International Journal of Electrical Engineering and Intelligent Computing (IJEEIC) EISSN 3031-5255
https://ejournal.unkhair.ac.id/index.php/ijeeic/index

Smart Parking based on Car Detection using Deep Learning YOLOv8

Waluyo Nugroho[1], Afianto[2], Mada Jimmy Fonda Arifianto[3]
[1,2,3] Mechatronics Department, Astra Polytechnic, West Java, Indonesia
nugroho.research@gmail.com

Article history : Received August 23, 2024 | Revised Des 22, 2024 | Accepted Des 24, 2024

In the context of rapidly growing urbanization, the need for efficient parking management solutions is becoming increasingly urgent. This research develops and implements a car detection system based on YOLOv8 (You Only Look Once Version 8) for smart parking applications using Raspberry Pi and the Node-RED platform. This system is designed to optimize the use of parking spaces and increase parking management efficiency by utilizing YOLO's real-time object detection capabilities. Data processed by the Raspberry Pi is sent to the Node-RED platform for Internet of Things (IoT) via MQTT protocol. Node-RED functions as a management and visualization system, allowing users to monitor parking status in real-time through an intuitive graphical interface. With Node-RED, users can find out which parking lots are full and which areas are still available.

Keywords: Internet of Things, MQTT, Node-RED, Raspberry Pi, YOLO

1. INTRODUCTION

The rapid urbanization and the surge in the number of vehicles on the road have led to a growing need for efficient parking solutions in urban environments [1]. Traditional parking management systems often rely on manual supervision or simplistic sensor-based technologies, which can result in inefficiencies, such as inaccurate space allocation, delays, and increased congestion. To address these challenges, there is a pressing need for smart parking systems that leverage advanced technologies to optimize parking space usage and enhance user experience [2].

This paper presents a Smart Parking system based on Car Detection using Computer Vision, specifically employing the YOLO (You Only Look Once) object detection algorithm [3]. YOLO is renowned for its real-time object detection capabilities, making it an ideal choice for dynamic environments like parking lots. By utilizing computer vision, the proposed system can accurately detect and monitor the occupancy of parking spaces in real time, reducing the need for physical sensors and enabling more efficient space management [4].

The implementation of YOLO in this context not only ensures high accuracy in vehicle detection but also contributes to a scalable and cost-effective parking solution [5]. This system can be integrated with existing infrastructure, providing a seamless upgrade to current parking facilities without the need for extensive modifications. The result is a smarter, more efficient parking system that can significantly reduce the time spent searching for available parking spaces, thus alleviating traffic congestion and enhancing the overall urban mobility experience. The following sections will delve into the design and implementation of the Smart Parking system, the advantages of using YOLO for car detection, and the potential impact of this technology on modern urban environments [6].

Previous research used Raspberry Pi and webcam as the main components to detect cars, with simple algorithms such as OpenCV-based image processing or threshold methods to detect objects [7]. While the current research uses Deep Learning YOLO for car detection, the detection results also interact with Node-RED as an automation platform that facilitates the integration of various system components, including data flow management, notification delivery, and physical control devices. Node-RED integration allows the system to create a more flexible workflow without having to write a lot of code, because Node-RED uses a graphical interface that is easier to connect hardware, software, and cloud services.

2. METHOD

This research method develops a Python-based real-time vehicle detection system using Raspberry Pi and Node-RED Integration. This process begins by taking real-time video input from a webcam camera. This camera captures images of the monitored parking area. This image taken directly will be input for the YOLO algorithm running on the Raspberry Pi. After YOLO processes the image, its classification results are output. This classification includes two main categories: "Cars" and "No Cars". The classification results are then sent to Node-RED, a platform for visualizing and managing data flows. Node-RED can be used to display detection results. Figure 1 shows the flow chart of the webcam and raspberry pi process system

DOI: 10.33387/ijeeic.v2i1.8692

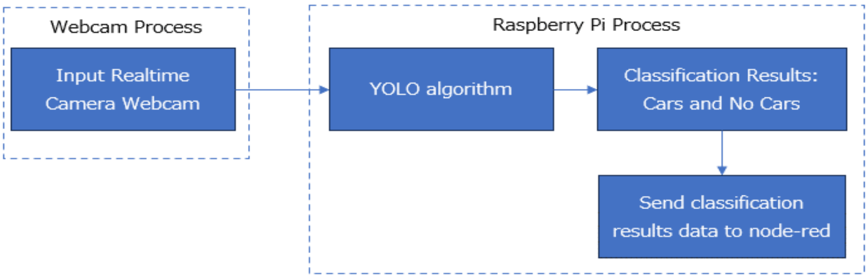

Webcam Process
Input Realtime Camera Webcam
Raspberry Pi Process
YOLO algorithm
Classification Results: Cars and No Cars
Send classification results data to node-red

Figure 1. Flowchart System

2.1 Datasets

This dataset is designed for the development and evaluation of smart parking systems using car detection techniques based on the YOLO (You Only Look Once) deep learning model. The dataset contains 1000 images and corresponding annotations of parking lots with vehicles and empty parking spaces, captured under various conditions. The dataset consists of high-resolution images of parking lots captured from various angles, including overhead and ground-level views [8]. The images were taken during different times of the day to include variations in lighting and weather conditions. Each image is annotated with bounding boxes that identify the location of vehicles within the parking lot [9]. The annotations also include labels indicating whether a parking space is occupied or vacant.

2.2 Vehicle Detection

The webcam is used as a camera for car detection with YOLO (You Only Look Once) using 1080p resolution, resulting in clear and detailed images. Good resolution helps YOLO detect objects (cars) more accurately. The frame rate used is 30 fps with auto focus. Frame rate is important to ensure that the captured video is not choppy, especially when detecting fast moving cars [10]. Higher frame rates allow the YOLO model to analyze more frames per second, improving detection accuracy. The results of reading the webcam image will be processed using a Raspberry Pi.

Raspberry pi will process data from the webcam using the YOLO algorithm, so it will be able to detect cars [11]. The detection results will be sent using MQTT communication and displayed on the Node-RED dashboard. The Raspberry Pi can be set up as an MQTT server, acting as a broker for communication between devices in an IoT network. MQTT is a lightweight communications protocol suitable for devices with limited resources, such as the Raspberry Pi. Figure 2 shows the Block diagram of smart parking system.

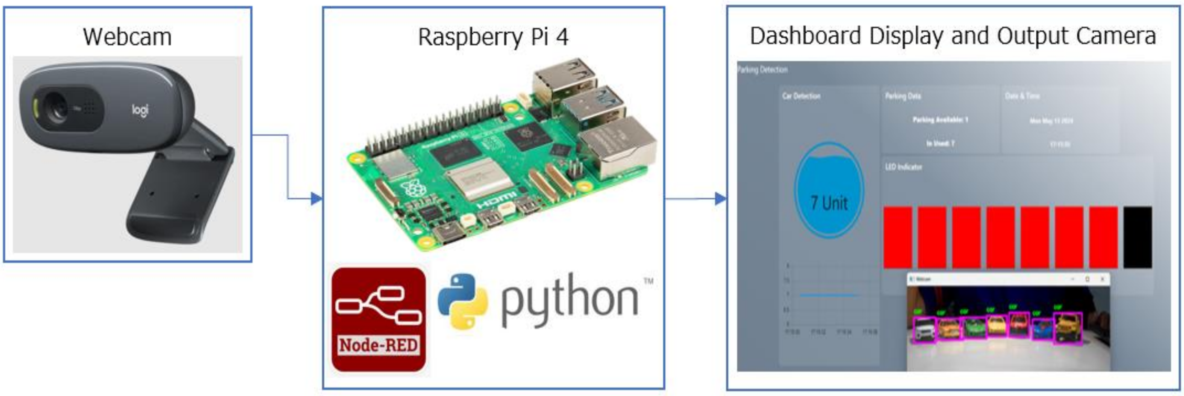


Figure 2. Block diagram of smart parking system

List of Hardware and Software:

1. Webcam Logitech C310 720p Camera with Auto Focus
2. Raspberry Pi 4B
3. Node-red v20.11.0
4. Python v3.12.2
5. Microsoft Visual Studio Code

System Algorithm:

The webcam functions as a sensor to capture images or videos from the parking area. The webcam installed in the parking lot will continuously record or take pictures of the parking area. This camera will capture live video to be explained by the system. Raspberry Pi is a mini computer used to process data received from the webcam. Raspberry Pi for the car detection algorithm and controls the overall parking system. Raspberry Pi receives images/videos from the webcam and processes them using the YOLO algorithm.

The YOLO algorithm breaks the image into a grid and predicts the possibility of an object (car) in each part of the grid. YOLO will provide output in the form of a bounding box around the detected car, along with the object class. Writing programming on Raspberry Pi and the YOLO algorithm uses the python language and the Microsoft Visual Studio Code compiler. Based on the car detection carried out by YOLO, the system can find out which parking slots are empty and which are occupied. Raspberry Pi will store this data for automatic parking lot management. If a car is detected, the system will record it as an occupied parking space. If a car is not detected in a particular parking slot, the slot will be considered empty. The results of the number of cars and parking area slots will be sent to Node-RED. Node-RED allows the integration of various devices and services without the need for much coding. Node-RED is used to manage communication between Raspberry Pi, YOLO algorithm, and other hardware or software such as additional sensors, information displays, or web applications. Node-RED also has a dashboard feature that allows the creation of a visual interface to unify and manage the parking system. This dashboard can display parking space availability in real-time. This information can be accessed from a web browser or other device with an internet connection.

2.3 Node-Red Design

Node-RED enables the integration of various devices and services without the need for much coding. In the context of a parking system, Node-RED can be used to manage the communication between the Raspberry Pi, the YOLO algorithm, and other hardware or software such as additional sensors, information displays, or web applications. When the Raspberry Pi detects a car in a parking slot, Node-RED can handle this data and send further information, such as updating the display in the user interface. A Node-RED flow diagram is shown in Figure 3.

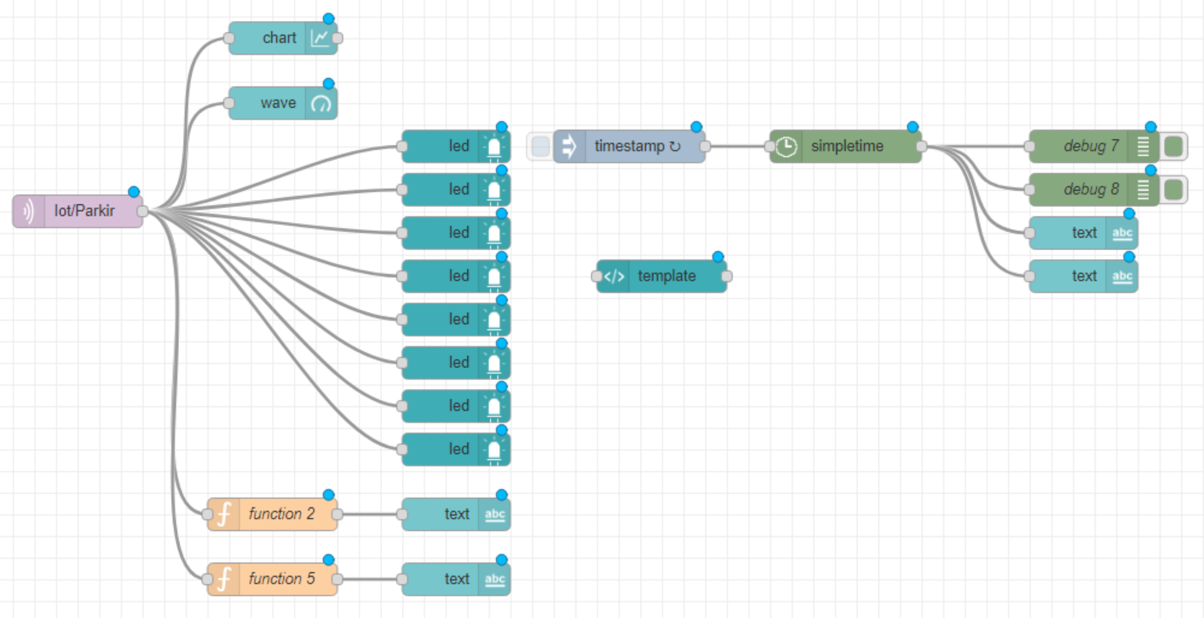


Figure 3. Block Diagram Flow Node-RED

2.4 YOLOv8 Object Detection

YOLOv8 (You Only Look Once version 8) represents the latest evolution in the YOLO series, designed for real-time object detection with improved efficiency, accuracy, and speed [12]. This version integrates detection and segmentation into a unified architecture, building upon the strengths of previous versions like YOLOv3, YOLOv4, YOLOv5, and YOLOv7. The YOLOv8 architecture is composed of several crucial components that work together to detect objects within images[13]. Below is a detailed explanation of each component such as Backbone, Neck, Head. The backbone is a convolutional network that extracts features from the input image [14]. YOLOv8 uses a lightweight yet powerful backbone inspired by modern architectures like CSP Darknet (Cross Stage Partial Network) and CSP ResNet. The backbone's main role is to capture various features at different levels of the image, such as edges, textures, shapes, and objects. The neck enhances the feature representation extracted by the backbone by merging information from different scales [15].

YOLOv8 utilizes structures like FPN (Feature Pyramid Network) and PAN (Path Aggregation Network) to maintain high-resolution information across multiple feature levels, which is crucial for detecting objects of varying sizes. FPN (Feature Pyramid Network): Produces feature maps at different scales, which are then combined to enrich spatial and context-aware information at all resolution levels. PAN (Path Aggregation Network): Facilitates the bottom-up flow of information, improving the detection of smaller objects [16]. The head is responsible for making final predictions, including bounding boxes, confidence scores, and object classes[17]. YOLOv8 features multiple output heads connected to features at various scales, allowing the model to detect both large and small objects with high precision. Each head performs regression to generate bounding box predictions and object classification [18].

YOLOv8 can be implemented using deep learning frameworks like TensorFlow. It is suitable for applications requiring real-time object detection, such as video surveillance, self-driving systems, and augmented reality[19]. Additionally, with support for inference on resource-constrained devices, YOLOv8 is ideal for deployment on edge devices like smartphones and smart cameras [20]. YOLOv8 represents a significant advancement in object detection, offering an outstanding balance between speed and accuracy [21]. With its more efficient architecture and anchor-free approach, YOLOv8 is well-equipped to handle various challenges in complex environments for object detection. Figure 3 shows the architecture of YOLOv8.

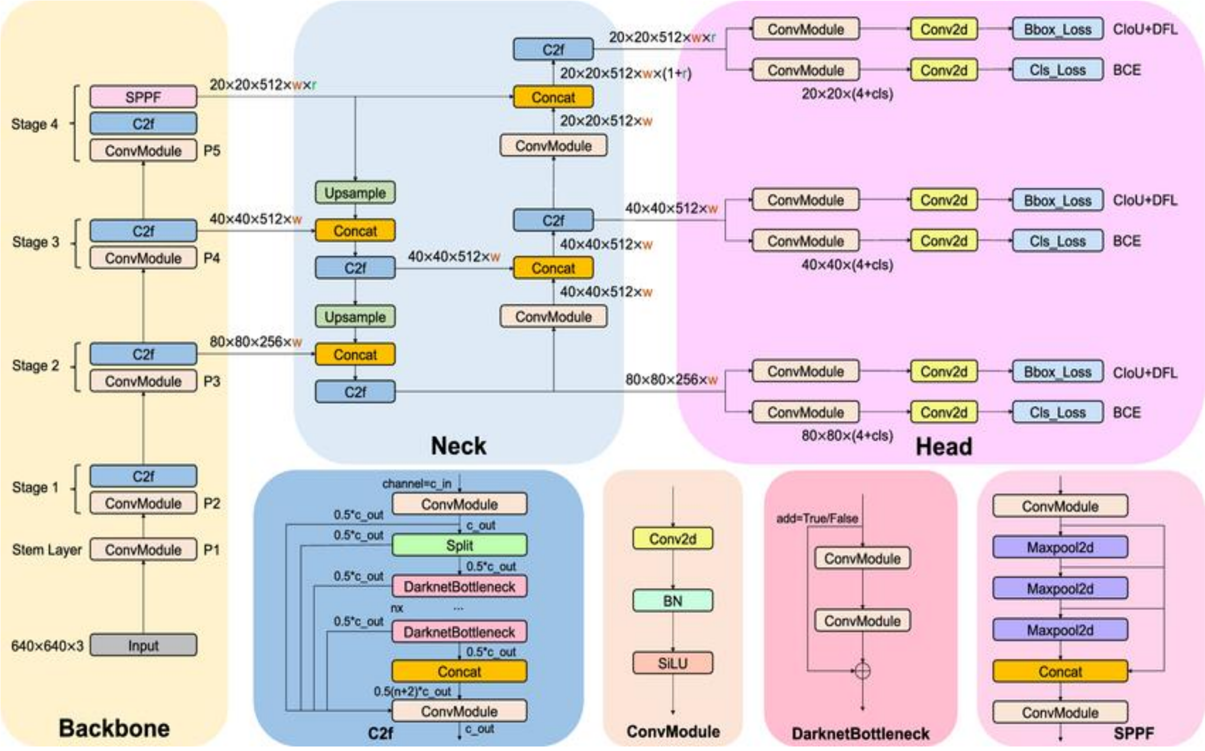


Figure 4. YOLOv8 object detection architecture

3. RESULTS AND DISCUSSION

The smart parking system uses the YOLO algorithm on Raspberry Pi and Node-RED is used to detect objects, such as cars to help vehicle users see empty and filled parking areas automatically. YOLO is a very efficient object detection algorithm and can be used in real-time systems. By utilizing the Raspberry Pi's computing capabilities, this system can operate independently and integrate with Node-RED to display smart parking data. Figure 4 shows the system developed successfully detects vehicles in real-time and manages parking space availability.

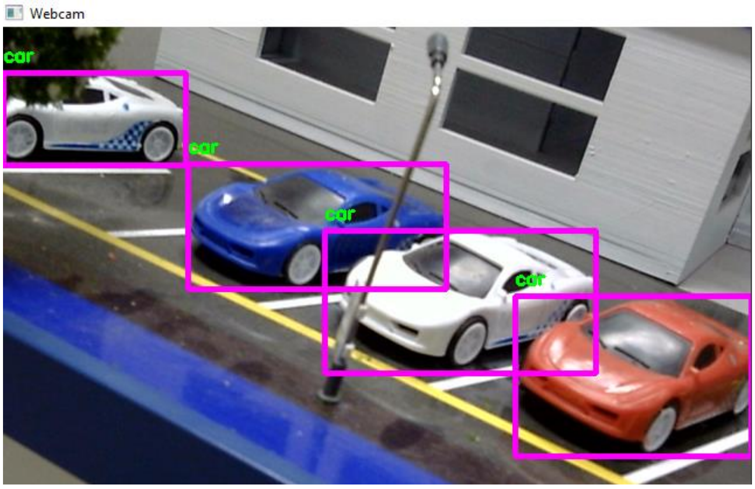
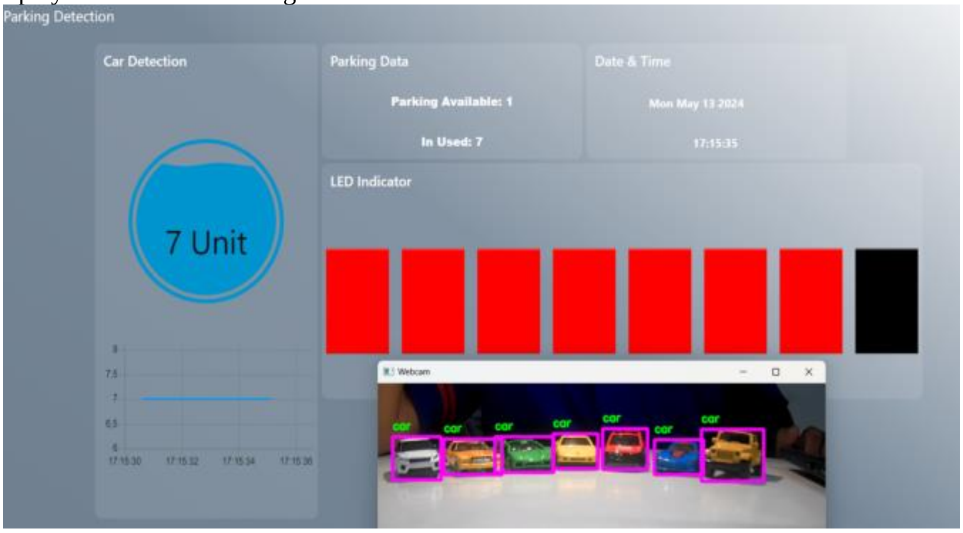


Figure 5. Car detection

Integration with Node-RED allows for easier monitoring. This program display is very useful in monitoring parking status in an area, providing real-time information about available and occupied parking areas, making it easier to manage parking and minimizing the time spent looking for a parking space. Figure 5 shows the dashboard display of car detection integration with Node-RED.


Parking Detection

Car Detection

7 Unit

Parking Data

Parking Available: 1

In Used: 7

Date & Time

Mon May 13 2024

17:15:35

LED Indicator

Figure 6. Car detection integration with Node-RED

4. CONCLUSION

The development of a real-time vehicle detection system using webcam, python pada raspberry pi, and Node-RED has been successfully carried out. This project utilizes computer image processing technology to detect moving vehicles in videos captured by live webcams. The use of Node-RED allows integration between the detection process and wider systems, such as the parking detection system that has been created.

The smart parking system using YOLO with Raspberry Pi and Node-RED provides an effective solution for automatic object detection in the vehicle parking process. Despite some limitations, especially in the computing capabilities of the Raspberry Pi, this system shows great potential for application in a variety of automated parking applications in the future. Further optimization and use of more powerful hardware will make this system more reliable and ready for commercial use.

REFERENCES

[1] S. S. Channamallu, S. Kermanshachi, J. M. Rosenberger, and A. Pamidimukkala, “A review of smart parking systems,” in *Transportation Research Procedia*, Elsevier B.V., 2023, pp. 289–296. doi: 10.1016/j.trpro.2023.11.920.

[2] A. Elomiya, J. Krupka, and S. Jovcic, “A Smart Parking System Using Surveillance Cameras and Fuzzy Logic: A Case Study at Pardubice University’s Campus,” in *Procedia Computer Science*, Elsevier B.V., 2023, pp. 4881–4890. doi: 10.1016/j.procs.2023.10.488.

[3] W. A. Jabbar, L. Y. Tiew, and N. Y. Ali Shah, “Internet of things enabled parking management system using long range wide area network for smart city,” *Internet of Things and Cyber-Physical Systems*, vol. 4, pp. 82–98, Jan. 2024, doi: 10.1016/j.iotcps.2023.09.001.

[4] M. S. Nazar, P. Jafarpour, M. Shafie-khah, and J. P. S. Catalão, “Optimal planning of self-healing multi-carriers energy systems considering integration of smart buildings and parking lots energy resources,” *Energy*, vol. 286, Jan. 2024, doi: 10.1016/j.energy.2023.128674.

[5] O. G. Ajayi, J. Ashi, and B. Guda, “Performance evaluation of YOLO v5 model for automatic crop and weed classification on UAV images,” *Smart Agricultural Technology*, vol. 5, Oct. 2023, doi: 10.1016/j.atech.2023.100231.

[6] C. Yu and Y. Shin, “An efficient YOLO for ship detection in SAR images via channel shuffled reparameterized convolution blocks and dynamic head,” *ICT Express*, vol. 10, no. 3, pp. 673–679, Jun. 2024, doi: 10.1016/j.icte.2024.02.007.

[7] T. Bawankule, P. Kalbhor, and A. Kumar Patil, “Smart Car Parking System Using Image Processing,” 2024. [Online]. Available: www.ijrpr.com

[8] L. Ortenzi *et al.*, “Automated species classification and counting by deep-sea mobile crawler platforms using YOLO,” *Ecol Inform*, p. 102788, Aug. 2024, doi: 10.1016/j.ecoinf.2024.102788.

[9] J. Feng and T. Jin, “CEH-YOLO: A composite enhanced YOLO-based model for underwater object detection,” *Ecol Inform*, vol. 82, Sep. 2024, doi: 10.1016/j.ecoinf.2024.102758.

[10] T. B. Pun, A. Neupane, R. Koech, and K. Walsh, “Detection and counting of root-knot nematodes using YOLO models with mosaic augmentation,” *Biosens Bioelectron X*, vol. 15, Dec. 2023, doi: 10.1016/j.biosx.2023.100407.

[11] Z. Dai, “Image acquisition technology for unmanned aerial vehicles based on YOLO - Illustrated by the case of wind turbine blade inspection,” *Systems and Soft Computing*, vol. 6, Dec. 2024, doi: 10.1016/j.sasc.2024.200126.

[12] L. Chen, G. Li, S. Zhang, W. Mao, and M. Zhang, “Journal Pre-proof YOLO-SAG: An improved wildlife object detection algorithm based on YOLOv8n YOLO-SAG: An Improved Wildlife Object Detection Algorithm Based on YOLOv8n,” 2024, doi: 10.1016/j.ecoinf.2024.102791.

[13] Y. Liu *et al.*, “Detection method of the seat belt for workers at height based on UAV image and YOLO algorithm,” *Array*, vol. 22, Jul. 2024, doi: 10.1016/j.array.2024.100340.

[14] A. Baccouche, B. Garcia-Zapirain, Y. Zheng, and A. S. Elmaghraby, “Early detection and classification of abnormality in prior mammograms using image-to-image translation and YOLO techniques,” *Comput Methods Programs Biomed*, vol. 221, Jun. 2022, doi: 10.1016/j.cmpb.2022.106884.

[15] Z. Liu, Y. Li, F. Shuang, Z. Huang, and R. Wang, “EMB-YOLO: Dataset, method and benchmark for electric meter box defect detection,” *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 2, Feb. 2024, doi: 10.1016/j.jksuci.2024.101936.

[16] N. Aishwarya, K. Manoj Prabhakaran, F. T. Debebe, M. S. S. A. Reddy, and P. Pranavee, “Skin Cancer diagnosis with Yolo Deep Neural Network,” in *Procedia Computer Science*, Elsevier B.V., 2023, pp. 651–658. doi: 10.1016/j.procs.2023.03.083.

[17] J. Huang, K. Zeng, Z. Zhang, and W. Zhong, “Solar panel defect detection design based on YOLO v5 algorithm,” *Heliyon*, vol. 9, no. 8, Aug. 2023, doi: 10.1016/j.heliyon.2023.e18826.

[18] Z. Situ *et al.*, “A transfer learning-based YOLO network for sewer defect detection in comparison to classic object detection methods,” *Developments in the Built Environment*, vol. 15, Oct. 2023, doi: 10.1016/j.dibe.2023.100191.

[19] E. C. Tetila *et al.*, “YOLO performance analysis for real-time detection of soybean pests,” *Smart Agricultural Technology*, vol. 7, Mar. 2024, doi: 10.1016/j.atech.2024.100405.

[20] Z. Zhao *et al.*, “YOLO-PAI: Real-time handheld call behavior detection algorithm and embedded application,” *Signal Process Image Commun*, vol. 120, Jan. 2024, doi: 10.1016/j.image.2023.117053.

[21] M. Hussain, “YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection,” Jul. 01, 2023, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/machines11070677.

BIOGRAPHIES OF AUTHORS

Waluyo Nugroho, lecturer in mechatronics at Astra Polytechnic (ASTRAtech). He is graduated of Diponegoro University Semarang (Master Degree) in the Electrical Engineering study program. Before his career as a lecturer, he worked for 4 years in the industry. Once works as a kaizen at PT. Federal Nittan Industries (Astra Group), PT. Indonesia Epson Industry, as well as hardware engineer supervisor at LG Research and Development Center. He worked at ASTRAtech starting in 2023 by teaching Digital Engineering courses, Advanced Microcontrollers, and Internet of Things (IoT). Fields of research that were developed at the time these are Robotics, Instrumentation Electronics, IoT, and Artificial Intelligence. Apart from that, he actively providing electronics and IoT training in several industrial companies.

Afianto, lecturer at Astra Polytechnic (ASTRAtech) in the field of Mechatronics, who completed his Bachelor's degree at UI and Master's degree at Artois University, France. His career began in 1997 in the engineering department at Federal Motor, now known as Astra Honda Motor, and since 2000 he has been assigned to be a full lecturer at ASTRAtech. Lectures taught in the field of electronics are Electrical Control, Sensors & Actuators and Microcontrollers, while The manufacturing sector has covered the fields of Plastic Injection and Automotive Painting. The superior product developed in his science is the Trackless AGV using a Local Positioning System (LPS) guide. Several scientific works have been published in the form of journals, monographs and in international proceedings, all of which are related to mechatronics science.

Mada Jimmy Fonda Arifianto is a lecturer in the Mechatronics Study Program at Astra Polytechnic. He has an educational background in Mechanical Engineering at ATMI Surakarta (Diploma Degree), Industrial Engineering at Bina Nusantara University (Bachelor Degree) and Mechanical Engineering in-depth study in Mechatronics at University of Duisburg-Essen (Master Degree). Apart from providing lectures in the field of robotics, microcontrollers, computer programming and industrial automation for more than 20 years, Mada Jimmy also active in research in the fields of CNC machine development, Trackless AGV, IoT-based energy monitoring and control, remote sensor application and renewable energy. These researches received funding from the Indonesian Government, the German Government and internally from Astra Polytechnic. He also provides consultation and training in the field of digitalization and automation for the community and manufacturing industry, and is active in community service activities, especially for teachers and vocational community students. Apart from conveying Tri Dharma activities, this lecturer is also active in developing competencies in the field of artificial intelligence in the international collaboration program between Astra Polytechnic and the National Institute of Technology (KOSEN), Kumamoto College.