EAI Endorsed Transactions
on AI and Robotics

EAI.EU Research Article

CAD-guided 6D pose estimation with deep learning in digital twin for industrial collaborative robot manipulation

Quang Huan Dong[1], The-Thinh Pham[2], Tuan-Khanh Nguyen[1,*], Chi-Cuong Tran[3], Hoang-Huy Tran[1], Tan Do-Duy[4], Khang Hoang Vinh Nguyen[1], Quang-Chien Nguyen[4]

[1]Vietnamese-German University, Vietnam
[2]Can Tho University of Technology, Vietnam
[3]National Taiwan University of Science and Technology, Taiwan
[4]Ho Chi Minh University of Technology and Education, Vietnam

Abstract

6D pose estimation in the bin-picking task has attracted increasing attention from researchers. CAD model-based method have been proposed, demonstrating its effectiveness. However, most existing research relies on point cloud registration from the RGB-D camera, which is often not robust to noise and low-light conditions, leading to degraded point cloud quality and reduced accuracy. Thereby, the method accuracy is significantly affected. Moreover, detecting objects correctly plays a vital role in multiple objects. Supervised deep learning takes consideration into this task, but it typically requires a large amount of labeled data. In industrial environments, sample collection and model retraining are limited. To address these challenges, we introduce the potential approach that integrates the zero-shot learning YOLOE and DEFOM-Stereo model. The YOLOE detects and localizes the object without requiring object-specific training, while DEFOM-Stereo generates point clouds for the CAD model-based pose estimaton. Extensive experiments demonstrate that the proposed approach achieves high accuracy in pose estimation, which is essential for grasp planning and manipulation tasks in robotics. Furthermore, the proposed approach is applied in a Unity3D-based digital twin, enabling enhanced virtual representation of a physical pickup target with an estimated pose. Hence, the research result supports more accurate and responsive digital twins for robotics toward the development of smart manufacturing systems.

Received on 05 July 2025; accepted on 04 September 2025; published on 02 October 2025

Keywords: pose estimation, computer vision, digital twin, industrial robot

doi:10.4108/airo.9676

1. Introduction

Object detection and pose estimation are essential tasks in computer vision, essential for enabling intelligent systems in areas such as robotics and digital twins [1]. Pose estimation involves determining an object's position and orientation in 3D space, which is crucial for physical interaction and manipulation [2]. Traditional methods based on geometric models and manually engineered features have been gradually outperformed by deep learning approaches due to their robustness and scalability [3]. By utilizing convolutional neural networks, deep learning-based object detection frameworks allow real-time performance and high detection accuracy [4]. However, their reliance on predefined object-specifics limits their adaptability in real-world scenarios [5]. Besides, integrating pose estimation with digital twins, the need to be considered is that it enables the creation of virtual representations of physical systems to enable simulation, monitoring, and control in various industrial applications. [6].

Despite the challenges of acquiring perfect CAD models, they remain a crucial component for robust pose estimation, particularly within advanced robotic

*Corresponding author. Email: khanh.nt@vgu.edu.vn

systems [3]. Methods operating without explicit CAD data often struggle with ambiguity, occlusion, and variations in lighting or texture. CAD models provide prior knowledge of an object's geometry, offering a strong constraint that significantly enhances both the accuracy and reliability of pose estimation. This is especially critical in industrial environments demanding high precision and repeatability. Furthermore, integrating CAD data enables more sophisticated approaches, such as model-predictive control and collision avoidance, crucial for safe and efficient human-robot collaboration [3]. Thus, the underlying geometric information represented by a CAD model remains a foundational element for achieving truly intelligent and adaptable robotic systems.

Towards more dependable and performant industrial automation, this work proposes a CAD-guided 6D pose estimation method with deep learning in digital twin for industrial collaborative robot arms. In summary, the primary contributions of this work are:

- A novel 6D pose estimation method leverages CAD models and zero-shot deep learning YOLOE to improve object detection and pose accuracy and adapt the arbitrary objects in industrial scenarios.
- An evaluation of object dimension estimation accuracy is performed using the RealSense D435 camera and the DEFOM-Stereo model, demonstrating the superior performance of the latter.
- The proposed method is implemented within a Unity3D-based digital twin to showcase its ability to generate a virtual representation of physical objects with estimated poses.

The remainder of the paper is structured as follows: Section 2 reviews related work on pose estimation approaches, deep learning techniques, evaluation metrics for object detection, and digital twin technologies; Section 3 outlines the proposed method, focusing on object discrimination alongside view selection, pose estimation based on CAD models, and eye-in-hand calibration; Section 4 presents experimental results, including comparisons of YOLOE models, object dimension estimation methods, pose estimation accuracy, as well as the integration of pose estimation in a Unity3D-based digital twin; and Section 5 concludes the study.

2. Related work

This section reviews key advancements in pose estimation methods, the development of the real-time detecting anything based on YOLO structure, the stereo matching of deep learning-based point clouds registration, and the emerging role of digital twins in representing physical systems virtually.

2.1. Pose estimation approaches

Research in pose estimation can be roughly categorized into three main strategies: feature-driven techniques, pattern alignment techniques, and techniques based on deep learning. Feature-driven techniques that employ 3D data provide robust solutions for recognizing objects [7], whereas pattern alignment techniques, which rely on RGB or RGB-D inputs, estimate an object's position and angle by analyzing information from 2D images. On deep learning approach, both traditional supervised learning techniques [8, 9] and modern deep learning techniques [2] are widely employed in improving object detection and pose estimation.

2.2. The zero-shot learning YOLOE

Several object detection techniques exist, including Faster R-CNN [10], SSD [11], and YOLO (You Only Look Once) [12]. YOLO is recognized for its computational efficiency and robust performance, making it a frequently selected approach for a wide range of applications. As a convolutional neural network, YOLO achieves real-time performance by combining object detection tasks—region identification, feature extraction, and classification—into a single procedure. This streamlined approach enhances performance, making YOLO models suitable for applications requiring quick decision-making.

Since YOLO-v1, the architecture has been gradually refined, with versions like YOLO-v3, YOLO-v5, and YOLO-v8 improving real-time object detection accuracy. The latest version YOLO-v11 [13] strengthens feature extraction by employing an improved backbone and neck architecture, thus, enables more precise object detection.

Existing YOLO models rely on object-specific recognition, which requires collecting large datasets. This approach is not flexible when objects change in dynamic industrial environments. The real-time seeing anything (YOLOE) [5] improves the real-time object detection model and handles limited data by combining a zero-shot architecture based on the YOLO structure. It is specifically designed to detect small and overlapping objects more effectively in complex visual scenes without retraining. YOLOE maintains the efficiency of the original YOLO framework while offering higher precision in object recognition.

2.3. Stereo matching and depth estimation

Among stereo matching and depth estimation techniques, there are PSMNet (Pyramid Stereo Matching Network) [14], GANet (Guided Aggregation Network) [15], and FoundationStereo [16]. PSMNet offers depth estimation by incorporating a pyramid pooling module, which captures contextual information

at multiple scales. GANet allows depth map, particularly in complex scenes, through a context-aware refinement network that leverages contextual relationships between adjacent pixels. FoundationStereo utilizes foundation models to improve stereo matching and depth estimation.

Recently, DEFOM-Stero (zero-shot learning) [17] offers a significant advantage by eliminating the need for task-specific training data, facilitating rapid prototyping and deployment in novel scenarios. This method demonstrates flexibility by readily adapting to new object categories through textual descriptions, enabling generalization beyond its initial training.

2.4. Digital twin

Digital twins are defined as virtual representations of real-world entities [18]. A digital twin enables real-time monitoring, simulation, and optimization. In the case of collaborative robots, it offers a synchronized digital representation of the robot, its working environment, and the objects it interacts with. By combining data from sensors with control inputs, the digital twin enables efficient design, testing, and adjustment of robotic tasks in a virtual environment before they are deployed in the real world. Furthermore, the digital twins facilitates precise navigation, context awareness, and multi-agent coordination [19, 20]. These innovations strengthen real-time system modeling in, e.g., agricultural automation, offering new possibilities in data-driven management and robot-to-robot interaction as well as the ability to operate as an adaptable educational tool [21].

With software that support flexibility and easy updates, digital twins assist the system in functioning better under different conditions [22]. Digital twins, particularly those built in interactive environments like Unity3D, provide real-time simulation and control capabilities, enabling safer testing and faster deployment of industrial robots [23]. The convergence of deep learning, robotics, and digital twins creates a robust framework for advancing autonomous systems and intelligent control. Deep learning enhances perception, decision-making, and adaptability in robotic systems [24]. Therefore, these virtual environments improve operational efficiency, reduce downtime, and support agile automation strategies across manufacturing, service robotics, and healthcare robotics [21].

Recently, a digital twin framework has been developed by the authors for a Universal Robot UR10e as part of a case study on industrial pick-and-place robotics [25]. In this work, a novel CAD-guided 6D pose estimation approach using deep learning is proposed to enhance object detection and visualization within the digital twin framework.

3. Method

The workflow illustrated in Figure 1 employs a step-by-step process to address key tasks. At 3D segmentation stage, captured images are analyzed by YOLOE to segment objects and generate bounding box proposals. These proposals are then refined by Shape Non-Max Suppression (Shape-NMS) method. The images are also processed by DEFOM-Stereo to generate point clouds. The next stage involves object discrimination and view selection to ensure the correctly detected object. This is followed by pose estimation stage. Finally, the position and orientation of the object is transferred to the robot coordinate by the eye-in-hand calibration.

By combining existing models, i.e. YOLOE and DEFOM-Stereo, the proposed approach offers an advancement beyond the capabilities of the individual components. YOLOE efficiently detects and localizes objects in 2D image space, while DEFOM-Stereo estimates dense depth maps from stereo image pairs. When integrated, the 2D detections can be projected into 3D space using the corresponding depth information, enabling accurate, labeled 3D object localization which is crucial for robotic tasks such as grasping and manipulation. This integration leverages YOLOE's real-time semantic recognition and DEFOM-Stereo's precise geometric modeling to detect object identities and positions in the scene. Thus, the system achieves a robust 3D perception capability that allows adaptability to different industrial, e.g. pick-and-place, scenarios without requiring extensive model retraining.

3.1. 3D segmentation

In the 3D segmentation stage, YOLOE identifies and localizes the objects with bounding boxes and its masks by anchor boxes. It needs some the postprocessing steps to filter out the redundant masks and select the most relevant ones. There is a notable step named NMS which serves as a crucial filtering mechanism. To identify overlaps, traditional NMS [26] techniques use bounding box information. Due to their restricted geometric representation, bounding boxes might, however, produce inaccurate overlap estimates when working with non-rectangular objects. Consequently, this study uses a shape of mask-NMS (Shape-NMS) (cf. Figure 2), a selecting method that makes advantage of object forms. Shape-NMS removes duplicates using instance masks instead of bounding boxes and requires no retraining.

After extracting various masks from the pictures, the model feeds the Shape-NMS algorithm to select n masks with the highest scores. The masks list M can be established as a tensor, $M = [m_1, m_2, .., m_3]$. The model returns a one-dimensional array, denoted as $S = [s_1, s_2, .., s_3]$, called the confidence score list S. Each

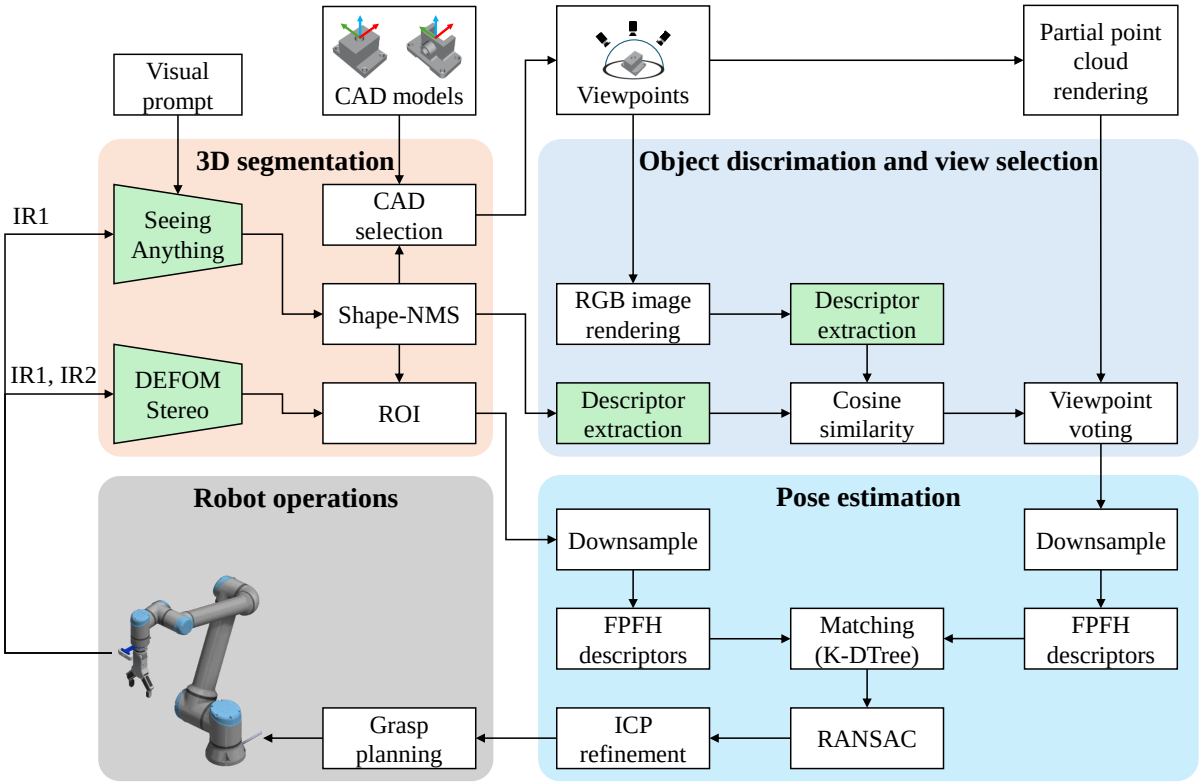


Figure 1. Proposed workflow for vision-guided 6D object pose estimation and grasp planning (green boxes indicate steps that involve deep learning methods).

score S_i is element-wise equivalent to a mask in the list M.

For clarity in describing the algorithm, every mask in the mask list should be represented as m_i, where $i = [1, 2, .., n]$, and its matching confidence score should be s_i. The area covered by mask m_i is denoted $area(m_j)$. Consider another mask m_j in the list (with $j = [1, 2, .., n]$ and $j \neq i$) that is partially adjacent to m_i, with an area of $area(m_j)$. The Intersection over Union (IoU) between two masks is calculate as:

$$\text{IoU}(m_i, m_j) = \frac{\text{area}(m_i \cap m_j)}{\text{area}(m_i \cup m_j)} \quad (1)$$

3.2. Object discrimination and view selection

The "Object discrimination and view selection" stage is presented in Figure 1 and its algorithm is shown in Algorithm 1. This stage focuses on identifying the target object and determining the best viewpoint for voting partial point cloud of the target object. Initially, rendered images (RGB and point clouds) from multiple viewpoints are generated using CAD models. Descriptor extractors (like DinoV2) analyze these images, creating unique feature representations for object recognition. Cosine similarity then compares these features to identify the most suitable viewpoints and the correctly recognized object in the scene. A voting mechanism selects the optimal view, which is then used for pose estimation – determining the object's 3D orientation.

3.3. Pose estimation

Figure 1 illustrates the pose estimation stage, with its algorithm detailed in Algorithm 2. This stage refers to the process of determining the 3D orientation and position of the target object in space. Initially, the received point cloud data from previous stages is downsampled to reduce computational load. The system then leverages feature matching between the observed point cloud and the known CAD model. Specifically, Fast Point Feature Histograms (FPFH) descriptors capture local geometric properties of the point cloud, enabling robust feature correspondence. These descriptors are efficiently searched using a K-DTree algorithm, accelerating the matching process. To handle potential outliers or incorrect matches, a Random Sample Consensus (RANSAC) method is employed, ensuring a robust and accurate pose estimation even with noisy data. Following initial

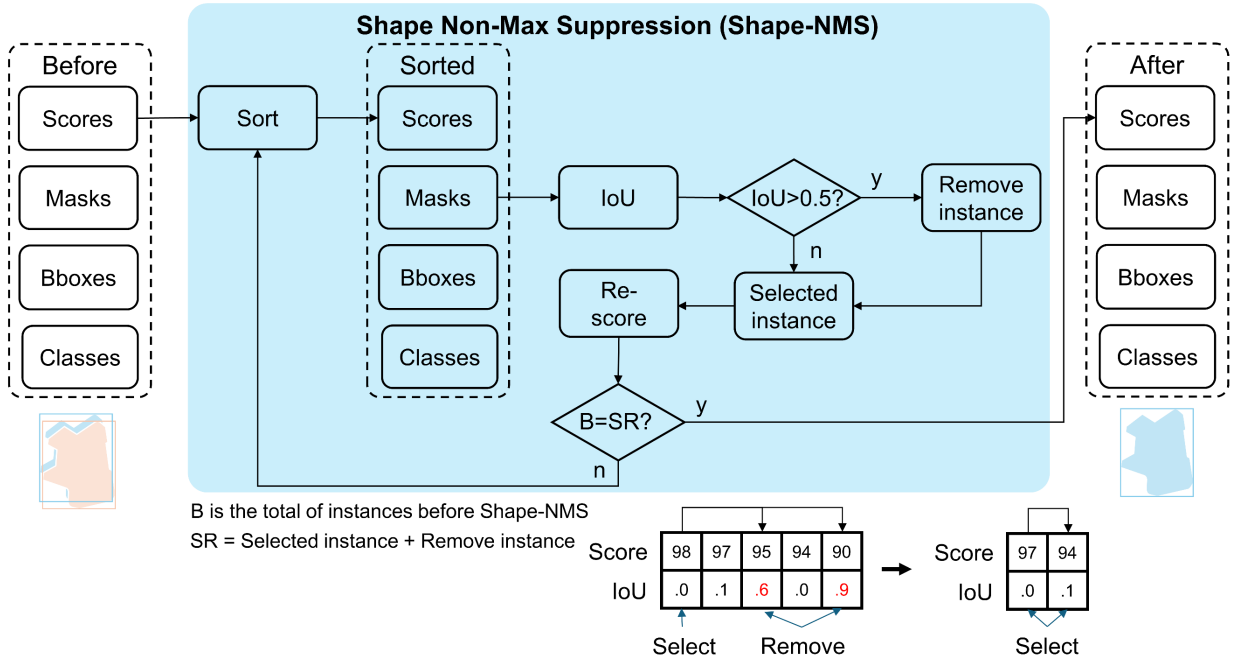


Figure 2. Illustration of the Shape Non-Maximum Suppression (Shape-NMS) process for instance selection and filtering based on score and Intersection over Union (IoU).

matching, an Iterative Closest Point (ICP) algorithm iteratively refines the pose by minimizing the distance between corresponding points. The resulting pose provides essential information for subsequent grasp planning.

3.4. Eye-in-hand calibration

The extrinsic parameters that define the relative position between a camera coordinate and a tool flange coordinate in robotic manipulation are determined via eye-in-hand calibration, as illustrated in Figure 3. An eye-in-hand transformation matrix is computed during this essential calibration process, which enables the robot to precisely observe and communicate with its environment. To provide precise calibration, the calibration object is maintained in an established position and orientation related to the robot base. The main objective is to maintain the target's coordinate system in a fixed position with regard to the robot's base coordinate system, as determined by Equation (2). A checkerboard (target's coordinate) is employed in this work to facilitate the calibration procedure.

$$ {}_{B}T^{W} = \left({}_{T}T^{B}\right)^{-1} \cdot {}_{T}T^{C} \cdot {}_{C}T^{W} , \tag{2} $$

$$ {}_{1}T^{2} = \begin{bmatrix} {}_{1}R^{2} & {}_{1}t^{2} \\ 0 & 1 \end{bmatrix} \in SE(3) , \tag{3} $$

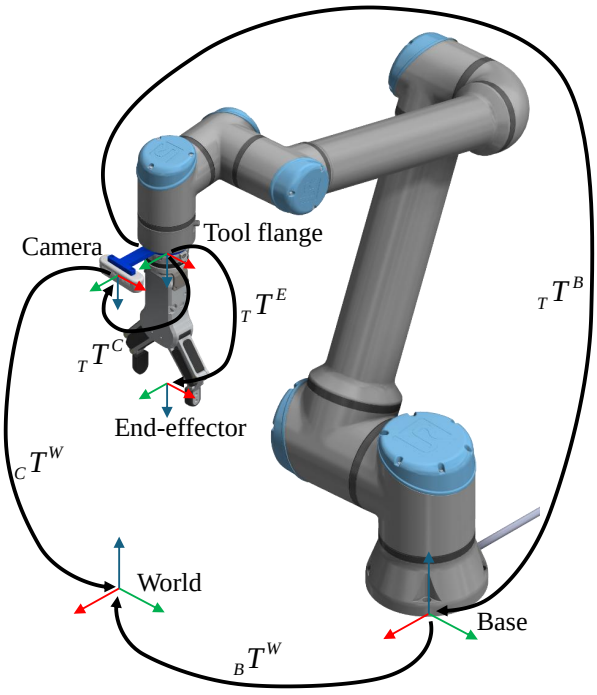


Figure 3. Coordinate the transformation relationship diagram and the system during the eye-in-hand calibration procedure.

where, ${}_{1}R^{2} \in SO(3), {}_{1}t^{2} \in R^{3}$ denotes the transformation matrix which includes a rotation and translation

Algorithm 1 Object Discrimination and View Selection

Require: Selected CAD model, Rendered Images (RGB & Point Clouds) from multiple viewpoints, and Detected Object Image
Ensure: Point Cloud of Selected viewpoint for pose estimation stage
Feature Extraction:
for each Viewpoint i **do**
  Extract RGB image and Point Cloud from selected CAD model via viewpoint i
  Extract descriptors (e.g., DinoV2) from RGB images
end for
Object Discrimination (Similarity Comparison):
Initialize a similarity score array
for each descriptor set from a viewpoint **do**
  Compare descriptor set with the Detected Object Image descriptors from DinoV2 extraction using cosine similarity
  Calculate a similarity score based on the comparison
  Store the similarity score and corresponding viewpoint
end for
Viewpoint Selection (Voting):
Initialize an empty viewpoint vote count array
for each viewpoint **do**
  if similarity score > threshold **then**
    Increment the vote count for that viewpoint
  end if
end for
Select Optimal Viewpoint:
Selected viewpoint = Argmax(viewpoint vote count)
Output:
RETURN Point Cloud of Selected viewpoint

Algorithm 2 CAD model-based 6D Pose Estimation

Require: Point cloud of the detected object in scene, Point Cloud from the selected viewpoint
Ensure: Estimated Pose (Rotation and Translation) of the object
Feature Matching:
Extract Point Cloud from the selected viewpoint using FPFH
Extract Point cloud of the detected object in scene using FPFH
Match Point Cloud features of the detected object to Point Cloud features from the selected viewpoint using a K-DTree for efficient search
Initial Pose Estimation:
Calculate initial rotation and translation using the matched feature correspondences (e.g., using Procrustes analysis or a similar method)
Iterative Refinement (ICP):
Repeat until convergence:
  Find the closest point in the CAD Model for each point in the Point Cloud
  Calculate the error (distance) between corresponding points
  Estimate the optimal transformation (rotation and translation) that minimizes the error
  Apply the transformation to the Point Cloud
EndRepeat
Outlier Rejection (RANSAC):
Apply RANSAC to identify and remove outliers from the matched feature correspondences
Refine the pose estimation using the inlier correspondences
Output:
RETURN Estimated Pose (Rotation and Translation)

for transforming coordinates from {1} to {2}; The robot base coordinate is represented by $\{B\}$, the world coordinate by $\{W\}$, the tool flange coordinate by $\{T\}$, the end-effector coordinate by $\{E\}$, and the camera coordinate by $\{C\}$. The $AX = YB$ problem [27] for N camera views can be used to define

$$ {}_{T}T_{i}^{B} \cdot {}_{B}T^{W} = {}_{T}T^{C} \cdot {}_{C}T_{i}^{W}. \tag{4} $$

Optimization approaches are used to solve the eye-in-hand calibration problems. For instance, this issue may also be directly addressed by global optimization frameworks [28]. The objective function, designed to minimize the difference between the left and right sides of the loop closure Equation (5), is expressed as

$$ \min_{{}_{B}T^{W}, {}_{T}T^{C} \in SE(3)} f = \sum_{i=0}^{N} \left\| {}_{T}T_{i}^{B} \cdot {}_{B}T^{W} - {}_{T}T^{C} \cdot {}_{C}T_{i}^{W} \right\|^{2}. \tag{5} $$

4. Experiment, evaluation and application

The section compares YOLOE object detection models, depth sensing technologies, and pose estimation—including its implementation in a Unity3D digital twin.

4.1. Metrics for evaluating object detection models

Coined by Everingham [29], classification-based metrics are used to evaluate how well object detection models perform. These include True Positives (TP), which are correctly detected objects; False Positives (FP), where the model detects something that is not actually there; and False Negatives (FN), where the

model fails to detect an existing object. Two important evaluation metrics based on these outcomes are Precision (P) and Recall (R). Precision measures how many of the predicted positive detections are correct:

$$P = \frac{TP}{TP + FP} \tag{6}$$

Recall measures how many of the real objects were correctly detected by the model:

$$R = \frac{TP}{TP + FN} \tag{7}$$

To measure how well a model performs across different confidence levels, the Average Precision (AP) metric is used. AP combines precision and recall into a single number for one object class by examining how well the model performs at multiple thresholds. To evaluate overall performance across all object types, the mean Average Precision (mAP) metric is calculated by averaging the AP values for all classes:

$$\text{mAP} = \frac{1}{C} \sum_{c=1}^{C} \text{AP}_c \tag{8}$$

Here, C is the total number of classes. AP is often computed at a fixed Intersection over Union (IoU) threshold (e.g., 0.50, denoted as mAP50). More comprehensive evaluations average AP across multiple IoU thresholds from 0.50 to 0.90 in steps of 0.05, referred to as mAP50-90.

4.2. Performance comparison of YOLOE family models

An analysis on performance of the YOLOE model family is essential for understanding the performance and limitations of each model in different object detection tasks. Table 1 presents a comparative performance analysis of six YOLOE models – ranging from YOLOE-v8s (smallest) to YOLOE-v11l (largest) – across two core object detection tasks: bounding box prediction and instance segmentation (mask prediction). The employed metrics are: P as defined in Equation (6); R as defined in Equation (7); mAP50 and mAP50-90 as defined in Equation (8).

Regarding the results on bounding box detection, YOLOE-v11l demonstrates the highest Precision (0.92) indicating a strong ability to accurately detect objects. YOLOE-v11m has the highest Recall (0.747) indicating the ability in identifying a large number of actual objects in the images. However, YOLOE-v11m's mAP50 (0.762) and mAP50-90 (0.75) are lower than those of other models, suggesting that while it could detect many objects, it may also include more false positives. YOLOE-v8l achieves the highest mAP50 (0.912) and mAP50-90 (0.898), demonstrating a well-balanced performance across different IoU thresholds. This suggests that YOLOE-v8l is capable of accurately detecting objects while maintaining a good balance between Precision and Recall.

For the instance segmentation (mask prediction), the results at mAP50-90 show slightly lower performance compared to bounding box detection. YOLOE-v8l emerges as the strongest performer in this task, achieving the highest mAP50-90 (0.854), demonstrating consistent performance across both tasks.

The model size (indicated by the suffixes 's', 'm', 'l') plays a crucial role. Larger models (like YOLOE-v8l or YOLOE-v11l) generally offer higher accuracy, while smaller models (like YOLOE-v8s) may offer lower accuracy. Therefore, a careful consideration of these trade-offs is essential when selecting the appropriate YOLOE model for a given task.

4.3. Comparison of RealSense camera and DEFOM-Stereo in object dimension estimation accuracy

The experimental setup is illustrated in Figure 4. Specifically, the Intel RealSense D435 camera was mounted at the end of the robotic arm, while the checkerboard was placed on a table. To capture data from various perspectives, the camera was positioned at five different angles.

Figure 5 shows point cloud data captured from two different methods — the Intel RealSense D435 camera and the DEFOM-Stereo method — at five different angles. A point cloud is a group of points in 3D space that represents the shape or surface of

Table 1. Performance comparison of YOLOE model family

Model	Bounding box				Mask			
	P	R	mAP50	mAP50-90	P	R	mAP50	mAP50-90
YOLOE-v8s	0.863	0.663	0.881	0.854	0.863	0.663	0.881	0.802
YOLOE-v8m	0.913	0.74	0.85	0.834	0.913	0.74	0.85	0.796
YOLOE-v8l	0.877	0.718	**0.912**	**0.898**	0.877	0.718	0.912	**0.854**
YOLOE-v11s	0.888	0.707	0.907	0.894	0.888	0.707	0.907	0.848
YOLOE-v11m	0.879	**0.747**	0.762	0.75	0.879	0.747	0.762	0.703
YOLOE-v11l	**0.92**	0.717	0.858	0.842	0.92	0.717	0.858	0.796

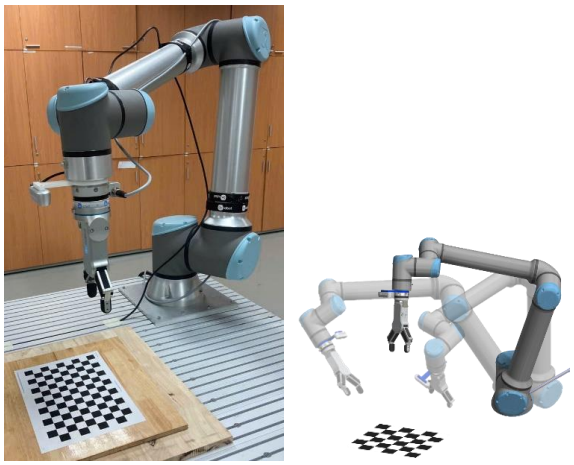


Figure 4. Experimental setup (left) a physical robot and (right) a visual robot at different angles.

an object. The top row presents the point clouds from the RealSense camera. These appear noisy and sparse, with missing parts, especially in darker areas. The bottom row shows the results from the DEFOM-Stereo method. These point clouds are much denser, clearer, and more complete, even in low-light conditions. Thus, the DEFOM-Stereo method seems to provide a more detailed and accurate view of the checkerboard pattern, suggesting that it is less sensitive to noise and could produce more reliable 3D reconstructions.

A comparative analysis of object dimension estimation accuracy between the RealSense camera and DEFOM-Stereo method is presented in Table 2. The results were evaluated based on checkerboard's height and width errors. On overall, DEFOM-Stereo achieves significantly lower values than the RealSense camera on error metrics RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error). The RMSE of 1.19 for DEFOM-Stereo is notably lower than the RealSense camera's 1.54, and the MAE of 0.96 for DEFOM-Stereo is better than the RealSense camera's 1.34. This suggests that the DEFOM-Stereo method could provide more consistent and reliable dimension estimations.

4.4. Pose measurement comparison

This analysis details the performance of the experiment designed to determine the object's location and orientation, which is referred to as its pose. The pose estimations from image processing were compared to the ground truth to evaluate its performance.

The ground truth was obtained by moving the robot end-effector to top of the object which is identified using ChArUco board detection (cf. Figure 6), and the position and orientation of the end-effector displayed on the robot teach pendant were recorded.

The pose estimation results from image processing are presented in Figure 7 which depicts a top-down view of the robotic platform used in this experiment. The black cube-shaped object is positioned on a white, flat surface. Coordinate axes (red, green, blue) are applied on the cube to indicate its position and orientation.

Pose measurement comparison is summarized in Table 3. The performance varied across different poses P1-P5, with the highest position absolute error observed in the pose P4 (3.39 mm for Y) and the highest orientation absolute error observed in the pose P1 (3.52 degrees for Yaw). Rigidly aligning point clouds with planar surfaces or symmetric features is challenging due to geometric ambiguities. These ambiguities often arise because planar or symmetric structures can lead to multiple valid alignment solutions, making it hard to determine the correct transformation (rotation and translation) without additional constraints or information.

In the pose P1, the camera collects the point cloud orthogonal to the object planes, resulting in parallel planes (flat parallel planes head-on). Matching a single plane between two point clouds is under-constrained, as the plane's normal only fixes two rotational degrees of freedom, allowing sliding within the plane and rotation about the normal vector. Adding a second parallel plane helps constrain translation (e.g., distance

Table 2. Comparison of RealSense camera and DEFOM-Stereo in object dimension estimation accuracy

Metric	Method	Angle				
		1	2	3	4	5
Height error	RealSense camera	-1.044	-0.95	1.517	-0.664	1.31
	DEFOM-Stereo	0.25	1.071	0.053	0.281	1.377
Width error	RealSense camera	0.962	0.071	-2.105	1.998	2.81
	DEFOM-Stereo	0.503	-0.949	-0.806	2.33	1.931
RMSE	RealSense camera	1.54				
	DEFOM-Stereo	**1.19**				
MAE	RealSense camera	1.34				
	DEFOM-Stereo	**0.96**				

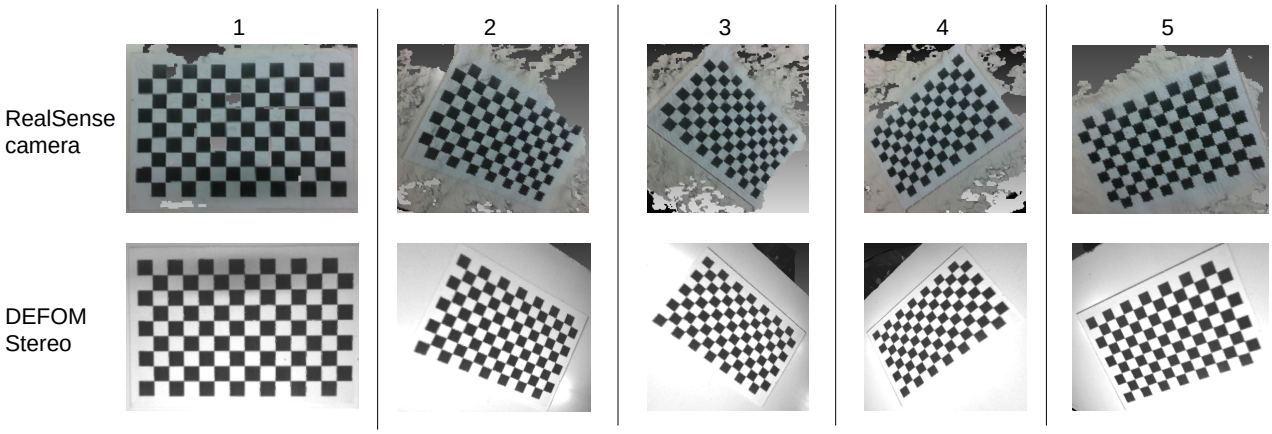


Figure 5. Excerpts of point clouds obtained from RealSense camera and DEFOM-Stereo at five different angles.

between planes) but does not resolve in-plane drift or rotation ambiguities. Introducing a non-parallel second plane further constrains rotation by defining two independent normal directions, reducing rotational ambiguity. However, translation along the planes' intersection line and rotation about it remain under-constrained, leaving the rigid transformation partially ambiguous (as shown in pose P4).

At least three non-parallel, all rotational degrees of freedom are fixed by the 3D basis formed by the three different normal vectors. All translational degrees of freedom are constrained at the planes' crossing point, which defines a distinct location. Six independent constraints are provided by this corner-like structure, three for translation and three for rotation. This makes it possible to determine the rigid transformation precisely and robustly.

The performance was evaluated based on the average of absolute errors between the estimated pose and the ground truth. The results indicate that the system performed well in estimating the object's position, with an average error of approximately 1.5 millimeters across all poses. This suggests that the system is able to determine the object's location in three-dimensional space. The orientation estimation also yielded promising results, with an average error of around 1.4 degrees. However, this level of accuracy needs an improvement for applications where precise positioning is crucial.

Figure 6. Ground truth obtainment for an object with ChArUco marker plane (left) and marker plane with end-effector (right).

4.5. Pose estimation in a digital twin within the Unity3D environment

The proposed pose estimation method could enhance the digital twin for industrial collaborative robot manipulation. The RealSense D435 camera is a common choice for implementing digital twins in robotics, but it lacks integrated object detection capabilities and is sensitive to noise. YOLOE offers effective object detection without the need for retraining, saving valuable time and resources. Comparative analysis using RMSE and MAE metrics, illustrated in Table 2, demonstrates that DEFOM-Stereo yields more accurate dimension estimations compared to D435 camera. In addition, DEFOM-Stereo generates a higher quality point cloud (depicted in Figure 5) compared to the native D435 output. The improved point cloud data is then used to visualize the corresponding detected objects within a Unity3D digital twin environment, thereby enhancing the virtual-physical synchronization of detected objects in manipulation tasks.

To assess synchronization accuracy more comprehensively, the position discrepancy between the tool center point (TCP) of the Unity3D-based digital twin robot and the physical UR10e robot arm is measured and reported in [25] based on values retrieved from kinematic models and UR10e sensors. Three indicators are used to evaluate discrepancies caused by modeling errors and sensor noise: RMSE, MAPE (Mean Absolute Percentage Error), and R2 (coefficient of determination). The reported RMSE ranges from approximately 3 mm to 8 mm for the TCP's X, Y, or Z position; the MAPE ranges from 0.492% to 1.182%; and the R2 ranges

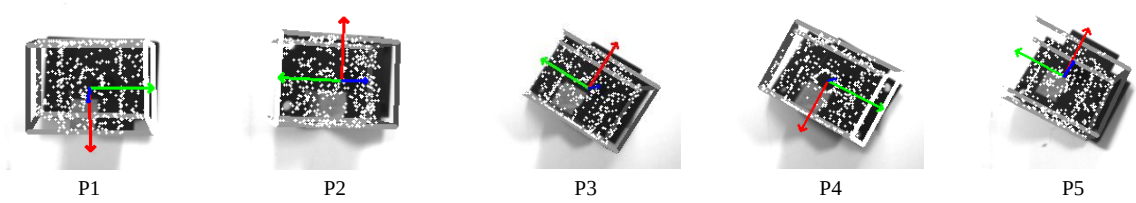


Figure 7. Pose estimation results visualized in the IR images at different poses P1–P5.

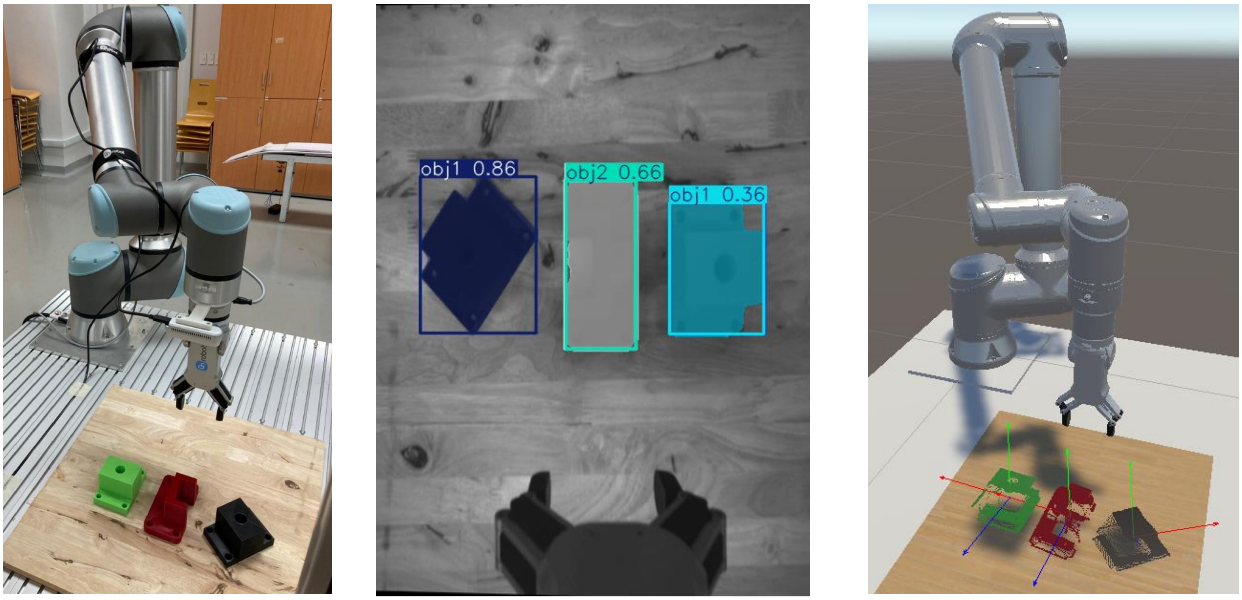
Table 3. Pose measurement comparison between ground truth (reference data) and image processing (information gained from images)

Pose\ Measurement		Position (in mm)			Orientation (in degrees)		
		X	Y	Z	Roll	Pitch	Yaw
P1	Ground truth	433.25	37.03	213.40	178.51	0.32	-4.36
	Image processing	431.51	39.75	213.58	176.64	1.18	-0.84
	Absolute error	1.74	2.72	0.18	1.87	0.86	3.52
P2	Ground truth	480.99	34.30	225.30	162.94	-0.61	178.06
	Image processing	479.80	33.91	223.05	163.02	-4.56	176.32
	Absolute error	1.12	0.39	2.25	0.08	3.95	1.74
P3	Ground truth	474.35	-51.83	220.78	-168.44	-1.37	145.53
	Image processing	473.33	-52.23	219.73	-170.35	-0.37	145.83
	Absolute error	1.02	0.40	1.05	1.91	1.00	0.30
P4	Ground truth	399.58	86.64	222.44	-178.01	-14.92	-40.20
	Image processing	402.95	90.03	221.36	-179.65	-15.33	-39.38
	Absolute error	3.37	3.39	1.08	1.64	0.41	0.82
P5	Ground truth	566.13	32.93	220.82	179.16	-8.85	154.69
	Image processing	566.44	34.81	217.32	179.10	-7.04	154.02
	Absolute error	0.31	1.88	3.50	0.06	1.81	0.67
Average error		**1.51**	**1.76**	**1.61**	**1.11**	**1.61**	**1.41**

from 0.992 to 0.999 for the same axes. The relatively low error values and high R2 indicate a reasonable level of synchronization, suggesting that the digital twin approximates the real robot's positional data. Still, further improvements are required to improve synchronization accuracy of the Unity3D-based digital twin robot.

Thanks to the UR10e built-in real-time controller, the robot's response times are considered fixed [25]. The UR10e is equipped with an integrated controller operating at 500 Hz, which enables it to update or publish its joint state with low latency. In addition, the Unity3D virtual environment can directly subscribe to the joint state publisher to update the virtual robot's joint positions accordingly. Furthermore, the communication is facilitated in a stable and managed network (e.g., a dedicated local wired connection); therefore, the virtual-physical communication latency is considered as a fixed value. Hence, the robot could be able to execute movements smoothly without relying on external control inputs.

The object detection and pose estimation results visualized in a prototypical digital twin within the Unity3D environment is shown in Figure 8. The digital twin in this setup includes the robot arm and the object being manipulated mounted on a fixed table. The object is placed on a wooden base plate to reduce noise for the camera. The employed objects are Obj1 and Obj2, which were 3D-printed from the models of the public dataset available in [30]. The point cloud obtained and pose estimation results are sent from Python program to Unity3D environment by employing a ROS bridge. The results enable manipulating the object in the digital twin environment, which is essential for further work, such as simulating and testing pick-and-place tasks with different object types in different settings before deploying them to production.



(a) Real obj1 (black, green) and obj2 (red)

(b) Object detection with confidence score

(c) Digital twins with pose estimation

Figure 8. Pose estimation in a prototypical digital twin (a: Physical environment; b: Object detection; c: Results visualized in Unity3D).

5. Conclusion

This study explored the integration of the zero-shot learning object detection models with pose estimation techniques based on CAD models, particularly focusing on the YOLOE family models and DEFOM-Stereo model, and their application in digital twin environment for a vision-guided UR10e robot arm. Through experiments, the object detection performance of different YOLOE versions was compared, demonstrating a trade-off between speed and accuracy, where YOLOE-v8l achieved high object detection accuracy while balancing precision and recall effectively. The object dimension estimation capabilities of the RealSense D435 camera and the DEFOM-Stereo method were also evaluated, showing that the DEFOM-Stereo provided more reliable measurements. Additionally, pose measurement accuracy was assessed to validate the effectiveness of the proposed approach. Finally, the presented pose estimation method was incorporated into a digital twin within the Unity3D environment to enhance the virtual representation of physical target object in a pick-and-place scenario. Overall, the findings suggest that combining YOLOE-based object detection with precise pose estimation and calibration could significantly improve the accuracy and usability of digital twins, supporting their application in robotics, simulation, and smart manufacturing. Future work could explore the zero-shot 6D pose estimation using an end-to-end deep learning approach integrated with digital twin interactions.

Acknowledgement. This research is funded by the Vietnam Ministry of Education and Training under grant number B2024-VGU-03.

References

[1] Zhou, X., Xu, X., Liang, W., Zeng, Z., Shimizu, S., Yang, L.T. and Jin, Q. (2022) Intelligent small object detection for digital twin in smart manufacturing with industrial cyber-physical systems. *IEEE Transactions on Industrial Informatics* **18**(2): 1377–1386. doi:10.1109/TII.2021.3061419.

[2] Zhao, Z.Q., Zheng, P., Xu, S.T. and Wu, X. (2019) Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems* **30**(11): 3212–3232. doi:10.1109/TNNLS.2018.2876865.

[3] Liu, J., Sun, W., Yang, H., Zeng, Z., Liu, C., Zheng, J., Liu, X. *et al.* (2024) Deep learning-based object pose estimation: A comprehensive survey URL https://arxiv.org/abs/2405.07801. 2405.07801.

[4] Bochkovskiy, A., Wang, C.Y. and Liao, H.Y.M. (2020) Yolov4: Optimal speed and accuracy of object detection URL https://arxiv.org/abs/2004.10934. 2004.10934.

[5] Wang, A., Liu, L., Chen, H., Lin, Z., Han, J. and Ding, G. (2025) Yoloe: Real-time seeing anything URL https://arxiv.org/abs/2503.07465. 2503.07465.

[6] Barricelli, B.R., Casiraghi, E. and Fogli, D. (2019) A survey on digital twin: Definitions, characteristics,

applications, and design implications. *IEEE Access* **7**: 167653–167671. doi:10.1109/ACCESS.2019.2953499.

[7] Vidal, J., Lin, C.Y., Lladó, X. and Martí, R. (2018) A method for 6d pose estimation of free-form rigid objects using point pair features on range data. *Sensors* **18**(8): 2678. doi:10.3390/s18082678.

[8] Bishop, C.M. and Nasrabadi, N.M. (2006) *Pattern recognition and machine learning*, **4** (Springer).

[9] Kotsiantis, S.B. (2007) Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering* **160**(1): 3–24.

[10] Ren, S., He, K., Girshick, R. and Sun, J. (2016), Faster r-cnn: Towards real-time object detection with region proposal networks. URL https://arxiv.org/abs/1506.01497. 1506.01497.

[11] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C. (2016) Ssd: Single shot multibox detector. *Computer Vision – ECCV 2016* **9905**: 21–37. doi:10.1007/978-3-319-46448-0_2.

[12] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*: 779–788. doi:10.1109/CVPR.2016.91.

[13] Jocher, G. and Qiu, J. (2024), Ultralytics yolo11. URL https://github.com/ultralytics/ultralytics.

[14] Chang, J.R. and Chen, Y.S. (2018) Pyramid stereo matching network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*: 5410–5418. URL https://arxiv.org/abs/1803.08669. 1803.08669.

[15] Zhang, F., Prisacariu, V., Yang, R. and Torr, P.H. (2019) Ga-net: Guided aggregation net for end-to-end stereo matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*: 185–194.

[16] Wen, B., Trepte, M., Aribido, J., Kautz, J., Gallo, O. and Birchfield, S. (2025) Foundationstereo: Zero-shot stereo matching. *CVPR* URL https://github.com/NVlabs/FoundationStereo. 2501.09466.

[17] Jiang, H., Lou, Z., Ding, L., Xu, R., Tan, M., Jiang, W. and Huang, R. (2025) Defom-stereo: Depth foundation model based stereo matching. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

[18] Vogel-Heuser, B., Bi, F., Wittemer, M., Zhao, J., Mayr, A., Fleischer, M., Prinz, T. *et al.* (2023), Literature collection of digital twin definitions from various domains, https://mediatum.ub.tum.de/doc/1716587.

[19] Moshayedi, A.J. *et al.* (2023) Integrating virtual reality and robotic operation system (ros) for agv navigation. *EAI Endorsed Transactions on AI and Robotics* **2**. doi:10.4108/airo.v2i1.3181.

[20] Durojaye, A. *et al.* (2023) Immersive horizons: exploring the transformative power of virtual reality across economic sectors. *EAI Endorsed Transactions on AI and Robotics* **2**. doi:10.4108/airo.v2i1.3392.

[21] Durojaye, A., Kolahdooz, A. and Hajfathalian, A. (2025) Enhancing virtual reality experiences in architectural visualization of an academic environment. *EAI Endorsed Transactions on AI and Robotics* **4**. doi:10.4108/airo.8051.

[22] Dimosthenopoulos, D., Basamakis, F.P., Glykos, C., Bavelos, A.C., Mountzouridis, G. and Makris, S. (2024) A digital twin-based paradigm for programming and control of cooperating robots in reconfigurable production systems. *International Journal of Computer Integrated Manufacturing* : 1–21doi:10.1080/0951192X.2024.2428683.

[23] Sujatha, A., Kolahdooz, A., Jafari, M. and Hajfathalian, A. (2025) Simulation and control of the kuka kr6 900ex robot in unity 3d: Advancing industrial automation through virtual environments. *EAI Endorsed Transactions on AI and Robotics* **4**. doi:10.4108/airo.8026.

[24] Moshayedi, A.J. *et al.* (2022) Deep learning application pros and cons over algorithm. *EAI Endorsed Transactions on AI and Robotics* **1**(1). doi:10.4108/airo.v1i.19.

[25] Dong, Q.H., Nguyen, T.K., Tran, C.C., Pham, T.T., Do, D.T., Nguyen, H.V.K. and Nguyen, Q.C. (2025) Effectiveness of digital twin framework for collaborative robotic manipulation. *Journal of Technical Education and Science* In press.

[26] K. Noh, S. Ki Hong, S.M. and Lee, Y. (2024) Enhancing object detection in dense images: Adjustable non-maximum suppression for single-class detection. *IEEE Access* **12**: 30253–130263. doi:10.1109/ACCESS.2024.3459629.

[27] Ha, J. (2023) Probabilistic framework for hand–eye and robot–world calibration ax=yb. *IEEE Transactions on Robotics* **39**(2): 1196–1211. doi:10.1109/TRO.2022.3214350.

[28] Wu, J., Liu, M., Zhu, Y., Zou, Z., Dai, M.Z., Zhang, C., Jiang, Y. *et al.* (2021) Globally optimal symbolic hand-eye calibration. *IEEE/ASME Transactions on Mechatronics* **26**(3): 1369–1379. doi:10.1109/TMECH.2020.3019306.

[29] Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J. and Zisserman, A. (2010) The pascal visual object classes (voc) challenge. *International Journal of Computer Vision* **88**(2): 303–338. doi:10.1007/s11263-009-0275-4.

[30] ZJU-IVI (2023), Rt-less_10parts: Reflective textureless dataset, https://github.com/ZJU-IVI/RT-Less_10parts.