A Digital Twin Simulation Framework for Conveyor-Based Construction Waste Sorting Using RGB-Derived Mesh Assets

Shanuka Dodampegama[1], Avish Singh[1], Lei Hou[1], Ehsan Asadi[1], Kevin Zhang[1], and Sujeeva Setunge[1]

[1]Centre for Future Construction, School of Engineering, RMIT University, Melbourne, VIC 3001, Australia
s3975925@student.rmit.edu.au, s3898103@student.rmit.edu.au, lei.hou@rmit.edu.au, ehsan.asadi@rmit.edu.au, kevin.zhang@rmit.edu.au, sujeeva.setunge@rmit.edu.au

Abstract

Construction and demolition waste (CDW) sorting is increasingly being automated using robotics and AI to improve safety, efficiency, and material recovery. However, developing and validating perception and control algorithms directly on physical conveyor systems is costly, time consuming, and difficult to reproduce due to variability in object layouts and operating conditions. The lack of scalable and repeatable benchmarking environments limits rapid experimentation and fair comparison of alternative approaches. This paper presents a digital twin (DT) simulation framework for conveyor-based CDW sorting that enables systematic and data-driven evaluation of robotic strategies. The proposed system integrates three main components: an RGB annotation-driven mesh generation pipeline that converts real waste images into 3D object assets, a Unity based physics simulation that models the conveyor and robotic workspace, and a ROS 2 communication layer that enables real time interaction with external perception and control modules. The generated meshes preserve object footprint and spatial relationships while remaining computationally efficient, allowing realistic cluttered scenes to be reproduced automatically. The framework also provides structured ground truth and event-based signals to support quantitative evaluation. Experimental results show that mesh generation requires approximately 24 ms per image and 5.3 ms per object, while the simulation maintains stable real time performance at around 46–47 frames per second across varying conveyor speeds. These results indicate that the platform is suitable for large scale experimentation. The proposed DT offers a practical and extensible testbed for developing, benchmarking, and transferring robotic waste sorting algorithms from simulation to real world deployment.

Keywords –

Digital Twin; Robotics; Construction and Demolition Wastes; ROS 2

1 Introduction

Construction and demolition waste (CDW) represents a large proportion of global solid waste and poses significant environmental and economic challenges due to its volume and heterogeneity. Automated robotic systems have the potential to improve CDW sorting efficiency by reducing manual labour, increasing material recovery, and supporting circular economy goals. However, the irregular shapes of waste objects, high inter-class variability, and dynamic conveyor conditions make perception, manipulation, and control particularly challenging for robotic systems [1].

Digital twin (DT) technologies have recently emerged as a promising approach for accelerating development and testing of robotic systems in simulation before real world deployment. A DT refers to a virtual representation of a physical system that enables real time communication and interaction between the digital model and its physical counterpart, supporting design, optimisation, and testing tasks without the cost and risk associated with physical experimentation [2]. In robotics, DT platforms integrated with robotic middleware such as the Robot Operating System (ROS) enable evaluation of perception, planning, and control algorithms in controlled environments.

Despite these advances, most existing DT frameworks target structured industrial tasks or generic manipulation scenarios. In contrast, CDW sorting requires handling irregular objects, cluttered conveyor streams, and variable spatial layouts derived from real waste scenes. Existing research on robotic waste sorting largely focuses on object detection and classification, with limited infrastructure for reproducible evaluation of complete perception-to-action pipelines.

This paper presents a data-driven DT framework for conveyor-based CDW sorting that integrates ROS 2 with Unity and generates simulation assets from annotated RGB waste images. The framework enables realistic reproducible scene generation, object state publishing, and event-based evaluation within a controllable environment. The contributions of this work are threefold: (i) an automated pipeline for generating 3D mesh assets from annotated RGB images of CDW while preserving object footprint, spatial layout, and key geometric attributes relevant for simulation, (ii) a synchronised Unity–ROS 2 DT environment capable of representing conveyor-based CDW sorting scenarios with dynamically generated waste streams derived from real images, and (iii) a configurable benchmarking framework including scenario definitions and performance metrics that enable systematic evaluation of sorting systems in simulation under varying object layouts and conveyor operating conditions.

2 Related Work

2.1 DTs and Simulation Environments for Robotics

DTs are increasingly used for developing and validating robotic systems, enabling safe, repeatable experimentation and rapid iteration of perception, planning, and control algorithms. In construction and built environment applications, recent reviews highlight rapid growth in DT research while identifying challenges in interoperability, real time data integration, and scalable cyber physical architectures that combine sensing, simulation, and decision making [3].

Within robotics, Unity is increasingly adopted as a practical simulation platform due to its high-quality visualisation, flexible scene design, and interactive modelling capabilities. It is commonly integrated with ROS or ROS 2 to connect virtual scenes with external robotic software. Prior work has demonstrated Unity-based DT pipelines and evaluated aspects such as communication latency, synchronisation, and real time interaction [2]. Comparative studies with robotics focused simulators such as Gazebo indicate that Unity supports rapid scene prototyping and visual realism, while Gazebo provides tighter native ROS integration and physics modelling depending on the application [4].

Researchers have also explored more extensible simulation architectures that combine Unity with external computation or messaging layers to support collaborative experiments and modular system design. For example, multi-user simulation environments with Unity-MATLAB bridge have been proposed to enable colloborative experiments and evaluate alternative communication mechanisms [5]. Similarly, DT-driven control platforms have been developed in which Unity-based visual environments interact with industrial robot controllers through bidirectional state and command synchronisation [6]. These studies demonstrate that the DT frameworks can significantly reduce system integration effort and accelerate the development of robotic applications.

However, most existing DT platforms are designed for structured industrial tasks or general manipulation scenarios. They rarely consider domain specific challenges such as conveyor-based handling of irregular waste objects, cluttered object streams, and dynamically generated scenes derived from real observations. As a result, current DT environments provide limited support for benchmarking robotic sorting algorithms under realistic CDW conditions.

2.2 Robotic and AI for CDW Sorting

Robotic sorting has been widely investigated as a means of improving recycling efficiency and reducing manual labour in hazardous and labour-intensive environments. Despite these advances, automated waste sorting remains challenging because CDW streams typically contain cluttered scenes, irregular object shapes, and significant material variability, requiring robust perception and decision making capabilities [1]. Recent research in CDW automation has primarily focused on vision-based detection and classification methods. Advances in deep learning have enabled improved recognition and segmentation of waste categories, supporting automated material recovery systems. They also highlight important practical considerations for real world deployment, including distributed processing between edge and cloud systems and coordination between multiple robotic units in large facilities [7].

Several works have extended these perception capabilities by integrating object detection with robotic manipulation. In such systems, detected objects are localised and then handled using robot kinematic control to perform pick and place or transfer operations [8]. At the same time, sensitivity analyses and field studies emphasise that real world viability depends not only on recognition accuracy, but also on consistent operation under varying throughput, conveyor speeds, and labour or operational constraints [9].

These observations indicate that evaluation of robotic sorting system must consider the full perception-to-action workflow rather than isolated algorithmic components. However, most existing studies validate their approaches using limited experimental setups or fixed test scenarios. Infrastructure that enables controlled, repeatable, and configurable benchmarking of sorting strategies remains relatively limited.

2.3 Data-driven Geometry Generation for Simulation Assets

Realistic simulation environments require representative object geometry to accurately reflect real world operating conditions. In many robotic applications, the quality of object models strongly influences the validity of experimental results, particularly when evaluating perception and manipulation strategies. Manually modelling large numbers of irregular objects is time consuming and often impractical, motivating the development of automated asset generation methods.

In the computer vision community, significant research has focused on reconstructing 3D structure from a single RGB image. These approaches include methods that predict depth maps, point clouds, or mesh representations directly from image features. Survey studies highlight the diversity of reconstruction techniques as well as the inherent difficulties caused by ambiguity, occlusion, and incomplete observations [10]. Although such learning-based methods can produce detailed geometry, they are computationally demanding and may not generalise well to the diverse and irregular shapes commonly found in CDW streams.

Consequently, many robotics and DT simulations instead rely on predefined CAD models or manually curated asset libraries. Although these assets are suitable for structured industrial components, they do not capture the variability and randomness observed in real waste streams. Existing Unity-ROS DT setups have been used to simulate manipulation tasks, but they typically assume that object models are known in advance rather than derived directly from real images [11].

As a result, current approaches either focus on high fidelity reconstruction for visual realism or depend on handcrafted object libraries. Neither direction adequately supports lightweight and scalable geometry generation that can rapidly create large numbers of representative assets for benchmarking cluttered waste sorting scenarios.

2.4 Research Gap and Positioning of This Work

The literature highlights three complementary research directions; DT frameworks for robotic simulation, AI-based perception and manipulation methods for waste sorting, and data-driven approaches for generating 3D geometry from images. However, these areas remain largely disconnected in the context of conveyor-based CDW sorting. Existing DT platforms provide general simulation capabilities but rarely incorporate data-driven scene generation from real waste observations. Meanwhile, robotic CDW sorting research primarily focuses on perception or manipulation algorithms without providing infrastructure for systematic benchmarking under controlled conditions. In parallel, RGB-based 3D reconstruction methods aim for detailed geometry recovery but are often computationally demanding and unsuitable for rapid asset generation.

Consequently, there remains a gap between realistic scene modelling and practical benchmarking environment for evaluating robotic sorting systems. To address this gap, this work proposes a data-driven DT framework that integrates RGB annotation-based mesh generation with a synchronised Unity-ROS 2 simulation environment tailored to conveyor-based CDW sorting scenario. The framework enables automatic generation of realistic cluttered scenes, controlled scenario definition, and event-based performance evaluation, providing a reproducible platform for benchmarking robotic sorting strategies.

3 System Architecture

This section presents the architecture of the proposed DT framework for conveyor-based CDW sorting. The system integrates an RGB driven mesh generation module, a Unity-based simulation environment, and a ROS 2 communication layer. Together, these components enable controllable scenario generation and real time interaction with external robotic software, allowing perception, planning, and control algorithms to be evaluated in a consistent and repeatable environment.

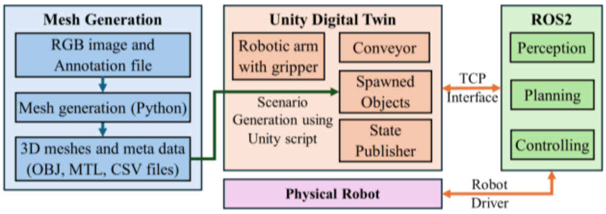

Figure 1. Overall system architecture and data flow between modules.

3.1 Overall Framework

Figure 1 illustrates the architecture of the proposed DT system, which comprises four components: an offline mesh generation module, a Unity based simulation environment, a ROS 2 coordination layer, and an optional physical robot interface. These components are loosely coupled so that scenario generation and simulation remain independent from external control algorithms. The mesh generation module converts annotated RGB images into lightweight 3D mesh proxies. Object boundaries extracted from polygon annotations are exported as OBJ files together with metadata describing class, dimensions, orientation, centroid, and estimated mass. The generated assets are stored locally and imported into Unity for conveyor sorting scene creation.

The Unity environment includes a conveyor, a simulated robot arm, and dynamically spawned waste objects positioned according to the source image layout using metric scaling. Communication between Unity and ROS 2 is achieved via a TCP interface that supports bidirectional data exchange. Unity publishes object states and scene variables, while ROS 2 nodes provide control inputs such as belt speed and task priorities. This architecture allows the same interfaces to be used in both simulation and physical robotic system.

3.2 Mesh Generation

The mesh generation module converts annotated RGB images into lightweight 3D object assets used to instantiate realistic waste scenes in the Unity DT. Instead of reconstructing exact geometry, the goal is to generate physically plausible mesh proxies that preserve object footprint, class identity, spatial layout, and key geometric attributes required for benchmarking. The overall pipeline is illustrated in Figure 2. The dataset used for asset generation consists of 800 RGB images from a public dataset, containing approximately 1077 concrete, 1146 brick, 1655 wood, and 931 rubber objects [12]. Polygon-based instance annotations compatible with the YOLO format were generated in previous work and reused here to extract object boundaries.

For each image, segmentation annotations are read to obtain polygon boundaries of individual objects. Polygon vertices are converted from image coordinates to metric units using a scale derived from the conveyor belt width, preserving relative spacing in the simulation. Each footprint is then extruded vertically to form a closed triangular mesh, with the height sampled from class specific ranges to introduce realistic thickness variation. The resulting mesh is centred at its centroid to simplify spawning within Unity.

Algorithm 1. RGB annotation-based mesh and metadata generation

Input: RGB image, polygon annotation file, belt width, class height ranges, material densities

Output: Mesh files and metadata for all objects in the image

1. Compute pixel to metric scale from belt width and image width
2. Load polygon annotations and convert to pixel coordinates
3. Remove objects that are truncated near image boundaries
4. For each polygon instance do
   a. Convert polygon vertices to metric coordinates
   b. Estimate geometric properties including area, centroid, and orientation
   c. Sample extrusion height from the class specific height range
   d. Triangulate polygon and extrude to generate a closed mesh
   e. Compute volume and derived mass using material density
   f. Shift mesh so centroid becomes local origin
   g. Export mesh as OBJ and MTL files with unit and axis conversion
   h. Save metadata including class label, centroid position, dimensions, orientation, area, volume, height, and estimated weight
5. Save all object records to a scene level CSV file

Figure 2. Algorithm used for mesh generation.

Most waste objects exhibit approximately convex shapes, while rubber fragments frequently contain concave boundaries. To support both cases, polygon triangulation is performed using an ear-clipping algorithm capable of generating stable meshes from arbitrary simple polygons. Basic geometric validation steps are also applied during preprocessing, including polygon orientation correction and triangulation checks, ensuring robust mesh generation even when annotations contain irregular vertex distributions or minor noise.

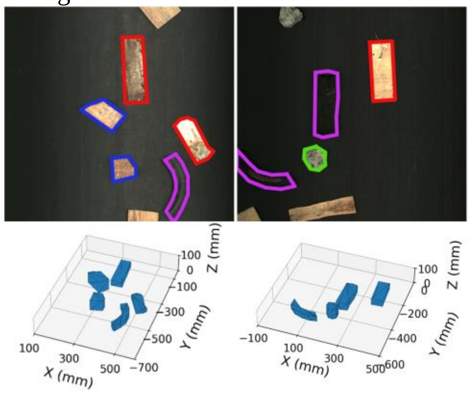

Figure 3. Example of RGB driven mesh generation. Top row shows the annotated RGB image with polygon boundaries and class labels. Bottom row shows the generated 3D meshes obtained by polygon extrusion and metric scaling

For compatibility with Unity, vertices are converted to metres and aligned with the simulation coordinate convention before export. The final assets are saved as OBJ and MTL files, together with a CSV file storing object metadata such as class, position, dimensions, orientation, and estimated mass. This automated process enables rapid and repeatable generation of simulation scenes without manual modelling. Figure 3 shows examples of the generated meshes alongside their annotated RGB images, demonstrating that the resulting assets preserve object layout while providing efficient volumetric representations for physics-based simulation.

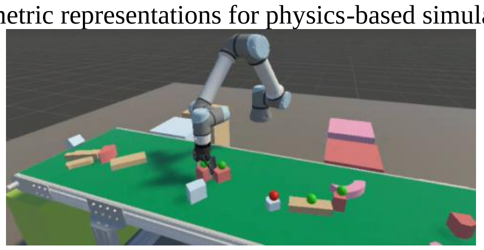

Figure 4. Unity based conveyor DT with dynamically spawned waste objects generated from RGB annotations. The environment includes the conveyor, simulated robot arm, and physics-based object interactions.

3.3 Conveyor DT in Unity

Unity serves as the DT of the physical conveyor sorting system, providing a controllable and repeatable environment for testing perception and decision making algorithms. The scene includes a conveyor belt, a simulated robotic arm with a gripper, and dynamically spawned waste objects generated by the mesh pipeline.

Unity is adopted for its flexible scene configuration, real time rendering, and built in physics capabilities.

At the start of each simulation episode, mesh assets and their metadata are loaded and used to spawn objects on the conveyor as shown in **Error! Reference source not found.**. Object attributes such as class, centroid, dimensions, and orientation are applied to preserving the relative layout of objects through metric scaling, enabling realistic cluttered scenes to be reproduced without manual placement.

Conveyor motion is simulated within Unity, while belt speed is controlled externally through ROS 2 commands and can include smooth variations to emulate realistic operating conditions. The environment continuously publishes object states and receives control inputs, allowing Unity to function as an interactive and measurable testbed for benchmarking rather than a passive visualisation tool.

3.4 ROS2 Interface and Topics

ROS 2 acts as the coordination layer linking the Unity DT with external perception, planning, and control modules. Data exchange is performed through ROS 2 topics via a TCP based interface, enabling bidirectional real time communication between the simulation and higher-level algorithms.

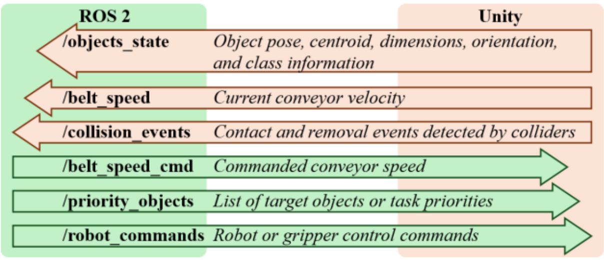

Figure 5. Main ROS 2 topics used for bidirectional communication between Unity and external modules. Arrows indicate data flow direction between Unity and ROS 2.

During operation, Unity publishes object states and global variables such as conveyor speed, while external nodes subscribe to this information for tracking, scheduling, and decision making. Control commands from ROS 2 are then sent back to Unity to influence the simulation, allowing the DT to operate as an interactive and modular experimental platform. Figure 5 summarises the main topics used in the current implementation, providing the necessary state and control information while maintaining a lightweight interface suitable for real time operation.

3.5 Event Detection via Colliders

To enable quantitative evaluation, the Unity DT uses physics-based colliders to automatically detect interaction events during simulation. Instead of relying on visual inspection, these triggers generate discrete signals for contacts, workspace transitions, and object removal, which are published to ROS 2 for real time monitoring and analysis. In addition, a collider at the tool centre point (TCP) is used to approximate grasp interaction, enabling evaluation of picking alignment without explicitly simulating detailed gripping physics.

Four types of colliders are used. Triggers attached to the gripper fingers detect contact with waste objects. A TCP collider provides a simplified representation of grasp interaction. Boundary colliders define the active conveyor workspace and record object entry and exit events. An additional trigger below the belt detects fallen objects and removes them from the scene to maintain stable simulation performance.

All events are transmitted through ROS 2 topics, allowing external nodes to compute statistics such as processed counts, missed interactions, and failures. This enables the DT to function as a measurable and repeatable benchmarking testbed.

4 Scenario Generation and Benchmarking Metrics

The proposed DT functions as both a simulation environment and a benchmarking platform for evaluating perception, scheduling, and manipulation strategies. It enables controlled experimentation through automatic scenario generation, structured ground truth, and quantitative performance metrics, supporting repeatable and fair comparison of algorithms under varying conveyor conditions.

4.1 Scenario Randomisation

To reflect real world variability and avoid overfitting to fixed layouts, the DT supports automatic randomisation of scene parameters at the start of each simulation episode. Randomisation is applied at both object and system levels. At the object level, scenes are generated by selecting mesh assets from different annotated RGB images and spawning them using their recorded spatial layout. Parameters such as extrusion height and material thickness are sampled within predefined ranges to introduce realistic variations in volume and mass while preserving object footprints.

At the system level, conveyor dynamics are varied by adjusting belt speed and adding smooth stochastic perturbations. Additional factors, including object ordering, spawn timing, and scene density, further diversify operating conditions. Because scenarios are generated programmatically, large numbers of repeatable test cases can be created efficiently, enabling statistical evaluation across multiple runs and reducing bias from fixed layouts.

4.2 Available Ground Truth

A key advantage of the DT is the availability of precise ground truth information for every object and event in the scene. Unlike physical experiments, where sensing noise and occlusion may limit measurement accuracy, the simulation environment provides direct access to exact geometric and state variables.

For each object spawned, the system maintains attributes including class label, centroid position, orientation, footprint dimensions, area, volume, height, and estimated mass. During runtime, Unity continuously publishes object poses and scene states to ROS 2, enabling external algorithms to access accurate and synchronised measurements. In addition, collider-based triggers generate event signals such as object contact, workspace entry or exit, and object removal.

This comprehensive ground truth supports detailed analysis of algorithm behaviour, debugging, and objective evaluation of success and failure conditions without manual annotation.

4.3 Evaluation Metrics

The DT facilitates quantitative assessment of sorting and manipulation strategies through measurable metrics derived from ground truth and event signals. These metrics can be computed online via ROS 2 nodes or offline from logged data.

Typical indicators include the number of processed objects, task completion rate, and the proportion of correctly selected or prioritised targets. Timing related metrics such as processing time per object and throughput are obtained from timestamps associated with entry and exit events. Contact and removal signals enable detection of failed interactions, such as missed picks or unintended collisions. Example metrics are defined as

$$\text{Throughput} = n_{success} / T \quad (1)$$

$$\text{Success rate} = n_{success} / N \quad (2)$$

$$\text{Miss rate} = (N - n_{success}) / N \quad (3)$$

where $n_{success}$ denotes the number of successfully handled objects, T is the total time, and N is the total number of objects. As these measures are computed directly from ground truth rather than sensor estimates, they provide consistent and repeatable benchmarks across different algorithms and experimental settings.

Overall, the combination of automated scenario generation, structured ground truth, and quantitative metrics enables the framework to function as a controlled testbed for developing and comparing robotic sorting strategies prior to deployment on physical hardware.

5 Experimental Demonstration and Evaluation Workflow

This section presents system-level demonstrations to validate the computational efficiency and runtime stability of the proposed DT framework. The focus is not on evaluating specific picking strategies, but on confirming that the mesh generation pipeline and Unity-based simulation operate in real time and support large scale experimentation under varying conveyor conditions.

5.1 Asset Generation Performance

The computational cost of the mesh generation stage was evaluated to assess the feasibility of creating simulation assets directly from annotated RGB images. For each image, the pipeline extracts polygon instances, constructs extruded meshes, computes geometric descriptors, and exports mesh and metadata files without manual intervention.

Across multiple images, the average processing time was approximately 24 ms per image. At the object level, average mesh construction time was 5.3 ms per instance. These results indicate that multiple objects can be generated within a few milliseconds, enabling rapid batch creation of large numbers of simulation scenarios. The low computational overhead demonstrates that the proposed RGB driven geometry generation approach is efficient and scalable, making it practical for repeated asset generation during experimentation.

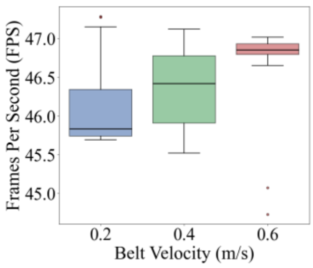

Figure 6. Variation of FPS for three different conveyor speeds measured for 1 minute.

5.2 Simulation Stability and Throughput

The runtime performance of the Unity-based DT was evaluated under cluttered scenes and varying conveyor speeds. Generated mesh assets were dynamically spawned using recorded spatial layout, resulting in multiple interacting objects on the conveyor.

Across tested speeds of 0.2 m/s, 0.4 m/s, and 0.6 m/s, the simulation maintained stable frame rates of approximately 46-47 frames per second (FPS), as shown in Figure 6. Only minor fluctuations were observed, with no noticeable degradation at higher belt velocities. This confirms that the lightweight mesh representation and physics configuration impose minimal computational overhead and enable real time execution on a standard desktop workstation. Communication performance was also evaluated, showing stable round-trip latency over time with an average round-trip time of approximately 100 ms between Unity and ROS 2. This indicates that the system supports reliable real time interaction for control and state feedback.

To emulate realistic operating conditions, conveyor velocity can be modulated with smooth stochastic variations around a nominal speed. Figure 7 shows an example profile with a base speed of 0.2 m/s and small perturbations. This variability enables testing of perception and scheduling algorithms under non constant motion while maintaining stable simulation performance. Overall, these results demonstrate that DT supports efficient asset generation, stable real time simulation, and reliable communication, providing a practical foundation for large scale benchmarking of robotic sorting strategies.

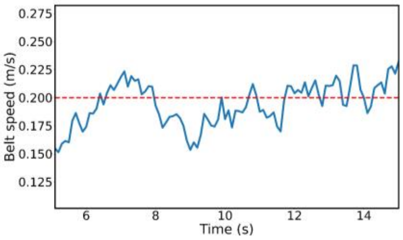

Figure 7. Example belt velocity profile generated from ROS2 side with a base speed of 0.2m/s.

5.3 Evaluation Workflow and Impact of Mesh Approximation

To illustrate how the proposed DT supports benchmarking of robotic sorting systems, a simplified picking experiment was conducted using a velocity feed-forward and feedback-based controller. The objective is not to evaluate the controller itself, but to demonstrate the interaction between perception, control, and the DT, and how performance metrics are computed.

In this framework, detailed physics-based gripping is not simulated. Instead, grasp interaction is approximated using colliders attached to the gripper fingers and tool centre point (TCP). A pick is considered successful when the gripper aligns on opposing sides of an object and the TCP overlaps the object region for a short dwell period (≈0.6 s) before moving to the sorting bin. Based on these events, metrics such as throughput, success rate, and miss rate can be computed using n_success as defined in Equations (1)–(3). To further demonstrate the evaluation workflow, tracking performance was measured by recording the time required for the end-effector to align with a moving target such that the positional error falls below 4 mm. Across multiple trials, an average tracking time of approximately 2.3 s was observed. This result indicates that DT enables consistent evaluation of perception-to-action behaviour under dynamic conveyor conditions.

Under this evaluation, the influence of mesh approximation primarily relates to object footprint, centroid location, and effective grasp width. As meshes are generated directly from annotated polygon boundaries, these planar characteristics are preserved. While fine-scale surface details and complex concavities are not explicitly modelled, their impact is limited for the system-level interaction criteria used in benchmarking.

The mesh generation pipeline is also robust to minor annotation noise. Variations in polygon vertices mainly affect local contour smoothness without significantly altering object footprint or layout. Geometric validation steps, including orientation correction and triangulation checks, ensure stable mesh construction from imperfect annotations. Consequently, the DT is designed for system-level evaluation of perception-to-action behaviour, object localisation, and motion alignment rather than detailed contact dynamics. The use of lightweight proxy meshes enables efficient generation of diverse CDW objects while maintaining sufficient geometric fidelity for reproducible benchmarking.

6 Discussion and Future Integration

The proposed DT framework provides a practical approach for bridging real world waste observations with controllable simulation-based experimentation. By combining RGB driven mesh generation, a Unity-based physics environment, and ROS 2 integration, the system enables repeatable and instrumented evaluation of robotic sorting strategies. Automatic conversion of annotated images into 3D assets eliminates the need for manual modelling and supports rapid generation of diverse scenarios. Together with structured ground truth and event signals, this enables systematic benchmarking that is difficult to achieve in physical experiments.

A key strength of the framework lies in its modular architecture. Asset generation, simulation, and control components are loosely coupled, allowing independent modification of perception, planning, or control modules without altering the environment. This reduces development time and enables fair comparison of alternative approaches under consistent conditions and supports large scale experimentation across varying object layouts, densities, and conveyor dynamics.

Despite these advantages, several limitations remain. The generated meshes are simplified geometric proxies based on polygon extrusion and do not capture fine scale surface details or complex material properties. While this is sufficient for system-level evaluation of perception, localisation, and motion planning, it may limit applicability for detailed grasp contact modelling. In addition, simulation physics cannot fully represent real world effects such as friction variability or deformability. Therefore, results obtained in the DT should be interpreted as indicative of system-level performance rather than exact predictions of real world behaviour.

Future work will focus on integrating the framework with physical robotic systems and perception pipelines. As both simulated and real setups share the same ROS 2 interfaces, control algorithms developed and tested in the DT can be transferred to the physical robot with minimal modification. Planned extensions include camera-based detection and tracking, implementing grasp planning and scheduling strategies, and conducting closed loop picking experiments on the conveyor. Additional improvements will explore more advanced geometry modelling, domain randomisation of physical parameters, and expansion of the asset library using larger and more diverse datasets.

Overall, the framework provides a scalable foundation for iterative development of AI-driven robotic sorting systems. By enabling safe, repeatable, and data rich experimentation in simulation prior to hardware deployment, the DT reduces integration risk and accelerates the pathway from algorithm design to real world application.

7 Conclusions

This paper presented a DT simulation framework for conveyor-based CDW sorting that integrates RGB driven mesh generation, a Unity-based physics environment, and a ROS 2 communication layer. The system automatically converts annotated RGB images into 3D mesh assets to generate realistic and repeatable sorting scenarios. By coupling data-driven asset generation with real time interaction and structured ground truth, the framework enables systematic benchmarking of perception, scheduling, and manipulation strategies without the cost and risk of physical experimentation. Experimental results confirmed that the mesh generation pipeline operates efficiently, requiring only milliseconds per object, while the Unity simulation maintains stable real time performance across varying conveyor speeds. The framework also supports the evaluation of perception-to-action behaviour and motion alignment under dynamic conditions. Although the meshes are simplified geometric proxies, they preserve key planar characteristics required for benchmarking and are sufficient for system-level evaluation. The modular design supports integration with ROS 2 based algorithms and future deployment on physical robotic platforms, providing a practical and extensible testbed that facilitates rapid development and evaluation of AI-enabled robotic waste sorting systems.

References

[1] S. Dodampegama, L. Hou, E. Asadi, G. Zhang, and S. Setunge, ‘Revolutionizing construction and demolition waste sorting: Insights from artificial intelligence and robotic applications’, *Resources, Conservation and Recycling*, vol. 202, p. 107375, Jan. 2024.

[2] M. Singh *et al.*, ‘Unity and ROS as a Digital and Communication Layer for Digital Twin Application: Case Study of Robotic Arm in a Smart Manufacturing Cell’, *Sensors (Basel)*, vol. 24, no. 17, p. 5680, Aug. 2024.

[3] V. V. Tuhaise, J. H. M. Tah, and F. H. Abanda, ‘Technologies for digital twin applications in construction’, *Automation in Construction*, vol. 152, p. 104931, Aug. 2023.

[4] M. Singh *et al.*, ‘Comparative Study of Digital Twin Developed in Unity and Gazebo’, *Electronics*, vol. 14, no. 2, p. 276, Jan. 2025.

[5] A. Haces-Garcia and W. Zhu, ‘Building an Accessible and Flexible Multi-User Robotic Simulation Framework with Unity-MATLAB Bridge’, *Computers*, vol. 13, no. 11, p. 282, Nov. 2024.

[6] W. Chang, W. Sun, P. Chen, and H. Xu, ‘Construction and Validation of a Digital Twin-Driven Virtual-Reality Fusion Control Platform for Industrial Robots’, *Sensors*, vol. 25, no. 13, p. 4153, Jan. 2025.

[7] Z. Wang, X. Yang, X. Zheng, D. Huang, and B. Jiang, ‘Computer Vision System for Multi-Robot Construction Waste Management: Integrating Cloud and Edge Computing’, *Buildings*, vol. 14, no. 12, p. 3999, Dec. 2024.

[8] H. T. N. Le and H. Q. T. Ngo, ‘Application of the vision-based deep learning technique for waste classification using the robotic manipulation system’, *International Journal of Cognitive Computing in Engineering*, vol. 6, pp. 391–400, Dec. 2025.

[9] X. Liu, Z. Farshadfar, and S. H. Khajavi, ‘Computer Vision-Enabled Construction Waste Sorting: A Sensitivity Analysis’, *Applied Sciences*, vol. 15, no. 19, p. 10550, Jan. 2025.

[10] M. S. U. Khan, A. Pagani, M. Liwicki, D. Stricker, and M. Z. Afzal, ‘Three-Dimensional Reconstruction from a Single RGB Image Using Deep Learning: A Review’, *J Imaging*, vol. 8, no. 9, p. 225, Aug. 2022.

[11] J. Yun *et al.*, ‘Digital twin model construction of robot and multi-object under stacking environment for grasping planning’, *Applied Soft Computing*, vol. 149, p. 111005, Dec. 2023.

[12] T. Ji, J. Li, H. Fang, R. Zhang, J. Yang, and L. Fan, ‘Rapid dataset generation methods for stacked

construction solid waste based on machine vision and deep learning', *PLOS ONE*, vol. 19, no. 1, p. e0296666, Jan. 2024.