Robotics: Science and Systems (RSS 2023), July 10 - 14, 2023, Daegu, Republic of Korea
The published version can be accessed from https://roboticsconference.org/program/papers/023

Demonstrating Large-Scale Package Manipulation via Learned Metrics of Pick Success

Shuai Li*, Azarakhsh Keipour*, Kevin Jamieson*†, Nicolas Hudson*, Charles Swan* and Kostas Bekris*‡
*Amazon Robotics, Seattle, Washington 98109, USA
†University of Washington, Seattle, Washington 98105, USA
‡Rutgers University, Piscataway, New Jersey 08854, USA
Email: {amzshua, keipourv, jamikevi, hudnco, cswan, bekris}@amazon.com

***Abstract*—Automating warehouse operations can reduce logistics overhead costs, ultimately driving down the final price for consumers, increasing the speed of delivery, and enhancing the resiliency to workforce fluctuations. The past few years have seen increased interest in automating such repeated tasks but mostly in controlled settings. Tasks such as picking objects from unstructured, cluttered piles have only recently become robust enough for large-scale deployment with minimal human intervention.**

This paper demonstrates a large-scale package manipulation from unstructured piles in Amazon Robotics' Robot Induction (Robin) fleet, which utilizes a pick success predictor trained on real production data. Specifically, the system was trained on over 394K picks. It is used for singulating up to 5 million packages per day and has manipulated over 200 million packages during this paper's evaluation period.

The developed learned pick quality measure ranks various pick alternatives in real-time and prioritizes the most promising ones for execution. The pick success predictor aims to estimate from prior experience the success probability of a desired pick by the deployed industrial robotic arms in cluttered scenes containing deformable and rigid objects with partially known properties. It is a shallow machine learning model, which allows us to evaluate which features are most important for the prediction. An online pick ranker leverages the learned success predictor to prioritize the most promising picks for the robotic arm, which are then assessed for collision avoidance. This learned ranking process is demonstrated to overcome the limitations and outperform the performance of manually engineered and heuristic alternatives.

To the best of the authors' knowledge, this paper presents the first large-scale deployment of learned pick quality estimation methods in a real production system.

I. Introduction

Automation in the industrial, manufacturing, and warehouse sectors has the potential to lower overhead expenses associated with producing, handling, and sorting goods. The increased speed and precision for handling each product can lower customer costs and improve product quality. Furthermore, it can reduce risks to humans in manual operations and enhance resilience to fluctuations in the labor market and overall economy.

Robot manipulation systems have already gained significant traction across industries, from car and garment manufacturing to crating apples [16, 13, 20, 19]. Many repeated operations in industrial settings include pick-and-place tasks using robot arms [17, 3]. Induction robots pick items from one location (e.g., a conveyor belt, a tote, or a box) and place them in another spot with the goal of singulating the items or feeding them to another machine (e.g., a sortation machine). Automation is still far from perfect, however. Some robot settings require a simplified environment to operate (e.g., only a single rigid object placed around the center of the conveyor belt), while others can only deal with a subset of the target objects, and the rest are passed on to humans (e.g., apple crating). Success metrics and cost benefits vary across tasks, and while these systems are largely beneficial, recent advances in robotics, computer vision, and machine learning are providing additional opportunities for robots to become financially viable in rather complex manipulation operations.

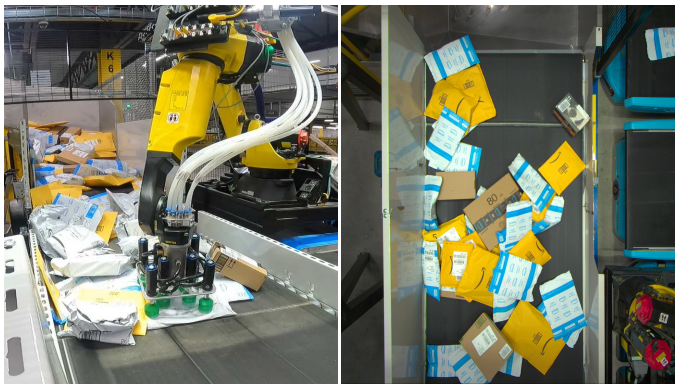


Fig. 1: A robot induction (Robin) workcell used for the statistics of this demonstration. The robotic arm is used for automated package singulation by Amazon.com, Inc. It picks packages from an unstructured pile on a conveyor belt and places them on mobile drive robots.

This work presents the learned pick quality system used in the Robot Induction (Robin) fleet of Amazon.com, Inc., which sorts several million packages per day [1]. Figure 1 shows a Robin workcell picking packages from a conveyor belt, which has been used for the statistics presented in this demonstration. Once a package is picked, Robin scans and places the package on a mobile drive unit to be routed to an appropriate drop point.

There is variation in the arm setup among workcells due to real-world constraints of industrial facilities, and the exact information about the incoming items is generally unknown. For this reason, a perception system has been developed, which aims to provide information regarding the packages on the

conveyor belt. Even with effective perception in the loop, however, several challenges remain for successful picking, including:

- The packages have different types of material (e.g., rigid or non-rigid, smooth or rough, etc.), requiring different picking strategies. For example, packages can be rigid boxes, deformable polybags, or semi-rigid containers in a mail sorting application.
- While the perception system can estimate an object's dimensions and material type, the mass and mass distributions of the incoming objects are more difficult to evaluate.
- A typical scene of Robin will contain many packages, often in a pile, with many objects only partially observed or some wholly buried in the pile.
- The fleet of robots is generally heterogeneous across workcells. There are variations in the workcell design, the operation environments (e.g., the surface where the packages are picked from or placed on), and the manipulator arm models, and different end-of-arm tools (EoAT) may be used due to changes in hardware over the deployment period.

A crucial metric in the robotic induction task is the success rate of picking attempts. Ideally, in a pile of objects, the robot should pick them one by one and place them in the target area without dropping any items. Two significant types of failure are possible: either the robot fails to find a suitable pick in the scene (e.g., due to potential collisions or if the objects are somehow out of reach), or the pick attempt is unsuccessful (e.g., the object is dropped after being picked). We call the former as *planning failure* and the latter as *holding failure*. An additional type of failure arises in item singulation when inadvertently more than one item is picked and placed at the same time. We call this type of failure *multi-pick failure*.

To deal with planning failures, in theory, it is possible to check for reachability and collisions for a large finite number of picks in the scene until a viable pick is found. In practice, however, computing collision-free arm trajectories and performing reachability checks are expensive, and the high throughput requirements of the industrial operation limits the number of picks that can be analyzed for each scene. Even if viable picks are found that pass the checks, there is a need to choose the picks that succeed in holding and transferring the package to the mobile robot.

In this context, this work demonstrates how machine learning models trained on historical pick outcomes from a production system can be leveraged to overcome these challenges and improve the performance of such large-scale deployments. Moreover, as the model's ability to predict outcomes improves with more data, the estimated pick qualities become more accurate over time. In particular, the contributions of this demonstration can be summarized as follows:

- We demonstrate a large-scale system for predicting pick qualities using machine learning. During our evaluation, this system picked up to 5 million packages daily (i.e., over 200 million packages over the corresponding period).
- We describe a ranking strategy for the picks, which uses the learned pick quality prediction system. This strategy has improved the production system's metrics compared to manually engineered, heuristic methods.
- We show that retraining the model on more recent data improves performance, indicating that this learning system is already effective with smaller amounts of data but can also improve over time with more recent and increasing datasets.

The rest of this demonstration paper is organized as follows: Section II reviews the related work and state of the art; Section III formalizes the problem considered in this work; Sections IV and V explain the methods for pick success prediction and learned pick ranking used by the demonstrated system; Section VI describes the evaluation performed and discusses the results of the corresponding tests in the production system; finally, Section VII discusses the lessons learned and future efforts in this area.

II. Related Work

Having an induction scene with several objects (shorthanded as an *induct*), the robot needs to execute one pick to move an object out of the pile and place it in the desired spot at each step. The robot may consider many candidate picks, but at each stage needs to choose and execute just the one with the highest predicted chance of success.

Ideally, the highest-ranking candidate pick should be the one chosen for execution, eliminating the need for suggesting more than one candidate pick. However, a complete feasibility evaluation of all potential candidates is usually impractical due to computational and time constraints. Therefore, an ordered list of candidate picks should be provided to the robot; the robot will evaluate the candidate picks one by one and execute the highest-ranked candidate that passes the feasibility checks (e.g., collision and robot arm reachability checks).

Various strategies can be devised to order the candidate picks. Some of these methods are based on hand-crafted intuitive heuristics, such as prioritizing the larger objects or the objects at the top of the pile and preferring the picks closer to the center of the objects and the ones having a higher number of activated suction cups during EoAT's contact with the object. In practice, such heuristics may work for a nominal induct but fail in complex scenarios and edge cases. Trying to manually handle all possible scenarios with more heuristics quickly becomes intractable. As an alternative, we can consider a data-driven approach that uses machine learning to *learn* a metric for the success probability of a particular pick and then rank the picks based on this score.

Pick selection based on a score learned from data has been an active research area in the past two decades. Morrison et al. [12] learn to estimate grasp qualities, angles, and gripper widths for each pixel in a depth image, assuming that the parallel jaw gripper's center is aligned with that pixel. The learning model is trained using a dataset of actual and simulated grasps. However, it does not work for non-vertical grasps and requires only a single object in its region of interest.

Morales et al. [11] use a set of visual and geometric features with K-Nearest Neighbor clustering to predict grasp success

for BarrettHand™. However, it is only suitable for 2-D planar shapes and does not generalize to the 3-D cluttered scene of industrial inducts.

Araki et al. [2] propose a learning method for simultaneous object detection, semantic segmentation, and grasping detection. However, as a black box, it is difficult to improve the grasp quality using this method. A more modular approach would allow simpler debugging for settings outside the lab's controlled environment.

Mahler et al. [9, 10] show the feasibility of directly predicting grasp qualities from point clouds with sufficient training data. It is designed for a gripper and a single suction cup at the EoAT. Similar to the work by Araki et al. [2], the biggest drawback of this method is that the output is hard to interpret, and it is difficult to intelligently improve the performance beyond increasing the training dataset size. However, many ideas from this work inspired our solution.

Zeng et al. [18] focus on robotic manipulators with multiple EoATs. They propose having various sets of grasps for different EoAT types based on the robot's perceived environment and selecting the EoAT based on the highest predicted success probability. It considers each affordance (e.g., which suction cups are active or inactive on the EoAT) as context rather than as a feature, producing many grasp sets to evaluate against the scene when there are many affordances. This drawback makes the approach infeasible for real-time applications with EoATs that have many affordances.

Some scenes may be void of any picks estimated to succeed with high probability. In this case, Liu et al. [7] propose a novel interactive exploration strategy that learns to push the objects around to obtain a better set of possible grasps in a complicated environment. While this method can potentially help in scenarios with no feasible picks or grasps, it is time-consuming and cannot be directly deployed in fast-paced industrial tasks.

Most of the research on grasp and pick quality and success prediction has been done in controlled lab settings, allowing to hold assumptions such as having singulated, rigid, or 2-D shaped objects or only performing vertical picks. In order to have a working system in real-world uncontrolled settings, such as a package fulfillment center or a mail package sortation facility, the method should be able to overcome these limiting assumptions.

In our proposed method, we have identified a selection of relevant features and have developed models that can assess the pick's quality in a cluttered uncontrolled scene without the limitations of other methods and within the industrial computational and timing constraints. The methods are deployed across a fleet of Robin manipulator robots in fulfillment centers and have been responsible for picking over 200 million packages during our evaluation period. To the best of our knowledge, our work is the only method for predicting the pick quality and ranking of picks that can work with different EoAT orientations, uncertain object material and properties, and cluttered environments.

III. Problem Statement

Consider the picking task illustrated in Figure 1. The task is initiated when a scene of cluttered packages of different types arrives at a reachable area via a conveyor belt. The conveyor belt and the scene remain static throughout the picking process until the scene is "cleared," which occurs when no reachable packages remain or when some exception occurs, and the next scene arrives.

The action of picking is performed by an induction manipulation robot consisting of a multiple-DoF arm with an end-of-arm tool (EoAT). The EoAT may consist of one or more suction cups. Depending on the EoAT design, each suction cup may be controlled individually, only as groups, or only all together.

Each *pick* is defined as a set of variables determining the actions of the robot: a 3-D point in space (i.e., the desired pick point where the EoAT makes contact with an item's surface), the desired 3-D orientation of the EoAT at the pick point, and a set of desired active suction cups on the EoAT. Note that a single package or even a package segment may be associated with many candidate picks.

At each time t, we will use x_t to represent the state of the current scene and $\mathcal{P}_t$ to denote the complete set of possible picks over the scene, determined by an elementary filtering process (e.g., making sure the pick point is on an item).

Given a scene x_t and any pick $p \in \mathcal{P}_t$, we can construct a d-dimensional feature vector $\phi(x_t, p) \in \mathbb{R}^d$ that encodes not only the parameters defining the pick p but also how the pick relates to the scene. For example, it may include the distance from the bottom of each suction cup to the surfaces of the packages beneath, estimated from point-cloud data. See Section IV-B for the extracted features used in our current deployment.

The role of the pick ranker is to use $\phi(x_t, p)$ for each $p \in \mathcal{P}_t$ to define an ordering over the candidate picks $\mathcal{P}_t$. This ordered list is passed through an final filtering step (e.g., checking the feasibility of planning the arm motion without a collision), and the robot executes the first feasible pick. The scene is cleared if no viable picks are found (i.e., *planning failure*). Once a feasible pick is executed, whether successful or not, the process starts all over again on the next scene, which may be a slightly modified or entirely new scene.

Ideally, the pick ranker would have perfect knowledge of the final filtering step and would dictate just a single pick to be executed. In practice, the final filtering process can vary by location and other constraints of the particular deployed robot, which may not be known beforehand. To minimize complexity, we only consider memoryless pick ranking systems: only the current scene is considered when choosing a pick, and there is no effort to plan ahead a sequence of picks.

We assume there exists a function $F : \mathbb{R}^d \to [0, 1]$ such that for a scene x_t and any pick $p \in \mathcal{P}_t$ the probability that a pick p will be successful is equal to $F(\phi(x_t, p)) \in [0, 1]$. Note that this model assumes $\phi(x_t, p)$ contains all necessary information about whether a pick will be successful. This simplifying assumption does not reflect that there may be unobserved factors

influencing pick success, such as the weight distribution inside the package. Extending our model to handle such partially observed settings is ongoing work.

To maximize the probability of a successful pick at time t, an ideal pick ranker would rank the picks of $\mathcal{P}_t$ in decreasing order of $F(\phi(x_t, p))$. Note that under this model, the success probability $F(\phi(x_t, p))$ is agnostic to picks p that do not pass the final filtering process and thus can take an arbitrary value.

In practice, we evaluate a surrogate for $F(\phi(x_t, p))$ (which takes a non-negligible amount of computation), so ideally, $\mathcal{P}_t$ would only include those picks that pass the final filtering process.

Of course, the true F is unknown, but we can estimate it with data and an appropriate machine learning model (see Section IV).

IV. Learning to Estimate Pick Success

This section describes our data-driven approach to estimating the probability of success for a given pick in a scene. First, we provide the details for the training datasets used in our work, and then we briefly outline the features extracted for our model. Finally, we explain the details of the models developed for this project.

A. Training Dataset

We compiled three datasets from hundreds of actual induction cells in Amazon fulfillment centers. Due to the nominal success of pick ranking heuristic methods used in the past, an independent and identically distributed (IID) random draw of inducts would lead to a severely imbalanced dataset, with pick success examples vastly outnumbering the failure examples. Therefore, we oversampled failures in all the training datasets to create a more balanced dataset.

- **TrainDataset-Center:** This dataset contains ~395K robotic inducts composed of 335,226 successful and 59,646 failed examples. Additionally, due to the used heuristics, the location of the picks in the dataset is as close to the center of package segments as possible.
- **TrainDataset-Random:** For this dataset, ~41K randomly selected inducts from *TrainDataset-Center* are replaced with new inducts that were randomly distributed to be picked anywhere on the packages' segments with a higher chance of being close to center when a center pick is possible. The new set of inducts comprises 34,715 successful and 6,673 failed picks, and the total size of the newly-created dataset is the same as *TrainDataset-Center*.
- **TrainDataset-Past:** This dataset is compiled from historical data collected from inducts executed before the timeframe of *TrainDataset-Center* and *TrainDataset-Random* inducts. It contains ~230K inducts composed of 195,408 successful and 34,482 failed examples. Similar to *TrainDataset-Center*, the location of the picks in the dataset is as close to the center of package segments as possible.

Each induction consists of the RGB image data captured by a camera at the top of the workcell looking straight down, depth images, and metadata. The metadata includes information on the induct, such as the ground truth on the success or failure, as well as information about the workcell (e.g., the station code, the type of manipulator arm and EoAT).

B. Feature Extraction

We compute a set of features for each induct using the metadata, RGB, and depth images. Specifically, the camera data is processed by our perception system to generate segments of the packages and tag each segment with an associated package type label. Additional statistics are computed for each segment using depth information (e.g., surface normals and the quality of plane fitting). An overview of our perception system design to extract the required features is shown in Figure 2.

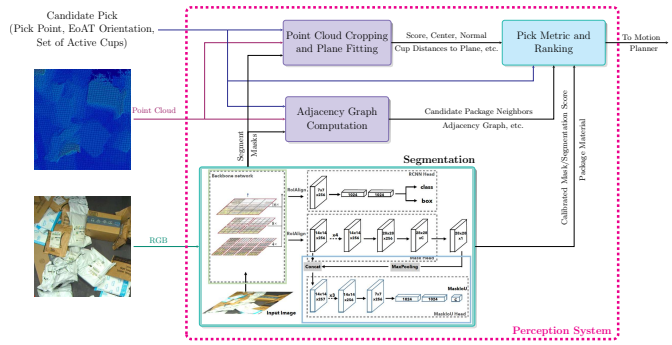


Fig. 2: An overview of the perception system design to extract features for the learned pick success model.

The segmentation module is a deep network based on Mask Scoring R-CNN [6] with a Swin-T backbone [8] for predicting the package material, instance segmentation masks, and the classification and segmentation scores. The rest of the features extracted for the model are directly computed from the input pick and point-cloud data.

Based on our feature importance studies, we identified the following features as significant predictors:

- **Package height:** We believe this feature correlates with the package's momentum and, therefore, can impact the shear force at the suction cups and the pick's stability.
- **Quality of plane fitting:** We fit a plane on each segment, and we speculate that a better plane fit correlates with a better seal between the suction cups and the package.
- **Number of activated suction cups:** More active suction cups can mean a more stable pick, reducing the failure probability.
- **Alignment quality between the suction cups and the package surface:** This feature is computed as the offsets between the package surface normal vector and the normal vector of the suction cups. We also expect this feature to be significant since a better alignment indicates a better seal between the suction cups and the package surface.

In addition to the above and other segment-specific features, we compute features that describe each segment's relationship with its surroundings, including the number of nearby segments and the *adjacency graph* features. To compute the

adjacency graph features, we construct a graph that captures the topological order of the package segments. This graph captures each detected segment's relative height with respect to its adjacent neighbor segments. Figure 3 shows an example where the numbers represent the relative position ranking of the segment among its neighbors.

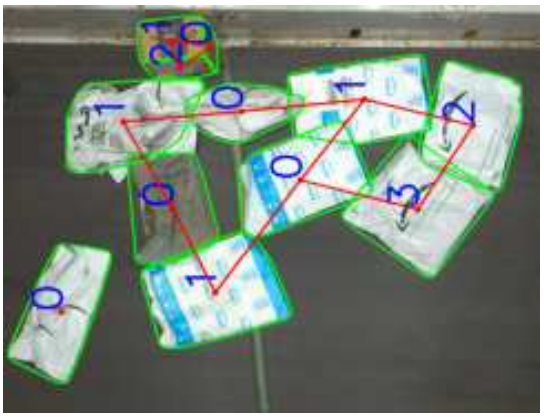


Fig. 3: Example of an adjacency graph for a cluster of items.

C. Pick Success Model

The features described in Section IV-B, along with the ground truth knowledge of induct success or failure, were extracted for the inducts in the training datasets (Section IV-A). The AutoGluon library [4] was leveraged for training, model selection, and hyperparameter tuning for the pick success prediction binary classification task. Many models showed similar performance, but a gradient boosting tree, specifically a CatBoost model [15], was among the top performers and was chosen for our implementation. We also evaluated other machine learning libraries and modeling options, such as multilayer perceptron (MLP) with Scikit-learn [14]; however, models trained with AutoGluon showed superior performances.

In contrast to our strategy of extracting interpretable (tabular) features, prior works have trained models that directly predict the pick success from some combination of the input RGB image, depth, and pick features. We also benchmarked that approach but did not see a significant improvement over our model's performance. Moreover, our method has a few advantages over the pixels-to-prediction approach:

1) **Interpretability:** We found it challenging to understand what made a particular scene easy or difficult with the pixels-to-prediction approach and why a specific pick failed. In contrast, the tabular features we extracted were easy to interpret and allowed us to characterize different failure modes. This helps to identify weaknesses in the training dataset to refine it later.
2) **Computation:** A gradient boosting tree is much faster to train and evaluate than a typical image classification model. This means that at deployment, we can evaluate many picks very quickly.
3) **Uncertainty quantification:** The CatBoost package natively supports sampling models from a posterior, which can be used to generate ensembles. In anecdotal studies, we found that these ensembles capture uncertainty very well. Because our training data does not cover all possible picks for all scenes, capturing such uncertainty helps provide more conservative predictions.

V. Pick Ranking

Let us assume that the robot has selected several picks in the scene. For example, the system may generate one or more picks per each object segment in the robot's region of interest. The model described in Section IV can output estimated probabilities of successfully picking up and holding the desired items for each candidate pick. Assuming the estimate closely correlates with the actual probability of success, the picks with a higher likelihood of success should be prioritized.

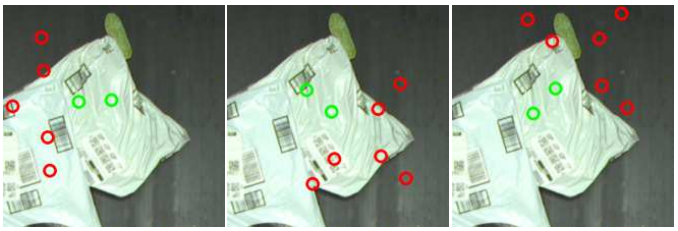
Ranking the picks based on their success probability estimate provides two benefits: the throughput of the workcell is improved due to the higher chance of picking each item up on the first try, and, by picking up the "easier-to-pick" items first, the "harder-to-pick" items become easier to pick (e.g., the occluding or very close items are removed around them, leaving them singulated).

In our work, we rank the picks in two steps. First, the picks are grouped by package segments, and the pick success probabilities or other heuristics are leveraged to rank the segments. When picking success probability is employed directly, we estimate the success probability of a segment as $P_{segment} = \max_{i=1...n} P^i_{pick}$, where P^i_{pick} is the success probability of the segment's i^{th} pick. Finally, once the segments are ranked, for each segment, we order its picks based on their success probability predictions. The two-step ranking is due to the logistics behind our system structure and the desired flexibility to try different methods for the two steps.

Figure 4 shows two examples of package rankings using our model, where the flat and large package segments are prioritized over the crumpled and small ones.

Fig. 4: Examples of ranking the packages based on the highest success probability estimate of their corresponding picks. A smaller rank number represents a higher priority.

Figure 5 shows the predicted success probability of three picks on a deformable package for picking with the suction cup arrangement illustrated in Figure 6(b). All three picks have two activated suction cups on the package. However, the model appears to prioritize the pick configurations where the EoAT is less likely to collide with the surrounding packages and is more likely to succeed.



(a) $P_{pick} = 0.708$ (b) $P_{pick} = 0.756$ (c) $P_{pick} = 0.825$

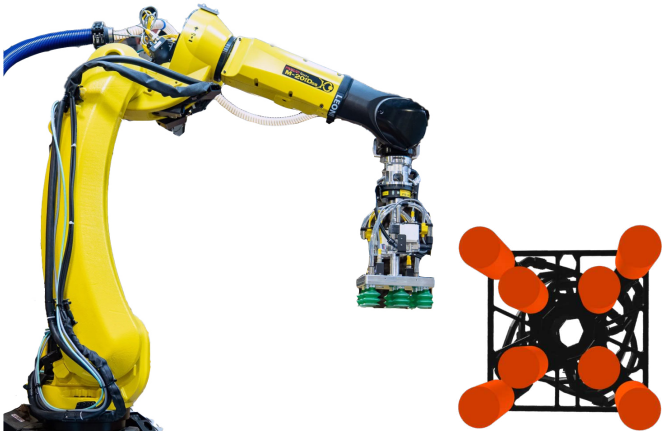
Fig. 5: Estimated success of different picks on a deformable package. The circles correspond to the suction cups. The green and red circles are the active and inactive suction cups.

VI. Experiments and Results

The methods proposed in Sections IV and V have been deployed and tested in multiple Amazon sites worldwide. This section presents our testing conditions, experiment results, and our analysis.

A. Hardware

The proposed method is implemented for Robin robot [1] used in Amazon fulfillment centers. The main arm consists of FANUC M-20iD/35 with six controlled axes, 35 kg payload, and 1831 mm reach (Figure 6(a)). The EoAT consists of 8 suction cups arranged in an "X" configuration with a size of 25×25 cm. Each suction cup can be controlled individually. Figure 6(b) illustrates this EoAT configuration.



(a) (b)

Fig. 6: Robin robotic arm used in our experiments. (a) Manipulator arm. (b) A simulation of the end-of-arm tool design with eight suction cups.

B. Baselines and Experiments

To evaluate the performance of different *pick success* modeling options, we first consider the following two baselines:

- **AlwaysSuccess:** Always predicting pick success;
- **BoostedTree-Past:** Our pick success model described in Section IV-C trained with the historical *TrainDataset-Past* data (see Section IV-A);

Historically, our robots were programmed to pick up packages at poses close to the package centers. However, there are cases where the robots must choose picks further away from the package center to avoid collisions, such as with other packages or fixtures on the conveyor belt. To be able to see the effect of choosing off-center picks, we trained two pick success models:

- **BoostedTree-Center:** Our pick success model described in Section IV-C trained with the *TrainDataset-Center* data (see Section IV-A);
- **BoostedTree-Random:** Our pick success model described in Section IV-C trained with the *TrainDataset-Random* data (see Section IV-A).

CNN-Center: Finally, we also report the performance of an image-based model, which is a network trained on *TrainDataset-Center* using RGB image crops around the target packages (Figure 7).

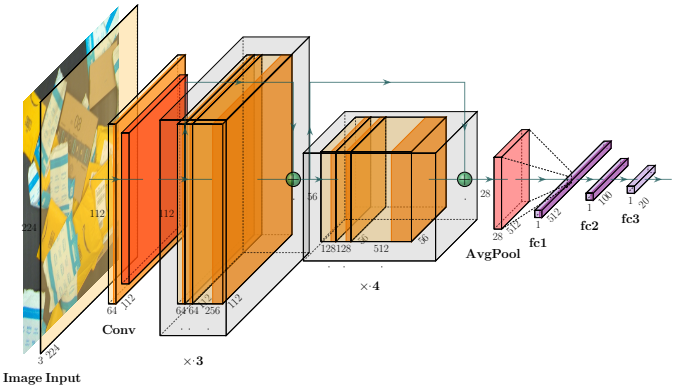


Fig. 7: Network architecture for extracting the RGB image embeddings. The input is a 400×400 patch around the target pick point (resized to 224×224), and the output is the pick success prediction. The blocks before the AvgPool layer are the first three convolution layers of a pre-trained Resnet50 model. The output of the second fully-connected layer (fc2) is used as the embedding. The size of the fc2 layer output (fc3 layer input) is set to 20, 48, or 96 for the desired image embedding sizes.

The following testing datasets are designed to evaluate the pick success estimation models:

- **EvalDataset-Center:** Consisting of ~60K picks that are close to the package center. This dataset has an overall pick success rate of 94.40%.
- **EvalDataset-Random:** Consisting of ~38K picks randomly chosen to be anywhere within the package segment. This dataset has an overall pick success rate of 94.01%.

For *ranking* segments on the actual robots, we considered several heuristic approaches, including:

- *Z-order:* Ranking the picks by the package's target surface elevation. This heuristic is motivated by the assumption that a package whose surface is at the top of the pile is easily reachable, and the robot can avoid collisions or mistakenly pick other occluding packages when trying to pick it up. This heuristic is simple to implement but omits information about actual occlusions and, in practice, fails for unreachable picks and in instances where a portion of the package is at the top of the pile, but the rest is buried under other packages. In addition, since this method is only concerned with the elevation of the package's surface, it is useless for packages lying on the conveyor belt.
- *Package size:* Ranking the picks by the package size (or segmentation area). This heuristic assumes that picking packages with a larger visible area will have higher success and that moving out these larger packages first will help declutter the scene, making picking smaller packages easier. A major issue with this heuristic is when smaller packages lie on or overlap with larger ones. This can result in collisions with other packages, difficulty lifting a larger package from under smaller packages, or picking more than one package simultaneously, resulting in holding or multi-pick failures.
- *Topological order:* Ranking the picks based on the order of occlusion of the packages. Picks to package faces that appear unoccluded get the best score. Picks on package surfaces that are only occluded by unoccluded packages get the next highest score, and so on. This heuristic is moderately simple to implement and prioritizes the unoccluded packages to improve the success probability. However, it fails to recognize the unreachable package surfaces and the occluded packages with unoccluded surfaces, and similar to the Z-order method, it does not differentiate between the picks on the same package surface. Moreover, this method heavily relies on an accurate perception system to compute the segment overlaps.

Any combination of the above heuristics may also be employed. For example, a topological order method can be used as the primary ranking criteria, with the Z-order method as the tie-breaker when two packages have the same topological ranks. Some other heuristic approaches are also worth mentioning, such as the measure of how quadrilateral a package is or the confidence score given by the instance segmentation method. These approaches try to indirectly measure a package's occlusion or deformation level but have their own drawbacks and challenges.

The heuristics above only rank the segments to be picked and cannot differentiate between various picks inside a segment. Once the segments are ranked, a reasonable heuristic approach for ranking picks inside a segment can be giving a higher score to the picks closer to the center of the segment and the ones with a higher number of activated EoAT suction cups.

Having experimented with these heuristics, we establish the following baseline and experiment to evaluate our proposed pick ranking approach:

- **Baseline:** Topological order with Z-order as the tie-breaker for ranking segments; picks with EoAT poses with a higher number of active suction cups, and then the ones with the center of active cups closer to the center of the segment are given higher rank within segments;
- **Experiment:** Topological order with learned pick success estimation as the tie-breaker for ranking segments; picks within segments are also ranked with the learned pick success estimation.

To evaluate our approach without bias to a particular planning strategy, we allow the robots to select EoAT poses randomly anywhere as long as the poses are within 30 cm distance from the center of the packages on the estimated package surface plane.

Moreover, additional large-scale experiments were performed through A/B tests on the Robin fleet to assess the performance of different methods. Based on the possible heuristic approaches, we established the following baselines to evaluate against our proposed pick ranking approach:

1) **TopoZ-Center:** Topological order with Z-order as the tie-breaker for ranking segments; picks within segments chosen close to the center of the package's segment with higher rank given to the EoAT poses with a higher number of active suction cups;
2) **Z-Center:** Same as *TopoZ-Center*, but only Z-order used directly for ranking segments;
3) **TopoZ-Random:** Same as *TopoZ-Center*, but picks within segments are chosen randomly anywhere in the package's segment, with a higher chance for picks closer to the segment's center.

For evaluation of the ranking methods through learned pick success estimation, we designed the following experiments:

1) **TopoLPR-Center:** Topological order with learned pick success estimation as the tie-breaker for ranking segments; picks within segments chosen close to the center of package's segment with learned pick success estimation used for ranking;
2) **LPR-Center:** Same as *TopoLPR-Center*, but learned pick success estimation is directly used for ranking the segments;
3) **LPR-Random:** Same as *LPR-Center*, but picks within segments can be chosen randomly anywhere in the package's segment.

C. Evaluation Metrics

Due to the rarity of failed inducts, our evaluation datasets are severely imbalanced. As a result, evaluating the classification *accuracy* does not serve as an informative metric for this task (for example, a baseline that always predicts pick success will get over 94% accuracy).

On the other hand, the main objective of a pick success classifier is to use its output estimates for ranking the picks so that picks with a higher chance of success are ranked higher.

We choose Receiver Operating Characteristic (ROC) Area Under the Curve (AUC) score as the metric for evaluating the pick success models. Mathematically, the ROC-AUC score is the same as the probability of a classifier ranking a randomly-chosen positive example higher than a randomly chosen negative example, i.e., $P(score(x^+) > score(x^-))$ (see [5] for the proof). Therefore, when all the successful picks in the testing datasets are ranked higher than all the failed picks, the ROC-AUC score would be 1.0, and when all the failed picks are ranked higher than successful picks, the ROC-AUC score would be 0. Therefore, the ROC-AUC score is a good metric for evaluating the pick-ranking ability of different models.

To evaluate the pick ranking system on the actual hardware, we also compute the percentage of picks when the robots fail to transfer a package from the conveyor belts to the mobile robots. We call this metric as *failure rate*.

D. Results

All models introduced in Section VI-B were evaluated on both testing datasets *EvalDataset-Center* and *EvalDataset-Random*. Table I presents the ROC-AUC scores of these models with confidence intervals.

TABLE I: ROC-AUC scores with confidence intervals of different models for the pick success estimation task.

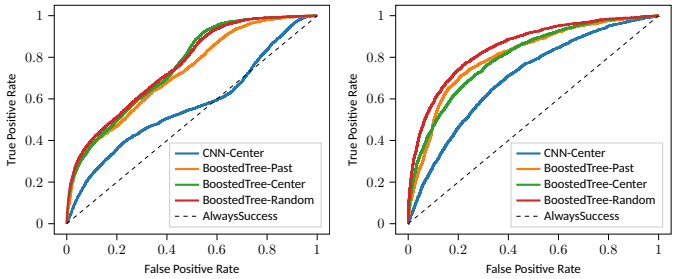
Model	EvalDataset-Center	EvalDataset-Random
AlwaysSuccess	0.5 (0.5, 0.5)	0.5 (0.5, 0.5)
BoostedTree-Past	0.725 (0.717, 0.732)	0.807 (0.799, 0.815)
BoostedTree-Center	0.755 (0.748, 0.761)	0.802 (0.792, 0.810)
BoostedTree-Random	0.758 (0.752, 0.765)	0.848 (0.840, 0.855)
CNN-Center	0.570 (0.560, 0.579)	0.703 (0.693, 0.712)

As seen from Table I, all the machine learning models beat the naive baseline *AlwaysSuccess* that always predicts pick success. Additionally, all models perform better on *EvalDataset-Random* compared to *EvalDataset-Center*. From Figure 8, we observe that the ROC curves on the *EvalDataset-Center* dataset (Figure 8(a)) are flatter than the ROC curves on the *EvalDataset-Random* dataset (Figure 8(b)) for middle range false positive rates. This indicates that there are more pick failure examples that are hard to differentiate from the pick success examples in the *EvalDataset-Center* dataset.

Upon further investigation of the datasets, we found that a sizeable portion of pick failure examples is due to factors not included in our feature set, such as suction cup degradations. We believe the proportion of such examples is more significant when the robot action space is more constrained, such as always attempting to pick packages close to the center.

When comparing models trained with data from different time ranges (i.e., *BoostedTree-Past* vs. *BoostedTree-Center* and *BoostedTree-Random*), we observe the model performance improves when it is trained with more recent data.

Additionally, we find that the model trained with picks sampled anywhere within the segment (i.e., *BoostedTree-Random*) slightly outperforms the model trained with picks always close to the center when evaluated on *EvalDataset-Center*. On the other hand, there is a more significant margin between *BoostedTree-Random* and *BoostedTree-Center* when they are evaluated on *EvalDataset-Random*. This shows that our pick success model can interpolate between centered picks and picks that are more off-center, and including the off-center picks in the training dataset helps with predicting pick success for larger varieties of picks, which can be beneficial for packages that are hard to reach and the robot has to select from a list of off-center picks. Finally, the RGB image-based model *CNN-Center* performs much worse than the other modeling options. Given *CNN-Center* makes predictions only based on the package appearance, it shows that additional information about the picks is critical even if the robots attempt to pick up the packages close to the center.



(a) (b)

Fig. 8: ROC curves of all models described in I tested on evaluation datasets (a) EvalDataset-Center and (b) EvalDataset-Random.

The *Experiment* ranking method (see Section VI-B) was deployed for production on Amazon's fulfillment center robotic fleet and has been used for picking up over 200 million inducts with a success rate of 98%. We analyzed ~180K random robotic inducts performed using *Baseline* and *Experiment* ranking methods to validate our proposed method. Table II summarizes the results, which shows that *Experiment* method improves the pick success rate by about 1.18%. This 23.7% reduction in failures, when deployed at a large scale (e.g., millions of picks per day performed on our fleet), has a significant impact on the operation costs.

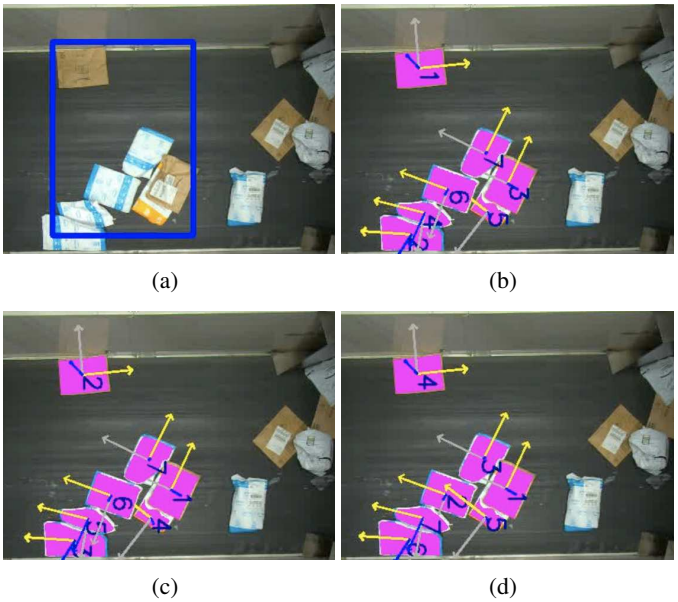
TABLE II: Test results for validation of pick-ranking learned method.

Method	Total Picks	Pick Success	Pick Failure	Success Rate
Baseline	89,162	84,718	4,444	95.02%
Experiment	90,127	87,700	3,427	96.20%

To better understand the cases where these methods rank the segments differently, we present a qualitative comparison for two cases that assist with understanding the behavior of the learned pick success method. In general, given similar conditions for two packages, the learned pick success estimation method seems to prefer flatter surfaces and packages with less occlusion while disliking the packages close to the conveyor

wall or at hard-to-reach angles.

1) Case 1: Figure 9 shows an induct with its segment ranking results for three methods: topological order with Z-order for tie-breaking, topological order with learned pick success estimation method for tie-breaking, and the learned pick success estimation method. It can be seen that the learned approach prioritizes the packages closer to the center of the conveyor belt (away from the conveyor walls), where the robot is less likely to have difficulties with reaching the pick at the desired angle or colliding with the conveyor wall.



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

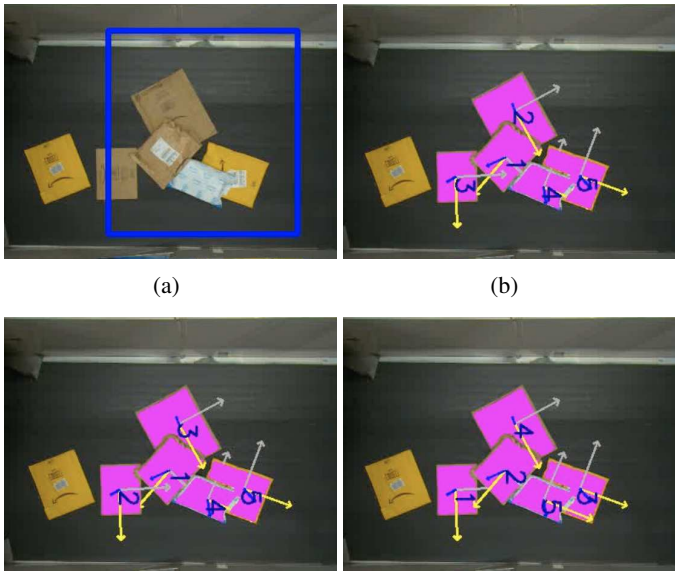
Fig. 9: Qualitative comparison of different segment ranking methods. (a) An example induct scene with a marked ROI (the blue rectangle). Results for: (b) topological order of segments with Z-order used for tie-breaking, (c) topological order of segments with learned pick success estimation used for tie-breaking, and (d) learned pick success estimation method directly used for segment ranking.

2) Case 2: Figure 10 shows an induct with its segment ranking results for the same methods as Case 1. Comparing the ranking for segments ranked 2 and 3 in Figures 10(b) and 10(c), it is evident that the learned method prioritizes the packages with less occlusion where the collision with the occluding packages is less likely. On the other hand, when the constraints of topological order are removed, it can be seen in Figure 10(d) that the learned method slightly prefers the packages with flatter surfaces (the box over the deformable packages) where the chances of holding failure are lower.

Finally, to find the best method, the large-scale A/B test was deployed across the fleet with a small percentage of total inducts allocated to each experiment group in Section VI-B. Table III summarizes the results of the A/B experiments.

The results show that using the learned pick success estimation to rank the segments (i.e., TopoLPR-Center and LPR-Center) improves the pick success for the robots compared with the manual heuristic ranking methods (i.e., TopoZ-Center and Z-Center). Interestingly, the best improvement comes from the more aggressive approach where we directly apply the learned pick success estimation for ranking (LPR-Center). Given that the heuristic ranking methods heavily depend on the heights of the packages, this suggests that promoting high packages can lead to more challenging picks, such as tall but unstable packages. On the other hand, the pick success model considers the package height only as an input feature along with other information such as the package position, surface normal, and adjacency graph features. Therefore, the pick success model can reason better and deprioritize unstable packages. It is also worth noting that if the picks are chosen randomly, the pick success improvement from the heuristic ranking method (TopoZ-Random) to the learned pick success estimation ranking method (LPR-Random) is even more significant (i.e., from 90.57% to 93.10%).

The pick success estimation model deployed in these A/B



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

Fig. 10: Qualitative comparison of different segment ranking methods. (a) Another example induction scene with a marked ROI (the blue rectangle). Results for: (b) topological order of segments with Z-order used for tie-breaking, (c) topological order of segments with learned pick success estimation used for tie-breaking, and (d) learned pick success estimation method directly used for segment ranking.

TABLE III: Results of A/B experiments for pick-ranking methods.

Method	Total Picks	Failed Picks	Success Rate
TopoZ-Center	1,158,353	89,378	92.28%
Z-Center	1,157,739	89,866	92.24%
TopoZ-Random	1,158,479	109,193	90.57%
TopoLPR-Center	1,156,697	83,535	92.78%
LPR-Center	1,160,005	72,789	93.73%
LPR-Random	1,157,342	79,820	93.10%

experiments predicts the pick success probability by taking the average of the predictions from five CatBoost models. The number of trees in the five CatBoost models is 2236, 1069, 799, 1464, and 1208 respectively. For all five models, the depth of the trees is 6, and we used a learning rate of 0.05.

VII. Conclusion

In this paper, we presented a large-scale deployed system for package manipulation, which estimates the pick success using a machine learning model. We demonstrated the effectiveness of this system by evaluating it on over 200 million picks and comparing it to heuristic baselines.

We believe that the recent developments in vision transformers combined with a large amount of induction data from our robotic fleet can improve our image-based network and may provide more valuable image embeddings, enhancing the prediction quality of the overall method.

Additionally, going through the mistakes made by our model, we realized that a sizable portion of them are due to hardware issues such as a dysfunctioning suction cup. In the future, we intend to leverage the developed pick success model for monitoring the health and analyzing the errors in our robot fleet.

Acknowledgments

The work would not have been possible without the support of the wider Amazon Robotics team. More specifically, the authors would like to thank previous and current members of the Robin and Janus teams, who established the underlying technology, provided support, and helped shape and deploy these ideas. The authors would like to give special thanks to David Oreper and Sicong Zhao for enabling deployment and to Shuai Han, Qiujie Cui, Mansour Ahmed, and Andrew Marchese, whose help was critical in the realization of this work.

Demonstration at RSS 2023

During the presentation of this paper, we intend to have a live demonstration of our evaluation or production workcells. The demonstration will present live webcams from different sites across the globe showing Robin robots picking and placing incoming packages.

References

[1] Amazon Science. Amazon Robotics: Robin, 2022. URL https://www.amazon.science/latest-news/robin-deals-with-a-world-where-things-are-changing-all-around-it.

[2] Ryosuke Araki, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. MT-DSSD: Multi-task deconvolutional single shot detector for object detection, segmentation, and grasping detection. *Advanced Robotics*, 36(8):373–387, 2022. doi: 10.1080/01691864.2022.2043183. URL https://www.tandfonline.com/doi/full/10.1080/01691864.2022.2043183.

[3] Andreas Björnsson, Marie Jonsson, and Kerstin Johansen. Automated material handling in composite manufacturing using pick-and-place systems – a review. *Robotics and Computer-Integrated Manufacturing*, 51: 222–229, 2018. ISSN 0736-5845. doi: 10.1016/j.rcim.2017.12.003. URL https://www.sciencedirect.com/science/article/pii/S0736584517301758.

[4] Rasool Fakoor, Jonas W Mueller, Nick Erickson, Pratik Chaudhari, and Alexander J Smola. Fast, Accurate, and Simple Models for Tabular Data via Augmented Distillation. In *Advances in Neural Information Processing Systems*, volume 33, pages 8671–8681, 2020. URL https://proceedings.neurips.cc/paper/2020/file/62d75fb2e3075506e8837d8f55021ab1-Paper.pdf.

[5] J A Hanley and B J McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, 1982. doi: 10.1148/radiology.143.1.7063747. URL https://doi.org/10.1148/radiology.143.1.7063747.

[6] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask Scoring R-CNN. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6402–6411, 2019. doi: 10.1109/CVPR.2019.00657. URL https://ieeexplore.ieee.org/document/8953609.

[7] Huaping Liu, Yuhong Deng, Di Guo, Bin Fang, Fuchun Sun, and Wuqiang Yang. An Interactive Perception Method for Warehouse Automation in Smart Cities. *IEEE Transactions on Industrial Informatics*, 17(2):830–838, 2021. doi: 10.1109/TII.2020.2969680. URL https://ieeexplore.ieee.org/document/8970574.

[8] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021. doi: 10.1109/ICCV48922.2021.00986. URL https://ieeexplore.ieee.org/document/9710580.

[9] Jeffrey Mahler, Florian T Pokorny, Brian Hou, Melrose Roderick, Michael Laskey, Mathieu Aubry, Kai Kohlhoff, Torsten Kröger, James Kuffner, and Ken Goldberg. Dex-Net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1957–1964, 2016. URL https://ieeexplore.ieee.org/document/7487342.

[10] Jeffrey Mahler, Matthew Matl, Vishal Satish, Michael Danielczuk, Bill DeRose, Stephen McKinley, and Ken Goldberg. Learning ambidextrous robot grasping policies. *Science Robotics*, 4(26):eaau4984, 2019. doi: 10.1126/scirobotics.aau4984. URL https://www.science.org/doi/abs/10.1126/scirobotics.aau4984.

[11] A. Morales, E. Chinellato, A.H. Fagg, and A.P. del Pobil. Experimental prediction of the performance of grasp tasks from visual features. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol-

ume 4, pages 3423–3428 vol.3, 2003. doi: 10.1109/IROS.2003.1249685. URL https://ieeexplore.ieee.org/document/1249685.

[12] Douglas Morrison, Peter Corke, and Jürgen Leitner. Learning robust, real-time, reactive robotic grasping. *The International Journal of Robotics Research*, 39(2-3):183–201, 2020. doi: 10.1177/0278364919859066. URL https://journals.sagepub.com/doi/10.1177/0278364919859066.

[13] Huong Giang Nguyen, Marlene Kuhn, and Jörg Franke. Manufacturing automation for automotive wiring harnesses. In *8th CIRP Conference of Assembly Technology and Systems*, volume 97, pages 379–384, 2021. doi: https://doi.org/10.1016/j.procir.2020.05.254. URL https://www.sciencedirect.com/science/article/pii/S2212827120314761.

[14] Pedregosa, Fabian, Varoquaux, Gaël, Gramfort, Alexandre, Michel, Vincent, Thirion, Bertrand, Grisel, Olivier, Blondel, Mathieu, Prettenhofer, Peter, Weiss, Ron, Dubourg, Vincent, Vanderplas, Jake, Passos, Alexandre, Cournapeau, David, Brucher, Matthieu, Perrot, Matthieu, Duchesnay, and Édouard. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. doi: 10.5555/1953048.2078195. URL https://www.jmlr.org/papers/v12/pedregosa11a.html.

[15] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. CatBoost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, volume 31, pages 1–11. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf.

[16] Mir Salahuddin and Young-A Lee. *Automation with Robotics in Garment Manufacturing*, pages 75–94. Springer International Publishing, Cham, 2022. ISBN 978-3-030-91135-5. doi: 10.1007/978-3-030-91135-5_5. URL https://doi.org/10.1007/978-3-030-91135-5_5.

[17] Nazib Sobhan and Abu Salman Shaikat. Implementation of Pick & Place Robotic Arm for Warehouse Products Management. In *2021 IEEE 7th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, pages 156–161, 2021. doi: 10.1109/ICSIMA50015.2021.9526304. URL https://ieeexplore.ieee.org/abstract/document/9526304.

[18] Andy Zeng, Shuran Song, Kuan-Ting Yu, Elliott Donlon, Francois R. Hogan, Maria Bauza, Daolin Ma, Orion Taylor, Melody Liu, Eudald Romo, Nima Fazeli, Ferran Alet, Nikhil Chavan Dafle, Rachel Holladay, Isabella Morona, Prem Qu Nair, Druck Green, Ian Taylor, Weber Liu, Thomas Funkhouser, and Alberto Rodriguez. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. *The International Journal of Robotics Research*, 41(7):690–705, 2022. doi: 10.1177/0278364919868017. URL https://journals.sagepub.com/doi/10.1177/0278364919868017.

[19] Kun Zhang and Hua Zhang. Design and implementation of automatic apple crating robot technology. In *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, pages 617–621, 2021. doi: 10.1109/ICBAIE52039.2021.9389974. URL https://ieeexplore.ieee.org/abstract/document/9389974.

[20] Zhao Zhang, Anand Kumar Pothula, and Renfu Lu. A Review of Bin Filling Technologies for Apple Harvest and Postharvest Handling. *Applied Engineering in Agriculture*, 34(4):687–703, 2018. ISSN 0883-8542. URL https://elibrary.asabe.org/abstract.asp?aid=49538&t=3.