table, and placing it onto the appropriate shelf section.

To achieve autonomous book sorting, our system leverages advanced computer vision techniques, specifically object detection and segmentation using Mask R-CNN [1]. This enables the system to get the precise position and orientation of the book. Additionally, depth information captured by a RealSense depth cloud is utilized to ascertain 3D pose of the books accurately. To further refine the sorting process, Optical Character Recognition (OCR) is employed to extract book titles or relevant metadata, facilitating precise placement onto the correct shelves.

Moreover, to ensure efficient and collision-free movement of the robotic arm, a trajectory planner is incorporated into the system. This planner orchestrates the entire path, navigating around obstacles to execute the sorting task seamlessly.

In summary, our proposed system integrates object detection, depth sensing, OCR, and trajectory planning to autonomously categorize and organize a collection of books. Through this endeavor, we aim to streamline the book sorting process, reducing human intervention and enhancing efficiency in various settings such as libraries, warehouses, and retail environments.

The subsequent sections of this report delve into a detailed overview of related work on autonomous manipulation and book sorting systems. Section III provides insights into the system architecture and the algorithms employed for book detection, grasping, and path planning. Section IV presents the experimental setup and results, followed by concluding remarks and discussions on future directions in Section V.

II. Relevant Work

Autonomous manipulation, particularly in the context of the bin-picking problem, has garnered significant attention due to its broad applicability across industries [1]. Similarly, sorting a collection of books into different categories presents analogous challenges, requiring precise identification and placement onto shelves marked with corresponding categories.

To address this, our system integrates a 6 DOF robotic arm equipped with a gripper and camera for book handling [2]. Leveraging advanced computer vision techniques like Mask R-CNN enables precise book detection and segmentation. Depth information from RealSense depth cloud enhances accurate 3D pose estimation [3]. Optical Character Recognition (OCR) facilitates book title extraction for precise sorting.

Additionally, trajectory planning ensures efficient and collision-free movement of the robotic arm during sorting [5]. Through this integration, our system aims to streamline book sorting processes and enhance efficiency in various settings.

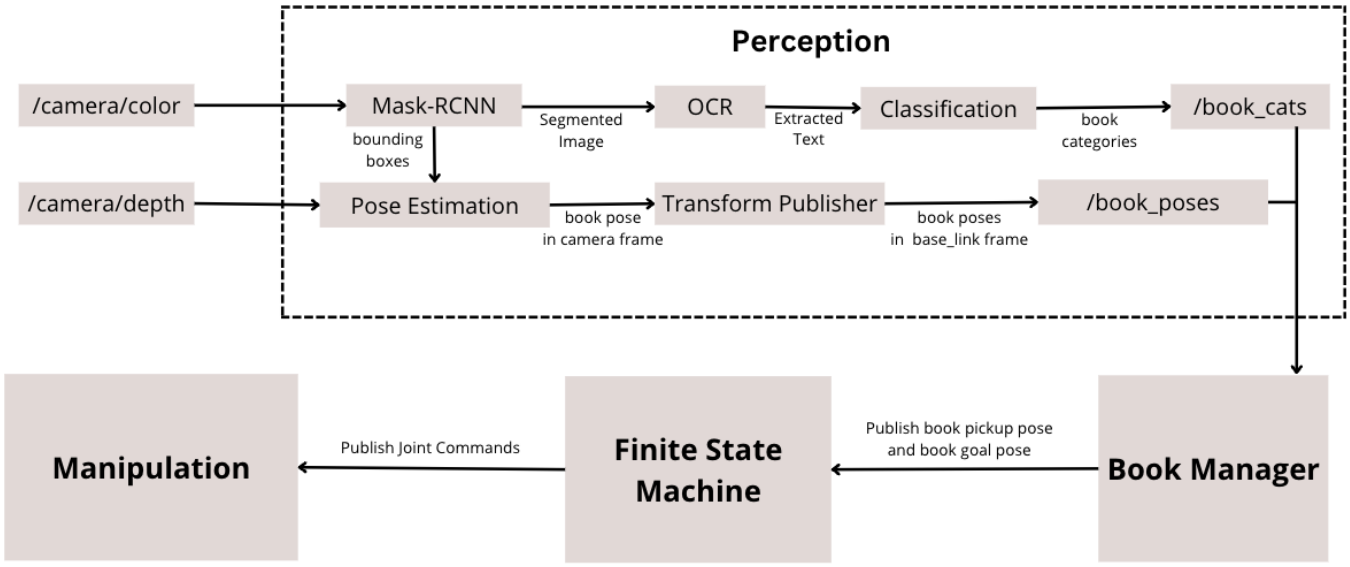

Fig. 1. System Architecture

III. Methodology

In this section, we'll elaborate on our system architecture and subsystems designed to autonomously handle book sorting tasks within a library environment. Our objective is to seamlessly pick books from a cluttered setting and place them onto designated shelves according to their respective categories. The system is engineered to withstand operational failures and resume tasks successfully. Within the library setting, we define specific areas:

A. System Architecture

Our system architecture is divided into several key components: Perception, Manipulation, Finite State Machine, and Book Manager.

In the Perception segment, the process begins with the reception of visual data from two sources, /camera/color and /camera/depth. The color camera data is processed through a Mask R-CNN model which provides bounding boxes for detected objects (books). Simultaneously, the depth camera data aids in the pose estimation of the books, determining their position and orientation relative to the camera frame. This pose data is then published through a Transform Publisher to convert the local camera frame coordinates to the base_link frame (panda_link0).

After the segmentation by the Mask R-CNN, the segmented images are passed through an OCR (Optical Character Recognition) module to extract text from the images. This extracted text is then classified into two categories based on their genre, fictional and non-fictional, and published to the /book_cats ROS topic. The classified data along with the pose data is further aggregated and published to the /book_poses topic, providing a holistic view of each detected book's textual and spatial information.

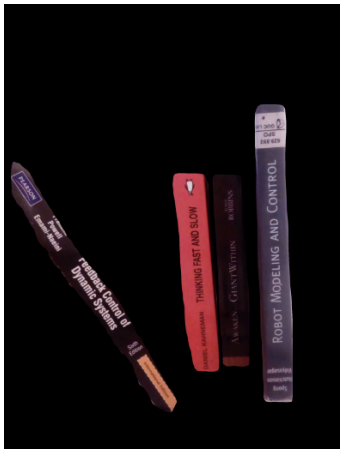

Fig. 2. Segmented Instances

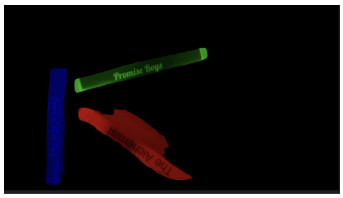

Fig. 3. Segmentation

The book manager subsystem maintains a record of the filled positions in each of the category in shelf. Based on this information, and the categories of the detected books, it assigns the goal location. The FSM interacts closely with the

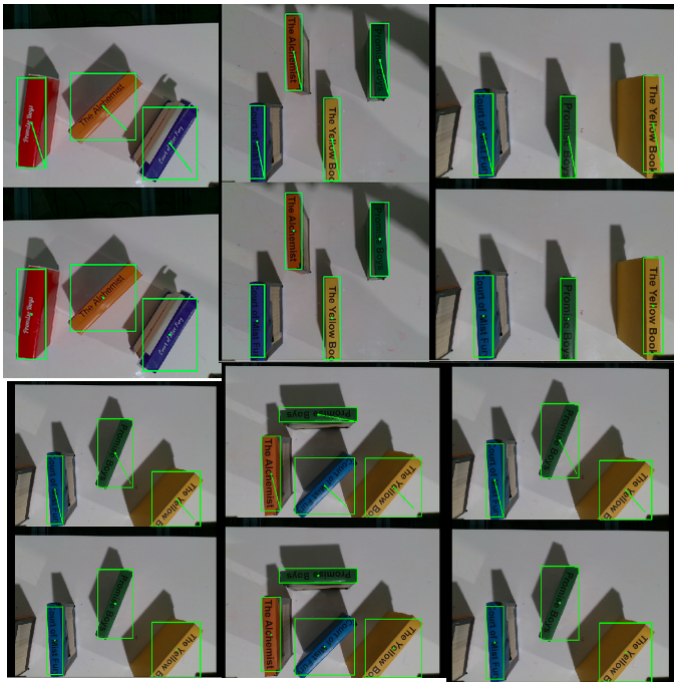

Fig. 4. Bounding Box after segmentation

Book Manager, which orchestrates the sequence of actions needed to manipulate the books, such as picking them up, and placing them at the designated location. The Book Manager uses the data from the /book_poses topic to decide the specific actions, like where and how a book should be picked up, leveraging the detailed pose information.

In terms of actuation, the Manipulation block receives commands to manipulate the robot's joints to align with the book pickup pose. These commands are determined by the Finite State Machine (FSM), which controls the flow of operations based on the state of the system and the incoming data from the Book Manager block.

Overall, this architecture highlights a sophisticated integration of vision-based perception, data processing, and robotic manipulation, tailored to autonomously handle books within a defined workspace. This setup is ideally suited for environments such as libraries or bookstores, where automated systems can enhance operational efficiency and accuracy.

B. Book Detection and Segmentation

In our project, to detect book shapes within an image and subsequently segment them by their spines, we employed the sophisticated Mask R-CNN (Mask Region-based Convolutional Neural Network) architecture. Mask R-CNN represents a state-of-the-art instance segmentation framework capable of accurately delineating object boundaries and identifying individual instances within a scene. To optimize its performance specifically for book spine detection, we fine-tuned the pre-trained Mask R-CNN model on a dataset comprising 220 annotated images. This fine-tuning process enabled the model to adapt to variations in book orientations, thereby enhancing its segmentation accuracy.

Upon successful segmentation of book spines, we proceeded to localize the grasping points for robotic manipulation. Utilizing the bounding boxes generated around each segmented book, we extracted the centroid as the optimal grasping point for the robotic gripper. Additionally, we leveraged depth information obtained from the RGBD camera to determine the precise spatial coordinates of these grasping points within the point cloud data. Furthermore, we calculated the orientation of each book by analyzing the relative positions of the centroid and bounding box points, facilitating accurate manipulation planning for the robotic system.

It's worth noting that our decision to utilize Mask R-CNN over other object detection frameworks, such as YOLOv3 (You Only Look Once), was driven by the superior performance of Mask R-CNN in accurately detecting and segmenting book spines. While YOLOv7 [2] offers real-time object detection capabilities, our evaluation revealed suboptimal results in book spine detection, necessitating the adoption of Mask R-CNN for its superior segmentation accuracy and instance-level recognition capabilities.

In summary, our approach to book spine detection involved fine-tuning a Mask R-CNN model to accurately segment individual instances of book spines within an image. This segmentation enabled precise localization of grasping points for robotic manipulation, facilitated by depth information from an RGBD camera. Our rationale for selecting Mask R-CNN over alternative frameworks was driven by its superior performance in accurately delineating book spines, thus ensuring robust and reliable detection for subsequent robotic manipulation tasks.

Algorithm 1 Segmentation and Grasp Position Detection

```
procedure HIGHLEVELPLANNER
    Initialize hover position
    while true do
        Capture 2D image from table
        Detect book spine
        if no book detected then
            break
        end if
        for each detected object do
            Perform Instance Segmentation
            Create a bounding box around it
            Get the centre of the bounding box
            Obtain the point cloud for the centre
            Obtain grasp pose in the frame of robot base
        end for
        Send the grasp point array in base frame to planner
    end while
end procedure
```

C. OCR Detection and Shelf Information Retrieval

In order to identify which book should be placed on which shelf, we utilized OCR (Optical Character Recognition) tech-

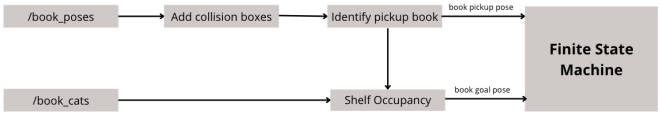

Fig. 5. Book Manager

nology. Specifically, we employed Tesseract-based OCR [4] to extract text from segmented images. The OCR process involves extracting text from the segmented image and obtaining the results in the form of bounding boxes containing the text. Subsequently, we extracted the bounding boxes of the books and concatenated all the text associated with each bounding box. We then compared this collated text with the book names stored in a CSV file, which contains information about all the books and their corresponding shelf numbers. The text with the highest confidence level was selected, and the shelf number associated with it was determined as the designated shelf for the respective book. This approach allowed us to efficiently identify the appropriate shelf for each book based on its extracted text.

D. Book Shelf Manager

Initially, the process receives data from two primary sources. The /book_poses topic provides poses of the books detected in the environment as PoseArray. This data is first utilized to ”Add collision boxes,” a crucial step for mapping out the physical space each book occupies and ensuring that the robot can navigate around or reach for books without causing disturbances or collisions.

Following this, the ”Identify pickup book” step determines which book to retrieve based on predefined criteria, such as retrieval order or priority. The decision at this stage is influenced by the spatial data, ensuring that the robot can physically access the chosen book.

Simultaneously, the /book_cats topic, which categorizes books based on its genre, is integrated into the ”Shelf Occupancy” process. The shelf occupancy process assigns the goal pose of the book detected based on both the category of the detected book and the books already present in the shelf.

The outputs from both the ”Identify pickup book” and ”Shelf Occupancy” processes converge to define the ”book pickup pose” and ”book goal pose.” The ”book pickup pose” specifies the exact position and orientation the robot’s end effector should assume to successfully pick up the book. Conversely, the ”book goal pose” designates the target location for placing the book, whether returning it to a different position or sorting it into a categorized arrangement.

These determined poses are then fed into the Finite State Machine, which orchestrates the robot’s movements and actions to execute the book pickup and placement tasks. The FSM is crucial for managing the transition between different operational states, ensuring the system responds dynamically to real-time sensory inputs and task requirements.

E. Finite State Machine

In our project, we’ve designed and implemented a Finite State Machine (FSM) to manage the tasks involved in organizing books on a shelf autonomously. The FSM orchestrates the actions of a robotic arm, controlling its movements to pick up books from a starting position, orient them correctly, and place them at predefined locations on the shelf. This FSM ensures efficient and reliable execution of the book management process.

The FSM operates based on distinct states, each representing a specific phase of the book management task. These states include:

1) **RESET**: In this initial state, the robot moves to a predefined reset position, ensuring a consistent starting point for subsequent tasks.
2) **MOVE_TO_BOOK**: Upon receiving information about the starting position and the goal position of the book from the *Book_manager* node, the FSM transitions to this state. Here, the robot plans and executes a trajectory to approach the book’s initial location.
3) **PICK_BOOK**: Once the robot reaches the starting position of the book, it closes its gripper to grasp the book securely. This action prepares the robot to lift the book from its initial position.
4) **ORIENT_BOOK**: After picking up the book, the robot needs to orient it correctly before moving it to the shelf. In this state, the robot adjusts the orientation of the book, ensuring it is aligned properly for placement.
5) **PLACE_BOOK**: With the book properly oriented, the robot moves it to the designated goal position on the shelf. This involves a sequence of movements to reach the precise coordinates for placement.
6) **SAFE_PLACE**: Once the book is placed on the shelf, the robot transitions to this state to ensure safe handling. Here, it opens the gripper to release the book and moves to a safe distance from the shelf to avoid collisions.

The FSM then loops back to the **RESET** state to prepare for the next book management task. Central to the FSM’s operation are the start_pose and book_pose, which are obtained from the Book_manager node. These poses provide crucial information about the initial position of the book and its desired location on the shelf, enabling the FSM to plan and execute the required movements accurately.

By organizing the book management process into distinct states and defining clear transitions between them, our FSM facilitates efficient and autonomous operation of the robotic arm, streamlining the task of organizing books on the shelf. This modular and structured approach enhances the reliability and scalability of our book management system, making it suitable for a variety of real-world applications.

F. MoveIt Manipulation Stack

The provided code showcases the utilization of the `moveit_class` module within a Finite State Machine (FSM) framework for trajectory planning and execution. This

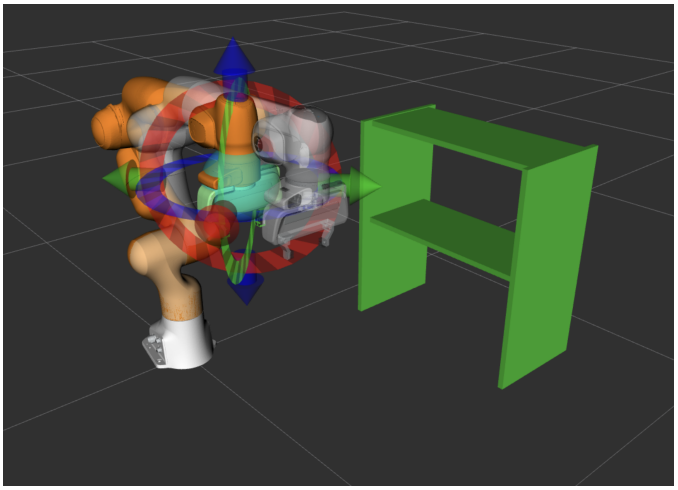

Fig. 6. Collision box of the Book Shelf in MoveIt

module serves as a crucial component for visualizing the trajectory planning and execution.

The `get_plan_given_pose()` function generates a trajectory plan based on a specified goal pose, enabling the robot to compute a feasible path from its current position to the desired end configuration.

Following trajectory plan generation, the `execute_plan()` function translates the abstract trajectory plan into tangible motion. It orchestrates the sequential movement of the robot's joints, ensuring each step of the trajectory plan is followed. Additionally, the `reset_joints()` function serves as a safety net, allowing the robot to revert to a predefined state in case of unexpected events or errors during operation. Furthermore, the `get_current_pose()` function provides real-time feedback on the robot's current position and orientation, crucial for ensuring accurate trajectory planning and execution.

In summary, the `moveit_class` module plays a multi-faceted role within the FSM framework, providing essential functions for trajectory planning, execution, safety, and real-time feedback. It serves as the backbone of the robot's motion control system, facilitating seamless and precise movement in complex environments.

IV. Evaluation

For the evaluation of our project, the primary metrics are centered on the effectiveness of the perception subsystem and the manipulation subsystem. Within the manipulation subsystem, the success criteria are defined by the robot's ability to accurately detect all books, ascertain their precise locations, and correctly place them in their designated goal positions.

To rigorously test these capabilities, we conducted a series of experimental runs, varying both the number of books and the spacing between them. Specifically, trials were performed with configurations of two and four books, introducing different spatial separations to simulate varying degrees of clutter

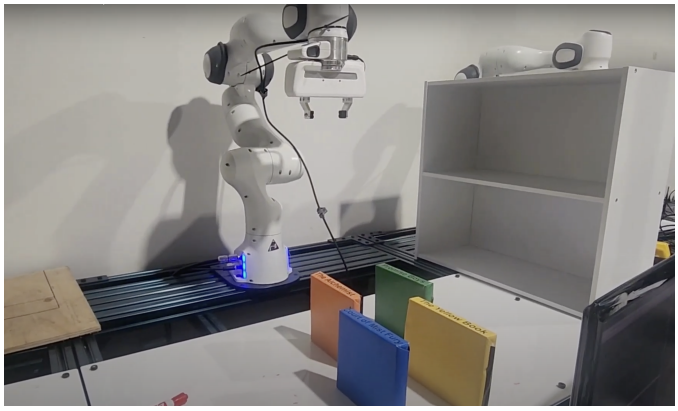

Fig. 7. Initial Environment Setup

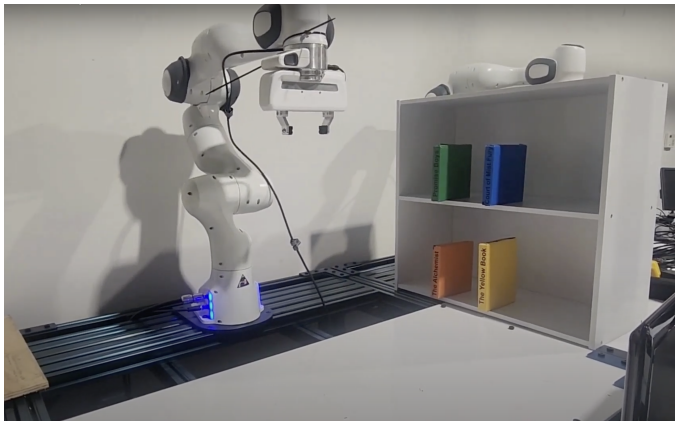

Fig. 8. Final Organised Setup

and overlap as might be encountered in real-world scenarios. These variations were intended to challenge the robot's ability to distinguish individual books and execute the necessary manipulations in a constrained space.

Additionally, the time taken to complete each task was recorded to evaluate the efficiency of the robot in performing the required tasks under varying conditions. This comprehensive evaluation approach allowed us to not only assess the effectiveness of individual subsystems but also to understand the integrated system performance in handling complex book manipulation tasks in a dynamic environment.

V. Challenges

A. Integration Complexity

Integrating the camera with the Franka manipulator proved challenging due to the need for precise alignment between detected objects and the robot's end-effector. Transformations between camera and robot frames were prone to errors, especially during calibration shifts. Additionally, integrating perception and planning modules was hindered by the need for synchronizing planner code execution with our ROS module, particularly in managing launch files. Achieving seamless interoperability required meticulous coordination and adjustments.

B. Book Spine Detection and Instance Segmentation

Book spine detection and instance segmentation presented challenges, with the former exhibiting inaccuracies that impacted the overall performance. Despite efforts to optimize spine detection, achieving precise localization remained a hurdle, necessitating further refinement of the detection algorithm. Additionally, while instance segmentation generally performed well, occasional instances of misclassification or incomplete segmentation posed obstacles in achieving reliable object identification.

C. Environmental Constraints

The environment presented numerous challenges due to the abundance of rigid bodies, including shelves and books arranged in the shelf and on the ground. These objects introduced constraints that limited the manipulator's freedom of movement and posed challenges in trajectory planning and execution. Accounting for these constraints required intricate modeling and precise collision detection algorithms to ensure safe and efficient operation within the environment.

VI. Future Work

Future enhancements in robotic book handling will significantly focus on improving the adaptability and versatility of robots in recognizing and picking up books from any orientation and side. This capability is crucial in environments where books are not uniformly placed, such as in returned piles in libraries or disorganized stock in bookstores. This requires us to develop more complex algorithms for spatial analysis that allow robots to assess and interact with books lying flat, upright, or in stacks. These algorithms would utilize depth sensing to accurately determine the orientation and adjust the manipulator accordingly.

Future developments will also refine the robot's ability to intelligently pick up a book based on its start location, intended goal location, and the understanding of surrounding collision boxes, optimizing path planning and handling efficiency. This requires us to implement algorithms capable of dynamic recalculations of paths based on real-time environmental data. This involves sophisticated modeling of the environment and potential obstacles, integrating collision avoidance into the core of the path planning process. This also requires us to develop systems that not only recognize the book and its immediate surroundings but also understand the broader context, such as the desired shelf location and optimal paths for placement without disrupting other objects. We will also need predictive models to forecast potential issues in paths or manipulations, such as predicting the likelihood of causing a book to fall from a shelf during manipulation, allowing preemptive adjustments to the approach.

VII. Conclusion

In conclusion, our project aimed to tackle the complex task of object stacking using robotic manipulation, integrating perception, planning, and execution seamlessly. Through meticulous coordination and adaptation, we navigated challenges such as integration complexities, book spine detection, instance segmentation, and environmental constraints. Despite these hurdles, our efforts yielded valuable insights and advancements in robotic manipulation, demonstrating the feasibility of automating tasks traditionally performed by humans. Moving forward, our work sets a strong foundation for future research and applications in automation and robotics offering promising opportunities to enhance productivity and efficiency across various industries.

Acknowledgment

The authors would like to thank Prof. Oliver Kroemer for their guidance throughout this project and also our TA's Abhinav and Vibhakar. We would also like to thank the classmates in MRSD for making our stay in autonomy lab fun!

References

[1] He, K., Gkioxari, G., Dollár, P., Girshick, R. (2017). Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV).

[2] A. Chench, "ROS-YOLOv7: ROS package for YOLOv7 Object Detection," GitHub repository, GitHub, 2022. https://github.com/alexandrefch/ros-yolov7.

[3] H.-I. Lin and S. Nanda, "6 DOF Pose Estimation for Efficient Robot Manipulation," in 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS), Tampere, Finland, 2020, pp. 279-284.

[4] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV).