Volume 10, Issue 3, March – 2025 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar609

Artificial Intelligence and Image Processing Based Part Feeding Control in a Robot Cell

Enesalp ÖZ[1,2]; Muhammed Kürşad UÇAR[3,4]
[1]Electrical and Electronics Engineering, Institute of National Science, Sakarya University, Serdivan, Sakarya, Turkey
[2]Toyota Motor Manufacturing, Body - Weld Engineer, Arifiye/Sakarya, Turkey
https://orcid.org/0000-0002-3467-704X
[3]Electrical and Electronics Engineering, Faculty of Engneering, Sakarya University, Serdivan, Sakarya, Turkey
[4]MKU Technology, Sakarya University Technopolis Region, Serdivan, Sakarya, Türkiye
https://orcid.org/0000-0002-0636-8645

Publication Date: 2025/03/20

Abstract: In this study, an artificial intelligence-assisted image processing system was developed to prevent errors in part feeding processes within an industrial robot cell. Using the YOLOv7-tiny model, accurate detection of parts was ensured, enabling effective quality control. While PLC communication was established via the ModBus protocol, the system hardware included an NVIDIA JETSON AGX ORIN, a BASLER acA2500-60uc camera, and a Raspberry Pi WaveShare monitor. A total of 2400 data samples were used for model training, achieving an accuracy rate of 98.07%. The developed system minimized human errors by preventing incorrect part feeding issues and significantly improved efficiency in production processes. Notably, the system's superior accuracy and processing speed demonstrated its suitability for real-time applications. In conclusion, this study highlights the effective implementation of artificial intelligence and image processing techniques in industrial manufacturing processes.

Keywords: *Artificial Intelligence, Image Processing, YOLOv7-tiny, Industrial Automation, Part Inspection.*

How to Cite: Enesalp ÖZ; Muhammed Kürşad UÇAR (2025). Artificial Intelligence and Image Processing Based Part Feeding Control in a Robot Cell. *International Journal of Innovative Science and Research Technology*, 10(3), 455-465. https://doi.org/10.38124/ijisrt/25mar609

I. INTRODUCTION

The industrial sector has increasingly turned to industrial robots to optimize production processes and enhance efficiency during the Fourth and Fifth Industrial Revolutions. In this period, robots have undertaken various tasks in production environments, including part handling, process monitoring, and collaboration with operators. As a result, many manufacturing facilities have improved efficiency and ensured production continuity [1]. However, in factories and workshops where human labor still plays a crucial role, issues such as quality defects, missing parts, and insufficient production speed persist. In this context, integrating the advantages of automation with human flexibility and sensitivity in environments where industrial robots collaborate with humans is essential. This approach enhances interaction between industrial robots and humans, enabling more efficient and effective management of production processes. In the automotive industry, various issues arise in processes involving human workers. In welding factories, vehicle bodies are assembled and welded by robots. To form the body, multiple subcomponents are welded together. The part assembly process is divided into two main production lines. The first type consists of fully automated lines where humans are not involved. In these lines, parts and body structures are transferred using automated equipment, positioned by robots and fixtures, and welded through intercommunicating robotic stations. The second type consists of side processes where humans are actively involved in assembling fundamental vehicle components, transferring parts, and positioning them correctly. Various errors, such as part damage, missing or excessive part assembly, and incorrect part feeding, frequently occur in these side processes. Accurately detecting and identifying objects in production processes is critical for the efficiency and quality of industrial manufacturing facilities [2]. Correctly determining object characteristics such as color, shape, orientation, and texture enables various improvements in production processes. This detection and identification process ensures the selection of correct parts and contributes to the early detection of potential defects. Consequently, overall efficiency increases, and product quality improves in industrial production facilities. Additionally, accurate object detection helps reduce human errors, minimizing production defects and enhancing workplace safety. Therefore, object detection and identification play a fundamental role in improving manufacturing efficiency and quality [2]. The

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar609

positions and characteristics of objects are detected using various methods in both fixed and dynamic systems. These detection processes are typically performed using machine vision and vision sensors. However, such systems are vulnerable to environmental factors in the working environment. In particular, areas exposed to challenging conditions, such as welding factories with metal debris, dust, and smoke, may limit the effectiveness of machine vision and sensor-based detection. In such environments, external factors can negatively impact the sensors' detection and processing capabilities. Thus, more robust and durable sensors or alternative detection methods may be required to ensure reliable and stable results in production processes [1]. For example, a study on ceramic tile production examined how simple image processing techniques could be used to improve quality control. The study specifically focused on the automatic detection and classification of cracks, stains, and other defects on tile surfaces [3]. While the system successfully addressed quality control issues using an existing product, the project environment had controlled lighting conditions, preventing external influences. However, such a system would be easily affected in environments like welding factories, where metal debris and sudden light sources are present.

II. LITERATURE REVIEW

As a result of the literature review, both basic image processing and artificial intelligence-supported projects have been examined. Basic image processing methods have been used for the automatic classification of agricultural products, focusing on analyzing features such as the size, color, and shape of fruits and vegetables [4]. This enables automatic classification and quality control of products. In an example application, the characteristic features of an apple were extracted for quality control; however, edge detection was used to identify defects on the apple. Extracting object characteristics using such an edge detection algorithm is unreliable, as it is highly susceptible to external light sources. The literature review highlights several key issues that underline the insufficiency of basic image processing techniques. One issue is the problem of part recognition and classification. Basic image processing methods struggle to differentiate between subtle differences among parts [5]. Basic image processing methods are insufficient in identifying complex patterns and variations in parts [6]. For advanced defect detection and segmentation problems, basic image processing techniques may not be effective [7]. Additionally, they are inadequate for handling noisy data and distortions [8]. Another issue is image segmentation and defect detection. Basic image processing techniques are not sufficient for accurately detecting and segmenting complex defects [9]. They struggle to identify complex defects in real-world data [10]. They may not be effective in detecting and classifying intricate defects [11]. The third issue involves handling heavy noise and distortions. Basic image processing techniques are inadequate for dealing with noisy data and distortions [12]. They fail to effectively detect and classify complex defects in production lines [13]. The final issue is related to real-time processing requirements. Basic image processing techniques may fail to meet the real-time processing requirements of industrial applications [14]. A study was conducted to address issues in weld inspection. Various methods were explored, including a traditional manual image processing procedure for feature extraction, followed by defect classification using a Support Vector Machine and defect localization via template matching [15]. However, such conventional methods are easily affected by environmental factors. As an alternative, artificial neural networks have been employed. Despite the effectiveness of ANN-based systems in specific tasks, they require extensive expertise for design, integration, and optimization. Moreover, large-scale implementation in industrial manufacturing settings necessitates a well-structured monitoring and control mechanism. The need for extensive experience in ANN-based defect detection presented a major challenge [16]. AlexNet was introduced in 2012 as a solution for such inspection applications, achieving a Top-5 classification error rate of 16.4%, compared to the 28.2% of traditional methods—a significant 11.8% improvement. This model was trained using over 14 million images and categorized into 21,841 classes. Following these advancements, deep learning networks became the preferred approach for industrial inspection applications. Researchers have since developed object detection applications using various deep learning methods [16]. For example, Huifan applied the RCNN framework to detect welding defects, achieving an accuracy rate of 58.54%. Wenhui Hou utilized a deep convolutional method, attaining a classification accuracy of 97.2%. Additionally, another researcher employed the YOLOv3 object detection framework, achieving a 75% accuracy rate [16]. Instead of traditional feature extraction processes, multi-layered neural networks—commonly known as deep learning—were adopted. The target object is continuously fed into the deep network with labeled data, enabling the network to learn the object's characteristic structures. Matthew D. Zeiler was the first to analyze deep learning and found that each layer was designed similarly to traditional feature extraction methods. The first layer extracts basic color characteristics, the second layer identifies textures, and the third layer detects object shapes. This automated feature learning approach eliminates much of the design workload, allowing object recognition systems to be built without requiring extensive prior knowledge. After seven years of development, deep learning has become the most sustainable and high-performing approach for object recognition, gaining significant attention from both industry and academia. In recent years, researchers have introduced numerous creative deep learning architectures. One such architecture is the YOLO network, designed by Joseph Redmon and Ali Farhadi at the University of Washington [5]. Another notable model is Faster RCNN, developed by Kaiming He at Facebook Research. Many object detection algorithms have been developed for use in both industry and academia, but two have gained widespread popularity: Faster RCNN and the YOLO series. While Faster RCNN provides better object detection accuracy, YOLO outperforms it in real-time data processing applications. To address the problem discussed in this study, the YOLO deep learning model, known for its superior real-time processing performance, was used. Various YOLO versions with different features were analyzed to select the most suitable and up-to-date model. As a result of this research, YOLOv7-tiny was determined to be

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar609

the best choice in terms of both performance and reliability. The model is lightweight, fast, and delivers high accuracy. Based on these findings, the image processing system was built using the YOLOv7-tiny model. This study aims to develop an AI-supported image processing system to ensure quality control in part placement processes on a production line. The system will be used in a production station where parts are manually placed by workers, ensuring that parts are correctly positioned in real-time. The first step involves training an object detection model to determine whether parts are placed correctly. For this purpose, the open-source Darknet deep learning framework will be utilized to train the YOLO model. The YOLO model is chosen for its speed and efficiency in object detection, making it suitable for real-time processing requirements. Next, a Programmable Logic Controller will be used to communicate with the production station. The ModBus protocol will be employed for PLC communication, allowing the system to activate the camera upon receiving a trigger signal from the production station and perform object detection. The detected information will then be visualized on a user interface developed using PyQt5. This interface will enable workers to monitor the part placement process and receive alerts in case of incorrect placements. In conclusion, the developed AI-assisted image processing system will enhance quality control in part placement processes on the production line and prevent incorrect placements. This system can be effectively used to increase efficiency and accuracy in industrial automation applications.

III. MATERIALS AND METHODS

The workflow diagram of the study is illustrated in Fig-1. In the hardware setup, equipment such as a Jetson PC, camera, display, and Adam IO were installed for the AI-based system. Data collection involved gathering image data from the production site to create the dataset required for model training. During the data preprocessing stage, the collected images were processed to ensure they were suitable for training the AI model. Model training was conducted using the preprocessed data to develop the AI system. A user interface was designed to display images and results. The trained AI model was tested in the production environment and deployed into the system for practical implementation.

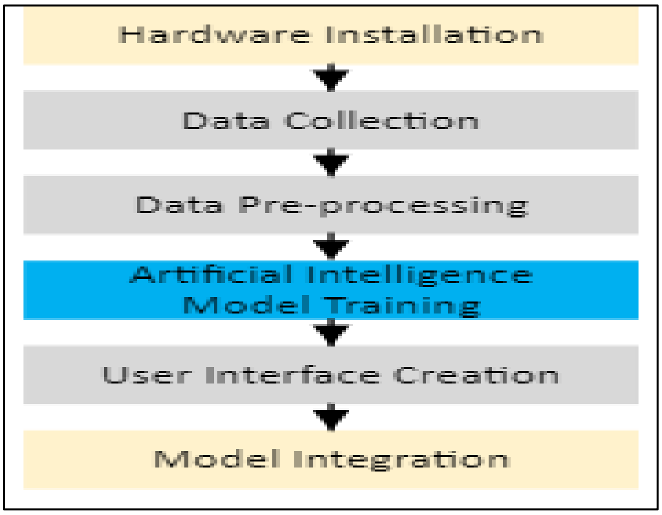

Fig 1 Workflow Diagram

➢ *FSM RR TACK 2 LH Robot Process*

Information about the production process mentioned in the introduction will be provided in this section. The system setup will be implemented in the Front Side Member Rear Tack 2 robot process, which is one of the sub-processes of the welding factory. The process image can be seen in Fig.1. In the FSM RR Tack 2 robot process, the side member components forming the vehicle's shell body are assembled. The FSM RR Tack 2 process consists of two separate operations, one for the right side and one for the left side. In this process, two different parts are joined together. The main part, shown in Fig.2, is common to both the right and left processes. However, the second part to be assembled differs depending on whether it belongs to the right or left side. The operator may mistakenly place the wrong part onto the main part. In the production condition, detecting a misfeed is only possible after the shell body has been assembled. If the incorrect part feeding is not detected during the process, an entire body may be scrapped. To prevent human-induced incorrect part placement, a system will be implemented to inspect and verify the assembly process. The joining operation will only be permitted after the system confirms the correct part placement.

➢ *Hardware Installation in the Field*

The system installation consists of three main sections. The first is the installation of a camera. In the era of automation and smart factories, cameras have become a necessity for implementing intelligent systems, and there are many camera brands and models with different features [17]. There are several critical factors to consider when installing cameras in industrial production environments. To ensure the camera captures clear and accurate images, the installation area must be stable and positioned in a way that prevents vibrations. While the camera should be mounted on a vibration-free, fixed surface, allowing it to rotate provides flexibility for easier image adjustment. After determining the installation location and structure, it is essential to protect the camera, especially in industrial production environments. In the welding factory process, robots perform spot welding, which generates metal splatter. One of the most sensitive and easily damaged components of a camera is its lens. To protect the lens from welding splatter, a transparent cover has been used.

Fig 2 Process Image

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar609

Fig 3 Main Part Image

➢ *Data Collection*

Different methods can be used for data collection. In this study, data collection was conducted on-site using a Python script. The program was executed to capture images of both OK and NG part placements. Fig.4 illustrates the amount of OK and NG data collected. A total of 2400 images were gathered, with 1200 OK and 1200 NG samples.

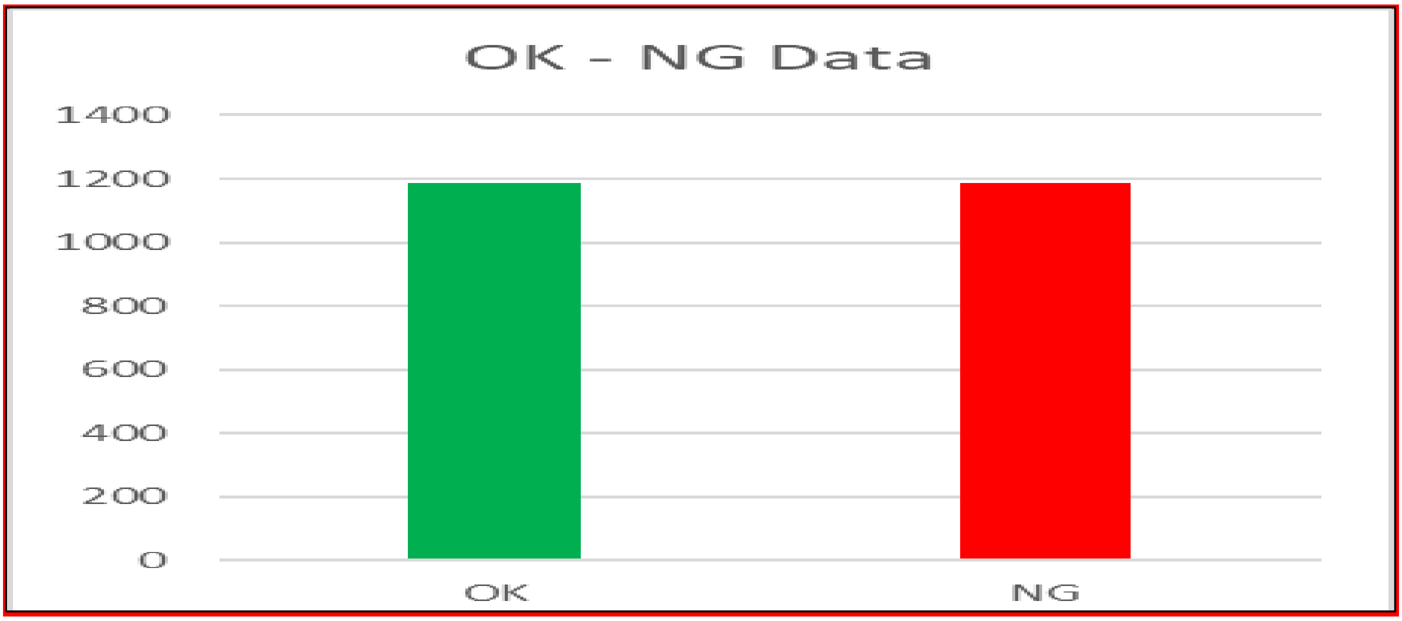

Fig 4 Data Collection Amount

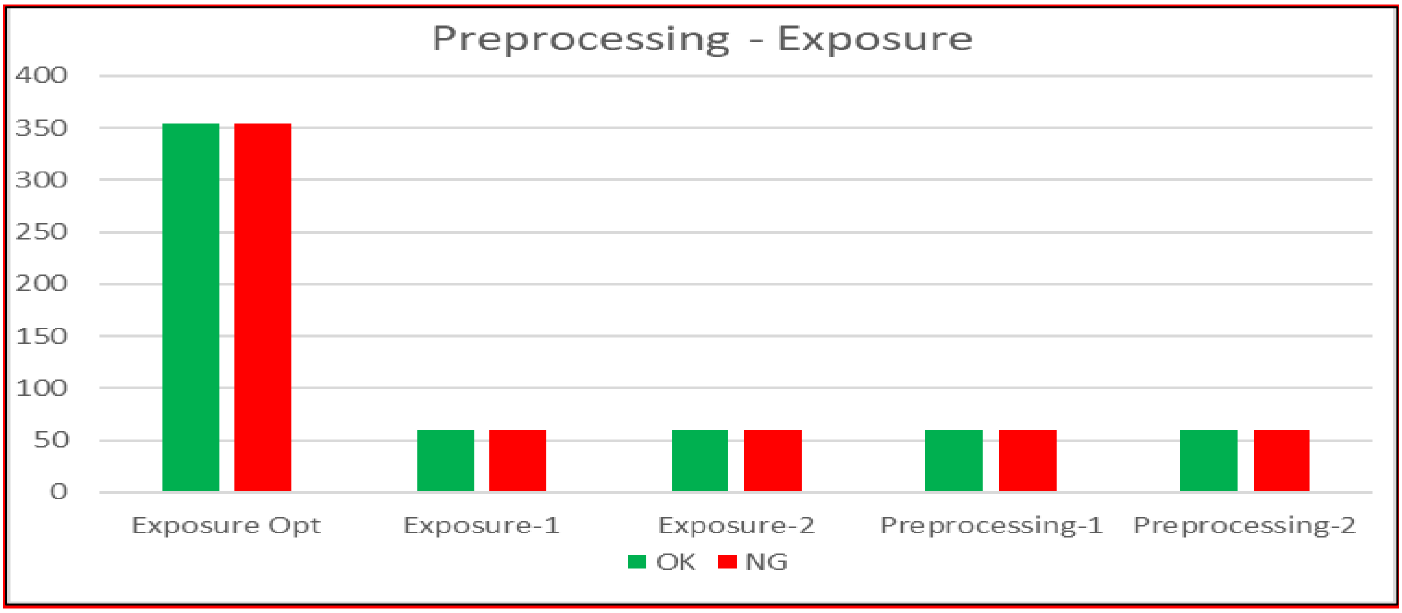

Fig 5 Data Preprocessing

After completing the data collection process, data diversity must be ensured, and data augmentation should be performed to enhance the accuracy and sensitivity of the AI model. Data augmentation has been categorized as shown in Fig.6.

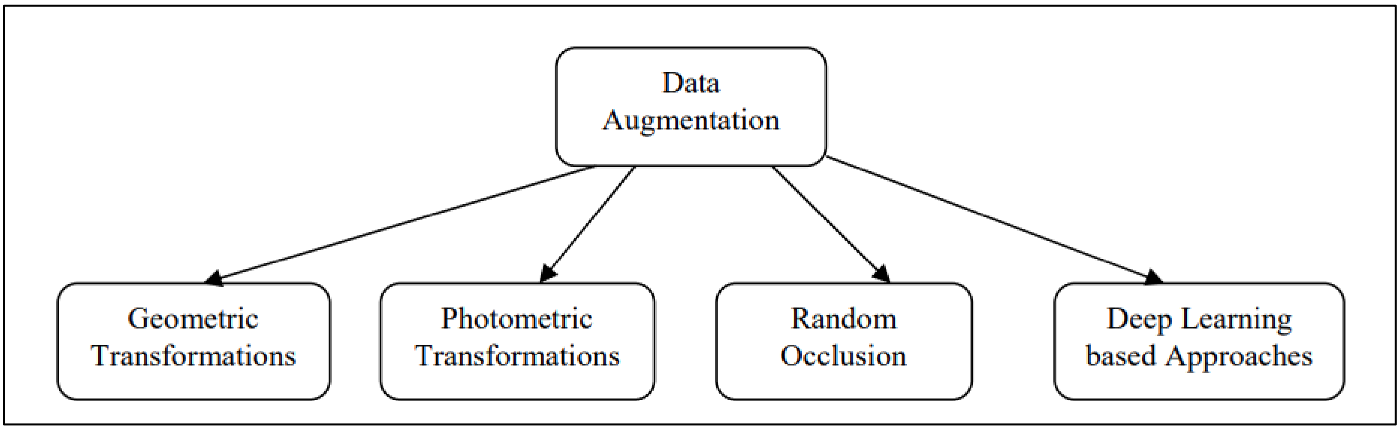

Fig 6 Data Augmentation Methods [18]

Geometric transformations refer to operations such as rotation and cropping. However, since the camera is fixed in this system, geometric transformations were not necessary. Photometric data augmentation, on the other hand, involves altering the pixel colors of existing images to generate additional data. In this project, the photometric data augmentation method was applied. During image collection, the camera was first adjusted to an optimal exposure setting, as shown in Fig.7. Subsequently, to simulate environmental effects and improve model training, the exposure settings were varied to collect images under different conditions, as shown in Fig.8. This approach helps simulate real-world factors such as shadows and lighting variations, allowing the AI model to function more accurately.

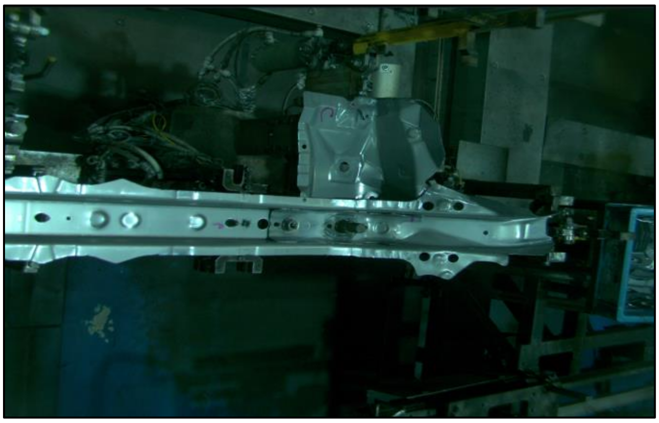

Fig 7 Image with High Exposure

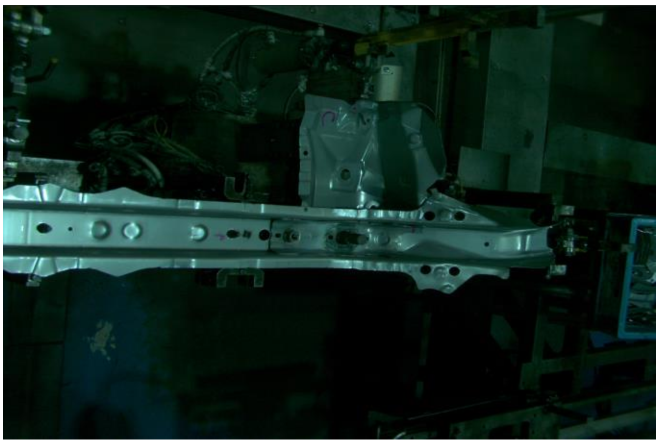

Fig 8 Image with Low Exposure

The random occlusion technique involves modifying collected images by cutting or reducing certain parts. The final method, deep learning-based data augmentation, generates additional data by recreating existing objects using a trained AI model.

➢ *Image Processing - Labeling*

As mentioned earlier in the image collection process, the exposure time was adjusted to simulate environmental effects in the production line. In addition to exposure adjustments, shadows were intentionally created by positioning objects near the structures where the parts are placed, further enhancing model training. Furthermore, with advancements in technology and evolving needs, various algorithms categorized under image preprocessing are used to both augment and diversify the data. Filters such as median filtering were applied to achieve data augmentation and diversification. The increased data variety obtained through preprocessing significantly strengthens the model's accuracy [19].

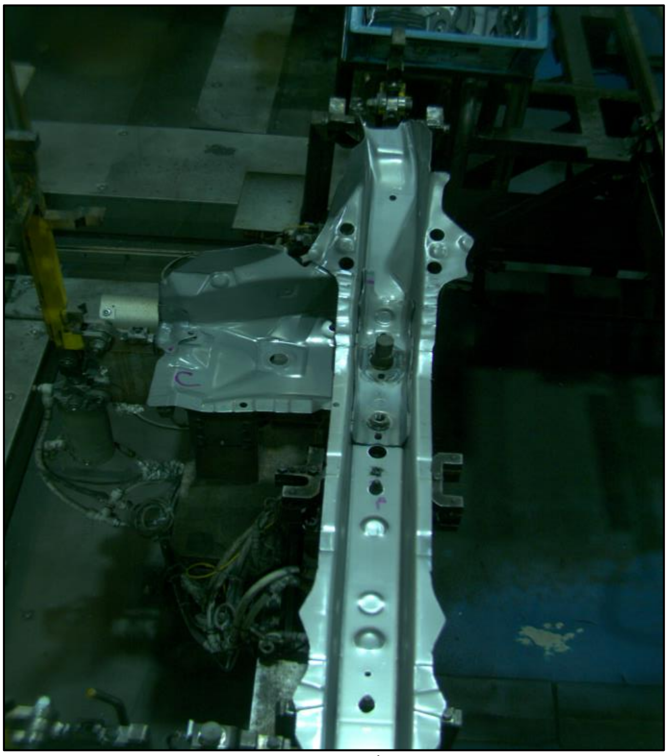

Fig 9 Normal İmage

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar609

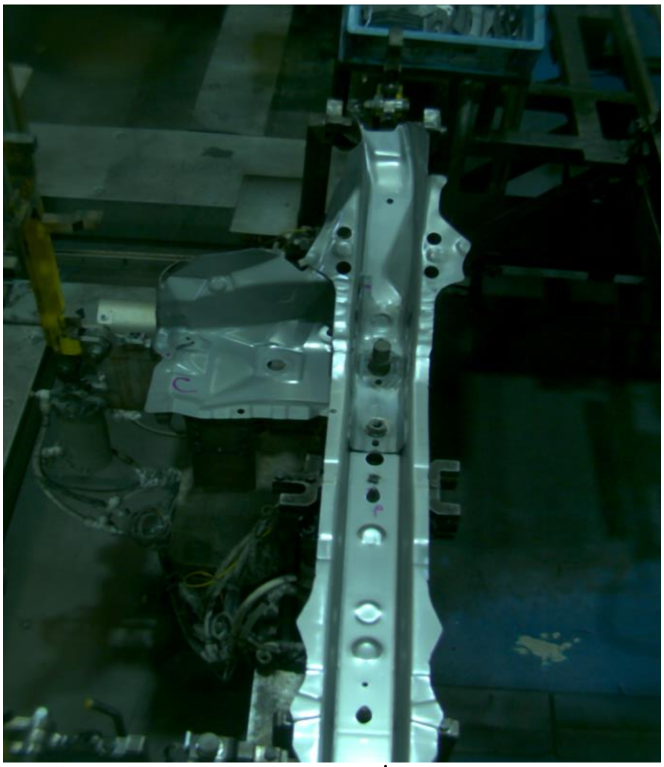
Fig 10 Filtered İmage

After the image collection process, the images need to be labeled to train the AI model and enable it to accurately distinguish between correct and incorrect parts. Image labeling refers to the process of marking the location, bounding box, and class of the object to be detected within an image. The format of the labeling process may vary depending on the requirements of the AI model to be trained. The labeling process was conducted using the open source labelImg program. In each image, the target object's location was marked and classified. To train the AI model, the labeled objects must be documented in .txt format, specifying their respective regions. The labelImg program was used to generate these .txt format files [20].

➢ *Training of the Artificial Intelligence Model*

Bu In this project, the YOLO model was used for object detection. YOLO was first introduced in 2015 through the paper "You Only Look Once: Unified, Real-Time Object Detection" published by Joseph Redmon [5]. As mentioned in the introduction, the YOLO algorithm has outperformed other object detection algorithms in real-time object tracking based on performance evaluation criteria. Since 2015, YOLO has evolved, and multiple versions have been developed. One of the latest and most proven versions, YOLOv7, is an open-source object detection algorithm based on deep learning, specifically convolutional neural networks (CNNs). The YOLOv7 model builds upon previous YOLO versions while providing a unified framework for optimized training models, offering higher speed and accuracy. YOLOv7 is a state-of-the-art object detection algorithm that outperforms many other object detection techniques in both speed and accuracy. By incorporating new techniques in deep learning and computer vision, it represents an advancement over previous YOLO versions such as YOLOv3. Fig.11 illustrates the different versions of YOLO that have been developed over time.

Fig 11 Chronological Development of YOLO

With its advancements, YOLOv7 has started to become an industry standard. The primary reason for this is embedded in its name, "You Only Look Once." As the name suggests, YOLO analyzes an image in a single pass, making it highly efficient and well-suited for real-time applications and environments with limited computational resources. When comparing different YOLO versions, each iteration, including v2, v3, v4, v5, v6, and v7, has introduced key improvements while maintaining the fundamental steps of the YOLO framework. YOLOv2 integrated anchor boxes and introduced

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar609

a new loss function. YOLOv3 implemented a new CNN architecture, replaced bounding boxes with those of varying scales and aspect ratios, and introduced Feature Pyramid Networks. YOLOv4 utilized a new CNN architecture, implemented K-means clustering for anchor boxes, and adopted GHM loss. YOLOv5 incorporated the EfficientDet architecture, dynamic anchor boxes, and Spatial Pyramid Pooling. YOLOv6 used the EfficientNet-L2 architecture and improved dense terminal bounding boxes. YOLOv7, the latest version, introduced nine bounding boxes, optimized feature fusion techniques, and enhanced accuracy and speed.A comparison table of YOLOv7's speed and accuracy against other object detection models can be seen in Fig.12.

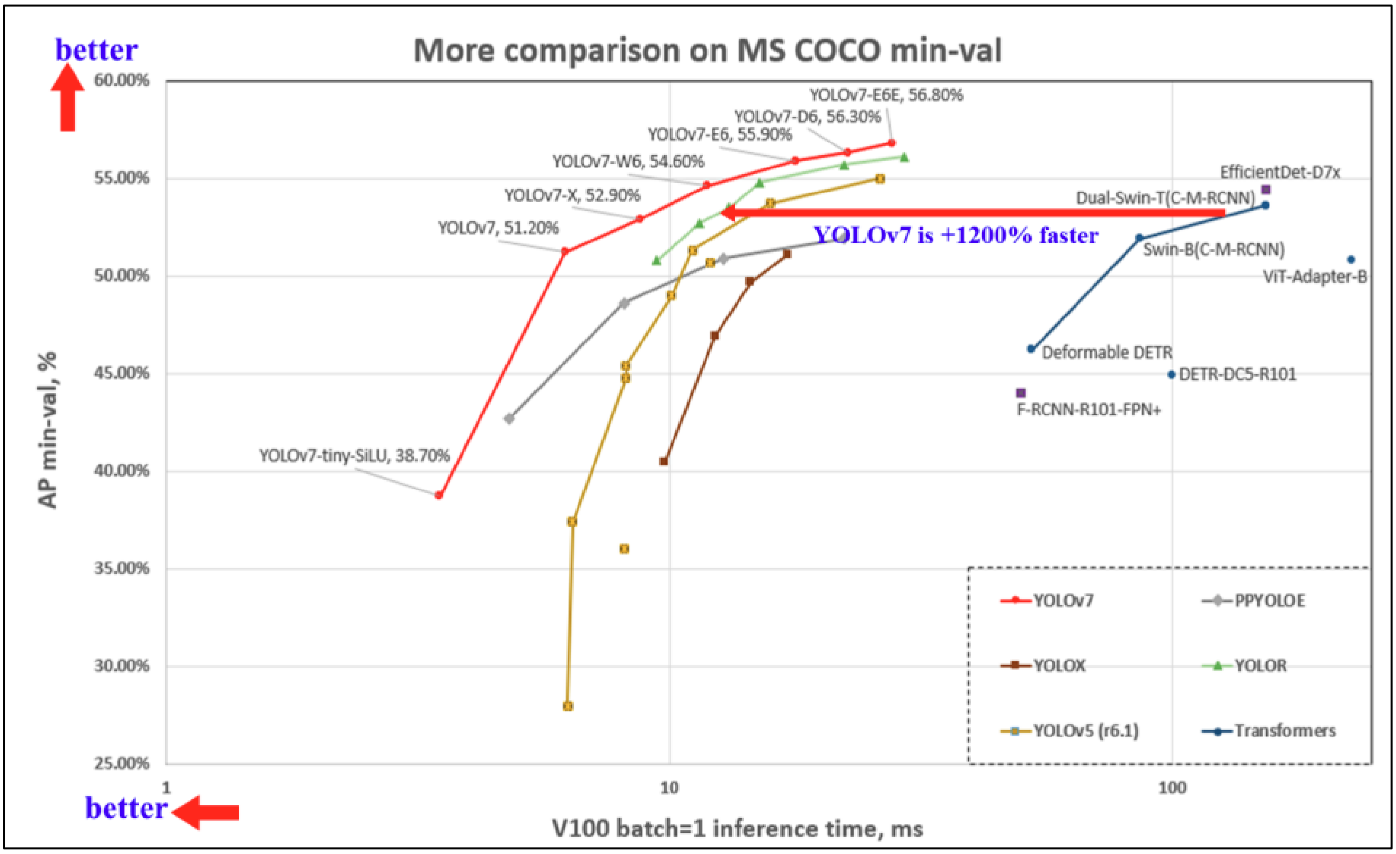

Fig 12 Two stage [21]

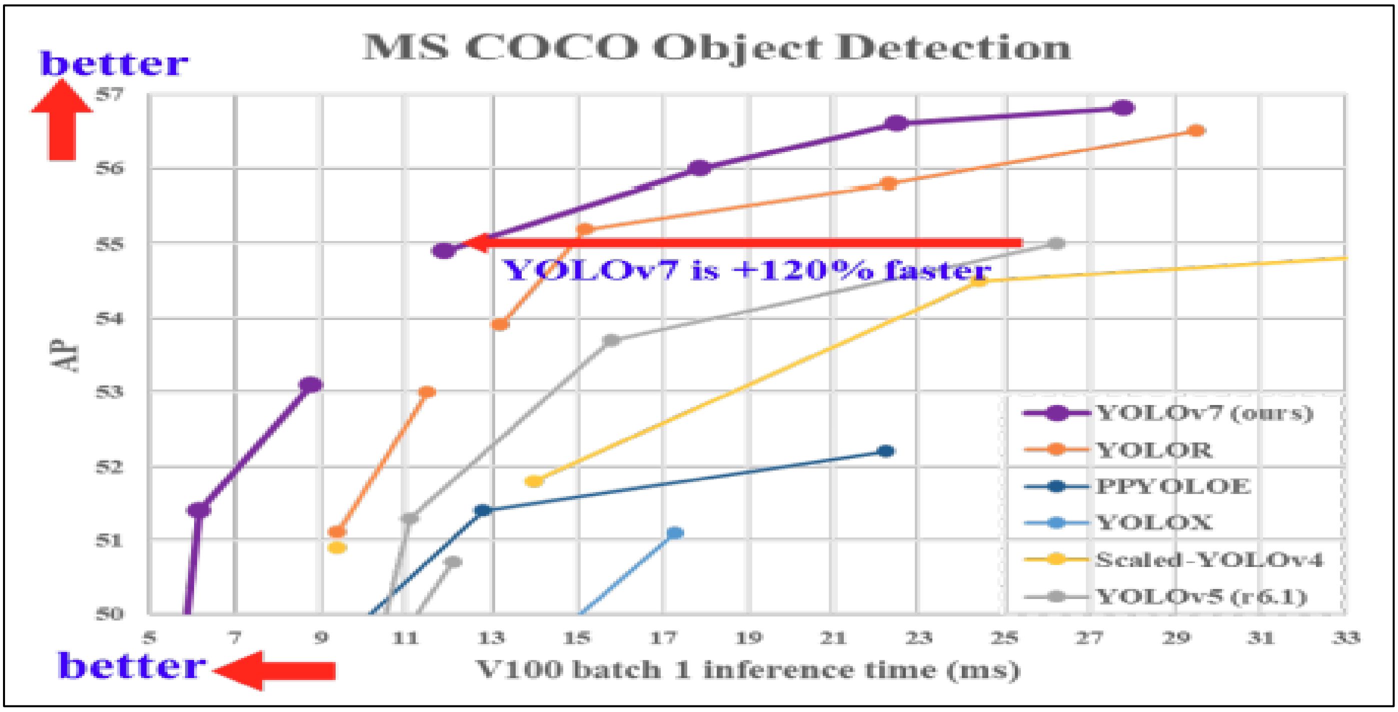

Fig 13 Single Stage [21]

The comparison of speed and accuracy between YOLOv7 and other real-time object detection algorithms can be seen in Fig.13.

As a result of these comparisons, the latest YOLO version, YOLOv7, was chosen for this project. Within YOLOv7 itself, there are different variants designed for various applications. For this project, the YOLOv7-tiny model was selected, as it is optimized for edge devices. The objects were labeled accordingly, and the next step was to train the model. There are multiple methods for model training. Frameworks such as TensorFlow and Keras can be used to create CNN models. However, for better speed and efficiency, the open-source DarkNet framework was preferred. DarkNet is written in C programming language, making it more efficient in terms of performance. Using DarkNet, the YOLOv7-tiny model was trained with the collected dataset.

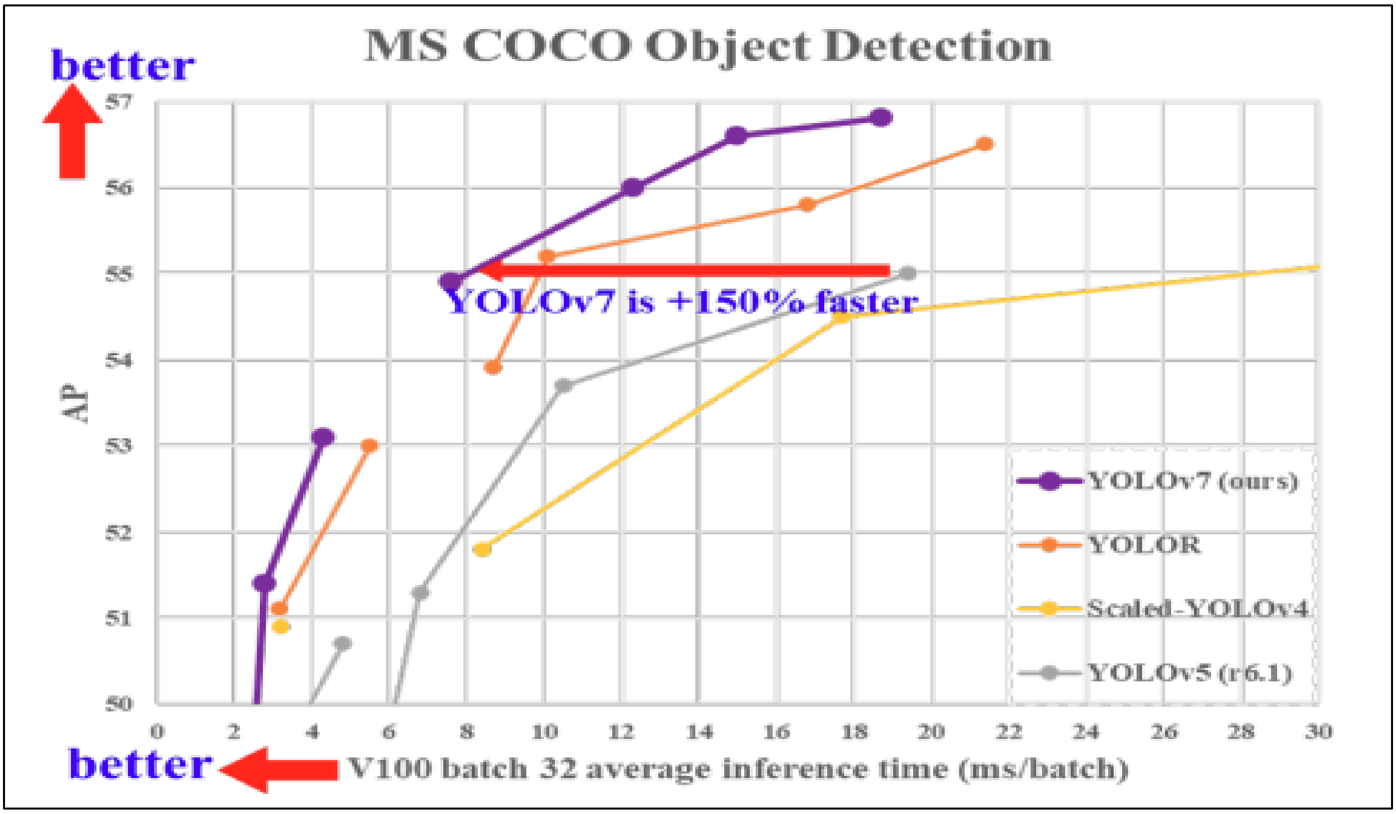

Fig 14 Comparison of YOLO Models [21]

➤ *System Installation, Interface Design, and Model Integration*

For the system installation, a step-by-step approach must be followed. The first step involves conducting a site inspection to analyze the characteristics of the object to be detected, review the conditions of the process where the system will be installed, and assess the site requirements. In the second step, the camera installation area is determined, ensuring that the camera can capture the required angle with an appropriate lens. Additionally, if the camera is installed in an environment affected by external factors, protective equipment must be considered. After setting up the camera, data collection, preprocessing, model selection, and model training are performed. These steps complete the artificial intelligence-related components of the system. To integrate the system into production and inform operators, a control and visualization design is required. First, the process workflow must be reviewed to determine when data should be collected, and which signals will trigger the system. In this project, at the FSM RR Tack 2 station, after the main part and sub-part are positioned on the fixture, a "go to robot" (start welding) signal is sent by pressing a button, which then transmits a signal to the PLC. The system's objective is to inspect the part placement before the "go to robot" signal is sent. The signal from the button to the PLC must also be transmitted to the edge device to capture an image. Various methods, including Ethernet TCP/IP, GPIO, and ModBus, can be used to transfer the PLC signal. In this project, ModBus communication protocol and Adam IO were used to transfer the PLC signal to the edge device. Adam IO is a device that collects dry contacts from the PLC or any other device and transmits them via ModBus. Using Adam IO, the "go to robot" signal from the PLC was sent to the edge device. This setup ensures that the system is triggered by the signal, captures an image, and performs part feeding verification using the AI model. To implement the "go to robot" signal control, modifications were required in the PLC software. The AI model's control signal was added as a condition before executing the "go to robot" command. The AI model's verification signal is transmitted to the PLC via Adam IO, and the robot proceeds only if the verification is successful. Once the inspection is completed, the result should be displayed to the operator via the user interface. The user interface can be designed using various Python libraries such as Tkinter, Kivy, wxPython, and PyQt. In this project, the PyQt5 library was used for UI design. The interface dynamically updates based on the AI object detection

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar609

algorithm. After capturing an image, the AI model detects the object, classifies it, places a bounding box around the detected object, and updates the UI display. Depending on whether the part is correctly or incorrectly placed, the designated area in the UI changes color to indicate the result. For the system to function properly, different components must operate independently of one another. A key point to emphasize is that the image capture signal from ModBus must be continuously monitored, while the UI must also remain operational. Alternatively, after receiving the image capture signal, both the AI model and the UI program must run continuously. In other words, a process must be capable of handling multiple tasks simultaneously. This is achieved using threads. Threads are also referred to as lightweight processes, and the concept of multi-threading allows multiple threads to run within a single process. On multi-core processors, these threads can run concurrently on different cores, a technique known as parallel programming. In summary, threads are utilized in this system to ensure that different functions operate simultaneously without interference, enabling continuous operation.

➢ *Performance Evaluation Criteriar*

In this study, statistical metrics such as accuracy, sensitivity (recall), specificity, precision, and F1-score were used to evaluate the performance of the model. These metrics were selected to assess the model's classification accuracy, its ability to minimize false positives and false negatives, and its reliability independent of random predictions.

- *The Performance values were Calculated as Follows:*

$$\textbf{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$\textbf{Sensitivity} = \frac{TP}{TP+FN} \quad (2)$$

$$\textbf{Specificity} = \frac{TN}{TN+FP} \quad (3)$$

$$\textbf{Precision} = \frac{TP}{TP+FP} \quad (4)$$

$$\textbf{F1-Score} = 2 \text{ x } \frac{Precision \text{ X } Sensitivity}{Precision + Sensitivity} \quad (5)$$

TP, FP, TN, and FN concepts originate from the confusion matrix, which is used to analyze the performance of a classification model in detail. This matrix categorizes predicted results against actual results into four groups:

- True Positives (TP): Cases where the model correctly predicts a positive outcome.
- False Positives (FP): Cases where the model incorrectly predicts a positive outcome (actually negative).
- False Negatives (FN): Cases where the model incorrectly predicts a negative outcome (actually positive).
- True Negatives (TN): Cases where the model correctly predicts a negative outcome.

The confusion matrix is structured to provide a detailed analysis of correct and incorrect predictions. If TP and TN values are high, it indicates that the model has a high success rate in making correct predictions. Conversely, low FP and FN values suggest that the model makes fewer errors.

IV. RESULTS

The model's performance is visualized using the **Confusion Matrix**, as shown in **Table 1**.

The preprocessing steps applied significantly improved the model's performance. Initially, with raw data, the accuracy was 88%, sensitivity 90%, specificity 70%, precision 95%, and F1-score 87%. The first step, grayscale conversion, resulted in slight improvements, increasing accuracy to 89% and sensitivity to 91%. Noise reduction further enhanced the model's performance, raising accuracy to 90% and sensitivity to 92%, marking a significant improvement. Data normalization improved the results even more, increasing accuracy to 91% and specificity to 78%. Following this, edge detection helped refine the model's accuracy to 94% and precision to 99%, contributing significantly to performance enhancement. Finally, by combining all preprocessing techniques, the results became highly satisfactory, with accuracy reaching 98.07%, sensitivity 98.07%, and specificity 98.15%. The precision improved to 99.89%, while the F1-score reached 98.97%, significantly boosting the model's overall success. These improvements clearly demonstrate how the preprocessing steps in the image processing workflow have enhanced the model's accuracy, sensitivity, specificity, and overall performance.

Table 1 Confusion Matrix

	Actual Positive	Actual Negative
Predicted Positive	915	1
Predicted Negative	18	53

Table 2 Pre-processing and Results

İşlem	TP	FP	TN	FN	Accuracy	Sensitivity	Specificity	Precision	F1-Score
Raw Data	900	20	50	30	0,88	0,9	0,7	0,95	0,87
Grayscale Conversion	905	18	52	28	0,89	0,91	0,72	0,96	0,88
Noise Reduction	910	15	53	25	0,9	0,92	0,75	0,97	0,89
Data Normalization	912	10	54	22	0,91	0,93	0,78	0,98	0,9
Edge Detection	913	5	55	18	0,94	0,95	0,8	0,99	0,92
Result	915	1	53	18	0,9807	0,9807	0,9815	0,9989	0,989

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar609

V. DISCUSSION

This study focuses on industrial object recognition systems using the YOLOv7-tiny algorithm. Similarly, the study "Decision Support System Based on YOLOv7 Algorithm for Brain Tumor Diagnoses" explored the application of YOLOv7 and YOLOv7-tiny algorithms in medical image analysis [22]. Both studies demonstrate the effectiveness of the YOLO algorithm across different fields and its capability to provide practical solutions. In this study, the YOLOv7-tiny model achieved an accuracy rate of 98.07%. In contrast, the study "Comparative Analysis of Deep Learning Image Detection Algorithms" reported accuracy rates of 85% for Faster R-CNN, 74% for SSD, and 80% for YOLOv3 [23]. Similarly, the "Decision Support System Based on YOLOv7 Algorithm for Brain Tumor Diagnoses" study reported an accuracy rate of 97%. These results highlight the superior accuracy of the YOLOv7-tiny model in both industrial and other domains. Additionally, YOLOv7-tiny achieved a speed of 160 FPS, significantly outperforming other models. In the "Comparative Analysis of Deep Learning Image Detection Algorithms" study, the FPS values for Faster R-CNN, SSD, and YOLOv3 were 8 FPS, 46 FPS, and 25 FPS, respectively [23]. This clearly demonstrates the superior speed and accuracy of YOLOv7-tiny in real-time applications.

VI. CONCLUSION

This study has demonstrated that YOLOv7-tiny outperforms other models in terms of both speed and accuracy. The significant advantage of this speed difference is particularly evident in real-time applications. The developed system achieved 98.07% accuracy in an industrial production line, effectively minimizing human errors in real-time operations. The modern architecture and optimized features of YOLOv7-tiny have proven to provide a highly efficient solution. For industrial and real-time applications, choosing YOLOv7-tiny offers a highly efficient and effective approach, making it an ideal model for automation and quality control systems.

REFERENCENS

[1]. K. Xu, N. Ragot, and Y. Dupuis, "View Selection for Industrial Object Recognition," in IECON Proceedings (Industrial Electronics Conference), IEEE Computer Society, 2022. doi: 10.1109/IECON49645.2022.9969121.

[2]. A. Shrestha, N. Karki, R. Yonjan, M. Subedi, and S. Phuyal, "Automatic Object Detection and Separation for Industrial Process Automation," in 2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science, SCEECS 2020, Institute of Electrical and Electronics Engineers Inc., Feb. 2020. doi: 10.1109/SCEECS48394.2020.195.

[3]. C. Boukouvalas et al., "ASSIST: automatic system for surface inspection and sorting of tiles," 1998.

[4]. H. M. T. Abbas, U. Shakoor, M. J. Khan, M. Ahmed, and K. Khurshid, "Automated sorting and grading of agricultural products based on image processing," in 2019 8th International Conference on Information and Communication Technologies, ICICT 2019, 2019. doi: 10.1109/ICICT47744.2019.9001971.

[5]. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016. doi: 10.1109/CVPR.2016.91.

[6]. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Trans Pattern Anal Mach Intell, vol. 39, no. 6, 2017, doi: 10.1109/TPAMI.2016.2577031.

[7]. W. Liu et al., "SSD: Single shot multibox detector," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2016. doi: 10.1007/978-3-319-46448-0_2.

[8]. J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017. doi: 10.1109/CVPR.2017.690.

[9]. R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2014. doi: 10.1109/CVPR.2014.81.

[10]. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2015. doi: 10.1007/978-3-319-24574-4_28.

[11]. L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," IEEE Trans Pattern Anal Mach Intell, vol. 40, no. 4, 2018, doi: 10.1109/TPAMI.2017.2699184.

[12]. X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in Proceedings of the 28th International Conference on Machine Learning, ICML 2011, 2011.

[13]. A. Raghunathan, S. M. Xie, F. Yang, J. C. Duchi, and P. Liang, "Understanding and mitigating the tradeoff between robustness and accuracy," in 37th International Conference on Machine Learning, ICML 2020, 2020.

[14]. E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," IEEE Trans Pattern Anal Mach Intell, vol. 39, no. 4, 2017, doi: 10.1109/TPAMI.2016.2572683.

[15]. L. Binyan, W. Yanbo, C. Zhihong, L. Jiayu, and L. Junqin, "Object detection and robotic sorting system in complex industrial environment," in Proceedings - 2017 Chinese Automation Congress, CAC 2017, 2017. doi: 10.1109/CAC.2017.8244092.

[16]. Y. Zuo, J. Wang, and J. Song, "Application of YOLO Object Detection Network in Weld Surface Defect Detection," in 2021 IEEE 11th Annual International Conference on CYBER Technology in Automation,

Control, and Intelligent Systems, CYBER 2021, Institute of Electrical and Electronics Engineers Inc., Jul. 2021, pp. 704–710. doi: 10.1109/CYBER53097.2021.9588269.

[17]. "Industrial Cameras, Technical Features, and Market," Optik & Photonik, vol. 13, no. 1, 2018, doi: 10.1002/opph.201870108.

[18]. P. Kaur, B. S. Khehra, and E. B. S. Mavi, "Data Augmentation for Object Detection: A Review," in Midwest Symposium on Circuits and Systems, 2021. doi: 10.1109/MWSCAS47672.2021.9531849.

[19]. Y. L. Ao, "Introduction to Digital Image Pre-processing and Segmentation," in Proceedings - 2015 7th International Conference on Measuring Technology and Mechatronics Automation, ICMTMA 2015, 2015. doi: 10.1109/ICMTMA.2015.148.

[20]. K. E. Varnima and C. Ramachandran, "Real-time Gender Identification from Face Images using you only look once (yolo)," in Proceedings of the 4th International Conference on Trends in Electronics and Informatics, ICOEI 2020, 2020. doi: 10.1109/ICOEI48184.2020.9142989.

[21]. C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," 2023. doi: 10.1109/cvpr52729.2023.00721.

[22]. S. YILMAZ, "Beyin Tümörü Tanıları İçin YOLOv7 Algoritması Tabanlı Karar Destek Sistemi Tasarımı," Kocaeli Üniversitesi Fen Bilimleri Dergisi, vol. 6, no. 1, pp. 47–56, Jul. 2023, doi: 10.53410/koufbd.1236305.

[23]. S. Srivastava, A. V. Divekar, C. Anilkumar, I. Naik, V. Kulkarni, and V. Pattabiraman, "Comparative analysis of deep learning image detection algorithms," J Big Data, vol. 8, no. 1, p. 66, Dec. 2021, doi: 10.1186/s40537-021-00434-w.