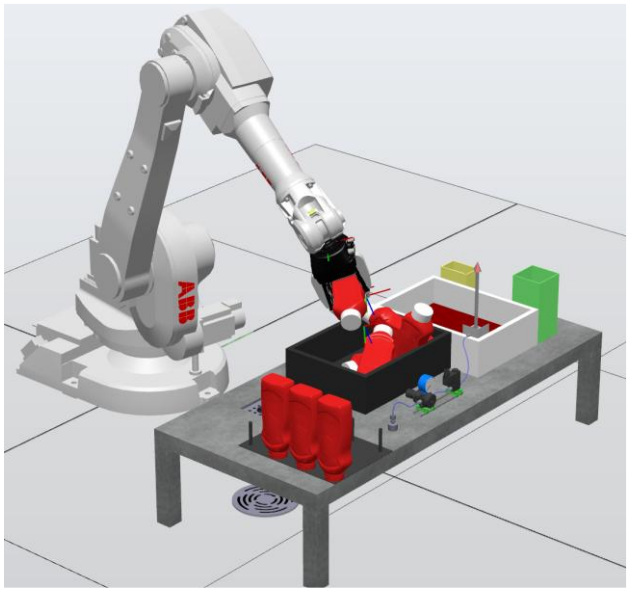
DEGREE PROJECT FOR MASTER OF SCIENCE WITH SPECIALIZATION IN ROBOTICS
DEPARTMENT OF ENGINEERING SCIENCE
UNIVERSITY WEST

Comparing Vision Methods for Pose Estimation in Robotic Bin-Picking:

- The RE:WORK Project – Product Recovery from Food Packaging using Flexible Cells

Mete Kurt Böler



UNIVERSITY WEST

A THESIS SUBMITTED TO THE DEPARTMENT OF ENGINEERING SCIENCE

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE WITH SPECIALIZATION IN ROBOTICS

AT UNIVERSITY WEST

2026

Date:	September 14, 2026
Author:	Mete Kurt Böler
Examiner:	Mikael Ericsson
Advisor:	Anders Appelgren
Programme:	Master Programme in Robotics
Main field of study:	Automation with a specialization in industrial robotics
Credits:	60 Higher Education credits (see the course syllabus)
Keywords:	6DoF pose estimation, Robotic bin-picking, Deep learning, Object localisation, Occlusion, Feature-Based Methods, Model-Free, RGB-D, Gripper evaluation
Template:	University West, IV-Master 2.10
Publisher:	University West, Department of Engineering Science S-461 86 Trollhättan, SWEDEN Phone: + 46 520 22 30 00 Fax: + 46 520 22 32 99 Web: www.hv.se

Summary

The food industry faces urgent sustainability challenges, prompting initiatives like RE:WORK to automate food recovery from packaging. For SMEs, this requires cost-effective, flexible robotic cells capable of handling mixed-texture, reflective objects in unstructured bins. This thesis investigates the feasibility of vision-based 6-DoF pose estimation for rigid food packaging, comparing FoundationPose against a custom hybrid method (SuperGlue-Enhanced MultiCue) across graded clutter levels using an Intel RealSense D435 camera.

Results demonstrate that FoundationPose significantly outperforms the hybrid method, achieving an overall precision of 43.3% compared to 21.1% under a 10°10cm criterion. FoundationPose proved highly robust in high-clutter scenarios (60.0% vs. 11.8%) by leveraging neural implicit representation to handle severe occlusion. Conversely, the hybrid method's reliance on visible geometric labels limited its performance in dense configurations. A critical discovery was the "low-clutter paradox," where sparse configurations degraded performance for both methods due to sensor-algorithmic degeneracy. Furthermore, dynamic sequential evaluation revealed that detection reliability (recall) dropped to ~16–17% for both methods, indicating that operational feasibility in continuous workflows is limited by detection frequency rather than pose accuracy.

In conclusion, while vision-based pose estimation is feasible, method selection depends heavily on occlusion levels and throughput requirements. The study establishes that symmetry-aware evaluation is strictly necessary to reflect physical graspability. To bridge the gap between perception uncertainty and physical manipulation, adaptive underactuated grippers are identified as the optimal paradigm, providing the mechanical compliance needed to absorb pose errors while maintaining torsional stiffness for downstream decapping. Finally, the end-to-end feasibility of the vision-to-robot pipeline was validated through a simulated digital twin in ABB RobotStudio, confirming robust socket communication and manipulation sequencing.

Affirmation

This master's degree report, *Comparing Vision Methods for Pose Estimation in Robotic Bin-Picking:*, was written as part of the master's degree work needed to obtain a Master of Science with specialization in Robotics and Automation degree at University West. All material in this report, that is not my own, is clearly identified and used in an appropriate and correct way. The main part of the work included in this degree project has not previously been published or used for obtaining another degree.

Mete Kurt Böler	2026-09-13
Signature by the author	Date

Contents

Main Chapters

SUMMARY III

AFFIRMATION IV

CONTENTS V

SYMBOLS AND GLOSSARY VII

1 INTRODUCTION 1

1.1 Aims and Research Questions 1
1.2 Contributions 2
1.3 Limitations 2

2 THEORETICAL FRAMEWORK AND RELATED WORK 4

2.1 Fundamentals of 6D Pose Estimation 4
2.2 Perception Challenges in Robotic Bin-Picking 5
2.3 Methodological Categories and Characteristics 7
2.4 Grasp Planning & Manipulation Considerations 9
2.5 Summary & Knowledge Gap 11

3 METHOD 15

3.1 Vision Method Selection Process & Criteria 16
3.2 Hardware, Software & Experimental Environment 21
3.3 Implementation & Validation Workflow 22
3.4 Experimental Setup & Scenarios 23
3.5 Experimental Data & Methods 27
3.6 Preprocessing & Validation 30
3.7 Evaluation Methods & Metrics 31
3.8 Error Sources & Uncertainty Analysis 33
3.9 Reproducibility Protocol 36
3.10 Specification-Based Gripper Evaluation Methodology 37

4 IMPLEMENTATION & SYSTEM INTEGRATION 40

4.1 Pipeline Architecture and Dual-Mode Execution 40
4.2 Hardware Configuration and Two-Photo Strategy 41
4.3 ArUco-Based Ground Truth System 42
4.4 Vision Method Implementation 44
4.5 Unified Scoring System 48
4.6 Evaluation Methodology 50
4.7 Simulation and RobotStudio Integration 53
4.8 Interactive Parameter Tuning and Validation 56
4.9 Computational Infrastructure and Cloud-Based Inference 57
4.10 Directory Structure and Parameter Locking 57
4.11 Execution and Data Collection 58

5 RESULTS AND DISCUSSION 59

5.1 Specification-Based Gripper Evaluation: Results and Recommendation 59

5.2 Quantitative Evaluation of Vision Methods 62
5.3 Discussion and Analysis 70
6 CONCLUSION 74
6.1 Conclusions 74
6.2 Future Work and Research 75
6.3 Critical Discussion 78
6.4 Generalization of Results 80
7 REFERENCES 82
A. APPENDIX 84

Symbols and glossary

6D/6-DoF Six Degrees of Freedom. Refers to the six independent parameters (3 for translation: x, y, z; and 3 for rotation: roll, pitch, yaw) that fully define an object's spatial configuration in 3D space.

ABB Robot Studio A robotics simulation and offline programming software developed by ABB, used for modelling robotic cells, validating process flows, and testing vision-to-robot pose transfer in virtual environments.

ADD A pose accuracy metric that calculates the average Euclidean distance between corresponding points on a 3D model transformed by the estimated pose versus the ground truth pose. A detection is considered successful if the error is below 10% of the object's diameter.

ADD-S A symmetry-aware variant of ADD that calculates the minimum distance from each transformed model point to the nearest point in the ground truth point cloud, treating geometrically equivalent poses (due to rotational symmetry) as valid.

ArUco A minimal marker-based system for pose estimation that uses square fiducial markers with binary patterns. Provides deterministic, closed-form pose recovery via Perspective-n-Point (PnP) solvers with low computational overhead.

BundleSDF A neural implicit surface reconstruction framework that builds a 3D representation (NeRF/SDF) of an object from a small set of reference images. Used in this study to generate the initial 3D mesh for the FoundationPose method.

BOP **(Benchmark for 6D Object Pose Estimation)** A standardized framework and challenge for evaluating 6D object pose estimation methods. The evaluation metrics (such as Recall score and Precision) in this study are adapted from the BOP challenge protocols.

CAD **(Computer-Aided Design)**
Digital 3D representations of objects with precise geometric specifications. Used in model-based pose estimation methods for matching sensor data against known geometry.

CLAHE **(Contrast Limited Adaptive Histogram Equalization)** An image processing algorithm used to improve the contrast in images. Applied in this study to the red channel of the high-resolution images to enhance the local contrast of ArUco markers before detection.

DoF **(Degrees of Freedom)** The number of independent parameters required to specify an object's configuration in space.

DRM **(Design Research Methodology)**
A structured research framework providing guidelines for engineering research, including problem clarification, artifact development, and empirical validation phases.

FoundationPose A modern few-shot learning method for 6-DoF pose estimation that generalizes to novel objects using only ~12-16 reference images without requiring CAD models or task-specific fine-tuning. Utilizes neural implicit representation and render-and-compare refinement.

Google Cloud A cloud-based platform providing GPU-accelerated computing resources. Used in this project for FoundationPose Phase 1 (NeRF reconstruction via BundleSDF) and for attempted 3D reconstruction via MASt3R.

HSV **(Hue, Saturation, Value)** A colour space that represents colours in a way that is more intuitive for segmentation than RGB. Used extensively in this study's preprocessing pipeline to isolate the red and white regions of the target object's labels.

ICP **(Iterative Closest Point)** An iterative algorithm for refining pose estimates by minimizing the distance between corresponding points in two point clouds.

Intel RealSense D435 An RGB-D camera utilizing active stereo vision with optional infrared projection to deliver synchronized color and depth streams. Selected for this project due to its cost-effectiveness and balance between resolution and noise characteristics.

JSON **(JavaScript Object Notation)** A lightweight data interchange format used in this project for storing pose estimation results, detection confidences, and metadata in a structured, human-readable format.

NeRF **(Neural Radiance Field)** A method that synthesizes novel views of complex scenes by optimizing an underlying continuous volumetric scene function using a sparse set of input views. Used in this project (via the BundleSDF pipeline) to generate the 3D proxy mesh required for FoundationPose inference.

OpenCV **(Open Source Computer Vision Library)** An open-source library providing comprehensive tools for computer vision tasks, including camera calibration, feature detection, pose estimation (PnP, ArUco), and image processing. Used extensively across Methods 1, 3, and 4.

PCA **(Principal Component Analysis)** A statistical procedure that uses an orthogonal transformation to convert a set of observations into a set of linearly uncorrelated variables called principal components. In this study, it is used to estimate the primary geometric axes of the object from 3D point cloud data.

PnP **(Perspective-n-Point)** An algorithm for estimating the pose (position and orientation) of a camera from a set of known 3D points and their corresponding 2D projections in the image. Used by ArUco for closed form pose recovery.

PPF **(Point Pair Features)** A geometric descriptor encoding the distance and normal directions between pairs of points in a 3D point cloud.

PNG **(Portable Network Graphics)** A lossless image format. RGB images are saved as standard PNG, while depth maps are exported as 16-bit PNG to preserve metric accuracy at millimetre scale.

Precision: A performance metric representing the proportion of reported detections that are correct ($Tp/(Tp + Fp)$). It indicates the reliability of the vision system when it decides to act.

PyTorch An open-source machine learning framework used for implementing and running deep learning-based methods. FoundationPose is built on PyTorch.

RANSAC **(Random Sample Consensus)** A robust estimation algorithm that iteratively selects random subsets of data to fit a model, filtering out outliers and identifying inliers that support the model.

RAPID The proprietary programming language used for ABB industrial robots. Utilized in this study to program the simulated manipulation, grasping, and downstream process sequences within the ABB RobotStudio digital twin.

Recall score: Adapted from the BOP benchmark, this metric represents the fraction of test targets (ground truth instances) for which a correct pose is estimated ($Tp/(Tp + Fn)$). It indicates the system's completeness in finding accessible objects.

RE:WORK An initiative by Sweden Food Arena aimed at developing automated solutions for recovering sorted food volumes from packaging and returning them to production, enhancing circularity in the food value chain.

RGB-D Sensor data combining **Red-Green-Blue** colour information with per-pixel **depth** measurements. Enables fusion of appearance and geometric cues for robust pose estimation.

SDK **(Software Development Kit)** A collection of software tools and libraries for developing applications for specific hardware. The Intel RealSense SDK provides access to camera intrinsics, calibration data, and streaming controls.

S-E-M **SuperGlue-Enhanced MultiCue**: A custom hybrid 6-DoF pose estimation method developed in this study. It combines geometric 3D-PCA pose estimation with deep learning-based orientation classification (using SuperPoint and SuperGlue) to resolve the 180° rotational ambiguity inherent in pure geometric approaches on symmetric objects.

Smart Components Virtual devices and logic elements in ABB RobotStudio used to simulate sensor feedback, object spawning, and dynamic scene manipulation. Used in this study to dynamically position virtual bottles at received pose coordinates within the digital twin.

SME **(Small and Medium-sized Enterprises)** Businesses with limited employees and revenue. The target beneficiaries of this project, as they face barriers to robot adoption including low capitalization and limited technical competence.

SuperGlue/ SuperPoint A deep learning-based feature matching framework. SuperPoint detects and describes keypoints in an image, while SuperGlue uses a graph neural network with optimal transport to match these keypoints between images. Used in this study to resolve the 180° rotational ambiguity of the geometric pose estimation.

SVD **(Singular Value Decomposition)** A matrix factorization method used to compute the Principal Component Analysis (PCA). In this study, SVD is the mathematical engine that extracts the eigenvectors representing the object's orientation, which is central to the discussion on rotational ambiguity and the "low-clutter paradox".

TCP **(Tool Centre Point)** The reference point on a robot's end-effector used for motion control and calibration. Requires preliminary calibration to define the tool frame relative to the robot flange.

TCP/IP **(Transmission Control Protocol/Internet Protocol)** A suite of communication protocols used to interconnect network devices. Used in this study to establish socket communication between the Python vision pipeline and the RAPID robot controller.

Two-Photo Strategy A data acquisition technique employed to satisfy conflicting resolution requirements. Photo 1 (1920×1080 RGB) is captured for high-resolution ArUco ground truth establishment, while Photo 2 (1280×720 RGB-D) is captured for vision method processing at native depth resolution, avoiding interpolation artifacts.

WSL2 **(Windows Subsystem for Linux)** A compatibility layer for running Linux binary executables natively on Windows. Used in this project's computational setup to run the local CPU/GPU vision pipelines (ArUco and SuperGlue) while maintaining access to the host system's hardware.

1 Introduction

The food industry faces significant sustainability challenges, with approximately one-third of all food produced globally being lost or waste[1]. This contributes to substantial environmental impact, economic inefficiencies, and resource waste across the value chain. Sweden's 2024 bio-waste legislation [2], requiring source separation of organic waste from packaging, creates an urgent need for scalable, automated sorting solutions that enable compliance and circular resource flows. However, implementing such automation remains difficult for many small and medium-sized enterprises. Recent research highlights that barriers such as low technical competence outside the core business, low capitalization problems, and difficult access to finance hinder robot adoption in such companies; financially attractive, lightweight robots that can be easily moved between processes and reprogrammed as needed offer a pathway to overcome these constraints [3].

This degree project is conducted within the RE:WORK framework [4], an initiative by Sweden Food Arena addressing the strategic need to minimize food waste and enhance circularity across the value chain [5]. The framework aims to develop automated solutions that enable "Rework"—the recovery of sorted food volumes for return to production. Within this context, the present thesis evaluates machine vision methods for pose estimation in cluttered bin configurations, with the goal of enabling flexible, conveyor-free robotic cells that can separate packaging from food content. To address this challenge, this thesis pursues the following aims and research question:

1.1 Aims and Research Questions

Main Goal:

The primary aim of this project is to implement and evaluate vision methods across analytical and data-driven paradigms for 6-DoF pose estimation of rigid food packaging in cluttered bin configurations. The evaluation includes controlled experiments measuring detection frequency, pose accuracy, with conditional validation through simulation and, if feasible, physical hardware. Additionally, the project aims to survey industrial gripper types suitable for handling the target object, evaluating their documented mechanical specifications against the object's physical properties (mass, dimensions, surface material).

Sub-goals:

1. **Conduct a literature synthesis** on 6-DoF pose estimation paradigms to establish technical prerequisites and method-specific filtering criteria.
2. **Implement and configure three vision methods** (ArUco, FoundationPose, and SuperGlue-Enhanced MultiCue) for processing synchronized RGB-D data, including preprocessing pipelines, parameter tuning, and unified output formatting.
3. **Perform a controlled experimental evaluation** across three static clutter levels (3, 5, and 8 objects) and a dynamic sequential picking protocol (8 down to 0

objects), with quantitative assessment of pose accuracy (translation and rotation errors), detection frequency, and coordinate stability.

4. **Identify candidate gripper types** based on documented mechanical principles and class-level characteristics and assess their mechanical suitability for the target object geometry.
5. **Validate vision-to-robot pose transfer through simulation in ABB Robot Studio** as the primary technical validation, with a conceptual animation to illustrate the process flow; conditional physical robot validation will be conducted per the criteria defined in Section 1.3.

Research Question:

The project addresses the following central research question: To what extent is vision-based 6-DoF pose estimation feasible for detecting and localizing rigid food packaging in cluttered bin configurations?

1.2 Contributions

This degree project provides computational and simulation-based insights into the intersection of flexible robotics, vision-based pose estimation, and sustainable waste handling. The primary contributions are:

- **Scientific:** A structured evaluation framework for comparing vision-based bin-picking methods under realistic hardware constraints. The framework enables empirical assessment of detection performance and pose estimation accuracy for rigid food packaging in cluttered configurations, while documenting limiting factors for vision-based automation in waste handling contexts.
- **Practical:** A specification-based evaluation of gripper types suitable for rigid, glossy squeeze-bottle geometry. The analysis identifies mechanical prerequisites (payload, geometry accommodation, surface compatibility) for reliable handling, derived from literature synthesis and a mechanical analysis of the object's physical properties.

1.3 Limitations

This study is subject to the following limitations that define its scope:

1. **Object scope**: The evaluation uses a single representative object: a standard 1 kg ketchup bottle (Brand: Eldorado Tomat Ketchup, dimensions: 235 mm length × 96 mm maximum width × 64 mm maximum depth, mass: 1 kg when full, smooth glossy red surface, white screw cap). This object serves as a case study of rigid food packaging. Objects with significantly different geometries, materials, or surface properties are outside project scope. Physical deformation of the container is neglected in the perception algorithms, consistent with the rigid-body assumption.
2. **Gripper evaluation scope:** Gripper suitability is assessed solely through theoretical matching of documented paradigm characteristics and mechanical principles to object properties. Physical grasp testing, dynamic alignment tolerance analysis, or experimental validation of grasp success rates are not included.
3. **Conditional physical robot validation**: Physical robot validation depends on the availability of hardware, successful camera-to-robot calibration, and time

constraints. Regardless of whether physical validation is conducted, simulation in ABB RobotStudio serves as the primary validation method, focused on vision-to-robot pose transfer and process-flow illustration via a digital twin with socket communication.

4. **Approximated ground truth techniques**: Ground truth establishment relies on approximated measurement techniques due to the absence of calibrated robotic positioning in the experimental setup, which limits absolute accuracy claims but maintains validity for comparative evaluation.
5. **Implementation fallback**: If experimental implementation of the selected vision methods proves infeasible within project constraints, the evaluation will rely on literature-reported performance data to ensure methodological continuity.

2 Theoretical Framework and Related Work

2.1 Fundamentals of 6D Pose Estimation

Definition and scope

In computer vision and robotics, 6D pose estimation (also termed 6-DoF) refers to the task of extracting the target area from input sensor data and determining the object's full spatial configuration, specifically encompassing three-degree-of-freedom translation and three-degree-of-freedom rotation [6]. Unlike two-dimensional representations that capture only planar position, 6D pose estimation determines an object's full spatial configuration by quantifying both its three-dimensional position and orientation [6]. Mathematically, this corresponds to six degrees of freedom: three for translation (x, y, z) defining spatial location, and three for rotation defining orientation. While translation is directly expressed as a 3D vector, rotation can be parameterized through established compact forms such as rotation matrices, quaternions, or Euler angles [6]. Regardless of the chosen formalism, these representations encode three independent rotational degrees of freedom.

Importance in industrial applications

Effective robotic grasping in industrial bin-picking relies on accurate 6D pose estimation to determine object position and enable precise manipulation [6]. As automation becomes increasingly critical for productivity and operational safety, 6D pose estimation has emerged as a core enabling technology. Beyond direct manipulation, accurate pose data reduces environmental uncertainty through two primary mechanisms: lowering the probability of robot collisions and providing necessary support for object transportation and related handling activities [6].

Standard estimation pipeline

A standard pipeline for vision-based pose estimation typically follows a sequential structure of feature extraction, correspondence matching, and geometric solving [6]. The process begins by acquiring geometric primitives such as points, edges, and lines directly from input images or point clouds. These features are then matched against a reference model, enabling the spatial relationship to be resolved by measuring corresponding coordinates across multiple spatial coordinate systems. Prior to geometric solving, camera calibration parameters—intrinsic parameters for correcting lens distortion and extrinsic parameters for relating camera to scene coordinates—are applied to map 2D image measurements into three-dimensional space [7]. The final 6D pose is then computed using algorithms such as the Perspective-n-Point (PnP) method, which determines the camera's pose from known 3D spatial points and their 2D projections [6].

Calibration and coordinate frames

Accurate pose measurement requires calibration of both camera intrinsic parameters and the vision system's external parameters relative to the robot workspace [7]. Camera calibration determines intrinsic parameters—such as focal length and distortion coefficients—enabling the geometric mapping between three-dimensional world coordinates and two-dimensional image coordinates according to the pinhole camera model, which ensures that captured images are of accurate dimensions and free of lens distortion [7]. Subsequently, hand-eye calibration establishes the spatial relationship between the camera and the robot—regardless of whether an eye-in-hand (camera mounted on the end-effector) or eye-to-hand (camera fixed in the workspace) configuration is employed—by computing the relative 3D position and orientation, expressed as a homogeneous transformation, between two coordinate frames: one centred at the camera lens and the other at the robot end-effector [7]. This problem is mathematically formulated as AX=XB, where A and B represent known relative motions of the camera and robot hand between two configurations, respectively, and X denotes the unknown transformation to be solved [7]. The calibration operation involves obtaining sets of corresponding camera and robot hand poses, where multiple motion pairs are required to accurately estimate the system variables by solving the resulting overdetermined equation[7]. Together, these procedures ensure reliable mapping between visual measurements and robotic coordinate frames, which is particularly crucial when sub-millimetre perceptual accuracy is required [7].

2.2 Perception Challenges in Robotic Bin-Picking

Textureless surfaces

In robotic bin-picking, an object is considered textureless in the absence of any texture, which challenges feature-based matching methods that require discriminative visual cues sensitive to intra-class variations such as colour and texture [8]. These perception challenges are compounded in industrial scenarios, which are characterised by extreme conditions including varying visibility and the typical presence of reflective and irregular objects [8]. To address these limitations, analytical approaches such as edge-based methods that leverage image contours and region-based methods that utilise statistical colour information have been shown to provide robust localisation for textureless objects [8]. Furthermore, the integration of 2.5D and 3D data, particularly through point clouds, has enabled substantial improvements in gras prediction for complex bin-picking scenarios involving occlusion and textureless surfaces [8].

Occlusion and clutter

High levels of clutter in industrial bin-picking scenarios are characterised by densely packed objects that cause significant mutual occlusion, frequently obscuring critical geometric cues [8]. This widespread occlusion limits the visible surface area available for analysis, making the capture of complete object geometry challenging [8]. To address these limitations, the literature identifies several effective strategies: two-stage coarse-to-fine frameworks that enable hierarchical refinement under partial visibility [8], voting-based pose aggregation that resolves ambiguity from occluded surfaces [8], and iterative refinement techniques that progressively correct initial pose hypotheses [8]. Additionally, hybrid approaches that combine analytical optimisation with data-driven

components demonstrate superior robustness in heavily cluttered scenes [8]. Cross-modal fusion further mitigates the impact of missing or corrupted data by leveraging complementary geometric and appearance information across RGB and depth modalities [8].

Geometric symmetry and rotational ambiguity

In robotic perception, rotational symmetry occurs when multiple distinct object orientations produce identical visual or geometric appearances from a given viewpoint, inherently introducing pose ambiguity [6]. Industrial bin-picking environments frequently contain textureless and symmetric objects, which compound this challenge by depriving feature-based algorithms of discriminative surface cues and generating indistinguishable projections across multiple viewpoints [6]. This ambiguity disrupts standard estimation pipelines, as mathematically distinct poses cannot be uniquely resolved from single-view observations. In feature-based and geometric approaches, symmetry generates redundant pose hypotheses that complicate clustering and voting mechanisms, while data-driven methods struggle to distinguish between equivalent symmetric configurations without explicit constraints [6].

Consequently, conventional evaluation metrics are inadequate unless adapted to remain invariant under an object's symmetry group. To address this, perception frameworks employ symmetry-adapted metrics such as Average Distance of model points (ADD) and its symmetry-aware variant (ADD-S), which computes the average minimum distance between transformed model points and the closest ground-truth surface points, rather than enforcing rigid one-to-one correspondence [6]. Benchmark evaluations confirm that estimation success rates decline significantly when symmetry-aware strategies are omitted [6]. To facilitate systematic validation, standardised datasets such as T-LESS, YCB-Video, and Occluded-LINEMOD (LM-O) have been established, providing rigorous testbeds that explicitly incorporate symmetric, textureless, and heavily occluded objects for performance assessment [6].

Sensor limitations

Beyond object-specific challenges, the physical characteristics of consumer-grade RGB-D sensors introduce constraints that affect pose estimation accuracy. For active stereo configurations, depth noise typically increases quadratically with distance [9], and the no-data 'shadowing' region caused by non-overlapping stereo views is larger at close range (e.g., ~0.4 m) [9]. Furthermore, the active illumination used to project texture onto unstructured scenes is prone to saturation when imaging glossy surfaces with specular reflections, which can result in missing depth values or outliers [8]. These sensor limitations compound the challenges posed by textureless surfaces and clutter, necessitating pose estimation methods that are robust to noisy or incomplete depth data. This limitation is particularly pronounced for industrial objects with reflective materials, where active illumination frequently saturates, rendering depth streams unreliable [8]. Consequently, perception pipelines that incorporate RGB-driven or edge-based fallback mechanisms are essential for maintaining robustness when depth data is compromised, as cross-modal fusion strategies can compensate for missing information in one modality by leveraging complementary geometric and appearance cues [8].

Clutter taxonomy

To systematically evaluate pose estimation methods, literature classifies bin-picking scenarios based on environmental complexity. Cordeiro et al. [8] propose a qualitative taxonomy that categorises environments according to object density and degree of mutual occlusion, described as follows:

- **Low-level clutter:** Characterised by well-spaced objects with minimal occlusion, where geometric features remain largely visible and simpler RGB-based methods often perform reliably [8].
- **Medium-level clutter:** Defined by numerous objects with substantial occlusion counts but typically low-to-moderate overlap between instances, representing typical industrial handling tasks where items are grouped but not completely buried [8].
- **High-level clutter:** Represents the most extreme scenario, with unstructured environments abundant in bulk objects and extensive occlusion, necessitating robust perception strategies such as deep learning or advanced sensor fusion [8].

This taxonomy establishes that bin-picking complexity is determined by object spacing, instance overlap, and mutual occlusion levels. As environmental density increases, the visibility of discriminative geometric features diminishes, implying a need for progressively more robust perception strategies [8]. This framework informs the experimental design of the present study, which employs systematically graded scenarios to evaluate algorithmic performance across these defined clutter levels.

2.3 Methodological Categories and Characteristics

Methodological categories and taxonomies

6D pose estimation methods are broadly categorised into two primary approaches: instance-level and category-level [6], [8]. A key distinction in the literature lies in how these approaches are further classified. Cordeiro et al. classify methods into instance-level, category-level, and model-free approaches, where instance-level methods typically require dedicated CAD models for high precision, while model-free approaches operate without prior object knowledge [8].In contrast, Guan et al. structure their taxonomy around in-put data modalities (RGB, point cloud, or RGB-D) and methodological techniques, such as regression-based, template-based, and feature-based methods [6]. Despite these structural differences, both reviews agree that instance-level methods are designed to detect and localise specific object instances, typically requiring a dedicated 3D model for each object to achieve high precision [6], [8]. Category-level methods, conversely, aim to estimate the poses of arbitrary shapes within a known object class, enabling the handling of new variants within the same category without requiring instance-specific CAD models, though often facing challenges with intra-class variations and generally exhibiting lower precision compared to instance-level counterparts [6], [8]. When models are utilised, pose candidates are generated by matching sensor data against precise 3D representations, yielding high geometric reliability [8]. Model-free approaches forgo this prior knowledge, relying instead on general visual cues, but tend to be less stable in complex environments [8]. Deep learning-based solutions offer flexibility but demand significant computational resources for both training and inference

[6]. Supervised learning approaches typically require large, annotated datasets, which represent a significant bottleneck for industrial deployment [6]. To balance these limitations, model-based geometry is sometimes combined with data-driven components, which the literature indicates results in more robust systems overall [6], [8]. However, both approaches face challenges with textureless surfaces, specular reflections, and symmetric objects, which can degrade pose estimation accuracy [6], [8]. In multi-object scenarios, instance-level methods typically prioritize high-precision localization, while category-level approaches can handle multiple instances simultaneously but with reduced precision [6], [8].

Model and data dependencies

A critical differentiator across paradigms is their dependency on prior models and training data. Traditional geometric approaches operate without task-specific training; instead, they depend on accurate CAD models to extract and match reference features directly, making them highly suitable for industrial settings where product designs are available [6], [8]. Conversely, hybrid approaches combine model-based geometry with data-driven components to leverage the strengths of both paradigms, though they require careful balancing of computational complexity and annotation requirements [6], [8].

Purely supervised data-driven methods typically require large, annotated datasets, which represent a significant bottleneck for industrial deployment [6]. To address this limitation, researchers increasingly utilize synthetic data generation pipelines, including Physically Based Rendering techniques, though bridging the domain gap between simulation and reality remains challenging [6]. Recent work explores self-supervised learning to reduce annotation dependencies, yet high-quality geometric supervision remains necessary for achieving reliable pose estimation performance [6], [8].

Generalization and multi-object handling

The capacity to generalize to unseen objects and handle multi-object scenes varies fundamentally across categories. Instance-level methods achieve high precision for known objects but typically fail when encountering novel categories without retraining or dedicated CAD models [6], [8]. Category-level and model-free approaches attempt to learn generalized shape priors or grasp affordances, enabling operation on unseen objects, but typically sacrifice metric pose accuracy for flexibility [6], [8]. In multi-object bin-picking scenarios, traditional pipelines process items sequentially via a detect-then-pose workflow, which ensures stability but at the cost of reduced efficiency [6]. Modern learning-based frameworks increasingly support simultaneous multi-instance pose estimation through instance segmentation masks or dense key point voting, improving throughput; however, performance degrades under severe mutual occlusion and dense clutter, necessitating robust iterative refinement or multi-view strategies [6].

Fiducial marker systems

Beyond instance-level and category-level paradigms, fiducial marker systems represent a deterministic, pattern-based approach to pose estimation. These methods bypass texture and occlusion challenges by utilizing high-contrast, geometrically encoded references that enable closed-form pose recovery via Perspective-n-Point solvers. Experimental comparisons of widely adopted open-source marker packages demonstrate that

ArUco achieves high detection rates and competitive localization accuracy while maintaining significantly lower computational overhead than alternative frameworks [10]. This efficiency makes it particularly suitable for resource-constrained platforms and establishes it as a reliable baseline for coordinate validation in robotics applications [10]. While not designed for unstructured bin-picking of textureless objects, fiducial systems provide a critical reference standard for evaluating the accuracy and stability of more complex perception pipelines under controlled conditions [10].

2.4 Grasp Planning & Manipulation Considerations

Object requirements and constraints

The transition from pose estimation to physical handling requires matching the object's physical properties to available end-effector specifications. This section outlines the mechanical criteria used to identify gripper types suitable for the target object, independent of vision system output.

The target object has a mass of approximately 1 kg. Industrial grippers must therefore provide a rated payload exceeding this value, typically incorporating a safety margin (e.g., ≥2 kg) to account for dynamic accelerations during pick-and-place cycles. Grip force specifications must ensure sufficient friction to prevent slippage, particularly on smooth plastic surfaces.

The object has a squeeze-bottle geometry with approximate dimensions of 235 mm height, 90 mm width, and 60 mm depth. Mechanical grippers require a stroke or opening width sufficient to encompass the object's width, while vacuum-based systems require cup configurations capable of establishing adequate seal area on curved or flat sections of the packaging. The physical envelope of the gripper must also accommodate the object's height without collision during approach.

The object features a glossy, textureless plastic surface. Mechanical parallel-jaw grippers rely on friction coefficients that may be reduced on smooth plastics, potentially requiring textured finger pads or increased normal force. Vacuum grippers require non-porous, relatively smooth surfaces to maintain pressure differentials, making them mechanically suitable for this material class provided surface curvature does not compromise seal integrity.

Scope of assessment

While vision methods estimate full 6-DoF pose, the mechanical suitability of a gripper is primarily determined by its ability to physically enclose or adhere to the object geometry. This survey identifies gripper types that meet baseline mechanical criteria; quantitative mapping of pose uncertainty to mechanical tolerance envelopes is outside the scope of this specification-based assessment.

Industrial gripper taxonomy

The selection of an appropriate end-effector is a critical determinant of overall manipulation efficiency and performance in robotic systems [11]. To ensure optimal operation, multiple interacting factors must be considered, including the object's shape, geometry, material stiffness, and weight, as well as manipulation criteria such as speed and task-specific constraints [11].

Industrial grippers can be classified along several dimensions. Based on mechanical configuration, grippers range from basic two-fingered designs—valued for their simplicity, cost-effectiveness, and suitability for standard pick-and-place operations—to three-fingered grippers offering enhanced precision for fragile objects, multi-fingered adaptive hands capable of grasping irregularly shaped or soft objects, bellows grippers designed for cylindrical surfaces, and grain-filled flexible ball grippers that conform to diverse geometries through form-fitting [11]. From an actuation perspective, grippers may be driven by cables, vacuum, pneumatics, hydraulics, or servo-electric motors, each presenting distinct trade-offs in force capacity, dimensional compactness, controllability, and maintenance requirements [11]. Based on material stiffness, rigid grippers dominate traditional industrial environments where high precision and large gripping forces are required, whereas soft or flexible grippers offer the ability to manipulate a larger variety of object shapes, making them particularly suitable for delicate or irregularly contoured items [11].

Tolerance analysis framework

The precision of a pick-and-place operation cannot be assessed solely through manipulator repeatability; it requires a systematic tolerance analysis of the entire robotic work cell from the operation point of view [12]. The cumulative tolerance stack-up can be modelled as a transformation chain of low-order kinematic pairs linking the workpiece, the gripper, the manipulator, and other work cell components, incorporating tolerances from manipulation, seizing, releasing, manufacturing, and metrology processes [12]. In flexible applications such as bin-picking, where the workpiece pose is not known precisely in the design phase, the precision of the metrology system used to resolve pose uncertainties constitutes an additional tolerance component in the chain [12].

To compensate for accumulated deviations, mechanical guiding features—such as chamfers, cones, or kinematic constraints introduced by closing gripper fingers—can act as self-alignment mechanisms that saturate tolerances in guided directions and improve achievable precision during the seizing or releasing phase [12]. By evaluating the transformation chain via Monte Carlo simulation, designers can predict failure rates at critical contact points and validate the feasibility of functional requirements in the early design phase. Experimental validation has demonstrated a strong correlation between simulated tolerance predictions and physical execution outcomes [12].

Application to the RE:WORK context

In the context of the present study, the tolerance analysis framework described above implies that the gripper must not only satisfy static payload and geometric requirements but also accommodate the positional uncertainties inherent to vision-based pose estimation. The seizing and releasing processes introduce geometric errors through contact transitions between components [12], and the selection of gripper configuration—rigid versus compliant, parallel versus adaptive—directly influences the magnitude of these errors and the system's ability to self-align during grasping [11], [12]. Consequently, the gripper evaluation presented in Chapter 3.10 assesses candidate paradigms not only against the object's physical properties but also against their capacity to tolerate the pose estimation uncertainties documented in the experimental evaluation. Furthermore, within the RE:WORK framework, bin-picking is not an isolated task but the initial step of a sequential recovery process that extends beyond the grasp itself

to include downstream operations such as opening, emptying, rinsing, and placement. Given the project's target context of small and medium-sized enterprises—where limited capital and technical competence constrain the complexity of viable system architectures—a single-robot system employing a stationary, form-fitting fixture for these downstream operations represents a realistic and robust premise. This architectural premise directly influences the gripper selection, as the end-effector must be capable of securely transporting the object and subsequently interfacing with the stationary fixture without inducing kinematic conflicts or excessive cycle times.

2.5 Summary & Knowledge Gap

Literature synthesis on pose estimation

The absence of surface texture is identified as a primary limitation for feature-based and regression-based methods, as it deprives algorithms of discriminative visual cues [6], [8]. Consequently, when conventional feature matching is unfeasible, literature recommends shifting toward analytical approaches that rely on geometric contours, edge-based segmentation, or region-based statistics [6], [8]. Furthermore, integrating depth information with RGB data (RGB-D) is emphasized as a critical strategy to enhance robustness in complex industrial environments compared to single-modality pipelines, as geometric depth cues compensate for the lack of colour features [6], [8].

Hybrid architectures and model-free geometric approaches are identified as the most viable solutions for industrial bin-picking, offering a balance between robustness and computational efficiency [6], [8]. Hybrid frameworks that combine data-driven localisation with analytical pose estimation address precision requirements while maintaining adaptability to unseen objects, without the heavy dependency on large-scale annotated datasets [8]. Concurrently, model-free geometric pipelines, such as those utilising Point Pair Features (PPF), enable precise pose estimation directly from sensor data without requiring explicit CAD models, thereby ensuring real-time performance in resource-constrained environments [6], [8]. Furthermore, methodologies that leverage edge-based information have demonstrated superior reliability for textureless objects, as they rely on geometric contours rather than appearance-based descriptors [6], [8].

Few-shot neural generalization

FoundationPose [13] exemplifies how unified foundation models can bridge model-based and model-free paradigms, achieving strong generalization to novel objects using either a CAD model or as few as ~12–16 reference images, without task-specific fine-tuning. Its neural implicit representation enables efficient novel-view synthesis for render-and-compare refinement, while large-scale synthetic training, augmented by language-guided texture generation and contrastive learning, yields robustness across challenging scenarios including textureless surfaces, dense clutter, and symmetric objects [13]. Although computationally demanding during inference (~1.3 s/object on GPU), such methods represent a significant step toward flexible, data-efficient pose estimation that maintains competitive accuracy with instance-level approaches despite reduced assumptions [13].

Transformer-based RGB-D fusion for occlusion robustness

Complementing few-shot approaches, Zhou et al. [14] propose a Deep Fusion Transformer (DFTr) network that aggregates cross-modality features by modelling global semantic similarity between appearance and geometry information. Their weighted vector-wise voting algorithm replaces iterative clustering with a non-iterative global optimization strategy, achieving near real-time inference (~48 ms/frame) while maintaining superior accuracy on benchmarks involving heavy occlusion, reflective surfaces, and symmetric objects [14]. By capturing long-range dependencies across RGB and depth modalities, the approach mitigates perturbations caused by missing data or sensor noise—critical for industrial bin-picking where single-modality pipelines often fail [14]. Although transformer-based architectures entail higher parameter counts than classical geometric methods, such fusion strategies represent a significant advance in handling the visual ambiguities inherent to densely cluttered environments.

Efficient PPF-based pipelines for industrial bin-picking

Complementary to foundation models, classical geometric approaches remain highly relevant for resource-constrained deployments. Yan et al. [15] demonstrate that a voting-based PPF pipeline, augmented with adaptive threshold segmentation and multilayer picking strategies, achieves 91.25% success rate and 0.265 s average estimation time on real-world bin-picking tasks involving textureless, thin, and partially occluded industrial objects. Their method constructs offline databases from either CAD models or real scene point clouds, enabling rapid adaptation to new object types without extensive retraining [15]. By leveraging geometric descriptors (contour, normal distribution, curvature) rather than appearance-based features, the approach maintains robustness under varying illumination and low-texture conditions—critical for industrial environments where RGB-only methods often fail [15]. Although less flexible than few-shot neural methods for novel categories, such geometric pipelines offer transparency, deterministic behaviour, and CPU-efficient execution that align well with the operational constraints of small-scale industrial systems.

Limitations of leading methods

Despite progress, significant barriers remain for industrial implementation. Leading learning-based methods typically require high computational resources, often with dedicated GPU infrastructure exceeding standard operational budgets [6]. Methods relying on generalisation often retain dependence on precise CAD models, creating a bottleneck when digital twins or engineering drawings are unavailable [8]. Furthermore, purely data-driven pipelines necessitate extensive training datasets or complex domain adaptation from synthetic to real-world environments, introducing significant annotation overhead and temporal constraints [6].

Identified knowledge gaps

Despite extensive research in 6D pose estimation, significant gaps remain in translating academic methods to practical, resource-constrained industrial automation. Based on the literature synthesis, two primary knowledge gaps are identified:

- **Resource and Implementation Gap in Practical Industrial Operations:** State-of-the-art methods are typically optimized for high-performance GPU environments and presuppose access to precise CAD models as well as extensive, manually annotated datasets [6], [8]. In practical industrial applications, these prerequisites are often constrained; real-world operational environments are characterized by reliance on standardized computational hardware, absence of detailed 3D models, and limited resources for data collection and annotation [3]. Consequently, there is a lack of validated, lightweight perception pipelines that combine model-free generalization with efficient inference on available hardware, without compromising robustness in complex environments.
- **Lack of Frameworks for Uncertainty Propagation to Gripper Execution:** Pose accuracy in the literature is reported almost exclusively via geometric metrics such as ADD and ADD-S [6], [8]. These quantify errors in translation and rotation but lack explicit connection to downstream manipulation. There is a clear shortage of methods that systematically map how perception uncertainties—such as detection frequency, coordinate frame stability, and rotational ambiguity—propagate to the kinematic tolerance requirements imposed on physical grippers. Consequently, the quantitative dependency between estimation errors and actual grasp success remains largely unaddressed.

These gaps directly inform the selection criteria and the comparative evaluation protocol presented in Chapter 3.

Linking challenges to system design

The challenges described in section 2.2, together with the manipulation constraints outlined in section 2.4, directly inform the filtering criteria applied in Section 3.1. The presence of textureless surfaces necessitates a shift from appearance-based matching to edge-driven and geometric voting strategies [6], [8]. High-clutter conditions further validate the selection of algorithms capable of aggregating sparse geometric cues rather than relying on dense feature correspondence [6]. The literature highlights that hybrid fusion architectures can improve robustness for textureless objects by leveraging both appearance-based and geometric cues [8].

These theoretical requirements align with practical limitations in resource-constrained industrial environments. The literature emphasises that methods for textureless objects must be robust to incomplete point clouds, sensor noise, and specular reflections – challenges particularly prominent when using cost-effective RGB-D sensors [6], [8]. While edge-based architectures exhibit strong performance on textureless objects [6], their hardware requirements often conflict with CPU-based evaluation constraints. Consequently, geometric approaches complemented by lightweight feature extraction remain viable for squeeze packages with limited symmetry when computational efficiency is prioritised [6]. Furthermore, reliable pose estimation is a prerequisite for downstream manipulation tasks, as inconsistent coordinate frames can lead to grasp failures or collisions [8].

Methodological motivation and concrete contribution

The identified gaps and operational constraints motivate a resource-efficient system design. By combining an analytical RGB-D fusion pipeline for pose estimation with

deep learning-based feature matching for orientation classification, the evaluation framework mitigates the computational burden of deep learning while maintaining robustness against texture absence and partial occlusion. This approach circumvents the CAD-dependence of generalisation methods and eliminates the annotation overhead of supervised networks, directly addressing the resource limitations characteristic of small-scale industrial environments [6], [8].

The concrete contribution of this work is a validated, hardware-constrained perception pipeline tailored for mixed-texture squeeze packages with high-contrast labels in cluttered bin configurations. By empirically benchmarking marker-based, geometric, few-shot, and hybrid fusion methods under standardised occlusion, this project establishes a reproducible evaluation framework bridging the gap between academic pose-estimation benchmarks and practical, budget-aware robotic integration. The minimal viable contribution ensures that perception outputs are rigorously benchmarked under standardized conditions, providing documented performance profiles for method selection in similar perception tasks. This design choice explicitly prioritizes deployability over absolute benchmark performance. While recent reviews note that end-to-end learning frameworks can achieve state-of-the-art accuracy, their substantial computational and data requirements often conflict with the hardware constraints of small-scale industrial environments [3], [6], [8]. The modular, hybrid architecture adopted here thus represents a deliberate trade-off, optimizing for transparency, maintainability, and CPU-efficient execution without sacrificing the robustness required for sequential bin-picking.

3 Method

Research design and experimental strategy

This thesis is structured according to the Design Research Methodology (DRM) framework [16], which provides the structural basis for this work: problem clarification and literature synthesis (Ch. 1-2), artifact development (Sec. 3.1-3.5), empirical validation (Sec. 3.6-3.9), and a complementary specification-based gripper evaluation (Sec. 3.10). Within this framework, the evaluation phase employs a controlled experimental design to establish causal relationships between algorithmic choices and performance outcomes. Following engineering research principles for systematic parameter variation [16], clutter density is manipulated across three levels (3, 5, and 8 objects) while all environmental and sensor parameters are held constant. By exposing each method to identical inputs, ground truth, and evaluation protocols, observed differences are isolated to algorithmic properties rather than external confounders. This comparative design directly addresses the central research question and aligns with established benchmarks for graded complexity in bin-picking scenarios, while adhering to reproducible laboratory practices and statistical validation protocols. To complement this static baseline, a secondary dynamic evaluation was designed. This protocol manipulates the scene sequentially from high clutter (8 objects) down to zero, capturing the compounding environmental variables of a real-world continuous workflow.

Evaluated methods

The three evaluated methods have been selected based on the selection criteria established for the study, with explicit links to established literature:

- **ArUco (Analytical, Marker-based):** Selected as a reliable reference baseline due to its established PnP pose estimation via OpenCV and well-defined coordinate frame [10]. In this study, the method serves as a practical control for the ground truth chain and calibration protocol.
- **FoundationPose (Data-driven, RGB-D):** Represents modern few-shot architectures that generalize to unseen objects with minimal reference data (~12-16 images) [13].
- **SuperGlue-Enhanced MultiCue (Hybrid, RGB-D):** A custom fusion strategy that combines geometric 3D-PCA pose estimation with deep learning-based orientation classification using SuperPoint and SuperGlue. This hybrid approach resolves the 180° rotational ambiguity inherent in pure geometric PCA by classifying label orientation (front/back), following literature recommendations for multimodal approaches that combine analytical and data-driven components [6], [8], [17], [18].

Experimental protocol

The experiments are conducted according to a strict chronological flow:

1. Camera calibration & ground truth establishment.
2. Data collection across three clutter levels.

3. Uniform preprocessing (native resolution preservation without spatial resizing or point cloud downsampling to maintain maximum geometric fidelity).
4. Sequential method execution (processing the vision pipelines one at a time to prevent GPU memory fragmentation and allow fair timing comparison) without parameter tuning against test data.
5. Metric evaluation with descriptive statistics (mean, median, min, max) for uncertainty quantification.

3.1 Vision Method Selection Process & Criteria

Technical prerequisites per method group

This section outlines the method selection process in four steps. First, the fundamental technical prerequisites for each methodological category are clarified, drawing on the perception challenges outlined in Section 2.2. Second, project-specific selection criteria are defined based on operational constraints and literature recommendations (Section 2.2). Third, these criteria are applied within established taxonomic frameworks to filter and identify methods that are both scientifically sound and practically implementable (Section 2.3). Finally, the selected methods are summarized and linked to the experimental configuration used for comparative evaluation (Section 3.4).

Before applying project-specific selection criteria, it is essential to clarify the fundamental technical prerequisites for each methodological category. These prerequisites represent the minimum conditions under which a method can function as intended, independent of project constraints.

Prerequisites per Method Group:

- **Marker-based:** Physical markers, known marker geometry (size & pattern), visible markers, adequate lighting, detectable pattern for ID recognition [10].
- **Feature-based (CAD-free):** Geometric features (contours, normals, curvature), filtered point clouds (noise removal), robustness to occlusion in cluttered scenes [8], [15].
- **Data-driven:** Reference images or annotated training data, GPU-accelerated inference, RGB-D or depth input [8], [13].
- **Hybrid: RGB-D** input with spatial alignment, multimodal compensation for weak modalities, and fusion strategies combining geometric and appearance cues [6], [8].

These prerequisites contextualize the application of project-specific criteria in the following section.

Selection criteria

To ensure a systematic and transparent method selection, candidate approaches were evaluated against criteria informed by literature-identified challenges [6], [8] and project-specific operational constraints. Each criterion is explicitly justified and mapped to the filtering logic applied in the following subsection.

1. **Model Independence & CAD Availability:**
    - **Justification:** Contemporary literature identifies CAD dependency as a practical constraint in unstructured or non-standardized environments [6], [8]. For this project, no precise 3D CAD models are available for the target objects (ketchup bottles).
    - **Filtering Criterion:** Methods requiring CAD files or extensive template libraries are excluded. Priority is given to model-free paradigms and data-efficient approaches that can generalize from limited reference samples.

2. **Computational Requirements & Hardware Compatibility:**
    - **Justification:** Review studies emphasize the trade-off between accuracy and inference latency, noting that many state-of-the-art learning-based methods demand high computational resources [6], [8].
    - **Filtering Criterion:** This project has access to a standard workstation GPU (NVIDIA Quadro P2000, 5 GB VRAM) but lacks high-performance clusters or enterprise-grade GPUs. Selected methods must therefore operate on this standard hardware or Google Cloud GPU resources. Approaches requiring dedicated GPU clusters or extensive training time are excluded due to these hardware constraints.

3. **Data Efficiency & Annotation Overhead:**
    - **Justification:** Manual annotation of 6D pose data is labour-intensive and often prohibitive for low-volume production [8]. Recent advances demonstrate that high performance can be achieved with minimal human-labelled examples via few-shot or synthetic-augmented pipelines [6], [13].
    - **Filtering Criterion:** Methods are favoured if they require $\leq$20 reference images per object or leverage automated pipelines. Approaches demanding thousands of annotated frames per instance are excluded due to timeline constraints.

4. **Environmental Robustness (Clutter & Occlusion):**
    - **Justification:** Bin-picking scenarios are inherently characterised by random object placement and partial visibility. Literature consistently identifies mutual occlusion and cluttered backgrounds as primary failure modes for appearance-only and single-modality methods [6], [8], [13].
    - **Filtering Criterion:** Candidates must demonstrate documented robustness to partial occlusion in multi-object scenarios involving randomly placed items. Methods integrating depth information or employing sensor-fusion architectures are prioritised for handling fragmented visibility and overlapping contours [6].

5. **Sensor Modality Compatibility:**
    - **Justification:** The project utilizes an Intel RealSense D435, providing synchronized RGB and depth streams. Studies document that RGB-D fusion improves robustness and accuracy compared to single-modality pipelines in complex perception tasks [6].

- **Filtering Criterion:** Methods must natively support RGB-D input or be readily adaptable to point cloud + image pairs. Monocular or LiDAR-dependent approaches are excluded.

6. **Implementation Feasibility & Open-Source Availability:**
    - **Filtering Criterion:** Preference is given to methods with actively maintained open-source repositories, clear documentation, and standard dependency stacks (e.g., PyTorch, OpenCV).

7. **Object-Specific Properties (Geometry & Contrast):**
    - **Justification:** Target objects (ketchup bottles) exhibit a "squeeze bottle" geometry (rounded sides, flat front) and mixed texture properties (textureless red body with high-contrast white label). Literature confirms that multimodal strategies leveraging geometric cues alongside appearance-based features effectively address textureless or visually challenging surfaces [6], [8].
    - **Filtering Criterion:** Selected methods must handle non-primitive 3D shapes. Preference is given to approaches leveraging high-contrast labels for positioning while using depth to handle the textureless body.

8. **Robustness to Illumination & Reflections:**
    - **Justification:** Target objects have glossy plastic surfaces creating specular reflections that can saturate RGB sensors and structured IR depth projectors. Literature identifies reflective surfaces as a major challenge and well-documented limitation for industrial optical sensors [6], [8].
    - **Filtering Criterion:** Priority is given to methods utilizing geometric information (point clouds/depth) or designed to be illumination-invariant. Methods requiring strictly controlled lighting are excluded.

These criteria form the filtering framework applied in the following subsection to identify and select methods that satisfy both academic standards and project constraints.

Method identification and selection

Building on the filtering criteria defined in the "Selection criteria" subsection, candidate approaches were evaluated within established taxonomic frameworks for 6-DoF pose estimation. Drawing on these frameworks, contemporary literature commonly groups methods by their dependency on prior geometric knowledge into model-based, model-free, and data-driven paradigms:

- Model-based approaches, requiring explicit 3D object representations,
- Model-free/feature-based approaches, extracting geometric or visual cues without predefined models, and
- Data-driven approaches, leveraging neural networks for direct pose regression.

Complementing this classification, methods are differentiated by input modality (RGB, depth, RGB-D) and algorithmic strategy. Recent reviews also highlight hybrid/fusion-based architectures as a robust solution for handling severe occlusion and textureless surfaces [6], [8].

By cross-referencing the nine filtering criteria with these taxonomies, three methods were identified as optimal for experimental evaluation:

- **ArUco (Analytical, Marker-based):** Selected as a deterministic analytical baseline and ground-truth reference. While marker-based methods are not covered in the data-driven taxonomies presented in [6], [8], ArUco is essential for validation due to its closed-form PnP solution, native OpenCV integration (satisfying criterion #6) [10], computational simplicity consistent with criterion #2
- **FoundationPose (Data-driven / RGB-D):** Embodies the modern few-shot learning paradigm. It addresses criterion #1 (model-free generalization), criterion #3 (requires only ~12–16 images), criterion #5 (native RGB-D input) [13], and criterion #8 (robustness in challenging scenarios via neural implicit representation) [13]. Despite lower data demands, it maintains state-of-the-art robustness on cluttered benchmarks [13].
- **SuperGlue-Enhanced MultiCue (Hybrid, RGB-D):** A custom fusion strategy that combines geometric 3D-PCA pose estimation with deep learning-based orientation classification using SuperPoint and SuperGlue [17], [18]. During development, the initially separate pipelines were merged to resolve the 180° rotational ambiguity inherent in pure geometric PCA. This hybrid approach classifies label orientation (front/back) to apply exact geometric corrections, satisfying criterion #7 (mixed-texture handling) and criterion #8 (robustness in challenging scenarios). ***Note:*** *The RED_BODY detection path was intentionally excluded from the final evaluation due to unreliable poses.*

Exclusion of geometric point-cloud methods

Approaches requiring precise 3D CAD models or extensive template libraries were systematically excluded, as they violate criterion #1 and are infeasible given the project's operational constraints. While literature acknowledges their high accuracy when models are available [6], [8], this study prioritizes paradigms designed for zero-CAD generalization. However, an attempt was made to generate the required 3D reference model using MASt3R [19], a transformer-base reconstruction framework. This proved infeasible, as the framework's memory requirements exceeded the available cloud GPU resources (NVIDIA T4 and L4) when processing the bottle's glossy, textureless surface, confirming that geometric point-cloud methods such as PPF could not be supported within the project's constraints.

Table 1: Selected methods, taxonomic position, satisfied criteria, and operational justification

Method & Taxonomic Position	Primary Satisfied Criteria	Key Operational Justification
ArUco (Analytical, Marker-based)	#2, #6	Deterministic analytical baseline and ground-truth reference; leverages closed-form PnP solution and native OpenCV integration.
FoundationPose (Data-driven, RGB-D)	#1, #3, #5, #8	Modern few-shot learning paradigm achieving state-of-the-art robustness on cluttered benchmarks; utilizes neural implicit representation for challenging scenarios with minimal reference data (~12-16 images).
SuperGlue-Enhanced MultiCue (Hybrid, RGB-D)	#2, #7, #8	Hybrid method merging geometric 3D-PCA with deep learning-based orientation classification. Resolves 180° rotational ambiguity via SuperGlue front/back matching while maintaining computational efficiency. Note: RED_BODY path excluded due to unreliable poses on symmetric surfaces.

Table 1 summarizes the three selected methods, mapping their taxonomic position, primary satisfied criteria, and key operational justifications. These methods form the basis for the experimental configuration and comparative evaluation protocol detailed in Section 3.4.

Infeasibility of Model-Based Evaluation Metrics

Standard benchmarks for 6-DoF pose estimation, such as ADD and ADD-S, require an accurate 3D mesh or point cloud representation of the target object. In this study, a proxy mesh was successfully generated using BundleSDF, which reconstructed a Neural Radiance Field (NeRF) from 24 reference images and extracted a textured 3D mesh (model.obj). However, due to time constraints, it was not possible to conduct a rigorous quality validation of this reconstructed mesh against physical ground truth measurements to determine whether it meets the accuracy requirements for ADD/ADD-S evaluation. The mesh was generated from limited viewpoints (24 images) and exhibited artifacts from depth data holes and texture mapping warnings, introducing reconstruction uncertainty that cannot be separated from pose estimation errors in an ADD/ADD-S calculation. Therefore, model-based metrics (ADD/ADD-S) were excluded from this study. Instead, the evaluation relies on pose error metrics computed directly against ArUco marker-based ground truth (Translation Error, Rotation Error) and detection-based metrics (Detection Frequency, and Precision/Recall evaluated at the 5°5cm and 10°10cm criteria), as described in Section 3.7.

Experimental configuration and comparative evaluation protocol

To enable a fair comparison, an evaluation framework has been established using established metrics and testing principles from the literature [6], [8], [13], [15]. This configuration ensures that the three selected methods are assessed under equivalent conditions, while considering the project's specific constraints. The protocol is structured around four main components: sensor configuration, computational environment, performance metrics, and dataset and test scenarios.

- **Hardware and Sensor Configuration:** The project utilizes the Intel RealSense D435 RGB-D camera for data acquisition. The choice is motivated by the sensor's ability to deliver synchronized RGB and depth streams, which is a prerequisite for evaluating multimodal methods in accordance with Criterion #5 (Sensor Modality Compatibility) [6]. Considering the project's budget constraints, a used unit was acquired. The D435 is described in the literature as a cost-effective sensor well-suited for experimental research in mobile robotics, as it offers a balance between resolution and noise characteristics (Gaussian distribution) [9]. The sensor's active stereo architecture with optional IR projection assists depth estimation in low-texture conditions [9], which is critical for handling glossy, textureless surfaces [8]. The camera is mounted in an eye-to-hand configuration (fixed above the workspace), which enables stable ground truth establishment via ArUco markers and provides a wide field of view for detection in cluttered scenarios [6].

- **Performance Metrics:** The evaluation focuses on pose accuracy. Because model-based metrics such as ADD/ADD-S require a 3D reference model that is unavailable in this study, pose accuracy is instead quantified using the error-

and detection-based metrics defined in Section 3.7, computed against the ArUco marker-based ground truth. These metrics account for both translation and rotation in 3D space, which is crucial for robotic grasp planning [6]

- **Dataset and Experimental Protocol:** The evaluation uses a custom dataset consisting of ketchup bottles (squeeze-bottle geometry with textureless red body and high-contrast white label). A custom dataset was chosen to ensure that the evaluation addresses the specific challenges of the project's target objects in accordance with Criteria #7 and #8, enabling a more applied assessment than general benchmark datasets. Three scenarios are tested: (1) Low clutter with 3 randomly dumped bottles for baseline accuracy, (2) Medium clutter with 5 randomly dumped bottles, and (3) High clutter with 8 randomly dumped bottles to simulate Criterion #4 (Environmental Robustness).

Images are captured for each scenario until a sufficient data volume for statis-tical comparison has been achieved. The sample size is applied consistently across all methods and scenarios to ensure fair comparison within the project's scope.

Ground truth is established using the ArUco marker baseline (Method #1). This constitutes an approximated measurement technique that limits absolute accuracy claims but maintains validity for comparative evaluation within this project's scope.

- **Clutter Simulation & Comparative Protocol:** To evaluate robustness against clutter under reproducible conditions, realistic scenarios are created by randomly dumping 8 ketchup bottles into a standardized container. Method performance is compared relatively based on precision, recall score, and pose error per scenario, enabling a fair assessment that aligns with the project's operational constraints. Ground truth methodology and occlusion-related validation logic are detailed in Section 3.5 and Section 3.6.

3.2 Hardware, Software & Experimental Environment

Hardware specifications

The development and evaluation environment consists of:

- **Desktop workstation:** Intel Core i7-8700K (6 physical cores, 12 logical threads) @ 3.70 GHz, 32 GB RAM, NVIDIA Quadro P2000 (5 GB VRAM, CUDA-capable), running Windows 11 (64-bit).
- **Camera:** Intel RealSense D435 (active stereo). The D435's global shutter depth sensors minimize motion artifacts during data capture [20].
- **Turntable:** Manual rotation stage (360°) used for multi-view capture of the isolated object, both for acquiring the reference images for FoundationPose and for the attempted MASt3R 3D reconstruction.

Software environment

The local Python environment is configured with the following versions::

- **Python:** 3.13.7
- **PyTorch:** 2.8.0 (CUDA-enabled for local SuperGlue inference and remote FoundationPose inference)

- **OpenCV:** 4.12.0.88 (incl. opencv-contrib-python)
- **Other dependencies:** NumPy 2.2.6, Matplotlib 3.10.6, Pandas 2.3.3
- **Package manager:** pip 26.0.1

Computational strategy

The computational load is distributed based on hardware requirements as follows:

- **Phase 1 (NeRF reconstruction via BundleSDF**): Due to the high VRAM requirements of NeRF training, this phase is executed on a Google Cloud virtual machine equipped with a Tesla T4 GPU (15 GB VRAM).
- **Phase 2 (FoundationPose inference):** Executed remotely on the same Google Cloud virtual machine (Tesla T4, 15 GB VRAM) via SSH/SCP. The local pipeline prepares the scene data (RGB-D images, camera intrinsics, and HSV-based instance mask), uploads it to the VM via SCP, triggers inference remotely using xvfb-run, and downloads the resulting pose estimate. This architecture was adopted because the Quadro P2000's 5 GB VRAM is insufficient for FoundationPose's inference requirements, even with the small, reconstructed mesh.
- **ArUco and SuperGlue-Enhanced MultiCue:** Executed locally within a WSL2 (Windows Subsystem for Linux) environment on the desktop workstation. The ArUco pipeline and geometric detection components of SuperGlue-Enhanced MultiCue run on the CPU (Intel Core i7-8700K), while the SuperGlue keypoint matching component utilizes the local GPU (NVIDIA Quadro P2000, 5 GB VRAM). The Intel RealSense D435 camera is connected to the Windows host and accessed from WSL2 via interop (python.exe).

Calibration and data acquisition

- **Intrinsic Calibration:** Factory-calibrated intrinsics retrieved from the Intel RealSense D435 firmware via the RealSense SDK were used for all experiments.
- **Extrinsic calibration (eye-to-hand):** established via ArUco marker detection + PnP in the static lab setup, and via standard hand-eye (AX=XB) solvers during robot integration. In both cases, the resulting transformation matrix $T_{cam_to_base}$ links the camera frame to the world/robot base frame.
- **Dataset:** Real-time data captured from the Intel RealSense D435.

The 50 mm stereo baseline, retrieved from firmware, is fundamental to the depth triangulation process.

3.3 Implementation & Validation Workflow

Vision-to-robot integration is validated through a three-tiered approach, with ABB Robot Studio serving as the primary validation platform.

1. **Primary track – Process-flow verification in ABB Robot Studio:** A virtual robotic cell is modelled to verify that pose estimates from the evaluated vision methods can be successfully transformed to the robot's coordinate frame. The minimum success criterion is that pose data (6-DoF coordinates + uncertainty

metrics) can be transferred from the vision pipeline to the robot controller without logical errors. The target outcome is a conceptual animation demonstrating the intended process flow (pick → open → empty → rinse → place) based on vision-generated poses, illustrating that system integration is feasible. Simplified CAD models are used, as geometric fidelity in simulation is secondary to data-flow validation.

2. **Ambition – Hybrid simulation with real sensor data:** Where feasible, the project aims to enhance realism by using actual RGB-D images captured with the Intel RealSense D435 as input to the vision methods, with resulting poses transferred to Robot Studio for execution. This hybrid approach (real perception → virtual manipulation) is pursued if time and technical constraints permit but is not required for methodological validity.
3. **Conditional extension – Physical robot validation:** If compatible hardware is available, hand-eye calibration succeeds, and resource constraints (time, geographic distance, budget) permit, the same pose-transfer logic is tested on a physical robot. This extension contextualizes simulation findings but is not required for comparative evaluation of vision methods or gripper-vision compatibility analysis.

Success across all tracks is defined consistently: the ability to transform vision output into robot-executable coordinates and demonstrate logical process flow. Partial success (e.g., data transfer without full animation) still yields valid insights into the design of the vision-to-robot interface.

3.4 Experimental Setup & Scenarios

To evaluate the robustness and performance of the methods under varying industrial conditions, three distinct scenarios are defined. These are designed to progressively increase complexity regarding object density and occlusion, within the physical constraints of the experimental setup.

Clutter scenarios

The experiments are conducted in a standardized plastic bin measuring 400 × 300 × 117 mm. A total of eight ketchup bottles is available, setting the upper limit for object density. The scenarios are defined as follows:

- **Scenario 1 (Low clutter):** 3 bottles placed with minimal overlap and varied orientations well-spaced and exhibiting minimal occlusions[8]. Three bottles are chosen for this scenario even though the bin can accommodate three bottles without any occlusion, as four bottles would guarantee occlusion, and depending on how the bottles are placed the occlusion would vary between minimal and severe due to the geometry of the bottle. This scenario establishes a baseline for method performance under low-clutter conditions while introducing moderate perceptual complexity compared to a single-object scenario.

Figure 1: Typical outcomes of the low-clutter scenario (3 bottles). (Left) A less common configuration where bottles settle at varied angles, introducing minimal occlusion. (Right) The most frequently observed configuration, where all three bottles have their labels facing the camera, resulting in zero mutual occlusion. This predominance of non-occluded arrangements justifies the selection of 3 bottles as the baseline "low-clutter" condition, aligning with Cordeiro et al.'s [8] definition of low-level clutter as characterised by well-spaced objects with minimal occlusion.

- **Scenario 2 (Medium clutter):** 5 bottles placed with varying orientations and partial overlap. Guarantees moderate occlusion and tests robustness against limited visual information and increased object density.
- **Scenario 3 (High clutter):** All eight bottles tightly packed with random orientations and heavy mutual occlusion. This amount is chosen as more bottles would mostly end up outside the bin. Simulates realistic bulk scenarios where only parts of the objects' surfaces are visible. This scenario represents the highest clutter level evaluated in this study, with all eight bottles placed in the bin to create substantial mutual occlusion and limited visible surface area.

Table 2: scenario definitions

Scenario	Number of objects	Occlusion level	Evaluation purpose
1 (Low clutter)	3	Light	Baseline performance under low clutter
2 (Medium clutter)	5	Medium	Partial occlusion & orientation variation
3 (High clutter)	8	Heavy	Robustness at maximum density & limited visibility

Figure 2: Visual representation of the three experimental clutter scenarios. (Left) Low clutter scenario with 3 bottles, exhibiting minimal occlusion and well-spaced objects. (Centre) Medium clutter scenario with 5 bottles, showing partial overlap and moderate object density. (Right) High clutter scenario with 8 bottles (maximum capacity), demonstrating severe mutual occlusion and dense packing. These scenarios operationalize the qualitative clutter taxonomy proposed by Cordeiro et al. [8] and enable systematic evaluation of method robustness across graded levels of environ-

Rationale: The number of objects and occlusion levels are chosen to gradually increase perceptual complexity within the physical constraints of the experimental setup (bin: 400×300×117 mm, max eight objects). The setup operationalizes the qualitative clutter taxonomy proposed in recent perception surveys [8], evaluating performance across graded levels of object density and mutual occlusion.

Figure 3: Overview of the experimental setup used for the bin-picking evaluation. The setup features the Intel RealSense D435 camera mounted on a static wooden rig in an eye-to-hand configuration, overlooking the standardized target bin. The image captures the bin placed in the fixture of the wooden bin. This illustrates the resource-constrained, non-laboratory environment utilized to validate the system's feasibility for SME applications.

The black roller blind was installed directly behind and on top of the wooden rig. This physical mitigation was initially deployed to prevent the HSV-based red segmentation pipeline from erroneously triggering on reddish wood tones, structural shadows, or textures of the rig itself. Subsequently, when Region of Interest (ROI) masking was implemented in the software pipeline to isolate the bin, the black roller was left redundant.

Physical protocol for scenario generation

To ensure randomness and avoid placement bias, a standardized physical protocol was followed for each experimental trial:
Pouring: The bottles are poured from a container into the box in a random manner.
Shaking: The box is shaken gently to allow the bottles to settle naturally and interact with one another, creating realistic and unpredictable orientations.
Adjustment: Any bottles that have ended up outside the camera's field of view (ROI) are gently pushed inward so that they fall within the defined workspace.
Static scene: The scene is then left completely undisturbed throughout the entire image capture sequence. This is a strict requirement for the two-image strategy described in Section 4.2 to function, and for the ArUco ground truth and pose estimation to be compared accurately.

Test object properties

The test object is a standard plastic ketchup bottle (Eldorado Tomat Ketchup, 1 kg).

- **Dimensions:** Height 23.5 cm, width ~9.6 cm, depth ~6.4 cm.
- **Material & surface:** Glossy plastic with a largely textureless red body and white labels. The low texture variation and reflective properties pose a challenge for feature-based methods.
- **Symmetry properties:** The object lacks rotational symmetry around its longitudinal axis due to its flattened cross-section (flat front/back surfaces with rounded sides) and visually distinct label designs on the front and back. Although the white screw cap is circular, its diameter is smaller than the bottle body, leaving the asymmetric shoulder geometry visible from top-down views. Approximate left-right symmetry may occur across the sagittal plane when considering geometric structure alone, but asymmetric label content generally enables unambiguous pose determination. Pose accuracy is instead quantified using translation error and rotation error against ArUco marker-based ground truth, along with detection-based metrics (see Section 3.7).

- **Rigid Body Assumption:** Despite the "squeeze-bottle" classification referring to the container's physical properties, the object is modelled as a rigid body for the purpose of 6-DoF pose estimation. The perception algorithms assume a fixed geometric structure to calculate a stable coordinate frame; physical deformation is neglected in the pose calculation logic, consistent with the project scope focusing on rigid packaging.

Figure 4: ArUco marker placement

Surface analysis reveals mixed texture properties consistent with Cordeiro et al.'s classification[8]. The red plastic body lacks high-frequency gradients and is therefore textureless, necessitating geometry- or depth-based estimation for these regions. In contrast, the white label fields containing black and red text provide high-contrast edges and localized texture, enabling template-driven methods. Detection-based metrics are additionally reported to assess robustness under partial occlusion, where temporary loss of asymmetric cues may introduce local pose ambiguity.

Lighting conditions

All experiments were conducted under reproducible lighting conditions to isolate algorithmic performance from external sources of variation.

- **Configuration:** Lighting was held constant throughout data collection using two fixed incandescent lamps with diffuse scattering via ceiling and wall reflections. Variable light sources (e.g., daylight) were excluded, and camera exposure, gain, and white balance were manually locked via the Intel RealSense SDK. While exact spectral or intensity characterization was not performed, the constant illumination ensures that comparative rankings between methods are not confounded by lighting variations. Maintaining controlled illumination conditions and fixed camera parameters is emphasized in perception literature as critical for isolating algorithmic performance from environmental variability [8].
- **Object-related rationale:** The test object's glossy, textureless surface is sensitive to specular reflections. The diffuse, indirect overhead lighting minimizes high-frequency light variations that could disrupt feature extraction and degrade depth data quality from structured light [8].
- **Industrial context & comparability:** The choice of controlled lighting aligns with literature recommendations for mitigating illumination-related artifacts in industrial perception tasks [8]. By keeping lighting conditions constant, illumination is eliminated as a confounding variable, enabling an evaluation of how each method handles the object's inherent challenges (symmetry, texture absence, occlusion).

Dynamic sequential experiment

While Scenarios 1–3 evaluate static states of clutter to isolate algorithmic accuracy, a second, distinct experiment was designed to evaluate the system during a continuous, sequential picking workflow.

Design and Protocol: The dynamic experiment consists of 10 independent sequences. Each sequence begins with a high-clutter state (8 bottles). Following pose estimation, the target bottle is physically removed by the operator to emulate a robotic pick, and the scene is immediately re-evaluated. This loop continues until the vision system detects no remaining bottles.

Rationale: This design shifts the focus from pure pose accuracy in isolated frames to system-level robustness under continuous operational conditions. Unlike the static scenarios, this protocol tests how the system performs as the scene progressively changes, capturing the cumulative effects of sequential picking, evolving occlusion patterns, and the dynamic availability of ArUco ground-truth markers. This enables the validation of method stability and detection consistency across a complete bin-clearing cycle.

3.5 Experimental Data & Methods

To evaluate the performance of various machine vision methods in cluttered environments, an experimental framework has been established. This section describes the data collection protocol, the establishment of a reference system (ground truth), data management, and the two specific methods evaluated.

Data collection protocol

Data is collected using an Intel RealSense D435 (RGB-D, structured light). To balance the need for flexibility during calibration with the need for stability during experiments, a two-step mounting strategy is employed:

1. **Calibration & Reference Collection (Flexible mounting):** For the collection of ~12–16 reference images for FoundationPose the camera is mounted on an adjustable wooden batten clamped to a workbench. This facilitates data collection from a broad spectrum of angles and distances. The camera's focus lock is engaged immediately after collection to ensure that the intrinsic parameters remain constant.
2. **Experimental Evaluation (Static mounting):** For the actual tests of scenarios 1, 2, and 3, the camera is moved to a static wooden rig. The camera is oriented perpendicular to the workspace (top-down). This static setup eliminates vibrations and ensures that the world coordinate system (established via ArUco markers on the bin) remains stable over time.

    - **Resolution & frame rate:** RGB: 1280×720 @ 6 fps, Depth: 1280×720 @ 6 fps. These values were chosen as a balance between image quality and computational cost.
    - **Camera settings:** To eliminate variations caused by automatic exposure, the camera's exposure, gain, and white balance are manually locked via the RealSense SDK before each experimental sequence.

- **Scenarios:** The tests are performed in three configurations: (1) Low clutter (3 bottles), (2) Medium clutter (5 bottles), and (3) High clutter (8 bottles).
- **Sampling:** For each scenario, 10 unique images are collected. Frames with extreme motion blur or where the camera is completely obstructed are manually excluded prior to analysis.

The 87° Horizontal Field of View (FOV) at 1280×720 resolution informed the ~55 cm mounting height to ensure full bin coverage.

Ground truth establishment

To evaluate the accuracy of the methods, a fixed reference system is required. This is handled differently depending on the experimental environment:

- **In a laboratory environment (without a robot):** A world coordinate system is established using an ArUco marker attached to the outer edges of the bin. By detecting this marker and applying a PnP solution with known intrinsic camera parameters, the camera's position and orientation relative to the bin are calculated [10]. This transformation serves as a common reference frame to convert all methods' pose estimations into a unified coordinate system. ArUco serves as a practical reference for comparative evaluation rather than an absolute ground truth, consistent with the approximated measurement techniques defined in Section 1.3.
- **During physical validation (with a robot):** The reference system is established via standard hand-eye calibration (AX=XB). After preliminary TCP calibration (which defines the tool frame relative to the robot flange), multiple robot poses and corresponding camera observations of a fixed calibration target are collected. The transformation matrix Tcam_to_base is then computed using an established solver (e.g., a vendor-specific routine or OpenCV's calibrateHandEye). This links the camera's coordinate frame directly to the robot base, enabling pose transfer without relying on external workspace markers.

It is important to note that the ArUco marker on the bin serves solely as a reference system and not as a perception method for the bottles. For Method 1 (ArUco), separate markers are placed on the bottles for the actual detection.

The evaluation is designed around a sequential picking paradigm rather than static full-scene analysis. In industrial bin-picking workflows, only the topmost or sufficiently visible object is relevant for manipulation at any given cycle. Consequently, ground truth is established per accessible object and pose accuracy metrics (translation error and rotation error) are computed only for instances where the marker is detectable. This reflects the practical workflow logic where the bin is progressively emptied; exclusion of deeply occluded objects from accuracy evaluation is not a methodological shortcoming, but a direct consequence of these objects not yet being manipulable targets. This approach ensures that the evaluation directly validates each method's ability to solve the actual task: sequential identification and localization of pickable objects.

Data management and reproducibility

To ensure reproducibility, all collected data is structured according to a unified file system where each scenario is assigned its own main folder.

- **File format:** RGB images are saved as lossless PNG files. Depth maps are exported as 16-bit PNGs to preserve metric accuracy (mm scale).
- **Intrinsic parameters:** The camera's calibration data (focal length, principal point, distortion coefficients) is stored in a JSON document (camera_intrinsics.json) for easy access by scripts, dynamically loaded to ensure consistency across all methods at 1280×720 resolution.
- **Result logs:** Calculated poses and detection confidences are logged as structured JSON per image.
- **Versioning:** All scripts for data collection and evaluation are version-controlled, with a separate file documenting exact dependencies and configuration parameters.

Evaluated methods overview

The following two vision methods have been selected for comparative pose estimation evaluation, alongside the ArUco baseline used for ground truth establishment. A common selection logic (Unified Scorer) ranks all detected instances using a weighted aggregate score. This score is derived from six normalized factors: detection area, number of agreeing methods, match distance to ground truth, Z-height, detection type, and validation strictness, preceded by hard filters (e.g., maximum distance to ArUco ground truth) and 3D clustering (30 mm). To ensure that Methods 2 and 3 evaluate the object geometry and texture rather than the fiducial markers used for ground truth, an automatic masking step is applied. The bounding regions of detected ArUco markers are zeroed out in RGB and Depth streams prior to processing by Methods 2 and 3. The detailed algorithmic implementation of these methods is presented in Section 4.4.

Method 1: ArUco (Marker-based)

Serves as the deterministic analytical baseline and ground-truth reference. It leverages a closed-form PnP solution and native OpenCV integration, providing high detection rates and low computational overhead. Each marker is assigned a unique ID to unambiguously determine the visible side of the object, aiding in orientation-aware manipulation without additional computational cost. As it serves as the ground truth reference for the other methods, it is excluded from the pose accuracy evaluation, and its performance is instead quantified via detection frequency.

Method 2: FoundationPose (Data-driven / Few-shot AI)

Represents a modern, data-driven method that generalizes to novel objects using a small number of reference images (~16 images) without requiring a CAD model. It employs an object-centric neural implicit representation for efficient novel view synthesis and refines the 6-DoF pose using a render-and-compare mechanism. Its main strength is robustness under high clutter and textureless surfaces, though its accuracy is inherently coupled to the geometric quality of the reconstructed proxy mesh used during inference.

Method 3: SuperGlue-Enhanced MultiCue (Hybrid, RGB-D)

A custom hybrid method designed to resolve the 180° rotational ambiguity inherent in pure geometric PCA. It first uses a custom geometric detection pipeline (colour segmentation and 3D-PCA) to identify candidate objects. To resolve the front/back ambiguity, it utilizes SuperPoint and SuperGlue to match key points against reference images, classifying the visible label orientation. This classification is passed to the geometric pose estimator to apply exact geometric corrections. (***Note:*** *The purely geometric RED_BODY detection path was intentionally excluded from the final evaluation due to unreliable poses under specular reflections, as detailed in Section 4.6*).

3.6 Preprocessing & Validation

To ensure that the input data to the evaluated methods is of high quality and that the results are comparable, a unified preprocessing and validation protocol is applied. This section describes the steps applied to point clouds and RGB images, as well as the criteria used to validate or exclude individual frames from analysis.

Two-Photo Strategy

Unlike traditional pipelines that rely on point cloud filtering or spatial resizing, the evaluated methods operate directly on native-resolution RGB-D data to preserve maximum geometric fidelity. To simultaneously satisfy the conflicting resolution requirements of the ground-truth establishment and the vision methods, a Two-Photo Strategy is employed using the Intel RealSense D435:

- Photo 1 (1920×1080 RGB): Captured exclusively for ArUco marker detection. The high resolution ensures that the smallest fiducial markers maintain a sufficient pixel footprint for robust sub-pixel corner refinement and PnP solving.
- Photo 2 (1280×720 RGB-D): Captured for the vision methods. This resolution matches the native maximum depth resolution of the D435, avoiding interpolation artifacts that would otherwise corrupt 3D back-projection and PCA calculations.

To guarantee identical lighting conditions between the two photos, a warmup of 30 frames is performed to allow the sensor's auto-exposure algorithm to stabilize. Following warmup, fixed camera settings (exposure: 312, gain: 0, white balance: 4000 K) are applied by disabling automatic mode. These hardcoded values were determined during pilot testing to provide optimal image quality for the specific lighting conditions (dark environment with curtain) and are explicitly re-applied during the capture of Photo 2 to ensure consistency across all experimental cycles.

Validation Criteria

To ensure that only frames of sufficient quality were included in the final evaluation, the following exclusion criteria were applied:

- **Visibility threshold:** An empirically determined visibility threshold was applied to exclude extremely occluded objects. The threshold was determined through a pilot study, and frames where the object's estimated visible volume fell below this value were excluded from the main evaluation. The final value was documented prior to the main runs and chosen to balance dataset size against the

requirement to only include practically graspable objects. Pose estimation under extreme occlusion was thus considered a separate challenge lying outside the primary focus of this study.

- **Visual inspection:** As a final step, a manual visual inspection was conducted where the estimated coordinate system axes (X, Y, Z) were projected onto the original image to verify that the pose origin and orientation aligned consistently with the object's visible geometry. Frames where the projected axes were clearly misaligned with the object's visible geometry were flagged for manual review and excluded in case of doubt. This step ensured that automatic metrics did not approve systematically incorrect poses that nevertheless met numerical thresholds.

3.7 Evaluation Methods & Metrics

To answer the study's research question and enable a fair, reproducible comparison between the two evaluated vision methods, performance is operationalized through a set of quantitative metrics. These are directly linked to the project's sub-goals and follow established evaluation protocols in 6D pose estimation, with relevance to bin-picking scenarios, including standardized error- and detection-based metrics [6], [21]. Below, the evaluation metrics, matching protocol, and performance indicators to be applied are defined.

Pose Error Metrics

Pose error quantifies the deviation between the computed 6-DoF pose and the ground truth pose. Because model-based metrics such as ADD/ADD-S require a 3D reference model that is unavailable in this study, pose accuracy is instead quantified using error metrics computed directly against the ArUco marker-based ground truth:

- **Translation Error:**
  - **Definition:** The Euclidean distance between the estimated object position and the ground truth position, measured in millimetres.
  - **Application:** Quantifies positional accuracy of the estimated pose.

- **Rotation Error:**
  - **Definition:** The angular difference between the estimated object orientation and the ground truth orientation, measured in degrees. Computed as the angle of the relative rotation matrix:

$$\theta = arccos\left(\frac{trace\left(R_{est}R_{gt}^{T}\right) - 1}{2}\right)$$

  - **Application:** Quantifies orientational accuracy of the estimated pose.
- **Symmetry-Aware Rotation Error:**
  - **Definition:** To account for the inherent 180-degree rotational ambiguity of the Principal Component Analysis (PCA) used in the geometric pose estimation (Method 3), a symmetry-aware rotation error is computed. Because the bottle's geometric body is symmetric along its lateral axes, the PCA eigenvectors for the X and Z axes may flip by 180 degrees without altering the physical graspability. The symmetry-aware error is defined as the minimum of the standard rotation error and the rotation

error calculated after applying a 180-degree rotation around the bottle's longitudinal Y-axis to the estimated pose: $\theta_{sym} = \min(\theta, \theta_{Y-flip})$.
    - **Application:** Prevents penalizing geometrically valid but mathematically flipped poses. This ensures the evaluation reflects the actual physical grasp success rather than strict mathematical orientation, aligning with the symmetry-adapted metrics such as ADD-S
- **5°5cm Criterion:**
    - **Threshold:** Rotation error < 5° ∧ Translation error < 5 cm.
    - **Application:** Binary complementary metric commonly used in industrial applications to define a "sufficiently accurate" pose for robotic manipulation tasks [6].
- **10°10cm Criterion:**
    - **Threshold:** Rotation error < 10° ∧ Translation error < 10 cm.
    - **Application:** A less stringent binary metric for assessing pose usability under higher clutter levels while still enabling successful grasping.
- **Detection Frequency:**
    - **Definition:** The proportion of frames in which the method returns at least one valid pose estimate, regardless of accuracy.
    - **Application:** Quantifies the method's robustness and reliability across all scenarios, independent of pose precision.

Matching Protocol

To handle multiple detections per frame and avoid double-counting, a deterministic matching procedure is applied:

- Nearest-GT matching: Each detection is paired with the ground truth bottle whose position is closest (smallest 3D Euclidean distance).
- Threshold condition: A match is accepted as a True Positive only if the translation and rotation errors fall within the thresholds defined by the 5°5cm or 10°10cm criteria.
- One-to-one mapping: Each ground truth instance is matched at most once. The detection with the lowest translation error claims the object; all others are marked as false positives.

Aggregated Performance Metrics

Aggregated metrics are calculated over the entire test dataset and reported per scenario. The evaluation logic strictly follows the matching protocol where each detection is paired with the nearest ground truth instance. Here, T_p denotes true positives (correctly matched detections), F_p denotes false positives (detections not matching any ground truth), and F_n denotes false negatives (ground truth instances not matched by any detection):

Precision: $P = T_p / (T_p + F_p)$. Measures the proportion of reported detections that are correct.

Recall: $R = T_p / (T_p + F_n)$. Adapting the evaluation methodology from the BOP benchmark [21], the recall score is the fraction of test targets (ground truth instances) for which a correct pose is estimated.

Precision evaluated at 5°5cm and 10°10cm thresholds: The proportion of reported detections that satisfy the strict 5°5cm or the more lenient 10°10cm accuracy criteria.
Detection Frequency: Proportion of frames where the method returns at least one valid pose, independent of accuracy. Reported separately for Scenarios 1–3.

Note on Ground Truth Filtering: All detection-based metrics are computed relative to the subset of objects with reliable ground truth availability. Heavily occluded instances without visible ArUco markers in the current frame are dynamically excluded from the evaluation pool (F_n is only calculated against visible markers). This aligns with the sequential picking paradigm.

3.8 Error Sources & Uncertainty Analysis

To ensure a transparent and scientifically sound evaluation, the primary error sources, uncertainty margins, and methodological limitations that may affect the study's results are identified and quantified below. These aspects are documented to enable correct interpretation of the performance metrics and to strengthen the study's reproducibility.

Systematic Error Sources

- **ArUco Detection:**
  - **Expected Impact:** False negatives under challenging lighting conditions such as low contrast, shadows, or non-uniform illumination . May cause missed detections in Scenarios 2–3.
  - **Mitigation:** Adaptive thresholding parameters (adaptiveThreshWinSizeMin/Max/Step) can be adjusted to adapt detection to varying lighting conditions [OpenCV, 2024]. Detection frequency is reported per scenario to quantify the method's sensitivity.
- **Domain Gap (Synthetic → Real):**
  - **Expected Impact:** FoundationPose is pretrained on synthetic data and may initially be sensitive to real sensor noise and lighting variations.
  - **Mitigation:** The model utilizes a few-shot paradigm and is adapted (trained) on approximately 12-16 real reference images of the ketchup bottles to bridge the domain gap and generalize to the specific sensor [13].
- **Occlusion & Clutter:**
  - **Expected Impact:** Missed detections or incorrect poses when objects are partially covered by other bottles.
  - **Mitigation:** Scenario stratification: results are reported separately for Scenario 1 (low), 2 (medium), and 3 (high clutter). Extremely occluded cases are excluded via a visibility threshold.
- **Manual Marker Application Variance:** A bounded but unmeasured systematic error is introduced during the physical application of the ArUco fiducials to the bottles. Four coupled sources contribute to this placement variance:

- **Manual positioning tolerance:** Markers are aligned by eye using physical references (mold seams, label edges), introducing a bounded translational offset (estimated ≤ ±3 mm) between the marker frame and the intended object frame.
- **Surface non-planarity (marker tilt):** Because the bottle surfaces are locally convex or concave, no marker can be applied perfectly planar. The out-of-plane sagitta at each corner ($h \approx L^2/8R$) violates the coplanarity assumption of the PnP solver, introducing a small systematic rotational bias that is static per instance but varies across instances. Because the out-of-plane deviation of a marker of side length L on a surface of radius Rscales as $L^2/(8R)$, reducing the marker from 20 mm to 15 mm on the most convex region decreases the curvature-induced planarity violation by approximately 44% (from ~1.4 mm to ~0.8 mm at $R \approx 35$mm, i.e., from ~4 px to ~2 px at the 55 cm working distance). The resulting loss in pixel footprint is compensated by the mounting height, which keeps even the smallest marker above ~40 px in the 1920×1080 capture.

Figure 5: Example of ArUco marker placement on the bottle's convex surface. The marker is applied manually to the planar elliptical side surface of the ketchup bottle. The annotated ring highlights the upper-right corner of the marker, which extends beyond the planar region and adheres to the curved shoulder of the bottle. This causes local deformation and perspective distortion of the printed pattern, violating the coplanarity assumption of the PnP solver and introducing a small systematic rotational bias. This visual evidence directly supports the documented placement tolerance and the curvature-induced planarity violation.

- **Inter-instance label offset:** The commercial labels used as alignment references are not positioned identically across the eight bottles (manufacturing tolerance of the label application on the production line). Consequently, the reference lines derived from label edges carry an inter-instance offset that propagates to the marker frame.
- **Adhesion-induced deformation:** During application, the flexible sticker may micro-stretch, wrinkle, or trap air bubbles on the glossy plastic, introducing sub-millimetre distortions of the printed marker pattern.

Impact on Comparative Validity: Critically, these errors are static per object instance and constitute the identical ground-truth baseline for Methods 2 and 3 within each evaluation frame. They therefore act as a documented systematic noise floor for the absolute pose accuracy metrics but do not systematically favour any specific vision method. Consequently, while absolute accuracy is bounded by this manual placement tolerance (precluding claims of sub-millimetre metric accuracy), the validity of the relative comparative analysis between the evaluated methods is fully preserved: any observed performance difference between methods reflects genuine algorithmic properties rather than artifacts of the ground-truth establishment.

Mitigation via Robot-Assisted Calibration (Not Performed): If a robotic manipulator were available, this placement variance could be quantified and compensated on a

per-instance basis by mounting each bottle in a fixed reference fixture and teaching the marker's four corners with a calibrated TCP, or by computing the fixture-to-marker offset transformation via PnP. The resulting per-instance transformation matrix would convert the ArUco-detected pose into the object's geometric frame without reliance on manual measurement, reducing the systematic noise floor to a quantified, compensated offset. In the present study, this mitigation is not performed due to hardware unavailability; the manual placement tolerance is instead documented as a bounded systematic uncertainty.

Uncertainty Margins

To quantify the statistical uncertainty in the reported performance metrics, the following procedure was applied:

Descriptive Statistics: To quantify the statistical variation in the reported performance metrics, descriptive statistics (mean, median, minimum, and maximum) are computed for all primary metrics (Precision, Recall score, Precision at 5°5cm, Precision at 10°10cm, and translation/rotation error values). This enables an assessment of method robustness and empirical variation across scenes and clutter levels without relying on resampling techniques.

Methodological Limitations

The following limitations are inherent to the study's design and must be considered when interpreting the results:

- **Absolute Values Not Directly Comparable to Literature:** Due to the unique experimental setup (specific bottle, D435 sensor, turntable, lighting, distance), absolute performance figures cannot be directly compared to external benchmark studies. Only relative comparisons between the evaluated methods under identical conditions are methodologically valid.
- **ArUco Occlusion Bias:** Since Method 1 requires a clear line of sight to the marker, it automatically excludes cases where the marker is covered. This introduces a selection bias toward objects with detectable markers, which is consistent with the sequential picking paradigm defined in Section 3.5. In this workflow, the set of manipulable targets at any cycle consists of objects with visible markers; objects without detectable markers are not yet accessible for verification or manipulation. Consequently, the validation baseline is limited to this verifiable subset rather than the full static scene.
- **CPU vs. GPU Comparability:** ArUco runs exclusively on the workstation's CPU within WSL2, while SuperGlue-Enhanced MultiCue utilizes both CPU and GPU (Quadro P2000) for the SuperGlue matching component. FoundationPose requires GPU acceleration and is therefore executed remotely on a Google Cloud VM (Tesla T4).
- **Factory Calibration Not Validated:** The camera's intrinsic parameters were used as provided by the manufacturer. To ensure accuracy, an independent validation was conducted via a reference-distance test. The test confirmed that the factory calibration was correct and that any systematic error was negligible. This validates the geometric mapping and depth data used across all evaluated methods.

Data Utilization for Downstream Analysis

The experimental results—pose uncertainty distributions, detection frequency, and coordinate stability metrics—are documented to support vision method selection. A supplementary specification-based review identifies gripper types mechanically compatible with the target object based on documented mechanical principles and class-level characteristics. No experimental validation of grasp success or kinematic tolerance mapping is included within the project scope.

3.9 Reproducibility Protocol

To ensure that the experimental pipeline can be independently verified and replicated, all software dependencies, algorithmic parameters, and stochastic elements have been explicitly documented and locked prior to the main evaluation phase.

- **Version Locking & Dependency Management:** All Python-based methods are governed by a strictly version-controlled dependency file (requirements.txt). This file specifies exact package versions (e.g., opencv-contrib-python==4.12.0.88, numpy==2.2.6, torch==2.8.0) to prevent dependency drift and ensure consistent behaviour across execution environments. The file is included in the supplementary repository alongside a README.md with step-by-step environment setup instructions.
- **Deterministic Execution:** The evaluated methods are largely deterministic by design. ArUco marker detection uses OpenCV's deterministic PnP solver with fixed adaptive thresholding parameters. The SuperGlue-Enhanced MultiCue pipeline uses deterministic geometric operations (3D-PCA via SVD, Canny edge detection, Hough transforms) combined with SuperGlue's graph neural network matching, which operates in inference mode without stochastic sampling. FoundationPose inference is executed in deterministic evaluation mode on the remote GPU. Each scenario is executed with a fixed number of cycles (10 per clutter level), and the experimental protocol ensures identical input data across all methods to enable fair comparison.
- **Parameter Locking:** All critical hyperparameters and preprocessing thresholds were determined during pilot studies and frozen before the main data collection phase. No post-hoc tuning was performed based on test-set results. Locked parameters include: Canny edge detection thresholds, ArUco detection parameters (minMarkerPerimeterRate, adaptiveThreshWinSizeMax), SuperGlue match thresholds, and the fixed 28.5 mm surface-to-centre offset. The complete parameter set is documented in Appendix 1.A
- **Hardware & Computational Context:** The ArUco and SuperGlue-Enhanced MultiCue methods are evaluated locally within a WSL2 (Windows Subsystem for Linux) environment on the desktop workstation (Intel Core i7-8700K). FoundationPose is split into two phases, both executed on Google Cloud: Phase 1 (NeRF reconstruction via BundleSDF) and Phase 2 (pose inference) run on a Google Cloud virtual machine equipped with a Tesla T4 GPU (15 GB VRAM). The local pipeline orchestrates the remote execution via SSH/SCP, preparing scene data locally and downloading results after inference. The hardware specifications for both environments are documented in Section 3.2.
- **Data & Code Availability:** The experimental code, evaluation scripts, and a minimal subset of representative frames are provided in a version-controlled

repository. Full dataset availability is restricted by storage constraints, but the data collection protocol (Sec. 3.5) and preprocessing pipeline (Sec. 3.6) are fully specified to enable independent reconstruction of the evaluation set.

- **Replication Scope & Limitations:** While absolute hardware equivalence cannot be guaranteed, the combination of version-locked dependencies, fixed random seeds, and documented parameter sets ensures that relative performance rankings and methodological conclusions are robustly reproducible. Researchers seeking to replicate the full pipeline are advised to match the CPU/GPU context per method and to account for minor sensor-specific variations when reproducing depth-based metrics.

3.10 Specification-Based Gripper Evaluation Methodology

While the primary focus of this thesis is the evaluation of vision methods for 6-DoF pose estimation, the overarching goal of the RE:WORK framework is the physical recovery of food products. Consequently, the perception pipeline must be compatible with a viable manipulation strategy. To address Sub-goal 4, a systematic, specification-based evaluation methodology was developed to identify industrial gripper paradigms mechanically suited to the target object (a 1 kg plastic ketchup bottle of the "squeeze-bottle" type). The purpose is not to select a specific commercial model, but rather to determine, at a principled level, which type of gripper is most suitable for the task and to ultimately recommend one or two paradigms. Consistent with the limitations defined in Section 1.3, this evaluation relies solely on the theoretical matching of documented mechanical principles and paradigm-level characteristics against the object's physical properties, excluding physical grasp testing and dynamic tolerance simulations.

Evaluation Criteria

To ensure a structured assessment, the candidate grippers were evaluated against five core criteria derived from the object's geometry (Section 3.4), the environmental constraints of the bin-picking scenario (Section 3.4), and the tolerance analysis framework and system architecture presented in Section 2.4:

1. **C1: Geometric Envelope and Stroke:** The gripper stroke must be dimensioned to cover the object's full grasp span and to support the two grasping strategies required by the scenario. When the bottle lies down, the grasp is applied at the designated grasping location, where the width is approximately 87 mm and the depth approximately 56 mm (approximately 50 mm in a closed clamping grasp), either from the side or against the front/back surfaces. To safely pass around and encompass the bottle's maximum width (approximately 96 mm), a maximum opening of at least approximately 106 mm (96 mm + 10 mm tolerance) is further required, yielding a required stroke span of approximately 50–106 mm, which also accommodates the cap diameter (approximately 55 mm) in accordance with Criterion 5. When the bottle stands upright or upside down in the bin, the gripper must instead be capable of performing a top-down grasp around the exposed portion of the bottle body, without grasping the cap. Which strategy is feasible is determined by the object's estimated pose, thereby directly linking the perception stage to the manipulation stage. At the same time, the gripper must retain a compact physical footprint and a finger

geometry that accommodates the non-cylindrical, flattened cross-sectional profile (flat front and back, rounded sides) in order to avoid collisions with the bin walls (400 × 300 mm) and adjacent objects in high-clutter scenarios.

2. **C2: Payload Capacity and Friction Management (Lifting Phase):** The end-effector must safely carry a 1 kg payload under dynamic accelerations in both grasping strategies (side grasp at the grasping location and top-down grasp, respectively). Since the grasp in both configurations may be located at a distance from the centre of gravity, tipping moments arise about the contact points, which the gripper must counteract through sufficient friction and form closure (e.g., compliant, textured, or profiled finger pads) – without requiring excessive clamping forces that risk deforming the bottle walls or otherwise damaging the container. Given the glossy plastic surface, mechanisms that maximise friction are prioritised. Lifting the bottle by the cap alone is considered infeasible: in addition to the unfavourable leverage effects and the risk of fracture at the small load-bearing surface, the cap is of flip-top type with a hinge, meaning that a lifting grasp on the cap risks actuating the opening mechanism (lifting the lid half) and thereby compromising the closure and the integrity of the product.

3. **C3: Vision-Error Tolerance (Mechanical Compliance):** As established in Section 2.4, vision-based pose estimation inherently introduces translational and rotational uncertainties. Grippers featuring mechanical compliance or self-centring kinematics are evaluated favourably for their capacity to absorb minor approach errors and thereby limit the accumulated tolerances during the seizing phase [12].

4. **C4: Accessibility in Clutter:** The kinematic profile of the gripper during the approach and retraction phases is assessed. For lying bottles, free lateral access at the grasping location is required, whereas standing bottles can only be grasped from above, since the bin walls (117 mm in height) block horizontal approach to the bottle body. The fingers must therefore be sufficiently slender and long to be lowered alongside a standing bottle without colliding with the bin walls or adjacent objects. Mechanisms that expand significantly outside the object's bounding box during actuation (e.g., certain angular grippers) are penalised due to the high collision risk in densely packed bins.

5. **C5: Compatibility with Downstream Processes (Fixture and Decapping):** In line with the system architecture discussed in Section 2.4 and the intended process flow (pick → open → empty → rinse → place) described in Section 3.3, the grasped object must ultimately be placed in a stationary fixture. Once the bottle body is securely locked in the fixture, a further critical functional requirement is imposed on the gripper: it must be able to interact with the cap and apply sufficient torque to unscrew it, without deforming or destroying the cap. The cap consists of two cylindrical parts joined by a hinge (flip-top design), which imposes specific requirements on the decapping grasp: the torque must be transmitted to the lower, threaded cylinder – by grasping around both cylinders or only the lower one – and must never be applied to the upper cylinder alone, as this would load the hinge without releasing the cap. This can be performed either from above (grasping around both cylinders) or from the side

(grasping around the lower cylinder, or both cylinders). The gripper's kinematics and finger design must therefore be capable of handling both the bottle body width (approximately 96 mm) and the cap diameter (approximately 55 mm) within the same cycle, which imposes specific requirements on the stroke span and finger geometry (e.g., a profiled jaw form that switches between body grasp and cap grasp).

Candidate Paradigms

Drawing upon the taxonomic classification of industrial grippers outlined in Section 2.4, six primary gripper paradigms were selected for theoretical evaluation:

1. **Rigid Parallel-Jaw Grippers:** Standard two-fingered configurations utilising linear actuation.
2. **Angular Grippers:** Two-fingered configurations operating about a central pivot point.
3. **Vacuum Grippers:** End-effectors relying on pressure differentials (e.g., suction cups or Bernoulli principles).
4. **Multi-Fingered Dexterous Grippers (Non-adaptive):** Rigid, multi-jointed systems requiring independent servo control per digit.
5. **Adaptive Mechanical Grippers (Underactuated):** Multi-fingered systems utilising mechanical linkages to passively conform to the object geometry.
6. **Adaptive Soft Finger Grippers:** End-effectors utilising elastomeric materials or pneumatically driven soft actuators.

Evaluation Protocol

Because manufacturer datasheets describe individual commercial products rather than abstract paradigms, the assessment was based on documented mechanical principles and, where available, the class-level characteristics reported in the gripper classification literature [11], together with the specification requirements derived from the object's physical properties; no model-level datasheet matching was performed. The characteristic parameter ranges of each paradigm (e.g., achievable stroke, payload capacity, actuation type, and finger material properties) were cross-referenced against the evaluation criteria (C1–C5) defined above. For transparency and comparability, each paradigm was documented along four dimensions: (i) design considerations and kinematic footprint, (ii) finger design and stroke requirements, (iii) environmental and maintenance factors, and (iv) integration and cost factors. The outcome of this methodological process is a comparative matrix detailing the mechanical advantages, inherent failure modes, and integration constraints of each paradigm relative to the specific challenges of the target object, ultimately enabling the recommendation of the one or two paradigms most suited to the task. The detailed results of this evaluation, including the final recommended gripper architecture, are presented and discussed in Chapter 5.

4 Implementation & System Integration

This chapter presents the complete implementation of the bin-picking system, from sensor data acquisition through perception, evaluation, and robotic manipulation. The system was designed to address the research aims stated in Chapter 1 by enabling a fair, reproducible comparison of modern 6-DoF pose estimation methods under realistic industrial occlusion conditions, and by demonstrating end-to-end integration with a robotic digital twin. The implementation follows a modular architecture that separates concerns into distinct pipeline stages, each of which can be independently tested, replaced, or extended.

4.1 Pipeline Architecture and Dual-Mode Execution

The top-level orchestration is implemented in two complementary Python scripts, main_loop.py and main_loop_dryrun.py, which share the same vision and evaluation subroutines but differ in their interaction with the robotic cell.

main_loop.py operates as a TCP server that awaits handshake messages from the RAPID program running in RobotStudio. Upon receiving a READY signal from the robot, it executes a full pick cycle: capturing fresh RGB-D frames, running all vision methods, evaluating detections against ArUco ground truth, ranking candidate bottles through the unified scorer, and finally transmitting the selected pose together with grasp-synthesis parameters back to RAPID. Between cycles the operator is prompted to physically remove the previously picked bottle, ensuring that each new capture represents a static scene.

main_loop_dryrun.py is a functionally identical orchestrator that omits the socket communication and robot interaction. It accepts a --bottles argument purely for logging purposes and auto-increments a cycle counter stored in output/evaluation_results/. This dual-mode design was motivated by three practical considerations:

1. **Isolated algorithm testing:** Vision methods can be evaluated without the overhead of establishing a socket connection or waiting for the robot to complete its motion sequence, dramatically shortening the debug-edit-run cycle.
2. **Fail-fast error reporting:** The dry-run script aborts immediately if any vision method returns a non-zero exit code, preventing wasted GPU time on downstream evaluation when an upstream stage has already failed.
3. **Reproducibility:** Each cycle writes a structured cycle_XXX_metrics.json file containing per-detection translation error, rotation error, and binary success flags, enabling post-hoc statistical analysis across all 30 experimental cycles.

Vision methods are executed sequentially rather than in parallel, despite the availability of a ThreadPoolExecutor. This decision was made after empirical observation that concurrent execution of SuperGlue and FoundationPose caused GPU memory fragmentation on the Quadro P2000 (5 GB VRAM), resulting in non-deterministic out-

of-memory crashes. Sequential execution also yields fairer wall-clock timings per method, which is relevant for the computational-cost analysis in Chapter 5.

Sequential Execution Pipeline

To facilitate the dynamic sequential experiment described in Section 3.4, the pipeline was extended with a sequential execution mode. This mode automates the picking loop: it triggers the capture, runs the vision methods, evaluates the detections, and prompts the operator to physically remove the target bottle before repeating the cycle. It continuously tracks the number of remaining bottles via the ArUco ground truth and logs the per-cycle metrics (translation error, rotation error, execution time, and grasp parameters) into a structured CSV file for post-hoc statistical analysis.

4.2 Hardware Configuration and Two-Photo Strategy

RGB-D data is acquired from an Intel RealSense D435 depth camera mounted 56.5 cm above the bottom of the bin. At this mounting distance, all bottles lie within the sensor's valid measurement range (Minimum-Z: 280 mm at 1280×720).

To handle the conflicting resolution requirements of the ground-truth establishment and the vision methods, the capture_d435.py script implements a two-photo strategy using two separate pipeline sessions:

Phase 1: High-resolution RGB for ArUco. A pipeline is configured for 1920×1080 RGB at 6 fps. After a warmup of 30 frames, exposure (312), gain (0), and white balance (4000 K) are locked by disabling automatic mode. A single frame is saved as data/bin_rgb_aruco.png.

Phase 2: Matched RGB-D for pose methods. A new pipeline is started at 1280×720 for both colour and depth streams. The locked settings from Phase 1 are explicitly applied to the colour sensor. The laser power is set to 150 mW. Since both streams have the same resolution, rs.align performs a 1:1 projection with no upsampling. RGB and depth are saved as data/bin_rgb.png (8-bit PNG) and data/bin_depth.png (16-bit PNG, mm scale), respectively. The camera intrinsics are exported to data/camera_intrinsics.json.

Phase 3: Intrinsics export. The intrinsic matrix is persisted as JSON so that all downstream scripts read the same calibrated values.

The scene is kept static throughout the capture sequence to ensure accurate comparison between the high-resolution reference image and the RGB-D streams.

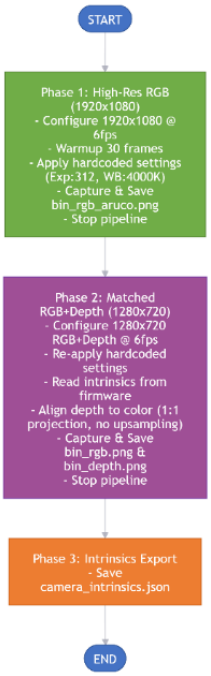


Figure 6: Two-Photo Strategy for RGB-D data acquisition using Intel RealSense D435. Phase 1 captures high-resolution RGB (1920×1080) for ArUco marker detection with locked camera settings (exposure: 312, white balance: 4000K). Phase 2 reconfigures the pipeline to matched RGB+Depth resolution (1280×720) to avoid interpolation artifacts from depth upsampling, re-applying the locked settings to ensure identical lighting conditions. Phase 3 exports camera intrinsics to JSON for consistent use across all downstream scripts. The static scene guarantees both photos represent the same physical configuration.

4.3 ArUco-Based Ground Truth System

Ground truth poses are generated using ArUco fiducial markers (cv2.aruco.DICT_4X4_100). Each bottle carries up to seven markers placed at known positions: back_lower, back_upper, front_upper, front_lower, right_lower, left_lower, and cap. An additional marker (corner) is placed just outside one corner of the bin as a global reference.

The nominal marker sizes range from 10 mm (back_upper) to 40 mm (corner). The actual sizes are 1.25 times the nominal values due to print scaling, yielding a real range of 12.5–50 mm. The calibration file aruco_calibration.json maps each marker ID to its position on the bottle, its actual size in millimetres, and its 3D offset (marker_center_in_bottle_frame) relative to the bottle's geometric centre.

Detection

The script run_aruco_ground_truth.py loads the high-resolution RGB image (bin_rgb_aruco.png, 1920×1080). The red channel is extracted (BGR format) for better contrast against the red plastic surfaces, and CLAHE (clipLimit=3.0, tileGridSize=8×8) is applied to enhance local contrast. The ArUco detector is configured with minMarkerPerimeterRate=0.01, adaptiveThreshWinSizeMax=33, and CORNER_REFINE_SUBPIX.

Pose computation

For each detected marker, the following is performed:

1. A 2D–3D correspondence is built from the marker's actual size: four corners in the marker's local plane (±L/2, ±L/2, 0) , where L is the marker's physical side length in millimetres.
2. cv2.solvePnP solves the perspective problem and returns the rotation vector rvec and translation vector tvec describing the marker's pose in camera coordinates (R_{cm}, t_{cm}) , where R_{cm} is the 3×3 rotation matrix and t_{cm} is the 3×1 translation vector.
3. The marker's orientation relative to the bottle (R_{bm}) is retrieved from a predefined set of rotations based on the marker's position (e.g., 180° around the x-axis for front-side markers, 180° around the z-axis for back-side markers).
4. The bottle's translation in camera coordinates is computed as $t_{cb} = t_{cm} + R_{cm} \cdot w$, where $w = -R_{bm}^T \cdot v$ and v is the marker's known offset from the bottle centre.
5. The bottle's rotation is computed as $R_{cb} = R_{cm} \cdot R_{bm}^T$, where R_{cb} is the bottle's rotation matrix in camera coordinates.

Per-bottle aggregation

Marker IDs are mapped to bottle IDs via position-based offset tables (e.g., back_lower: ID 0–7 → bottle 0–7, front_upper: ID 24–31 → bottle 0–7). When multiple markers on the same bottle are visible, the bottle's final position is computed as the mean of all marker translations. The rotation is taken from the marker whose translation lies closest to the mean.

Camera intrinsics

The intrinsics are read from camera_intrinsics.json (generated at 1280×720 in Phase 2) and scaled to the 1920×1080 resolution by the factor $s = 1920/w_{720}$. This ensures that the PnP solution uses the correct focal lengths and principal point for the high-resolution image.

The result is saved as data/aruco_ground_truth.json with position_mm and euler_deg per bottle. Bottles without visible markers are excluded.

ArUco masking

Before the vision methods run, mask_all.py detects all ArUco markers in both the 1080p and 720p images, expands each marker polygon by a calibrated margin (in millimetres, converted to pixels via the local marker scale), and inpaints the masked regions using cv2.inpaint (Telea method) with a cosine-weighted feather edge. This prevents the vision methods from exploiting the high-contrast fiducial patterns as orientation cues.

4.4 Vision Method Implementation

Two vision methods are implemented and evaluated: SuperGlue-Enhanced Multi-Cue (a hybrid method combining geometric analysis with deep learning-based classification) and FoundationPose (a mesh-based method executed on external GPU hardware).

SuperGlue-Enhanced MultiCue

This method is implemented in run_superglue_enhanced_multicue.py and combines geometric label detection with SuperPoint/SuperGlue-based front/back classification to resolve the 180° ambiguity in PCA-based (SVD) pose estimation, where the signs of the eigenvectors cannot be determined, meaning the X and Z axes have a 50% probability of flipping 180°. The method is custom-made for the specific ketchup bottle used in this work, and all geometric parameters (thresholds, search regions, label signatures) are based directly on the physical dimensions of the bottle and its labels.

Pipeline

1. **Image loading and ROI masking:** The RGB image (bin_rgb_masked.png) and depth image (bin_depth.png) are loaded. An ROI mask is applied to filter out the background outside the bin.

2. **Geometric label detection:** The detect_labels function (from utils.bottle_detectors) segments red and white regions via HSV thresholding and extracts contours. Labels are divided into front-side and back-side labels based on their aspect ratio (threshold 0.6).

Front-side labels (aspect_ratio ≥ 0.6) have a 5-sided geometry with two parallel sides running along the bottle's length axis (house-shaped), where the tip points towards the bottom of the bottle. Since the label's width (55 mm) and height (52 mm) are nearly identical, the difference is so small that it is easy to get the direction wrong if relying on only a single method. Hough line detection (step 1) also required careful parameter tuning and is sensitive to image quality and perspective distortion. Therefore, a hierarchical fallback strategy with five steps is used to determine the Y-axis:

I. **Hough lines:** The Probabilistic Hough Line Transform detects line segments in the label's edge image. Lines are grouped by angle (tolerance 10°), and only groups with at least two parallel lines are accepted. The longest line from the parallel groups defines the Y-axis direction (perpendicular to the line).
II. **Longest edge:** The label's contour is approximated to a polygon, and the longest edge is identified as the label's base. The point furthest from the base is defined as the tip. The Y-axis is set perpendicular to the base, pointing towards the tip.
III. **Fingerprint matching:** The polygon's corner angles are calculated and matched against the house-shape signature: tip (133°), shoulders (113°), and corners (~90°). If a tip is found, the Y-axis is defined from the centre towards the tip. If two corners are found, the base is defined between them, and the Y-axis is set perpendicular to the base.

IV. **Largest angle:** If no fingerprint match succeeds, the largest angle in the polygon is searched for (requirement >123°). This angle is assumed to be the tip, and the Y-axis is defined from the centre towards the tip.

V. **Minimum bounding rectangle (minAreaRect):** As the final fallback, the smallest rectangle enclosing the contour is calculated, and the longest side is used as the Y-axis.

Each step is verified by matching the Y-vector against a detected cap within a geometrically constrained search region. The search distance of 60–170 mm along the axis was chosen to provide margin around the nominal distance (~106 mm) between the label's centre and the cap, since the projected distance varies when the bottle tilts at different angles. The lateral tolerance of ≤53 mm was chosen with consideration to the cap's diameter of 55 mm, to avoid matching against caps on adjacent bottles. If no cap is found in a step, the next step is tried.

Back-side labels (aspect_ratio < 0.6) are elongated rectangles where the Y-axis is determined directly from the longest side of the rectangle (minAreaRect), without the need for a fallback strategy.

Figure 7: Geometric properties of the target labels utilised by the SuperGlue-Enhanced MultiCue detection pipeline. (Top) The front "house-shaped" label, where the purple arrow indicates the tip pointing towards the bottle base (used for Y-axis direction in the finger-print and largest-angle fallback steps). The two light-blue parallel lines highlight the longitudinal edges targeted by the Probabilistic Hough Line Transform in the first fallback step. (Bottom) The elongated back label, outlined in green to indicate its high aspect ratio (< 0.6), which allows direct Y-axis determination via the minimum area rectangle without requiring the hierarchical fallback strategy. These geometric cues collectively enable robust orientation estimation even when individual detection steps fail due to partial occlusion or perspective distortion.

3. **Validation:** The validate_object function applies five filters to ensure that only valid bottle detections are accepted:
   - **Area (7000–112000 px):** Filters out detections that are too small (image noise, partial occlusions) or too large (background regions, merged bottles).
   - **Solidity (>0.82):** Solidity is defined as the contour's area divided by the area of its convex hull. A value close to 1.0 indicates that the contour is nearly convex, which is expected for a complete label. Low values indicate strong deformation, holes, or that multiple objects have merged.
   - **Aspect ratio (1.0–4.0):** The ratio between the longest and shortest sides of the minimum bounding rectangle (minAreaRect). This ensures that the label has reasonable proportions — neither too narrow (like a line) nor too wide (like a circle).
   - **Dynamic Z-spread:** The standard deviation of depth values (Z) within the contour is calculated and compared against a dynamic threshold: threshold = (median_z × 0.05) + 10.0. If the Z-spread exceeds the threshold, it indicates that the contour is not a planar label but may cover multiple bottles at

different depths, or that the depth data is noisy. The threshold scales with the median distance to allow greater absolute variation at longer distances.

- **Angle of incidence (>30°):** The angle between the bottle's main axis (calculated via SVD on the 3D point cloud) and the camera's optical axis (Z-axis) is calculated. If the angle is less than 30°, it means the bottle is pointing almost directly at the camera, causing the label to be strongly foreshortened and pose estimation to become unreliable.

4. **SuperGlue classification:** For each valid label, correct_pose_with_pnp is run:
   - A 100 px-padded region around the label is extracted and converted to gray-scale.
   - Eight in-plane rotations (0°–315° in 45° steps) are tested to find the alignment that maximizes keypoint matches.
   - The crop is matched against two pre-captured reference images (reference_front_masked.png, reference_back_masked.png) using the SuperGlue indoor-weights model.
   - The side (front or back) yielding the highest match count is selected, producing a Boolean is_backside flag.

Figure 8: Reference images used for SuperGlue front/back orientation classification. (Top) Front view (refer-ence_front_masked.png) showing the house-shaped label. (Bottom) Back view (reference_back_masked.png) showing the elongated label and the small lower label. The ArUco fiducial markers have been seamlessly removed using a custom masking pipeline that combines OpenCV's Telea inpainting algorithm with a cosine-weighted feathering mask. This ensures the deep learning matcher relies solely on the label geometry and text for classification, preventing it from exploiting the high-contrast fiducial patterns as orientation cues while avoiding edge artifacts.

5. **3D-PCA pose estimation:** The calculate_6dof_pose function (from utils.pose_estimator) computes the 6-DoF pose:
   - All valid depth pixels within the contour are back projected to 3D camera coordinates using the pinhole model.
   - The centroid is computed and the point cloud is centred.
   - Singular Value Decomposition (SVD) is applied to find the principal axes.
   - The Y-axis is aligned with the cap direction using cork_pos_2d.
   - The X-axis is constrained along the width (96 mm) and re-orthogonalized via Gram-Schmidt.
   - The Z-axis is computed as X × Y to guarantee a strict right-handed system (det(R) = +1).
   - If is_backside=True, the Z and X axes are flipped to ensure the Z-axis always points towards the bottle's physical front face.

6. **Geometric corrections:** Two deterministic offsets are applied:
   - A 28.5 mm surface-to-centre offset along the local Z-axis translates the origin from the visible label surface to the bottle's geometric centre.
   - A Y-axis centre correction shifts the origin along the bottle's length axis to compensate for the fact that the detected label's centroid does not coincide with the bottle's geometric centre. The correction magnitude is chosen via nearest-neighbour matching against known label heights (99 mm for the upper back label, 52 mm for the front-side label).

7. **Output.** The result is saved as output/SuperGlueEnhancedMultiCueOutput_Masked/enhanced_multicue_detections.json with position, rotation, area, 2D origin, and the is_backside flag per detection.

FoundationPose

FoundationPose is implemented in run_foundationpose_masked.py and executed on an external GPU instance (Google Cloud VM with NVIDIA Tesla T4, 15 GB VRAM) due to local VRAM limitations (Quadro P2000, 5 GB).

Before inference can be run, a 3D reference model of the bottle must be created. This reconstruction was performed in Google Cloud using the BundleSDF pipeline, which uses a NeRF-based approach to fuse multi-view RGB-D images into a signed distance field. Training data consisted of 24 images taken from 3 different perspectives around the bottle (8 images per perspective, total 3×8=24). The resulting mesh (model_centered.obj) has its origin at the bottle's geometric centre.

Figure 9: Manual rotation stage (turntable) utilized for acquiring the 24 reference images required for FoundationPose's few-shot adaptation and BundleSDF mesh reconstruction. Reference image preprocessing pipeline for FoundationPose and BundleSDF mesh reconstruction. (Left) Original capture showing the ketchup bottle on the manual turntable with visible grid pattern and alignment markers. (Centre) Manually edited version where the turntable and background have been removed using image editing software to isolate the object and eliminate potential reconstruction artifacts from non-object geometry. (Right) Binary segmentation mask generated via HSV thresholding, used by FoundationPose during inference to localize the target object within the scene.

Figure 10: Reference image acquisition sequence for FoundationPose. The setup captures 24 images from 3 different elevation perspectives (8 images per perspective). This figure shows one complete 360° rotation sequence from a single perspective, with images captured at 45° intervals. Alternating images show the original capture with the turntable visible (left) and the manually masked version with the turntable removed (right). The top row displays the first 180° of rotation (0°-180°), while the bottom row shows the remaining 180° (180°-360°). This systematic rotation provides sufficient multi-view parallax for the BundleSDF NeRF reconstruction and subsequent few-shot pose estimation.

FoundationPose Training Pipeline

1. **Local preparation:** The script copies bin_rgb_masked.png and bin_depth.png to foundationpose/FoundationPose/demo_data/bin_picking/.
2. **Scene preparation:** prepare_bin_scene.py creates a bin_picking_scene directory containing:

- rgb/000000.png and depth/000000.png (copies of the input data)
- cam_K.txt (camera intrinsics from camera_intrinsics.json)
- masks/000000.png (a binary mask generated via HSV thresholding for red + white, followed by morphological cleaning and connected component analysis; the largest component is selected as the target bottle)

3. **Remote execution:** The scene directory is transferred to the VM via SCP. Inference is run via SSH using the command xvfb-run -a python run_demo.py --mesh_file ... --test_scene_dir .... xvfb provides a virtual framebuffer for OpenGL-based rendering.
4. **Result download:** The pose file debug/ob_in_cam/000000.txt (a 4×4 homogeneous transform) is retrieved via SCP.
5. **Export:** export_fp_detections.py converts the mesh-frame pose to the thesis coordinate system (X_th=X_mesh, Y_th=Z_mesh, Z_th=-Y_mesh) and saves the result as output/FoundationPoseOutput/foundationpose_detections.json in the unified detection format.

4.5 Unified Scoring System

To combine detections from multiple vision methods and select the optimal bottle for robotic grasping, a unified scoring system is implemented in unified_scorer.py. The system uses a two-stage process: first, hard filters are applied for immediate rejection of detections that do not meet critical requirements; then, remaining detections are ranked based on six weighted factors.

Hard filters

Three hard filters are applied sequentially. Detections that fail a filter are immediately rejected and excluded from scoring.

Filter 1: ArUco matching. Each detection's 3D position is compared against all ArUco markers in the ground truth (aruco_ground_truth.json). The detection position is matched to the nearest ArUco marker (smallest Euclidean distance). If this distance exceeds 100 mm, the detection is rejected. This threshold ensures that only detections that can be associated with a known bottle in the scene are accepted.

Filter 2: 2D/3D conflicts. Detections with the has_2d_3d_conflict field set to True are rejected. This captures cases where 2D image analysis and 3D pose estimation yield contradictory results.

Filter 3: Duplicate removal. After scoring, detections are sorted in descending order by total score. Detections closer than 50 mm to an already retained detection are rejected as duplicates. This ensures that the same physical bottle is not selected multiple times.

Scoring Factors

Remaining detections are scored based on six factors, each with equal weight (1/6 ≈ 16.7%). Each factor is normalized to the range [0.0, 1.0] before being multiplied by its weight.

Factor 1: Visible surface (area). Measures how large a portion of the bottle is visible in the image. A larger area indicates less occlusion and a better data foundation for pose estimation. Normalization is performed using min-max scaling across all detections in the batch: $norm_area = \frac{area - \min_area}{\max_area - \min_area}$, where area is the detection's 2D contour area in pixels, and min_area and max_area are the minimum and maximum areas among all detections in the batch, respectively. If all detections have the same area, norm_area is set to 0.5 (neutral value).

Factor 2: Number of detecting methods. Measures how many different vision methods have detected the same physical bottle. First, all detections are clustered based on 3D distance: if the distance between two detections is less than 30 mm, they are grouped as the same bottle. For each cluster, the number of unique methods is counted. Normalization uses a stepwise scale:

- 1 method → 0.25
- 2 methods → 0.50
- 3 methods → 0.75
- 4+ methods → 1.00

More methods agreeing increases reliability.

Factor 3: Match distance to ArUco ground truth. Measures the physical 3D distance between the vision method's estimated position and the ArUco marker's known position. A smaller distance indicates higher accuracy. Normalization uses linear decay:

$$norm_match_distance = 1.0 - \frac{match_distance}{MAX_MATCH_DISTANCE}$$

, where match_distance is the 3D Euclidean distance between the estimated position and the nearest ArUco ground truth position, and MAX_MATCH_DISTANCE is the maximum allowed distance for normalization (50 mm).

Factor 4: Z-height. Measures the bottle's depth from the camera. A lower Z-value means the bottle is closer to the camera (higher up in the bin) and thus easier to reach. Normalization uses min-max scaling, inverted so that lower Z yields a higher score:

$$norm_z_height = \frac{\max_z - z}{\max_z - \min_z}$$

, where z is the detection's depth coordinate, and min_z and max_z are the minimum and maximum depth values among all detections in the batch, respectively.

Factor 5: Detection type. Prioritizes detections based on the type of detection. LABEL detections (from label analysis) provide more unambiguous orientation data than RED_BODY detections (from red body segmentation). FoundationPose is given the same priority as LABEL. Normalization uses fixed values:

- LABEL → 1.0
- FOUNDATIONPOSE → 1.0
- RED_BODY → 0.35

Factor 6: Validation type. Prioritizes detections based on how strict the validation was. strict validation (geometric and depth-based filters, see Section 4.4) is geometrically more reliable than permissive validation. Normalization uses fixed values:

- strict → 1.0
- permissive → 0.35

The total score for each detection is computed as a weighted sum:

$$total_score = \frac{1}{6} \cdot (norm_area + norm_methods + norm_match_distance + norm_z_height + norm_detection_type + norm_validation_type)$$

, where norm_area, norm_methods, norm_match_distance, norm_z_height, norm_detection_type, and norm_validation_type represent the normalized scores of the six respective factors, and total_score ∈[0.0,1.0].

The detection with the highest total_score is selected as the best bottle for robotic grasping. The result is sent to RobotStudio via a socket interface with position (position_mm) and rotation (euler_deg).

4.6 Evaluation Methodology

To enable a fair and reproducible comparison of the implemented vision methods, a dedicated evaluation tool (evaluation_metrics.py) computes pose estimation errors against the ArUco-based ground truth. The metrics, matching logic, and binary success criteria strictly follow the definitions established in Section 3.7.

Implementation of the evaluation pipeline

For each matched detection, the script computes the Translation Error, Rotation Error, and Symmetry-Aware Rotation Error (RESA) as defined in Section 3.7. The matching procedure pairs each detection with the closest ground truth bottle, applying the 5°5cm and 10°10cm thresholds to classify True Positives, False Positives, and False Negatives. Aggregated metrics (Precision, Recall, and Precision at Threshold) are then calculated per scenario based on these classifications.

Exclusion of the RED_BODY Detection Path

During the experimental phase, the purely geometric RED_BODY detection path (based on HSV thresholding and watershed segmentation) was intentionally excluded from the final comparative evaluation. Empirical testing revealed that under the strong, directed lighting conditions required for this setup, this path yielded highly imprecise poses. The combination of specular reflections on the glossy plastic and the unresolvable 180-degree PCA ambiguity on textureless surfaces made the RED_BODY outputs unreliable for safe robotic manipulation. Consequently, the final evaluation focuses exclusively on the LABEL detection path, where SuperGlue successfully mitigates the primary rotational ambiguities.

Filtering of Ground Truth Data

To prevent the evaluation from penalizing vision methods for failing to detect bottles that are physically occluded from the camera's view, the ground truth dataset is filtered dynamically per cycle. Before metrics are computed, the script performs a fresh ArUco detection on the high-resolution reference image (bin_rgb_aruco.png). Only ground truth bottles where at least one ArUco marker is successfully detected in the image are included in the evaluation pool for that specific cycle.

Cycle-Based JSON Logging

To meet the reproducibility requirements described in Section 3.9, the evaluation script does not merely print results to the terminal. Instead, all per-detection metrics (TE, RE_{raw}, RE_{SA}, and binary flags) as well as aggregated scores (Precision, Recall, TP, FP, FN) are serialized into a structured JSON file: output/evaluation_results/cycle_XXX_metrics.json. The cycle number is either passed explicitly via a command-line argument or automatically detected from the results directory. This structured logging enables post-hoc statistical analysis across all 30 experimental cycles.

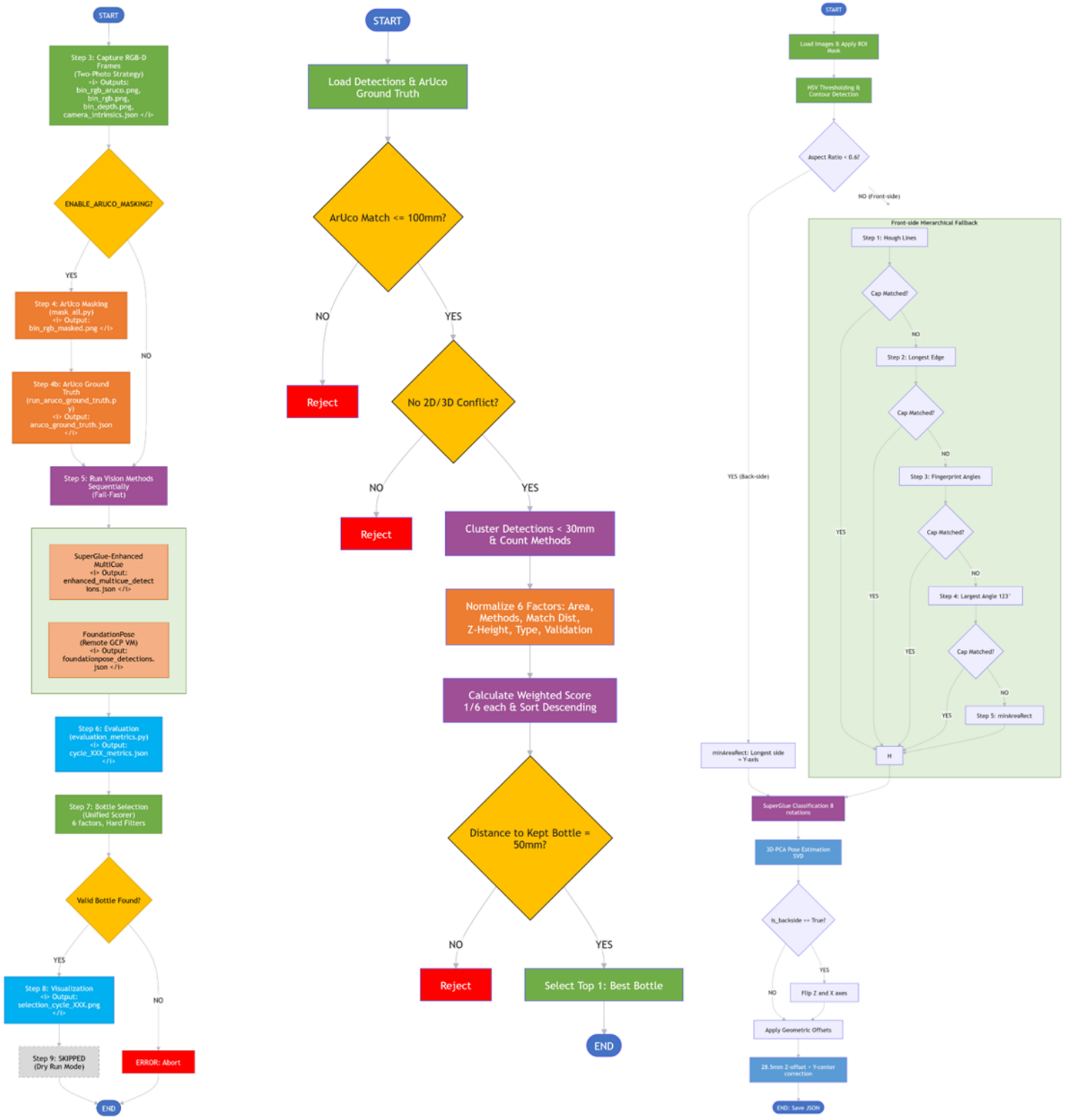


Figure 11: Overview of the system architecture and core processing pipelines. (Left) The overall comparative evaluation pipeline, illustrating the sequential execution from data capture (Two-Photo Strategy) through ArUco preprocessing, vision method execution, evaluation, and final bottle selection. (Centre) The Unified Scorer's two-stage selection process, detailing the hard filters (ArUco match ≤ 100 mm, 2D/3D conflict check), 3D clustering, normalization of six scoring factors, and final duplicate removal. (Right) The SuperGlue-Enhanced MultiCue pipeline, showing the HSV thresholding, aspect-ratio-based branching, hierarchical Y-axis fallback strategy for front-side labels, SuperGlue front/back classification, and 3D-PCA pose estimation with geometric offsets.

4.7 Simulation and RobotStudio Integration

Socket Protocol and Simulation-Based Validation

Communication between Python and the RAPID program in RobotStudio was implemented via a TCP/IP socket connection. Python acts as the TCP server (listening on port 30000), while RAPID acts as the TCP client and connects at startup. The protocol follows a strict handshake logic to ensure synchronization:

Protocol sequence per cycle:

1. RAPID sends "READY" to signal completion of the previous cycle.
2. Python responds with "START" to acknowledge receipt.
3. Python sends the number of bottles (nTotalBottles).
4. Python sends the index of the target bottle (nTargetIndex).
5. For each bottle, Python sends the pose in the format "x,y,z,rx,ry,rz" (mm + degrees, Euler XYZ).
6. For the target bottle, Python additionally sends grasp data (approach/grasp positions and quaternions, grip_mode, grasp_from_top, cap_up).

After each message, Python waits for an "OK" acknowledgement from RAPID before sending the next message.

Simulation-Based Validation: The overarching goal of the digital twin integration is to transmit real-time 6-DoF poses, estimated by the vision pipeline from the physical bin, directly to the robot controller. However, this requires a rigid spatial transformation between the camera frame and the robot base frame (wobj0), which relies on the physical mounting and calibration of the corner ArUco marker on the bin.

Because this physical integration was not completed within the project timeframe, the current validation operates as a simulated proxy. A standalone Python script transmits hardcoded "ground truth" poses (extracted from RobotStudio) augmented with synthetic Gaussian noise (±3 mm translation, ±2° rotation) to emulate the uncertainty of the vision pipeline.

Upon receiving these simulated poses, RAPID utilizes Smart Components (SC_BottlePoseInBin) to dynamically spawn and position virtual bottles at the exact received coordinates within the virtual bin. The robot then executes the full manipulation sequence (approach, grasp, fixture placement, and unscrewing) on these virtually positioned bottles.

This approach successfully validates the socket communication protocol, RAPID's data parsing capabilities, and the heuristic grasp synthesis logic. The transition to a fully spatially accurate digital twin, driven by real-time camera data via the corner ArUco anchor, is identified as future work in Section 6.2.

Grasp Synthesis and RAPID Execution

To enable autonomous manipulation based on estimated 6-DoF poses, a heuristic grasp synthesis was implemented in Python (compute_grasp() in robot_socket.py), alongside a sequential manipulation logic in RAPID. Together, these handle the entire pipeline from pose analysis to physical (or simulated) task execution.

Grasp Synthesis in Python

The compute_grasp() function analyses each bottle pose (represented as a 4×4 homogeneous transformation matrix) and determines the grasping strategy based on its orientation relative to the world Z-axis (upward). The algorithm categorizes bottles into two states:

- **Standing or upside-down bottle (|Z_world[2]| > 0.7):**
  The bottle's length axis is nearly vertical. The gripper approaches along the bottle's length axis, always from above in world coordinates, but directed towards either the cap or the bottom depending on the bottle's orientation. Flags are set to grasp_from_top = 1 and grip_mode = 2 (wide grip, 96 mm). The cap_up variable is determined by whether the length axis points upward (1) or downward (0). An offset of 70 mm is applied along the length axis from the centre towards the cap to prevent the gripper's palm from colliding with the bottle body. The approach position is calculated an additional 100 mm along the same direction. To select which side the gripper should face, the bottle's side normals (+X, -X, +Y, -Y) are projected onto the horizontal plane, and the side whose normal best aligns with the direction towards the robot base is selected.

- **Lying bottle (|Z_world[2]| ≤ 0.7):**
  The bottle is lying on its side. The algorithm selects the side whose world nor mal has the largest Z-component (the side pointing most upwards and thus most accessible). The grasp templates (GRASP_TEMPLATES) define a local grasp offset, the gripper's Z-axis (approach direction), Y-axis (along the bottle length), and grip_mode (1 for narrow 64 mm sides, 2 for wide 96 mm sides) for each side. The approach position is placed 100 mm above the grasp point in the world Z-direction.

For each grasp, a local gripper frame (X, Y, Z axes) is constructed to ensure correct orientation, converted to quaternions in ABB format ([qw, qx, qy, qz]), and sent to RAPID along with position data and control flags.

RAPID Execution and Smart Components

The RAPID program (MainModule) receives the poses and parameters via the socket protocol and executes a predefined sequence:

1. **PlaceAllBottles:** For each received bottle, the pose (RV_x–RV_rz) is parsed and sent to the Smart Component SC_BottlePoseInBin via the signal DI_MoveBottleToPose. The component uses a Source node to dynamically spawn a virtual bottle at the received coordinates within the virtual bin. Once placed, the component returns DO_BottleMovedToPose, and RAPID acknowledges Python. For the target bottle, the approach and grasp poses, as well as control flags (grip_mode, grasp_from_top, cap_up), are additionally parsed.
2. **GripTargetBottle:** The robot moves to the approach position (MoveJ), sets the gripper orientation (do_ApprochFrontBackPosition), performs a linear approach to the grasp position (MoveL), and activates the appropriate gripper mode based on grip_mode (do_GripRLSidePosition for narrow grip,

do_GripFrontBackPosition for wide grip). The bottle is then lifted, and the robot retracts to a safe position.

3. **PlaceInFixture:** The fixture is designed to receive the bottle in two orientations: either bottom-down (standard position for unscrewing) or cap-down (intermediate position for grip changing) (see *Figure 12***Fel! Hittar inte referenskälla.**). Based on grasp_from_top and cap_up, the appropriate orientation is selected. If the bottle was grasped from the bottom (cap_up = 0), it is first placed cap-down in the fixture. This allows the robot to release its bottom grip, take a new side grip on the bottle, lift it, flip the bottle 180° in the air, and finally place it bottom-down in the fixture for unscrewing. This sequence is handled by FlipBottleInFixture.
4. **UnscrewCapPath:** When the bottle is standing bottom-down in the fixture, the robot switches to the cap gripper (AGS_TCP_CAP) and performs three repeated 180° rotations (via UnscrewCapMid and UnscrewCapStop) to unscrew the cap. The cap is then gripped, detached from the fixture (do_DetachCapFixture), and transported to CapDropPoint for disposal.
5. **PickBottleToPourKetchup and PourKetchupPath:** The robot grasps the bottle from the fixture and transports it to the PokeHole station. This station consists of an air-filling nozzle (like those used for industrial bottle filling) equipped with a razor-sharp cross-shaped conic cutting tool on top that pierces the bottle's plastic membrane (see *Figure 12* below). This must be custom made for this application. When the bottle is placed over the nozzle, compressed air is activated (do_PressureOn), which flows up through the nozzle and out under the cutting tool to empty the bottle's contents using air pressure.

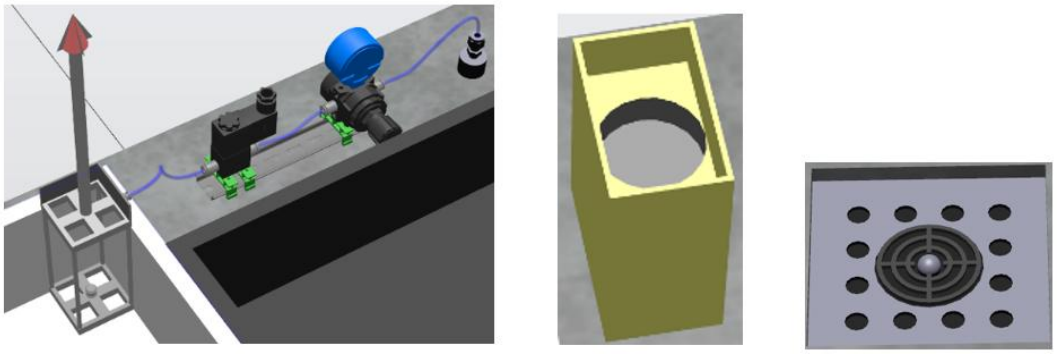


Figure 12: (Left)The air-filling nozzle equipped with the razor sharp cross-shaped conic cutting tool on top that pierces the bottle's plastic membrane. This must be custom made for this application. (Centre) The custom-made fixture that can receive the bottle in two orientations, bottom-down or cap-down. (Right) The bottle rinser.

6. **RinseBottlePath and PlaceInDishwasher:** The bottle is rinsed at the Target_120 position (the bottle rinser, see *Figure 12***Fel! Hittar inte referenskälla.**) and subsequently placed in a dish rack (at one of eight predefined positions, Target_30 to Target_110), which are selected sequentially via a counter (nDishwasherIndex).

This sequence validates that the entire manipulation flow, from pose estimation to a fully processed bottle in the dish rack, can be executed correctly in the simulation environment, including the handling of various bottle orientations, grasp strategies, and downstream operations (see front cover for a visual representation of the station from RobotStudio).

4.8 Interactive Parameter Tuning and Validation

To ensure methodological rigor and eliminate arbitrary "trial-and-error" in setting image processing parameters, dedicated interactive tuning tools were developed. Rather than hardcoding values based on visual guesswork, two distinct OpenCV-based tuning scripts were built to quantitatively validate the parameters for the two most critical geometric stages of the pipeline: label edge detection and cap identification.

Tool 1, Label and Edge Detection Tuner (tune_mask_and_hough.py):

This tool is designed to tune the geometric label detection used in the SuperGlue-Enhanced MultiCue pipeline. It exposes parameters across four synchronized windows: the original image, the binary mask, the Canny edge map, and a parameter panel.

- **Adjustable Parameters:** HSV thresholds for white and red segmentation, morphological kernel sizes (with automatic enforcement of odd sizes), Canny thresholds (with validation ensuring low < high), and Hough Line Transform parameters (threshold, minimum length, maximum gap).
- **Quantitative Validation Logic:** The tool goes beyond visual inspection by performing real-time geometric analysis. For every detected label contour, it extracts all Hough lines within its bounding box and groups them by angle (10° tolerance). Because the front "house-shaped" label physically contains parallel edges, the tool verifies that exactly one group of parallel lines is detected. It highlights this status in real time (green for "OK", red for "FAIL"). Additionally, a histogram of line lengths is rendered to help the operator distinguish true label edges from high-frequency noise.

Tool 2, Cap Detection Tuner (interactive_cap_tuner.py):

This tool addresses a specific domain challenge: distinguishing bottle caps from white labels that contain text. Both appear as white blobs, but caps lack the dense text patterns of labels.

- **Adjustable Parameters:** Minimum/maximum area limits, circularity threshold (MIN_CIRC), HSV thresholds for dark/text pixels (BLACK_V_MAX), morphological dilation iterations, the analysis radius (ANALYS_RADIE_PX), and the text percentage threshold (TEXT_THRESH).
- **Quantitative Validation Logic:** For every detected white blob (candidate cap), the tool calculates its centroid. It then counts the percentage of "text pixels" (segmented via red/black HSV thresholds) within a circle defined by the analysis radius. If the text percentage is below TEXT_THRESH, the blob is classified as a valid cap ("OK"). If it exceeds the threshold, it is rejected as a label ("TXT"). The tool prints a sorted, real-time list of all detected objects with their exact coordinates and text percentages, allowing the operator to precisely tune the threshold to maximize true positives while eliminating false positives.

Reproducibility and Configuration Locking

Both tools share a rigorous documentation workflow. Once optimal parameters are identified, the operator can save the state, which automatically generates:

1. A PNG image of the visualized scene with the chosen parameters and validation overlays embedded.

2. A structured JSON file containing a timestamp, the exact integer values for all adjusted parameters, and an array of detection statistics (e.g., label_stats or object text percentages).

This workflow guarantees that every threshold value in the pipeline has a traceable, empirical justification grounded in visual and statistical data. The locked_params used in the production code are derived directly from these validated JSON logs, effectively eliminating the risk of configuration drift between the tuning phase and the final experimental evaluation.

4.9 Computational Infrastructure and Cloud-Based Inference

As detailed in Section 3.2, the local workstation's NVIDIA Quadro P2000 (5 GB VRAM) is insufficient for FoundationPose's inference requirements. Therefore, the computationally intensive steps are delegated to a Google Cloud virtual machine equipped with an NVIDIA Tesla T4 GPU (15 GB VRAM). Prior to inference, the required 3D reference model (mesh) was generated in the cloud using the BundleSDF pipeline, fusing 24 images taken from three different perspectives around the ketchup bottle into a final mesh (model_centered.obj).

To integrate the cloud-based inference into the local workflow without manual intervention, an automated wrapper script (run_foundationpose_masked.py) was developed. This script orchestrates the entire process via SSH and SCP:

- **Local Preparation:** The script copies the masked input files and executes prepare_bin_scene.py locally to create a scene directory containing the RGB image, depth image, binary mask, and camera intrinsics.
- **Cleaning and Transfer:** Any previous results on the VM are cleaned via SSH, after which the new scene directory is securely transferred to the VM via SCP.
- **Remote Inference:** The inference is triggered via SSH. Because FoundationPose relies on OpenGL for rendering, the command is executed using env -u QT_QPA_PLATFORM xvfb-run -a python run_demo.py --mesh_file ... --test_scene_dir .... This provides a virtual display environment (X Virtual Framebuffer) that allows the headless cloud server to render graphical contexts. The absolute path to the Python interpreter in the VM's Conda environment is used to ensure the correct environment.
- **Result Retrieval and Export:** The resulting 4×4 transformation matrix is downloaded locally via SCP, and export_fp_detections.py converts it into the unified JSON format.

4.10 Directory Structure and Parameter Locking

The data management policy defined in Section 3.5 is implemented through a strict directory structure and automated logging. The data/ directory contains all inputs (raw and masked RGB-D images, aruco_ground_truth.json, and camera_intrinsics.json). The output/ directory stores all pipeline results, and the config/ directory holds static configuration files, such as aruco_calibration.json.

To meet traceability requirements, the evaluation_metrics.py script generates a dedicated file (output/evaluation_results/cycle_XXX_metrics.json) per experimental cycle. Furthermore, the interactive tuning tools (Section 4.8) generate JSON logs (e.g.,

fas1_...json) that document exactly which parameters were tested alongside detection statistics and timestamps.

To eliminate the risk of configuration drift, the final validated parameters (locked_params) from the tuning tools are hardcoded directly into the production scripts. This guarantees that the exact parameters validated interactively are the ones executed during the 30 experimental cycles, fulfilling the reproducibility requirements outlined in Section 3.5.

4.11 Execution and Data Collection

A total of 30 experimental trials were conducted, divided into three sequential blocks based on occlusion level. First, 10 trials with high occlusion (8 bottles) were executed, followed by 10 trials with medium occlusion (5 bottles), and finally 10 trials with low occlusion (3 bottles). Within each block, a new, random layout was generated for each trial using the physical protocol described in Section 3.4. This block structure enables a clear and isolated statistical analysis of how each vision method performs at specific levels of visual complexity and occlusion.

5 Results and discussion

5.1 Specification-Based Gripper Evaluation: Results and Recommendation

This section presents the results of the evaluation methodology, including the six candidate paradigms and the five criteria C1–C5, as defined in Section 3.10. These were assessed through qualitative matching of documented mechanical principles and class-level characteristics [11] against the physical properties of the target object.

Assessment per Paradigm and Comparison Matrix

All assessments below are qualitative syntheses: each symbol indicates the degree to which a paradigm's documented mechanical principles and class-level characteristics satisfy the respective criterion in Section 3.10. Justifications are provided cell by cell in the text, and Table 3 summarises the results. Key: ++ strong advantage; + advantage; ± mixed; – disadvantage; — critical weakness.

Row 1: Rigid parallel-jaw grippers (Overall: ±)

- **C1 (+):** Linear, parallel kinematics enables the design of fingers and the selection of actuators that can be dimensioned to cover the required span (approximately 50–106 mm); profiled jaws accommodate the flattened profile and top-down grasping.
- **C2 (+):** Stable form and friction closure with textured pads; clamping force is controllable (electric/pneumatic actuation). Although a filled bottle (1 kg) is stiff and resists deformation, sufficient and even clamping force is required to counteract slipping against the glossy plastic under dynamic accelerations – provided that the jaw profile is not purely conformal, since otherwise the object may rotate about its own axis within the grasp.
- **C3 (–):** No passive compliance – does not absorb positional errors from the vision stage.
- **C4 (±):** Slender fingers are possible, but the actuator requires lateral clearance for lying bottles.
- **C5 (±):** Profiled jaws can in principle enclose the lower cylinder, but a purely semicircular (conformal) profile provides no form closure against rotation about the object's own axis – the torque must be transmitted solely through friction, with the risk that the cap or bottle slips/rotates within the grasp. Anti-rotation features (planar contact surfaces, ribs, or V-geometry) are required for reliable decapping.

Overall (±): Feasible but demanding; strong on geometry and payload, demanding on decapping (requires anti-rotation design), weak against perception uncertainty.

Row 2: Angular grippers (Overall: —)

- **C1 (–):** The arcuate motion yields poorly controllable contact geometry across the full span.

- **C2 (–):** Pinch grasp against a glossy surface provides poor resistance for 1 kg; risk of slippage and ejection.
- **C3 (–):** Rigid kinematics; angular errors are amplified at the fingertips.
- **C4 (—):** The jaws splay outwards in the open position – unacceptable collision risk in the bin.
- **C5 (–):** Difficult to direct and transmit torque to the lower cylinder with the arcuate closing motion.

Overall (—): Excluded due to collision risk and payload uncertainty.

Row 3: Vacuum grippers (Overall: —)

- **C1 (—):** The geometry lacks completely flat surfaces for suction cups. The relatively flat front and back surfaces often exhibit slight convexity, while the right- and left-hand sides show somewhat greater convexity. For standing bottles, the only surface exposed from above is the small, irregular cap – an impermissible grasp point due to leverage effects and the risk of actuating the flip-top hinge.
- **C2 (—):** 1 kg on a small suction cup produces leverage effects and a risk of cap fracture/hinge actuation; the glossy curved surface compromises the seal. Although the bottle is filled (1 kg) and therefore stiff, suction cups cannot transmit the enormous shear forces arising during lifting and acceleration of a 1 kg load. Since the surface is often slightly convex, the perfect seal required to maintain the vacuum against these shear forces is absent, causing the grasp to lose the object. When lifting from the side, peeling moments are added, which a vacuum cup mechanically cannot withstand.
- **C3 (±):** A compliant cup damps minor approach errors, but loss of seal is a critical failure mode.
- **C4 (+):** Compact profile without lateral sweep – good accessibility in clutter.
- **C5 (—):** Cannot transmit any torque – decapping impossible.

Overall (—): Excluded: incompatible with the lifting and decapping requirements.

Row 4: Multi-fingered dexterous grippers, non-adaptive (Overall: –)

- **C1 (+):** Independently controlled fingers cover the span and can be positioned for both strategies.
- **C2 (+):** Force distribution over multiple contact points provides a stable 1 kg grasp without high local clamping force.
- **C3 (±):** The rigid links and stiffly controlled (non-compliant) joints of the fingers lack mechanical compliance, but the multi-fingered envelopment provides some geometric error tolerance.
- **C4 (–):** Bulky hand with many actuators – collision risk in the narrow bin.
- **C5 (±):** Can in principle grasp the lower cylinder, but complex coordination is required to avoid loading the hinge.

Overall (–): Excluded: high weight/complexity without decisive advantage over paradigm 5.

Row 5: Adaptive mechanical grippers, underactuated (Overall: ++)

- **C1 (+):** Passive linkage conformation handles the flattened profile and both grasping strategies within the span.
- **C2 (+):** Self-centring, enveloping grasp distributes the force without excessive clamping force.
- **C3 (+):** Mechanical compliance absorbs minor approach errors (cf. self-centring/leading features in [12]).
- **C4 (±):** More compact than a dexterous hand, but enveloping fingers require some clearance around a standing bottle.
- **C5 (+):** Sufficient torsional stiffness through the linkage arms to transmit breakaway torque to the lower cylinder.

Overall (++): Best balance across all five criteria.

Row 6: Adaptive soft finger grippers (Overall: +)

- **C1 (++):** Ultimate conformation – handles all geometries and both strategies with minimal precision requirements.
- **C2 (±):** Gentle force distribution protects the bottle, but friction against the glossy 1 kg surface is uncertain at high accelerations.
- **C3 (++):** Elastic fingers damp approach errors and collision forces.
- **C4 (+):** Slender, compliant fingers with mild collision consequences.
- **C5 (–):** Low torsional stiffness – cannot reliably transmit the decapping torque.

Overall (+): Strongest in the lifting phase, weakest in the decapping phase.

Table 3: Comparison matrix of the six candidate paradigms against criteria C1–C5

Paradigm	C1	C2	C3	C4	C5	Overall
Rigid parallel-jaw grippers	+	+	–	±	±	±
Angular grippers	–	–	–	—	–	—
Vacuum grippers	—	—	±	+	—	—
Multi-fingered dexterous grippers (n	+	+	±	–	±	–
Adaptive mechanical grippers (under	+	+	+	±	+	++
Adaptive soft finger grippers	++	±	++	+	–	+

Recommendation

The evaluation recommends **adaptive mechanical grippers (underactuated)** as the first choice, with **adaptive soft finger grippers** as the second choice. The choice stands between the two paradigms that manage the tension between criterion 3 (compliance against vision errors) and criterion 5 (torsional stiffness for decapping): the soft finger gripper maximises form compatibility and error tolerance in the lifting phase, whereas the underactuated mechanical gripper offers the stiffness required to transmit the decapping torque without loading the flip-top hinge. This trade-off illustrates the coupling between perception uncertainty and manipulation requirements identified as a knowledge gap in Section 2.5.

Furthermore, it should be noted that because the target object features a glossy plastic surface, any recommended enveloping or friction-based gripper paradigm must

be equipped with high-friction, textured elastomeric finger pads to prevent slippage under dynamic accelerations of the 1 kg payload.

Limitations of the Evaluation

In accordance with the limitations defined in Section 1.3, this evaluation is solely theoretical and specification based. This means that the assessment is based on matching the object's physical specifications against the documented mechanical principles of each paradigm. No physical grasp tests, dynamic tolerance simulations, or empirical measurements of actual grasp success rates have been performed. The results therefore constitute a qualitative ranking of paradigms rather than a quantitative performance measurement.

5.2 Quantitative Evaluation of Vision Methods

Overall Detection and Pose Accuracy(Static Scenarios)

The aggregated performance of SuperGlue-Enhanced MultiCue and FoundationPose across the 30 experimental cycles (Scenarios 1–3) is presented in Table 4. FoundationPose achieved a Precision at 10°10cm of 43.3%, nearly double that of SuperGlue-Enhanced MultiCue (21.1%). Under the more stringent 5°5cm criterion, FoundationPose reached 13.3% compared to SuperGlue's 3.3%. The methods exhibited different detection behaviours: FoundationPose achieved a Recall score of 0.111, indicating a conservative strategy with fewer false positives relative to its detections. SuperGlue-Enhanced MultiCue showed a marginally higher Recall score (0.132), suggesting a more permissive detection approach.

For successfully matched detections (within 10°10cm), FoundationPose demonstrated superior positional accuracy with a median Translation Error of 17.1 mm (mean: 20.4 mm), compared to SuperGlue's 33.1 mm (mean: 31.6 mm). Both methods achieved comparable Symmetry-Aware Rotation Errors (7.5° and 6.2° median, respectively), with maximum errors remaining below the 10° threshold.

Table 4: Overall performance metrics for the evaluated vision methods across all 30 experimental cycles

Method	Recall score (mean)	Precision (5°5cm)	Precision (10°10cm)	Median Trans. Error	Median Rot. Error (SA)
S-E-M	0.132	3.3 %	21.1 %	33.1 mm	6.2°
FP	0.111	13.3 %	43.3 %	17.1 mm	7.5°

To provide a more granular view of these results, Table 5 details the performance metrics across the three specific clutter scenarios (High, Medium, and Low). This breakdown highlights the variation in precision and translation error depending on the occlusion level, complementing the aggregated data presented in Table 4.

Table 5: performance metrics across the three specific clutter scenarios (High, Medium, and Low)

Scenario	Method	Precision (5°5cm)	Precision (10°10cm)	Median Trans. Error
High Clutter (8 bottles)	**S-E-M**	0.0 %	11.8 %	23.1 mm
	FP	20.0 %	60.0 %	16.2 mm
Medium Clutter (5 bottles)	**S-E-M**	4.5 %	31.8 %	31.1 mm
	FP	20.0 %	50.0 %	14.6 mm
Low Clutter (3 bottles)	**S-E-M**	8.7 %	21.7 %	39.2 mm
	FP	20.0 %	20.0 %	39.1 mm

Precision and Recall per Cycle

The cycle-level distributions reveal the stability of each method across the 30 trials. At the stringent 5°5cm criterion (*Figure **13***), the distributions are heavily collapsed at zero, resulting in no visible bars for the vast majority of cycles. The plot only reveals single outlier cycles for each method: approximately 0.33 precision for SuperGlue and 1.0 for FoundationPose, illustrating the extreme difficulty of meeting this strict threshold. At the primary 10°10cm criterion (*Figure 14*), FoundationPose demonstrates a distribution heavily concentrated at 1.0, indicating that when it produces detections, they are almost exclusively true positives. SuperGlue's distribution is lower and wider, with a maximum near 0.46 and a median around 0.46, reflecting a higher rate of false positives across cycles.

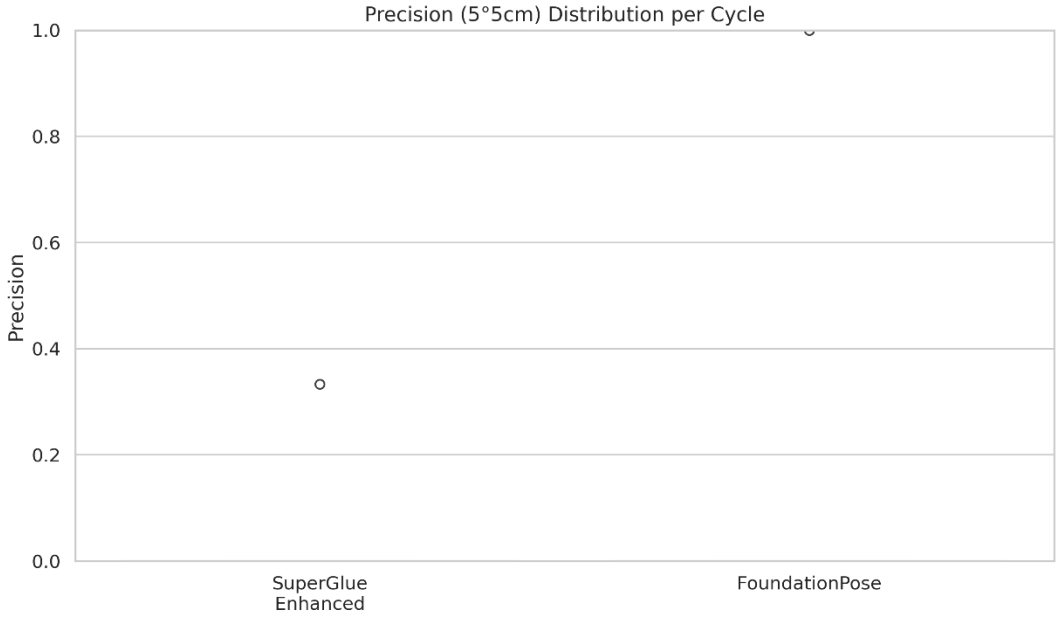


Figure 13

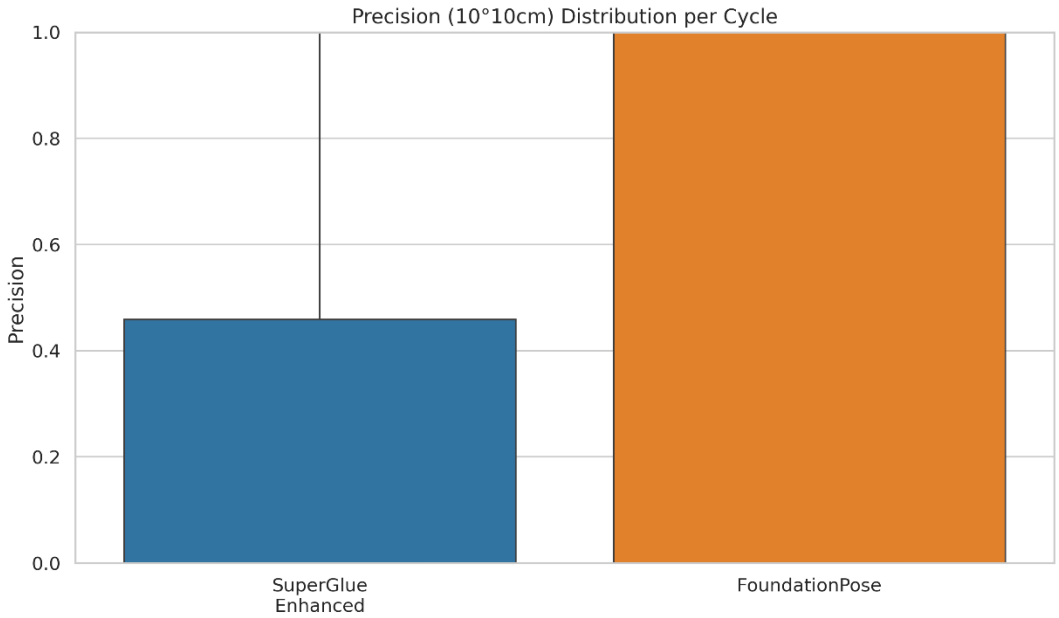


Figure 14

The Recall score distribution per cycle (*Figure 15*) shows that SuperGlue achieves a slightly higher mean recall (0.132, max 0.50) compared to FoundationPose (0.111, max 0.33), confirming its more permissive detection strategy at the cost of lower precision.

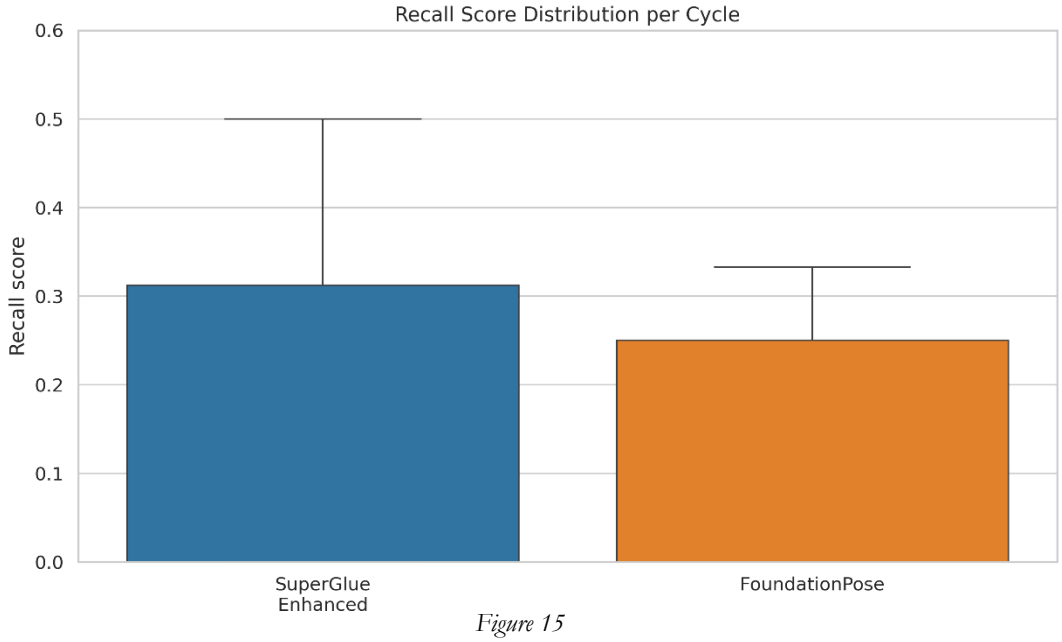


Figure 15

Performance Across Clutter Levels (Static Scenarios)

The scenario-based analysis reveals FoundationPose's clear advantage under high occlusion, achieving 60.0% Precision at 10°10cm with 8 bottles compared to Super-Glue's 11.8% (*Figure 16*). At the stricter 5°5cm criterion, FoundationPose reaches 20.0% while SuperGlue achieves only 11.8%. This performance gap narrows at medium clutter (5 bottles), where FoundationPose maintains 50.0% at 10°10cm against SuperGlue's 31.8%. Contrary to expectations, both methods degraded significantly in the low-clutter scenario (3 bottles), with FoundationPose dropping to 20.0% and SuperGlue to 21.7% at 10°10cm.

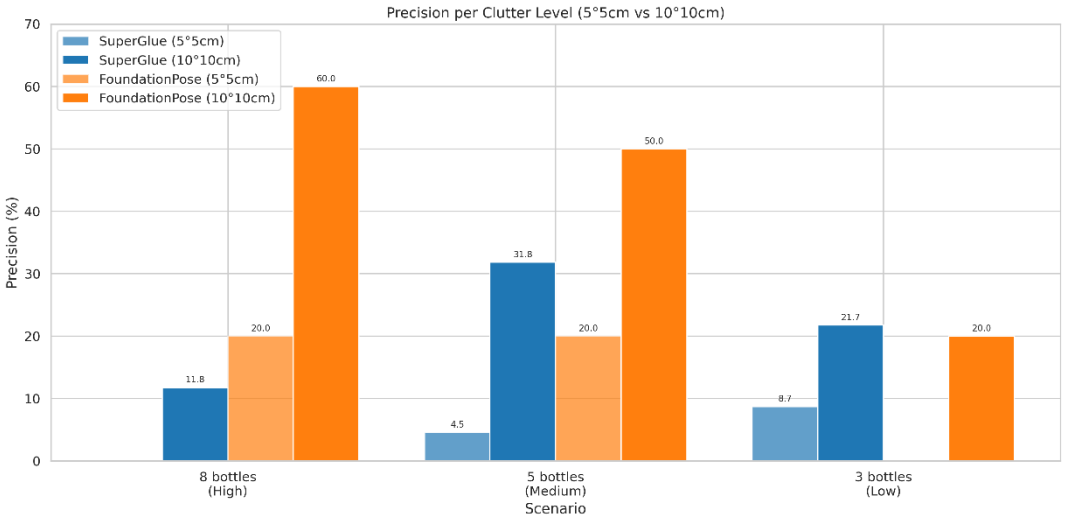


Figure 16

Recall scores per clutter level (*Figure 17*) follow a similar pattern. FoundationPose achieves 12.2% Recall at 10°10cm with 8 bottles, while SuperGlue reaches 4.1%. At medium clutter, SuperGlue's Recall increases to 20.0%, surpassing FoundationPose's 14.3%. In the low-clutter scenario, SuperGlue maintains 19.2% Recall compared to FoundationPose's 7.7%.

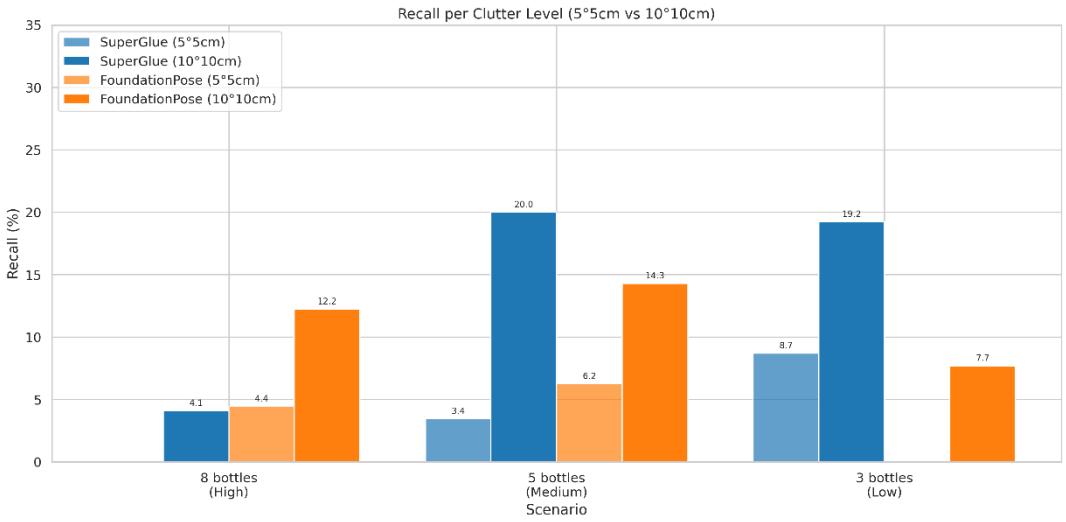


Figure 17

Error Distribution for Successful Detections

For detections meeting the 10°10cm criterion, the Translation Error distributions (*Figure 18*) show FoundationPose's tighter interquartile range (approximately 14–20 mm) and minimum error of 3.1 mm, indicating high precision under favourable conditions. SuperGlue exhibited a wider spread (approximately 25–40 mm interquartile range) with a minimum of 10.5 mm. FoundationPose's maximum error reached 60.6 mm (one outlier), while SuperGlue's reached 58.7 mm. Both methods produced errors well within the 100 mm threshold for the majority of successful detections.

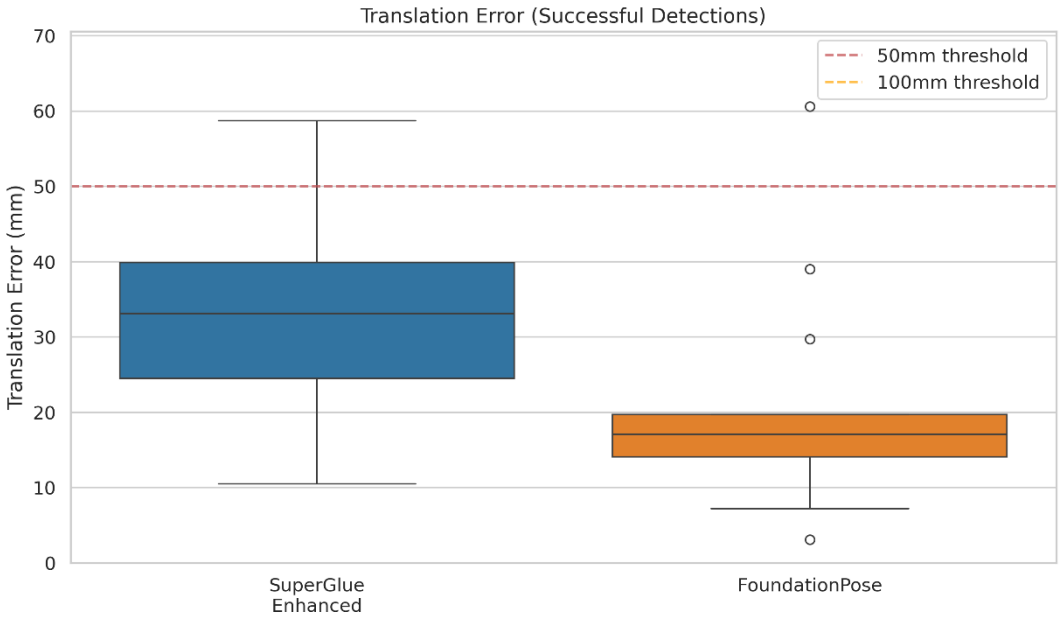


Figure 18

Symmetry-Aware Rotation Errors (*Figure 19*) were remarkably similar for both methods, with medians of 7.5° (FoundationPose) and 6.2° (SuperGlue), and maximum errors of 9.5° for both. The interquartile ranges were tight (approximately 5–8° for SuperGlue and 5–8° for FoundationPose), with the 5° threshold line intersecting both boxplots near their lower quartiles, indicating approximately 25% of successful detections meet the stricter 5°5cm criterion.

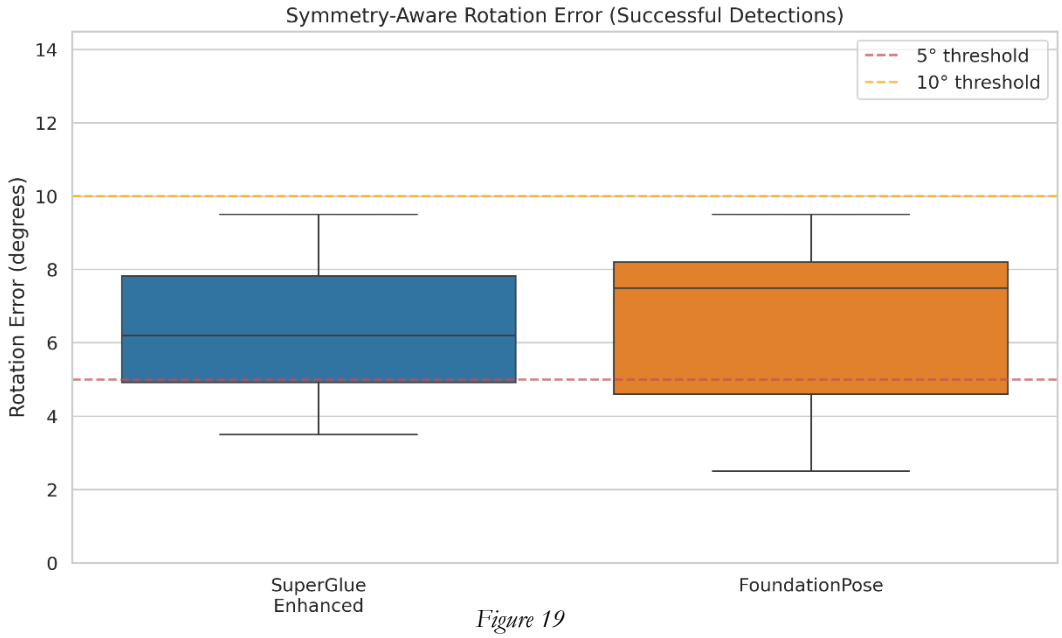


Figure 19

Handling of Rotational Ambiguity

Figure 20 illustrates the critical impact of applying symmetry-aware metrics to the raw rotation errors. For SuperGlue-Enhanced MultiCue, the raw errors are predominantly below 10°, with only a single outlier (Detection Index 13) at ~175°. In stark contrast, FoundationPose exhibits raw rotation errors clustering around 170–180° for the majority of detections. However, when the symmetry-aware correction is applied, these massive errors for FoundationPose collapse to the 0–10° range, aligning closely with SuperGlue's performance. The underlying architectural causes of this raw rotational ambiguity are discussed in detail in Section 5.3.

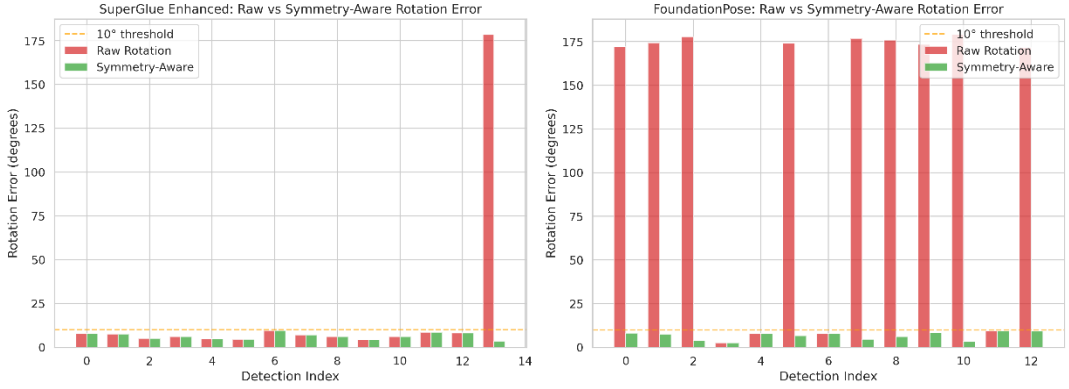


Figure 20

Dynamic Sequential Bin-Picking Evaluation

The dynamic experiment (10 sequences, 61 total observations) revealed substantially different performance characteristics from the static evaluation. The comprehensive results are summarized in Table 6. Overall Recall at 10°10cm dropped to 17.4% for FoundationPose and 16.1% for SuperGlue, with 0.0% Recall at 5°5cm for both methods. When detections succeeded, median Translation Errors were 17.5 mm (FoundationPose) and 22.8 mm (SuperGlue), with Symmetry-Aware Rotation Errors of 7.6° and 4.6°, respectively.

Table 6

Sequential Bin-Picking Experiment - Summary Table

Metric	FoundationPose	SuperGlue Enhanced
TOTAL - Recall (10°10cm) (%)	**17.4**	**16.1**
TOTAL - Recall (5°5cm) (%)	**0.0**	**0.0**
TOTAL - TE Mean (mm)	**36.8**	**24.0**
TOTAL - TE Median (mm)	**17.5**	**22.8**
TOTAL - RE_SA Mean (°)	**7.6**	**5.7**
TOTAL - RE_SA Median (°)	**7.6**	**4.6**
TOTAL - Time Mean (s)	**26.660**	**4.977**
TOTAL - Time Median (s)	**26.656**	**6.210**
1 bottles - Recall (10°10cm) (%)	20.0	20.0
1 bottles - Recall (5°5cm) (%)	0.0	0.0
1 bottles - TE Mean/Median (mm)	97.6 / 97.6	36.7 / 36.7
1 bottles - RE_SA Mean/Median (°)	7.6 / 7.6	2.0 / 2.0
1 bottles - Time Mean/Median (s)	26.668 / 26.370	2.692 / 1.825
2 bottles - Recall (10°10cm) (%)	57.1	28.6
2 bottles - Recall (5°5cm) (%)	0.0	0.0
2 bottles - TE Mean/Median (mm)	17.4 / 17.7	14.1 / 14.1
2 bottles - RE_SA Mean/Median (°)	7.2 / 7.1	9.3 / 9.3
2 bottles - Time Mean/Median (s)	26.885 / 26.789	6.210 / 6.080
3 bottles - Recall (10°10cm) (%)	29.4	17.6
3 bottles - Recall (5°5cm) (%)	0.0	0.0
3 bottles - TE Mean/Median (mm)	17.5 / 17.2	22.5 / 17.0
3 bottles - RE_SA Mean/Median (°)	8.4 / 8.7	8.6 / 8.9
3 bottles - Time Mean/Median (s)	26.656 / 26.645	7.751 / 8.457
4 bottles - Recall (10°10cm) (%)	5.3	26.3
4 bottles - Recall (5°5cm) (%)	0.0	0.0
4 bottles - TE Mean/Median (mm)	37.2 / 37.2	22.8 / 10.9
4 bottles - RE_SA Mean/Median (°)	8.7 / 8.7	4.6 / 4.8
4 bottles - Time Mean/Median (s)	26.458 / 26.362	7.831 / 8.514
5 bottles - Recall (10°10cm) (%)	10.0	20.0
5 bottles - Recall (5°5cm) (%)	0.0	0.0
5 bottles - TE Mean/Median (mm)	14.3 / 14.3	23.8 / 23.8
5 bottles - RE_SA Mean/Median (°)	6.0 / 6.0	4.0 / 4.0
5 bottles - Time Mean/Median (s)	27.111 / 26.911	6.730 / 6.479
6 bottles - Recall (10°10cm) (%)	0.0	0.0
6 bottles - Recall (5°5cm) (%)	0.0	0.0
6 bottles - Time Mean/Median (s)	26.390 / 26.390	1.815 / 1.815
7 bottles - Recall (10°10cm) (%)	0.0	0.0
7 bottles - Recall (5°5cm) (%)	0.0	0.0
7 bottles - Time Mean/Median (s)	26.455 / 26.455	1.814 / 1.814

Recall variation across clutter levels in the sequential protocol (*Figure 21*) shows that FoundationPose peaked at 57.1% Recall with 2 bottles remaining but dropped to 5.3% with 4 bottles. SuperGlue maintained more consistent performance (17.6–28.6% for 2–5 bottles) but never reached FoundationPose's peak. Both methods failed completely (0.0% Recall) with 6–7 bottles remaining, indicating fundamental limitations in detecting objects under these specific sequential conditions.

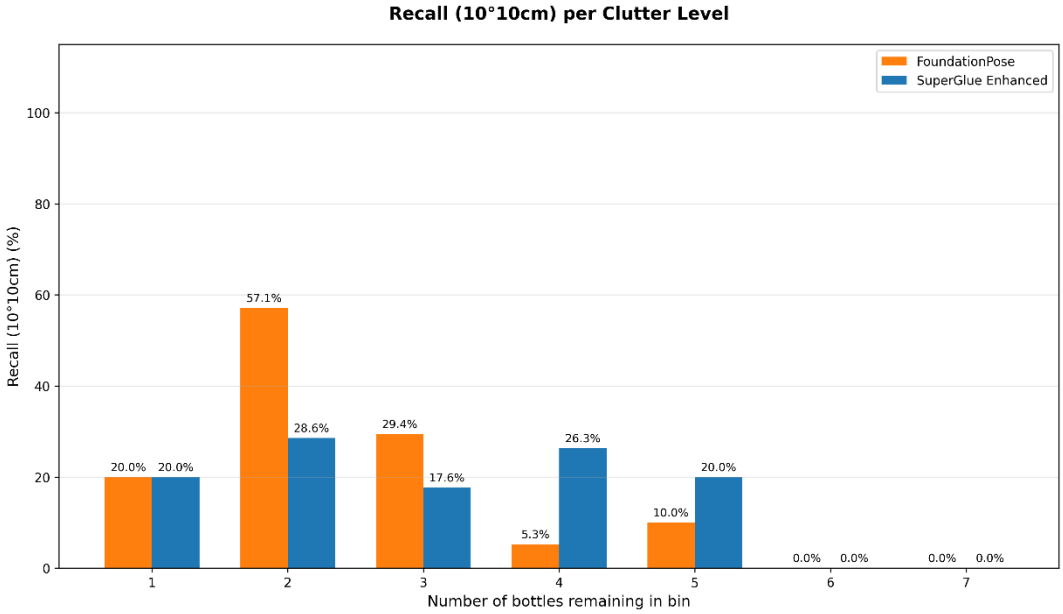


Figure 21

The Symmetry-Aware Rotation Error per clutter level (*Figure 22*) shows that both methods remain well within the 10° threshold for successful detections across all clutter levels. FoundationPose's median errors range from 2.0° (1 bottle) to 8.7° (3 bottles), while SuperGlue's range from 2.0° (1 bottle) to 9.3° (2 bottles). The 5° threshold line intersects several boxplots, indicating that a substantial portion of successful detections meet the stricter criterion.

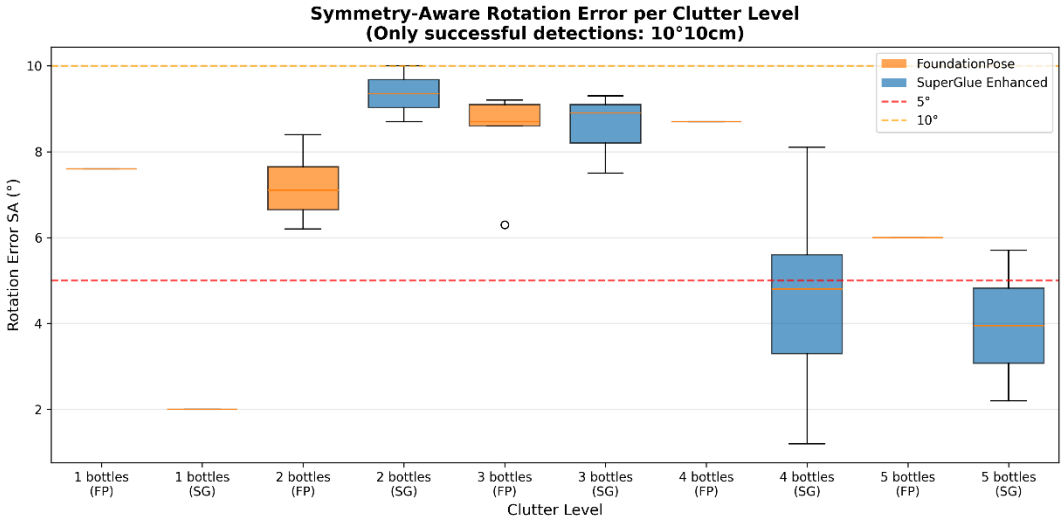


Figure 22

The Translation Error distribution per clutter level (*Figure 23*) reveals that FoundationPose maintains low errors at 2 bottles (median 17.7 mm) and 3 bottles (median 17.2 mm), but exhibits high variance at 1 bottle (median 97.6 mm, single observation). SuperGlue shows more consistent performance across clutter levels, with medians ranging

from 10.9 mm (4 bottles) to 36.7 mm (1 bottle). Both methods produce errors well within the 100 mm threshold for the majority of successful detections.

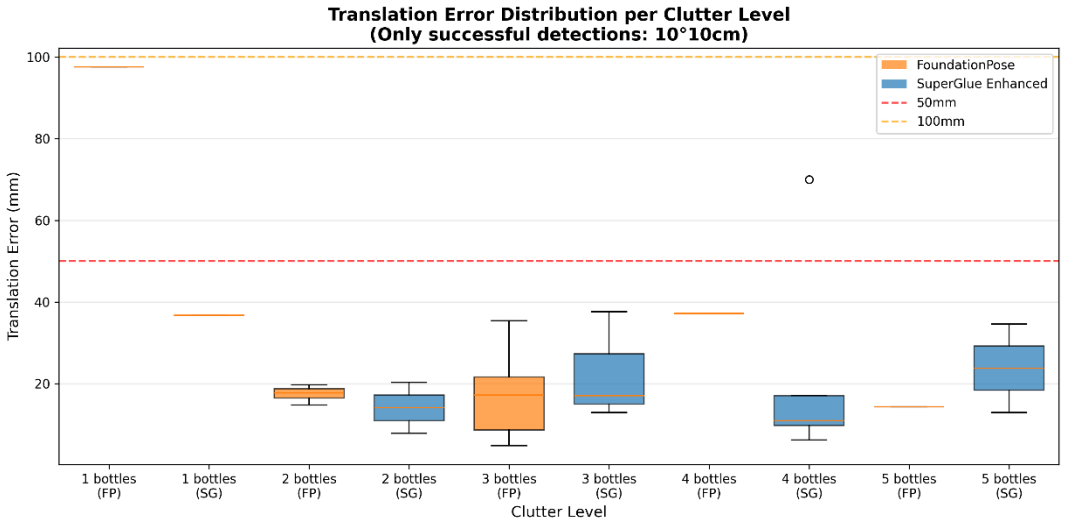


Figure 23

Execution time (*Figure 24*) represented a critical operational trade-off: FoundationPose required a median of 26.656 seconds per evaluation (mean: 26.660 s), while SuperGlue operated at 6.210 seconds median (mean: 4.977 s)—more than 4× faster. This substantial time difference reflects the computational cost of FoundationPose's remote GPU inference and render-and-compare refinement versus SuperGlue's local geometric processing.

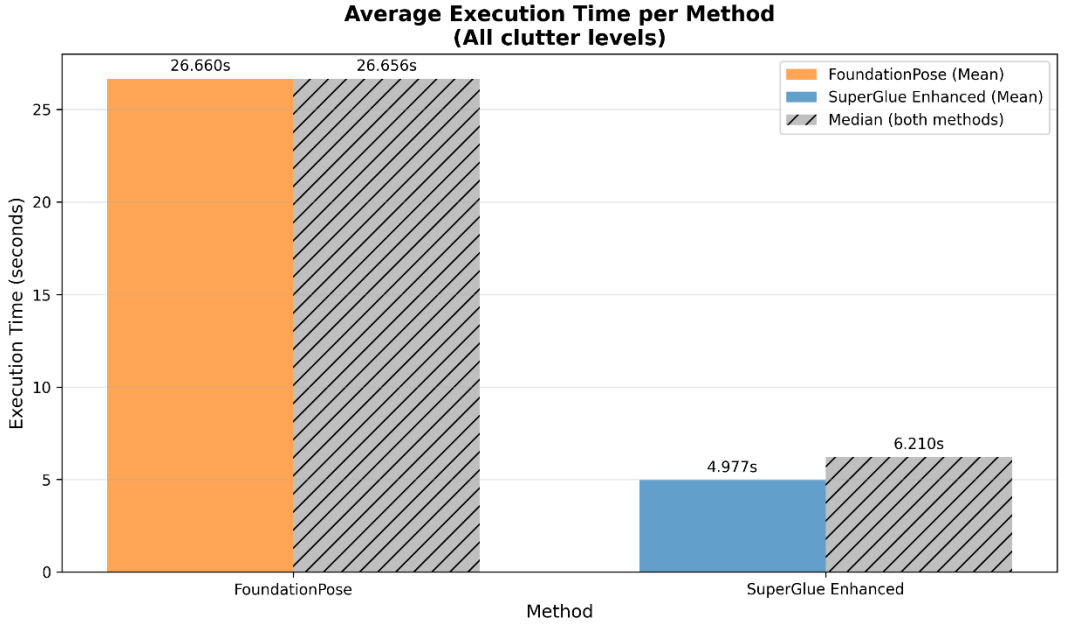


Figure 24

5.3 Discussion and Analysis

Methodological Strengths and Weaknesses in Relation to Literature

The experimental results from the static evaluation demonstrate a clear performance divergence between the two evaluated vision methods, with FoundationPose achieving an overall precision of 43.3% (10°10cm criterion) compared to SuperGlue-Enhanced MultiCue's 21.1%. However, the dynamic sequential experiment revealed substantially lower performance for both methods (~17% recall: 17.4% for FoundationPose and 16.1% for SuperGlue), with detection frequency rather than pose accuracy emerging as the primary bottleneck. This performance gap is particularly pronounced under high-clutter conditions, where FoundationPose reached 60.0% precision at 8 bottles while SuperGlue degraded to 11.8%.

FoundationPose's superiority at high occlusion directly validates its core architectural innovation: the use of an object-centric neural field based on signed distance functions, combined with iterative render-and-compare refinement [13]. Because this mechanism synthesizes novel views from a small set of reference images, it can effectively interpolate plausible geometry from partial observations. This aligns with findings on benchmark datasets like Occluded-LINEMOD and YCB-Video[13], confirming that the method maintains robustness in industrial bin-picking scenarios involving glossy, textureless objects where traditional feature extraction fails.

Conversely, the performance profile of SuperGlue-Enhanced-MultiCue reflects the inherent limitations of its hybrid geometric dependency. The method relies on SuperPoint for keypoint detection [17] and an attentional graph neural network to solve feature matching via optimal transport [18]. While this architecture provides a significant computational advantage, operating in approximately 69 ms per image pair, it is strictly contingent on the availability of visible surface features. Cordeiro et al. [8] note in their review of bin-picking perception that geometric methods complemented by lightweight feature extraction remain viable for specific packaging types but are fundamentally constrained by their reliance on visible geometric primitives. The experimental data confirms this assessment; the hierarchical fallback strategy of the label detection pipeline cannot compensate for the complete occlusion of the white labels in the 8-bottle scenario.

The "Low-Clutter Paradox" and Sensor Limitations

An unexpected finding in the experimental data is the performance degradation observed at low clutter (3 bottles), where FoundationPose drops to 20.0% and SuperGlue-Enhanced MultiCue to 21.7%. Empirical observation confirms that in these sparse scenarios, the bottles consistently settle flat in the bin with the label facing directly toward the camera (a fronto-parallel configuration). Consequently, in this specific top-down experimental setup, the clutter level acts as a confounding variable for object orientation; it is methodologically difficult to isolate a pure 'clutter effect' from an 'orientation effect'. This counterintuitive result reveals fundamental limitations in both data-driven and geometric approaches when confronted with geometrically degenerate configurations and sensor-specific physical constraints.

For FoundationPose, the lack of contextual 3D cues from neighbouring objects removes the occlusion boundaries and depth contrasts that normally anchor the render-and-compare optimization in 3D space. Technically, FoundationPose minimizes a pho-

tometric and geometric loss function between the rendered neural implicit representation (SDF/NeRF) and the observed RGB-D data. In a sparse, fronto-parallel scene, the Jacobian of the loss function with respect to the 6-DoF pose parameters—specifically the gradients for roll, pitch, and Z-translation—becomes extremely shallow. Without the high-frequency spatial gradients provided by adjacent occluding objects, the optimization landscape becomes locally flat, causing the iterative refinement solver to suffer from vanishing gradients and stall in local minima. This corroborates the findings of Cordeiro et al. [8], who demonstrate that purely data-driven pipelines necessitate extensive training datasets or complex domain adaptation, indicating that few-shot adaptation lacks the parallax cues required to constrain the 6-DoF pose in isolation.

For SuperGlue-Enhanced MultiCue, this orientation introduces geometric degeneracy in the 3D-PCA estimation that fundamentally undermines the Singular Value Decomposition (SVD) process. When the label is perfectly perpendicular to the camera, the depth variation across the detected point cloud is minimal. Mathematically, let the centred 3D point cloud be $P \in R^{N\times 3}$. The covariance matrix is computed as $C = P^T P$, which is decomposed via SVD into $C = V\sum V^T$, where $\sum$ contains the eigenvalue $\sigma_1^2 \geq \sigma_2^2 \geq \sigma_3^2$ representing the variance along each principal axis.

In a fronto-parallel configuration, the depth values z_i across the label are nearly constant, causing the variance along the camera's optical axis (σ_3^2) to approach zero. Consequently, the covariance matrix C becomes numerically rank-deficient (rank 2 instead of 3). The condition number of the matrix, defined as $\kappa(C) = \sigma_1^2/\sigma_3^2$, approaches infinity. In this ill-conditioned state, the eigenvector v_3 (which should represent the bottle's surface normal/Z-axis) becomes highly sensitive to infinitesimal perturbations.

These perturbations are directly dictated by the physical hardware architecture of the Intel RealSense D435. According to the D400 Series Datasheet [20] and the noise modeling by Ahn et al. [9], the D435 utilizes a 50 mm stereo baseline and an 850 nm active infrared projector to assist Semi-Global Matching (SGM) on textureless surfaces. In standard stereo triangulation, the depth resolution scales quadratically with the working distance and inversely with the baseline and focal length. At the experimental working distance of 565 mm, the triangulation angle is inherently narrow, physically bounding the Z-axis precision.

Furthermore, when this active illumination strikes the glossy, clear-coated red plastic at normal incidence, the specular reflection saturates the OmniVision OV9282 global shutter sensors. While the global shutter eliminates motion artifacts, it cannot prevent pixel saturation from the IR burst. This saturation creates depth holes and high-frequency, zero-mean noise in the disparity map. This phenomenon aligns with Cordeiro et al. [8], who note that model-free geometric approaches are prone to estimating low-dimensional results with low precision values for textureless objects, as they rely heavily on distinct geometric primitives that are absent in flat, specular regions. When this sensor noise is injected into the near-zero σ_3^2 eigenvalue of the rank-deficient covariance matrix, it causes the estimated normal vector v_3 to swing arbitrarily within the plane orthogonal to the well-defined axes. This violates the fundamental assumption of PCA that distinct, well-separated principal components exist, leading to erratic Y-axis and Z-axis estimates despite correct X-axis detection.

This phenomenon creates a compound failure mode: the geometric degeneracy amplifies the physical sensor noise, a combination that neither the hierarchical fallback strategy nor the SuperGlue classification can fully compensate for. Cordeiro et al. [8]

emphasize that overcoming such limitations requires hybrid strategies that localize objects through data-driven models and estimate their poses using analytical techniques, thereby combining robustness with adaptability. The SuperGlue-Enhanced MultiCue method was designed precisely as such a hybrid strategy; however, in the extreme low-clutter degeneracy, even the deep-learning classification struggles to find reliable keypoints on a saturated, featureless plane.

This is not unique to this specific experimental setup but reflects a broader challenge in industrial robotics. As Guan et al. [6] highlight, objects in industrial applications frequently exhibit smooth, reflective surfaces with minimal texture information. In such scenarios, methods relying purely on RGB data or distinct visual features become highly ineffective, necessitating robust RGB-D fusion or point-cloud-based strategies to compensate for the lack of discriminative appearance cues. The glossy red plastic of the target bottle perfectly exemplifies this challenge, where specular reflections disrupt both feature matching and active stereo depth projection.

Furthermore, with only three objects per cycle, the cycle-level recall becomes highly sensitive to individual failures; a single missed detection reduces the cycle recall by 33 percentage points, magnifying the statistical impact of these micro-variances. This small-sample sensitivity, combined with the geometric and sensor degeneracy, explains why both methods perform worse in the seemingly easier low-clutter scenario - a finding that underscores Guan et al.'s [6] warning that pose accuracy metrics must be interpreted in the context of environmental complexity, not just object count.

Handling of Rotational Ambiguity

The critical importance of symmetry-aware metrics is illustrated in *Figure 20*. For SuperGlue-Enhanced MultiCue, the integration of SuperPoint and SuperGlue resolves the inherent 180° ambiguity of the PCA-based pose estimation by actively classifying whether the visible label is the house-shaped front or the rectangular back. Consequently, the raw rotation errors for SuperGlue are predominantly below 10°, with only a single outlier (Detection Index 13) exhibiting an error of ~175° where this classification failed.

In contrast, FoundationPose exhibits raw rotation errors clustering around 170–180° for the majority of detections. As described in the FoundationPose methodology [13], the method relies on a render-and-compare strategy driven by a neural implicit representation (SDF). Since the bottle's underlying geometry (the red plastic body) is rotationally symmetric, it is solely the unique visual information on the label that breaks this symmetry. When the label is occluded by other bottles, the algorithm lacks this visual reference. Consequently, the optimization cannot distinguish between the front and back, leading it to occasionally converge to a mathematically flipped pose (180° error). When the symmetry-aware correction is applied, these errors for FoundationPose collapse to the 0–10° range, confirming that the metric is necessary for correctly evaluating the method, as the flipped pose remains physically valid for the robotic grasp.

Coupling Perception and Manipulation

The experimental results highlight the critical importance of linking perception accuracy to downstream manipulation requirements. FoundationPose achieved a median translation error of 17.1 mm, whereas SuperGlue-Enhanced MultiCue exhibited a median error of 33.1 mm. Using a 100 mm threshold (the 10°10cm criterion) to define a

"successful" detection is an established standard metric in BOP research, rather than an absolute physical limit for a successful grasp.

Cordeiro et al. [8] explicitly identify a major knowledge gap in the current literature: pose accuracy is almost exclusively reported via geometric metrics such as ADD and ADD-S, which quantify spatial errors but lack an explicit connection to physical manipulation tasks. This study addresses that gap through the specification-based gripper evaluation detailed in Section 5.1. According to this theoretical evaluation, it is the gripper's mechanical compliance (Criterion C3) that handles the uncertainty from the perception stage. For the recommended underactuated gripper, it is specified that its mechanical compliance absorbs minor approach errors, unlike rigid grippers which lack this capability. However, it is important to emphasize that the evaluation in Section 5.1 is qualitative; no physical grasp tests have been performed to empirically verify exactly how large errors (e.g., 33 mm) can actually be absorbed in practice.

The recommendation of underactuated grippers as the optimal paradigm is directly informed by this perception-manipulation coupling. They manage the tension between Criterion C3 (compliance against vision errors) and Criterion C5 (torsional stiffness for decapping). This validates the literature's emphasis on conducting systematic tolerance analyses across the entire robotic work cell, rather than evaluating perception algorithms in isolation.

Beyond spatial accuracy, the temporal performance introduces a critical operational trade-off. The substantial difference in execution time detailed in Section 5.2 implies that for continuous, high-throughput industrial bin-picking operations, the choice of method cannot be based on peak pose accuracy alone. The significantly faster operation of SuperGlue-Enhanced MultiCue may outweigh FoundationPose's positional superiority in scenarios where overall cycle time and detection frequency are the primary bottlenecks.

Detection Frequency Challenges in Sequential Operations

The dynamic sequential experiment exposed a critical limitation not fully apparent in the static evaluation: both methods exhibited extremely high "Not Detected" rates (82.6% for FP, 83.9% for S-E-M) across the sequential picking workflow. This finding suggests that while both methods can achieve acceptable pose accuracy when detections succeed, their detection mechanisms are highly sensitive to the specific scene configurations encountered during continuous operation. The complete failure of both methods at 6–7 bottles remaining (100% not detected) indicates that the sequential picking process may create scene configurations—such as specific occlusion patterns, lighting interactions, or ArUco marker visibility constraints—that fall outside the operational envelope of both paradigms. This highlights a crucial distinction between static benchmark performance and operational robustness in continuous workflows.

6 Conclusion

This chapter summarizes the main results of the study and draws conclusions based on the research questions formulated in Chapter 1. The results are discussed in relation to the literature, and the validity of the study, future research needs, and generalizability are addressed.

6.1 Conclusions

This study has evaluated the feasibility of vision-based 6-DoF pose estimation for the detection and localization of rigid food packaging in cluttered bin configurations. The experimental results answer the central research question by demonstrating that while modern pose estimation is technically feasible, achieving orientation and translation accuracy sufficient for robotic grasping when a detection occurs, it is not yet operationally viable for continuous industrial deployment. The primary bottleneck is not pose accuracy, but detection reliability (recall) under evolving scene configurations.

Main results

In the static evaluation, FoundationPose demonstrated superior performance with an overall precision of 43.3% (10°10cm criterion) compared to SuperGlue-Enhanced MultiCue's 21.1%. This performance gap was particularly pronounced at high occlusion (8 bottles), where FoundationPose achieved 60.0% precision while SuperGlue dropped to 11.8%. FoundationPose also provided higher positional accuracy with a median translation error of 17.1 mm compared to SuperGlue's 33.1 mm. However, for successful detections, both methods achieved comparable symmetry-aware rotational accuracy (medians of 7.5° and 6.2°, respectively), confirming that for successful detections, both methods achieve pose estimation accuracy within the evaluated 10°10cm criterion.

Dynamic sequential evaluation

The dynamic sequential experiment exposed the critical limitation of both methods in continuous workflows. Overall recall dropped to 17.4% for FoundationPose and 16.1% for SuperGlue, accompanied by high "Not Detected" rates (82.6% and 83.9%, respectively). Furthermore, a critical operational trade-off was identified: FoundationPose's higher peak accuracy requires ~26.7 seconds per cycle, whereas SuperGlue-Enhanced MultiCue operates at ~6.2 seconds (>4× faster). This demonstrates that operational feasibility in continuous workflows depends heavily on throughput and detection reliability, not just peak pose accuracy.

Methodological contribution – SuperGlue-Enhanced MultiCue

The primary scientific and practical contribution of this thesis is the design and implementation of the hybrid method SuperGlue-Enhanced MultiCue. By combining geometric 3D-PCA pose estimation with deep learning-based orientation classification (SuperPoint/SuperGlue), the method resolves the frontal/rear (Z-axis) 180° rotational ambiguity inherent in pure PCA-based methods. Because the sides of the bottle lack

labels and consist of textureless red plastic, SuperGlue cannot classify these sides, meaning the lateral (X-axis) 180° ambiguity remains. However, by classifying the visible label orientation and applying exact geometric corrections, the method offers a resource-efficient and transparent pipeline particularly adapted for environments with limited hardware resources. The implementation of symmetry-aware rotation metrics further validated that the mathematically flipped poses remain physically valid for a gripper, bridging the gap between strict mathematical error metrics and downstream manipulation requirements.

Gripper evaluation

The specification-based evaluation identified adaptive underactuated grippers as the recommended paradigm. These offer an optimal balance between mechanical compensation for pose uncertainty (Criterion C3) and sufficient torsional stiffness for cap removal (Criterion C5). This result directly addresses the identified knowledge gap regarding the coupling between perception accuracy and manipulation requirements.

Experimental validity

All experiments were conducted under realistic, resource-constrained conditions—in a home environment with an Intel RealSense D435 camera mounted on a simple wooden rig. This demonstrates that advanced pose estimation can be evaluated without access to expensive laboratory setups, which is highly relevant for the SME sector. It must be explicitly noted that the RobotStudio simulation was used solely to illustrate the process flow and validate the socket communication and RAPID logic between the vision pipeline and a potential robot controller. It did not validate the actual spatial vision-to-robot pose transfer, which remains a subject for future physical validation.

6.2 Future Work and Research

Several research directions are identified to further develop this work:

Digital twin integration via corner ArUco calibration

In the current implementation, RobotStudio functions solely as an illustrative simulation where synthetic poses with applied noise are sent via socket—not as a real digital twin linked to the physical scene. A fully integrated digital twin would be highly beneficial, as it would allow for the safe testing, debugging, and optimization of robot trajectories and grasp strategies in a virtual environment before executing them on a physical robot. A crucial next step to achieve this is to utilize the corner ArUco marker already placed outside the bin as a common spatial reference between the physical camera and the virtual robot cell. Since the camera is statically mounted (eye-to-hand) and the marker's position in the robot's base coordinate system is known, the camera's extrinsic transformation can be directly calculated using solvePnP, without the need for a more complex, motion-based hand-eye calibration (AX=XB). By detecting this marker in the high-resolution reference image and solving the perspective problem, all pose estimates from the vision pipeline are transformed directly into the robot's base frame. This would enable a true digital twin where the physical bin scene is mirrored in real-time in RobotStudio.

Evaluation of CAD- and mesh-based methods

In this study, traditional model-based methods (such as Point Pair Features, PPF, or strict CAD-dependent template matching) were excluded due to the lack of precise CAD models for the target object, and because attempts to generate a reference mesh via MASt3R failed due to local hardware limitations. Future work should include a comparative evaluation of these methods in scenarios where manufacturer CAD data or high-quality 3D scans are available. This would enable a direct comparison between model-free and data-efficient methods (like FoundationPose and SuperGlue-Enhanced MultiCue) and classical, geometrically exact CAD-based pipelines.

Physical robot validation

Once the digital twin integration above is established, the next step should be physical validation with a robot to quantify how the documented pose errors translate into actual grasp success. This would enable experimental verification of the gripper's tolerance to perception uncertainty under realistic conditions.

Custom bin design for unobstructed grasping

The standard rigid-walled bin used in this study occasionally restricted the natural displacement of surrounding bottles when the gripper attempted a deep insertion, potentially causing jamming or secondary collisions. For future physical robot validation, a custom-made bin featuring elastic mesh or flexible netting on its side walls should be employed. This would allow peripheral objects to yield and shift out of the gripper's path during approach, ensuring that grasp success is evaluated purely on the vision system's pose accuracy and the gripper's mechanics, rather than being hindered by rigid boundary constraints.

Extended object diversity

The evaluation was limited to a single ketchup bottle. To increase generalizability, future work should first evaluate the methods on additional variants of ketchup bottles, differing in size, label design, and material, to test their robustness to variation within the same packaging category. FoundationPose would require new reference images and a new neural implicit representation for each variant (consistent with the few-shot paradigm), while SuperGlue-Enhanced MultiCue would require a new, object-specific hierarchical pipeline optimized for each variant's unique geometric primitives. This controlled expansion should precede any attempts to generalize to fundamentally different packaging types.

Hybrid method development

FoundationPose's superiority at high occlusion and SuperGlue's higher recall suggest the potential for an adaptive hybrid architecture that dynamically switches between methods based on real-time assessment of occlusion level and visible geometric cues.

Optimization of detection pipelines

The dynamic sequential experiment revealed a high "Not Detected" rate (82.6% for FP and 83.9% for S-E-M) across both methods. Given that the 100 mm translation threshold is physically justified by the target object's maximum width (96 mm), this high

failure rate represents a genuine algorithmic limitation rather than an artifact of an overly strict scoring metric. Future work should therefore focus on improving the robustness of the detection mechanisms themselves under changing scene configurations—such as enhancing feature matching resilience to dynamic occlusions or varying lighting—rather than relaxing the physically grounded evaluation criteria.

Standardization of ground truth markers

The manual application of ArUco markers introduces a systematic positional and orientational uncertainty. Future work should investigate standardized, repeatable methods for applying these fiducial markers to minimize placement errors. This could involve the design and use of custom 3D-printed jigs or alignment fixtures that ensure each marker is applied to the exact same location and with identical orientation on every object instance. Minimizing this manual placement variance would lower the ground truth noise floor, allowing for even more precise evaluation of the pose estimation algorithms.

Multi-view strategies

The study used a single top-down camera view, which imposes a fundamental geometric limitation: the sides of the bottles (which lack labels and exhibit textureless red plastic) are largely invisible from this vantage point. This limitation directly contributes to the high "Not Detected" rate (82.6% for FP and 83.9% for S-E-M) observed in the dynamic sequential experiment, where bottles settle in orientations that occlude the label-bearing surfaces from the camera's line of sight.

Future work should investigate multi-view architectures to address this blind spot. Two primary approaches are viable: (1) a multi-camera setup with 2–4 synchronized RGB-D sensors positioned at different angles around the bin, enabling fusion of complementary viewpoints and significantly reducing occlusion-related dropouts; or (2) an eye-in-hand configuration where a single camera is mounted on the robot's end-effector, allowing the robot to actively inspect the scene from multiple poses before grasp execution.

Importantly, multi-view strategies may also resolve the **low-clutter paradox** identified in Section 5.3, where both methods exhibited degraded performance with only 3 bottles. The fronto-parallel configurations prevalent in sparse scenes create geometric degeneracy for PCA-based estimation (SuperGlue) and vanishing gradients for render-and-compare optimization (FoundationPose). A secondary camera positioned at an oblique angle would break this degeneracy by providing the depth variation and parallax cues that are absent in single top-down views, potentially restoring performance in low-clutter scenarios to expected levels. Both strategies would require extending the unified scoring system (Section 4.5) to handle multi-view detection fusion.

Quantification of Algorithmic Stochasticity

Future work should include a controlled experiment where the evaluated methods are executed repeatedly (e.g., 50–100 iterations) on identical, static input data. This would isolate the algorithmic variance from the physical variance, enabling a more rigorous decomposition of the total pose estimation uncertainty. Such an analysis would provide method-specific confidence intervals and validate whether the deterministic

execution claims hold in practice across different GPU architectures and CUDA versions.

6.3 Critical Discussion

During the course of the work, several methodological challenges emerged that affected the execution and results of the study.

Experimental environment

Conducting the experiments in a kitchen with a simple wooden rig and an Intel RealSense D435 camera was both a limitation and a strength. The limitation lay in the absence of controlled laboratory conditions, where lighting could not be controlled in the same way as in a dedicated research environment. The strength was that this demonstrates that advanced vision research can be conducted with minimal resources, which is the core of the RE:WORK project's goal to help SME companies. The manual scenario generation (pouring bottles into the box, shaking, adjusting) introduced variation that is difficult to reproduce exactly, but at the same time reflects realistic industrial conditions better than a perfectly controlled environment.

Development of SuperGlue-Enhanced MultiCue

The most significant methodological work lay in designing and implementing the hybrid method. The main challenge during this work was solving the technical and practical problems that continuously arose during the development and testing of the pipeline. The hierarchical fallback strategy for Y-axis determination (Hough lines → longest edge → fingerprint → largest angle → minAreaRect) required extensive interactive parameter optimization via the developed tuning tools. The exclusion of the RED_BODY detection path from the final evaluation was a necessary methodological choice based on empirical observations of unreliable poses under specular reflections.

RobotStudio simulation

The simulation was used solely to illustrate the process flow and validate the socket communication—not as the primary validation tool. The fact that physical robot validation was not conducted limits the practical validation of the study, but does not affect the scientific validity of the relative method comparison.

Ground truth bias

ArUco-based ground truth introduces a selection bias where only objects with visible markers are included in the evaluation. This is methodologically justifiable within the framework of the sequential picking paradigm (only manipulable objects are relevant), but limits the ability to evaluate the methods' ability to detect deeply occluded objects. The manual application of the markers also introduced a systematic uncertainty of ±3 mm in translation and ±2° in rotation, which, however, constitutes a static noise floor that did not systematically favour any of the evaluated methods.

Translation vs. Rotation Complexity

The observed discrepancy between the tight translation error distributions and the higher variance in rotation estimates is consistent with fundamental challenges in 6DoF pose estimation. Guan et al. [6] emphasize that estimating rotation is inherently more complex and difficult than estimating translation, largely because common rotation representations are discontinuous in 3D Euclidean space, posing significant optimization challenges for both analytical and learning-based networks. This theoretical constraint contextualizes why achieving sub-centimetre translation accuracy is often more attainable than achieving sub-degree rotational precision in unstructured, textureless bin-picking scenarios.

Low-clutter paradox

The unexpected result that both methods performed worse with 3 bottles compared to 5-8 bottles raises questions about the robustness of the experimental design. Although this can be explained by geometric degeneracy phenomena and statistical sensitivity with small samples, it indicates that further validation is needed to rule out systematic error sources in the scenario generation.

Notably, the fronto-parallel configurations prevalent in low-clutter scenarios represent a fundamental geometric degeneracy that could potentially be resolved through multi-view architectures. For FoundationPose, a secondary camera positioned at an oblique angle would restore the depth gradients across the bottle surface that are absent in fronto-parallel top-down views, break the roll/pitch symmetry that causes the optimization Jacobian to flatten, and recreate occlusion-like geometric cues via the bottle's silhouette against the bin boundary—effectively substituting for the contextual 3D information that neighbouring objects provide in high-clutter scenarios. For SuperGlue-Enhanced MultiCue, the oblique view would restore the depth variation required for stable PCA estimation, preventing the covariance matrix from becoming rank-deficient. This suggests that the low-clutter paradox may not be an inherent limitation of the methods themselves, but rather a consequence of the single-view experimental design.

Static vs. Dynamic discrepancy

The significant drop in detection performance from the static evaluation (where Precision at 10°10cm ranged from 21.1% to 43.3%) to the dynamic experiment (where overall Recall dropped to ~16–17%, and to 0.0% at the stricter 5°5cm criterion) highlights a critical methodological insight: static benchmarks, while essential for isolating algorithmic accuracy, fail to capture the compounding variables of continuous operation. The fact that both methods exhibited comparable performance degradation in the dynamic experiment, despite their divergent static performance, suggests that operational robustness is constrained by factors beyond pure pose estimation - particularly detection reliability under evolving scene configurations. This underscores the necessity of including dynamic, sequential protocols in future bin-picking evaluations.

Hardware constraints on 3D reconstruction for geometric pipelines

The exclusion of classical model-based methods (such as PPF combined with RANSAC and ICP) was driven by hardware limitations. An attempt to generate the required reference mesh using the transformer-based MASt3R[19] framework proved infeasible, as its memory requirements exceeded the capacity of the available Google cloud GPUs

(NVIDIA T4 and L4) when processing the target object. Without a valid 3D mesh, geometric pipelines like PPF could not be executed. This highlights a practical bottleneck in resource-constrained evaluations: classical methods strictly depend on high-quality 3D models, and the failure of the reconstruction step precludes the evaluation of the pose estimation step itself, regardless of the algorithm's theoretical merits.

Algorithmic Stochasticity and Determinism Assumptions

The evaluation protocol in Section 3.9 assumes deterministic execution of the vision methods, based on framework documentation and the use of deterministic modes where available. However, this assumption has not been empirically validated through repeated execution on identical inputs. GPU-based neural network inference (particularly via cuDNN) may introduce minor floating-point variations between runs, independent of physical sensor noise. This represents a methodological limitation: while the relative comparison between methods remains valid (since both are subject to the same computational environment), the absolute confidence intervals for pose estimation accuracy cannot be rigorously established without accounting for this potential source of variance.

6.4 Generalization of Results

Object generalization

The results are specific to the evaluated object type—a rigid plastic ketchup bottle with a textureless red surface, a high-contrast white label, and limited rotational symmetry. The conclusions are directly transferable to similar "squeeze bottle" geometries with mixed texture properties, but cannot be generalized to objects with high rotational symmetry (cylindrical objects without distinct cues), deformable materials, or highly specular surfaces.

Method generalization

FoundationPose's superiority at high occlusion is likely generalizable to other textureless objects with available reference images, as the method's neural implicit representation is designed for exactly this challenge. SuperGlue-Enhanced MultiCue's reliance on visible geometric primitives is a fundamental limitation that applies to all hybrid geometric methods.

Context generalization

The results are particularly relevant for the SME sector with limited hardware resources, circular economy applications within the food industry, and bin-picking scenarios with sequential picking logic. The fact that all experiments were conducted in a kitchen with a simple wooden rig and a cost-effective camera demonstrates that advanced vision research does not require expensive laboratory environments. This makes the study's methodology particularly valuable for researchers and integrators with limited resources.

Research contribution

The study combines the development of the hybrid method SuperGlue-Enhanced MultiCue, adapted for resource-constrained environments, with a comprehensive experimental evaluation under realistic, non-laboratory conditions. The documented two-photo strategy, interactive parameter optimization, dual-protocol evaluation framework (static and dynamic sequential), and JSON-based logging constitute a reproducible methodology that can be applied in similar contexts.

In summary, the results of the study are most generalizable to similar industrial bin-picking applications within the food sector, where mixed-texture packaging (e.g., smooth surfaces with distinct labels), limited resources, and demands for circular material handling coexist. The methodological contribution, the development of SuperGlue-Enhanced MultiCue and the resource-efficient experimental strategy, constitutes the main scientific value of the study.

7 References

[1] United Nations Climate Change, 'Food loss and waste account for 8-10% of annual global greenhouse gas emissions; cost USD 1 trillion annually | UNFCCC'. Accessed: Feb. 22, 2026. [Online]. Available: https://unfccc.int/news/food-loss-and-waste-account-for-8-10-of-annual-global-greenhouse-gas-emissions-cost-usd-1-trillion

[2] Sveriges riksdag, *Avfallsförordning (2020:614) | Sveriges riksdag*. Sweden, 2020. Accessed: Apr. 14, 2026. [Online]. Available: https://www.riksdagen.se/sv/dokument-och-lagar/dokument/svensk-forfattningssamling/avfallsforordning-2020614_sfs-2020-614/

[3] A. Grau, M. Indri, L. Lo Bello, and T. Sauter, 'Robots in Industry: The Past, Present, and Future of a Growing Collaboration with Humans', *IEEE Industrial Electronics Magazine*, vol. 15, no. 1, pp. 50–61, Mar. 2021, doi: 10.1109/MIE.2020.3008136.

[4] 'RE:WORK: A Call to Action for Circular Food Production and Smarter Automation-Ketchup'.

[5] Sweden Food Arena, 'Innovations-och forskningsagenda för livsmedelssektorn 2023 Framtidens innovativa och hållbara tillväxtmotor', Stockholm, 2023. Accessed: Apr. 13, 2026. [Online]. Available: https://swedenfoodarena.se/wp-content/uploads/Innovations-och-forskningsagenda-2023_Sweden-Food-Arena.pdf

[6] J. Guan, Y. Hao, Q. Wu, S. Li, and Y. Fang, 'A Survey of 6DoF Object Pose Estimation Methods for Different Application Scenarios', Feb. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/s24041076.

[7] I. Enebuse, M. Foo, B. S. K. K. Ibrahim, H. Ahmed, F. Supmak, and O. S. Eyobu, 'A comparative review of hand-eye calibration techniques for vision guided robots', *IEEE Access*, vol. 9, pp. 113143–113155, 2021, doi: 10.1109/ACCESS.2021.3104514.

[8] A. Cordeiro, L. F. Rocha, J. Boaventura-Cunha, D. Figueiredo, and J. P. Souza, 'A review of visual perception for robotic bin-picking', *Rob. Auton. Syst.*, vol. 196, Feb. 2026, doi: 10.1016/j.robot.2025.105236.

[9] M. S. Ahn, H. Chae, D. Noh, H. Nam, and D. Hong, 'Analysis and Noise Modeling of the Intel RealSense D435 for Mobile Robots', *2019 16th International Conference on Ubiquitous Robots, UR 2019*, pp. 707–711, Jun. 2019, doi: 10.1109/URAI.2019.8768489.

[10] M. Kalaitzakis, S. Carroll, A. Ambrosi, C. Whitehead, and N. Vitzilaios, 'Experimental Comparison of Fiducial Markers for Pose Estimation', *2020 International Conference on Unmanned Aircraft Systems, ICUAS 2020*, pp. 781–789, Sep. 2020, doi: 10.1109/ICUAS48674.2020.9213977.

[11] Z. Samadikhoshkho, K. Zareinia, and F. Janabi-Sharifi, 'A Brief Review on Robotic Grippers Classifications', *2019 IEEE Canadian Conference of Electrical and Computer Engineering, CCECE 2019*, May 2019, doi: 10.1109/CCECE.2019.8861780.

[12] B. Tipary and G. Erdős, 'Tolerance analysis for robotic pick-and-place operations', doi: 10.1007/s00170-021-07672-5/Published.

[13] B. Wen, W. Yang, J. Kautz, and S. Birchfield, 'FoundationPose: Unified 6D Pose Estimation and Tracking of Novel Objects', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 17868–17879, 2024, doi: 10.1109/CVPR52733.2024.01692.

[14] J. Zhou, K. Chen, L. Xu, Q. Dou, and J. Qin, 'Deep Fusion Transformer Network with Weighted Vector-Wise Keypoints Voting for Robust 6D Object Pose Estimation', in *Proceedings of the IEEE International Conference on Computer Vision*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 13921–13931. doi: 10.1109/ICCV51070.2023.01284.

[15] W. Yan, Z. Xu, X. Zhou, Q. Su, S. Li, and H. Wu, 'Fast Object Pose Estimation Using Adaptive Threshold for Bin-Picking', *IEEE Access*, vol. 8, pp. 63055–63064, 2020, doi: 10.1109/ACCESS.2020.2983173.

[16] K. Säfsten and M. Gustavsson, *Research Methodology For Engineers and Other Problem-Solvers*, Second Edition. Lund, 2024. Accessed: Apr. 15, 2026. [Online]. Available: https://www.studentlitteratur.se/kurslitteratur/forskningsmetodik-och-vetenskapsteori/vetenskapsteori/research-methodology/

[17] D. Detone, T. Malisiewicz, and A. Rabinovich, 'SuperPoint: Self-supervised interest point detection and description', *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2018-June, pp. 337–349, Dec. 2018, doi: 10.1109/CVPRW.2018.00060.

[18] P. E. Sarlin, D. Detone, T. Malisiewicz, and A. Rabinovich, 'SuperGlue: Learning Feature Matching with Graph Neural Networks', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4937–4946, 2020, doi: 10.1109/CVPR42600.2020.00499.

[19] V. Leroy, Y. Cabon, and J. Revaud, 'Grounding Image Matching in 3D with MASt3R'.

[20] 'Intel ® RealSense TM D400 Series Product Family Datasheet', 2019.

[21] T. Hodaň *et al.*, 'BOP: Benchmark for 6D Object Pose Estimation', 2018. Accessed: Apr. 26, 2026. [Online]. Available: https://openaccess.thecvf.com/content_ECCV_2018/papers/Tomas_Hodan_PESTO_6D_Object_ECCV_2018_paper.pdf

A. Appendix

Intel RealSense D435 Technical Specifications

This appendix documents the key technical parameters of the Intel RealSense D435 camera used in this study. All values are retrieved from the official datasheet [20] or via the RealSense SDK at runtime, are unit-specific and stored in the camera's EEPROM; they were not modified or re-calibrated during the study. Factory calibration was validated via reference-distance test.

Table 7, Intrinsic Parameters (at 1280×720 resolution)

Parameter	Value	Unit	Relevance to Study
Focal length (fx, fy)	~1.93	mm	Determines projection geometry for pose estimation
Principal point (cx, cy)	Image centre (SDK-provided)	pixels	Used in PnP and back-projection calculations
Distortion coefficients	≤1.5% radial/tangential		Justifies minimal undistortion preprocessing
Horizontal FOV	87 ± 3	degrees	Motivated mounting height of ~55 cm for full bin coverage, ensuring the smallest markers (10–15 mm) maintain a sufficient pixel footprint (≥40 px) for robust detection and sub-pixel corner refinement.
Vertical FOV	58 ± 1	degrees	
Shutter type (depth)	Global shutter		Reduces motion artifacts during data capture

Table 8, Extrinsic & System Parameters

Parameter	Value	Unit	Relevance to Study
Stereo baseline	50	mm	Fundamental for depth triangulation; used in T_cam_to_bin calculations
Depth origin offset (X-Y)	17.5	mm	Compensation from tripod mount to optical centre
Depth start point (Z)	-4.2	mm	Reference plane for Z=0; added to measured values for ground truth
Minimum-Z (1280×720)	280	mm	Lower bound for valid depth data; informed camera placement

Table 9, Operational Settings Used in Experiments

Setting	Value	Justification
Resolution (RGB + Depth)	1280×720	Native sensor format; optimal balance of detail and computational cost
Frame rate	6 FPS	Sufficient for static bin; reduces USB bandwidth and thermal load
Laser power	150 mW (manual)	Set to compensate for IR absorption by red plastic surfaces while avoiding saturation.
Exposure / Gain	Locked (manual	Ensured consistent illumination across all experimental runs
USB connection	USB 3.1 Gen 1	Required for simultaneous RGB+Depth at target resolution

Retrieval Method
All intrinsic parameters were retrieved programmatically via the Intel RealSense SDK 2.0 at pipeline initialization:
```python
profile = pipeline.start(config)
intrinsics = profile.get_stream(rs.stream.depth).as_video_stream_profile().get_intrinsics()
# intrinsics contains: width, height, ppx, ppy, fx, fy, coeffs, model
```