Electronics and Information Technologies, 2026, 35, 121–130
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.93:621.3.049.77

DISTRIBUTED ANOMALY DETECTION SYSTEM IN MANUFACTURING USING EDGE AI AND COMPUTER VISION

***Olha Shymchyshyn*[1]*, Kostiantyn Pidhornyi*[1]*, Maryan Shymchyshyn*[2]
[1] *Lviv Polytechnic National University,*
12 Stepana Bandery Street, Lviv, 79013, UKRAINE
[2] *Lviv Branch of Private Higher Education Establishment "European University",*
5 Kushevycha Street, Lviv, 79019, UKRAINE

Shymchyshyn, O., Pidhornyi, K. & Shymchyshyn, M. (2026). Distributed Anomaly Detection System in Manufacturing Using Edge AI and Computer Vision. *Electronics and Information Technologies*, 35, 121–130. https://doi.org/10.30970/eli.35.9

ABSTRACT

Background. Modern automated production facilities are characterized by a high level of robotization and the drive to minimize human involvement in routine operations. At the same time, the complete elimination of the operator is unacceptable due to safety requirements and the need for decision-making in the event of anomalies. The use of centralized computer vision systems results in significant network load, limiting their scalability. One promising direction for solving this problem is the application of Edge AI technology.

Materials and Methods. A three-tier architecture for an anomaly detection system is proposed. The first tier is implemented based on the ESP32-P4 edge nodes with digital cameras, which perform local object detection using quantized computer vision models. Only structured metadata is transmitted to the central cluster via the CAN bus. The second tier performs motion trajectory analysis based on root mean square (RMS) deviation. The third tier implements an operator notification subsystem with automatic generation of an event video replay.

Results and Discussion. Offloading object detection to edge nodes significantly reduces network load and ensures system scalability regardless of the number of cameras. The use of the ESP32-P4 enables real-time video processing at up to 30 frames/s with a power consumption of less than 500 mW per node. RMS analysis ensures the detection of deviations from the reference trajectory in less than 1 s. After an anomaly is recorded, the operator receives a push notification and a 5-second video clip for rapid situation assessment.

Conclusion. The proposed architecture demonstrates the feasibility of practical Edge AI utilization for building scalable production process monitoring systems. The combination of local computer vision, transmission of structured data via the CAN bus, and centralized statistical analysis ensures continuous technological process control, reduces network load, and shortens response time to anomalous events. Further research will focus on the adaptive tuning of RMS thresholds and the implementation of federated learning for the models.

Keywords: edge computing, computer vision, ESP32-P4, anomaly detection, CAN bus, industrial automation

INTRODUCTION

Modern automated manufacturing enterprises increasingly operate with a minimal staff presence on the production floor. Robotic manipulators, conveyors, and automated guided vehicles (AGVs) perform the majority of physical operations, while personnel take

© 2026 Olha Shymchyshyn et al. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

on supervisory functions. However, the complete elimination of a responsible person is considered unacceptable both from a safety perspective and for regulatory reasons [1, 2].

The human-computer interaction scientific literature emphasizes the need for reliable two-way collaboration frameworks that allow operators to identify system model biases and increase decision-making reliability [3]. Traditional closed-circuit television (CCTV) systems require constant human attention and scale poorly for enterprises with dozens or hundreds of cameras. Centrally processed computer vision (CV) systems running on a server create significant network load when processing high-framerate video streams from multiple cameras simultaneously. This contradiction between automation requirements and the need for human supervision creates the demand for a system that combines continuous machine monitoring while retaining the operator as the ultimate arbiter upon detecting deviations.

The Edge AI approach, which involves performing data processing using machine learning models directly on embedded devices close to the data source, significantly reduces network load and processing latency, thereby ensuring system scalability [4]. Provided that object detection is performed directly on peripheral camera nodes, only compact structured metadata — specifically object identifiers, bounding box coordinates, and timestamps — are transmitted to the central cluster. This approach substantially reduces the volume of transmitted data and lowers network bandwidth requirements. The implementation of this paradigm has been made possible by the emergence of microcontrollers with specialized hardware accelerators for neural network computations, notably the ESP32-P4 with a dual-core RISC-V architecture and an integrated Image Signal Processor (ISP) accelerator. The purpose of this work is to develop a distributed computer vision system architecture for real-time anomaly detection in production objects based on edge data processing (Edge AI) and centralized cluster analysis with operator notification generation.

MATERIALS AND METHODS

General System Architecture

The proposed system comprises three hierarchical levels, as illustrated in **Fig. 1**. The first level consists of distributed edge nodes deployed across the enterprise. The second level is a centralized data processing cluster. The third level represents the operator notification subsystem.

Level 1 — Edge Nodes

Each edge node is implemented as an intelligent computer vision module consisting of a digital camera and an ESP32-P4 microcontroller. The primary purpose of the node is local video stream processing, object detection, and transmitting only compact structured metadata to the central cluster instead of the raw video stream.

The choice of the ESP32-P4 platform is driven by its architectural features focused on edge data processing tasks. The microcontroller is built on two 32-bit RISC-V cores with a clock frequency of up to 400 MHz and features an integrated Image Signal Processor (ISP) module that performs hardware preprocessing of video data. Optimized ESP-NN libraries are used to accelerate neural network model execution, enabling efficient execution of quantized models using integer arithmetic (INT8). This approach ensures real-time local execution of computer vision models with low power consumption of the edge node [5].

Object detection is performed using pre-trained and quantized computer vision models deployed directly on the edge node. Model training is carried out on external computational resources using representative samples of a specific production process, after which the model is optimized and converted for execution on the microcontroller.

The choice of the neural network architecture is determined by the specifics of the production process, the required detection accuracy, and the available computational resources of the edge node. Since Edge AI operates under constrained RAM and power

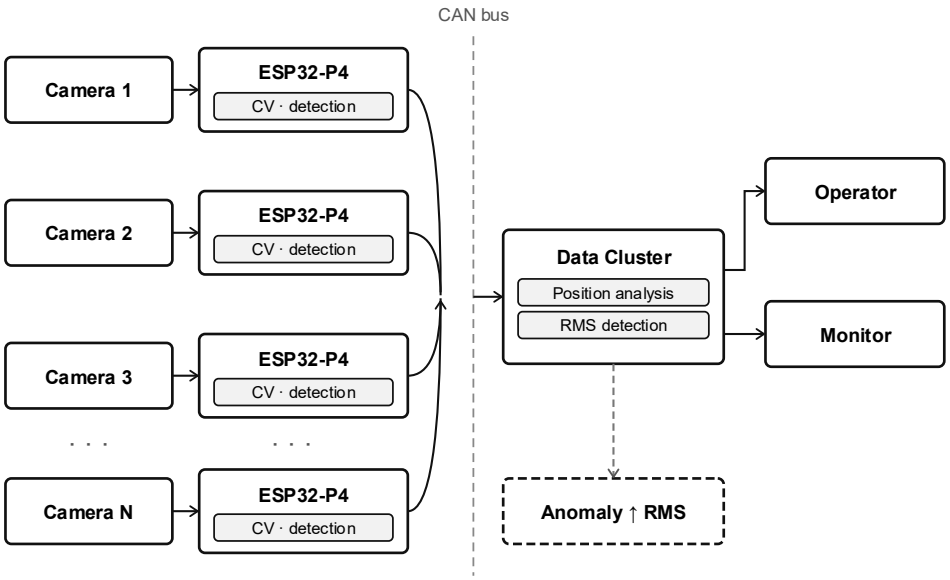


Fig. 1. Three-tier distributed architecture of the Edge AI-based anomaly detection system: (1) edge nodes (camera + ESP32-P4); (2) CAN bus; (3) data processing cluster; (4) operator notification system

consumption conditions, key criteria for model selection include performance, processing latency, object localization accuracy, and the amount of PSRAM used.

Comparative characteristics of the most common models suitable for implementation on the ESP32-P4 platform are presented in **Table 1**. The presented characteristics are generalized based on data [6–8] and may vary depending on the model configuration, input image resolution, and hardware platform parameters. As can be seen from **Table 1**, the use of more complex models ensures higher object localization accuracy, but is accompanied by an increase in processing latency and memory consumption. For high-speed production lines, the application of specialized regression models or NanoDet-m is appropriate, whereas for tasks involving complex scenes with a large number of objects, models of the YOLO family are more suitable. Therefore, the choice of a specific model is

Table 1. **Comparative characteristics of computer vision models for implementation on edge nodes**

Model	FPS	Latency, ms	Localization accuracy	PSRAM load	Recommended scenario
Custom Regressor	~45 – 60	< 20	Medium	Minimal	Fast lines, single stable object.
NanoDet-m	~18 – 25	40 – 55	High	Medium	Optimal for sorting parts.
SSD Lite (MobNetV2)	~12 – 18	60 – 80	High	High	Standard industrial tasks.
YOLOv5n	~9 – 14	70 – 110	Very high	High	Detection of multiple small objects.
YOLOv8n	~6 – 10	110 – 160	Very high	Very high	Complex scenes where accuracy is critical.
YOLO26n	~12 – 16	65 – 85	Extremely high	Very high	Precise coordinates + OBB.

determined by a trade-off between performance, object localization accuracy, processing latency, and the amount of available hardware resources of the edge node.

Inter-level Communication

To organize data exchange between edge nodes and the central processing cluster, the Controller Area Network (CAN) bus was chosen. This choice is driven by a number of characteristics critical for the system's operation in an industrial environment. Specifically, CAN ensures deterministic message delivery with a bounded maximum latency, which is essential for real-time systems. In addition, the protocol is characterized by high immunity to electromagnetic interference, the presence of a built-in multi-master arbitration mechanism without the need for network switches, and the widespread availability of hardware implementations and industrial transceivers [9].

The system utilizes the CAN 2.0B standard with an extended 29-bit identifier, allowing for flexible addressing of nodes and message types. Data transmission is carried out in frames of up to 8 bytes in length, the structure of which is optimized for transmitting compact metadata obtained at the edge level. The frame data field transmits: node identifier (1 byte), object class (1 byte), normalized X and Y centroid coordinates (2 bytes each, Q8.8 fixed-point format), and a relative timestamp (2 bytes). Representing coordinates in the Q8.8 fixed-point format provides a trade-off between positioning accuracy and message compactness, which is important for CAN networks with a small frame size. At a detection frequency of 20 frames/s, a single node generates about 160 bytes/s of overhead data, whereas transmitting an uncompressed VGA video stream requires over 18 MB/s. Thus, using local detection reduces network traffic by more than a factor of 10^5 (the order of magnitude depends on resolution, frame rate, and compression parameters).

Given that the CAN bus operates as a broadcast medium without built-in security protocols, addressing network security and message integrity is paramount, particularly when transmitting safety-critical data. In industrial environments, unprotected CAN frames are vulnerable to message spoofing, frame injection, and replay attacks. To mitigate these threats, the system architecture incorporates node authentication mechanisms, such as Secure On-Board Communication (SecOC) or Hash-based Message Authentication Codes (HMAC), appended to critical metadata payloads. This is especially vital when the CAN bus is leveraged for functional safety functions, such as transmitting emergency stop (E-stop) signals or monitoring hazardous operator zones. Ensuring data integrity and authenticating edge node identities guarantees that emergency overrides cannot be maliciously intercepted or altered, complying with industrial functional safety standards.

The proposed message structure allows transmitting only the information necessary for further analysis without forwarding the video stream. This significantly reduces the network load and ensures system scalability as the number of edge nodes increases. Unlike transmitting the full video stream, the proposed approach involves transmitting exclusively structured results of local processing, which reduces the volume of network traffic by several orders of magnitude. As a result, the performance of the central cluster is practically independent of the number of cameras, since the server analyzes only compact object descriptions rather than primary images.

Level 2 — Data Processing Cluster and Anomaly Detection

For each tracked object, the reference trajectory is formed during the system calibration stage under normal operating conditions. During calibration, the object's coordinates are recorded over several cycles of normal production line operation. The reference trajectory is generated by averaging the corresponding trajectories obtained during these cycles and is used as a baseline model of normal object motion. The use of a baseline model representing normal process behavior is consistent with the general principles of statistical process monitoring [10].

Anomaly detection is carried out by comparing the current coordinates of the object with the corresponding reference values. For each frame *i* within a sliding time window of duration τ, the instantaneous deviation $d(t_i)$ is computed as the Euclidean distance between the actual coordinates (x_{actual}, y_{actual}) and the planned reference coordinates $(x_{planned}, y_{planned})$:

$$d(t_i) = \sqrt{\left[\left(x_{planned} - x_{actual}\right)^2 + \left(y_{planned} - y_{actual}\right)^2\right]}.$$

The root mean square (RMS) deviation $RMS(t)$ is evaluated over a sample of $N = \tau \cdot f$, measurements (where f is the detection frame rate) within the sliding window τ:

$$RMS(t) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[d(t_i)]^2}.$$

The selection of the anomaly detection threshold RMS_{thresh} is grounded in statistical process control principles [10]. During the system calibration phase under standard operating conditions, the baseline distribution of the sliding RMS value is evaluated to calculate its historical mean μ_{RMS} and standard deviation σ_{RMS}. The detection threshold is then defined using the k-sigma statistical rule:

$$RMS_{thresh} = \mu_{RMS} + k \cdot \sigma_{RMS},$$

where k is a tuning parameter (typically $k \in [3, 6]$) that determines the system's sensitivity trade-off. Increasing k lowers the False Positive Rate (FPR), preventing nuisance alerts caused by minor vibration or lighting changes, but increases the risk of False Negatives (FN). Conversely, a smaller k maximizes sensitivity to micro-deviations at the expense of potential false alarms. The duration of the sliding window τ (and corresponding sample size $N = \tau \cdot f$) acts as a low-pass filter: longer windows smooth out transient sensor noise, whereas shorter windows ensure sub-second anomaly response times.

Exceeding the set RMS threshold ($RMS(t) > RMS_{thresh}$) is interpreted as an abnormal state of the object.

Simultaneously with processing the positional data, the cluster maintains a video ring buffer that receives a parallel low-bitrate H.264 video stream. This approach minimizes network bandwidth requirements while preserving the ability for subsequent visual verification of events. If an anomaly is detected, the corresponding video fragment, defined by the time interval from 2 s before the event to 3 s after it, is automatically extracted from the ring buffer and associated with the incident record in the system. This ensures the formation of an evidence base and simplifies the analysis of the causes of anomalies.

Level 3 — Operator Notification Subsystem

The third tier of the architecture implements operator decision-making support functions and ensures interaction between the automated anomaly detection system and the responsible person. The primary purpose of the subsystem is rapid notification of deviations from the standard production operating mode, providing tools for visual verification of the event, and storing information for subsequent analysis. Upon receiving a message from the cluster about exceeding the root mean square (RMS) threshold, the subsystem automatically

generates an incident notification. It includes the object identifier, object class, edge node identifier, event time parameters, and the calculated RMS value. The message is transmitted to the operator via a mobile application or another operational alerting channel. Simultaneously with generating the notification, a video recording fragment covering the time interval from two seconds before the anomaly detection moment to three seconds after it is automatically extracted from the video ring buffer. Playing back the corresponding video fragment at the operator station allows for a quick visual verification of the causes of the deviation without reviewing the full video archive. All registered incidents are logged in the event log along with the corresponding metadata, which ensures the possibility of subsequent analysis, system operation auditing, and the accumulation of statistics on the occurrence of anomalous situations. The final decision to confirm or dismiss the notification is made by the operator, which complies with the Human-in-the-Loop principle and allows combining the advantages of automated data processing with human expert evaluation. The operation algorithm of the proposed system is shown in **Fig. 2**.

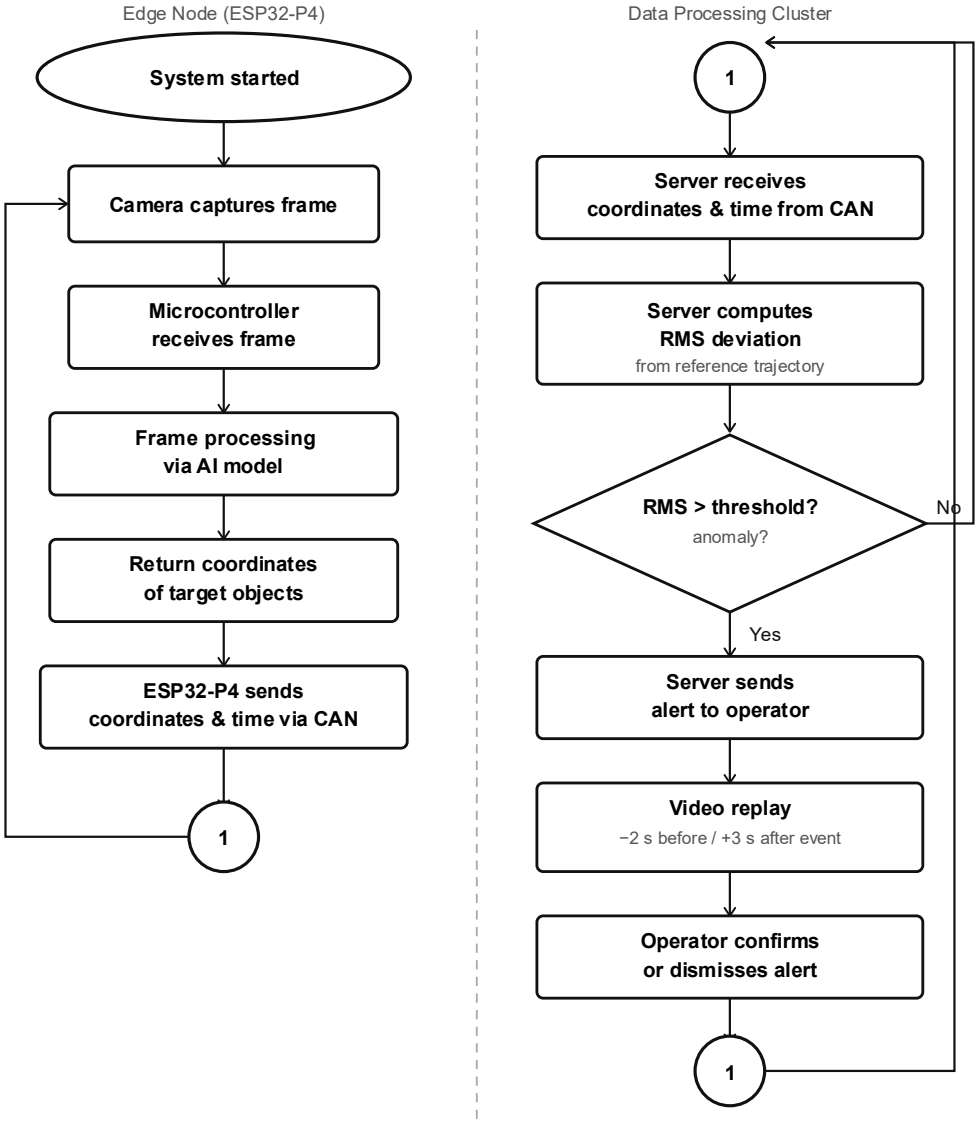


Fig. 2. Operation algorithm of the distributed industrial anomaly detection system

RESULTS AND DISCUSSION

One of the key advantages of the proposed architecture is its ability to horizontally scale without a significant increase in network load. In traditional centralized computer vision systems, each additional camera generates a separate video stream of approximately 50–200 Mbps (depending on resolution and codec), saturating the network interface and cluster computing power even with a relatively small number of cameras [11].

To mitigate this, offloading detection to embedded microcontrollers running quantized integer (INT8) models allows for deterministic metadata transport via industrial buses, which aligns with recent methodologies on microcontroller neural network quantization [12] and modern edge-assisted industrial analytics frameworks [13]. In the proposed architecture, each additional node adds at most ~10 kbps of CAN metadata traffic (**Table 2**). The computational load of the cluster grows only with the number of tracked objects, rather than the number of camera pixels, since object detection is offloaded to the edge.

It should be emphasized that the quantitative performance indicators presented in this study—specifically a local processing frame rate of up to 30 frames/s, power consumption below 500 mW per node, reaction time under 1 s, and network traffic reduction exceeding 10^5 times — are analytical and theoretical estimates. These metrics are calculated based on the technical specifications of the ESP32-P4 SoC (clock frequency, ISP performance, and power envelope), the parameters of quantized INT8 model architectures (**Table 1**), and the bandwidth requirements of standard 29-bit CAN 2.0B frames relative to raw video streams.

Anomaly detection in the proposed system is based on analyzing the deviation of the object's current trajectory from the reference trajectory formed during the production process calibration stage. The RMS deviation, calculated within a sliding time window, is used as the evaluation criterion, which allows accounting for random coordinate fluctuations without losing sensitivity to persistent deviations.

The RMS threshold value is determined based on statistical analysis of data obtained during normal production operation. This approach allows adapting the system to the specifics of a particular technological process and reducing the number of false positives.

Upon exceeding the established threshold, the cluster automatically generates a message to the operator and initiates the playback of a video fragment from the ring buffer, which provides the ability to quickly visually verify the cause of the anomaly.

The use of the Human-in-the-Loop principle makes it possible to combine automatic anomaly detection with the operator's expert assessment. The automatically generated video replay provides the necessary context for making a decision to confirm or dismiss the notification. In addition to rapid response, the incident log can be used for subsequent analysis of deviation causes and the accumulation of training data to improve computer vision models.

Table 2. **Comparison of theoretical/calculated characteristics of the centralized and proposed edge computer vision architecture**

Parameter	Centralized CV	Proposed (Edge AI)
Network traffic volume per camera	50–200 Mbps	< 10 kbps (calculated metadata)
Server scaling	Linear with cameras	Linear with objects
Detection/processing latency	50–500 ms (network + GPU)	< 33 ms (estimated locally at 30 FPS)
Edge node power consumption	– (server-side)	< 500 mW (per technical specs)
Video privacy	Transmitted to the server	Remains on the node

The proposed approach is primarily advisable for modernizing existing production lines equipped with previous-generation equipment that lacks built-in computer vision, self-diagnostics, or intelligent state monitoring tools. In such production systems, the implementation of a distributed architecture based on Edge AI allows realizing automatic monitoring and anomaly detection functions without the need to completely replace technological equipment.

While the analytical results validate the conceptual feasibility and architectural efficiency of the system, practical validation on a physical testbed or experimental production line constitutes a necessary next step of this research. Empirical testing will allow for the evaluation of real-world interference on the CAN bus, actual model latency under varying industrial illumination, and experimental verification of detection thresholds.

CONCLUSION

The paper proposes a three-tier architecture of a distributed industrial anomaly detection system that combines edge data processing using Edge AI, centralized statistical analysis, and an operator decision-making support subsystem. Unlike traditional centralized computer vision systems, the proposed approach involves performing object detection directly on edge nodes, which significantly reduces network traffic and increases system scalability regardless of the number of installed cameras.

It is shown that the use of the CAN industrial bus ensures deterministic exchange of structured metadata between edge nodes and the processing cluster, while the use of root mean square deviation as a criterion for evaluating deviation from the reference trajectory makes it possible to implement a computationally efficient real-time anomaly detection algorithm.

The proposed operator notification subsystem ensures the automatic generation of incident notifications, video fragment playback from the ring buffer, and event logging, which implements the Human-in-the-Loop principle and combines automated analysis with human expert evaluation.

A promising direction for further research is the experimental verification of the proposed architecture on a real production testbed, the development of adaptive methods for tuning anomaly detection thresholds, and the integration of federated learning mechanisms to improve model accuracy without transmitting primary video data outside the enterprise.

ACKNOWLEDGMENTS AND FUNDING SOURCES

This work was supported by a research project funded from the state budget (DB/FORS project: “Functional primary transducers of sensor devices based on liquid crystal materials for use in extreme conditions”, 0126U001141).

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that the research was conducted in the absence of any conflict of interest.

AUTHOR CONTRIBUTIONS

Conceptualization, [K.P., O.S.]; methodology, [M.S.]; validation, [O.S.]; investigation, [K.P.]; writing — original draft preparation, [O.S.]; writing — review and editing, [M.S.]; visualization, [K.P.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Romero, D., & Stahre, J. (2021). Towards the resilient operator 5.0: The future of work in smart resilient manufacturing systems. Procedia CIRP, 104, 1089–1094. https://doi.org/10.1016/j.procir.2021.11.183

[2] Hasan, M. M., Kasedullah, M., Ripon, M. B. B., & Khan, M. M. H. (2025). AI-Driven Quality Control in Manufacturing and Construction: Enhancing Precision and Reducing Human Error. Applied IT & Engineering, 3(1), 1–10. https://doi.org/10.25163/engineering.3110270

[3] Seeber, I., Bittner, E., Briggs, R. O., de Vreede, T., de Vreede, G. J., Elkins, A., Maier, R., Merz, A. B., Oeste-Reiß, S., Randrup, N., Schwabe, G., & Söllner, M. (2020). Machines as teammates: A research agenda on AI in team collaboration. Information & Management, 57(2), 103174. https://doi.org/10.1016/j.im.2019.103174

[4] Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S., & Zomaya, A. Y. (2020). Edge intelligence: The confluence of edge computing and artificial intelligence. IEEE Internet of Things Journal, 7(8), 7457–7469. https://doi.org/10.1109/JIOT.2020.2984887

[5] Espressif Systems. (2025). ESP32-P4 technical reference manual (Rev. v0.4, Pre-release). Espressif Systems (Shanghai) Co., Ltd. https://documentation.espressif.com/esp32-p4-chip-revision-v1.3 technical_reference_manual_en.pdf

[6] Espressif Systems. (2025). ESP-NN: Optimised neural network functions for Espressif chipsets [Software library]. GitHub. https://github.com/espressif/esp-nn

[7] Espressif Systems. (2025). ESP-DL: Espressif deep-learning library for AIoT applications (v3.3.8) [Software framework]. ESP Component Registry. https://components.espressif.com/components/espressif/esp-dl

[8] Lazarevich, I., Kozlov, A., Kaglinskiy, M., & Dukhan, M. (2023). YOLOBench: Benchmarking efficient object detectors on embedded systems. arXiv. https://arxiv.org/abs/2307.13901

[9] International Organization for Standardization. (2024). *Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical coding sublayer* (ISO Standard No. 11898-1:2024). https://www.iso.org/standard/86384.html

[10] Montgomery, D. C. (2019). *Introduction to statistical quality control* (8th ed.). John Wiley & Sons.

[11] Masykuroh, K., Hendrawan, Mulyana, E., & Krishna, F. (2025). A Survey on Video Compression Optimization Techniques for Accuracy Enhancement in Video Analytics Applications (VAPs). IEEE Access, V.13, 75822–75846. https://doi.org/10.1109/ACCESS.2025.3561063

[12] Peluso, V., Cipolletta, A., Calimera, A., Poggi, M., Tosi, F., & Mattoccia, S. (2025). Edge AI in practice: A survey and deployment framework for neural networks on embedded systems. Electronics, 14(24), 4877. https://doi.org/10.3390/electronics14244877

[13] Gauttam, H., Nain, G., Pattanaik, K. K., & Mendes, P. (2025). Edge-AI: A systematic review on architectures, applications, and challenges. *Journal of Network and Computer Applications*, 104375. https://doi.org/10.1016/j.jnca.2025.104375

РОЗПОДІЛЕНА СИСТЕМА ВИЯВЛЕННЯ АНОМАЛІЙ НА ВИРОБНИЦТВІ З ВИКОРИСТАННЯМ ПЕРИФЕРІЙНОГО ШТУЧНОГО ІНТЕЛЕКТУ ТА КОМП'ЮТЕРНОГО ЗОРУ

Ольга Шимчишин[1]*, Костянтин Підгорний[1], Мар'ян Шимчишин[2]
[1]Національний університет «Львівська політехніка»,
вул. Степана Бандери 12, м. Львів, 79013, УКРАЇНА
[2] Львівська філія ПВНЗ «Європейський університет»,
вул. Кушевича 5, м. Львів, 79019, УКРАЇНА

АНОТАЦІЯ

Вступ. Сучасні автоматизовані виробництва характеризуються високим рівнем роботизації та прагненням мінімізувати участь людини у виконанні рутинних операцій. Водночас повне усунення оператора є неприпустимим з огляду на вимоги безпеки та необхідність прийняття рішень у разі виникнення аномалій. Використання централізованих систем комп'ютерного зору призводить до значного мережевого навантаження, що обмежує їх масштабованість. Одним із перспективних напрямів вирішення цієї проблеми є застосування технології периферійного штучного інтелекту (ПШІ, англ. Edge AI).

Матеріали та методи. Запропоновано трирівневу архітектуру системи виявлення аномалій. Перший рівень реалізовано на базі периферійних вузлів ESP32-P4 із цифровими камерами, які виконують локальне детектування об'єктів за допомогою квантованих моделей комп'ютерного зору. До центрального кластера по шині CAN передаються лише структуровані метадані. На другому рівні здійснюється аналіз траєкторій руху на основі середньоквадратичного відхилення (СКВ, англ. RMS — root mean square). Третій рівень реалізує підсистему сповіщення оператора з автоматичним формуванням відеоповтору події.

Результати та обговорення. Перенесення детектування об'єктів на периферійні вузли суттєво зменшує мережеве навантаження та забезпечує масштабованість системи незалежно від кількості камер. Використання ESP32-P4 дозволяє виконувати обробку відеоданих у режимі реального часу зі швидкістю до 30 кадрів/с при енергоспоживанні менше 500 мВт на вузол. Аналіз RMS забезпечує виявлення відхилень від еталонної траєкторії менш ніж за 1 с. Після фіксації аномалії оператор отримує push-сповіщення та відеофрагмент тривалістю 5 с для оперативної оцінки ситуації.

Висновки. Запропонована архітектура демонструє можливість практичного використання ПШІ для побудови масштабованих систем моніторингу виробничих процесів. Поєднання локального комп'ютерного зору, передачі структурованих даних по шині CAN та централізованого статистичного аналізу забезпечує безперервний контроль технологічного процесу, зменшує мережеве навантаження та скорочує час реагування на аномальні події. Подальші дослідження будуть спрямовані на адаптивне налаштування порогів СКВ і впровадження федеративного навчання моделей.

Ключові слова: периферійні обчислення, комп'ютерний зір, ESP32-P4, виявлення аномалій, шина CAN, промислова автоматизація

Received / Одержано	Revision / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
12 August, 2026	14 September, 2026	21 September, 2026	28 September, 2026