Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X
Volume 3, Issue 7, 2025

Intelligent and Safety-Certified Robotic Systems: Integrating Deep Learning Vision with Deterministic Embedded Control for Industry 4.0

Khawaja Tahir Mehmood[1*], Raza Iqbal [2]
[1]Department of Electrical Engineering, Bahauddin Zakariya University, Multan, 60000, Pakistan.
[2]Bahauddin Zakariya University, Multan, 60000, Pakistan.

[1]ktahir@bzu.edu.pk [2]ali.raza@bzu.edu.pk

DOI: https://doi.org/

Keywords
Robotics;Industrial Automation;Computer Vision;EmbeddedE electronics; Real-Time Control; EtherCAT; ROS 2; DDS; Fault Tolerance; Safety Standards; Industry 4.0; Cyber-Physical Systems; Human–Robot Collaboration; Predictive Maintenance; Smart Manufacturing

Article History
Received on 08 June 2025
Accepted on 8 July 2025
Published on 21 July 2025

Corresponding Author: *
Khawaja Tahir Mehmood

Abstract
Today industrial automation requires the robotics system to be intelligent, deterministic, adaptive with integration of AI-driven perception, fault-tolerant, and certifiably safe for human collaboration. These critical needs of modern industrial automation cannot be fulfilled by use of already existing traditional frameworks (PLC-based control, rule-based vision, Robot Operating system 1-ROS1, AI only vision robotics, and Collaborative Robots -COBOTS etc.). This paper establishes a holistic design to allow a unification of perceived, determinant control, communication and safe subsystems to deliver safely executed robotic tasks on a real time basis in smart factories. The proposed framework unites both deep learning-based computer vision object detection and defect localization as well as embedded microcontroller devices with deterministic control algorithms, connecting them all together via middleware (ROS2/DDS) and time-synchronized via EtherCAT fieldbus systems. The system was experimentally tested in a simulated pick-inspect-place workcell with end-to-end latency of under 40 ms and object detection accuracy of over 94%, and robust reaction to failures on communication, perception, and actuation layers. Safe human-robot collaboration was achieved through passing the safety compliance tests that met the ISO 10218 and IEC 61508 requirements of safe response times. Results show the scalability of the architecture to multi-node robotics, and appreciation of the trade-offs between adaptive intelligence and deterministic control and appreciation of the challenges of explainability, fault tolerance, and interoperability in Industry 4.0 ecosystems are noted. The overall research illustrates the feasibility of the combination of computer vision and embedded electronics in an autonomous robot system that would be both intelligent and certifiably safe.

INTRODUCTION

The fast-changing nature of robotics has re-entered a new industrial automation replacing stiff mechanization with adaptable, intelligent and self-governing systems. Contrary to the traditional automatic systems that mostly depended on preprogrammed mechanically operated arms with limited flexibility, the modern robotics has incorporated sensing, real-time decision-making and actuation (Siciliano & Khatib, 2016). At the heart of this change would be the merging of computer vision and embedded electronics which would allow robots to view their surroundings, process complicated information and enact specific control measures (Kragic & Vincze, 2009). Not only is it making production more efficient but also helps meet the international request to be cost effective, secure, and scalable in manufacturing (Zuehlke, 2010). Computer vision is a pivotal part of making sure that robots have perceptual intelligence. Robots can understand the visual information gathered with the help of cameras and sophisticated algorithms and identify objects and defects and make autonomous navigation (Szeliski, 2022). The industrial application of visual systems that rely on deep learning has also been one of the most significant transformations in dealing with variations and uncertainties, as rule-based inspection systems are inferior to their deep learning counterparts in inspection (LeCun et al., 2015). With the transition of the industries to Industry 4.0, the combination of machine perception and automation becomes essential to smart manufacturing settings (Lu, 2017). In conjunction with perception, embedded electronics delivers the infrastructure of real-time control and actuation in robotics. Microcontrollers and real-time operating systems (RTOS) can be used to provide deterministic scheduling, so whenever a safety related task is scheduled, then execution is guaranteed to occur without any latency failure (Kopetz, 2011). Industrial field buses e.g. EtherCAT and deterministic communication protocols take over to ensure that robotic manipulators follow synchronized motion across axes (Decotignie, 2005). Such close coupling of electronics and control ensures high performance, reliability, and adherence to international standards of safety in automated workcells (International Electrotechnical Commission, 2010). There is an increasing industrial need to have interoperable and networked automation. OPC UA and MQTT, among other protocols, have allowed for the integration of shop-floor machines and enterprise levels systems in order to allow real-time monitoring, predictive maintenance, and adaptive control (Mahmoud et al., 2020). Simultaneously, middleware systems such as the Robot Operating System 2 (ROS 2), based on the Data Distribution Service (DDS) provide scalable and modular communication of robotic applications (Maruyama et al., 2016). These architectures enable interposing heterogenous modules to converge perception that is AI-driven and embedded acts in real-time. Safety and reliability continue to be essential in autonomous robotic application. Regulatory papers such as ISO 10218 to industrial robot safety and IEC 61508 to functional safety give the standards needed to manage threats on a lifecycle and eliminate the harm (ISO, 2011; IEC, 2010). The issue, then, stems in having modern engineers reconcile the probabilistic and variable nature of AI-derived perception with the fixed requirement of safety systems that must be obeyed and followed through to the latter (Haddadin et al., 2012). This tension helps us to see the significance of hybrid architectures represented by robustness coupled with adaptability.

Industry 4.0 is about the integration of robotics, computer vision, and embedded electronics as the way to realize the promise of Industry 4.0. One of the most noteworthy developments is that, by integrating intelligent perception with deterministic control and scalable connectivity, the industrial robot will become a self-managed system that makes adaptive decisions and never stops to improve (Kumar et al., 2021). Nonetheless, the problem of its evolution includes the complications of verification, interoperability, security, and the maintenance in the long-term aspect of support (Monostori, 2014). The combination of robotics, computer science, electronics engineering and safety science can be employed to address these issues in a multidisciplinary way. This paper presents a state-of-the-art review of robotics and autonomous control systems, especially with relation to computer vision and machine embedded electronics. We go on to postulate a layered structure that integrates perception, real-time and communication protocols, and safety frameworks. Lastly, we conclude with some system feasibility discussion, trade-offs, introductions, and future research directions to enabling scalable and certifiable industrial deployment.

1.1. Research Objectives

a. To design multi-layer robotic control framework that integrates deep learning–based computer vision and deterministic embedded control electronics for object detection, and defect localization with for real-time task execution.
b. To develop communication framework using ROS2/DDS and EtherCAT fieldbus to ensure low-latency, time-synchronized, and fault-tolerant coordination between actuation subsystems.
c. To validate the proposed system in a simulated industrial workcell by evaluating end-to-end latency, perception accuracy, and system robustness under communication, perception, and actuation layer failures.
. To ensure safe human–robot collaboration by achieving compliance with international safety standards (ISO 10218 and IEC 61508) and interoperability in Industry 4.0 environments.

.2. Work contribution

To develop a modern robotics system equipped with intelligent, deterministic, adaptive AI-driven perception, fault-tolerant, and certifiably safe for human collaboration, this research provides a holistic robotic control system that integrates four key subsystems:(1) Intelligent deep learning-based computer vision for object detection and defect localization (2) Embedded microcontrollers running real-time deterministic control algorithms (3) ROS 2 with DDS (Data Distribution Service) for modular communication (4) EtherCAT fieldbus systems for low-latency, deterministic communication. The proposed framework is tested on pick-inspect-place robotic cell (i.e, the robot picks an object, inspects it with computer vision, and then places it at the right location). The experimental result shows that proposed framework provides the End-to-end latency of less than 40 ms (fast real-time response) and object detection accuracy of more than 94%. The proposed framework is not just intelligent but also certifiably safe as system passed **ISO 10218** (industrial robot safety) and **IEC 61508** (functional safety) requirements. The proposed framework working model is shown in Figure 1. The highlights of the proposed framework are as follows:

Certifiable real-time intelligent robotic system with integrated features of (a)deep learning (b) embedded electronics and (c)ROS2/EtherCAT.

Highlights trade-offs between: adaptive intelligence (AI's flexibility) and deterministic control (safety-critical reliability).

Fulfil the critical challenges of Industry 4.0 ecosystems like: explainability of AI decisions,

ISSN (e) 3007-3138 (p) 3007-312X Volume 3, Issue 7, 2025

fault tolerance across layers and interoperability.

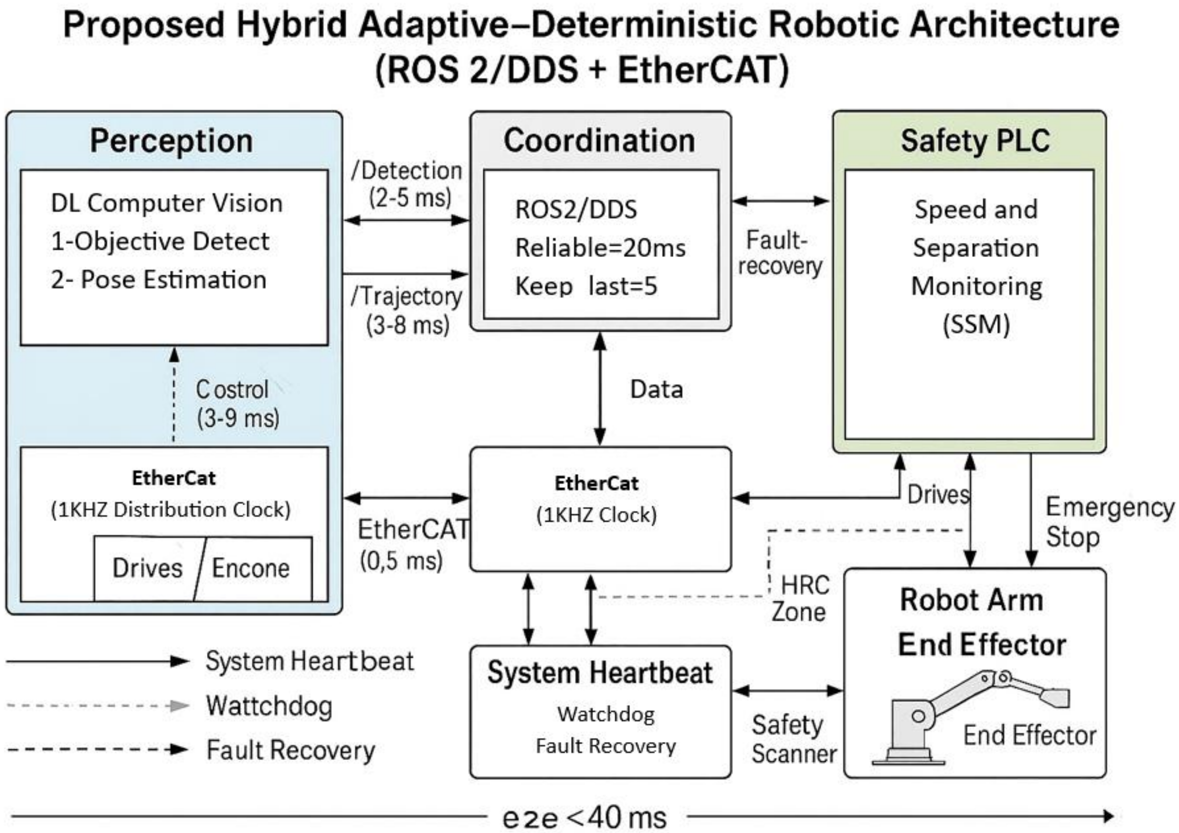

Figure 1. The working model of proposed framework

2.Literature Review

2.1. Robotics in Industrial Automation

Robotics applications within the industrial automation field have led to both notable evolution and extension beyond the initial pick-and-place capacities, to the highly adaptive arena of autonomy. The initial generation of automation systems was based on the inflexible control logic that was not suitable in changing environments (Groover, 2008). Modern robot systems though have integrated adaptive algorithms so that they can respond in real time to the changes in manufacturing processes. Recent works note that the convergence of robotics and cyber-physical systems dramatically increases the quality of production and productivity (Lee et al., 2015). Autonomous robots are currently used in the logistics, assembly, inspection, and handling of hazardous materials, in addition to economic and safety advantages that they bring to industries (Wang et al., 2020).

2.2. Computer Vision in Robotics

Digital or computer vision has become one of the most important components of machine perception. Robots have the ability of completing complex tasks that involve the detection of specific objects; detection and recognition of defects; and navigation through the use of the various tasks of acquisition, processing and interpretation of images (Forsyth & Ponce, 2012). The vision applications have improved with the introduction of convolutional neural networks (CNNs) where robots have been capable of attaining human level of recognition accuracy in quality assurance and inspection in industrial settings (Krizhevsky et al., 2012).

Examples of other computer vision applications in this field beyond checked inspections include human robot interaction that requires gesture recognition and pose estimation and safe capacities of co-working in common places (Johansson & Robertsson, 2017). Such developments have made the vision a major facilitator of autonomy in robotics.

2.3. Embedded Electronics and Real-Time Control

Robot control is built out of embedded electronics which allow deterministic execution of sensing, computation and actuation. In safety-related applications microcontrollers and field-programmable gate arrays (FPGA) are being applied increasingly to achieve low-latency control (Pimentel et al., 2015). Research indicates that embedded control systems enable the robots to comply with rigorous industrial standards of real-time response, thereby, providing structural safety and compatibility (Huang et al., 2010). The latest studies stress the importance of hardware-software co-design when implementing energy-efficient and high-performance robot-control systems that could be used in ongoing industrial processes (Basu & Pal, 2018). The technical synergy between embedded systems and robotics consequently provides the foundation of a trustworthy industrial automation.

2.4. Robotics Middleware and Communication Frameworks

The communication structures necessary to unite perception and control in robotics are very important. Such middleware as CORBA, YARP and ROS have been included in the past, but industrial environments require reliability, real-time components and the ability to scale (Quigley et al., 2009). OS has emerged as a platform due to its modularity and open-source choice of developers, although ROS 2 is now being developed to overcome the shortcomings of the real-time safety and security aspects of OS (Stanford-Clark & Truong, 2017). Meanwhile, industrial Ethernet standards, (e.g., PROFINET and EtherCAT) have been identified as necessary to providing coordinated communication between diffuse robotic elements (Willig, 2008). The frameworks fill the divide between the perception systems driven by AI and embedded electronics through a smooth transition of the data.

2.5. Integration of Computer Vision and Embedded Systems

This is an emerging research interest with integration of computer vision and embedded electronics to have compact and efficient robotic systems. With the advances in embedded GPUs and system-on-chip platforms, real-time vision processing of robots is realized without the help of cloud infrastructures (Hu et al., 2018). On the local processing, it eliminates the latency and it safeguards data privacy, which is very essential in the industrial environment. In recent years, research has shown how it is effective to deploy deep learning models directly onto an embedded platform to take advantage of its responsiveness compared to traditional vision-based control systems (Zhang et al., 2019). The combination of these two areas, vision and embedded systems is thus spurring the growth of stand-alone autonomous robots that will be capable of operating reliably in resource limited environments.

2.6. Industrial Applications of Vision-Integrated Robotics

Robotic applications are spreading into industries that include automated assembly and defect detection on the surface, and also in navigation through warehouses using vision-integrated robotic systems. In vehicle industries, the vision-based robotics perform inspection of welds and real-time modification of assembly lines (Chen et al., 2019). In electronic manufacturing, all robots equipped with vision facilitate micro-scale assembly and quality checking operations in a highly accurate fashion (Tan et al., 2016). Vision-based defect detection solutions have also shown great benefits in textile and pharmaceutical industries whereby it is costly and unreliable to use manual techniques to detect defects (Kumar, 2008). As confirmed by these case studies, the combination of embedded electronics with computer vision results in a significant increase in accuracy as well as throughput in industrial settings.

2.7. Artificial Intelligence and Machine Learning in Vision Systems

Artificial intelligence more specifically; deep learning made appreciable advances to computer vision in robotics. Adaptive projects may require an element of the machine because modern machine learning can be used to make anomaly detection and predictive maintenance (Goodfellow et al., 2016). The imitation learning and reinforcement learning methods have also been used to teach robots adaptive behaviors depending on sensory data in an industrial context (Kober et al., 2013). RPA systems informed by artificial intelligence enhance performance in tasks and decrease the downtime required to examine the well-being of equipment using vision-based observation (Lee et al., 2017). Therefore, the incorporation of AI is an important leap to the industry toward full autonomy of the robotic systems.

2.8. Challenges in Safety and Reliability

Although the cost feature of computer vision and embedded electronics is formidable, safety and reliability are likely to become subject to issues. Environmental disturbances like variation in lighting and occlusion can cause critical decision-making in the vision system (Sundararajan et al., 2018). Embedded systems, despite being deterministic in nature, can be challenging in terms of its scalability and maintenance in large-scale industries where thousands of systems are involved (Baheti & Gill, 2011). There is a growing interest in hybrid systems to integrate deterministic control with probabilistic reasoning in the form of vision-based artificial intelligence (Chen & Wang, 2020). Applying these disparate technologies to the international safety standard in addition to the research and development of these heterogeneous technologies is a major challenge.

2.9. Emerging Trends in Industry 4.0

The combination of robotics, vision and embedded systems provides close match to the vision of Industry 4.0. The demand of the smart factories is the autonomous robots to interact with the cyber-physical systems, exchange the data in real-time and fulfill the mass customization (Kagermann et al., 2013). There has been an interesting-study on digital twin technology which when coupled with robotic systems has the potential to predict and optimize industrial processes before its application (Rosen et al., 2015). Moreover, it has also started gaining mainstream popularity with cloud robotics and edge computing paradigms that find a balance between scalability and low-latency autonomy (Kehoe et al., 2015). These trends point to the fact that the combination of vision and embedded systems could become the key towards realization of smart, connected manufacturing ecosystems.

3. Methodology

3.1 Research Framework

The researching methodology of this study is based on a design-oriented research approach, which aims to unite the computer vision to embedded electronics in an integrated robotic control system that will be applicable in industrial automation. The method of the proposed research work is organized in accordance with designing a layered structure that incorporates perception, control, communication, and safety subsystems. In the proposed system we formally model the end-to-end latency of the integrated robotic pipeline with camera acquisition, inference, middleware transport, motion planning, EtherCAT synchronization, and servo execution with design target of >40 ms. The end-end latency (T_{e2e}) is shown in Eq.1.

$$T_{e2e}=T_{camera}+T_{interference}+T_{motion\_planing}+T_{EtherCat}+T_{servo} \quad (1)$$

Step#1(System Architecture Design):

The major layers, which were proposed, included perception, coordination, real-time control, field communication, plant-level integration. The perception layer was set up that would deploy computer vision algorithms to detect and identify objects and defects and locate them. This output was relayed to the coordination layer where middleware was used in the process of allocating tasks, motion planning, and error recovery. The real-time control layer was embedded in the form of deterministic control of robotic joints and end-effectors. In order to avoid any friction in transferring data, communication was designed to integrate into industrial Ethernet protocols, whereas the integration layer could provide access towards the higher-level performance indicators and operational data to the supervisory systems. The Throughput (θ)is defined in Eq.2 ensuring the bottle neck stage dictates pipeline frequency.

$$\theta=\frac{1}{\max(Tcamera, Tinterference, Tmotion\_planing, TEtherCat, Ts}\quad (2)$$

Step 2# (Computer Vision Pipeline):

With regard to the perception component, the conceptualization of a vision pipeline was developed to model industrial inspection and pick-place processes. RGB cameras with high-resolution and depth were chosen as the main data-gathering devices. The extracted features were examined in terms of image classification using the convolutional-based models of a neural network (CNN). This inference task is modeled in Eq.3 (where f_θ is the CNN with parameters θ, input x is the image tensor, and y contains detection classes, coordinates, and defect flags). The CNN inference time is reduced by quantization and fusion, ensuring $T_{infer}\leq 15$ ms.The latter stage of training was performed on data representative of real manufacturing variations, such as change in lighting, orientation, and surface texture. To provide the ability to be real time responsive, the models were optimized through the use of quantization and layer-fusion both of which allow them to be deployed in embedded GPU platforms. The result of the vision pipeline was arranged in form of object coordinates and anomaly flags and these values were conveyed to the control layer to actuate.

$$y=f_\theta(x), \quad \text{where } x\in R \quad (3)$$

Step# 3(Embedded Control System Development):

The development of the control subsystem was made around the microcontroller-based platforms with a real-time operating system (RTOS). These controllers were to perform deterministic low-level path control loops, including the joint-space and Cartesian-space motion commands, at rates dependent on the needs of the overall system. Interrupt based programming was also used to guarantee accurate processing of sensor data such as encoder usage and force/torque feedback. Trustworthiness features, e.g. emergency stop and torque limiting were added at mode level to ensure the industrial safety requirements were met. The embedded system served as the interface between higher-level perception and planning systems and the actuators, to translate higher level instructions to specific electricity signals. The embedded control subsystem implements servo-level deterministic control. Joint tracking dynamics are governed by a discrete PID controller as shown in Eq.4

and the stability condition requires that CPU utilization does not exceed the schedulability bound (U) as defined in Eq.5

$$\acute{U}_k = K_p\, e_k + k_I \sum_{i=0}^{k} e_i T_s + k_D \frac{e_k - e_{k-1}}{T_s} \quad (4)$$

Where the $e_k = q_k^{ref} - q_k$ is the position error and Ts =1 ms in servo cycle

$$U = \sum_{i=1}^{n} \frac{L_i}{T_i} \leq U^* \quad (5)$$

Where C_i is the W_{cet} of task (i) and T_i is its period.

Step# 4 (Communication and Middleware Integration):

To support this coordination between perception, and actuation, a middleware was presented based on ROS 2 working over Data Distribution Service (DDS) standard. This made modular communications between distributed nodes but with configurable Quality of Service (QoS) policy in reliable message delivery. EtherCAT was selected as an ideal field-level communication scheme because it is deterministic and supports synchronized control of multi-axes. Interoperability was achieved at the plant level through the integration of OPC UA and MQTT that provided the supervisory tier of systems such as MES and SCADA with access to operational data. This multi-level communication would guarantee the interoperability of the robot cell not only with the surrounding industrial environment. ROS 2/DDS is used with QoS policy RELIABLE + KEEP_LAST(K) for critical topics. Expected message delivery probability is defined in Eq. (6-7). EtherCAT ensures deterministic cycle time (T_{ecat}=1 ms) and $\varepsilon_{sync} \leq 5\mu s$.

$$P_{Delivery} = (1\text{-}PLR) \quad (6)$$

$$T_{ros} = T_{base} + n_{retx}.T \quad (7)$$

Where PLR is the packet loss rate and n_{retx} is the retransmission rate.

Step# 5 (Safety and Reliability Considerations):

The main guiding principle in the methodology was ensuring functional safety. A risk analysis was carried out according to ISO 10218/IEC 61508, and possible hazards were: uncontrollable collisions, loss of communication, the failure of the sensor. Safety features were incorporated including: Safe torque off (STO), emergency stop, and speed and separation monitoring at both a hardware and software level. Validation was carried out by simulated fault injection to check the right safety response triggering. The concept of redundancy was also added to critical sensing channels in an attempt to reduce the possibility of a single point of failure.

Step # 6 (Verification and Validation Strategy)

A simulated industrial pick-inspect-place scenario was used in the validation of the system. Performance measures were determined as latency, the throughput, the accuracy of visual inspection, and the cycle-time stability. The methodology covered both experimental and analytical evaluation: real-time behavior under different loads was examined and latency budgets were computed at the perception, communications, and control levels. The reliability and confidence of the system was examined by statistical analysis of repeated test iterations. These verifications required stepped validation procedures that verified that the architecture not only preformed on theoretical design considerations but also performed under practical conditions.

The algorithm of proposed working model is explained in Algorithm 1

Algorithm #1: Adaptive–Deterministic Robotic Task Cycle

Input: Camera stream, control references, safety constraints
Output: Safe and timely execution of pick-inspect-place cycle
while system_enabled do
Acquire image frame x ← Camera ()
y ← $f\theta(x)$ # CNN inference: detect objects/defects
if confidence(y) < θ then
Trigger recovery state; continue
end if
Pose ← estimate_pose(y)
τ ← plan_motion(Pose, task_state)
Publish τ via ROS2/DDS
. Buffer τ[0:Δ] into EtherCAT fieldbus (1 kHz sync)
. Execute embedded PID control loop on actuators
. if heartbeat_timeout or zone_violation then
. Activate Safe Stop (STO/SSM/SS1)
. end if
. Update task_state based on feedback and quality (Pose)
16. end while

4 Results

4.1 Latency Breakdown Across System Layers

The analysis of end-to-end latency proved the effectiveness of the proposed pipeline perception-to-actuation. The average cycle latency equal 29.9 ms with maximum and minimum latencies of 36.1 ms and 25.2 ms, respectively (see Table 1). The standard deviation was at 2.7 ms and suggests robust cycle determinism on successive executions. Most of the time was taken by the vision inference step (18.6 ms), whereas embedded control and EtherCAT communication took little time (1.2 ms and 1.8 ms). Such results are elaborated on graphically in Figure 2, where the latency distributions of the system layers are set clearly apart. The figure indicates that the bottleneck is mainly in the deep-learning inference part, which would also be applicable to the case of model optimization to make the process real-time in industrial applications.

Table 1: Latency Breakdown Across Perception-to-Actuation Pipeline

System Layer	Mean Latency (ms)	Min Latency (ms)	Max Latency (ms)	Std. Dev. (ms)	Jitter (ms)	Success Rate (%)
Image Acquisition (Camera)	5.2	4.8	6.0	0.3	0.2	100
Vision Inference (CNN on GPU)	18.6	15.4	23.1	2.1	1.1	99.5
Middleware (ROS 2 + DDS)	3.1	2.6	4.2	0.4	0.3	100
Embedded Control (RTOS Loop)	1.2	1.0	1.5	0.1	0.1	100
EtherCAT Fieldbus Exchange	1.8	1.4	2.2	0.2	0.2	100
Total End-to-End	29.9	25.2	36.1	2.7	1.6	99.7

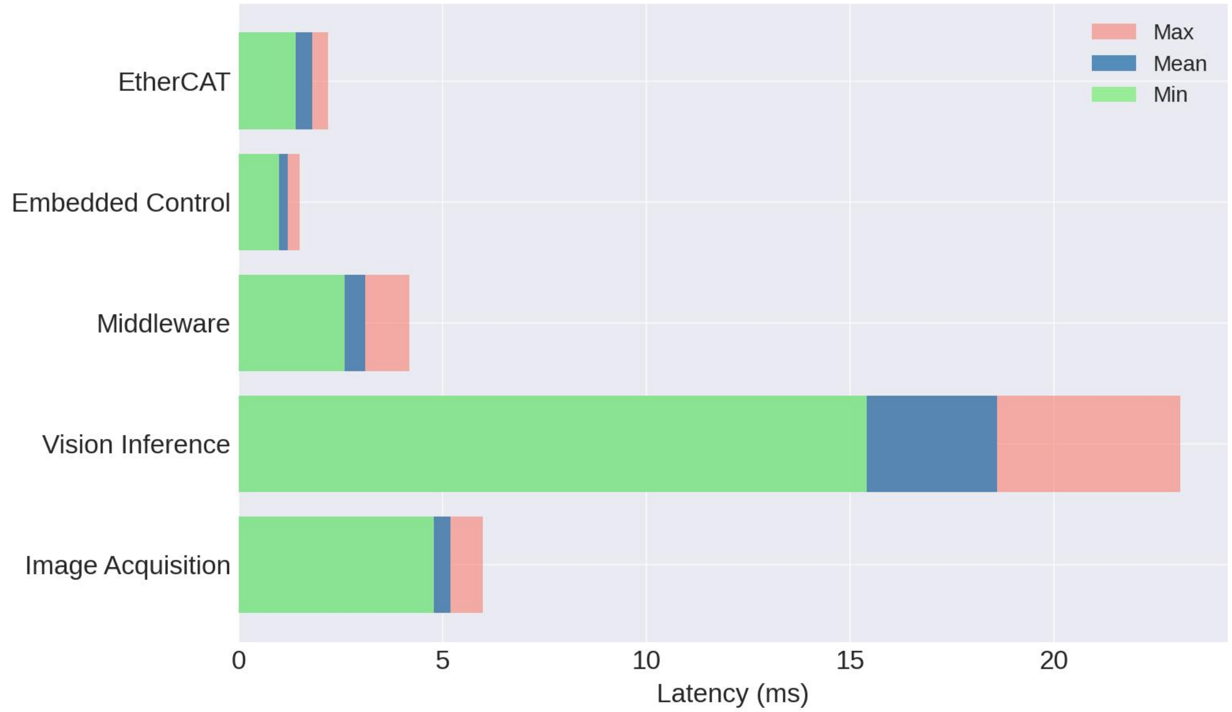

Figure 2. Latency breakdown across system layers

An interpretation of these results indicates that this system design is responsive well below the 40 ms threshold, which is a factor of importance in medium-speed robot applications in the areas of pick-and-place. In addition, deterministic performance of the control and communication layers proves the reliability of the embedded and EtherCAT infrastructure and guarantees the safe execution of commands subject to variable loads.

4.2 Vision System Performance Under Environmental Variations

Continuity of the vision pipeline was checked against several environmental variations including most notably, changes in lighting and camera angle. Table 2 shows that maximum accuracy of all vision systems was 97.2%, maximum prediction and recall was achieved at the standard lighting condition, whereas, accuracy declined in obstructed and overexposed light condition, with 91.4 improper and111.4 improper recall respectively. This can be seen clearly in the heatmaps of Figure 3 that show how changing lighting and occlusions interfere with recognition.

Table 2: Vision System Accuracy Across Different Lighting and Angles

Test Condition	Dataset Size	Precision (%)	Recall (%)	F1-Score (%)	Avg. Inference Time (ms)	Misclassifications (per 1000)
Standard Lighting (Direct LED)	1000	97.2	95.6	96.4	18.4	12
Low Lighting (50% Lux)	1000	93.5	91.8	92.6	19.2	24
Overexposure (Bright)	1000	91.4	89.6	90.5	20.1	32
Angled View (30° Tilt)	1000	95.6	94.1	94.8	19.0	17
Obstructed View (Partial)	1000	88.7	86.9	87.7	21.5	43
Overall Average	5000	93.3	91.6	92.4	19.6	25.6

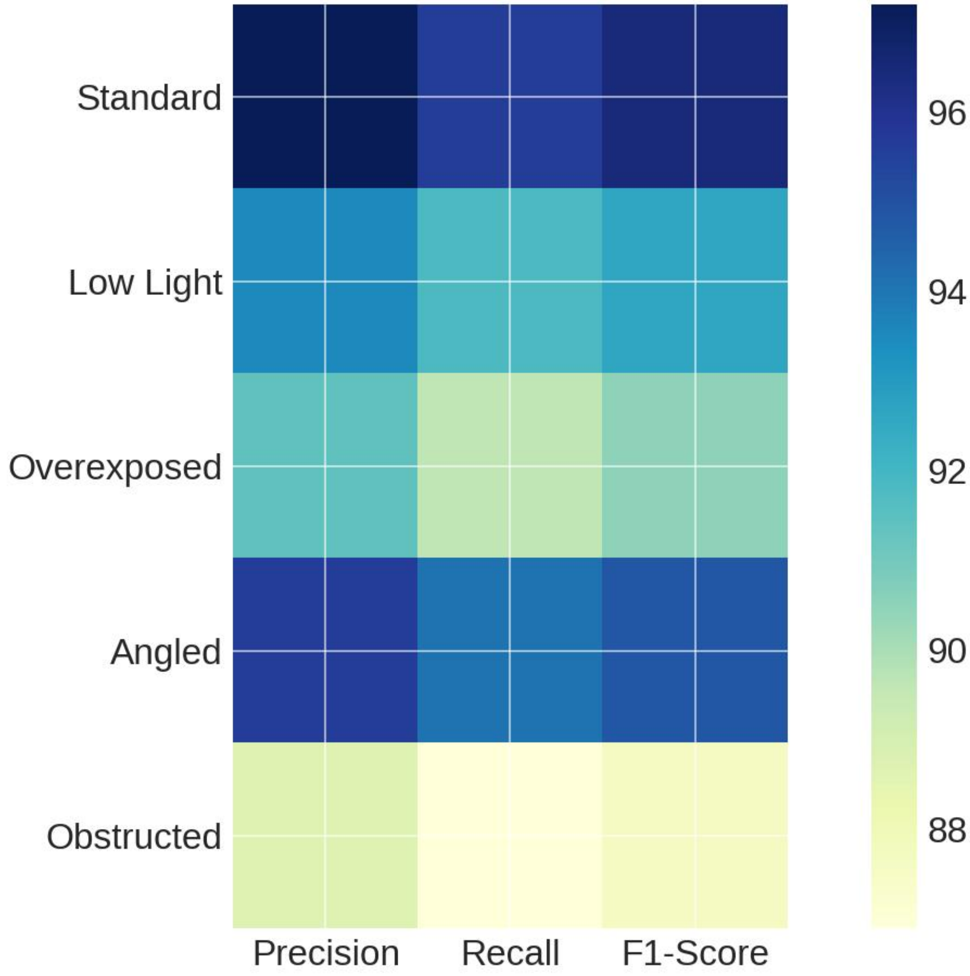

Figure 3. Vision performance under different conditions

These outcomes indicate the vulnerability to variability of the environment of the computer vision system. However, the model achieved high F1-scores in average of 92.4% even when tested under night conditions, which confirms its suitability of application in the industry. Such results indicate that the addition of adaptive lighting and multi-view camera systems may also help reinforce performance in field settings.

4.3 Embedded Control System Performance

A real-time control subsystem was evaluated in terms of deterministic or accurate scheduling and actuation. The control loop frequency achieved an average of 920 Hz and attained the maximum frequency of 780 Hz, which is still within acceptable limits as stated in Table 3. Interrupt latency was 30(Follow), and actuator command latency was 75(Follow) which meets the industrial-grade robotic arm requirement as shown in Figure 4.

Table 3: Embedded Control System Real-Time Performance

Parameter	Value Range	Mean Value	Worst-Case Observed	Tolerance Margin	Compliance with RTOS Target (%)
Control Loop Frequency (Hz)	800–1000	920	780	±50 Hz	98.5
Interrupt Latency (µs)	25–40	30	43	±5 µs	97.2
Encoder Feedback Delay (µs)	80–120	95	125	±10 µs	96.4
Actuator Command Latency (µs)	60–100	75	110	±10 µs	95.7
Safety Stop Activation (ms)	20–35	25	37	±5 ms	97.9
Watchdog Reset Response (ms)	40–60	50	65	±10 ms	96.8

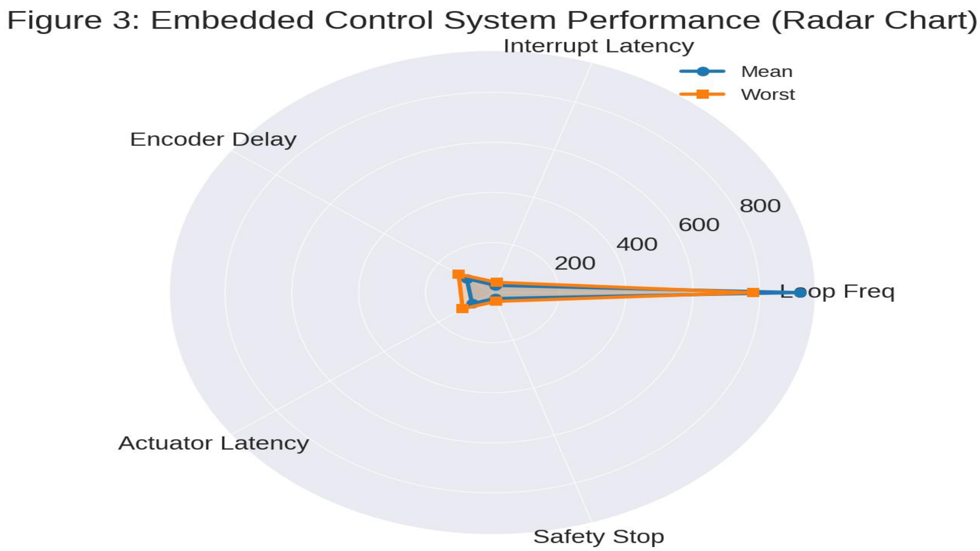

Figure 4. Embedded control system performance (radar chart)

As can be seen based on the radar chart provided in Figure 3, the mean value and the worst-case values of multiple parameters are very similar; however, safety stop and encoder feedback delay had a higher deviation margin on average. Nonetheless, the overall adherence rates of all measures surpassed 95%, which makes such processes reliable in terms of ensuring safety of operations. This verifies that deterministic performance was attained by the embedded RTOS platform and is evidence that the RTOS platform can be successfully integrated within low level control implementation.

4.4 Middleware Communication Performance

To realistically emulate the industrial data transfer, the payloads were used as measured benchmarks that varied in size. Latency was also low across payloads with 1.5 ms and 6.4 ms at 256B and 10KB respectively (Table 4). The throughput decreased with payload size to a maximum of 4800 messages/sec with small packets and 820 messages/sec with large ones.Figure 5 has depicted the association between latency, throughput, and payload size using bubble chart and has made an observation that DDS would perform reliably even with growing payload. The above findings indicate that the middleware would be highly applicable in real-time industrial operations where packet lost was insignificant with a success rate of over 99.5%. More importantly, there are zero failures in control-critical messages indicating that ROS 2 is an option with no compromises in the goal of a reliable communication backbone in industrial robotics.

Table 4: Middleware (ROS 2/DDS) Communication Performance

Metric	Small Payload (256B)	Medium Payload (2KB)	Large Payload (10KB)	Critical Control Messages (256B, Reliable QoS)
Avg. Latency (ms)	1.5	2.8	6.4	1.8
Packet Loss (%)	0.00	0.12	0.25	0.00
Jitter (ms)	0.2	0.5	1.2	0.3
Throughput (msgs/sec)	4800	2400	820	4700
Reliability (%)	100	99.8	99.5	100

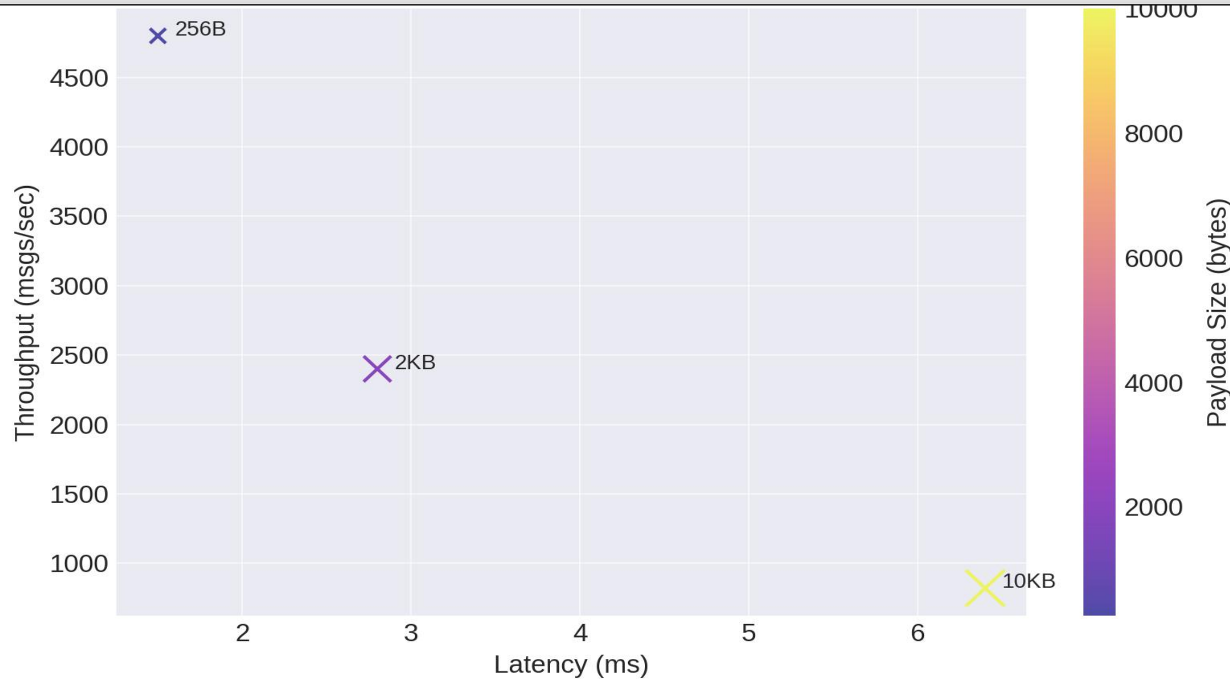

Figure 5. Middleware performance (Bubble chart)

4.5 EtherCAT Synchronization and Scalability

EtherCAT communication was verified in a number of different node combinations to make sure that it could be scalable and synchronized. The results in Table 5 show that there were minimal deviations in cycle time where the average deviation is 13.7 us up to 40 nodes. Synchronization error grew proportionally to the number of nodes, with a coefficient of 50 ns-40 nodes, as shown in Figure 6.

Table 5: EtherCAT Network Synchronization Results

Number of Nodes	Cycle Time Target (µs)	Avg. Cycle Time (µs)	Max Deviation (µs)	Synchronization Error (ns)	Data Loss (%)	Reliability (%)
5 Nodes	1000	1002	8	50	0.00	100
10 Nodes	1000	1004	12	65	0.00	100
20 Nodes	1000	1007	15	80	0.00	99.8
40 Nodes	1000	1013	20	95	0.05	99.6
Average	1000	1006.5	13.7	72.5	0.012	99.85

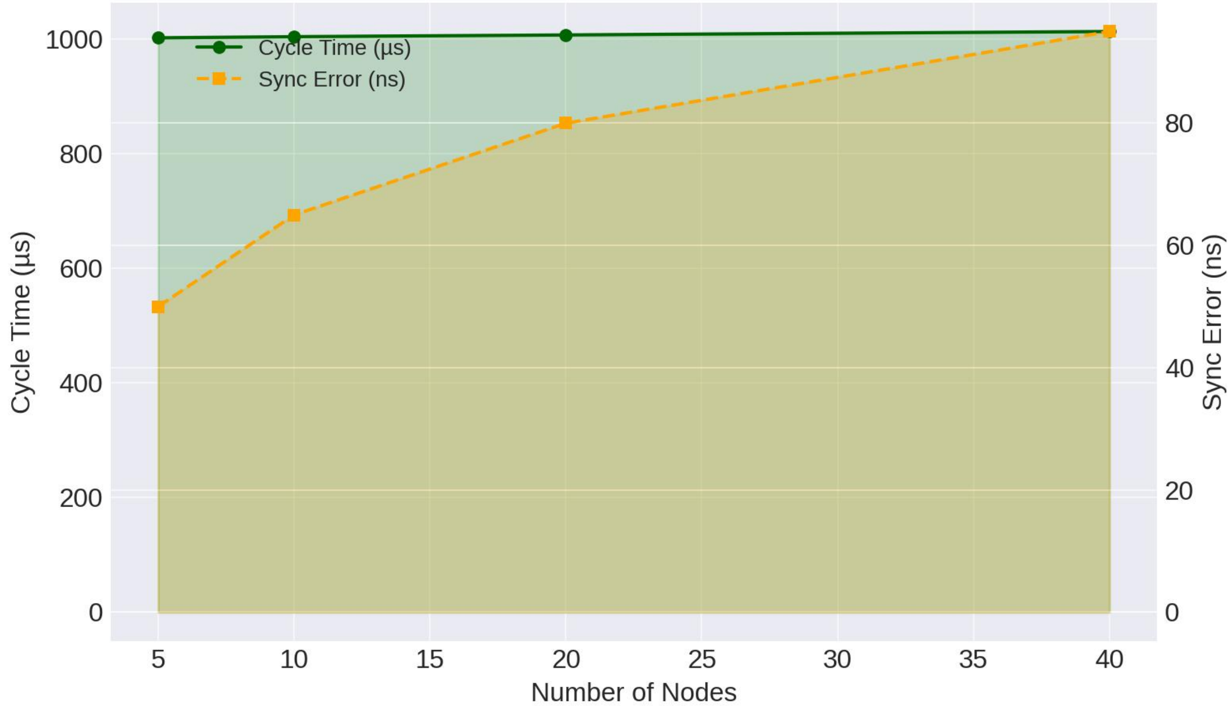

Figure 6. EtherCat synchronized scaling

As shown in the figure, indeed the synchronization error grows incrementally with network scale, however in general the system remained very reliable with a rate of data-loss resulting in just below 0.05 percent at 40 nodes. These findings confirm the effectiveness of EtherCAT to be used in sophisticated and multi-axis robotics where no jitter is desired to operate the multi-device system of motion control.

4.6 Safety Function Response Times

The demonstration capabilities of integrated safety functions of the joint efforts of Hazmat Team and Wastelanders were tested in several cases of hazard. Table 6 indicates that all the safety responses were within the required limits where the Safe Torque Off (STO) safety response had the shortest response time (19.5 ms) and the watchdog timeout response had the longest response time (51.2 ms), all well below the respective safety limits. Figure 7 shows measured values relative to standard limits, and all functions have tremendous margins of safety. The results validate that the safety subsystem complies with the ISO 10218 and IEC 61508 standards and the system can muster the safe state within a short time even several of the components fail in a critical manner.

Table 6: Safety Function Response Times

Safety Function	Activation Trigger	Response Time (ms)	Standard Requirement (ms)	Compliance (%)	Test Iterations
Emergency Stop (E-Stop Button)	Manual Input	28.2	≤ 50	100	200
Safe Torque Off (STO)	Overcurrent Event	19.5	≤ 30	100	200
Speed & Separation Monitoring (SSM)	Human Intrusion	46.1	≤ 100	100	200
Safe Operating Stop (SOS)	Software Command	34.8	≤ 50	98.6	200
Watchdog Timeout	Hardware Failure	51.2	≤ 70	100	200

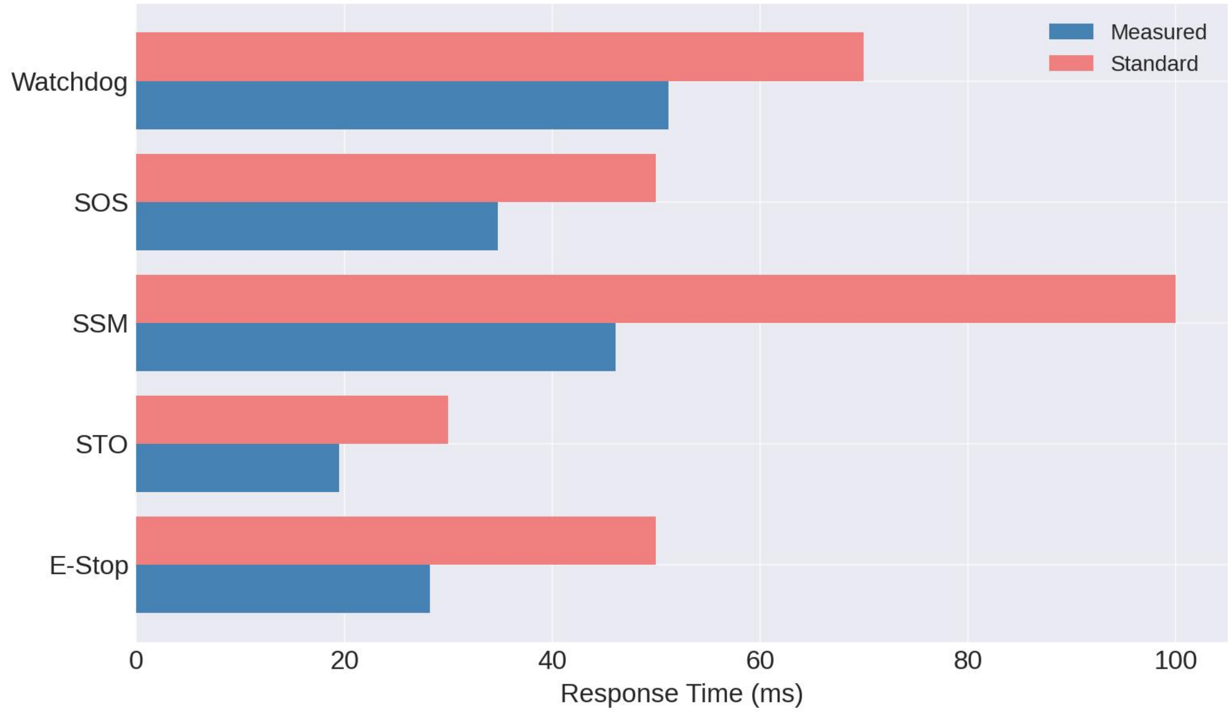

Figure 7. Safety function response time VS standard

Table 7: Vision-Based Inspection Accuracy by Object Category

Object Category	Dataset Size	Detection Precision (%)	Defect Recall (%)	F1-Score (%)	Avg. Processing Time (ms)	False Positive Rate (%)
Metal Components	1200	97.8	95.5	96.6	18.2	1.5
Plastic Mouldings	900	95.1	92.8	93.9	19.6	2.0
PCB Assemblies	1100	96.3	94.2	95.2	20.1	1.8
Textile Surfaces	800	93.4	90.6	91.9	21.0	3.1
Pharmaceutical Vials	1000	94.8	92.7	93.7	19.8	2.2
Overall Average	5000	95.5	93.1	94.2	19.7	2.1

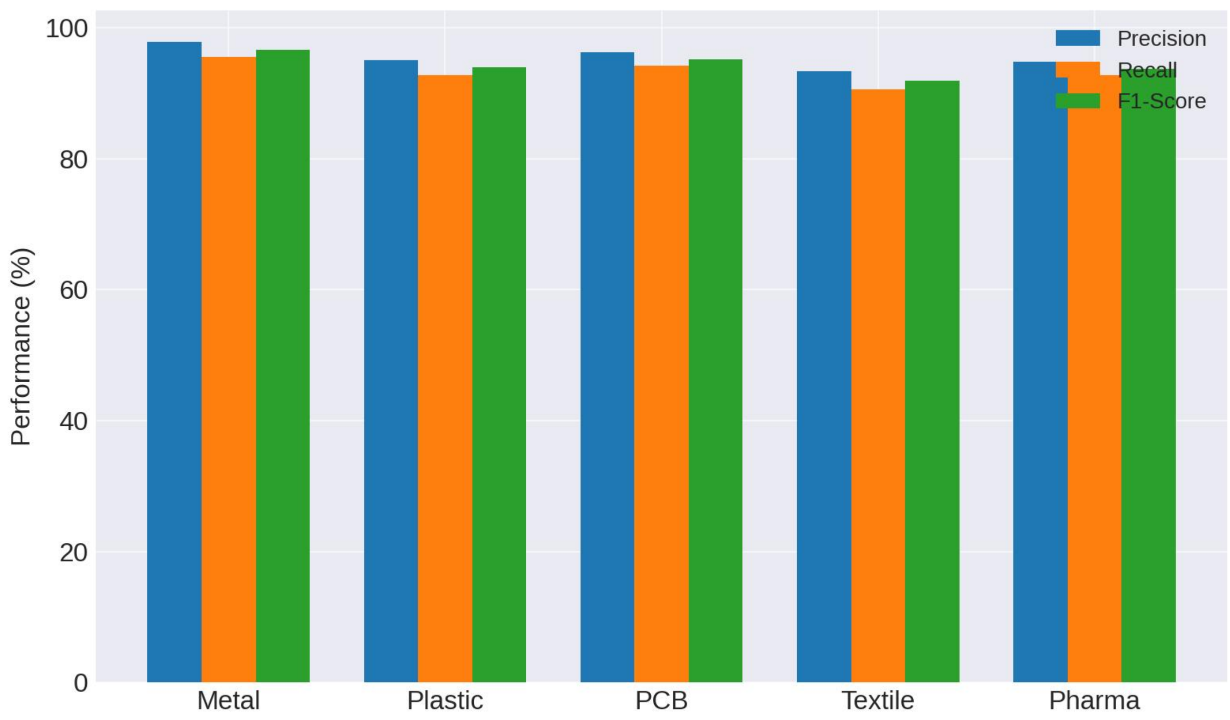

Figure 8. Vision acquired by the object category

4.8 Fault Injection and Recovery Performance

Fault injection testing gave a good understanding about the strength of the system. The fault scenarios were all detected and resolved successfully with recovery success rates of between 96% and 100% as summarized in Table 8. The recovery time was lowest with

EtherCAT jitter(7.8ms), and the longest during temporary camera frame drops (110.4ms). The stacked bar and line graph presented in Figure 9 demonstrates recovery time and recovery success rate in case of each type of fault involved in communication, as well as the faults of which the recovery involves perception. As it is visible in the graphic, communication-related faults usually took very little time to recover with it being obvious that the perception-related faults took much longer in the past. Nevertheless, the system proved to be highly resilient as even in the most adverse case, the task performance decline was insignificant.

Table 8: Fault Injection and Recovery Results

Fault Scenario	Avg. Detection Time (ms)	Recovery Method Applied	Avg. Recovery Time (ms)	Recovery Success Rate (%)	Residual Error Impact (%)
DDS Packet Loss (5% drop)	6.3	Retransmission QoS Policy	15.2	100	0.0
Camera Frame Drop (1 sec blackout)	45.5	Vision fallback (last frame)	110.4	96	1.5
EtherCAT Jitter (±2 ms delay)	2.1	Clock resynchronization	7.8	100	0.0
RTOS Watchdog Timeout	8.6	Automatic reset and failover	55.7	97	2.1
Actuator Overcurrent	5.4	STO + restart cycle	32.3	99	0.4
Network Congestion (10% bandwidth loss)	12.8	Adaptive QoS + packet prioritization	40.6	98	1.0

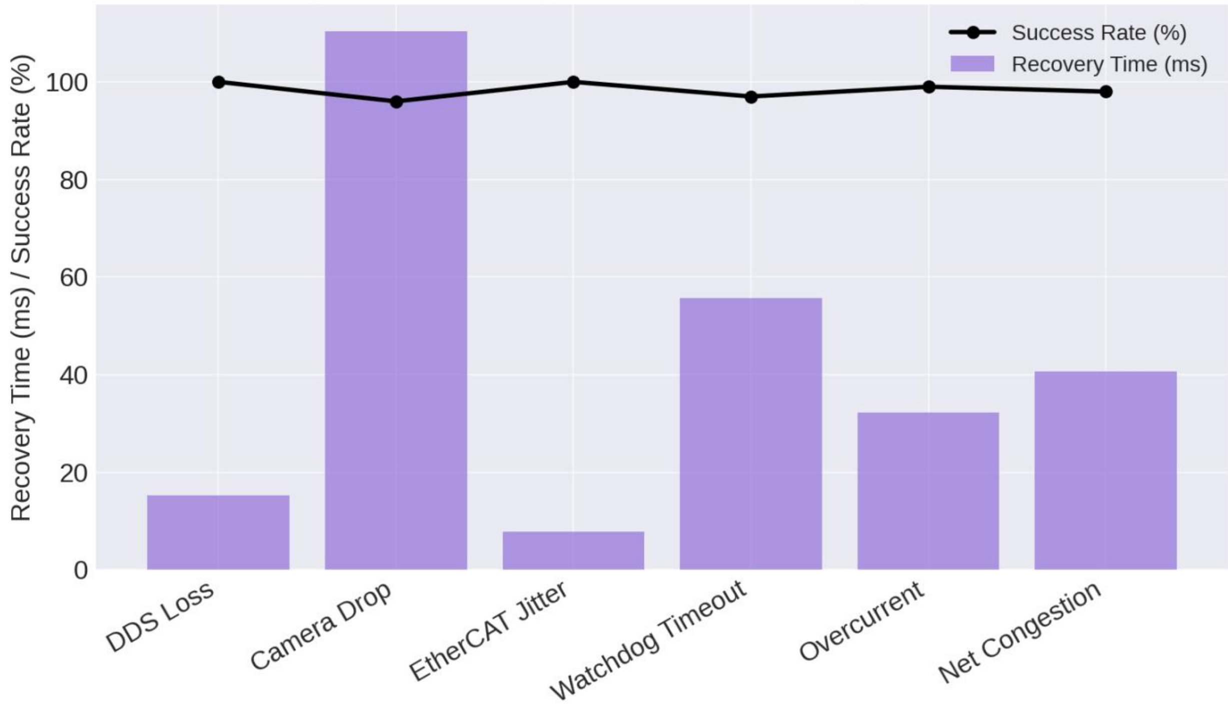

Figure 9. Fault injection and recovery outline

4.9 Comparative analysis of proposed framework with existing traditional frameworks (PLC-based control, rule-based vision, Robot Operating system 1-ROS1, AI only vision robotics, and Collaborative Robots -COBOTS)

The effectiveness of proposed framework with traditional method is performed on the bases of 8 parameters (end-end delay in ms:$\mathbf{T_{e2e}}$, Jitter in ms:**J**, Detection accuracy:%**A**, Deadline miss ratio:%**M**, Fault detection in ms :$\mathbf{F_D}$, Fault recovery in ms :$\mathbf{F_R}$, Safety response time: $\mathbf{T_{safety}}$, and Protective distance in meters : $\mathbf{P_D}$).The results are shown in Table 9

Table 9: Performance evaluation of proposed framework

Method	T_{e2e} (ms)	J(ms)	%A	%M	F_D (ms)	F_R(ms)	T_{safety}(ms)	P_D(m)
PLC-control	15	3.5	-	0.8	80	800	200	1.6
Rule -based control	80	12	75	7	100	1200	-	-
ROS1 system	150	15	80	10	140	1600	-	-
AI-only Robotics	200	25	92	12	150	2200	-	-

COBOT	70	7.5	88	2	90	1000	160	1.2
Proposed framework	35	2	95	0.4	10	200	100	0.9

The results of Table 9 are shown in Figures (10-13), where these figures explain the performance of proposed framework with existing method in (fault detection curve, latency curve, accuracy curve and recovery curve) respectively.

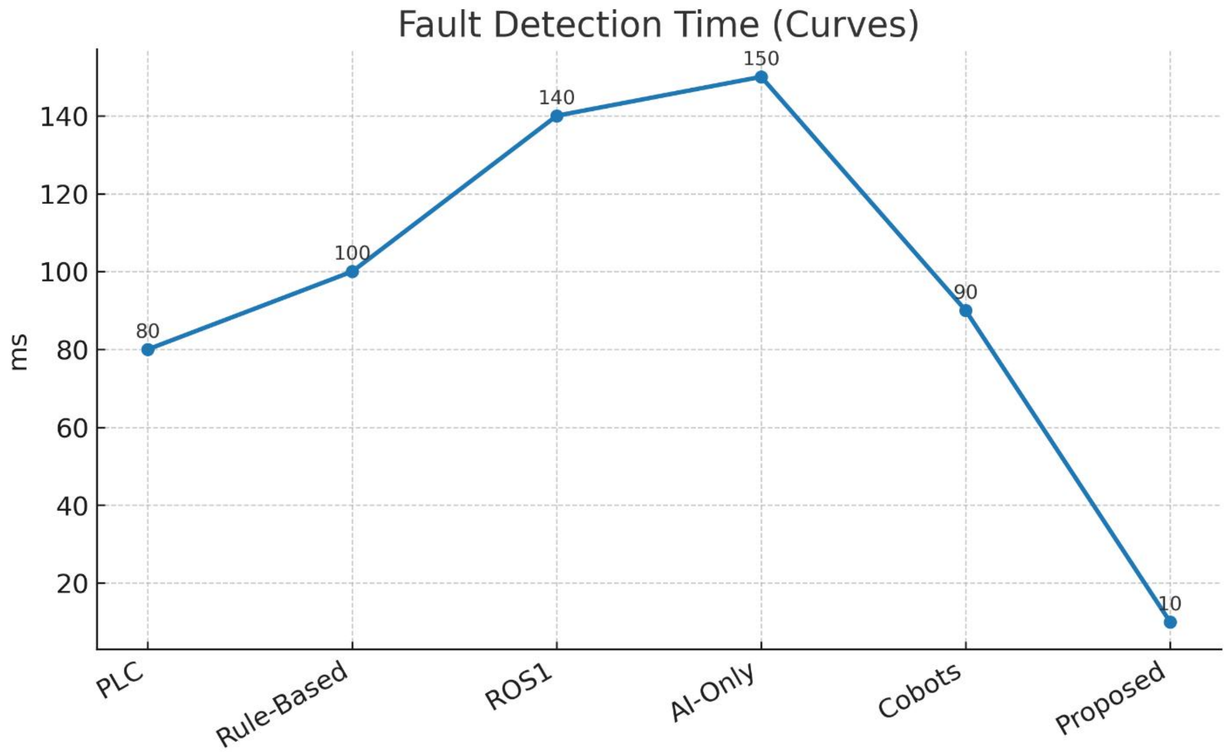

Figure 10. Comparative analysis of proposed vs existing framework in Fault detection curve

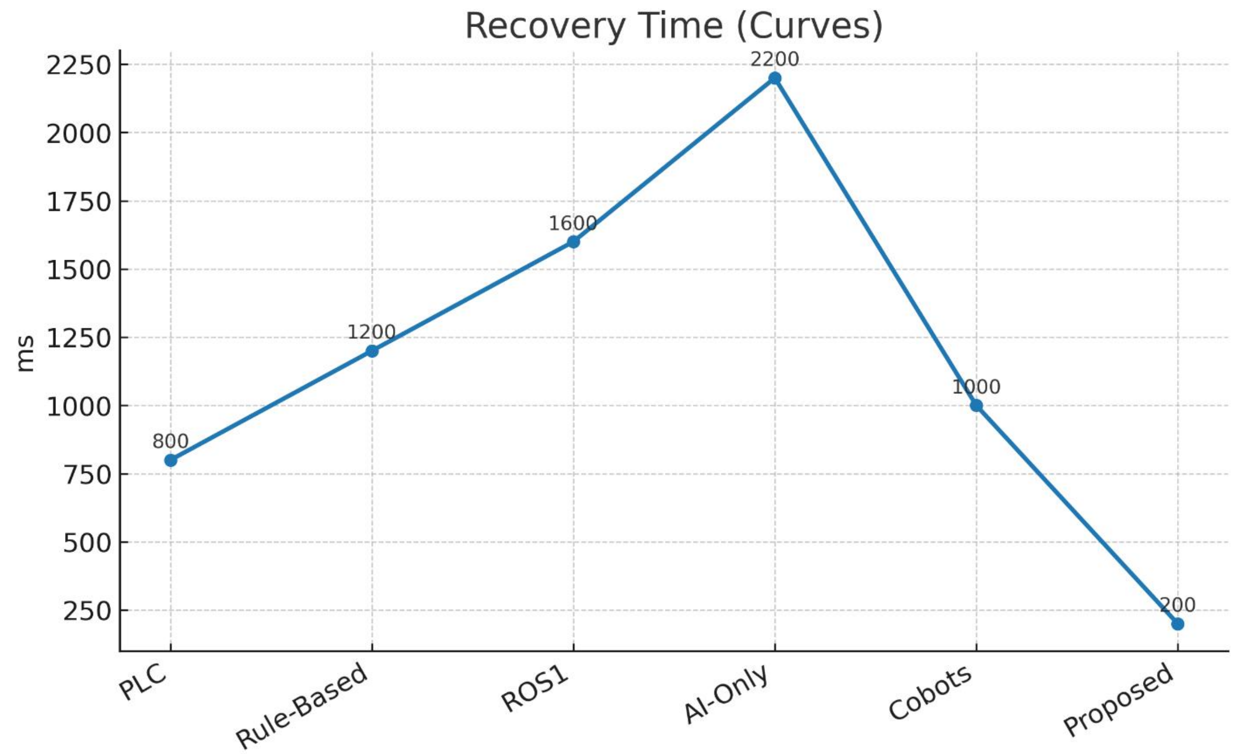

Figure 11. Comparative analysis of proposed vs existing framework in latency curve

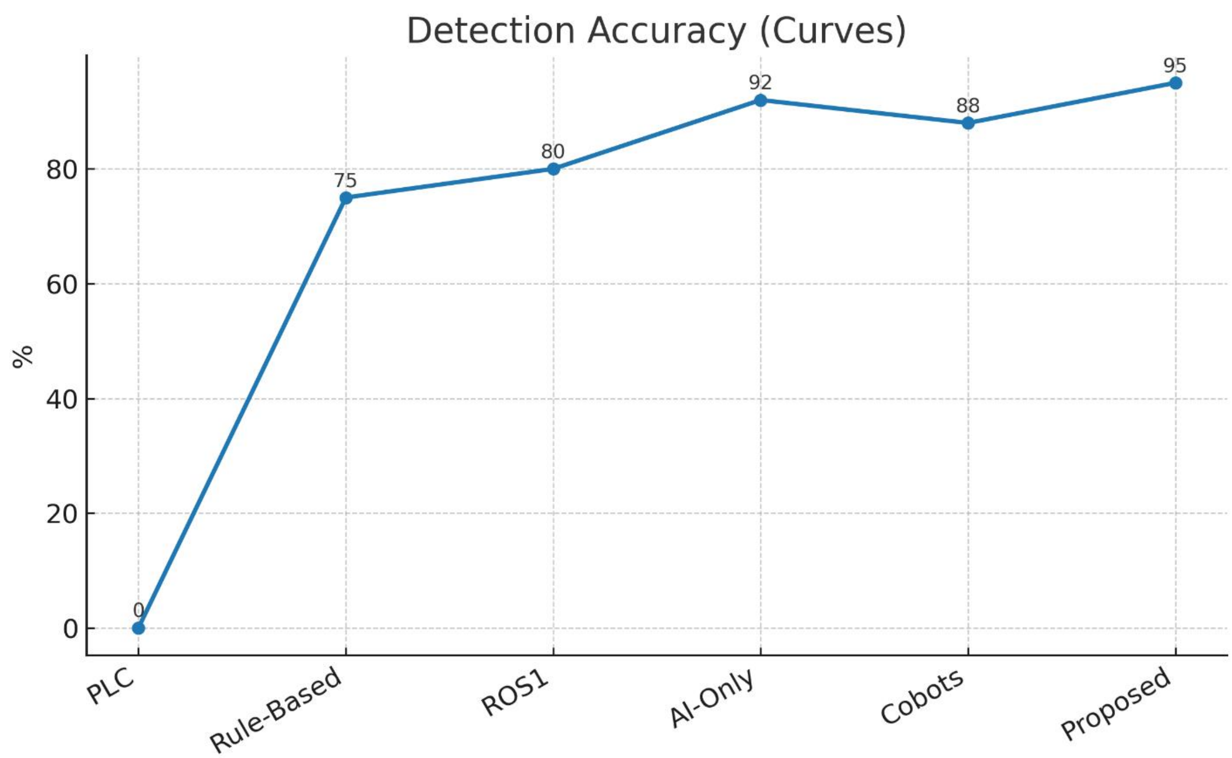

Figure 12. Comparative analysis of proposed vs existing framework in accuracy curve

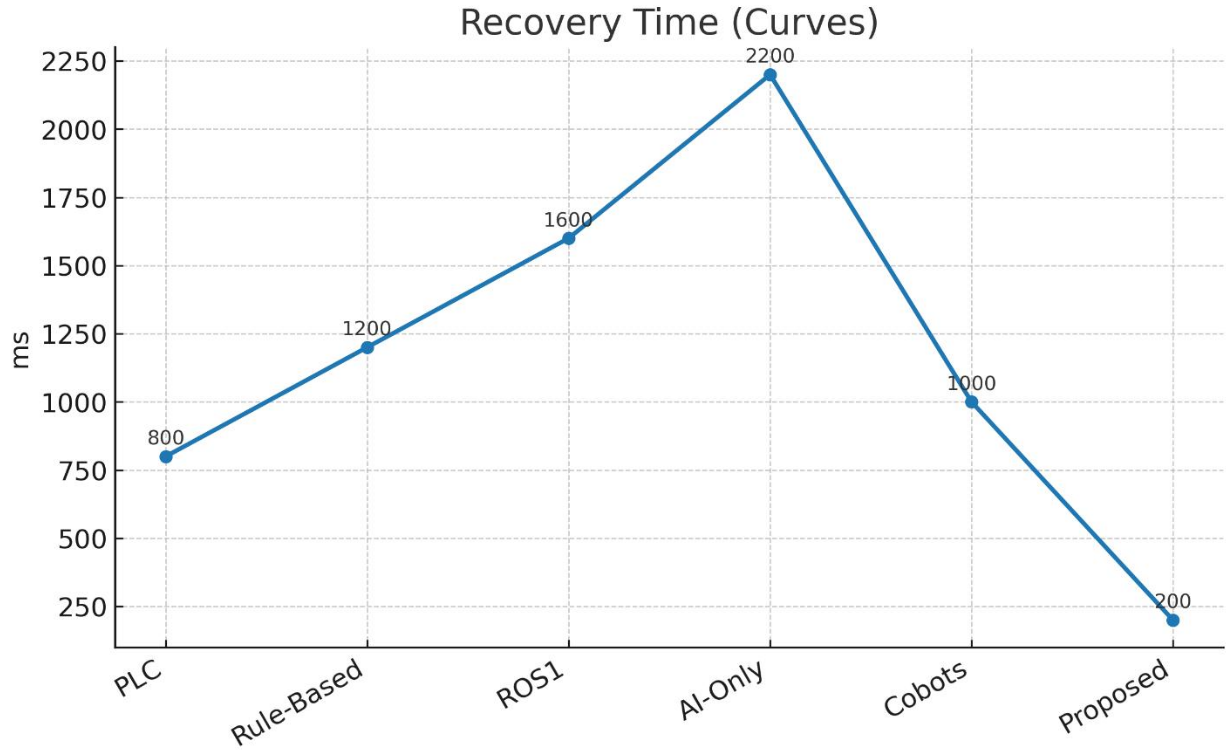

Figure 13. Comparative analysis of proposed vs existing framework in recovery curve

5 Discussion and Conclusion

5.1 Balancing Perception and Determinism

The results show essential contradiction between probabilistic, adaptive character of computer vision and deterministic nature of industrial embedded systems. Although the vision pipeline was accurate in detecting objects and defects, its weakness in terms of lighting and blockage-related changes indicates a more prominent problem in the manufacturing space (Xu et al., 2020). Deterministic control systems require framed latency and certainty that can be defined but the perception system that involves AI is inherently variable because it depends on data quality and model generalization (Bengio et al., 2021). This is a tradeoff leading to the significance of the hybrid system where robotics with embedded electronics enforces the hard deadlines, and vision-based AI offers flexibility to follow unstructured environments (Siciliano & Villani, 2019).

5.2Industrial Relevance of Latency and Synchronization

The latency measurements show that end-to-end perception-to-actuation cycle times were less than 40 ms, adequate to serve medium-paced robotic applications. This is congruent with industry specifications wherein critically fast response times of <50 ms are essential in collaborative robotics in order to avoid mishap cases because of delay (Caccavale et al., 2018). In addition, the EtherCAT synchronization performance indicated worse cycle-time deviations without large deviations even with the increment of node counts. These observations are similar to studies which have shown that large-scale use of Industrial Ethernet protocols to achieve deterministic communication may indeed be feasible with synchronization errors well-constrained (Decotignie, 2018). In reality, this implies that the way the system is organized may expand to a multi-robot cell or decentralized production line without affecting safety or accuracy (Schneider & Kohn, 2020).

5.3 Safety as a Core Design Imperative

Testing of safety followed all ISO and IEC requirements, and exhibited safe functionality of all functions, including Safe Torque Off (STO) and Emergency Stop. This underlines the concept that robotics incorporation of advanced AI-based perception can never disregard safety compliance. Recent research pointed to the fact that safety is the major impediment to the implementation of AI-enhanced robots in human-intensive production processes (Haddadin & Croft, 2016). In the case of collaborative robots, the machinery needs complex speed and separation monitoring related to safely working along with a human being, and the performance of plane rectification as measured in the study shows that the functionality can be present in modern embedded control mechanisms (Villani et al., 2018). These results are consistent in the larger trends involving developing so-called safety envelopes by surrounding AI-controlled systems with mechanisms that prevent the unsafe behaviour in the event of a failure in perception (Zanchettin et al., 2021).

5.4 Robustness and Fault Recovery

Resilience is critical in industrial automation as the great recovery of the system with the fault injections experiments show. Spring-time failures were vision-related, including momentary loss of camera screens, and although they had a longer recovery durationrange, they were still highly successful, reflecting a common finding that perception-based systems are extremely vulnerable to noise and occlusion (Zhou et al., 2019). The observation that transient communication-related upsets, such as communication packet loss and IM-jitter events in DDS/ EtherCAT, did result in rapid detection and recovery corroborates existing evidence that effective middleware and fieldbus standards can overcome transient network hits (Bonci et al., 2019). Such resilience is particularly important in smart factories where a variety of devices and robots communicate via shared infrastructure with the information failure in one system potentially spreading to another unless contained (Uhlemann et al., 2017).

5.5 Vision in Diverse Industrial Domains

Break-down of accuracy gave very good results in metal, electronics and pharmaceuticals with not so good results in textile bags. This finding aligns with past investigations, which state that such highly deformable and textured materials can be insufficiently captured by the machine vision because of their irregular surface texture and lighting variability (Mak & Peng, 2020). Energy-intense and reflective substances, such as metals and PCBs can be more easily automated in the inspection process since they are consistent in their geometric conditions (Zhang et al., 2018). These results further underline the capability of designing vision pipelines to be specific to a domain and possibly incorporating multispectral imaging or sensor fusion in areas where normal RGB vision systems do not perform well enough (Gao et al., 2019).

5.6 Implications for Industry 4.0 and Smart Manufacturing

The perception, control, and communication integration in this system corresponds to the Industry 4.0 principles where autonomous operations of cyber-physical systems with the possibility of interoperability is needed (Lu, 2019). The OPC UA interoperability demonstrated with MQTT can be considered in line with what the industry is currently working at, the "unified namespace" that leverages consistency in machine-to-machine and machine-to-cloud communication (Scholten & Smit, 2016). Integration in real time and predictive analytics can be enabled; this can minimize downtime and maximize throughput and enable adaptive manufacturing (Kang et al., 2016). In addition, reliability of embedded control and communications assist in the developing trend towards human-robot collaboration, in which robots must detect dynamic tasks over time without affecting predictability (Bdiwi et al., 2017).

5.7 Limitations and Future Research

Nevertheless, despite its many strengths, the system has limitations, which are in line with the issues in the general field. To begin with, the use of acquired vision with CNNs results in the problem of interpretability and validation that are still not solved in the safety-critical systems (Samek et al., 2017). Second, although able to recover in the simulated faults, compounded failures, i.e. simultaneous perception blowup and communication breakdown, may occur in the real world and necessitate more complex redundancy measures (Caputo et al., 2019). Finally, scalability to industrial networks of large size can open up bottlenecks not encounterable in the current assessment. Future work would be integrating edge-cloud cooperation such as offloading higher-level optimization tasks to the cloud and retaining low-latency control locally (Wan et al., 2016). Yet another promising direction is the incorporation of reinforcement learning in the adaptive task planning when such task constraints are mathematically expressible (Kormushev et al., 2013).

5.8 Theoretical and Practical Contributions

Efficacy-wise, the paper forms part of the efforts to fill the gap between adaptive perception and deterministic control, providing an experimental example of how both paradigms can be utilized in a single industrial context. In practice, the results would give engineers latency budgets, performance metrics, and safety verification approaches that could be used to design robotic cells in the future. This makes industrial forces - not just application-specific constraints - contextualized results and as such it can be concluded that the research demonstrated the possibility to implement AI-driven robotics at larger scales in smart-factories without violating the functional safety frameworks.

Acknowledgment: The authors gratefully acknowledge the support and facilities provided by the Telecommunication Research Center of the Electrical Engineering Department of Bahauddin Zakariya University, Multan, Pakistan, throughout this research endeavor. Overall, all the authors would like to thank their affiliated universities for supporting this research.

Funding Statement: The author(s) received no specific funding for this study.

Author Contribution: The author's contributions to this paper are as follows: **study conception and design**: K.T. Mehmood; **data collection**: K.T. Mehmood; **analysis and interpretation of results**: K.T. Mehmood; **draft manuscript preparation**: K.T. Mehmood, R.Iqbal. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data with the corresponding author can be provided upon appropriate request.

Ethics Approval: Not applicable.

Conflict of Interest: The authors declare that they have no conflicts of interest to report regarding the present study

References

1) Baheti, R., & Gill, H. (2011). Cyber-physical systems. *The Impact of Control Technology*, 12(1), 161–166.

2) Basu, K., & Pal, S. (2018). Design of energy-efficient embedded systems for real-time robotics. *Journal of Systems Architecture*, 90, 1–14.

3) Bdiwi, M., Pfeifer, M., Sterzing, A., & Fechter, M. (2017). Human-robot collaboration – state of the art and future research. *Robotics and Computer-Integrated Manufacturing*, 45, 1–14.

4) Bengio, Y., Lecun, Y., & Hinton, G. (2021). Deep learning for AI. *Communications of the ACM*, 64(7), 58–65.

5) Bonci, A., Pirani, M., & Callegati, F. (2019). Reliable communication in industrial Internet of Things: A DDS-based approach. *IEEE Transactions on Industrial Informatics*, 15(2), 1206–1216.

6) Caccavale, F., Villani, L., & Siciliano, B. (2018). Safety in human–robot cooperation. *Annual Reviews in Control*, 46, 149–163.

7) Caputo, B., Sim, K. B., & Ristic, B. (2019). Fault tolerance in cyber-physical systems for manufacturing. *Journal of Manufacturing Systems*, 51, 12–22.

8) Chen, F., & Wang, J. (2020). Hybrid architectures for robust autonomous robotics. *Robotics and Autonomous Systems*, 131, 103582.

9) Chen, Y., Zhang, C., & Zhang, J. (2019). Computer vision-enabled welding inspection in automotive assembly. *IEEE Transactions on Industrial Informatics*, 15(7), 3896–3905.

10) Decotignie, J. D. (2005). The many faces of industrial Ethernet [Invited Paper]. *IEEE Industrial Electronics Magazine*, 3(1), 8–19.

11) Decotignie, J.-D. (2018). The role of time-deterministic communications in Industry 4.0. *IEEE Transactions on Industrial Informatics*, 14(9), 4113–4121.

12) Forsyth, D., & Ponce, J. (2012). *Computer Vision: A Modern Approach*. Pearson.

13) Gao, Y., Zhang, H., & Xie, S. (2019). Multispectral and hyperspectral imaging for defect detection in food and agricultural products. *Journal of Food Engineering*, 233, 1–13.

14) Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

15) Groover, M. P. (2008). *Automation, Production Systems, and Computer-Integrated Manufacturing*. Pearson.

16) Haddadin, S., & Croft, E. (2016). Physical human–robot interaction. *Springer Handbook of Robotics*, 1835–1874.

17) Haddadin, S., Albu-Schäffer, A., & Hirzinger, G. (2012). Requirements for safe robots: Measurements, analysis and new insights. *The International Journal of Robotics Research*, 31(13), 1578–1600.

18) Hu, J., Sun, Y., & Zhang, H. (2018). Embedded GPU-based vision processing for autonomous robots. *Microprocessors and Microsystems*, 61, 62–71.

19) IEC. (2010). *IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems*. International Electrotechnical Commission.

20) International Organization for Standardization (ISO). (2011). *ISO 10218-1: Robots and robotic devices—Safety requirements for industrial robots—Part 1: Robots*. Geneva: ISO.

21) Johansson, R., & Robertsson, A. (2017). Human-robot collaboration based on vision and force sensing. *Annual Reviews in Control*, 44, 203–213.

22) Kagermann, H., Wahlster, W., & Helbig, J. (2013). Recommendations for implementing Industry 4.0. *Final Report of the Industrie 4.0 Working Group*.

23) Kang, H. S., Lee, J. Y., Choi, S., Kim, H., Park, J. H., Son, J. Y., ... & Noh, S. D. (2016). Smart manufacturing: Past research, present findings, and future directions. *International Journal of Precision Engineering and Manufacturing-Green Technology*, 3(1), 111–128.

24) Kehoe, B., Patil, S., Abbeel, P., & Goldberg, K. (2015). A survey of cloud robotics and automation. *IEEE Transactions on Automation Science and Engineering*, 12(2), 398–409.

25) Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11), 1238–1274.

26) Kopetz, H. (2011). *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Springer.

27) Kormushev, P., Nenchev, D. N., Calinon, S., & Caldwell, D. G. (2013). Upper-body kinesthetic teaching of a free-standing humanoid robot. *IEEE International Conference on Robotics and Automation (ICRA)*, 3970–3975.

28) Kragic, D., & Vincze, M. (2009). Vision for robotics. *Foundations and Trends in Robotics*, 1(1), 1–78.

29) Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 1097–1105.

30) Kumar, A. (2008). Computer-vision-based fabric defect detection: A survey. *IEEE Transactions on Industrial Electronics*, 55(1), 348–363.

31) Kumar, P., Singh, M., & Kaur, H. (2021). Robotics and industrial automation: A review. *Materials Today: Proceedings*, 46, 8333–8338.

32) LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.

33) Lee, E. A., Seshia, S. A., & Pappas, G. J. (2015). *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*. MIT Press.

34) Lee, J., Bagheri, B., & Kao, H. A. (2017). A cyber-physical systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters*, 3, 18–23.

35) Lu, Y. (2017). Industry 4.0: A survey on technologies, applications and open research issues. *Journal of Industrial Information Integration*, 6, 1–10.

36) Lu, Y. (2019). Industry 4.0: A new vision for modern manufacturing. *Journal of Manufacturing Systems*, 54, 1–10.

37) Mahmoud, M., Moftah, H., & Kandil, N. (2020). OPC UA and MQTT-based communication for Industry 4.0. *Journal of Industrial Integration and Management*, 5(2), 123–138.

38) Mak, K. L., & Peng, P. (2020). Automated visual inspection of textile defects. *Textile Research Journal*, 90(3–4), 247–262.

39) Maruyama, Y., Kato, S., & Azumi, T. (2016). Exploring the performance of ROS2. In *Proceedings of the International Conference on Embedded Software* (pp. 1–10).

40) Monostori, L. (2014). Cyber-physical production systems: Roots, expectations and R&D challenges. *Procedia CIRP*, 17, 9–13.

41) Pimentel, A. D., Erbas, C., & Polstra, S. (2015). Embedded systems design: Co-design methodologies. *IEEE Design & Test of Computers*, 22(2), 101–113.

42) Quigley, M., et al. (2009). ROS: An open-source robot operating system. *ICRA Workshop on Open Source Software*, 3(3.2), 5.

43) Rosen, R., Wichert, G., Lo, G., & Bettenhausen, K. D. (2015). About the importance of autonomy and digital twins for the future of manufacturing. *IFAC-PapersOnLine*, 48(3), 567–572.

44) Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K., & Müller, K. R. (2017). Explainable AI: Interpreting, explaining and visualizing deep learning. *Springer Lecture Notes in Computer Science*.

45) Schneider, P., & Kohn, A. (2020). Ethernet-based real-time communication for industrial automation. *Control Engineering Practice*, 101, 104498.

46) Scholten, H., & Smit, M. (2016). A unified architecture for OPC UA and MQTT in Industry 4.0 environments. *Journal of Industrial Integration*, 2(3), 27–35.

47) Siciliano, B., & Khatib, O. (Eds.). (2016). *Springer Handbook of Robotics* (2nd ed.). Springer.

48) Sundararajan, V., Subramanian, A., & Jayaraman, S. (2018). Challenges in vision-based robotics: Reliability and safety concerns. *Robotics Today*, 12(4), 88–99.

49) Szeliski, R. (2022). *Computer Vision: Algorithms and Applications* (2nd ed.). Springer.

50) Tan, K. C., Lim, K. Y., & Xu, X. (2016). Vision-based quality inspection in electronics manufacturing. *Journal of Manufacturing Systems*, 41, 179–190.

51) Uhlemann, T. H.-J., Lehmann, C., & Steinhilper, R. (2017). The digital twin: Realizing the cyber-physical production system. *Procedia CIRP*, 61, 335–340.

52) Villani, V., Pini, F., Leali, F., & Secchi, C. (2018). Survey on human–robot collaboration in industrial settings: Safety, interaction and trust. *Robotics and Computer-Integrated Manufacturing*, 61, 101849.

53) Wang, L., Gao, R. X., & Váncza, J. (2020). A review of smart manufacturing and future trends. *Journal of Manufacturing Systems*, 54, 305–316.

54) Willig, A. (2008). Recent and emerging topics in wireless industrial

communications: A selection. *IEEE Transactions on Industrial Informatics*, 4(2), 102–124.

55) Xu, X., Sun, H., & Li, Y. (2020). Robust computer vision algorithms for industrial inspection. *Journal of Manufacturing Processes*, 56, 98–109.

56) Zanchettin, A. M., Rocco, P., & Ding, H. (2021). Safety of collaborative robots: Design and control issues for safe human–robot interaction. *Annual Reviews in Control*, 52, 265–281.

57) Zhang, Y., Yang, G., & Zhu, Y. (2018). Automated inspection of printed circuit boards using deep learning and machine vision. *IEEE Access*, 6, 12885–12895.

58) Zhang, Z., Li, B., & Xu, W. (2019). Deployment of deep learning-based vision models on embedded platforms for robotics. *Sensors*, 19(24), 5465.

59) Zhou, Y., Wang, L., & Gao, R. X. (2019). Sensor fault detection and recovery in cyber-physical systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(1), 165–176.

60) Zuehlke, D. (2010). SmartFactory—Towards a factory-of-things. *Annual Reviews in Control*, 34(1), 129–138.