Advanced Robotic System for Efficient Pick-and-Place of Deformable Poultry in Cluttered Bin: A Comprehensive Evaluation Approach

Rekha Raja[1], Akshay K. Burusa[1], Gert Kootstra[1], Eldert J. Van Henten[1]

***Abstract*— This research paper presents an advanced robotic system designed for efficient pick-and-place of deformable poultry pieces from cluttered bins. The system incorporates a novel architecture with seamless integration of various modules, enabling the robot to handle deformable poultry with precision. It introduces a comprehensive evaluation approach to assess the system's performance, considering perception, state modeling, planning and control, gripping and manipulation. The experiments were conducted on two different samples of chicken pieces with varying weights and shapes, under complex and simple scenarios. Performance indicators, failure categories, and cycle time were used for evaluation. The evaluation revealed an overall success rate of 49.4% for picking and placing chicken pieces, with failure rates of 21.8% for perception, 30.7% for gripping, and 11% for manipulation modules. These results highlight areas of improvement, particularly in object detection, grasp pose estimation in clutter, and gripper designs for deformable products, to create a robust pick-and-place solution. The proposed robotic system and evaluation method hold immense potential for revolutionizing the meat processing industry and other food processing sectors, making automation more efficient and adaptable to meet the increasing demand in the food industry.**

***Index Terms*— Robotic system architecture, Deformable object, Machine learning, Bin picking, System analysis. Robotic system architecture, Deformable object, Machine learning, Bin picking, System analysis.**

I. Introduction

With the advent of industry 4.0 there has been a demand to transform food processing using robotics and artificial intelligent (AI) technology. However current robotic and AI technology is not able to deal with the large variations in shape, size, and softness of natural food products like meat. Although, worldwide, meat consumption has been rapidly increasing [1] due to a change in diet to include more meat. Today, poultry production accounts for roughly one-third of total meat production worldwide [2]. Since the demand for poultry products is increasing globally, the need to process more poultry is also increasing. Despite the fact that the poultry-processing industry has achieved a high degree of automation already, technical solutions mainly rely on mechanical engineering. Certain tasks, such as picking chicken pieces from a cluttered pile in a crate and placing them in order, are still a challenging problem to automate [4], and therefore currently this task is performed by human workers (see Figure 1). This so-called *bin-picking task* in poultry processing is technologically challenging due to the fact that poultry pieces come in varied shapes and sizes and are packed closely and disorderly piled up together in a bin which makes it extremely hard to segregate and place them in sequential order in another bin or on a conveyor belt in the desired position and orientation, using purely mechanical means. A different and more advanced kind of technology is called for.

[1]Farm Technology group, Wageningen University and Research, 6700AA, Wageningen, The Netherlands

Fig. 1: Poultry pieces are currently manually segregated from cluttered piles in bins to place them in order on the conveyor belt by human laborers for further processing in the processing line in an industry that is still labor intensive (source: [3]).

Robot technology has the potential to provide flexible solutions in the face of product and task variation. So far, robot technology has been extremely successful in, for instance, the manufacturing industry [5]. Repetitive operations on large numbers of objects that are very well-defined in terms of location, orientation, shape, and size were instrumental to this success. Robots have also entered the agro-food production chain, where adoption continues to grow rapidly [5]. This trend expresses the readiness of this industry to adopt more advanced technology. However, the robots currently used in the agro-food domain are dealing with simple operations on products without much variation and are essentially based on industrial robotic pick-and-place technology. Current robot technology still cannot meet the requirements of flexibility when dealing with variation within and between different classes of products. Switching between tasks is an additional challenge on which robotics still needs to deliver. And due to the characteristic of the meat industry, robot technology needs capabilities of perception and action to deal with complex manipulation tasks while handling products having a surface with low and varying friction characteristics and products that easily deform under external forces.

In the FlexCraft project [6], we aim to develop more flexible robotic technology for poultry processing. Automating the poultry bin-picking task with robots requires flexible capabilities, including perception, world modelling, planning and control, and gripping. However, integrating these modules robustly presents a challenge. Moreover, evaluating such a large complex system requires a systematic approach that considers the performance of individual modules that adds to the performance of the overall system. The existing approaches for evaluating robotic systems focus either on the whole system's performance without assessing individual modules or solely on the evaluation of individual modules without translating the results to the system's performance. A comprehensive evaluation method is needed to understand the system's performance, identify failures, and guide future research for improvement.

Therefore, the contributions of this work are as follows: i) we

a novel robotic system architecture is proposed and developed for picking poultry pieces from a cluttered pile in a bin and placing them in sequential order in another bin or on a conveyor belt, ii) a novel approach to evaluate the performance of such a large system of bin-picking robots is proposed by assessing individual modules as well as the overall system, and iii) using this procedure, we showed where and why the robotic bin picking system failed and where future research should focus to improve the system's performance and discuss the pros and cons of each sub-component of the systems and technologies including the details of the systems.

The paper is organized as follows. Section III-A describes the experimental environment. Section III-C introduces the hardware and software components of the systems. Section III-D describes the architecture of the proposed system. Section III-E presents algorithms proposed for the individual modules. Section IV presents the methods to conduct experiments and the novel procedure to analyze pick-and-place robotic systems. Section V provides the results and the discussion follows in Section VI.

II. Background

The literature on robotic automation in meat processing is limited, with a few notable studies focusing on specific aspects of meat handling [7], [8], [9]. Jorgensen et al. [10] presented a robotic system for pick and place operations of pork meat using suction cups. In [11] the same authors developed a vacuum gripper for picking and placing meat products by the robot. Joffe et al., [12] demonstrated an approach to handling poultry products that allowed picking up a whole chicken from an unordered bin using a suction cup gripper and placing it in a canonical orientation. However, these studies primarily evaluated the performance of individual modules rather than the entire system.

When it comes to assessing robotic systems, current literature shows two approaches: (1) the performance of a whole system is analyzed, without paying too much attention to the performance of individual modules in that system, or (2) the performance of individual module(s) is evaluated without translating the results to a performance of a total system. A clear example of the former approach is in the Amazon picking challenge [13]. The evaluation methods used in the Amazon picking challenge also focused on overall system performance without considering individual module performance [14]. However, an overall score of the performance does not measure the performance of individual modules comprising the system and is therefore not useful to understand where a system fails and where to pay attention in future research to improve the technologies to solve real-world problems.

Examples of the latter assessment approach, in which individual modules are evaluated and not the whole system, can be found in research on meat processing systems. For example, Jorgensen et al. [10] measured the performance based on the score that comprises three sub-scores for placing the meat at the desired position and orientation, and ensuring the safety of the product. In [11] the authors manually analyzed the performance of suction cup based system for grasping meat pieces and identified three categories of failures due to vacuum loss during lifting, vacuum loss during transferring, and when multiple objects were lifted during the grasp. However, they did not analyze the performance of the whole system. In [15], the author investigated the gripping mechanism for manipulating deformable meat products in the industry based on the criteria of maintaining safety (preventing visual and physical damage to the product) and hygiene standards. In their work Joffe et al., [12] did not show the evaluation method for the entire pipeline either but only evaluated the picking and placing performance considering whether an object was picked up or not by the suction cup and whether the breast of the chicken was facing the picking area and the neck of the chicken was pointing up towards the ceiling, respectively.

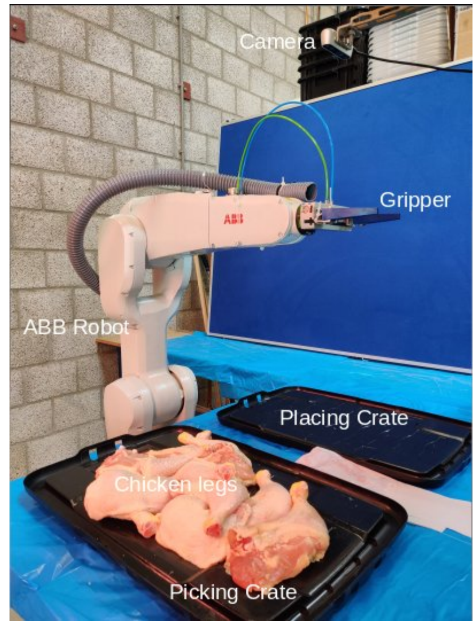


Fig. 2: Experimental setup for picking and placing of chicken pieces from a crate.

It appears in research on robotic automation of meat processing that a common evaluation process assessing both the performance of individual sub-processes as well as the overall system performance is lacking, and such a method could help to get more insight into the different sub-processes in the whole pipeline. In related robotics research on greenhouse automation Van Henten [16] and Bac et al., [17] proposed a robotic system for harvesting vegetables and their assessment procedure. Building on their work as a second contribution of this paper, we proposed and demonstrated an assessment procedure that evaluates both the individual modules as well as their contribution to the overall performance of the whole system for bin-picking robots for meat processing which could provide important insights into the performance of the whole system.

III. Materials and Methods

A. Experimental Environment

The pick and place environment was created in a laboratory setup shown in Figure 2. The chicken pieces were randomly positioned in a black crate having dimensions of 580 x 380 x 30 mm. There was ambient illumination only; no additional lighting was used. The set-up was designed to replicate a future industry set-up, that is, a robotic arm was installed next to the bin so that the bin was reachable by the robot to pick up objects and place them at a distance from one another on a conveyor belt. In order to obtain a top view of the bin, a camera (Model: Intel RealSense D435, 2017) was installed at the top of the bin at a height of 740 mm. A 6 DoF manipulator (Model: ABB IRB 1200) was placed at a distance of 250 mm and 80 mm away from the bottom-left corner of the crate along the x-axis and y-axis of the robot so that it could reach everywhere inside the crate. A gripper (Model: Festo Adaptive gripper fingers DHAS-ME-H9-120) was mounted at the end-effector of the robot which was used for picking and placing the chicken pieces.

B. Objects characteristics and variations

The experiments were performed at room temperature with two different samples of chicken pieces, one containing light-weight pieces and the other containing heavy-weight pieces. These chicken pieces exhibited variations in terms of sizes, shapes, and weights, as detailed in Section III-A. The heavy sample contained pieces having lengths ranging from 22 cm to 24 cm and widths ranging from 7 cm to 10 cm and a weight range from 330 grams to 370 grams; the light sample contained pieces having lengths ranging from 16 cm to 18 cm and the width ranged from 5 cm to 7 cm and the weight ranged from 280 grams to 320 grams. The utilization of different samples allowed us to analyze the potential influence of size and shape on the system's performance.

(a) (b)

Fig. 3: (a) Undesired mark on the chicken leg after picking it by the suction cup, (b) grasping chicken pieces by the 2-fingers Robotiq gripper was not very stable.

C. Hardware and software components of the systems

The robotic system was developed as part of the research project called FlexCRAFT which stands for "Cognitive Robots for Flexible Agro Food Technology" [6]. A detailed description of the hardware and software components of pick-and-place robots and a discussion of choices for each component is given in the following sections.

1) Hardware components: The robotic systems for picking and placing objects mainly consisted of three subsystems: a manipulator, an end effector, and a sensing system.

- *Manipulator* We used the ABB IRB 1200 manipulator having 6 degrees of freedom. The 6 degrees of freedom provide the robot with the ability to move in 3-dimensional space and to allow free positioning and posturing of the tip of end effectors to cope with jobs. This capability is very useful for picking and placing objects. Moreover, as it is not a redundant manipulator, analytical solutions to the inverse kinematic exist. Hence its calculation becomes simpler.
- *End effector* The chicken pieces are naturally deformable objects of varied shapes and sizes. To manipulate such kinds of objects without damaging them requires a robust, highly flexible gripper. [15] reviewed unilateral grippers for a robot to use in meat processing. However, unilateral gripping devices such as magnetic, needle, and adhesive gripers were disregarded as ineffective, damaging to the product, or unhygienic. To the best of our knowledge, at present, the gripper which can safely manipulate deformable poultry pieces is very rare in the market.
  We experimented with different grippers available in the market to pick and place chicken legs, for example, the suction cups, 2-fingers Robotiq gripper, and the Festo Fin Ray grippers. However, experimentally, it has been found that the suction cups are not suitable for picking chicken pieces as they leave undesired marks on the piece that is referred to as *visual damage*, as shown in Figure 3 and it does not fulfill the requirement of maintaining hygiene. Suction-type grippers based on Coanda and Bernoulli principles, including hybrid designs incorporating novel additions, have also been discounted due to hygienic issues in the meat processing environment [15]. The design of the robotic gripper is also not very suitable for manipulating chicken legs as its grasping is not as firm as required to transport chicken pieces. Even, there is more chance of physically damaging the product as it is very rigid. None of the grippers are 100% successful in picking and placing chicken pieces. Even, there are other grippers available from the company Soft Robotics such as mGrip™ [18] that could be useful to handle deformable meat pieces. However, they are very expensive compared to others.
  In this paper, we chose the Festo gripper (Model: Festo Adaptive gripper fingers DHAS-ME-H9-120) as an end-effector for the experiments as it is adaptive to shapes and sizes. This adaptive gripping can be successfully applied in food processing applications to process products of varying sizes and contours. A particular characteristic of the Fin Ray Effect® is the structure's ability to adapt to different component contours. Pressure-sensitive workpieces, in particular, can be displaced and deposited without damage.
  This simple Fin-ray gripper can reliably pick chicken legs from the narrow, deep shelf bins. The simple shape and thin fingertips of the end-effector reduce the need for complex collision avoidance or pre-grasp object manipulation, as it easily fits in between objects, pushing them aside if necessary. This simple choice for the end effector illustrates that an appropriate embodiment simplifies different aspects of the overall solution, including perception, planning, and grasping.
- *Sensors* An RGBD camera (Model: Intel RealSense D435) was used to capture both the RGB and the depth image of the crate from the top.
  In order to integrate all the hardware components, an interface was provided by a computer. The ABB IRC5 robot controller was connected to the computer through the gigabit Ethernet (GigE) interface to control the manipulator. The gripper was controlled via an Arduino, and it was directly connected to the Arduino board via a simple breadboard circuit. Serial communication between the Arduino controller board and the computer was done through a USB. The camera was connected directly to the computer through a USB cable.

2) Software Components: A generic modular software framework for the development of picking and placing robots was developed as part of the FlexCRAFT project. The goal was to provide a modular generic high-level functionality by structured programming, thus leading to faster and simplified development of robots. In theory, this allows users to have multiple modules - but they are different approaches to accomplishing the same task and they can be called whenever required. It becomes easier to remove an existing module or add a new module to be called in the main program. The software framework allows sequential/parallel execution of different modules to perform complex tasks. Each module was written in C++ or Python language.

A Linux (version: Ubuntu 20.04) operating system was installed on the computer with the necessary modules to connect to external devices. A Robot Operating System (ROS, Noetic) was utilized as a middleware running on Ubuntu to communicate between hardware and software modules while performing parallel execution of different modules of the robot. Each module was created as a ROS node that was able to communicate with other ROS nodes through a topic and respond to inputs. To perform the motion planning a MoveIt package was integrated with ROS and for the simulation and visualization, a Gazebo platform was used. These packages/platforms communicate with the system by digital I/O interface.

However, it is not always simple to develop code to manage these many modules to communicate with different devices. We designed a generic state machine that organizes and regulates the execution sequence of each module or function by operating its own ROS nodes. In each stage of the state machine, a task must be finished by one or more ROS nodes. The state machine managed and transfered data to other nodes as input as necessary. A node is actually just executable software that is part of a ROS package. To communicate with one another, ROS nodes employ a client library. Nodes have the ability to publish or subscribe to Topics. Services may also be provided by or used by Nodes. Depending on the condition or situation, the state machine made a transition to one of the possible next states. Implementing code and error handling functionality was made simpler by centralizing this transition mechanism into a general system component. The status of the software system was regularly checked by a performance monitor. An error handler determined the best course of action to do when an error was discovered, such as pausing the program and informing the user via an error message or resetting a node, thus realizing the performance of the whole system.

D. System Architecture

A novel system architecture was designed for a fully autonomous robot that was developed for picking chicken pieces from a bin of cluttered scenes to segregate and place them in order. The system architecture integrates four main modules: (A) perception module, (B) world modeling module, (C) motion planning and control module, and (D) gripping module, which addresses the specific challenges associated with the picking and placing of chicken legs.

- Perception: This module process sensor data and interpret it within the context of the task description, allowing the robot to understand its surroundings. For example, detecting and identifying the objects that has to be picked up by the robot, localizing them in the crate, and determining the corresponding real-world coordinates of the objects. It extracts relevant visual features and provides essential input to the subsequent modules.
- World Modelling: This module acts as a bridge between the other modules, storing and exchanging various types of information about the robot's environment, which other modules can read or write. It constructs a comprehensive representation of the environment, taking into account both static and dynamic elements. It incorporates information from the perception module, such as the geometry and deformability of the chicken legs, the cluttered bin, and other objects in the vicinity. This representation enables the robot to reason about the state of the environment, predict the behavior of the objects, and make informed decisions during the pick-and-place process.
- Decision Making and Planning: This module help the robot decide what actions it should perform at each moment to achieve its tasks effectively. For example, it generates collision-free trajectories for the robot's manipulator to reach and grasp the chicken legs. It utilizes the information from the world modeling module to plan efficient path that consider task-specific objectives.
- Gripping: It takes into consideration the information from the world modeling modules to determine the optimal grip configuration. It focuses on the control of an adaptive gripper or end-effector to securely grasp the target chicken legs. It takes action to open the gripper at appropriate time to grasp the target and close the gripper to release the target when it reached the desired placing location.

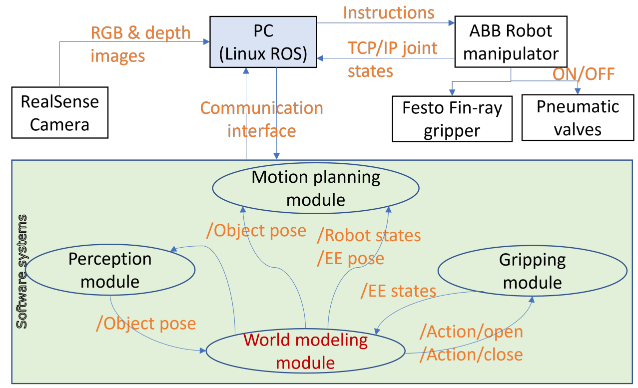


Fig. 4: The system architecture, a Task-Execution-Knowledge (TEK) framework. It emphasizes on the tasks that are translated into physical actions through the execution of skills, which are based on the robot's knowledge and capabilities. It highlighting the importance of task execution and knowledge-driven decision-making. In essence, the figure serves as a visual representation of the system's architecture and interactions without directly representing the software components themselves.

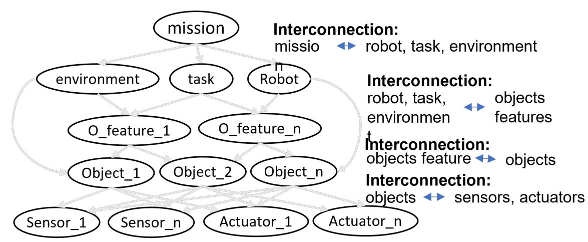


Fig. 5: The data association/interconnection hierarchy of dependencies between different components in the robotic systems.

The Figure 4 represents the robots system architecture, a *task-skill-motion* paradigm, that is how a robotic system connects the robot's task (what it needs to do) to its physical motion (how it performs the task), that is facilitated by the robot's *knowledge/skill*, which helps the robot make the right decisions at the right time. The *skill* is a set of interactions between hardware and software activities that enable the robot to realise above four mentioned responsibilities. In summary, the figure illustrates how the robot's task, knowledge, and motion are interconnected through discrete and continuous control, perception, world model, and gripping modules, enabling the robot to carry out its tasks efficiently and effectively. Figure 5 show how behavioral interactions occur at different levels of modeling the world:

- Actuators and sensors form the basic level, linking the robot to the physical reality. This level represents the "smallest world" where the robot operates.
- The next level involves models of relevant objects in the real world, containing information that needs to be connected to the sensors and actuators.

- The subsequent level relates these objects to (i) the robot itself, (ii) other objects in the environment, and (iii) the task requirements.

In summary, the figure demonstrates how different levels of modeling the world interconnect to enable the robot to perform tasks effectively, with a focus on sensor-actuator interactions, object modeling, task requirements.

These interconnections/data associations within the robotic system architecture enables a coherent and effective framework for the picking and placing of chicken legs. The architecture follows a modular design that enables seamless integration and interaction between modules, offering important system features.

- Modular and Scalable Design: The architecture follows a modular design, allowing for the seamless integration of the perception, world modeling, motion planning, and gripping modules. The novel contribution lies in the development of a scalable architecture that facilitates the addition of new modules or functionalities as the requirements evolve, making it adaptable to future advancements.
- Real-Time Decision Making: The architecture enables real-time decision making by integrating the perception, world modeling, motion planning, and gripping modules in a coherent manner. The novel contribution lies in the development of efficient communication protocols and data exchange mechanisms between these modules, allowing for timely and synchronized decision making to adapt to the changes in the environment.

Figure 6 shows a task sequence or flowchart diagram of the picking and placing robot. At the very beginning of an experiment, we manually checked if the computer was able to establish connections with other hardware devices such as the robot controller, the gripper controller, and the camera. Once everything was okay, the algorithm initialized the robot manipulator at the home position such that the picking crate was fully visible to the RealSense camera mounted on the top to capture RGBD images of the picking workspace. The RGBD images were registered and the RGB image was passed to the object detection and classification algorithm (III-E.1) to segment individual objects in the crate, and classify their poses whether they are facing up or down. The detected and classified fully visible pieces on the top, their mask and the classification results were fed as input to the next level pose estimation algorithm. At this stage, we utilized the depth image and fed it as well as input it to the pose estimation algorithm. The pose estimation algorithm provided the 3D position and 3D orientation of each detected fully visible piece. Then, based on the strategy algorithm selected the target piece to be picked up by the robot. Once, the grasp pose was estimated for the target piece based on the object pose and the gripper pose, the motion planner planed the motion of the manipulator from its current pose to the target pose. Before executing the motion the system checked the status of the gripper and ensured that it was open. The robot executed the motion to the target object position. Once it reached, its gripper closing action was triggered to grasp and picked up the target object and moved it to the placing position. Once the robot manipulator reached the desired placing position, the gripper opening action was triggered to release the object. Hence, it completed one cycle and then the algorithm went back to check if there was any object found in the crate and repeated this process until the crate was empty or there was no object detected by the robot.

Overall, the proposed architecture provides a systematic and comprehensive approach to address the challenges of picking and placing chicken legs. By combining perception, world modeling, motion planning, and gripping modules, the architecture enables the robot to perform delicate and precise manipulation tasks in real-world scenarios, with potential applications in the food industry and other domains requiring dexterous object handling.

E. Proposed Algorithms for Software Modules

In this section, we provide the details of underlying methods for each of the four modules described in the previous section.

1) Perception module: The perception module consisted of three sub-tasks: (1) object detection and classification, (2) object pose, and grasp pose estimation, and (3) target selection.

- *Object detection and classification* The object detection and classification module was continuously running to detect objects in the crate using the color images captured by the RealSense camera. Object detection and classification basically involved bounding box detection and mask detection for each class of objects. In recent times, the most popular way to detect objects is to apply deep convolutional neural networks (CNN) algorithms such as Faster R-CNN [19] achieves high accuracy but is slower and resource-intensive, and YOLO [20] enables real-time object detection, while SSD [21] balances accuracy and speed. U-Net specializes in medical image segmentation but is limited to instance segmentation tasks. Mask R-CNN [22] and YOLACT [23] offer accurate object detection and instance segmentation but sacrifice real-time performance. YOLACT++ [24] improves upon YOLACT with enhanced performance, while Poly-YOLO excels in handling objects with arbitrary shapes. Each algorithm possesses unique merits and demerits as listed in Table I, catering to specific needs in object detection and instance segmentation applications. As in semantic segmentation, the multiple objects are treated in a single category as one entity and in the use case, on the other hand, we needed to identify individual objects within these categories (fully visible top, fully visible bottom, partly visible top, partly visible bottom. An example of all these categories is shown in Figure 7).
  Hence, mask R-CNN is a preferred choice as accurate object detection and precise instance segmentation are of paramount importance for the our application of bin picking of chickens, although we have to compromise with real-time performance. Its ability to provide pixel-level segmentation and handle complex scenes makes it suitable for applications as detailed understanding of object boundaries is crucial.
- *Object pose and grasp pose estimation* The RealSense camera recorded images in color and depth at the same resolution. As a result, the depth image was directly covered with a mask created from the color image, and extracting the point cloud belonging to the mask resulted in the surface of the chicken piece that was visible to the camera. This was done to obtain the segmented point cloud for each individual chicken piece. This point cloud contained noise as a result of the mask quality, the effect of infrared light on the depth measurements, and areas of the image that were outside the depth range that the camera was calibrated on. The algorithm used an outlier removal technique to remove the noise and extract the point cloud that represented the chicken piece. To estimate the location of the chicken piece, the corresponding (x, y, z) coordinates of the point cloud were averaged, i.e. we calculated the mean of the point cloud associated with each chicken piece. The 3D pose of the object was calculated using the principal component analysis (PCA) of the extracted point cloud. The PCA uses singular value decomposition (SVD) to find the eigenvectors. The eigenvector associated with the largest eigenvalue represents the orientation of the chicken piece along the x-axis. This eigenvector passes through the midpoint (x, y, z) and

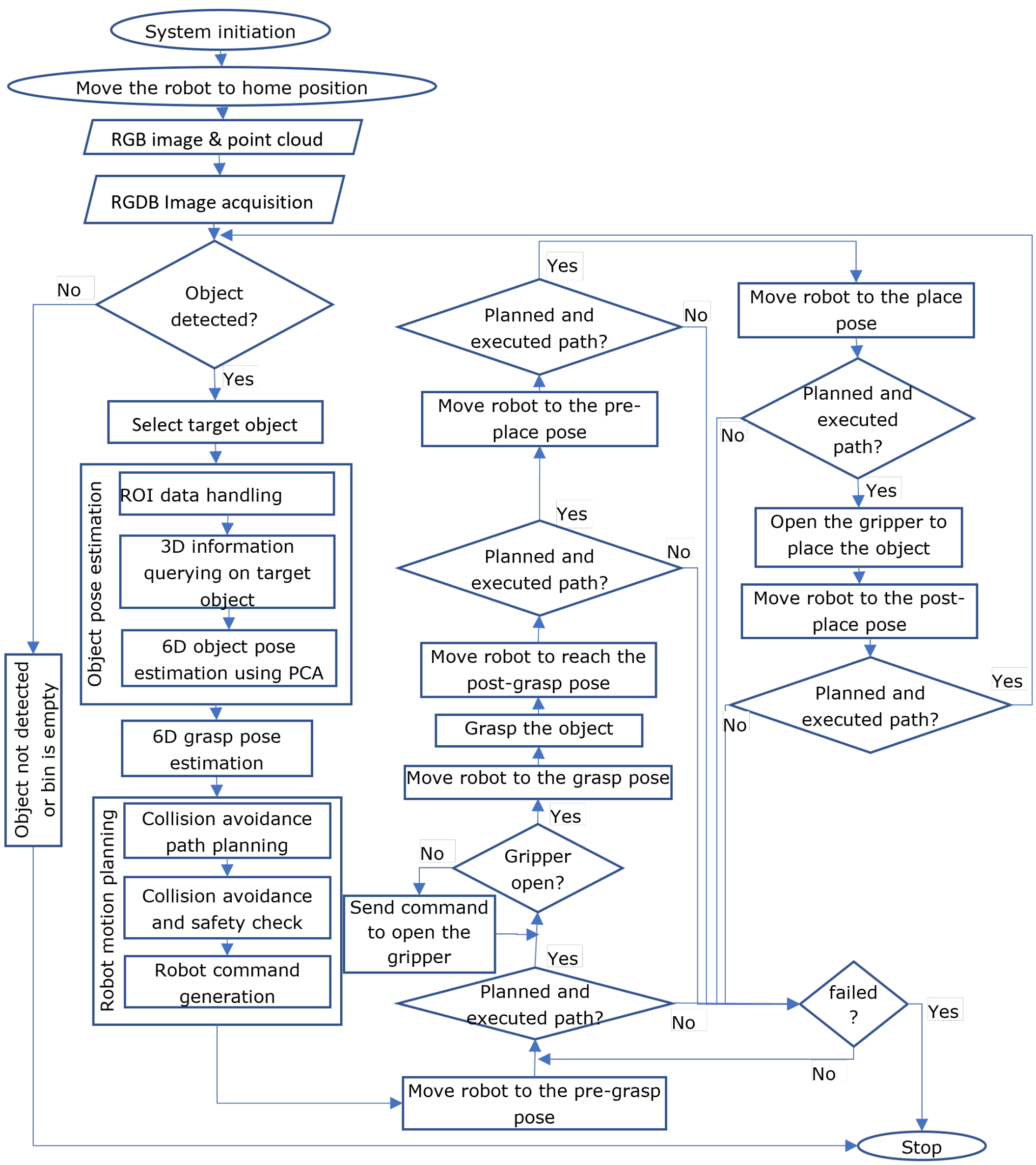


Fig. 6: Flowchart diagram of the task sequences for picking and placing the chicken pieces by the robot.

aligns with the outermost point on the drumstick side of the chicken leg. The second eigenvector, perpendicular to the first, represents the width of the chicken pieces. Finally, the third eigenvector signifies the thickness of the pieces. A more detailed explanation of the algorithm is described in our previous work [25].

The grasp pose was calculated based on the estimated 3D pose of the object. We rotated the object pose by 180^o about the x-axis to get a grasp pose of the end effector. However, the algorithm used only the angle calculated in the horizontal plane (XY plane), that is, the angle about the Z-axis of the end effector with respect to the position of the object. Considering the rotation angles about the X-axis and Y-axis of the grasp pose resulted in the worst grasping and picking performance. Probably, solutions to estimate the object pose under conditions of occlusions, cluttered and varying shapes and sizes are not very reliable. The grasp poses were converted from the camera coordinate frame into the world coordinate frame of the manipulator, using a transformation matrix between the camera coordinate frame and the world coordinate frame T_{world}^{camera} obtained using hand-eye calibration [26],

$$T_{world}^{grasp} = T_{world}^{camera} \cdot T_{camera}^{grasp}, \tag{1}$$

where T_{world}^{grasp} represented the transformation matrix between the world coordinate frame to grasp pose of the object coordinate frame; T_{camera}^{grasp} was the transformation between the camera coordinate frame and the grasp pose of the object coordinate frame. Thus, it calculated the necessary transformations for transforming image feature coordinates into real-world coordinates. The transformation matrix between the camera coordinate frame and the world coordinate frame T_{world}^{camera} was

$$T_{world}^{camera} = T_{world}^{base} \cdot T_{base}^{tool0} \cdot T_{tool0}^{camera} \tag{2}$$

where T_{world}^{base} was the transformation between the world coordinate frame and the base coordinate frame of the robot; T_{base}^{tool0} was the transformation matrix between the

TABLE I: Differences among state-of-the-art object detection algorithms with respect to the application.

Algorithm	Year	Architecture	Object Detection	Instance Segmentation	Real-time	Backbones Supported	Merits	Demerits
U-Net	2015	CNN	No	Yes	No	Customizable	Specialized for medical image segmentation, effective for small dataset	Limited to instance segmentation
Faster R-CNN	2015	CNN + R-CNN	Yes	No	No	VGG, ResNet	High accuracy	Slower than single-stage detectors, requires more computational resources than YOLO
SSD	2016	CNN	Yes	No	Yes	VGG, ResNet, etc.	Real-time, good accuracy	Struggle with small object and tends to generate false positives
Mask R-CNN	2017	CNN + R-CNN	Yes	Yes	No	ResNet, ResNeXt	High accuracy, instance segmentation, pixel-level segmentation, handles complex scenes	Relatively slow inference speed, requires significant computational resources
YOLACT	2019	CNN	Yes	Yes	Yes	ResNet-101	Real-time, instance segmentation, high performance, single-stage architecture	May have reduced accuracy compared to two-stage methods like Mask R-CNN
YOLACT++	2020	CNN	Yes	Yes	Yes	DarkNet-53	Improved performance compared to YOLACT	Still Relatively less accurate compared to Mask R-CNN and requires large GPU memory
Poly-YOLO	2021	CNN	Yes	Yes	Yes	CSPDarkNet-53	Real-time instance segmentation with high accuracy, handles objects with arbitrary shapes	May have reduced performance for small objects compared to other algorithms, memory-intensive

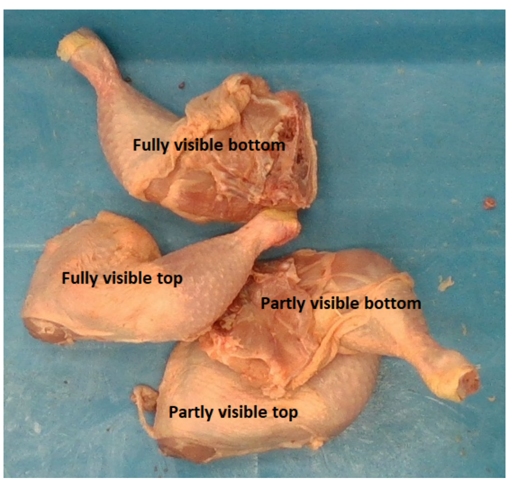


Fig. 7: An example of four different classes of poultry pieces: fully visible top, fully visible bottom, partly visible top, and partly visible bottom.

base coordinate frame and the end effector coordinate frame of the robot; T_{tool0}^{camera} was the transformation matrix between the end effector coordinate frame to the camera coordinate frame. Therefore transformation matrix T_{world}^{camera} remained fixed throughout the experiment as the camera was static with respect to the world.
It is wise to mention that the possible damage could occur to the costly robot grippers and the robot itself if the grasping position or orientation of the object is estimated and given incorrectly. Therefore, to minimize potential hazards to the property workspace constrains had been introduced by using "safe" maximum input levels.

- ***Target object selection* Among all the detected objects in the bin, we were required to select one target object to be picked up by the robot at a time, which raised the challenge of how to choose the best one to pick out of all the detected objects. Hence, we proposed a strategy to pick the object which was on the top of the pile and fully visible, considering that would be the easiest for the robot to pick.**
The detection and pose estimation algorithm provided the estimated pose for all the fully visible objects in cluttered scenarios in the crate. The object which was on the top of the pile was the one that is closest to the camera along the z direction. Hence, we calculated the distance between the camera's z-position (C_z) and the positions of all the objects that were completely visible (O_z^i) and selected the one with the minimum distance value by

$$target_{object} = min(C_z - O_z^1, C_z - O_z^2, ..., C_z - O_z^v) \quad (3)$$

where $i = 1, ..., v$, and v was the number of detected fully visible objects at time t.

2) World modeling module: **Current robotic systems are pre-programmed to deal with very specific tasks on a limited set of objects very well-defined in terms of location, shape, size, and material properties. However, to deal with variability and enable flexibility, robotic systems need to reason about the objects in their environment or world. Hence a representation**

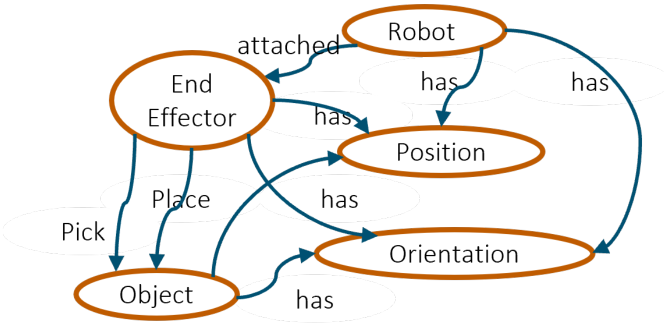


Fig. 8: Schematic diagram of the world model representation of the environment.

Fig. 9: Example picture of the Festo gripper that adapted to the shape of the chicken piece to firmly grasped it during the picking and placing operation in one of the experiments.

of the world is required, i.e., *world modeling* is defined as the process of creating a *numerical model* of a real-world environment, or workspace. This can be graphically displayed to provide the user with a 3D surface model of the workspace for simulations, analysis, and task planning [27]. In this work, a numerical task-centric world model was developed to interact with the environment shown in Figure 8. We stored information about the (static) obstacles in the workspace; the state of the robot, the end effector, the gripper; the information related to the objects in a numerical form. For example, if the state of the gripper was open it was numerically represented as 1, and if it was closed that state was represented as 0, and that information was stored in the world model and used by the gripping module whenever necessary. Similarly, the configuration of the manipulator was represented by Θ and the end effector position was represented by x, y, and z in a 3D space. The id and grasp pose of objects were stored over time to assist the perception module and task planning module in the presence of occlusion and variation.

3) Motion planning module: In the case of industrial manipulators, motion planning refers to providing suitable joint angle position (or velocity) trajectories to move the robot from one pose to another. Many state-of-the-art motion planning algorithms are available in the form of the open motion planning library (OMPL) [28] which has been integrated into several easy-to-use software packages like Moveit! [29]. In this paper, we used the Moveit! package in integration with ROS for building motion planning algorithms for ABB IRB 1200 robot manipulator. The simulation was carried out using Gazebo [30] environment. The motion planning for the picking and placing task involved a sequence of a few steps.

Once the first target object in the crate was detected, the estimated grasp pose of the target was sent to the Moveit! planner that planned a path from the current pose of the robot to the desired estimated pre-grasp pose above the target object. The Moveit! planner used the Bi-directional Rapidly-exploring Random Trees [31] algorithm to plan a path from initial pose to the desired goal pose. Then the robot started executing the path from its home pose to the pre-grasp pose. Then, from the pre-grasp pose it moved to the estimated grasp pose to grasp and pick the object and back to the post-grasp pose above the grasp pose. Then, the robot moved from the post-grasp pose to the predefined pre-place pose to the place pose to place the object. The placing pose of the object was predefined. After placing the object, the robot moved to the post-place pose. Meanwhile, if the next target object was found in the crate, the robot moved to the pre-grasp pose and repeated the process until the crate was empty. Else, the robot moved to the home pose and stopped the process.

4) Gripping module: The gripping module focuses on controlling the selected Festo Fin Ray gripper while picking and placing chicken pieces using robots. The gripper was operated through a pneumatic valve which was controlled using a microcontroller enables precise control and coordination of the gripper's actions. The microcontroller receives input signals from the personal computer (PC), processes them, and sends corresponding commands to the pneumatic valve, determining the gripper's opening and closing actions at appropriate time. This integration of technology allows for automated and programmable operation of the gripper enhances the efficiency and reliability of the pick-and-place process, reducing the risk of product quality issues and ensuring the integrity of the transported chicken pieces.

It is crucial to apply appropriate amount of force in handling delicate and deformable chicken pieces to prevent excessive force that may cause deformation or damage. Hence, we define deformability constrains as below:

- Determined Acceptable Deformation: defined the acceptable limits/thresholds for deformation considering acceptable amount of flattening or bending that can occur during the grasping process without compromising the overall quality or appearance.
- Established Force-Deformation Relationship: determined the relationship between the grasping force applied by the robot and the resulting deformation of the object. This involves experimental testing to measure the force-deformation behavior using Hooke's law [], that states a change in shape due to the application of a force is a deformation. Even very small forces are known to cause some deformation is written by,

$$F = k\Delta L \tag{4}$$

where ΔL is the amount of deformation produced by the force F, and ΔL. The deformation can be along any axis and it is proportional to the applied force. Based on this relationship, the maximum allowable force that the gripper can exert during grasping was set to stay within the predefined deformability constraints.

Figure 9 shows the Festo gripper (Model: Festo Adaptive gripper fingers DHAS-ME-H9-120) transporting chicken pieces by grasping it firmly.

IV. Evaluation and Experimental Procedures

A. Experimental testing scenarios

The experiments were conducted for both the light and heavy samples (as described in Section III-A) under two different

(a) (b)

Fig. 10: Example of (a) simple and (b) complex scenarios of chicken pieces in a crate.

testing scenarios: 1) a complex scenario, and 2) a simple scenario depending on the placement of chicken pieces in the bin. These scenarios were designed to assess the impact of occlusions and clutter on the system's performance. Each scenario was repeated five times for both the light and heavy samples. In the simple scenario (Figure 10(a)), 8 pieces were taken in total and randomly placed in one layer in the crate touching each other, whereas, for the complex scenarios (Figure 10(b)), 10 pieces were taken in total and randomly placed in three layers so that the pieces were overlapping and some pieces partially occluded other pieces. In total 20 sets of experiments were conducted for both samples, out of which 5 were with simple scenes and 5 were with complex scenes for each sample set.

B. Guidelines of the picking and placing task

A well-defined set of guidelines was established for the picking and placing task during the experiments, as follows:

- The robots were tasked to automatically pick one chicken piece at a time from a pile in the crate and place them into the desired location on another crate and repeat the process to empty the bin.
- In case of any failure at any stage, starting from detecting an object to placing an object during the execution, the detected target chicken pieces at that cycle were removed manually and the algorithm was continued to pick up the next object.

Subsequently, the system was evaluated based on identified sub-tasks (Section IV-B.1), and its performance was assessed at each step using various evaluation criteria. A specific protocol was followed, encompassing the recording of performance indicators (Section IV-B.2) and categorizing successes or failures (Section IV-B.3). These recorded measurements were employed to conduct experiments and determine the overall performance of the robotic system (Section IV-B.2).

1) List of sub-tasks: Nine distinct sub-tasks were identified for which the system's performance was evaluated, as listed below:

1) The robot must correctly identify and classify at least one chicken piece that lies on the top of the pile.
2) The robot must select the target chicken piece to be picked.
3) The robot must correctly estimate the pose of the target chicken piece.
4) The robot must correctly estimate the grasp pose of the end effector or the gripper.
5) The robot must reach the desired grasp pose of the target chicken piece.
6) The robot must grasp the target chicken piece firmly.
7) The robot must pick the chicken piece.
8) The robot must transport the chicken piece to the desired placing pose.
9) The robot must properly release the object at the desired placing pose.

2) performance indicators: The evaluation process employed the following performance indicators:

1) Object detection success (%): The number of chicken pieces successfully detected in the crate as a fraction of the total number of chicken pieces in the crate.
2) Target Selection success (%): The number of times the algorithm successfully selected the target piece per total number of chicken pieces in the crate.
3) Object pose estimation success (%): The number of times the algorithm successfully estimated the pose of the target chicken piece per total number of chicken pieces in the crate.
4) Grasp poses estimation success (%): The number of times the algorithm successfully estimated the grasp pose of the target chicken piece per total number of chicken pieces in the crate.
5) Reaching Target (%): The number of times the robot was able to plan the path and reach the desired target location per total number of chicken pieces in the crate.
6) Grasping success rate (%): The number of times the robot was able to grasp the target object successfully per total number of chicken pieces in the crate.
7) Picking success rate (%): The number of times the robot was able to pick the target object successfully per total number of chicken pieces in the crate.
8) Transporting success rate (%): The number of times the robot was able to transfer the target object successfully from its post-grasp position to the pre-placed position per total number of chicken pieces in the crate.
9) Placing success rate (%): The number of times the robot was able to place the target object successfully per total number of chicken pieces in the crate.
10) object damage rate (%): The number of damaged chicken pieces per total number of chicken pieces in the crate. A chicken piece was consider damaged if the robot cut more than 5 mm into the chicken piece or caused a bruise. Measuring meat damage was highly relevant for economic feasibility because a producer cannot market damaged meat.
11) Cycle time (s): Time of an average full picking and placing operation of one chicken piece, including object detection, pose estimation, target selection, reaching, grasping, picking, transporting, and placing the target at the desired location per total number of chicken pieces in the crate.

3) Failure Categories: The failures during the picking and placing attempts were categorized as follows:

1) Incorrect object detection: a) chicken piece was partially detected or b) if two or more pieces were detected as one, or c) if no piece was found.
2) Poor target selection: When there existed a fully visible piece that was easier to pick than the target piece. For example, the strategy to select the target piece was based on the highest point cloud among all detected fully visible pieces as discussed in Section III-E.1. However, this does not always give the best results, especially when the piece was tilted and some parts of the point cloud of the piece were at a higher level but the piece was cluttered, i.e., surrounded by other pieces, and was then difficult to grasp.
3) Incorrect object pose estimation: if the object pose was not estimated or the estimated object pose was not in the range of ground truth as discussed in Section III-E.1.

4) **Poor grasp poses quality:** (a) if the estimated grasp pose was not in the range of ground truth as discussed in Section III-E.1, (b) if the object did not fit into the gripper based on the pose, (c) if the robot picked more than one object at a time.
5) **Failed to reach grasp:** if the robot was not able to reach the desired grasp pose due to the inability to plan motion.
6) **Failed to grasp:** (a) if the object slipped while closing the gripper to grasp the object or (b) it failed to grasp due to incorrect action of the gripper motion or (c) while grasping if the the robot hit other objects and damaged.
7) **Failed to pick:** if the object slipped out of hand while moving from grasp pose to post-grasp pose after grasping. This could be caused due to (a) poor gripper design, (b) the target object sticking to the underneath object or the crate below it.
8) **Failed while transferring:** (a) if the object slipped out of the hand/gripper while moving from the post-grasp pose to the pre-place pose or (b) if the motion planner failed to compute a safe path between these two points.
9) **Failed to place:** if the robot did not place an object at the desired place position. In addition, it would be good to consider placing the object in the desired orientation, but we did not consider this for the experiments.

One failure category was assigned to each unsuccessful picking and placing attempt, i.e., where the system failed first out of nine defined categories. Let's say, a sample set contains X number of pieces in total, and the number of pieces that failed in i^{th} category is denoted by X_i. Then the failure rate Y_i for the i^{th} category was calculated as,

$$Y_i = \frac{X_i}{X - \sum_{j=1}^{i-1} X_i} * 100 \tag{5}$$

where i = 1,2,..., n, and n was the total number of categories, i.e., 9 for our case. Equation (5) was used to calculate the failure rate of each category in the experiments. So, out of 10 objects, if the robot failed 0 time in the first 4 categories, 2 times in category 5, i.e., failed to reached grasp, 1 time in the category 6, i.e, failed to grasp, then the failure rate for the category 5 and 6 would be $\frac{2}{10-(0+0+0+0)} * 100$ = 20% and $\frac{1}{10-(0+0+0+0+2)} * 100$ = 12.5%, respectively.

V. Results

The experimental results of the picking and placing of chicken pieces are described in this section. The quantitative results for the experiments of the simple and the complex scenarios for both the light and heavy samples are given in Table II for the light chicken pieces. For the complex scene, during the first experiment, out of 10 pieces, 6 times the robot failed to successfully pick and place chicken pieces. Out of 6 failures, 1 time it failed to detect and segment the object properly, 1 time it incorrectly estimated the pose, 1 time the grasp pose was not good, and 3 times the chicken piece slipped out of the gripper while picking. That means the robot could achieve a 40 % success rate in picking and placing chicken pieces in the first experiment. The success rates for the second, third, fourth, and fifth experiments were 40%, 30%, 50%, and 50%, respectively. On average, it is observed that out of 29 failures, 5 times the system incorrectly detected objects, 3 times failed to select the target, 2 times incorrectly estimated pose, 9 times incorrectly estimated grasp pose, 1 time failed to grasp the object, 7 times failed in picking an object and 2 times it failed to transfer object due to slip. Hence, the average success rate for the complex scene achieved was 42%. Similarly, for the simple scene, during the first experiment, out of 8 pieces, 3 pieces were not successfully picked and placed by the robot. Among these 3 pieces, the robot failed to grasp 2 pieces and 1 piece slipped out of the gripper during transfer to the placing location. That means the robot achieved a 62.5% success rate in picking and placing chicken pieces. The success rate for the second, third, fourth, and fifth experiments achieved was 62.5%, 37.5%, 50%, and 62.5%, respectively. On average, out of 18 failures, 4 times it failed to estimate grasp pose, 2 times it failed to grasp the object, 10 times it failed to pick the object, and 2 times it failed to transfer the object. Hence, the average success rate for the simple scene achieved was 55%. Overall, the success rate for the simple scene (55%) was higher than for the complex scene (42%).

Similarly, Table III shows the experimental result for picking and placing chicken legs for the heavy chicken pieces. The overall success rate for the complex scenes was 50% whereas for the simple scene it was a bit more, i.e., 52.5%. A similar trend with the light sample set was observed in the failure rate in the first three categories of perception for the simple scene, i.e., 0%, however, for the complex scene, there are 3, 1, 1 piece failed in detection, target selection, and the pose estimation categories. In terms of grasp quality, there were 7 pieces that failed in a complex scene compared to a simple scene. This shows that clutter was a serious issue while estimating grasp pose. The highest failure rate was observed in picking the objects for both the complex and simple scenes which were 9 out of 25 failures and 14 out of 19 failures for the complex and simple scenes, respectively. Overall, the robot achieved a success rate of 50.6% for picking chicken legs from a pile to placing them in order. The success rate for the heavy sample set was achieved at 51.2% including complex (50%) and simple scenes (52.5%), and for the light sample set, the success rate was 48.5% including complex (42%) and simple scenes (55%). So there was not much difference in success rate when picking and placing the lighter or heavier chicken pieces. However, it is observed that there was a pronounced difference in the overall success rate between complex scenes (46%) and simple scenes (53.7%), which clearly indicates that the scene complexity highly influenced the performance of the overall systems.

Table IV presents the performance of the individual categories for the experiments for both the heavy and light sample sets. It showed that for the simple scene, there were 0% failure cases in the first three categories of the perception module which include object detection, target selection, and object pose estimation, whereas for the complex scene the failure rate was 8%, 4%, and 3%, respectively. Even the robot failed more to estimate grasp pose for the complex scene (16%) than for the simple scene (7.5%). All these failures are due to the overlap, cluttered, and occlusion conditions in the crate. In terms of motion planning, the robot did not fail to reach the grasp pose till it placed the object for both scenes. However, the robot failed quite seriously whenever the gripper was in action irrespective of the complexity of the scene, i.e., while grasping the object, picking the object, and transferring the object the failure rate for the complex scene was 1%, 33.3%, 10.4%, and for the simple scene, it was 2.7%, 24%, and 5%, respectively. The robot failed to grasp as the gripper pushed the object away from the desired grasp point during the closing action. The robot failed to pick up the object as the object was slipping out of hand while pulling up. Similarly, the object was slipping out of hand while the robot was in motion to transfer the object. Overall, there was no damage to the chicken pieces found throughout the experiments. However, given how frequently the gripper failed to manipulate objects, it is clear that there is an immediate need to work on designing a suitable gripper to manipulate natural, soft, deformable objects like chicken legs as it leads to the highest failure. Moreover, the perception algorithm needs to be improved to deal with overlap, occlusion, and clutter, especially when estimating grasp pose. We demonstrated these

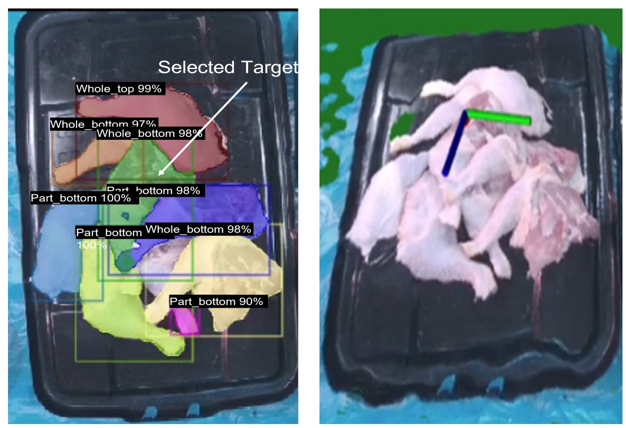
TABLE II: Results for each simple vs. complex scene of Light sample set.

Type of Scene	Simple						Complex						All
Experiments number	1	2	3	4	5	Total	1	2	3	4	5	Total	Total
Incorrect object detection	0	0	0	0	0	0	1	1	1	1	1	5	5
Poor target selection	0	0	0	0	0	0	0	0	2	1	0	3	3
Incorrect pose estimation	0	0	0	0	0	0	1	0	0	0	1	2	2
Poor grasp quality	0	1	1	1	1	4	1	2	2	1	3	9	13
Failed to reach grasp	0	0	0	0	0	0	0	0	0	0	0	0	0
Failed to grasp	0	0	0	0	2	0	0	0	1	0	1	2	3
Failed to pick	0	2	4	3	1	10	3	3	1	0	0	7	17
Slipped while transferring	1	0	0	0	1	2	0	0	1	1	0	2	4
Failed to place	0	0	0	0	0	0	0	0	0	0	0	0	0
Total failures	3	3	5	4	3	18	6	6	7	5	5	29	47
Total pieces	8	8	8	8	8	40	10	10	10	10	10	50	90
Success rate (%)	62,5	62,5	37,5	50	62,5	55	40	40	30	50	50	42	47.7

TABLE III: Results for each simple vs. complex scene of Heavy sample set.

Type of Scene	Simple						Complex						All
Experiments number	1	2	3	4	5	Total	1	2	3	4	5	Total	Total
Incorrect object detection	0	0	0	0	0	0	0	1	0	0	2	3	3
Poor target selection	0	0	0	0	0	0	0	0	1	0	0	1	1
Incorrect pose estimation	0	0	0	0	0	0	0	0	0	1	0	1	1
Poor grasp quality	0	0	0	1	1	2	2	3	1	1	0	7	9
Failed to reach grasp	0	0	0	0	0	0	0	0	0	0	0	0	0
Failed to grasp	0	0	0	0	0	0	0	0	0	0	0	0	0
Failed to pick	3	4	2	2	3	14	1	0	2	3	3	9	23
Slipped while transferring	0	0	2	0	1	3	1	2	1	0	0	4	7
Failed to place	0	0	0	0	0	0	0	0	0	0	0	0	0
Total failures	3	4	4	3	5	19	4	6	5	5	5	25	44
Total pieces	8	8	8	8	8	40	10	10	10	10	10	50	90
Success rate (%)	62,5	50	50	62,5	37.5	52.5	60	40	50	50	50	50	51.1

Fig. 11: Example of incorrect object segmentation, where two pieces are detected as one by the Mask R-CNN deep learning algorithm.



(a) (b)

Fig. 12: Example of poor target selection. (a) Segmented output image of Mask R-CNN of top-view RGB image, (b) RViz visualization of the point cloud image of the same scene showing grasp pose of the selected target piece.

failures at each step of perception as below:

- **Incorrect object detection: Figure 11 displays typical examples of incorrect object detection, where the top two pieces are detected as one piece.**
- **Poor target selection: Figure 12 shows an example of poor target selection. In the scene, the chicken piece on the top was selected as the target piece but that was surrounded by five other pieces in the neighborhood, hence it was harder to grasp. However, it can be seen that there were other fully visible (*whole_top* or *whole_bottom*) pieces placed in the less cluttered area (for instance, the bottom-right most piece) in the scene which was easier to grasp than the selected one. Hence a better target selection strategy/approach is required which should take into consideration the neighborhood pieces while selecting the target piece.**
- **Incorrect object pose estimation: Figure 13 shows an example of incorrect pose estimation of the target object, where the z-axis (red) was pointed downward. However, the z-axis should point upward. The reason for the incorrect estimation of the direction of the z-axis is the misclassification of the object side (top/bottom) from the mask R-CNN algorithm, i.e., the algorithm predicts the bottom side as the top side. It is explained in detail in Raja et. al., [25].**

(a) (b)

Fig. 13: Example of incorrect object pose estimation. (a) Segmented output image of Mask R-CNN of top-view RGB image of the scene, (b) RViz visualization of the point cloud image of the scene showing the object pose.

(a) (b)

Fig. 14: Example of poor grasp quality. (a) Segmented output image of Mask R-CNN of top-view RGB image of the scene, (b) RViz visualization of the point cloud image of the scene showing the estimated grasp pose on the selected target object.

- **Poor grasp quality: Figure 14 shows an example of poor grasp quality. It was not a good grasp pose for the target object as there was another object underneath, and consequently while grasping, the gripper grasped both objects. Of course, the quality of the grasp is highly related to the design of the gripper and how to approach the objects during grasping.**

During the experiments, we kept track of which objects failed at which sub-tasks and recorded that information and presented it in the form of a heatmap in Table V, where the x-axis represents the order of pieces picked and the y-axis represents the sub-tasks. The heatmap represents the results for all 5 experiments for each of the sample sets, Figure V(a), Figure V(b), Figure V(c), Figure V(d) presents heatmap for simple light, complex light, simple heavy and complex heavy sample sets, respectively. The darker color represents less number of failures and the lighter color represents high failures. The number in the map represents the total number of times the robot failed at particular sub-tasks during the picking-and-placing operations conducted for each sample set. That means, the maximum number in the map could be 5, i.e., the robot could fail a maximum of 5 times at each sub-tasks out of 5 conducted experiments and the minimum number could be 0, i.e., no failure. For the complex light sample set, the first piece selected as the target was incorrectly detected in one of the five experiments. The robot failed during the operation on the second object in line in three experiments out of five experiments, one time at poor grasp quality, one time failed to pick and one time slipped while transferring. Similarly, while operating the third piece, the robot failed three times (one time the object was detected incorrectly, one-time target selection was poor and one time the object slipped while picking) out of 5 experiments. While operating the tenth object, the robot failed one time while picking. It is visible from the map that, there was more failure in the perception module at the beginning compared to the later stage of the experiments. It might happen due to occlusion and being cluttered. That means as the robot removes objects from the crate, the complexity of the scene reduces, i.e., there is less clutter, overlap, or occlusion conditions, hence the performance of the perception module gets better at the later stage, which indicates that there is a need for better perception module that is robust to occlusion and cluttered.

The average time taken to successfully pick and place a chicken piece was 16 seconds.

VI. Discussion

The presented research demonstrates a significant step forward in the development of an advanced robotic system for the efficient pick-and-place of deformable poultry pieces from cluttered bins. The achieved overall success rate of 49.4% indicates that the system is capable of handling deformable objects to a reasonable extent. However, the identified failure rates in individual modules, such as perception (21.8%), gripping (30.7%), and manipulation (11%), reveal areas that require further attention for improvement. The practical experiments with the robots to pick and place real chicken pieces and the proposed evaluation method in this study has provided valuable insights into the system's performance, allowing us to pinpoint specific challenges and potential solutions. In the following sections, we discuss lessons learned from the experiments by reflecting on the properties of deformable chicken pieces, that is how the object stiffness or the size and shape affects the overall performance of the systems. Moreover, the discussion will reflect on the four different modules of perception, world modeling, motion planning, and gripping in the pipeline. Also, we would reflect on the proposed assessment procedure that combines the evaluation of all the modules.

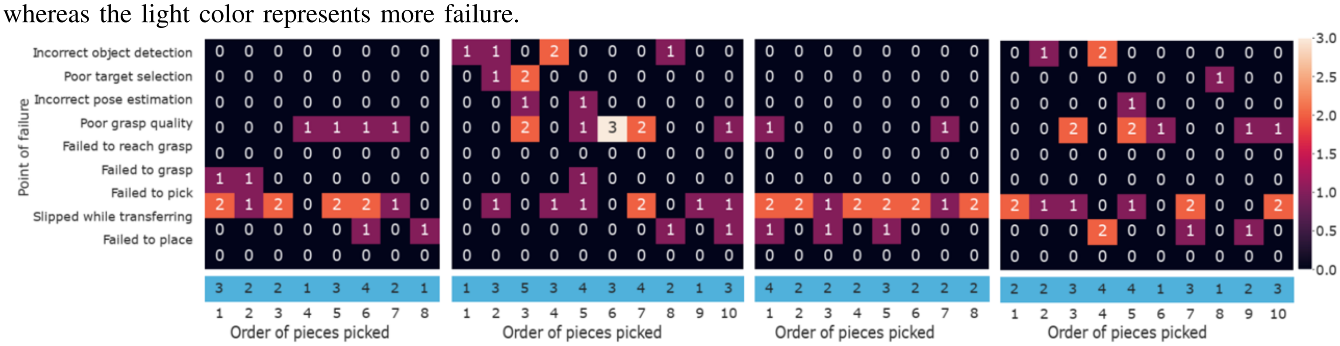
A. Objects

One crucial aspect that emerged from the experimental results is that the influence of cluttered scenarios on the overall system performance. The picking and placing success rate has reduced by 9.7% for the complex scene compared to the simple scene. The presence of cluttered and occlusion was one of the major reasons for the poor success. The effect of size of chicken pieces has a little influence on the success rate. However, the abrupt shape of chicken pieces has a lot of effect on the performance of perception, gripping, and manipulation modules, thus on the overall performance. A task for future work is, therefore, to look for better solutions for all these modules to improve their performance. In addition, there is a need to investigate the effects of the surface properties of poultry pieces on performance. As the surface of the chicken pieces was slippery and its friction properties changed based on stiffness that varies with the temperature and that has an

TABLE IV: Results of failures in individual categories for Heavy vs. Light sample set.

Sample set	Light		Heavy		All		All(%)
Type of scene	Simple(%)	Complex (%)	Simple(%)	Complex(%)	Simple(%)	Complex(%)	
Incorrect object detection	0	6	0	10	0	8	4.4
Poor target selection	0	2.1	0	6.6	0	4	2.3
Incorrect pose estimation	0	2.1	0	4.7	0	3	1.8
Poor grasp quality	5	15.5	10	22.5	7.5	16	13.3
Failed to reach grasp	0	0	0	0	0	0	0
Failed to grasp	0	0	5.5	3.2	2.7	1	2.1
Failed to pick	36.8	23.6	29.4	23.3	24	33.3	28.6
Slipped while transferring	12.5	13.9	8.3	8.7	5	10.4	11
Failed to place	0	0	0	0	0	0	0

TABLE V: Analysis of failure at sub-tasks during the experiments based on the order of picking of the objects for the datasets (a) simple light, (b) complex light, (c) simple heavy, (d) complex heavy. The dark color represents less failure whereas the light color represents more failure.



, (a) (b) (c) (d)

effect on the overall performance of the system. The poultry pieces were sticky which also adds to the poor manipulation success rate.

B. Cycle time

Human workers can pick chicken pieces at an approximate rate of 4-6 seconds per piece. In general, reaching that speed for handling man-made objects with an automated solution is likely as much a research problem as it is an engineering one, requiring fine-tuning computations of all algorithms as well as optimizing all robot motions. However, this is not the case for naturally deformable objects like poultry pieces. As while manipulation, to reduce the cycle time, velocities of the system need to increase, which might have an effect on the masses, forces, and friction coefficient of the deformable poultry pieces. When these characteristics alter, the ability of the object to be gripped by the gripper will be affected. Hence, in the future, it has to be investigated how the performance with the velocity of the robot and the system has to evaluate based on the time. Speed can be utilized as one metric of development, perhaps driving the choice of robotic mechanisms as well as algorithmic solutions, with the risk of leading the community to premature optimization rather than unorthodox creativity.

C. Perception module

The success rate for detecting, selecting the target, and estimating the pose of chicken pieces in the simple scene showed a remarkable improvement, reaching 100%, compared to 92%, 96%, and 97% in the complex scene. That means there was no failure in the simple scene, but faced challenges in the complex scene due to clutter and occlusions, resulting in some failures. Hence, this is a critical area that requires advancements in object detection and recognition algorithms, especially in cluttered environments. Developing sophisticated perception techniques that can accurately interpret the context provided by the task description is key to improving the overall performance of the system.

While estimating the grasp pose of the target chicken piece, the surrounding environmental situation such as clutter, occlusion, and even the noisy depth information was not considered. To the best of our knowledge, the literature on estimating the grasp pose of chicken pieces considering the surrounding environment is yet to appear. To effectively use the cameras in future research, the issue of poor depth estimates should be addressed. Another explanation for poor depth estimation is due to the lighting condition. The light may have partially reflected on chicken pieces and due to that some of the point clouds were missing or provided with NAN values, which can be overcome by having more uniform illumination systems on the crate. In the industry environment, the illumination can actually be controlled to overcome the issues.

D. World modelling

The world model made for this application was very simple, where the robot was able to successfully store necessary information in the memory and used it when necessary. However, to improve the performance of different modules of the robot, it might be required to build a more sophisticated world model in the future. For example, the robot could have information on the arbitrary shapes and sizes of objects, whether simple or complex, it can be rapidly modeled and store the information to allow a robot to interact with it. Moreover, the world model should be capable to add information on 3D surface maps, or models to form a 3D world model of the target workspace.

E. Motion planning

During the experiments, the path generated by *MoveIt!* planner to move the end-effector from one initial position to the other goal position looks good for the application of picking and placing objects. However, sometimes, MoveIt! generates weird complex motion trajectories for the robot arm even when moving between two closely adjacent points. Due to the bad motion, the object slipped out of hand while transferring from the picking location to the placing location. This affects the performance of the overall result significantly. A better motion planner could improve the performance of the system.

To follow the planned trajectory, the IRC5 ABB robot controller was used, which is able to follow the generated trajectory very accurately to reach the goal position. Hence, the robot could accurately reach the target location to pick the object and precisely place it at the desired location.

F. Gripping and manipulation

A number of issues related to object properties explained the performance of unsuccessful grasp and picking of the chicken pieces by the Festo gripper. However, the design of the end-effector mostly influenced the ability to manipulate the chicken pieces. The gripper was not very effective at handling the object's deformability and slippery nature. We tried to improve the grasping performance by not closing the gripper completely. When the gripper was completely closing it squeezed the object too much and due to that objects commonly slipped out of the gripper. It is desired to find a better gripper in the future which can provide stability by firmly grasping the chicken piece. Similarly, it needs to be investigated how much force is required to grasp the object as currently, we are providing constant force to pick all the chicken pieces. The gripper having a force-feedback capability could improve the gripping performance. Damages to the chicken pieces did not occur which is a very positive aspect of the gripper.

G. Systems and Assessment procedures

The modular design of the system architecture allowed for seamless integration and interaction between modules, facilitating efficient task execution. The combination of modules provided a comprehensive approach to address the complexities of deformable poultry handling. However, it lacks incorporating of learning and adaptation mechanisms, that can enable continuous improvement of the system's performance. This continuous learning and adaptive approach holds promise for creating a highly robust and efficient robotic system for pick-and-place tasks, making it more versatile and capable of handling real-world challenges in the food processing industry and beyond.

One of the notable strengths of this research lies in the proposed assessment procedure that combines the evaluation of all the modules. This comprehensive approach allows for a holistic understanding of the system's performance, offering valuable insights into the interplay between the different components and their collective impact on the overall success rate. Several aspects of hardware and software compatibility were verified using the proposed methodology - the effectiveness of the used 3D vision system, the gripper design, the perception algorithms, the motion planning algorithms, and the execution rate of the tasks. Moreover, during the experimental trial, the evaluation methodology immediately revealed the reason for failures and considerable limitations of that system, e.g., in one of the experiments the object detection algorithm detected two pieces as one, and in another experiment due to the bad motion of the manipulator object slipped out of the gripper. Furthermore, illumination proved to be a challenging task for object detection, etc. This individual reason for failure/success has an effect on the overall performance of the proposed system. We could not have gained these insights without this method of evaluation.

The system components and their assessment procedure can be used by developers or integrators that want to build and quickly evaluate the configuration of a sample bin-picking workstation. The performed experiments can therefore be directly comparable between the selected hardware and software components by various research teams or industries. In the practical context, the method of evaluation could be very useful for the industry. Though, the experimental results indicated that the robotic bin-picking system still lacks the speed, dexterity, and flexibility, possessed by a human worker. Despite all, the proposed systems showed that it is possible to handle deformable objects by the robot considering cluttered bin-picking scenarios. Although in many literature [32], researchers stated that it is even hard to pick and place simple objects that would match the human mind and hand-eye coordination. Further work is needed to improve the performance of the system and fine-tune the evaluation. Please note that different operations require a different assessment.

VII. Conclusion

This research presents an advanced robotic system for efficiently picking and placing deformable poultry pieces from cluttered bins. The modular design of the architecture allowed for seamless integration and interaction between modules, facilitating efficient task execution. The combination of perception, world modeling, motion planning, and gripping modules provided a comprehensive approach to address the complexities of deformable poultry handling. The proposed assessment procedure, which evaluates all modules' performance, proved to be effective in identifying strengths and weaknesses within the system. This evaluation framework indicated an overall success rate of 49.4%. We identified the key problems of the developed system by evaluating the performance against 13 performance indicators, and 9 failure categories by conducting experiments with two different sample sizes of poultry pieces in simple and complex scenes. The experimental results showed that the performance of the robot was improved by 7.7% for a simple scene compared to a complex scene. This difference mainly occurred due to the poor performance of the perception module for the complex scene compared to the simple scene. The performance of the gripping module was poor for both scenes as it highly depended on the interaction between the gripper and the target object. The performance of world modeling and motion planning modules was good for both scenes, i.e, the information was correctly stored in the world model and used by the other modules when required; and the robot was able to appropriately position its end-effector at the desired location. Hence it can be concluded that the gripping module, i.e., the gripper design should get more attention. Additionally, a robust perception module is needed for complex scenes in order to boost performance. Due to the proposed method of evaluation, we could have achieved these valuable insights. Overall, the developed systems have the potential to be deployed in the food processing industry to pick and place objects, and the assessment procedure may offer valuable insights in terms of practical application.

Acknowledgement

This research is supported by grants from the NWO perspective project FlexCRAFT under grant number P17-01. The authors would like to thank Sam Blaauw, Laurens Philippo, Bastiaan Vroegindeweij, David Rapado Rincon, Frans Kemp and Luis Ponce Pacheco for their technical contributions.

REFERENCES

[1] F. Agriculture Organization, "Gateway to poultry production and products," 2022. [Online]. Available: https://www.fao.org/

[2] A. Thornton, "Oecd-fao agricultural outlook," p. 238, 2022. [Online]. Available: https://www.weforum.org/agenda/2019/02/chart-of-the-day-this-is-how-many-animals-we-eat-each-year/

[3] Marel, "Marel poultry." [Online]. Available: https://marel.com/

[4] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 688–716, 2018.

[5] I. F. of Robotics, "World robotics," 2022. [Online]. Available: https://ifr.org/,accessed:October2022

[6] W. U. Farm Technology Group. (2018) Flexcraft program. [Online]. Available: https://flexcraftprogram.com/

[7] T. B. Jørgensen, K. Debrabant, and N. Krüger, "Robust optimization of robotic pick and place operations for deformable objects through simulation," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 3863–3870.

[8] J. Curhan, T. Womersley, and J. A. Lessing, "Robotic gripper for handling meat products," Feb. 7 2019, uS Patent App. 16/053,213.

[9] T. B. Jørgensen, B. R. Hansen, M. M. Pedersen, N. Krüger, and N. W. Hansen, "A flexible suction based grasp tool and associated grasp strategies for handling meat," in *Proceedings of the 2018 4th International Conference on Mechatronics and Robotics Engineering*, 2018, pp. 43–47.

[10] T. B. Jørgensen, S. H. N. Jensen, H. Aanæs, N. W. Hansen, and N. Krüger, "An adaptive robotic system for doing pick and place operations with deformable objects," *Journal of Intelligent & Robotic Systems*, vol. 94, no. 1, pp. 81–100, 2019.

[11] T. B. Jørgensen, N. Krüger, M. M. Pedersen, N. W. Hansen, and B. R. Hansen, "Designing a flexible grasp tool and associated grasping strategies for handling multiple meat products in an industrial setting," *International Journal of Mechanical Engineering and Robotics Research*, vol. 8, no. 2, pp. 220–227, 2019.

[12] B. Joffe, T. Walker, R. Gourdon, and K. Ahlin, "Pose estimation and bin picking for deformable products," *IFAC-PapersOnLine*, vol. 52, no. 30, pp. 361–366, 2019.

[13] Robohub, "Amazon picking challenge." [Online]. Available: https://robohub.org/tag/amazon-picking-challenge/

[14] C. Eppner, S. Höfer, R. Jonschkowski, R. Martín-Martín, A. Sieverling, V. Wall, and O. Brock, "Lessons from the amazon picking challenge: Four aspects of building robotic systems." in *Robotics: science and systems*, 2016.

[15] S. Ross, O. Korostynska, L. Cordova-Lopez, and A. Mason, "A review of unilateral grippers for meat industry automation," *Trends in Food Science & Technology*, 2021.

[16] E. J. Van Henten, J. Hemming, B. Van Tuijl, J. Kornet, J. Meuleman, J. Bontsema, and E. Van Os, "An autonomous robot for harvesting cucumbers in greenhouses," *Autonomous robots*, vol. 13, no. 3, pp. 241–258, 2002.

[17] C. W. Bac, E. J. van Henten, J. Hemming, and Y. Edan, "Harvesting robots for high-value crops: State-of-the-art review and challenges ahead," *Journal of Field Robotics*, vol. 31, no. 6, pp. 888–911, 2014.

[18] S. R. I. mGrip, "Soft robotics inc. mgrip." [Online]. Available: https://www.softroboticsinc.com/mgrip

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[22] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[23] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact: Real-time instance segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9157–9166.

[24] C. Zhou, *Yolact++ Better Real-Time Instance Segmentation*. University of California, Davis, 2020.

[25] R. Rekha, R. R. David, K. Frans, K. Gert, and V. H. Eldert, "posepp: 6d pose estimation of poultry pieces in a cluttered bin," *IEEE INTERNATIONAL CONFERENCE ON COMPUTATIONAL CYBERNETICS AND CYBER-MEDICAL SYSTEMS*, vol. 15, no. 1, pp. 172–188, 2022.

[26] M. Esposito, "easyhandeye: automated, hardware-independent hand-eye calibration." [Online]. Available: https://github.com/IFL-CAMP/easy_handeye

[27] C. Q. Littile and C. W. Wilson, "Rapid world modelling for robotics," Sandia National Labs., Tech. Rep., 1996.

[28] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, 2012.

[29] S. Chitta, I. Sucan, and S. Cousins, "Moveit![ros topics]," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 18–19, 2012.

[30] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2149–2154.

[31] M. Jordan and A. Perez, "Optimal bidirectional rapidly-exploring random trees," 2013.

[32] P. Bencak, D. Hercog, and T. Lerher, "Evaluating robot bin-picking performance based on box and blocks test," *IFAC-PapersOnLine*, vol. 55, no. 10, pp. 502–507, 2022.