Anomaly Detection in Industrial Production Products Using OPC-UA and Deep Learning

Henry O. Velesaca[1,2], Doménica Carrasco[1], Dario Carpio[1], Juan A. Holgado-Terriza[2], Jose M. Gutierrez-Guerrero[3], Tonny Toscano[1] and Angel D. Sappa[1,4]

[1]*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*

[2]*Software Engineering Department, University of Granada, 18014, Granada, Spain*

[3]*Abbott Laboratories, 18004, Granada, Spain*

[4]*Computer Vision Center, 08193-Bellaterra, Barcelona, Spain*

Keywords: Anomaly Detection, Industry 4.0, Deep Learning, OPC-UA, YOLO v8.

Abstract: In the realm of industrial manufacturing, detecting defects in products is critical for maintaining quality. Traditional methods relying on human inspection are often error-prone and time-consuming. However, advancements in automation and computer vision have led to smarter industrial control systems. This paper explores a novel approach to identifying defects in industrial processes by integrating OPC-UA and YOLO v8. OPC-UA provides a secure communication standard, enabling seamless data exchange between devices, while YOLO v8 provides accurate object detection. By combining these technologies, manufacturers can monitor production lines in near real-time, analyze defects promptly, and take corrective actions. As a result, product quality and operational efficiency are improved. A case study involving tinplate lid defect detection demonstrates the effectiveness of the proposed approach. The system architecture, including PLC integration, image acquisition, and YOLO v8 implementation, is detailed, followed by the performance evaluation of the OPC-UA server and YOLO v8 model integration. Results indicate efficient communication with low Round Trip Times and End-to-End delay, highlighting the potential of this approach for defect detection. The code is available at GitHub: https://github.com/hvelesaca/OPC-UA-YOLOv8-Lid-Anomaly-Detection, facilitating further research.

1 INTRODUCTION

In the highly competitive landscape of industrial manufacturing, the timely detection and correction of defects in products are critical to maintaining high standards of quality and operational efficiency (e.g., (Huang et al., 2017), (Monteiro et al., 2019)). Traditionally, this process has relied heavily on human inspection, which can be error-prone and limited in terms of speed and accuracy (Montgomery, 2019). However, with advancements in automation technology and computer vision, it has become possible to implement smarter and more effective industrial control systems ((Verkhivker et al., 2020), (Dey and Agrawal, 2016)).

In recent years, advances in industrial communication protocols have transformed the way quality assessments are performed in smart factories. These protocols enable smooth data exchange and communication among various elements of the manufacturing process, such as sensors, machinery, and quality control systems (Zheng et al., 2018). By harnessing these protocols, manufacturers can monitor production processes in real-time, analyze them for potential defects or anomalies, and swiftly implement corrective measures to uphold product quality and regulatory standards (Li et al., 2018).

This article focuses on exploring an innovative methodology for identifying defects in industrial manufacturing, leveraging two key technologies: OPC-UA (Foundation, 2023) (Open Platform Communications Unified Architecture) and YOLO v8 (Jocher et al., 2023) (You Only Look Once version 8). OPC-UA offers a robust and secure communication standard that facilitates interoperability between different devices and systems in the industrial environment. On the other hand, YOLO v8 is a cutting-edge object detection model in the field of computer

Velesaca, H., Carrasco, D., Carpio, D., Holgado-Terriza, J., Gutierrez-Guerrero, J., Toscano, T. and Sappa, A.
Anomaly Detection in Industrial Production Products Using OPC-UA and Deep Learning.
DOI: 10.5220/0012812600003756
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 13th International Conference on Data Science, Technology and Applications (DATA 2024)*, pages 505-512
ISBN: 978-989-758-707-8; ISSN: 2184-285X
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

vision, known for its speed and accuracy in identifying objects in images and videos.

In this context, we will explore how the integration of OPC-UA and YOLO v8 into industrial control systems can significantly improve the ability to detect and classify defects in real-time, enabling a faster and more efficient response to any anomalies in the manufacturing process. In addition, this work will examine case studies and practical applications that illustrate the benefits and potential limitations of this methodology, as well as future opportunities for its development and widespread adoption in the industry.

To address this work in detail, the manuscript is organized as follows. Section 2 introduces some related works on the use of OPC-UA, image identification, and the use of deep learning techniques within industrial processes. Section 3 presents the proposed approach to carry out the integration of OPC-UA and deep learning techniques. Then, section 4 shows the experimental results taking as reference a case study for the detection of anomalies in tinplate lids. Finally, conclusions are presented in Section 5.

2 BACKGROUND

As described previously, this paper presents an approach showing the integration of industrial control systems using OPC-UA and deep learning techniques as key elements. This section summarizes some of the most relevant techniques related to the topic of this work.

2.1 OPC-UA Overview

OPC-UA (Open Platform Communications Unified Architecture) is a communication standard widely used in industrial environments to facilitate interoperability between different devices, systems, and software platforms. Designed to address connectivity challenges in modern industrial environments, OPC-UA provides a robust and secure platform for near real-time data exchange and heterogeneous system integration.

However, OPC-UA is not only a communication protocol, but also allows other tasks to be carried out, such as, for example, providing a specification for Alarms (OPC-UA Part 9: Alarms and Conditions) which is a specific part of the OPC-UA standard that focuses on alarm and condition management in industrial environments. This standard specifies the guidelines for modeling, triggering, managing, and monitoring alarms and conditions in automation and control systems. It also includes a part (OPC-UA Part 11: Historical Access) which focuses on accessing and retrieving historical data stored on OPC-UA servers. This enables client applications to use historical information for analysis, reporting, trending, and other applications in industrial environments.

The study presented by (Georgi Martinov, 2017) demonstrates the use of the OPC-UA protocol for monitoring equipment with different kinematics. It highlights data acquisition from a CNC system, including motor positions, linear encoder readings, and electroautomatic device signals. The implementation does not require a separate adapter for the OPC server. Additionally, OPC-UA enables efficient client-side data retrieval through the "Publisher-Subscriber" model, allowing modifications to the kinematic schema. However, challenges may arise from network connectivity issues and the availability of OPC-UA servers in unreliable network environments.

Furthermore, (Nedeljkovic and Jakovljevic, 2020) addressed an issue in a manufacturing cycle using a "pick and place" system controlled by an Omron CP1L-EM40DT-D PLC. The problem emerges from components frequently entering the system incorrectly positioned. To resolve this, communication between the vision sensor and the handling system was established using the OPC-UA standard. This allowed the transmission of information about the parts' orientation and signaled the sensor when the manipulator was ready to receive data from the camera. Implementing OPC-UA provided the advantage of easily exchanging the smart vision sensor without reprogramming or modifying the wiring. However, transitioning from a previous OPC DA server introduced complexities, leading to the use of OPC Expert to adapt functionalities to the OPC-UA standard.

On the other hand, the paper presented by (Schäfer et al., 2022) outlines an architecture for integrating reinforcement learning (RL) into industrial environments to optimize operations and enhance decision-making, with a key emphasis on the use of OPC-UA for seamless and secure data exchange. The framework includes real-time data collection from sensors and IoT devices via OPC-UA, preprocessing of this data, and training of RL models that learn optimal strategies through continuous interaction with the environment. Once trained, these models are deployed within the industrial control system to make real-time decisions. The system also incorporates monitoring and feedback mechanisms to continuously improve and retrain the models, ensuring adaptability and scalability for various industrial applications, ultimately leading to more intelligent and autonomous processes.

Finally, based on object-oriented principles, Gutierrez et al. (Gutierrez-Guerrero and Holgado-Terriza, 2017) propose a metamodel iMMAS for conceptualizing industrial automation systems. This metamodel includes a concrete syntax and specific semantics that simplify the development and deployment of manufacturing control systems. In these systems, the models can be transformed into PLC programs and OPC-UA data models.

2.2 Object Recognition

Object recognition in industrial environments has emerged as a powerful tool to improve efficiency, quality, and safety in a wide range of industrial applications. With the advancement of computer vision technology and deep learning, companies are increasingly leveraging the ability of machines to visually analyze and understand their work environment. On the other hand, object recognition in industrial environments involves the use of machine learning algorithms and models to recognize specific objects, patterns, or features in images captured by cameras or vision devices. These systems can detect product defects, monitor equipment performance, perform quality inspections, and more.

For example, the paper by (Dominguez et al., 2006) addresses the complexities of object recognition and inspection in challenging industrial environments. They use advanced computer vision techniques, particularly the YOLO v8 convolutional neural network, to enhance object detection and classification in real-time. YOLO v8's capabilities enable fast and accurate detection even under adverse conditions, improving industrial inspection processes. However, challenges such as the need for diverse training datasets, adaptation to varying environmental conditions, and computational resource requirements may hinder successful implementation. Additionally, expertise in parameter fine-tuning is crucial for achieving optimal results in practical industrial settings.

On the other hand, the paper by (Rocha et al., 2014) introduces a cascade system designed for object recognition and pose estimation in industrial settings. This system comprises several stages, starting with object detection using deep learning techniques such as YOLO, followed by pose estimation algorithms to determine object orientation. By integrating these methods, the system demonstrates robust performance in identifying and accurately estimating object poses, thus enhancing automation and quality control processes in industries. However, challenges may arise due to occlusions, lighting variations, and complex backgrounds, potentially impacting the accuracy and reliability of object recognition and pose estimation. Moreover, the computational complexity of the cascade system may necessitate efficient hardware resources for real-time deployment in industrial environments.

2.3 Deep Learning Within Industrial Systems

The advancement of artificial intelligence, especially in the field of deep learning, has opened new frontiers in the optimization and automation of industrial systems. In this context, the use of deep learning techniques within industrial systems is gaining more and more relevance due to its ability to process large amounts of data, identify complex patterns, and improve operational efficiency. This article explores the impact and applications of deep neural networks in industrial environments, highlighting their potential to improve quality, predict failures, optimize processes, and reduce production costs.

The authors propose various deep learning techniques for fault diagnosis in industrial environments, emphasizing their ability to identify unique patterns and ensure accurate, timely detection. Challenges include interpreting model results and integrating them with existing monitoring systems. For example, the work presented by (Surendran et al., 2022) highlight the effectiveness of deep learning in analyzing large data sets for early fault detection but notes the high demand for training data and computational complexity. On the other hand, the article presented by (Iqbal et al., 2019) address fault detection and isolation, underscoring the capability of deep learning models to detect complex patterns but also recognizing the need for extensive data and significant computational resources. The work presented by (Sánchez Santalices et al., 2023) demonstrate neural network implementation for tray anomaly detection, capable of identifying irregular patterns for early problem detection. However, the need for representative training data and the potential for false positives or negatives necessitate careful model optimization and tuning.

3 PROPOSED APPROACH

This section details the different stages of the proposed approach. Figure 1 shows the system architecture of the proposed approach. The system architecture is based on a hierarchical industrial system with three layers: a) the control layer where the main PLC subsystem is in charge of controlling the plant; b) the

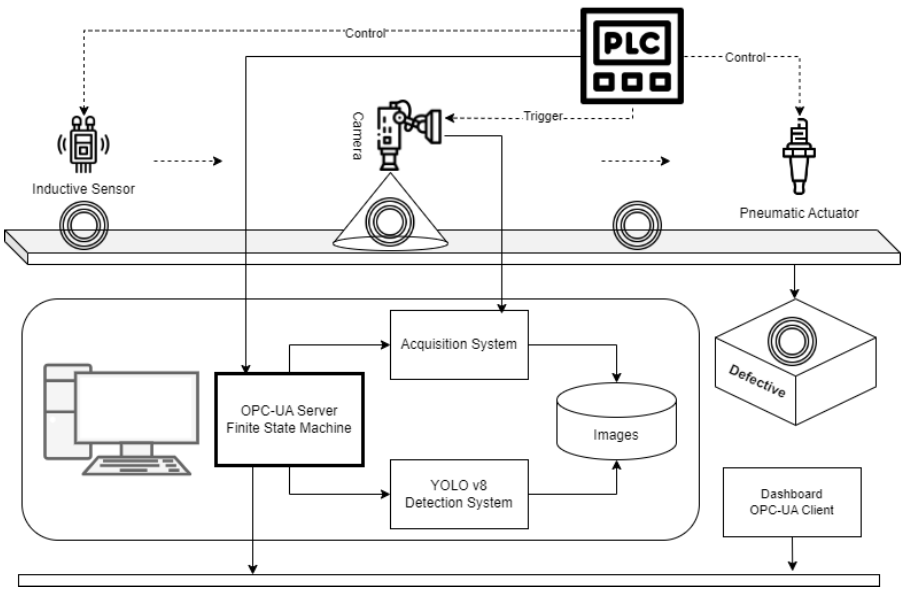

Figure 1: General outline of the proposed system architecture in this paper.

supervision layer where the main OPC-UA subsystem is responsible for retrieving the plant status, the image acquisition from the camera and the application of anomaly detector based on a deep neural network; and, finally, c) the visualization layer where operators can examine the evolution of the plant using a dashboard.

3.1 System Description

The evaluation of the proposal is based on a case study of a tinplate lid failure detection system. For the architecture proposed in this work, a conveyor belt, a PLC, and an inductive sensor are used to detect the passage of the lids. In addition, a vision system consisting of an industrial camera with a lighting system is used. In addition, the camera is connected to a workstation which also allows the execution of the OPC-UA Server and the deep neural network.

3.2 OPC-UA Server

In our proposal, the integration of an OPC-UA server in the defect detection system in industrial processes plays a crucial role by providing a unified and secure interface for near real-time data collection. By leveraging the interoperability and security capabilities of OPC-UA, it can ensure the integrity and reliability of data collected from different devices and systems on the plant floor. This integration allows for continuous and efficient process monitoring, providing a solid basis for defect detection.

3.3 YOLO v8

The neural network selected to be used in the image identification component of the detection subsystem is YOLO v8 (Jocher et al., 2023). It is selected for its efficiency and speed in detecting objects in images, which makes it suitable for real-time applications in industrial environments where fast responses are required. Additionally, its ability to identify multiple objects in a single pass makes it ideal for identifying defects in industrial products with multiple irregularities. YOLO v8 also offers a deep and flexible architecture that allows easy adjustment and optimization 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. Finally, it is open source and its extensive developer community makes it easy to implement and long-term maintain YOLO v8-based defect detection systems.

3.4 Integration of OPC-UA Server and YOLO v8

Combining the OPC-UA server with advanced defect detection algorithms, such as YOLO v8, the system's ability to identify and classify defects in near-real-time can significantly be improved. By using accurate and timely data provided by the OPC-UA server, detection algorithms can be effectively trained and tuned to recognize a wide range of defects with high accuracy. The integration of these technologies offers a comprehensive and effective approach to improving quality and efficiency in industrial processes. This approach simultaneously reduces downtime and production costs.

3.5 Metric Evaluation

For the evaluation of the proposed work, Round Trip Time (RTT) and End-to-End delay metrics will be calculated to establish the performance of the server. RTT is a measure of the amount of time it takes for a data packet to travel from the source point to the destination and then return to the source. E2E refers to the time it takes for information to travel from its source to its destination in a networked system. These measurements are important in evaluating network performance as it directly affect the communication speed and responsiveness of online applications and services. Both metrics are commonly used in computer networks to evaluate latency or network response time (Eckhardt and Müller, 2019).

4 CASE STUDY

This section presents the case study experimental results obtained with the proposed framework. For performance evaluation of the proposed approach, RTT and E2E measurements are used.

4.1 System Implementation

In consideration of the system architecture depicted in Figure 1, a system implementation for the detection of manufacturing defects in tin lids within a factory is presented. As the main element, a Siemens S7 1200 PLC is used in the control layer. Additionally, a vision system consisting of an Industrial Visible Spectrum Camera is used along with a lighting system. The system also has an Inductive Sensor used to detect the lids on the conveyor belt. For the supervision layer, a workstation is used with an Intel Core I9 3.3GHz CPU and NVIDIA Titan XP GPU for training and testing YOLO v8 for the identification of images. It is responsible for acquiring the images and also for running the OPC-UA Server and YOLO v8. Figure 2 shows the main components used in the system architecture.

4.2 Image Acquisition

As a first step, the acquisition of images of tin lids in good and defective condition is established. Among the most common defects are scratches, dents, and lack of rubber on the inside edge. Table 1 shows the distribution of data used for the training, validation, and testing stages used by YOLO v8 in a later stage.

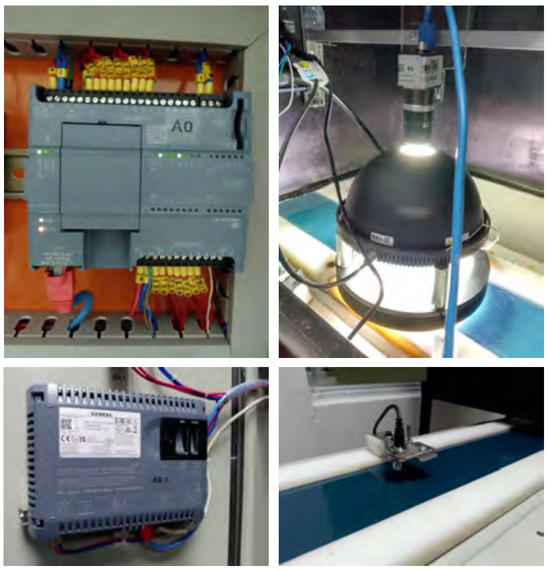

Figure 2: *(top-left)* Siemens S7 1200 PLC. *(bottom-left)* HMI TP700 comfort. *(top-right)* Vision system and lighting system. *(bottom-right)* Inductive Sensor.

Table 1: Distribution of data acquisition.

Task	**Good**	**Defective**
Training	457	588
Validation	85	112
Testing	29	36
Total	571	736

4.3 OPC-UA Server Implementation

The next step is the implementation of the OPC-UA server for which the design of the finite state machine (FSM) specified in Part 16 of OPC-UA has been defined (see Figure 3). Also to create the OPC-UA server, the behavior model has been defined based on the FSM and using the methodology proposed by (Velesaca et al., 2024). Starting from the model created in Figure 4 and with the help of the Free OPC-UA Modeler (FreeOpcUa, 2 28), the OPC-UA server is created in XML format.

After obtaining the model in XML format, the next step is deploying the server. The language used is Python, so to carry out the deployment of the OPC-UA server, two files are created: *Server.py* and *Utils.py*. The file *Server.py* contains the general structure of the server and defines a class that represents OPC-UA programs, specifically implements the methods that contain the behavior of the FSM,

Table 2: Finite State Machine transitions of the system.

No	Name Transition	Cause	Origin State	Destination State	Effect
1	HaltedToReady	Reset Method	Halted	Ready	Report Transition 1 Event/Result
2	ReadyToRunning	Start Method	Ready	Running	Report Transition 2 Event/Result
3	RunningToHalted	Halt Method or Internal (Error)	Running	Halted	Report Transition 3 Event/Result
4	RunningToReady	Internal	Running	Ready	Report Transition 4 Event/Result
5	RunningToSuspended	Suspend Method	Running	Suspended	Report Transition 5 Event/Result
6	SuspendedToRunning	Resume Method	Suspended	Running	Report Transition 6 Event/Result
7	SuspendedToHalted	Halt Method	Suspended	Halted	Report Transition 7 Event/Result
8	SuspendedToReady	Internal	Suspended	Ready	Report Transition 8 Event/Result
9	ReadyToHalted	Halt Method	Ready	Halted	Report Transition 9 Event/Result
10	DetectingToDetecting	Internal	Detecting	Detecting	Report Transition 10 Event/Result
11	DetectingToAcquiring	Internal	Detecting	Acquiring	Report Transition 11 Event/Result
12	AcquiringToAcquiring	Internal	Acquiring	Acquiring	Report Transition 12 Event/Result
13	AcquiringToClassifying	Internal	Acquiring	Classifying	Report Transition 13 Event/Result
14	ClassifyingToClassifying	Internal	Classifying	Classifying	Report Transition 14 Event/Result
15	ClassifyingToDetecting	Internal	Classifying	Detecting	Report Transition 15 Event/Result

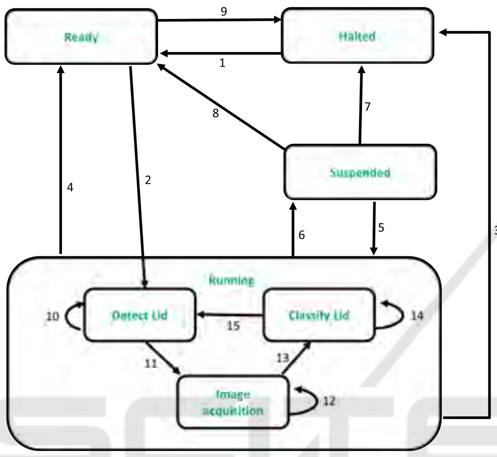

Figure 3: Finite State Machine for defect detection system in tinplate lids.

and uses the semantics offered by OPC-UA to validate transitions between states which are defined in the Table 2. *Utils.py* file contains additional functions for server creation. The last step is the execution of the server.

4.4 YOLO v8 Implementation

As a first step before starting the training phase, it is decided to increase the number of examples for training, for which the Albumentations library is used. This library allows for data augmentation and is commonly used in computer vision tasks, such as training object detection models such as YOLO v8. Albumentations offers a wide range of image transformations, such as cropping, rotations, brightness, and contrast changes, among others. These transformations can be applied in a random and controlled manner during preprocessing of the training data, generating additional instances of the original images with realistic variations.

Figure 5 shows the execution code using console mode with YOLO v8 for the classification task.

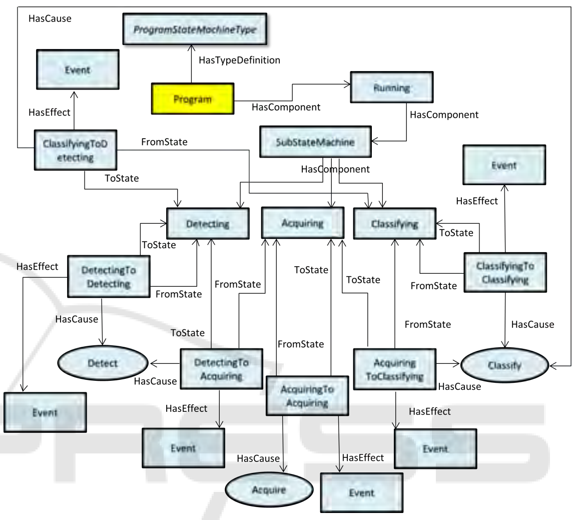

Figure 4: Behavior model in OPC-UA notation (Velesaca et al., 2024).

```
$ yolo classify train data="dataset/" \
model=yolov8m-cls.pt epochs=500 imgsz=640
```

Figure 5: Code executed to training data with YOLO v8.

The model *"yolov8m-cls.pt"* and 500 training epochs are used as execution parameters. After completing the training/validation tasks, the normalized confusion matrix shown in Figure 6 is obtained as a result. Furthermore, an accuracy of 93% is obtained.

For the testing task, the code shown in Figure 8 is used. Additionally, the qualitative and quantitative

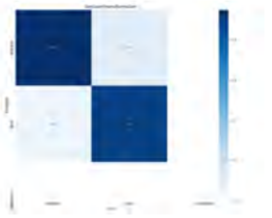

Figure 6: Normalized confusion matrix for the network trained using YOLO v8.

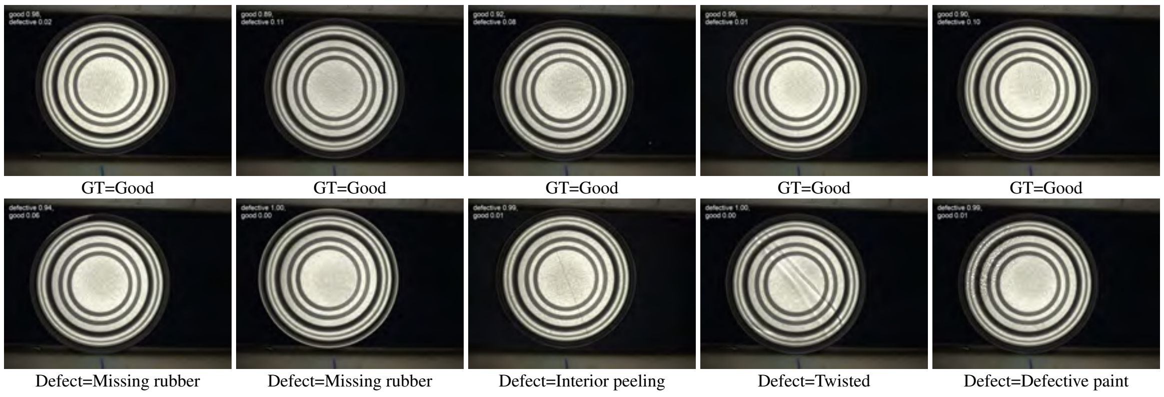

Figure 7: YOLO v8 prediction results. These example images are part of the testing set.

```
$ yolo classify predict source="test/" \
model="train/weights/best.pt"
```

Figure 8: Code executed to testing data with YOLO v8.

```
def run(self):
  while(True):
    if self.sCurId.get_value() == self.sRun.nodeid:
      if self.sCurId.get_value() == self.sDet.nodeid:
        self.Detecting()
      elif self.sCurId.get_value() == self.sAcq.nodeid:
        self.Acquiring()

        # predict on 1 images
        results = model(image_to_predict)
        #Index: defective (0), good (1)
        pred = results[0].probs.cpu().detach().numpy()

        if(pred[0] > pred[1]):
            defective = True
            activate_actuator()
            print("Defective")
        else:
            defective = False
            print("Good")

      elif self.sCurId.get_value() == self.sClas.nodeid:
            self.Classifying()

    time.sleep(0.00001)
```

Figure 9: Code executed in FSM in OPC-UA server with YOLO v8.

evaluation on a testing subset is shown in Figure 7.

4.5 Integration of OPC-UA Server and YOLO v8

To establish communication between the PLC and the OPC-UA server, the Snap7[1] library is used. One of the variables configured in the PLC is a trigger signal which captures the reading of the inductive sensor and based on the distance from the camera determines if it is necessary to perform a capture. Once the image is read, it is sent to the YOLO v8 prediction system so that it can determine if the tinplate lid is defective or good. If it is defective, the pneumatic actuator will be activated to separate the damaged item.

The execution of the model trained in YOLO v8 is carried out within the OPC-UA server scope, in addition, state changes are executed using the finite state machine model. Figure 9 shows the code in the OPC-UA server where the execution of the model trained in YOLO v8 is carried out.

[1]https://pypi.org/project/python-snap7/

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

Image Size		Payload	E2E	RTT
Width	Height	(kB)	(ms)	(ms)
1440	1080	1555200	237	275
1080	810	1166275	209	231
720	540	777450	178	189
360	270	388690	159	168

4.6 Metric Evaluation

As a last stage to evaluate the performance of the OPC-UA server together with YOLO v8 in near real-time two metrics are used End-to-end delay (E2E) and Round-Trip Time (RTT). E2E refers to the time taken for a packet to be transmitted across a network from source to destination. It is a common term in IP network monitoring and differs from RTT in that only the path in one direction from source to destination is measured. Additionally, 100 images are considered for the calculation of this measure.

5 CONCLUSIONS

The article proposes a methodology to improve anomaly detection in industrial processes by integrating OPC-UA and YOLO v8. Traditional human inspection methods are noted to be error-prone and slow, underscoring the need for more automated and accurate solutions. The article describes in detail the architecture of the proposed system, which includes components such as a PLC, an industrial camera, an OPC-UA server, and the YOLO v8 model. The interaction between these components is highlighted to achieve efficient near-real-time defect detection. Furthermore, experimental results are presented, including performance metrics such as RTT and E2E to evaluate the system efficiency.

Finally, the combined use of OPC-UA and YOLOv8 in industrial environments offers significant benefits such as secure, standardized communication, and interoperability between devices, along with near-real-time monitoring. OPC-UA enables seamless and protected data exchange, while YOLOv8 provides fast and accurate object classification, automating visual inspection and reducing human errors. Additionally, OPC-UA's capability to access and analyze historical data facilitates predictive maintenance and process optimization, enhancing operational efficiency and product quality.

ACKNOWLEDGEMENTS

This research has been partially supported by the ESPOL project "Automatización del proceso de detección de fallas en piezas de hojalata usando visión por computador" (CIDIS-004-2023), and by University of Granada.

REFERENCES

Dey, S. and Agrawal, M. K. (2016). Tinplate as a sustainable packaging material: Recent innovation and developments to remain environment friendly and cost effective. *Int. J. Res. IT Manag. Eng*, 8:9–22.

Dominguez, E., Spinola, C., Luque, R. M., Palomo, E. J., and Munoz, J. (2006). Object recognition and inspection in difficult industrial environments. In *Int. Conf. on Industrial Technology*, pages 989–993. IEEE.

Eckhardt, A. and Müller, S. (2019). 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 (ETFA)*, pages 161–167. IEEE.

Foundation, O. (2023). OPC Unified Architecture. Accessed: December 2023.

FreeOpcUa (último acceso: 2024-02-28). FreeOpcUa Modeler. https://github.com/FreeOpcUa/opcua-modeler.

Georgi Martinov, Roman Pushkov, S. E. (2017). Opc ua-based smart manufacturing: System architecture, implementation, and execution. *IEEE.*

Gutierrez-Guerrero, J. M. and Holgado-Terriza, J. A. (2017). iMMAS an Industrial Meta-Model for Automation System Using OPC UA. *Elektronika Ir Elektrotechnika*, 23((3)):3–11.

Huang, H.-W., Wu, S.-J., Lu, J.-K., Shyu, Y.-T., and Wang, C.-Y. (2017). Current status and future trends of high-pressure processing in food industry. *Food control*, 72:1–8.

Iqbal, R., Maniak, T., Doctor, F., and Karyotis, C. (2019). Fault detection and isolation in industrial processes using deep learning approaches. *Transactions on Industrial Informatics*, 15(5):3077–3084.

Jocher, G., Chaurasia, A., and Qiu, J. (2023). Ultralytics yolov8.

Li, Q., Tang, Q., Chan, I., Wei, H., Pu, Y., Jiang, H., Li, J., and Zhou, J. (2018). Smart manufacturing standardization: Architectures, reference models and standards framework. *Computers in Industry*, 101:91–106.

Monteiro, C. A., Cannon, G., Lawrence, M., Costa Louzada, M. d., and Pereira Machado, P. (2019). Ultra-processed foods, diet quality, and health using the nova classification system. *Rome: FAO*, 48.

Montgomery, D. C. (2019). *Introduction to statistical quality control*. John wiley & sons.

Nedeljkovic, D. M. and Jakovljevic, Z. B. (2020). Integration of smart vision sensor into manipulator control system using opc-ua. *IEEE.*

Rocha, L. F., Ferreira, M., Santos, V., and Moreira, A. P. (2014). Object recognition and pose estimation for industrial applications: A cascade system. *Robotics and Computer-Integrated Manufacturing*, 30(6):605–621.

Sánchez Santalices, J., Moya de la Torre, E. J., and Poncela Méndez, A. V. (2023). Implementación de una red neuronal para la detección de anomalías en bandejas. In *Jornadas de Automática*, pages 873–878. Universidade da Coruña. Servizo de Publicacións.

Schäfer, G., Kozlica, R., Wegenkittl, S., and Huber, S. (2022). An architecture for deploying reinforcement learning in industrial environments. In *Int. Conf. on Computer Aided Systems Theory*, pages 569–576. Springer.

Surendran, R., Khalaf, O. I., and Tavera Romero, C. A. (2022). Deep learning based intelligent industrial fault diagnosis model. *Computers, Materials & Continua*, 70(3).

Velesaca, H. O., Holgado-Terriza, J. A., and Gutierrez-Gutierrez, J. M. (2024). Optimizing Smart Factory Operations: A Methodological Approach to Industrial System Implementation based on OPC-UA. In *Int. Conf. of Applied Industrial Engineering*, pages 1–15.

Verkhivker, Y., Altman, E. I., Dotsenko, N. V., and Miroshnishenko, E. (2020). Commodity approach to materials when manufacturing containers for foods. *Iuniper Online Journal Material Science*, 6(3):555687.

Zheng, P., Wang, H., Sang, Z., Zhong, R. Y., Liu, Y., Liu, C., Mubarok, K., Yu, S., and Xu, X. (2018). Smart manufacturing systems for industry 4.0: Conceptual framework, scenarios, and future perspectives. *Frontiers of Mechanical Engineering*, 13:137–150.