Anomaly Detection in Industry 4.0: Supervised vs. Semi-supervised Approaches with OPC-UA integration

Henry O. Velesaca
ESPOL Polytechnic University, Escuela Superior Politécnica del Litoral, ESPOL.
Campus Gustavo Galindo Km. 30.5 Vía Perimetral, P.O. Box 09-01-5863, Guayaquil, Ecuador
hvelesac@espol.edu.ec

William Velesaca
ESPOL Polytechnic University, Escuela Superior Politécnica del Litoral, ESPOL.
Campus Gustavo Galindo Km. 30.5 Vía Perimetral, P.O. Box 09-01-5863, Guayaquil, Ecuador
wvelesac@espol.edu.ec

Abstract

Anomaly detection is critical for maintaining operational stability and preventing unplanned downtimes in industrial systems. This paper explores and compares supervised and semi-supervised deep learning based approaches within industrial systems, leveraging the OPC-UA protocol for data communication and execution of the evaluated algorithms within a finite state machine. This study presents a case study of anomaly detection in a tinplate lids system, demonstrating the performance of both approaches. Integrating with OPC-UA ensures near real-time data access, interoperability, and scalability across diverse industrial environments. Experimental results highlight the strengths and limitations of each method, providing insights into their applicability for modern industrial anomaly detection challenges. Average Accuracy, Execution Time (e.g., CPU and GPU), Round Trip Time, and End-to-End Delay are used to evaluate the performance of the proposed approaches.

1 Introduction

Industrial systems are the backbone of modern manufacturing and production processes, driving efficiency and productivity across various sectors. However, the complexity and scale of these systems also introduce vulnerabilities and challenges, particularly in maintaining operational stability and preventing unplanned downtimes. The identification of anomalies is critical for the prevention of significant problems, as it allows for the mitigation of potential issues at an early stage.

Traditional anomaly detection techniques based on statistical analysis or machine learning have been extensively applied to identify irregular patterns in industrial systems. One of the most common methods is statistical analysis, which involves using various statistical tests and models to determine whether a given data point significantly deviates from the expected behavior. Techniques such as the z-score, which measures how many standard deviations a data point is from the mean, and control charts, which monitor the stability of processes over time, are widely employed ([8], [5]). In the field of techniques based on machine learning, the Support Vector Machines (SVM) stands out as one of the most used. For example, Schölkopf et al. [22] describe how the SVM can be used for anomaly detection by learning the boundary that separates normal data from anomalies. Condition-based maintenance is widely used in industrial settings, employing sensors to monitor critical process variables like temperature and pressure in real time. By analyzing this data with machine learning and setting alert thresholds, potential issues can be identified before impacting product quality, allowing

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

for proactive adjustments and reducing defects, costs, and waste [1]. In contrast, rule-based systems use predefined rules and thresholds developed by experts to detect anomalies, triggering alarms when readings exceed certain limits [26]. While intuitive and easy to interpret, rule-based systems can be rigid, struggle with evolving conditions, and are labor-intensive to maintain, especially as data complexity grows ([20], [4]).

In recent years, deep learning has become a powerful tool for anomaly detection in industrial systems due to its ability to learn and extract complex patterns from large datasets. Autoencoders are a popular method for anomaly detection, identifying anomalies by measuring reconstruction errors from normal data [10]. This approach effectively captures intricate patterns, making it suitable for complex environments [19]. Additionally, recurrent neural networks (RNNs), especially Long Short-Term Memory (LSTM) networks, excel in detecting anomalies in time-series data by modeling temporal dependencies and predicting future values [3]. These techniques have shown great promise in monitoring dynamic industrial processes and providing early warnings of potential failures [7].

This paper explores and compares supervised and semi-supervised deep learning based approaches for anomaly detection in an industrial systems context, with a focus on the use of OPC-UA (Open Platform Communications Unified Architecture). OPC-UA is a widely adopted communication protocol in industrial automation, offering a robust framework for data exchange and interoperability between heterogeneous systems [15]. Also, a case study of the tinplate lids classification system is used to demonstrate the efficiency of the proposed pipelines.

To address this work, the manuscript is organized as follows. Section 2 presents works related to supervised and semi-supervised deep learning based approaches and also introduces OPC-UA. Section 3 presents the proposed pipeline to carry out the classification of *"good"/"defective"* lids. Then, Section 4 shows the experimental results taking as a reference a case study for anomaly detection in tinplate lids using both approaches. Also, OPC-UA Server and Vision Finite State Machine are implemented. Finally, conclusions are presented in Section 5.

2 Related Work

As described above, this paper presents the comparison of supervised and semi-supervised deep learning techniques as key components in the context of industrial control systems. In addition, OPC-UA is presented as a system communication standard that facilitates the collection, integration and analysis of operational data, providing valuable insights for process optimization and predictive maintenance. This section reviews some of the most relevant techniques related to these topics.

2.1 Supervised Approaches

In recent years, deep learning has gained popularity for its ability to automatically learn and extract features from large datasets. Pham et al. [21] present a real-time defect detection system using YOLO v5, ideal for industrial applications due to its high accuracy and speed, though sensitive to lighting and complex designs. Similarly, Beak et al. [2] use YOLO v7 for anomaly detection in cosmetics manufacturing, requiring well-curated training data and potentially needing retraining for new defect types. Klarak et al. [13] extend anomaly detection to defect classification using machine learning, providing detailed insights but requiring extensive labeled data. Kim et al. [12] propose a supervised approach for detecting anomalies in X-ray images of packaged food, effective but reliant on labeled data.

2.2 Not Supervised Approaches

Contrarily to supervised approaches, DRAEM [28] introduces an unsupervised method for surface anomaly detection using discriminatively trained embeddings, excelling in refined anomaly detection but requiring complex training and large datasets. The CutPaste method [16] uses self-supervised learning with synthetic anomalies, effective with minimal labeled data but possibly less accurate for real-world defects. Overall, deep learning models offer significant advantages over classical approaches, including handling complex datasets, reducing the need for manual feature extraction, and better adaptability to new data patterns. On the other hand, MemSeg [27] is a semi-supervised method designed for detecting surface defects in images by leveraging both differences and commonalities in visual data. Unlike traditional fully supervised approaches, MemSeg combines feature extraction with a memory bank of normal samples, enabling the detection of anomalies without requiring extensive defect annotations. By focusing on the distinctive characteristics of normal surfaces during training, the model can accurately identify deviations that signal defects during testing. The system incorporates advanced neural network architectures and anomaly detection techniques to provide a flexible and efficient solution for industrial applications where defect variability is high and labeled data is limited.

2.3 OPC-UA Overview

As the last part of the state-of-the-art review, OPC-UA is reviewed, which has been increasingly adopted in industrial systems for its robust and flexible framework for data exchange and interoperability. Although this standard is known for the characteristics mentioned above, it is not limited to those contexts since it has different parts (e.g., historical access, alarms & conditions, programs, finite state machines, and others) that support other tasks such as generation of datasets, training of models and support for industrial systems. On the other hand, several studies have explored its potential for enhancing anomaly detection. For instance, Lee and Kim [15] developed an OPC-UA-based monitoring system integrating machine learning for detecting anomalies in manufacturing processes, improving maintenance and reducing downtime. Similarly, Vogel-Heuser et al. [25] proposed an architecture using OPC-UA for early fault detection in automated production, leveraging advanced data analytics for better issue diagnosis compared to traditional methods.

On the other hand, in [23] Velesaca et al. explore the implementation of an anomaly detection system in industrial products by combining OPC-UA and deep learning techniques. This approach leverages OPC-UA's robust standardized communication capability to collect real-time data and applies deep learning algorithms to identify complex patterns and detect product defects. Although it presents significant advantages, such as high precision and the capacity for continuous improvement, it also faces challenges related to high computational requirements and the need for large volumes of labeled data. The same research team, in Velesaca et al. [24], propose a methodological approach to industrial system implementation based on OPC-UA, the objective is to propose an 11-steps methodology for the design and implementation of industrial systems using UML notation and then convert it to OPC-UA notation and finally generate an XML to deploy a server.

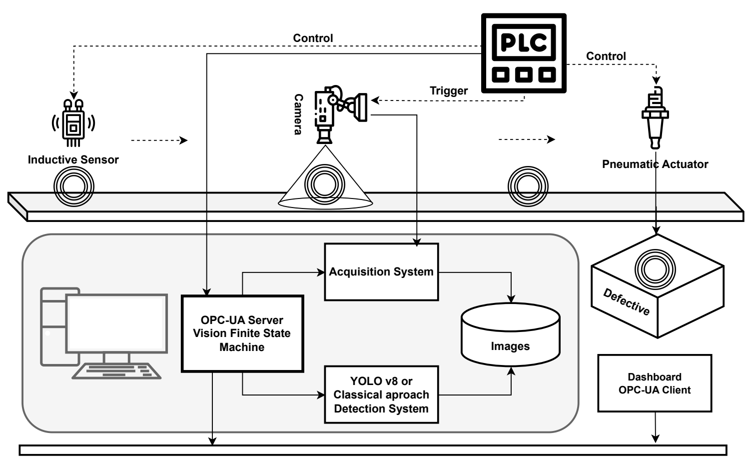


Figure 1: General outline of the system architecture proposed in this paper.

3 Proposed Approach

This section details the stages of the deep learning approaches (supervised and semi-supervised) compared in the current work. The system architecture shows in Fig. 1 is structured around a hierarchical industrial system comprising three layers: a) the control layer, where the primary PLC subsystem manages the plant's operations; b) the supervision layer, where the OPC-UA subsystem retrieves plant status, handles image acquisition from cameras, and applies anomaly detection using both supervised and semi-supervised models; and c) the visualization layer, where operators can monitor the plant's progression through a dashboard. Figure 2 illustrates the pipeline for both deep learning based approaches.

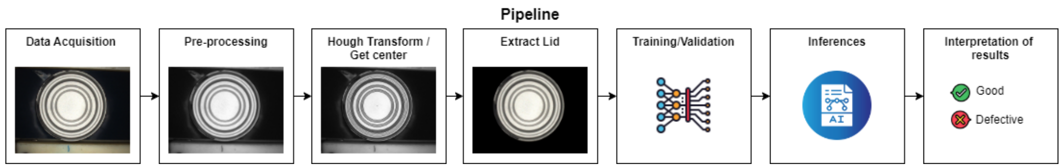


Figure 2: Overall pipelines of the proposed approaches

3.1 Pre-processing

As a first step and before starting the training process, the area that corresponds to the tinplate lid is segmented using the Hough Transform [14] and the background is removed so that in the classification task attention can be paid to the features of the object to be analyzed.

3.2 Supervised Technique

The deep neural network selected for image recognition of the anomaly detection task is YOLO v8 [11]. It has been chosen for its efficiency and speed in detecting objects in images, making it suitable for real-time applications in industrial environments where fast responses are needed. YOLO v8 also offers a deep and flexible architecture that allows easy adjustments and optimizations to adapt to different lighting conditions, viewing angles, and defect types. This makes it a viable option to address the complexity and diversity of defect detection challenges in industrial environments. Furthermore, it is open source and its extensive developer community makes it easy to implement and long-term maintenance of YOLO v8-based defect detection systems. Finally, for this work we consider using the models: *"yolov8m-cls.pt"* and *"yolov8s-cls.pt"* due to the restrictions of resources and low processing times necessary for the proposed solution and the industrial context.

3.3 Semi-supervised Technique

The semi-supervised anomaly detection method selected for surface defect detection is MemSeg [27]. MemSeg has been chosen due to achieves the state-of-the-art (SOTA) performance on MVTec AD datasets [17]. Its ability to effectively detect anomalies in image surfaces by leveraging both similarities and differences between regions, making it particularly suitable for environments where labeled defective samples are limited. This method utilizes a memory bank of normal images, storing relevant features that allow for efficient comparison when evaluating new inputs. MemSeg's structure is flexible, enabling it to adapt to different surface textures, lighting conditions, and defect types, which is crucial in industrial applications where variability in surface characteristics is common. Moreover, MemSeg's semi-supervised nature allows it to generalize well across different scenarios, reducing the need for extensive labeled datasets, which are often difficult to obtain. Given its balance between detection performance and resource efficiency, MemSeg offers a viable solution for real-time anomaly detection in industrial settings, where timely and accurate detection of surface defects is critical.

3.4 OPC-UA Vision Specification

The OPC-UA companion specification for machine vision aims to simplify the integration of machine vision systems into production control and IT systems. Therefore, part of the specification OPC 40100-1 [9] Machine Vision - Control, Configuration management, recipe management, and result management is created. Its purpose is not only to complement or replace existing interfaces between a machine vision system and its process environment with OPC-UA but also to establish new horizontal and vertical integration capabilities. This enables authorized process participants to communicate relevant data with each other, reaching up to the IT enterprise level. Consequently, the OPC-UA Vision interface facilitates the exchange of information between different machine vision systems, PLCs, or any software systems at the control device level accessing the machine vision system. In this work, it is proposed to use a state machine based on this standard to support the computer

Task	**Good**	**Defective**
Training	400	400
Validation	75	75
Testing	37	39
Total	512	514

Table 1: Distribution of data acquisition

vision techniques to be used. Using this specification a server is created and using a Vision Finite State Machine Type, integrating within the "ContinuousExecution" state both the supervised and semi-supervised models.

3.5 Metric Evaluation

For the proposed work, the evaluation of the Average Accuracy over the test set is used to establish the performance of the framework. Furthermore, the execution time using CPU and GPU is computed to measure the efficiency of the used techniques. Finally, Round Trip Time (RTT) and End-to-End Delay (E2E) metrics are calculated. RTT is a measure of the time it takes for a data packet to travel from the point of origin to the destination and then back to the source. E2E refers to the time it takes for information to travel from its source to its destination in a network system. Both metrics are commonly used in computer networks to evaluate latency or network response time [6].

4 Case Study

This section details the practical application of these approaches, consider a manufacturing plant where both supervised and semi-supervised deep learning methods are deployed for anomaly detection testing. Data from various sensors and equipment is collected using OPC-UA and fed into the anomaly detection systems. For overall performance evaluation, the Average Accuracy metric is used. Also, Execution Time using CPU and GPU is calculated to measure the speed of the techniques. Furthermore, in terms of latency, RTT and E2E metrics are used. Finally, one of the main restrictions of the system is that it can process 500 tinplate lids per minute, so the Execution Time should be equal to or less than 75ms.

4.1 System Implementation

This implementation addresses the detection of manufacturing defects in tinplate lids within a factory setting. At the core of the control layer, a Siemens S7 1200 PLC is used. Also, a vision setup comprising an industrial visible spectrum camera and a lighting system, along with an inductive sensor for detecting lids on the conveyor belt. In the supervision layer, a workstation is employed with an Intel Core I9 3.3GHz CPU and NVIDIA Titan XP GPU for training/validation and testing YOLO v8 for image recognition and MemSeg for anomaly segmentation. This workstation is tasked with acquiring images and running the OPC-UA server, YOLO v8 and MemSeg. Figure 3 highlights the key components utilized in the system architecture.

4.2 Dataset Acquisition

As a first step, images of tinplate lids in both *"good"* and *"defective"* conditions are acquired. The most common defects include interior peeling, defective paint, and the missing rubber on the inside edge. Table 1 illustrates the distribution of data used for the training, validation, and testing stages in a later phase with YOLO v8 and MemSeg.

4.3 YOLO v8

Before starting the training phase, the Albumentations library is used to increase the number of training examples. This library allows transformations, such as cropping, rotations, brightness, and contrast adjustments, enhancing the dataset with realistic variations. YOLOv8 models, specifically *"yolov8m-cls.pt"* and *"yolov8s-cls.pt"*, are used for classification tasks, trained with parameters:

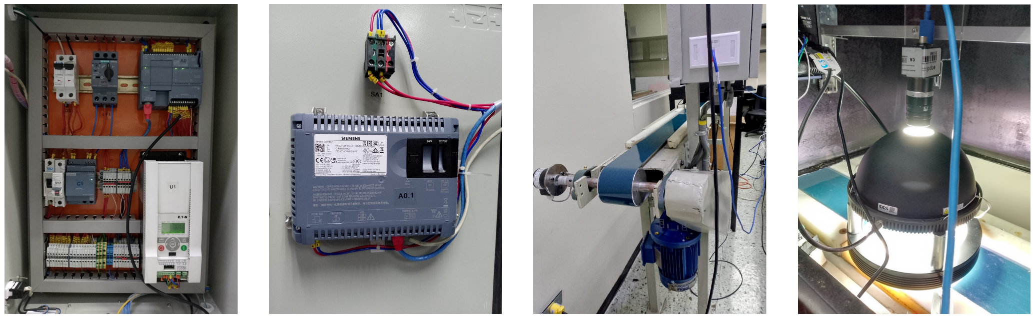
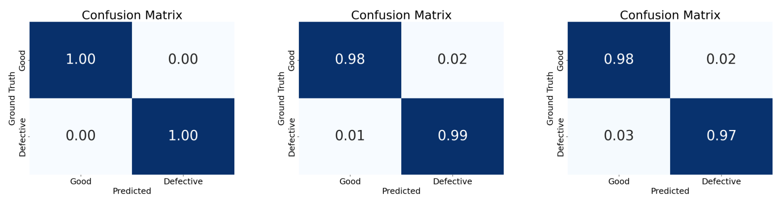


Figure 3: *(1st)* Siemens S7 1200 PLC. *(2nd)* HMI TP700 comfort. *(3rd)* Conveyor belt. *(4th)* Industrial camera and lighting system.



yolov8m-cls.pt *yolov8s-cls.pt* MemSeg

Figure 4: Normalized confusion matrix of each technique calculated on the test set.

epochs=500, imgsz=640, batch=18, pretrained=True, erasing=0.0, flipud=1.0, fliplr=1.0, mixup=0.0, patience=20, close_mosaic=0, auto_augment=autoaugment, mosaic=0.0, optimizer=SGD. After training and validation, normalized confusion matrices (Fig. 4) and Average Accuracy percentages (Table 2) are obtained. Finally, the EigenCAM [18] technique with GitHub implementation[1] is used to understand YOLOv8 model results, highlighting influential image regions. Activation maps for each model's last three layers are shown in Fig. 5.

4.4 MemSeg

The MemSeg model incorporates a memory bank mechanism that stores feature representations of normal images during the training phase. For this, a pretrained deep neural network, for this case study ResNet18 is used as a feature extractor. This model, trained with parameters similar to those used in classification tasks, is fine-tuned to capture key patterns that represent commonalities in non-defective surfaces. After training, MemSeg evaluates the model by calculating anomaly scores based on the differences between the current test image and the stored feature representations in the memory bank. Anomaly detection metrics, such as AUROC, Average Accuracy percentages and confusion matrix, are then calculated to assess model performance. These matrices help identify how well the model differentiates between defective and non-defective surfaces. To further understand how MemSeg highlights defects, heatmap are used to visualizing the defects area. These maps indicate which regions of the image the model focuses on when identifying defects. Such visual explanations help refine and validate the system's ability to detect anomalies effectively on various surfaces.

4.5 OPC-UA vision server implementation

The OPC-UA companion specification for machine vision aims to simplify the integration of machine vision systems into production control and IT systems. It enhances existing interfaces and establishes new horizontal and vertical integration capabilities, enabling authorized process participants to communicate relevant data up to the IT enterprise level. The "SingleExecution" and "ContinuousExecution" states are the part where the previously developed algorithms could be executed, specifically for the case study it would be executed in the "ContinuousExecution" state because the conveyor belt is running and the system would be continually waiting for an image of a tinplate lid. The other states are part of the OPC-UA specification and are part of the concept of a vision system that models the

[1] https://github.com/rigvedrs/YOLO-V8-CAM

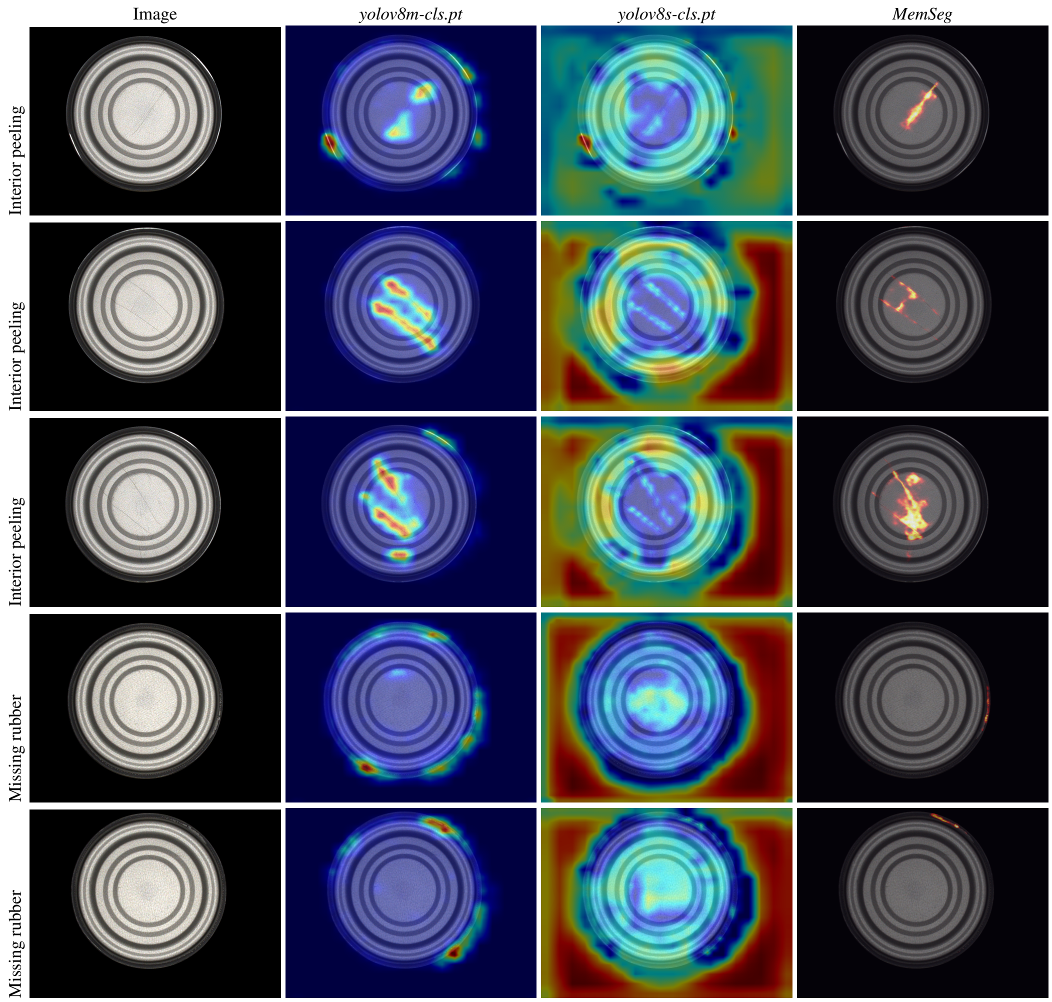
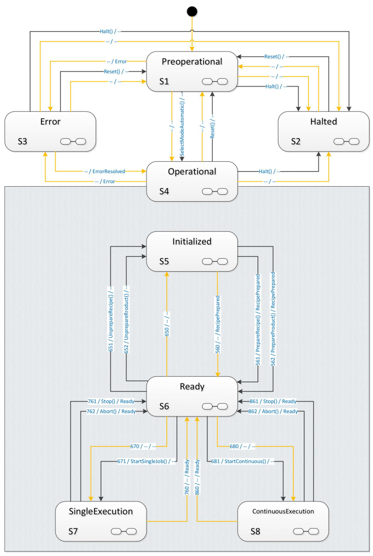


Figure 5: Qualitative evaluation of YOLO v8 and MemSeg prediction results on defective images of the testing set.

standard. The representation of the Vision Finite State Machine (VFSM) (Fig. 6 *(left)*) and the code of the implemented state machine (Fig. 6 *(right)*), which integrates the execution of both YOLO v8 and MemSeg techniques.

4.6 Experimental Results

For the evaluation of this study, Average Accuracy, Execution Time (e.g., CPU and GPU), RTT, E2E, and metrics are calculated to measure framework performance. Table 2 shows results from each technique calculated on the testing set. The technique that presents the best Average Accuracy is *"yolov8m-cls.pt"*, with a value of 1.00. In the case of Execution Time using CPU, MemSeg model presents the best performance with a time of 71.25ms, while using GPU the best result is presented by *"yolov8s-cls.pt"* with a time of 25.55ms. On the other hand, Fig. 4 shows the confusion matrix of each of the implemented techniques where it can be seen that *"yolov8m-cls.pt"* shows the best performance, allowing precisely differentiation between the *"good"* and *"defective"* classes. Also, in the case of MemSeg technique, the AUROC metric is used on the testing dataset, obtaining the following values AUROC-image: 0.992% and AUROC-pixel: 0.805%. Finally, Table 3 shows results from the performance evaluation in E2E and RTT metrics using different payloads, these metrics calculate the connection latency of the OPC-UA server and OPC-UA client created for this task.




```python
def process(self, image, threshold):
 if image:
    if self.sCurId.get_value() == self.sContExe.nodeid
        :
      if self.technique=="memSeg":
        # predict on 1 images
        results = model(image)
        #Anomaly score based
        anomaly_score = results[0][1].mean().item()

        if(anomaly_score > threshold):
            result = "Defective"
            activateActuator()
        else:
            result = "Good"
        print(f"Result: {result}")
      if self.technique=="yolo":
        # predict on 1 images
        results = model(image)
        #Index: defective (0), good (1)
        pred = results[0].probs.cpu().detach().numpy()

        if(pred[0] > pred[1]):
            result = "Defective"
            activateActuator()
        else:
            result = "Good"
        print(f"Result: {result}")
```

Figure 6: *(left)* States and transitions of the VisionAutomaticModeStateMachineType. Part OPC 40100-1 of the OPC-UA specification [9]. *(right)* Code executed in VFSM within OPC-UA server integrating and running YOLO v8 and MemSeg models.

Technique	Params. (M)	Execution Time CPU(ms)	GPU(ms)	Average Accuracy
MemSeg	11.70	**71.25**	39.90	0.976
yolov8m-cls.pt	42.70	218.90	67.50	**1.000**
yolov8s-cls.pt	13.50	99.65	**25.55**	0.991

Table 2: Performance evaluation Average Execution Time and Average Accuracy per each approach

Image Size Width	Height	Payload (kB)	E2E (ms)	RTT (ms)
1440	1080	1555300	121	238
1080	810	1166500	113	217
720	540	777700	98	186
360	270	388900	89	172

Table 3: Performance evaluation in End-to-End delay and Round Trip Time metrics in milliseconds.

5 Conclusion

This paper proposes a methodology to improve anomaly detection in industrial processes and compare supervised and semi-supervised deep learning techniques, with the integration of OPC-UA. YOLOv8 deep learning models (*"yolov8m-cls.pt"* and *"yolov8s-cls.pt"*) achieve average accuracies of 1.00 and 0.99, respectively. In contrast, MemSeg technique identifies anomalies with an average accuracy of 0.98. Furthermore, in relation to the Execution Time, *"yolov8s-cls.pt"* obtained a time of 25.55ms but an average accuracy lower than *"yolov8m-cls.pt"*. Also, all model implementations are within the system restrictions of staying within 75ms. On the other hand, E2E and RTT show that the system has a low latency connection with the OPC-UA server and can be used in near real-time systems. Finally, an OPC-UA server and a VFSM are implemented, and both models are executed within the "ContinuousExecution" state (see Fig. 6), thus allowing communication at a closer device level, which allows execution with low latency in near-real-time. On summary, the integration of OPC-UA and YOLOv8 in industrial environments offers notable advantages, including secure, standardized communication and device interoperability, alongside near-real-time defect detection. Although MemSeg's performance lagged slightly (by less than 2%) behind YOLOv8, its ability to detect future, unlabeled defects provides a valuable long-term benefit. OPC-UA ensures seamless, protected data exchange, while YOLOv8 delivers fast, precise object detection, automating visual inspections and minimizing human error.

References

[1] Rosmaini Ahmad and Shahrul Kamaruddin. An overview of time-based and condition-based maintenance in industrial application. *Computers & industrial engineering*, 63(1):135–149, 2012.

[2] Seunghyo Beak, Yo-Han Han, Yeeun Moon, Jieun Lee, and Jongpil Jeong. YOLOv7-Based Anomaly Detection Using Intensity and NG Types in Labeling in Cosmetic Manufacturing Processes. *Processes*, 11(8):2266, 2023.

[3] Mikel Canizo, Isaac Triguero, Angel Conde, and Enrique Onieva. Multi-head CNN–RNN for multi-time series anomaly detection: An industrial case study. *Neurocomputing*, 363:246–260, 2019.

[4] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys*, 41(3):1–58, 2009.

[5] Nwodo Benita Chikodili, Mohammed D Abdulmalik, Opeyemi A Abisoye, and Sulaimon A Bashir. Outlier detection in multivariate time series data using a fusion of K-medoid, standardized euclidean distance and Z-score. In *Int. Conf. on Information and Communication Technology and Applications*, pages 259–271. Springer, 2020.

[6] Andreas Eckhardt and Sebastian Müller. Analysis of the round trip time of opc ua and tsn based peer-to-peer communication. In *Int. Conf. on Emerging Technologies and Factory Automation*, pages 161–167. IEEE, 2019.

[7] Tolga Ergen and Suleyman Serdar Kozat. Unsupervised anomaly detection with LSTM neural networks. *Transactions on Neural Networks and Learning Systems*, 31(8):3127–3141, 2019.

[8] Erik M Ferragut, Jason Laska, and Robert A Bridges. A new, principled approach to anomaly detection. In *Int. Conf. on Machine Learning and Applications*, volume 2, pages 210–215. IEEE, 2012.

[9] OPC Foundation. OPC 40100-1 Machine Vision - Control, Configuration management, recipe management, result management. Technical report, OPC Foundation, 2024.

[10] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. In *Science*, volume 313, pages 504–507. American Association for the Advancement of Science, 2006.

[11] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023.

[12] Kangjik Kim, Hyunbin Kim, Junchul Chun, Mingoo Kang, Min Hong, and Byungseok Min. Real-time anomaly detection in packaged food x-ray images using supervised learning. *Comput. Mater. Contin*, 67(2):2547–2568, 2021.

[13] Jaromír Klarák, Robert Andok, Peter Malík, Ivan Kuric, Mário Ritomskỳ, Ivana Klačková, and Hung-Yin Tsai. From anomaly detection to defect classification. *Sensors*, 24(2):429, 2024.

[14] VF Leavers. Which hough transform? *CVGIP: Image understanding*, 58(2):250–264, 1993.

[15] Jayoung Lee and Youngmin Kim. Application of OPC-UA based on IoT for monitoring and controlling manufacturing processes. *Procedia Computer Science*, 155:154–161, 2019.

[16] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Int. Conf. on Computer Vision and Pattern Recognition*, pages 9664–9674, 2021.

[17] Jiaqi Liu, Guoyang Xie, Jinbao Wang, Shangnian Li, Chengjie Wang, Feng Zheng, and Yaochu Jin. Deep industrial image anomaly detection: A survey. *Machine Intelligence Research*, 21(1):104–135, 2024.

[18] Mohammed Bany Muhammad and Mohammed Yeasin. Eigen-cam: Class activation map using principal components. In *Int. Joint Conference on Neural Networks*, pages 1–7. IEEE, 2020.

[19] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton van den Hengel. Deep learning for anomaly detection: A review. *ACM Computing Surveys*, 54(2):1–38, 2021.

[20] Manuel Peña, Félix Biscarri, Juan Ignacio Guerrero, Iñigo Monedero, and Carlos León. Rule-based system to detect energy efficiency anomalies in smart buildings, a data mining approach. *Expert Systems with Applications*, 56:242–255, 2016.

[21] Dinh-Lam Pham, Tai-Woo Chang, et al. A YOLO-based real-time packaging defect detection system. *Procedia Computer Science*, 217:886–894, 2023.

[22] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[23] Henry Velesaca., Doménica Carrasco., Dario Carpio., Juan A. Holgado-Terriza., Jose Gutierrez-Guerrero., Tonny Toscano., and Angel Sappa. Anomaly detection in industrial production products using opc-ua and deep learning. In *Int. Conf. on Data Science, Technology and Applications*, pages 505–512. INSTICC, SciTePress, 2024.

[24] Henry O Velesaca, Juan A Holgado-Terriza, and Jose M Gutierrez Guerrero. Optimizing Smart Factory Operations: A Methodological Approach to Industrial System Implementation based on OPC-UA. In *E3S Web of Conferences*, volume 532, pages 1–15. EDP Sciences, 2024.

[25] Birgit Vogel-Heuser, Alexander Fay, Ina Schaefer, Matthias Tichy, Ansgar Trächtler, and Christoph Legat. Architectural design of multi-agent systems in the context of OPC UA and automationML. *Proceedings of the IEEE*, 104(5):947–967, 2016.

[26] Weng-Keen Wong, Andrew Moore, Gregory Cooper, and Michael Wagner. Rule-based anomaly pattern detection for detecting disease outbreaks. In *AAAI/IAAI*, pages 217–223, 2002.

[27] Minghui Yang, Peng Wu, and Hui Feng. Memseg: A semi-supervised method for image surface defect detection using differences and commonalities. *Engineering Applications of Artificial Intelligence*, 119:105835, 2023.

[28] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In *Int. Conf. on Computer Vision and Pattern Recognition*, pages 8330–8339, 2021.