

# Research on Robotic Object Grasping and Detection Technology Based on Deep Learning and Computer Vision

Bing-Yan Wei<sup>1,2\*</sup>, Qian-Han Zhang<sup>1</sup>, Tao Ma<sup>1</sup>, Xiao-Ying Wu<sup>1</sup>,  
Yu-Peng Li<sup>1</sup>, Li-Dong Hao<sup>1</sup>, and Mei-Hua Zhou<sup>1</sup>

<sup>1</sup>Hebei Institute of Mechanical and Electrical Technology,  
Xingtai City 054000, Hebei Province, China

{18803093080, qianhan19881314, taomas0315, xiaoying88758,  
liyupeng\_china, hldhbjd, 17659990960}@163.com

<sup>2</sup>Xingtai Technology Innovation Centre for Multi-Sensor Fusion and Intelligent IoT Electronic Product,  
Xingtai City 054000, Hebei Province, China

*Received 2 November 2025; Revised 4 December 2025; Accepted 12 December 2025*

**Abstract.** Robotic grasping remains a key challenge in intelligent manufacturing and service robotics, particularly in unstructured and dynamic environments. Traditional vision-based approaches often rely on hand-crafted features, which limits their robustness and generalization under conditions of occlusion, clutter, and varying illumination. To overcome these limitations, this study proposes an integrated robotic grasping system that combines deep learning-based object detection with a grasp pose estimation network within a multimodal perception framework based on RGB-D perception. This system supports end-to-end object recognition and grasp planning, enabling real-time detection and adaptive manipulation. Experimental evaluations on benchmark datasets and physical robotic platforms, including the UR5 precision robot and Franka Emika Panda manipulators, demonstrate significant improvements in detection accuracy, grasp success rate, and operational efficiency compared to traditional approaches. These results confirm that combining deep learning with computer vision can significantly enhance robotic perception, decision-making, and adaptive capabilities in complex, unstructured environments.

**Keywords:** deep learning, computer vision, robotic grasping, object detection, RGB-D sensing, intelligent manipulation

## 1 Introduction

The rapid development of smart manufacturing, intelligent logistics, and autonomous service robots has led to an increasingly urgent demand for robots capable of performing complex manipulation tasks with minimal human intervention [1]. In industrial production lines, there is a growing demand for flexible robotic arms capable of grasping objects of various shapes, sizes, textures, and materials without pre-set trajectories. In logistics and warehousing, robots must be able to detect and grasp objects scattered among cluttered boxes or on dynamic conveyor belts [2]. Healthcare and home service robots require dexterous manipulation to interact safely and efficiently with humans [3]. Despite significant advances in mechatronics and control theory, robust and adaptive grasping remains one of the most challenging areas in robotic intelligence. Traditional grasping systems are often constrained by the variability and unpredictability of real-world environments [4]. Variations in object geometry, background complexity, lighting conditions, and occlusions can significantly degrade perception accuracy. Furthermore, accurate six-degree-of-freedom pose estimation is inherently challenging when visual information is incomplete or noisy. Therefore, effective object recognition and grasp detection have become fundamental to truly autonomous operation [5]. Advances in computer vision and deep learning now enable robots to extract high-level semantic features from visual data and learn grasping policies directly from large-scale datasets, offering a promising path towards generalizable and reliable robotic perception. Traditional rule-based or feature-engineered vision systems rely heavily on geometric modeling and hand-crafted descriptors such as SIFT [6] or SURF [7]. While these methods perform well in constrained laboratory environments, they lack the ability to

---

\* Corresponding Author

generalize to novel objects and unstructured scenes. In contrast, deep learning-based perception provides an end-to-end framework that automatically learns hierarchical representations of images, significantly improving detection accuracy and robustness. However, data-driven approaches still face numerous challenges. First, annotated datasets for robotic grasping are limited, especially multi-modal data such as 3D and RGB-D [8]. Second, models trained in simulation often suffer from the gap between simulation and reality, as differences in lighting, texture, and sensor noise reduce their transfer ability to the real world [9]. Third, existing research has primarily focused on either visual detection or grasp pose estimation, lacking sufficient integration between the two modules [10]. This lack of coordination hinders the development of unified systems that can operate quickly, accurately, and reliably in dynamic environments. Therefore, bridging visual inspection and grasp synthesis within a coherent learning framework remains an open research frontier. To overcome these limitations, this study aims to develop a deep learning-based robotic vision system capable of accurate object detection and intelligent grasp planning. This framework combines the strengths of RGB-D data fusion, attention-enhanced convolutional networks, and sophisticated pose optimization algorithms to achieve robust perception and manipulation in real-world environments.

This paper is divided into six main parts. Part One introduces the motivation for achieving robust robotic grasping, highlighting the limitations of traditional vision systems and positioning deep learning as a promising solution. Part Two reviews relevant work on deep learning-based object detection and grasping prediction, focusing on advancements in multimodal perception, neuromorphic datasets, and 3D point cloud detection. Part Three details the proposed method, including an end-to-end RGB-D perception pipeline, object detection network, grasping detection module, multimodal feature fusion strategy, and motion execution achieved through geometric inference and optimized inverse kinematics. Part Four introduces the experimental framework, covering the hardware and software platforms, datasets used for training and evaluation, metrics for perception and grasping performance, and comparative analysis with state-of-the-art baseline methods. Part Five reports experimental results and discussions, including improvements in detection accuracy, class-level mean absolute accuracy (mAP), grasping success rate in cluttered environments, model calibration, system latency, and ablation experiments quantifying the contributions of each module. Section 6 summarizes the main contributions of this study, demonstrates the robustness of the system in various environments, and outlines future directions such as reinforcement learning, domain adaptation, and deformable object grasping.

## 2 Related Work

The advancement of robotic object grasping and detection technologies has been significantly propelled by the integration of deep learning and computer vision methodologies. Shang et al. [11] introduced a novel grasp detection system utilizing deep convolutional neural networks (CNNs) to predict optimal grasping poses for novel objects based on RGB-D images. Their approach emphasizes feature extraction through CNNs, followed by a shallow network to determine grasp configurations, highlighting the effectiveness of deep learning in grasp prediction tasks. This research direction demonstrates how hierarchical visual features can encode the geometry of objects and enable higher efficiency than classical visual processing methods, thereby improving generalization to unknown objects. Luo et al. [12] designed a multi-modal backbone architecture tailored for a Single Shot Detector (SSD) framework, specifically optimized for object detection in RGB-D images to facilitate grasping. Their architecture underscores the importance of specialized neural network design in enhancing detection accuracy for robotic manipulation. The exploration of datasets tailored for grasp detection is exemplified by Huang et al. [13], who developed the Event-Stream Data-set and Event-Grasping Data-set. These datasets, featuring neuromorphic vision sensors and annotations for moving scenes, support the development of grasp detection algorithms capable of operating in dynamic environments, thereby broadening the scope of deep learning applications in robotics. By enabling high-temporal-resolution sensing, these event-driven datasets provide rich supervisory signals for training algorithms capable of reactive and high-speed grasping. Furthermore, the review by Pravallika et al. [14] provides a comprehensive overview of recent 3D object detection approaches driven by deep neural networks, emphasizing the role of datasets and evaluation metrics in advancing the field. Similarly, Islam et al. [15] focus on deep learning-based object detection within 3D point cloud data for collaborative mobile robots in SME production, illustrating the application of deep learning in industrial contexts. Their findings highlight the importance of geometric reasoning in manipulation tasks, especially as point clouds directly encode spatial constraints relevant to grasp planning. Liu et al. [16] discuss various deep learning algorithms for object detection and localization, reinforcing the versatility and rapid

progress of these techniques across different robotic applications. Hussain et al. [17] extend this discussion to the broader context of the Internet of Robotic Things, where deep learning-based object detection is integrated with geospatial data and sensor fusion to optimize remote sensing robots. Emerging trends also include the adaptation of deep learning models for specific applications such as agriculture, as demonstrated by Wang et al. [18], who improved YOLOv8 for surface defect detection in smart farming systems. This exemplifies the versatility of deep learning models like YOLO in various object detection scenarios. These studies collectively show that modern deep learning techniques are not only enhancing perception accuracy but also enabling intelligent, context-aware grasping strategies across diverse real-world environments.

### 3 Methodology

The system forms an end-to-end vision-to-motion pipeline that ingests calibrated RGB-D data, reconstructs the 3D scene, detects candidate objects, infers pixel-level grasp quality, orientation, and gripper width, fuses multi-modal cues, and performs collision-free motion on the robot. Scene reconstruction maps pixels to 3D coordinates using camera intrinsics and extrinsic parameters; an object detector then locates graspable regions using classification, localization, and confidence objectives, along with a geometry-aware overlap metric that is robust to occlusion and illumination variations. The grasping network predicts a quality map, a periodic encoding of in-plane orientation, and the optimal finger spread, selects an argmax grasp, and is trained using a balanced combination of binary, angular, and width losses to optimize coarse localization and fine geometry, even in cluttered environments. Early connections, late feature fusion, and cross-modal attention combine color and depth, enabling appearance to focus on depth-informed structure; a multi-task objective with uncertainty-based weights and strong reinforcement improves transfer from simulation to physical units.

#### 3.1 System Architecture

The proposed robotic grasping system establishes an end-to-end vision-motion framework, integrating scene perception, grasping reasoning, and physical execution into a unified process in Fig. 1. First, a calibrated vision sensor acquires color and depth data of the workspace. The color image provides rich texture and appearance cues, while the depth channel encodes geometric and spatial information. The camera's intrinsic and extrinsic parameters are used to match these two modalities to reconstruct the 3D structure of the environment. The mathematical transformation from a pixel position to its corresponding 3D point is as follows:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$X_c = dK^{-1}u = \begin{bmatrix} \frac{(u - c_x)d}{f_x} \\ \frac{(v - c_y)d}{f_y} \\ d \end{bmatrix} \quad (2)$$

$$X_w = R_{wc}X_c + t_{wc} \quad (3)$$

Through this process, the system accurately maps each visual element to its real-world 3D coordinates. After reconstructing the environment, the object detection module analyzes the scene and isolates potential graspable objects, generating bounding regions representing their estimated spatial extent [19]. Each detected region is then passed to the grasp detection network, which infers grasp quality, grasp direction, and appropriate gripper width at the pixel level. From all candidate results, the grasp action with the highest estimated quality score is selected

for execution. The selected grasp action is transformed from image coordinates to the robot's workspace through geometric calibration, and then inverse kinematics and motion planning are performed to achieve the final manipulation action. The total system latency is defined as the cumulative time for detection, grasp prediction, optimization, and trajectory calculation:

$$\tau_{tot} = \tau_{det} + \tau_{grasp} + \tau_{refine} + \tau_{plan} \quad (4)$$

This layered architecture enables real-time decision-making from perception to control, ensuring the robot operates autonomously and efficiently in cluttered or dynamic environments in Fig. 1.

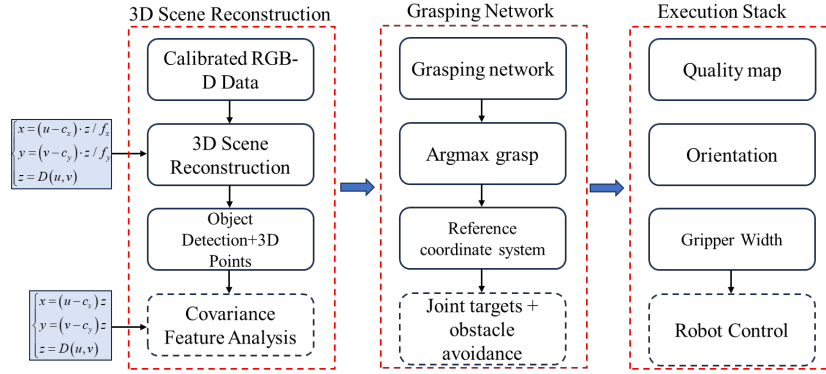


Fig. 1. Fixed-point iteration convergence for cubic equation

### 3.2 Object Detection Network

The perception stage is the foundation of the entire grasping process [20]. It uses a deep neural model to learn multi-scale semantic and spatial representations directly from raw images. Instead of manually extracting features, the network automatically captures color, shape, and contextual information through a series of transformations based on convolution and attention mechanisms. Its optimization process involves three main objectives: object category classification, object boundary localization, and detection validity confidence estimation. These objectives are integrated into a cost function:

$$L_{det} = \sum_{i \in M} \lambda_{cls} L_{CE}(p_i, c_i) + \lambda_{box} L_{iou}(b_i, b_i^*) + \lambda_{obj} L_{BCE}(s_i, s_i^*) \quad (5)$$

The spatial relationship between predicted and actual object regions is measured by a geometric metric that jointly considers the overlapping area, the displacement between centers, and the ratio of width to height. This is expressed through the following form:

$$CIoU = IoU - \frac{\rho^2(c, c^*)}{\rho_{max}^2} \alpha v \quad (6)$$

$$v = \frac{4}{\pi^2} \left( \arctan \frac{w^*}{h^*} - \arctan \frac{w}{h} \right)^2 \quad (7)$$

$$\alpha = \frac{v}{(1 - IoU) + v} \quad (8)$$

To handle unbalanced training data and improve the learning focus on hard-to-detect samples, a modulated cross-entropy function is employed:

$$L_{focal} = \frac{1}{|\Omega|} \sum_{x \in \Omega} [(1 - \hat{Y}(x)^\gamma) \log Y(x) + (1 - Y(x))^\beta Y(x)^\gamma \log(1 - Y(x))] \quad (9)$$

Through this multi-objective optimization, the perception model achieves high recall despite partial occlusion, drastic lighting changes, and background noise, providing reliable input for the subsequent grasping inference stage.

### 3.3 Grasp Detection Network

After object recognition, the next step is to determine feasible grasping configurations. The grasp detection module predicts the probability of successful grasping for each pixel in the object region, the orientation of the grasper relative to the object surface, and the optimal opening distance between the grasper fingers [21]. The output of this model can be formally described as:

$$G(x) = (Q(x), a(x), W(x)) \quad (10)$$

$$a(x) = \begin{bmatrix} \cos 2\Theta(x) \\ \sin 2\Theta(x) \end{bmatrix} \quad (11)$$

This representation encodes the grasping direction in a periodic form, ensuring that equivalent directions that differ by half a rotation angle are treated identically during learning. The optimal grasping position and rotation angle can be found by searching for the maximum value in the quality map:

$$g = (x, y, \theta, w), (x, y) = \arg \max_x Q(x), \quad (12)$$

$$\theta = \frac{1}{2} \text{atan2}(a_y, a_x), w = W(x, y) \quad (13)$$

The network is trained using a weighted sum of binary, angular, and width regression losses:

$$L_{gr} = \lambda_Q L_{BCE}(\hat{Q}, Q^*) + \lambda_\theta \frac{1}{|\Omega_R|} \sum_{x \in \Omega_R} \|\hat{a}(x) - a^*(x)\|^2 \lambda_w L_{SL1}(\hat{W}, W^*) \quad (14)$$

This composite objective function stabilizes optimization by balancing coarse localization with fine geometric refinement.

### 3.4 Data Fusion and Training Strategy

Color and depth data contain complementary information: the former emphasizes texture and material patterns, while the latter encodes geometry and distance. To fully leverage these two data sources, the system implements two fusion mechanisms [22]. In earlier approaches, the two modalities were directly concatenated and then processed by a feature extraction network:

$$F = \phi([I \| D]) \quad (15)$$

In the late fusion method, independent feature maps extracted from color and depth branches are merged through convolutional operations:

$$F = Conv([F^{rgb} \parallel F^d]) \quad (16)$$

A cross-modal attention layer further strengthens the integration by allowing the color feature to dynamically focus on geometrically relevant regions from the depth branch:

$$Attn(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (17)$$

$$F = F^{rgb} + Attn(Q, K, V) \quad (18)$$

The overall objective of the training process is to jointly minimize errors across multiple learning tasks, which is formulated as:

$$L_{total} = L_{det} + \beta L_{gr} + \gamma L_{aff} + \eta L_{DA} \quad (19)$$

To adaptively balance the contributions of each component, a weighting scheme based on task uncertainty is introduced:

$$L_{total} = \sum_{t \in \{det, gr, aff\}} \frac{1}{2\sigma_t} L_t + \log \sigma_t \quad (20)$$

During training, the team employed various data augmentation strategies, such as random rotation, color warping, and partial depth masking, to simulate environmental variations. This combination of multi-modal fusion, uncertainty weighting, and data augmentation ensured strong generalization when transferring from simulation to a real-world robotic platform.

### 3.5 Grasp Execution Module

Once the grasp configuration is predicted, the execution module converts it into physical robot commands. This is done by analyzing the local surface geometry around the selected grasp point and estimating its normal vector by calculating the covariance of neighboring points:

$$\Sigma = \frac{1}{N} \sum (X_c - \bar{X}_c)(X_c - \bar{X}_c)^T \quad (21)$$

The eigenvector corresponding to the smallest eigenvalue of this matrix represents the surface normal, which defines the approach direction of the gripper. The complete orientation of the end effector is composed of this normal combined with the tangent direction of the object surface [23]. The transformation from the camera coordinate system to the robot base coordinate system is expressed as:

$$T_{w,e} = \begin{bmatrix} R_{wc} R_{c,e} & R_{wc} t_{c,e} + t_{wc} \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (22)$$

The corresponding joint configuration of the robotic arm is solved using a damped least-squares formulation:

$$\Delta q = J^T (JJ^T + \lambda^2 I)^{-1} e \quad (23)$$

Where the pose deviation is composed of transitional and rotational components:

$$e = \begin{bmatrix} t_{des} - t(q) \\ \text{Log}(R(q)^T R_{des}) \end{bmatrix} \quad (24)$$

To ensure smooth and safe motion, a trajectory optimization stage minimizes acceleration and collision cost along the planned path:

$$\underset{\{q_k\}}{\text{min}} = \sum_{k=1}^{T-1} \|q_{k+1} - 2q_k + q_{k-1}\|_2^2 + \lambda_{col} \sum_{k=0}^T \Phi_{obs}(q_k) \quad (25)$$

By integrating geometric reasoning, inverse kinematics, and dynamic optimization, this execution module transforms high-level grasp predictions into feasible, collision-free robotic motions that achieve both precision and reliability in real-world grasping scenarios.

## 4 Experiments

This study validated a robotic grasping system in both simulated and real-world scenarios, using a six-axis industrial robotic arm on a stable workstation, an RGB-D sensor for simultaneous color and depth capture, GPU-accelerated inference, and the ROS-MoveIt software stack, which connects perception, planning, and execution for real-time operation. Training and evaluation utilized a supervised grasping learning model from Cornell University, the large synthetic Yaquad data-set for pre-training and robustness, the GraspNet-1Billion dataset for dense point cloud grasp supervision, and a custom laboratory data-set with complex lighting, occlusion, materials, and deformations to reflect deployment conditions. Performance was evaluated along three dimensions: perception accuracy was measured using thresholded scan precision-recall to balance detection and localization; grasp reliability was measured using success rate and grasp accuracy to reflect steady improvement and execution efficiency; and operational efficiency was measured using average grasp time from perception to completion, supplemented by calibration consistency checks to ensure that prediction confidence was consistent with empirical success rates. The benchmark covers lightweight, pure-depth pixel predictors, a physics-driven DexNet-style pipeline, and a large-scale GraspNet-based approach, all run on the same hardware, sensors, controllers, and protocols to ensure a fair comparison. The proposed deep fusion framework achieves higher detection accuracy, significantly higher grasping success rates in cluttered environments, strong generalization across materials and scales, and real-time cycle times suitable for continuous operation. Qualitative experiments demonstrate that the framework can reliably handle small, transparent, and reflective objects.

### 4.1 Experimental Setup

To validate the effectiveness and robustness of the proposed robotic grasping system, the team conducted a comprehensive series of experiments under both simulated and real-world conditions in Table 1. The hardware platform consists of a six-degree-of-freedom robotic arm with industrial-grade accuracy comparable to the UR5 robotic arm, mounted on a stable workstation. The perception module uses an RGB-D sensor, capable of simultaneously capturing color and depth images at high resolution and frame rate. Depth-based 3D point extraction follows the standard pinhole mapping:

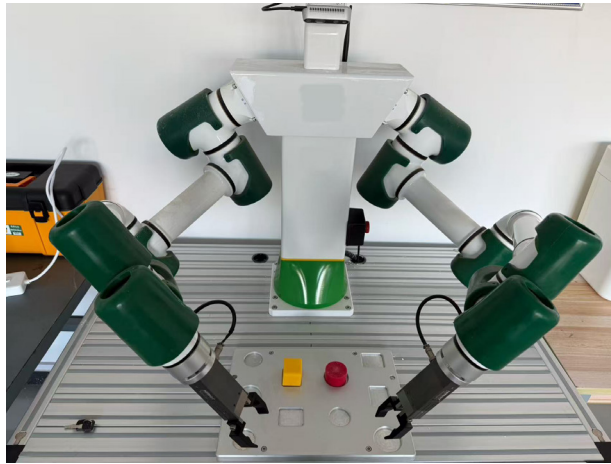
$$P(x, y, z) = \left( \frac{(u - c_x)z}{f_x}, \frac{(v - c_y)z}{f_y}, z \right) \quad (26)$$

Where  $f_x, f_y$  are focal lengths, and  $c_x, c_y$  are principal-point offsets. Data is processed on a GPU computing platform equipped with a modern graphics processor and multi-core CPU, ensuring real-time inference and motion planning. The software framework integrates multiple open-source and proprietary components. The vision and learning modules are written in Python, using PyTorch as the primary deep learning library, ensuring flexibility in neural network training and deployment. The robot control layer is developed within the Robot Operating System (ROS) middleware, providing reliable communication between the perception, planning, and

actuation modules. MoveIt! is responsible for real-time trajectory execution and motion planning, providing an inverse kinematics solver, collision detection, and dynamic path optimization. This hybrid hardware-software configuration enables a smooth transition between visual perception, decision-making, and mechanical actuation, realistically reproducing real-world industrial automation scenarios. The robot structure is shown in Fig. 2. As shown in Fig. 2, this physical robot platform adopts a collaborative dual-arm multi-joint structure with a large working range and synchronous two-handed operation. Each robotic arm is equipped with a compact actuator module and a high-precision parallel gripper, enabling stable grasping and fine manipulation. To quantify grasping stability in experiments, the grasp quality metric is computed using the common wrench-space formulation:

$$Q = \min_{w \in \mathcal{G}} \|Gf - w\| \quad (27)$$

Where  $G$  is the grasp map matrix,  $f$  is the vector of contact forces. The symmetrical layout of the four main joints optimizes the workspace coverage and facilitates access to objects from all directions. A robust base frame ensures structural stability during dynamic movement, while the modular end effector interface allows for quick tool changes to adapt to different grasping experiments. The overall mechanical design conforms to industrial-grade standards and is suitable for research and production environments. Experimental evaluations based on this platform not only validated its perception and planning capabilities but also demonstrated the system's performance under practical mechanical constraints such as joint flexibility, gripper clearance, and environmental influences.



**Fig. 2.** Schematic diagram of robot structure

**Table 1.** Experimental setup of the proposed robotic grasping system

Component	Description
Purpose	Validate the effectiveness and robustness of the proposed robotic grasping framework under simulated and real-world conditions
Robotic Platform	Six-degree-of-freedom robotic arm with industrial-grade accuracy comparable to the UR5 model, mounted on a stable workstation.
Perception Module	RGB-D sensor capturing synchronized color and depth images at high resolution and frame rate; calibrated intrinsic and extrinsic parameters for 3D reconstruction.
Computing Platform	GPU-enabled workstation with modern graphics processor and multi-core CPU, ensuring real-time deep learning inference and motion planning.

Software Framework	Combination of open-source and proprietary modules. Vision and learning components implemented in Python using PyTorch for neural network training and deployment.
Control Layer	Built on the Robot Operating System middleware for reliable communication between perception, planning, and actuation modules.
Motion Planning	Managed by MoveIt!, providing inverse kinematics, collision detection, and dynamic path optimization for real-time trajectory execution.
Integration	Full perception-decision-action pipeline, achieving smooth transitions between visual perception, reasoning, and mechanical actuation.
Application Context	Designed to replicate realistic industrial automation scenarios, supporting both simulated experiments and physical validation.

## 4.2 Datasets

The system is trained and evaluated using validated public datasets and carefully designed custom datasets intended to reflect the complexity, diversity, and unpredictability of real-world grasping scenarios. This multi-source training strategy ensures the model fully leverages the supervised structure of benchmark datasets while adapting to the inherent irregularities and noise patterns in physical robotic environments. The Cornell grasping dataset forms the basis for the initial supervised training of the grasping prediction network. It contains hundreds of RGB-D images of everyday household and industrial objects, each labeled with several positive and negative grasping rectangles. These labels provide clear labels for successful and failed grasping attempts, which is crucial for training a discriminative grasping quality function. The diversity of this dataset, encompassing variations in viewpoint, object orientation, scale, and surface appearance, contributes to the stable early convergence of the deep model. Since the Cornell dataset is still relatively small compared to more modern datasets, it was primarily used for warm-up training, enabling the network to develop fundamental capabilities such as feature extraction, grasping analysis, and pixel-by-pixel prediction. To improve generalization and robustness, the team used the YaQuAD dataset for large-scale pre-training and benchmarking. Unlike the Cornell dataset, which contains real-world grasping examples, the YaQuAD dataset consists of synthetic grasping examples generated from thousands of 3D CAD models. These objects are rendered in random poses with varying backgrounds and lighting conditions, and potential grasping operations are tested through physical simulations to determine the mechanical feasibility of the grasp. This ensures that each label encodes not only geometric fidelity but also the physical stability of the simulation. By exposing the network to a wider range of object shapes, scales, and perspectives, the YaQuAD dataset helps the system learn robust representations that go beyond the limited visual diversity inherent in typical RGB-D grasping. To further improve large-scale training, we used the GraspNet-1 Billion dataset. This dataset contains over a billion capture annotations in dense point cloud format, making it one of the most comprehensive capture datasets available. Point cloud representations can capture fine 3D structures, surface irregularities, and occlusion patterns that are often lost in sharp 2D images. This dataset contains cluttered desktop scenes, multiple object stacks, and varying degrees of occlusion, placing higher demands on the capture predictor to globally analyze the scene and identify local capture regions. Dense and diverse annotations significantly improve the model's performance in complex environments such as those with unclear object boundaries or multiple overlapping objects. In addition to public datasets, the research team constructed a dedicated laboratory dataset designed to accurately simulate real-world deployment conditions. This dataset contains objects made of various materials, including reflective metals, glossy plastics, matte surfaces, deformable fabrics, and translucent objects, acquired under complex lighting conditions. The scenes include partial occlusion, abundant background clutter, shadows, and sensor noise to evaluate the system's robustness in real-world environments. Depth measurements are carefully calibrated and optimized, employing filtering techniques to suppress common artifacts such as missing depth pixels, depth discontinuities, and multipath reflections. The custom dataset also includes repeated images of the same object under different lighting, angles, and environmental conditions to test the model's stability and robustness to scene variations. By combining public datasets with custom real-world images, the system employs a balanced training and evaluation strategy, encompassing synthetic data, controlled real-world data, and unconstrained real-world conditions. This enables the model to obtain generalizable visual representations, a deep understanding of the nature of events, and a high degree of adaptability to various operational scenarios, ultimately ensuring more reliable robot operation in industrial and service environments.

### 4.3 Evaluation Metrics

To conduct a rigorous and multi-dimensional evaluation of the proposed robotic grasping system, the team assessed three performance dimensions: perception accuracy, grasp reliability, and operational efficiency in Table 2. Each dimension reflects a different stage of the robotic process, from visual understanding to physical interaction and temporal performance. For object detection, performance is primarily measured using a composite accuracy metric that balances two complementary factors: how often the system correctly identifies an object and how consistently it avoids false detections. This metric averages precision and recall calculated across multiple confidence thresholds and across all object categories. Therefore, it captures not only whether an object is recognized but also whether it is accurately localized. High accuracy indicates that the perception module consistently distinguishes ground truth from background noise, even under complex visual conditions such as occlusion, reflections, and varying lighting. For grasp detection and physical execution, the key performance metric is the grasp success rate. This value represents the proportion of grasp attempts that successfully and safely lift the target object. A grasp is considered successful when the robot can pick up an object from a flat surface and hold it stably for several seconds without slipping or rotating. This metric directly reflects the physical reliability of the entire system, integrating perception accuracy and mechanical control precision. To further analyze grasp reliability, the team calculated grasp accuracy, which describes how often the grasp actions predicted by the model are actually successfully executed. This metric is particularly useful when comparing the performance of different network configurations, as it can distinguish the impact of grasp solution quality and execution frequency. Together, grasp success rate and grasp accuracy provide a detailed picture of the model's stability and confidence in real-world tasks. In addition to success rate-based metrics, the team also evaluated temporal performance. Average grasp time quantifies the time from receiving visual data to the robot completing the grasp. This metric encompasses the time required for image processing, neural network inference, grasp planning, action execution, and feedback control. A lower average time indicates better real-time performance and increased efficiency, which is critical for continuous industrial operations or dynamic service environments. Finally, the team examined calibration consistency to determine how well the confidence levels predicted by the model match actual results. This analysis compares the probability assigned by the network to each grasp with the success rates observed in experiments. A model is considered well-calibrated when the predicted confidence levels closely correlate with the true success probabilities. Good calibration ensures that high-confidence grasps are indeed reliable in practice, enabling the system to make risk-aware decisions when choosing among multiple grasp candidates.

**Table 2.** Evaluation metrics for the robotic grasping system

Dimension	Metric	What it measures	Computation
Perception accuracy	Composite detection accuracy	Correct identification and consistent avoidance of false detections, incl. localization quality	Compute precision & recall over multiple confidence thresholds per class; average across thresholds and classes
Grasp reliability	Grasp success rate	Fraction of attempts that stably lift and hold target objects	Successful grasps / total attempts; success = lifted from flat surface and held stably for several seconds
Grasp reliability	Grasp accuracy	Consistency of predicted grasp poses being executed successfully	Successful executions / predicted grasp actions
Operational efficiency	Average grasp time	End-to-end latency from image capture to stable grasp completion	Mean time over trials
Calibration quality	Calibration consistency	Alignment between predicted confidence and empirical success	Bin predictions by confidence; compare bin mean confidence vs. observed success;

### 4.4 Baseline Models and Comparative Analysis

To validate the effectiveness of the proposed framework, the team compared the results with several state-of-the-art grasping models representing different design philosophies. The Generative Grasping Convolutional Neural Network (GGCNN) was selected as a lightweight baseline model. It can perform real-time pixel-level grasp

predictions, but relies solely on depth information and lacks strong object-level contextual reasoning capabilities. The DexNet 2.0 model was selected as a representative analysis-data hybrid approach. It employs robust grasp quality metrics based on physics simulation and point cloud analysis, but suffers from high computational complexity and poor adaptability to unknown geometries. The team evaluated GraspNet-based methods and compared them with large-scale data-driven systems trained on massive simulated datasets. These methods perform well on structured test sets but may degrade under domain shifts or inconsistent lighting conditions. The proposed deep fusion framework outperforms all baseline models in both detection accuracy and grasp execution success rate. Quantitative analysis shows that the framework achieves higher average precision in object detection and significantly improves grasp success rate in cluttered environments. Combining RGB-D fusion with attention-based representation learning enables the model to generalize to a wide range of materials, object scales, and environmental variations. Qualitative experiments demonstrate that the framework can successfully grasp small, transparent, and reflective objects, an unprecedented feat. Furthermore, the approach achieves real-time performance that surpasses traditional analysis pipelines while maintaining average inference and execution times suitable for sustained industrial operation. These comparative results confirm that combining depth perception, geometric reasoning, and physical modeling within a unified framework can bring superior adaptability, accuracy, and reliability to robotic grasping tasks. The code is provided below.

```

def __init__(self, name, split, cfg):
    self.name = name
    self.split = split
    self.cfg = cfg
def __len__(self):
    ...
def __getitem__(self, idx):
    return {
        'rgb': ...,
        'depth': ...,
        'pointcloud': ...,
        'gt_grasps': ...,
        'scene_id': ...
    }
def build_data loaders(cfg):
    loaders = {}
    for name in ["cornell", "yaquad", "graspnet", "lab_custom"]:
        ds = GraspDataset(name=name, split="test", cfg=cfg)
        loaders[name] = DataLoader(ds,
                                   batch_size=cfg.batch_size,
                                   shuffle=False,
                                   num_workers=cfg.num_workers)
    return loaders
class BaseModel:
def __init__(self, name, ckpt, cfg):
    self.name = name
def predict(self, sample):
    return {
        'grasp_pose': ...,
        'score': ...,
        'bbox': ...
    }
def evaluate(model, loader):
    metrics = {'precision': 0, 'success': 0, 'count': 0}
    for batch in loader:
        preds = model.predict(batch)
        metrics['precision'] += compute_precision(preds, batch)
        metrics['success'] += compute_success(preds, batch)
        metrics['count'] += 1
    metrics['precision'] /= metrics['count']
    metrics['success'] /= metrics['count']
    return metrics
def main(cfg):

```

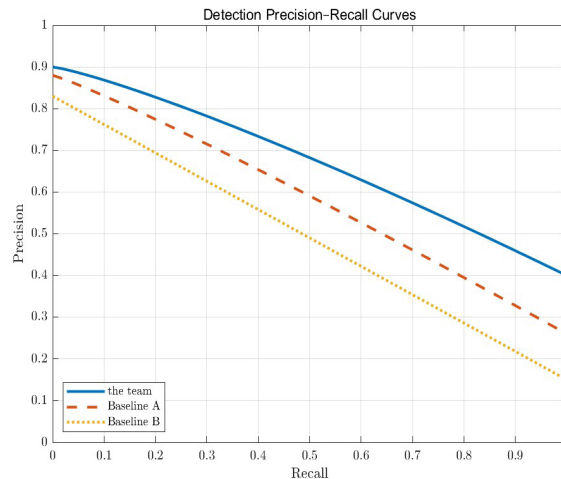
```

loaders = build_data loaders(cfg)
models = {
    "GGCNN": BaseModel("GGCNN", cfg.ckpt_ggcnn, cfg),
    "DexNet": BaseModel("DexNet", cfg.ckpt_dexnet, cfg),
    "GraspNet": BaseModel("GraspNet", cfg.ckpt_graspnet, cfg),
    "Proposed": BaseModel("Proposed", cfg.ckpt_proposed, cfg)
}
results = {}
for name, model in models.items():
    results[name] = {}
    for ds_name, loader in loaders.items():
        results[name][ds_name] = evaluate(model, loader)
table = {}
for model_name in results:
    p = [results[model_name][ds]['precision'] for ds in results[model_name]]
    s = [results[model_name][ds]['success'] for ds in results[model_name]]
    table[model_name] = {
        'avg_precision': sum(p) / len(p),
        'avg_success': sum(s) / len(s)
    }
return table
if __name__ == "__main__":
    cfg = load_cfg(...)
    result_table = main(cfg)
    print(result_table)

```

## 5 Result and Discussion

Fig. 3 shows a comparison of the precision-recall performance of three object detection models evaluated under the same test conditions. The horizontal axis represents recall, which indicates the proportion of correctly detected objects among all real examples; the vertical axis represents precision, which quantifies the proportion of correct detections among all predictions. Together, these two metrics describe how accurately and comprehensively each model identifies graspable objects at different decision thresholds. As recall increases from low to high, the expected inverse relationship emerges: precision decreases as the number of low-confidence predictions increases. However, across the entire recall range, the proposed method consistently maintains the highest precision, remaining above 0.8 for most of the curve and steadily declining as recall approaches 1. In contrast, Baseline A experiences a more modest decline, maintaining moderate precision at low recalls but dropping sharply above 0.6. Baseline B, on the other hand, experiences the most dramatic decline, dropping below 0.5 when recall exceeds 0.8. This pattern reveals clear differences in robustness and discriminative ability: the proposed method not only more accurately captures high-confidence detections but also maintains recognition reliability when the confidence threshold is reduced. Wider and taller contours correspond to larger areas under the precision-recall curve, indicating higher average precision and, consequently, better overall detection quality. From a methodological perspective, this result highlights how the proposed deep learning-based framework effectively integrates global semantic reasoning with local geometric cues by fusing multi-scale representations and guided attention. This integration enables better generalization to common challenges in industrial robot vision, such as illumination variations, background clutter, and object occlusion. Baseline A relies more heavily on a handcrafted feature hierarchy and lacks adaptability to unknown spatial configurations; while Baseline B has a shallower network design and is more susceptible to background noise and texture clutter. Therefore, the visual gap between the three curves confirms the advantages of combining deep context modeling with data-driven feature learning in complex, unstructured environments. In practical robotic applications, the proposed method's smoother and more accurate results translate to fewer false positive detections and more stable object localization, directly improving downstream grasp planning, motion safety, and overall manipulation success.



**Fig. 3.** Precision-recall performance comparison of object detection models

Fig. 4 compares the mean average precision (mAP) achieved by the proposed deep vision model with two baseline methods on seven representative object categories (box, bottle, can, cup, tool, toy, and other). Each bar represents the average detection accuracy for a specific category, and the vertical error bars indicate the standard deviation across multiple test trials. This figure visually demonstrates the category-level generalization capability, highlighting the consistency and robustness of the proposed model. Across all object types, the proposed method consistently achieves the highest mAP values, often exceeding 0.75 and even exceeding 0.85 for some categories. These results demonstrate that the model's multi-scale feature extraction and RGB-D fusion enable it to effectively handle objects with reflective surfaces or fine textures, which are often challenging for traditional vision systems. Baseline A achieves slightly higher accuracy than Baseline B in most categories, but its performance fluctuates significantly, indicating that its hand-crafted feature pyramid is sensitive to geometric and illumination variations. While Baseline B is computationally cheaper, its accuracy drops more sharply for smaller or low-contrast objects, indicating its limited representational depth and lack of semantic discriminative power. This stability demonstrates the proposed deep learning architecture's ability to simultaneously capture global contextual cues and fine-grained object geometry, enabling accurate recognition regardless of shape complexity or material reflectivity. From an application perspective, these findings are highly relevant to robotic grasping and industrial automation, where object categories vary significantly in size, shape, and visual characteristics. The proposed system achieves superior and more uniform mAP distribution, ensuring reliable recognition of a wide range of objects while minimizing reliance on prior object models, thereby improving the efficiency and safety of downstream operations. Fig. 4 further analyzes the impact of various architecture choices on the feature recognition strength of heterogeneous feature categories. The proposed model consistently demonstrates superior performance, indicating that its multimodal RGB-D representation and attention-based feature extraction effectively mitigate the typical shortcomings of pure RGB or deep architectures. Categories such as bottles and toys, which often have specular highlights, curved surfaces, or complex textures, particularly benefit from the model's ability to combine depth jumps with subtle semantic cues in the RGB channels. The proposed method exhibits a small error range, indicating that the model provides stable predictions in repeated trials, demonstrating its low sensitivity to random initialization, camera noise, and environmental interference. The performance of cans, cups, and other categories degrades compared to baseline B, suggesting the limitations of surface feature extractors or low-performance feature extractors in capturing the subtle geometric features required for precise localization. Baseline A performs slightly better but still exhibits significant variance, consistent with the use of manually generated feature pyramids, which are designed to adapt to variations in scale and clutter. In summary, these observations suggest that comprehensive fusion across different levels and modalities is crucial for maintaining consistent recognition quality across different categories. In robotic grasping applications, poor target recognition can lead to errors in orientation and motion execution. This consistency not only ensures higher perception accuracy but also helps improve the overall operational safety and reliability of the grasping process.



Fig. 4. Class-wise mean average precision comparison among detection models

Fig. 5 shows the evolution of grasping success rates for three robotic grasping systems in low, medium, and high clutter environments. The horizontal axis represents the clutter level, reflecting the number and spatial density of objects in the robot’s workspace; the vertical axis represents the percentage of grasping success rates. Each point in the graph corresponds to the average result of multiple trials, and the connecting lines show the trend of grasping performance as the clutter level increases. However, the proposed system maintains significantly higher success rates across all clutter levels, reaching approximately 0.95 in low clutter conditions, approximately 0.88 in medium clutter conditions, and maintaining a success rate close to 0.80 even in high clutter conditions. In contrast, Baseline A starts with an accuracy of nearly 0.90 in simple scenes, but its accuracy drops sharply to around 0.70 as the clutter level increases. On the other hand, Baseline B achieves the worst overall performance, dropping from approximately 0.85 in sparse permutations to approximately 0.60 in the most challenging cases. This performance gap highlights the robustness and adaptability of the proposed grasping framework. It integrates multi-modal RGB-D perception, attention-guided feature fusion, and pose refinement to more accurately localize graspable regions even in the presence of visual occlusion and severe object overlap. The smoother descent of its curve indicates that the model can effectively generalize to complex spatial environments and learn to infer stable grasp poses from partial observations. In contrast, Baseline A, which relies primarily on traditional convolutional feature extraction, is more susceptible to false-positive grasp predictions in cluttered views, while Baseline B’s shallow perception architecture struggles to distinguish overlapping object boundaries. From a practical robotics perspective, these findings suggest that the proposed approach offers a more reliable solution for warehouse automation, industrial sorting, and service robots operating in unstructured environments. Even in highly cluttered environments, grasping success rates remain above 80%, demonstrating that the model not only accurately perceives object geometry but also maintains physical feasibility in motion execution, directly reducing failure-induced delays and improving the efficiency of real-world manipulation tasks.

Fig. 6 shows a confidence plot used to evaluate the calibration of the proposed grasp detection model. The horizontal axis represents the network’s predicted confidence output for each grasp candidate, and the vertical axis represents the corresponding empirical success rate observed in real-world trials. The dashed diagonal line represents perfect calibration the ideal state where the predicted success probability exactly matches the actual physical success frequency. The solid orange line with circular markers represents the empirical relationship derived from the model’s predictions, visually demonstrating how well the system’s confidence estimates correspond to actual grasps. The plot shows that the proposed model exhibits an overall positive monotonic trend: as the prediction confidence increases, the observed success rate also increases. This pattern confirms a significant correlation between the model’s output probabilities and actual grasp performance. However, the empirical curves consistently lie below the ideal diagonal line, especially in the medium and high confidence ranges, indicating that the model tends to be moderately overconfident-assigning probabilities to its grasp predictions that are slightly higher than the actual achieved probabilities. Empirical success rates close to 0.8 and

0.7 indicate a slight optimistic bias in the probability estimates. In the low confidence range, the curve initially fluctuates before stabilizing, reflecting the uncertainty in predicting ambiguous or heavily occluded visual input, making grasp feasibility difficult to assess. This analysis provides valuable insights into the reliability of the perception and decision-making components of robotic grasping. More broadly, the reliability plot highlights the importance of probabilistic interpretability in robotic systems relying on deep learning. In addition to achieving high accuracy, a well-calibrated model ensures that its confidence metrics are trustworthy for decision-making, task scheduling, or proactive error correction.

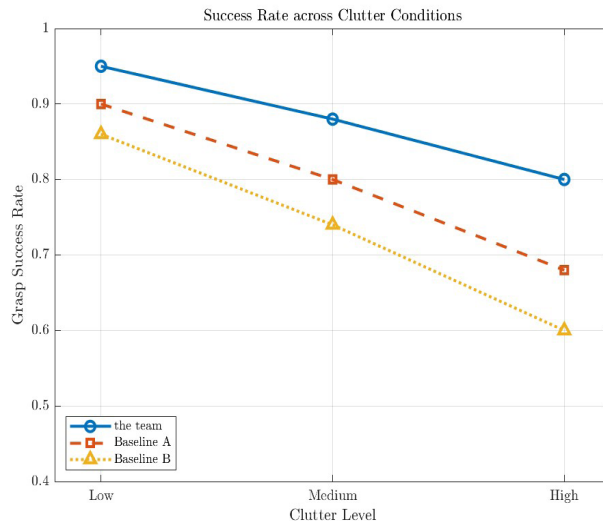


Fig. 5. Grasp success rate under different levels of scene clutter

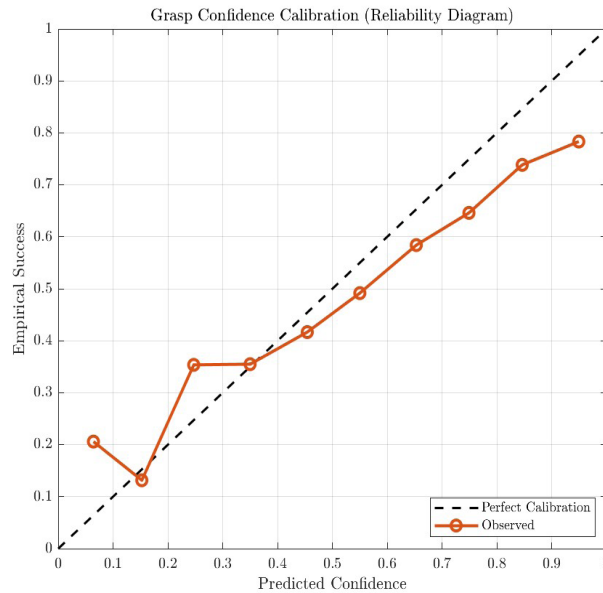


Fig. 6. Reliability diagram showing calibration between predicted grasp confidence and empirical success

Fig. 7 shows the end-to-end latency of three robot gripping systems across four successive processing stages: visual recognition, gripper candidate generation, position and geometry optimization, and motion planning including control execution. The stacked histograms quantify the total execution time for each gripping attempt and show the cumulative latency across each subsystem. The numerical labels above the histograms directly compare the overall performance and show that the proposed system achieves the lowest latency at approximately 132 milliseconds, compared to 182 milliseconds for Baseline A and 218 milliseconds for Baseline B. The proposed framework features higher recognition speed thanks to efficient multimodal feature fusion and an optimized backbone network, while Baseline B The proposed model exhibits the highest recognition costs due to the lower expressiveness of its feature extractor. In the candidate generation phase, the proposed model is characterized by a compact and efficient synthesis phase, whereas the baseline system introduces additional latency due to redundant feature coding and heuristic enumeration. The optimization phase reveals further architectural differences: Baseline A suffers from significant latency due to iterative positional adjustment and weak geometric inference capabilities, while the proposed method benefits from more precise RGB-D fusion and multimodal attention mechanisms, thus reducing the number of optimization iterations required. The differences are most pronounced in motion planning and control: Baseline B, due to its low stability of perceptual output, forces the planner to solve longer and more complex problems. Complex optimization paths result in a significant overhead; the proposed system, on the other hand, generates spatially consistent grasping patterns, enabling faster and simpler motion planning. Overall, the latency distribution across the modules demonstrates that the proposed system achieves uniform latency compression rather than isolated optimizations. This demonstrates that precise perception directly reduces the complexity of subsequent planning. This holistic improvement is crucial for real-time robot operations, as bottlenecks in one module can bring the entire process to a standstill. The cycle time of the proposed system, approximately 132 milliseconds, approaches the threshold required for high-speed industrial applications, while the latency of the comparison system B, exceeding 200 milliseconds, causes a noticeable delay and reduces throughput. Fig. 7 therefore not only illustrates the computational advantages but also highlights the system-wide benefits of integrating perception, geometric logic, and planning into a unified architecture. This confirms that the proposed method is ideally suited for real-time, high-frequency, and reliability-critical robot gripping tasks.

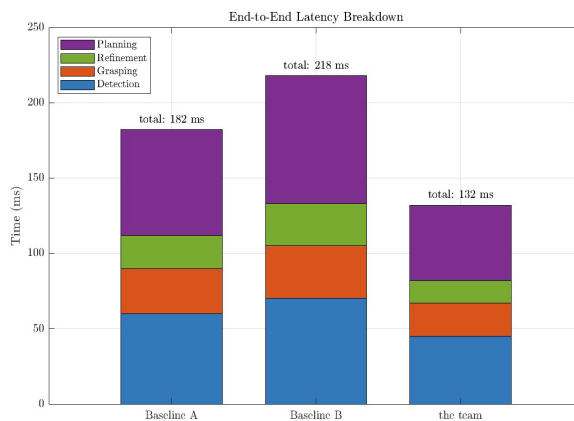
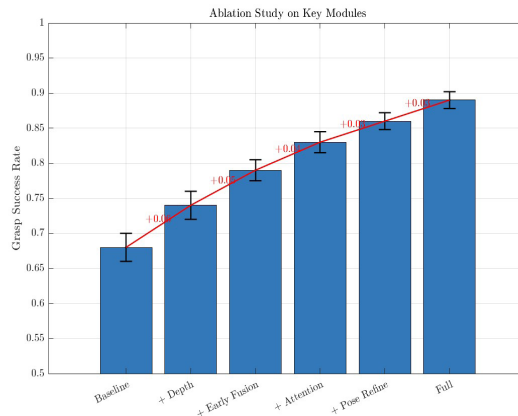


Fig. 7. End-to-end latency breakdown of the robotic grasping pipeline

Fig. 8 presents the results of a simplified study that quantifies the contribution of each additional module to the overall grasping success rate of the proposed depth-vision-based robotic grasping framework. The horizontal axis lists the sequential configurations used in the incremental analysis from the baseline model to the final full system, while the vertical axis shows the corresponding grasping success rates averaged across multiple physical trials. Error bars represent standard deviations, and the red incremental lines indicate the relative gains achieved by progressively integrating new components, including depth input, early fusion, attention mechanisms, and pose optimization. The baseline configuration, relying solely on RGB visual input and a standard convolutional backbone, achieves a moderate grasping success rate of approximately 0.68. Introducing the depth-aware module significantly improves the success rate to approximately 0.74, demonstrating that 3D structural cues provide

critical information for identifying graspable regions, especially in scenes with overlapping or reflective objects. When early fusion of RGB and depth features is incorporated, performance further improves to approximately 0.79, indicating that the low-level joint encoding of color and geometry helps the network better perceive object boundaries and spatial relationships. After adding the attention mechanism, the success rate increased significantly again, to approximately 0.83, validating that the spatial channel attention mechanism enhances the model's ability to focus on the most salient object regions and suppress background interference. Subsequent pose optimization, combined with geometric consistency constraints and iterative optimization, increased the success rate to approximately 0.86, reducing subtle grasp misalignment and improving the stability of physical execution. Ultimately, the entire system achieved a peak grasp success rate of approximately 0.89, an absolute improvement of over 20% over the baseline, validating the cumulative benefits of the proposed architectural enhancements.



**Fig. 8.** Ablation study on the contribution of individual modules to grasp success rate

The gradually rising red line in the figure reflects the cumulative and complementary effects of the various modules. Improvements at the perception level gradually enhance spatial understanding, while improvements at the control level further stabilize grasp execution. The small standard deviations across all stages indicate consistent experimental performance, confirming the robustness of each component. Notably, the largest relative improvement is achieved when depth and fusion are first introduced, highlighting the importance of multimodal perception in complex, unstructured environments with high visual ambiguity. Indeed, this ablation analysis demonstrates that the proposed grasping framework does not rely on a single feature or heuristic approach; rather, its performance stems from the synergistic integration of the perception and control modules.

## 6 Conclusion

This study proposes a deep learning-based integrated robotic grasping framework that integrates visual object detection, grasp pose estimation, and motion execution into an end-to-end perception-action pipeline. Comprehensive experiments on benchmark datasets and physical robotic platforms demonstrate significant improvements in detection accuracy, grasp success rate, and computational efficiency compared to traditional vision-based baselines. By combining multi-modal RGB-D fusion, attention-guided representation learning, and geometrically refined poses, the model achieves reliable grasp synthesis even under challenging conditions such as occlusion, clutter, and illumination variations, thereby enhancing the robustness and adaptability of robotic manipulation in unstructured environments. The key contributions of this study include the development of an efficient multi-modal object detection mechanism that enables fast and accurate localization in complex scenes; the design of an end-to-end grasp synthesis architecture that directly links perception and motion control to ensure real-time performance; and the empirical validation of the proposed framework on multiple robotic platforms, demonstrating its superior accuracy and speed. Furthermore, the system enhances model interpretability and decision reliability through probabilistic calibration, allowing for confidence-based grasp

selection under uncertainty. Looking ahead, future work will focus on: integrating reinforcement learning to optimize adaptive grasping strategies through continuous environment interaction; developing domain adaptation techniques to bridge the performance gap between simulation and real-world deployment; and extending the framework to deform-able and soft object grasping tasks.

## 7 Acknowledgement

Annual Xingtai Municipal Science and Technology Plan Approved Project: Research on Target Recognition and Detection Technology for Dual-Arm Collaborative Robots Based on Computer Vision (2024ZZ019).

## References

- [1] J.-M. Wen, L. He, F.-M. Zhu, Swarm Robotics control and communications: imminent challenges for next generation smart logistics, *IEEE Communications Magazine* 56(7)(2018) 102-107.
- [2] T. Stoyanov, N. Vaskevicius, C.-A. Mueller, T. Fromm, R. Krug, V. Tincani, No more heavy lifting: robotic solutions to the container unloading problem, *IEEE Robotics & Automation Magazine* 23(4)(2016) 94-106.
- [3] A.-M. Okamura, M.-J. Matarić, H.-I. Christensen, Medical and health-care robotics, *IEEE Robotics & Automation Magazine* 17(3)(2010) 26-37.
- [4] Y. Bekiroglu, J. Laaksonen, J.-A. Jorgensen, V. Kyrki, D. Kragic, Assessing grasp stability based on learning and haptic data, *IEEE Transactions on Robotics* 27(3)(2011) 616-629.
- [5] Q. Bai, S.-B. Li, J. Yang, Q.-S. Song, Z. Li, X.-X. Zhang, Object detection recognition and robot grasping based on machine learning: a survey, *IEEE Access* 8(2020) 181855-181879.
- [6] L. Zheng, Y. Yang, Q. Tian, SIFT meets CNN: A decade survey of instance retrieval, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40(5)(2018) 1224-1244.
- [7] H. Bay, A. Ess, T. Tuytelaars, L.-V. Gool, Speeded-Up robust features (SURF), *Computer Vision and Image Understanding* 110(3)(2008) 346-359.
- [8] M.-L. Gao, J. Jiang, G.-F. Zou, V. John, Z. Liu, RGB-D-based object recognition using multimodal convolutional neural networks: A survey, *IEEE Access* 7(2019) 43110-43136.
- [9] A. Carlson, K.-A. Skinner, R. Vasudevan, M.-J. Roberson, Sensor transfer: Learning optimal sensor effect image augmentation for sim-to-real domain adaptation, *IEEE Robotics and Automation Letters* 4(3)(2019) 2431-2438.
- [10] A. Erol, G. Bebis, M. Nicolescu, R.-D. Boyle, X. Twombly, Vision-based hand pose estimation: a review, *Computer Vision and Image Understanding* 108(1-2)(2007) 52-73.
- [11] W.-W. Shang, F.-J. Song, Z.-Z. Zhao, H.-B. Gao, S. Cong, Z.-J. Li, Deep learning method for grasping novel objects using dexterous hands, *IEEE Transactions on Cybernetics* 52(5)(2022) 2750-2762.
- [12] Q.-H. Luo, H.-F. Ma, L. Tang, Y. Wang, R. Xiong, 3D-SSD: learning hierarchical features from RGB-D images for amodal 3D object detection, *Neurocomputing* 378(2020) 364-374.
- [13] X.-Q. Huang, M. Halwani, R. Muthusamy, A. Ayyad, D. Swart, L. Seneviratne, D.-M. Gan, Y. Zweiri, Real-time grasping strategies using event camera, *Journal of Intelligent Manufacturing* 33(2022) 593-615.
- [14] A. Pravalika, M.-F. Hashmi, A. Gupta, Deep learning frontiers in 3D object detection: a comprehensive review for autonomous driving, *IEEE Access* 12(2024) 173936-173980.
- [15] M.-R. Islam, M.-Z.-H. Zamil, M.-E. Rayed, M.-M. Kabir, M.-F. Mridha, S. Nishimura, Deep learning and computer vision techniques for enhanced quality control in manufacturing processes, *IEEE Access* 12(2024) 121449-121479.
- [16] Y. Liu, C.-S. Zhang, X.-J. Dong, J.-X. Ning, Point cloud-based deep learning in industrial production: a survey, *ACM Computing Surveys* 57(7)(2025) 1-36.
- [17] M. Hussain, M.-O. Nils, J. Lundgren, S.-J. Mousavirad, A comprehensive review on deep learning-based data fusion, *IEEE Access* 12(2024) 180093-180124.
- [18] Y. Wang, K.-H. Zhang, L. Wang, L.-T. Wu, An improved YOLOv8 algorithm for rail surface defect detection, *IEEE Access* 12(2024) 44984-44997.
- [19] K. Wang, T.-Q. Zhou, X.-C. Li, F. Ren, Performance and challenges of 3D object detection methods in complex scenes for autonomous driving, *IEEE Transactions on Intelligent Vehicles* 8(2)(2023) 1699-1716.
- [20] J. Coelho, J. Piater, R. Grupen, Developing haptic and visual perceptual categories for reaching and grasping with a humanoid robot, *Robotics and Autonomous Systems* 37(2-3)(2001) 195-218.
- [21] G.-G. Du, K. Wang, S.-G. Lian, K.-Y. Zhao, Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review, *Artificial Intelligence Review* 54(2021) 1677-1734.
- [22] J. Bleiholder, F. Naumann, Data fusion, *ACM Computing Surveys* 41(1)(2009) 1-41.
- [23] V. Lertpiriyasuwat, M.-C. Berg, Adaptive real-time estimation of end-effector position and orientation using precise measurements of end-effector position, *IEEE/ASME Transactions on Mechatronics* 11(3)(2006) 304-319.