Autonomous Shelf Arrangement: A Robotic System for Precise Placement and Organization

Shahram Najam Syed
snsyed@andrew.cmu.edu
Carnegie Mellon University

Louis Plottel
lmplotte@andrew.cmu.edu
Carnegie Mellon University

Gaurav Sethia
gsethia@andrew.cmu.edu
Carnegie Mellon University

Michael Gromis
mgromis@andrew.cmu.edu
Carnegie Mellon University

***Abstract*— This report presents the development and implementation of a robotic system for automating shelf placement tasks as part of a semester project. Utilizing a prebuilt Franka Emika robotic arm equipped with an Intel RealSense camera, this project explores methods for object manipulation in structured environments. The initial phase employed contour detection for object pose estimation, providing a fundamental understanding but revealing limitations in accuracy, particularly in depth perception and orientation. To improve the pose estimation capabilities, we transitioned to using oriented bounding boxes, which improved the detection of object orientations. This improvement, however, was insufficient for precise manipulation tasks particularly the heuristic of using the centroid of the point as the grasp point. Consequently, we integrated keypoint detection with oriented bounding boxes, significantly refining our system's ability to determine accurate grasp points based on object poses. This dual approach allowed for a more detailed analysis of object geometry, significantly improving the accuracy of grasp point detection and overall manipulation efficacy. In order to train our vision system effectively with limited data availability, a pseudo single-shot learning approach was employed. For each of the nine distinct object classes only a single image was initially captured. To enhance the dataset and improve the robustness of our oriented bounding box detector and keypoint estimation, we implemented advanced augmentation techniques by segmenting these objects and repositioned them in a multitude of orientations and scales within the training images as well as applying a technique known as random erasing to simulate environmental perturbations such as variable lighting and shadows. This approach not only expanded the effective size of our training dataset but also significantly bolstered the system's ability to handle real-world variability in object placement tasks. To complement this enhanced vision system, we developed a hierarchical planning algorithm equipped with advanced memory management features. This strategic component enabled the robotic arm to execute a series of defined poses, optimizing the movement trajectory for each object from pick-up to precise placement on the shelf. This project underscores the potential of integrating simple yet effective vision techniques and motion planning strategies to enhance robotic automation in real-world applications.**

I. INTRODUCTION

Robotic automation in logistics and warehouse management is a rapidly evolving field, promising to enhance operational efficiency and reduce human labor in repetitive and physically demanding tasks. This potential was prominently showcased in the Amazon Picking Challenges of 2015 and 2017, which were designed to spur advancements in robotic technologies capable of navigating complex, semi-structured environments to pick and place diverse inventory items with the long-term goal of warehouse automation [1][2]. These competitions highlighted both the progress and the limitations in current robotic systems, particularly in terms of adaptability, speed, and precision in handling a wide range of objects [3].

Our project draws inspiration from these challenges, seeking to address specific shortcomings identified in robotic manipulation during these competitions. While the challenges demonstrated significant achievements in integrating perception, grasp planning, and motion control, they also revealed critical gaps in precise object manipulation and pose estimation—capabilities essential for practical deployment in real-world settings.

Our approach utilizes a prebuilt Franka Emika robotic arm integrated with an Intel RealSense camera, focusing on enhancing the robot's ability to perform shelf placement tasks with high precision. The initial phase involved using contour detection for object pose estimation, which, though foundational, proved limited in handling the depth and orientation complexities presented by varied objects. To refine our capabilities, we shifted to a combination of oriented bounding boxes and keypoint detection, aiming to improve both the accuracy of pose estimation and the effectiveness of grasp point determination.

In tandem with these technical enhancements, we embraced a pseudo single-shot learning approach for training our vision system, crucial for overcoming the challenges posed by limited available data. By applying advanced augmentation techniques such as object segmentation and repositioning, alongside random erasing for environmental robustness, we significantly expanded the training dataset's diversity and utility. This method not only enhanced the system's adaptability to real-world variations but also aligned with insights gained from the Amazon Picking Challenge regarding the importance of robust, flexible perception systems in dynamic environments.

Through these innovations, our project contributes to the ongoing discourse on robotic automation, emphasizing the integration of advanced computer vision techniques and

strategic motion planning to tackle the complexities of modern logistics operations.

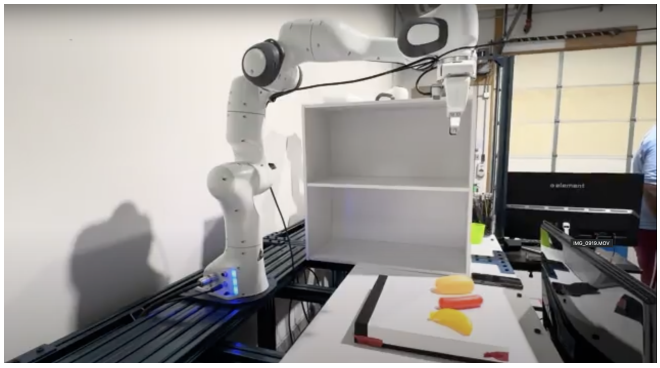

Fig. 1. Structured layout of the experimental setup

II. RELEVANT WORK

Recent advancements in deep learning have significantly enhanced the capabilities of automated bin picking systems and shelf replenishment, which are integral to modern industrial robotics. Researchers have focused on improving the accuracy and efficiency of these systems through image processing and object recognition techniques. Cordeiro et al. (2022) [4] provide a comprehensive overview of the state-of-the-art deep learning approaches utilized in bin picking, highlighting the integration of convolutional neural networks (CNNs) with 3D vision to enhance depth perception and object differentiation. This integration is crucial for identifying and manipulating objects in cluttered environments.

Further developments include the work of Mahler et al. (2017) [5], who introduced Dex-Net, a robust grasp planning system that uses deep neural networks to generate grasp plans from 3D point clouds. This system significantly improves the success rate of picking operations in cluttered scenes by enabling more accurate grasp predictions. Similarly, Morrison et al. (2020) [6] extended this approach by implementing generative grasping convolutional neural networks that can generalize grasps for a variety of objects unseen during training, thereby enhancing the adaptability of robotic systems to new environments.

Additionally, the role of synthetic data in training these models cannot be overstated. Zeng et al. (2018) [7] demonstrated how training with synthetic data can improve the robot's ability to generalize across different tasks, a critical factor in environments where variability in object types and arrangements is common. These advancements collectively push the boundaries of what automated systems can achieve in practical, everyday tasks within structured environments.

Deep reinforcement learning (DRL) has emerged as a transformative approach for teaching robots to perform complex manipulation tasks autonomously. Levine et al. (2016) [8] showcased the application of DRL to automate fine motor skills in robots, enabling them to perform tasks that require high dexterity, such as screwing caps on bottles and handling delicate materials. Their work illustrates the potential of DRL in enhancing the precision and adaptability of robotic systems.

Building on these foundations, Gu et al. (2017) [9] explored the use of DRL in environments characterized by high-dimensional sensory inputs. They highlighted how DRL helps robots adapt to dynamic conditions and perform precise manipulations, which are essential for tasks involving intricate assembly operations. Moreover, Kumar et al. (2019) [10] discussed the integration of DRL with simulation environments, which allows for efficient and risk-free training of robots in complex manipulation tasks, thus reducing the costs and hazards associated with direct physical training.

The synergy between these deep learning methodologies and reinforcement learning strategies represents a significant leap forward in the field of robotic manipulation, offering robust solutions for the automation of complex tasks in industrial and commercial settings

III. METHODOLOGY

This section outlines the comprehensive methodology employed in the development and implementation of our robotic system for automating shelf placement tasks. The objective was to design a system capable of accurate and efficient object manipulation within a structured environment using a prebuilt Franka Emika robotic arm coupled with an Intel RealSense camera. Our approach integrates advanced computer vision techniques and robotic motion planning to address the challenges associated with object detection, pose estimation, and precise manipulation.

The development process was structured into several phases, each focusing on a specific aspect of the system's capabilities:

1) **Contour Detection and RANSAC Refinement:** We began with basic computer vision techniques to establish a foundational layer for object detection and initial pose estimation. This phase involved classical methods like contour detection to identify object boundaries, followed by a RANSAC-based refinement to improve the accuracy of bounding box estimations.
2) **Oriented Bounding Box Detection:** Recognizing the limitations of classical methods under varying environmental conditions, we incorporated a deep learning approach using YOLOv8 for oriented bounding box detection. This shift aimed at enhancing the robustness and accuracy of pose estimation across different object orientations and complex scenarios.
3) **Keypoint Detection for Enhanced Manipulation:** To further refine our system's interaction with objects, especially for determining viable grasp points, we integrated keypoint detection. This step was crucial for enabling precise manipulation beyond the capabilities of traditional bounding box approaches.
4) **Advanced Data Augmentation Techniques:** Given the limited availability of diverse training data, we employed sophisticated data augmentation strategies to simulate a wide range of environmental variations,

thereby enhancing the generalization capability of our models.

5) **Hierarchical Planning Algorithm:** The final integration involved a sophisticated planning algorithm that orchestrates the sequence of detection, manipulation, and placement tasks. This algorithm is designed to optimize the manipulation process, reduce operational time, and ensure the reliability of object interactions.

The methodologies adopted in this project not only address the immediate needs of automated shelf placement but also contribute to the broader field of robotics by demonstrating how integrated systems can leverage both classical and modern approaches to solve complex real-world challenges. The following subsections detail each of these methodologies, elucidating the technical processes and the rationale behind our approach.

A. Contour Detection

The primary motivation for utilizing classical computer vision techniques, specifically contour-based pose estimation, was due to their ease of implementation and robustness in structured environments. This method is foundational in our system's object detection and localization pipeline.

The process begins by converting the captured RGB images from the RealSense camera into the grayscale color space. This transformation simplifies the image data, reducing computational overhead while maintaining the necessary details for effective edge detection. Following this, a Gaussian blur is applied to smooth out the image, reducing noise and improving the reliability of subsequent edge detection operations.

Edge detection is performed using the Canny algorithm, where we carefully tune the threshold values to capture significant edges while minimizing the impact of trivial discrepancies in the image. This step is crucial for delineating the precise boundaries of objects within the scene.

After edge detection, the OpenCV function `cv2.findContours` is employed to extract contours from the binary image. These contours represent potential object boundaries but are often affected by environmental factors such as variable lighting, shadows, and partial occlusions, which can complicate the detection accuracy.

Given the limitations posed by simple contour detection—particularly its susceptibility to environmental noise and its inability to capture depth information—we integrated a two-step approach to refine our object localization:

- **Initial Bounding Box Estimation:** Using the contours obtained, we initially fit bounding rectangles around these contours. This provides a coarse estimation of the object's location and orientation but was often inaccurate due to the irregular shapes of objects and overlapping contours.
- **Custom Two-Step RANSAC Algorithm:** To enhance the accuracy of our bounding boxes, we implemented a custom two-step RANSAC algorithm. Initially, this algorithm samples various rectangle configurations within predefined bounds in the image. It then evaluates these based on an inlier ratio, determining the best fit for each contour. This process, however, yielded coarse estimates due to the extensive search space and computational constraints.
- **Refinement Phase:** In the subsequent refinement step, we constrained our search to a smaller area around the initially estimated bounding boxes. This localized search allowed for more precise adjustments, signif-

Algorithm 1 Algorithm-1: Object Pose and Depth Estimation using Contour Detection and RANSAC

```
procedure DETECTOBJECTS
    Initialize and start the RealSense pipeline
    Configure depth (640x480, 30fps) and color (640x480, 30fps) streams
    while system active do
        frames ← Capture frames from the pipeline
        color_frame ← frames.get_color_frame()
        depth_frame ← frames.get_depth_frame()
        if not color_frame then
            Continue
        end if
        gray ← ConvertToGrayscale(color_frame)
        blurred ← ApplyGaussianBlur(gray, (5, 5), 0)
        edges ← PerformCanny(blurred, 75, 150)
        contours ← FindContours(edges)
        for each contour in contours do
            Initialize BestRatio ← 0
            Initialize BestBox ← null
            for NumIters ← 1 to MaxIters do
                SampleParams ← randomly sample position, size, and angle
                Box ← construct bounding box with SampleParams
                Ratio ← ComputeInliers(depth_frame, Box, contours)
                if Ratio > BestRatio then
                    BestRatio ← Ratio
                    BestBox ← Box
                end if
            end for
            rect ← BestBox
            depth ← CalculateDepth(depth_frame, rect)
            point ← DeprojectTo3D(depth_frame, rect.center, depth)
            DrawBoxAndInfo(color_image, rect, depth, point)
        end for
        Display(color_image)
        if 'q' key is pressed then
            break
        end if
    end while
    Stop pipeline and clean up
    return points, angles
end procedure
```

icantly improving the accuracy of our bounding box estimations as validated by our empirical results.

While contour-based pose estimation is not without its challenges, adding in RANSAC as a pose refinement step improved the estimated pose, making it an indispensable component of our methodology. It lays the groundwork for more sophisticated techniques required for the techniques mentioned in succeeding sections that require a preliminary understanding of object locations and poses.

Fig. 2. Example Image Showing Detected Bounding Boxes using Contour Detection and RANSAC Refinement

B. Oriented Bounding Box

Given the inherent limitations in the contour detection followed by RANSAC algorithm—primarily its susceptibility to false positives and negatives due to variable lighting conditions, complex backgrounds, and partial occlusions—we pivoted towards a more robust, learning-based approach using YOLOv8. This transition was motivated by the need for a more accurate, versatile, and data-driven method for pose estimation, particularly in dynamic environments where object orientations can vary significantly.

YOLOv8 introduces the concept of Oriented Bounding Boxes (OBB), which are particularly useful for detecting objects at various angles and rotations, thereby providing a more precise pose estimation. The implementation began with configuring the YOLOv8 model to detect OBB, using the pretrained yolov8m-OBB.pt model as a starting point to leverage transfer learning, thereby enhancing learning efficiency and initial accuracy.

The data for training was sourced from an augmented dataset specifying object locations and orientations, ensuring a variety of scenarios to enhance the model's generalization capabilities. This approach was designed to improve the model's robustness against diverse environmental conditions.

The training was conducted using the following specifications detailed in the YAML configuration:

YAML Configuration for YOLOv8 Training

```
epochs: 100
batch: 16
imgsz: 640
optimizer: auto
pretrained: true
augment: false
```

The core algorithm for pose estimation with YOLOv8 can be outlined as follows:

Algorithm 2 Pose Estimation with YOLOv8

1: **procedure** ESTIMATEPOSE
2: Load pre-trained YOLOv8 model
3: Configure model with training parameters
4: Train model on annotated dataset with diverse poses
5: Validate model on separate validation set
6: Deploy model for real-time object detection and pose estimation
7: **end procedure**

The pose estimation is based on predicting the parameters of the oriented bounding box, which are given by:

$$B = (x, y, w, h, \theta)$$

where (x, y) represents the center of the box, w and h denote the width and height, and θ is the rotation angle of the box. The loss function used during training incorporates aspects unique to pose estimation, such as the orientation angle, which is optimized alongside traditional parameters like the bounding box dimensions and object classification scores.

Upon deployment, the YOLOv8 model exhibited outstanding performance metrics on an NVIDIA RTX 3060 GPU, achieving a frame rate of 30 FPS with a mean Average Precision (mAP) of 0.925 on the validation set. These metrics illustrate the model's capability to provide highly accurate and real-time pose estimations, significantly enhancing the system's operational efficiency.

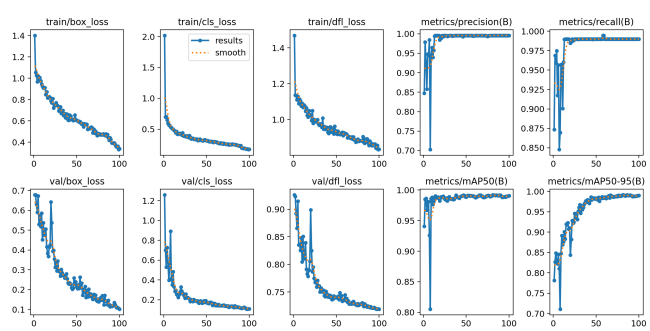

Fig. 3. Training and validation curves for Oriented Bounding Box Detection

Fig. 4. Example Image Showing Detected Oriented Bounding Boxes using YOLOV8

C. Oriented Bounding Box w/ Keypoint detection

While oriented bounding boxes significantly improved our system's ability to detect the orientation and position of objects, the centroids provided by these boxes often do not correspond to feasible grasp points, especially for irregularly shaped or asymmetric objects. Keypoints, which can be strategically defined at parts of the object most suitable for grasping (like handles or specific edges), offer a solution to this limitation.

The integration of keypoint detection allows the system to identify and utilize these specific points, thereby enhancing the robotic arm's interaction with objects in terms of both accuracy and reliability. This is particularly beneficial in cluttered environments where precise interaction with individual objects is required.

To implement this enhanced detection mechanism, we adapted our model to concurrently detect oriented bounding boxes and keypoints within those boxes. The following algorithm outlines the steps involved in this process:

Upon deployment, the YOLOv8 model exhibited outstanding performance metrics on an NVIDIA RTX 3060 GPU, achieving a frame rate of 30 FPS with a mean Average Precision (mAP) of 0.975 on the validation set. These metrics illustrate the model's capability to provide highly accurate and real-time pose estimations, significantly enhancing the system's operational efficiency.

Algorithm 3 Pose Estimation with OBB and Keypoint Detection

```
procedure DetectAndOrientObjects
    Initialize YOLOv8 with capabilities for both OBB and keypoint detection
    Load the model pre-trained on a relevant dataset
    Continuously capture images from the Intel RealSense camera
    Preprocess the images for real-time processing
    for each frame in the video stream do
        Detect OBBs and corresponding keypoints within those boxes
        Analyze keypoints to determine viable grasp points
        Optimize the grasp strategy based on keypoint configuration
        Calculate the most effective approach vector for the robotic arm
        Execute the grasping action
    end for
    Provide real-time adjustments dibased on sensory feedback
end procedure
```

Fig. 6. Example Image Showing Detected Oriented Bounding Boxes and Kepoints for optimal grasps using YOLOV8

D. Data Augmentation

Our project significantly leveraged advanced data augmentation techniques to enhance the training robustness and efficacy of our vision systems. Specifically, we implemented "Random Erasing" and "Cut, Paste and Learn" methods, which were pivotal in augmenting our limited dataset.

The Random Erasing augmentation technique introduces a simple yet effective method to enhance the model's ability

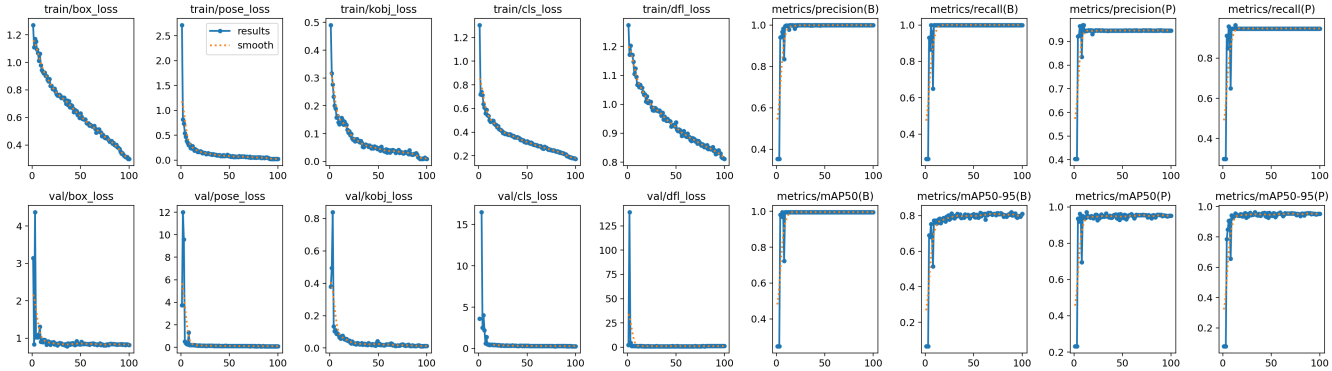

Fig. 5. Training and validation curves for Key Points of Oriented Bounding Box Detection

to handle occlusion. During training, Random Erasing randomly selects a rectangle region within an image and erases its pixels with random values, as described by Zhong et al. (2017) [11]. This process generates images with various levels of occlusion, thereby reducing overfitting and enhancing the model's robustness to real-world scenarios where objects may be partially obscured.

Algorithm 4 Random Erasing Procedure

Require: Input image I; Image size W and H; Area of image S; Erasing probability p; Erasing area ratio range s_l and s_h; Erasing aspect ratio range r_1 and r_2.
Ensure: Erased image I^*.

$p_1 \leftarrow \text{Rand}(0, 1)$
if $p_1 \geq p$ **then**
  $I^* \leftarrow I$
  return I^*
else
  while True **do**
    $S_e \leftarrow \text{Rand}(s_l, s_h) \times S$
    $r_e \leftarrow \text{Rand}(r_1, r_2)$
    $H_e \leftarrow \sqrt{S_e \times r_e}, W_e \leftarrow \sqrt{\frac{S_e}{r_e}}$
    $x_e \leftarrow \text{Rand}(0, W), y_e \leftarrow \text{Rand}(0, H)$
    if $x_e + W_e \leq W$ and $y_e + H_e \leq H$ **then**
      $I_e \leftarrow (x_e, y_e, x_e + W_e, y_e + H_e)$
      $I(I_e) \leftarrow \text{Rand}(0, 255)$
      $I^* \leftarrow I$
      return I^*
    end if
  end while
end if

The "Cut, Paste and Learn" approach by Dwibedi et al. (2017) [12], focuses on the synthesis of new training samples by 'cutting' object instances from source images and 'pasting' them onto random background scenes. This method not only addresses the scarcity of labeled data but also ensures that the detection models are trained on images that vary significantly in terms of background, object orientation, and scale. By implementing this technique, we could effectively train our model to recognize and localize objects across diverse conditions without the need for extensive data collection.

Algorithm 5 Cut, Paste and Learn Procedure

Require: Source image I_s; Target background image I_t; Object mask M.
Ensure: Augmented image I_{aug}.

Extract object from I_s using mask M
$O \leftarrow I_s$ where M is true
Select a random location (x, y) on I_t
Calculate placement bounds based on object dimensions and I_t size
if object fits within I_t at (x, y) **then**
  Paste O onto I_t at (x, y)
  Update M to new location on I_t
  Apply blending techniques if necessary to integrate O into I_t
  $I_{aug} \leftarrow I_t$ with O pasted
  Optionally apply further image processing (e.g., color adjustment)
else
  Adjust (x, y) or resize O and retry
end if
return I_{aug}

Both augmentation strategies were integrated into our training pipeline to prepare our models for the complexities of real-world applications. The Random Erasing was particularly used to introduce randomness in object appearance by occluding different parts of the objects during training. Meanwhile, the "Cut, Paste and Learn" method allowed us to simulate a wide range of environmental variations, significantly improving our model's exposure to different scenarios.

E. Planner

The planner orchestrates a series of operations that integrate perception, motion planning, and manipulation. It is structured to first identify the position and orientation of target objects, then compute a trajectory for approach, grasp, and placement. This sequence minimizes unnecessary

movements and adjusts in real-time to sensory feedback, enhancing both precision and reliability.

The planning strategy is divided into several key stages, each responsible for a segment of the task:

- Initialize robot: Reset joints and open gripper.
- Move to observation position: Navigate the camera to a predefined location for object viewing.
- Object detection cycle: Continuously capture images and detect objects.
- For each detected object:
  - Estimate object pose: Determine the position and orientation.
  - Plan approach: Compute the trajectory for moving towards the object.
  - Execute movement: Move the robot arm along the planned path.
  - Grasp object: Adjust the gripper to pick up the object securely.
  - Calculate placement position: Identify the appropriate shelf location for placement.
  - Place object: Deposit the object at the designated location.
- Return to start position: Move the robot back to the initial position to begin a new cycle or end the sequence.

The pseudocode below outlines the procedural logic implemented by our hierarchical planning algorithm:

Fig. 7. Placing Operation In Action

IV. EVALUATION

To evaluate the system, we randomly scatter the 3 instances of the classes in the workspace (mindful of the virtual walls) to examine the successful grasp and stocking in the shelf. The entire run can be viewed at `https://youtu.be/Csf0lCaH_Tg`.

The system was successfully able to detect, grasp, move, and stock the shelf for 3 iterations of the detected objects with a grasp success rate of approximately 75%. The system successfully recovered from failed grasps according to the behavior designated in the behaviour planner. No human interventions were required to push the objects into the pickup region.

Algorithm 6 Hierarchical Planning for Robotic Manipulation

```
procedure EXECUTEMANIPULATIONCYCLE
    InitializeRobot()                          ▷ Reset joints and open gripper
    MoveToInitialPosition()                    ▷ Position robot at camera perch
    while True do                              ▷ Continuous operation loop
        image ← CaptureImage()                 ▷ Capture view from camera
        objects ← DetectAndEstimateObjects(image)   ▷ Object detection and pose estimation
        if objects is empty then
            break                              ▷ Exit loop if no objects are detected
        end if
        for each object in objects do
            success ← ApproachAndGrasp(object)     ▷ Approach, align, and grasp
            if not success then
                continue                       ▷ Skip to next object if grasp fails
            end if
            position ← DeterminePlacementPosition(object)   ▷ Calculate where to place object
            PlaceObject(position)              ▷ Deposit object at target location
            ReturnToObservationPosition()      ▷ Return to start for next cycle
        end for
    end while
    CleanUp()                                  ▷ Finalize and clean up resources
end procedure
```

V. CHALLENGES

The development of an automated robotic system for shelf placement using a prebuilt Franka Emika robotic arm and an Intel RealSense camera presented several technical challenges that were pivotal in shaping the project's trajectory. These challenges stem primarily from limitations in initial detection methods, complexities in improving pose estimation, and constraints related to data handling and algorithmic integration.

1. Contour Detection Limitations: Initially, we employed contour detection for object pose estimation, which, while providing a fundamental understanding of object shapes, revealed significant limitations in depth perception and orientation accuracy. The method struggled particularly with objects that had complex geometries or were partially occluded, leading to erroneous pose estimations in addition to environment perturbations that could not reliably inform the robotic arm's manipulation tasks.

2. Transition to Oriented Bounding Boxes: To improve pose estimation, we transitioned to using oriented bounding boxes. Although this technique enhanced our ability to detect object orientations, it was still insufficient for precise manipulation tasks. The primary challenge was the heuristic

use of the centroid of the bounding box as the grasp point, which did not consistently correspond to the most stable grasp points on irregularly shaped objects.

3. Integration of Keypoint Detection: The integration of keypoint detection with oriented bounding boxes aimed to refine the system's grasp point determination. However, accurately aligning keypoints with the robot's grasping mechanism required complex calibration and continuous adjustments to ensure precision. This process was particularly challenging due to varying object sizes and shapes, which sometimes led to misalignment between detected keypoints and actual feasible grasp points.

4. Pseudo Single-Shot Learning Constraints: With limited data availability, we adopted a pseudo single-shot learning approach, which posed its own set of challenges. Generating a robust dataset from a single image per object class involved sophisticated augmentation techniques, including object segmentation and random erasing to simulate environmental perturbations like variable lighting and shadows. However, ensuring that these augmentations preserved the essential features of the objects for accurate detection and pose estimation proved to be a complex and iterative process.

5. Hierarchical Planning Algorithm Complexity: Developing a hierarchical planning algorithm equipped with advanced memory management features required intricate programming and testing to optimize the sequence of movements from pick-up to shelf placement. The challenge was compounded by the need to ensure that the algorithm could adapt to real-time feedback from the vision system and adjust the robot's trajectory dynamically to account for any discrepancies in object placement or orientation.

VI. FUTURE WORK

The project has set a strong foundation for further enhancements with concentrated future efforts focusing on refining the system's capabilities, expanding its applicability, and ensuring it can operate under increasingly complex conditions. The following are the primary areas for future development:

1. Enhanced Object Recognition and Classification: To improve the system's ability to handle a wider variety of objects, further development in object recognition and classification is essential. Implementing more advanced neural networks, such as deeper convolutional neural networks (CNNs) or capsule networks, could provide better feature extraction and recognition capabilities. Additionally, incorporating more diverse and extensive datasets, possibly through crowd-sourced image collection or partnerships with retail organizations, will help train these models to recognize a broader array of object types and conditions.

2. Advanced Pose Estimation Techniques: While the integration of oriented bounding boxes and keypoint detection has improved pose estimation, there is room for further enhancement. Future work could explore the use of 3D reconstruction techniques or stereo vision to gather more accurate depth information. Leveraging state-of-the-art methods in computer vision, such as deep learning models trained on 3D data, could significantly refine the system's ability to ascertain object orientation and position.

3. Adaptive Grasping Mechanisms: Developing adaptive grasping mechanisms that can adjust based on the object's material, weight, and fragility will make the robotic system more versatile and sensitive to different handling requirements. Research into variable stiffness grippers and tactile sensors that provide feedback on grip strength and object stability could lead to more nuanced and effective manipulation strategies.

4. Machine Learning Enhancements: Enhancing the machine learning algorithms used for path planning and manipulation is crucial. The integration of reinforcement learning with real-time feedback loops could allow the system to learn from each interaction, improving its efficiency and adaptability over time. Moreover, exploring transfer learning to apply knowledge gained from one task to different but related tasks could accelerate the learning process and enhance system performance.

5. Robustness to Environmental Variations: Increasing the system's robustness to changes in the environment, such as lighting conditions, background clutter, and unexpected obstacles, is essential for real-world deployment. Future work will involve developing algorithms that are more resilient to such variations, possibly through augmented reality simulations that can train the system under a range of scenarios.

VII. CONCLUSION

This report detailed the development and implementation of a robotic system designed to automate shelf placement tasks, leveraging a prebuilt Franka Emika robotic arm and an Intel RealSense camera. Throughout the semester project, we explored and integrated various methods to enhance the robot's capabilities in object manipulation within structured environments.

Initially, our system employed contour detection for object pose estimation, which provided a basic understanding but was limited in accurately determining object depth and orientation. Recognizing these limitations, we transitioned to utilizing oriented bounding boxes, which improved the detection of object orientations but still lacked in precision for effective manipulation. Consequently, we integrated keypoint detection with oriented bounding boxes, significantly enhancing the accuracy of grasp point determination and the overall manipulation efficacy.

To address the challenges posed by limited training data, we adopted a pseudo single-shot learning approach, significantly expanding our training dataset through advanced augmentation techniques. These included segmenting objects and repositioning them in a variety of orientations and scales, as well as employing random erasing to simulate environmental perturbations. This approach not only enriched the dataset but also bolstered the system's robustness to real-world variability in object placement tasks.

Furthermore, we developed a hierarchical planning algorithm equipped with advanced memory management features to optimize the robotic arm's movement trajectory for each

object from pick-up to precise placement on the shelf. This strategic component was crucial in enabling the robotic system to execute a series of defined poses efficiently.

The project demonstrated the potential of integrating simple yet effective vision techniques and motion planning strategies to enhance robotic automation in real-world applications. It highlighted the importance of robust and flexible perception systems capable of adapting to dynamic environments, which is crucial for the practical deployment of robotic systems in industries such as logistics and warehousing.

In conclusion, while our project made significant strides in automating shelf placement tasks, the exploration and integration of advanced technologies highlighted the ongoing need for innovation in robotic manipulation. Future work will focus on refining these techniques to improve the adaptability, efficiency, and reliability of robotic systems, ensuring they can meet the diverse and evolving demands of real-world applications.

ACKNOWLEDGMENT

We extend our deepest gratitude to the Robotics Institute at Carnegie Mellon University for providing the resources and environment necessary for conducting this research. Special thanks are due to Prof. Oliver Kromer, the course instructor for 16662 Robot Autonomy, whose guidance and expertise were invaluable throughout the development of this project. We also appreciate the dedicated assistance and insightful feedback from our teaching assistants, Abhinav Gupta and Vibhakar Mohta, whose support played a crucial role in the success of our work. Their contributions were instrumental in helping us navigate the complexities of robotic system design and implementation.

References are important to the reader; therefore, each citation must be complete and correct. If at all possible, references should be commonly available publications.

REFERENCES

[1] Baker, Peter, and Zaheed Halim. "An exploration of warehouse automation implementations: cost, service and flexibility issues." Supply Chain Management: An International Journal 12.2 (2007): 129-138.

[2] D'Andrea, Raffaello. "Guest editorial: A revolution in the warehouse: A retrospective on kiva systems and the grand challenges ahead." IEEE Transactions on Automation Science and Engineering 9.4 (2012): 638-639.

[3] Correll, Nikolaus, et al. "Analysis and observations from the first amazon picking challenge." IEEE Transactions on Automation Science and Engineering 15.1 (2016): 172-188.

[4] Cordeiro, Artur, et al. "Bin picking approaches based on deep learning techniques: A state-of-the-art survey." 2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC). IEEE, 2022.

[5] Zhang, Harry, et al. "Dex-Net AR: Distributed deep grasp planning using a commodity cellphone and augmented reality app." 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020.

[6] Morrison, Douglas, Peter Corke, and Jürgen Leitner. "Learning robust, real-time, reactive robotic grasping." The International journal of robotics research 39.2-3 (2020): 183-201.

[7] Zeng, A., Yu, K.-T., Song, S., Suo, D., Walker, E., Rodriguez, A., Xiao, J. "Multi-view self-supervised deep learning for 6D pose estimation in the Amazon Picking Challenge." In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017.

[8] Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., Quillen, D. "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection." International Journal of Robotics Research, vol. 37, no. 4-5, pp. 421–436, 2018.

[9] Gu, S., Holly, E., Lillicrap, T., Levine, S. "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates." In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017.

[10] Kumar, V., Todorov, E., Levine, S. "Optimal control with learned local models: Application to dexterous manipulation." In Proceedings of the 2019 IEEE International Conference on Robotics and Automation (ICRA), Montreal, Canada, 20–24 May 2019.

[11] Zhong, Zhun, et al. "Random erasing data augmentation." Proceedings of the AAAI conference on artificial intelligence. Vol. 34. No. 07. 2020.

[12] Dwibedi, Debidatta, Ishan Misra, and Martial Hebert. "Cut, paste and learn: Surprisingly easy synthesis for instance detection." Proceedings of the IEEE international conference on computer vision. 2017.