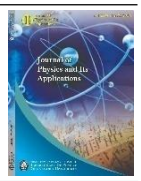
Journal of Physics and Its Applications

Journal homepage : https://ejournal2.undip.ac.id/index.php/jpa/index



Classification of Box Sizes in Automated Systems Using YOLOv8 and Computer Vision

Ayu Sarah Annisa and Meqorry Yusfi*

Department of Physics, Faculty of Mathematics and Natural Sciences, Andalas University, Padang, Indonesia

**Corresponding author: meqorryyusfi@sci.unand.ac.id*

A R T I C L E I N F O

Article history:
Received: 15 October 2025
Accepted: 22 May 2026
Available online: 30 May 2026

Keywords:
YOLO
MQTT
ESP32-CAM
Computer Vision
Automatic Sorting

A B S T R A C T

The logistics and distribution industry requires a fast and accurate automated sorting system to improve operational efficiency. This research develops a computer vision-based automated sorting system using YOLO (You Only Look Once) to detect and classify box sizes in real time. The system consists of an ESP32-CAM as a visual sensor, an ESP8266 NodeMCU as a microcontroller, and a servo motor as an actuator using the MQTT communication protocol. The detection results are sent through a local MQTT broker for low latency processing without the internet. The YOLOv8 model used successfully achieved a detection and classification accuracy of 98.15%. The top camera showed more stable performance (89-96%) than the front camera (83-96%) due to the influence of distance and angle of image capture. The tests were conducted under fixed lighting conditions and only distinguished between small (< 5×5×5 cm) and large (≥ 5×5×5 cm) boxes, with a maximum load limit of 700 grams. The system is still limited in classifying objects close to the size limit, and is not optimal for variable lighting.

1. Introduction

In the modern logistics age, the need for an automated sortation system is becoming increasingly important as the volume of goods to be handled on a daily basis increases [1]. Sorting systems are used to group goods according to certain characteristics, such as size, to improve shipping efficiency and storage space utilisation. However, many industries still use manual methods that are slow, inaccurate and dependent on human labour, leading to inefficiencies [2], [3], [4].

The application of computer vision and object recognition algorithms, such as YOLO, provides an effective solution for automated sorting systems. YOLO is known for its ability to detect objects in real time with high accuracy and good computational efficiency [5], [6]. Previous studies have shown the success of YOLO in various applications such as face recognition, vehicle detection, and industrial object classification [7], but its application for size classification in automated sorting systems is still limited.

This research aims to design and implement an automatic sorting system based on YOLO and ESP32-CAM, which is able to classify boxes based on size (large ≥ 5×5×5 cm and small < 5×5×5 cm). This system is integrated with NodeMCU ESP8266 and servo motor and uses MQTT as communication protocol. It is expected that this system will be able to increase the efficiency of the sorting process and reduce the classification errors.

2. Method

This study designed an automatic sorting system to classify box sizes using the YOLOv8 algorithm integrated with IoT hardware. The system consists of NodeMCU ESP8266 as the main controller, ESP32-CAM as the image sensor, DC motors as conveyor drives, and servo motors as sorting actuators.

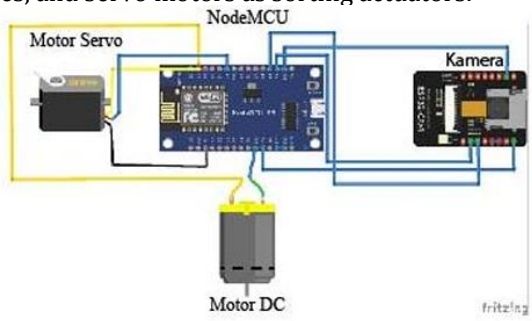


Fig. 1: Circuit design

Based on Fig. 2, conveyor belt (50 × 10 cm) is used to move the test objects. The camera is placed in two positions, namely the top camera at a distance of 65 cm to capture the length and width of the object, and the front camera at a distance of 30 cm to capture the height of the object.

DC motors are chosen because they have linear characteristics between input voltage and rotational speed. This principle is explained by the electromagnetic force equation (1):

$$F = I\,B\,L\cos\alpha \qquad (1)$$

where the current I is proportional to the voltage V supplied [8].Thus, the conveyor speed can be controlled by varying the voltage. A servo motor is used as a sorting actuator, controlled by a PWM signal generated by NodeMCU. Servo torque is formulated as equation (2):

$$\tau = F.r = mgr \qquad (2)$$

so that servo performance can be analyzed based on the load applied [9]. The ESP32-CAM camera operated at a standard VGA resolution of 640 x 480 pixels running under controlled indoor lighting conditions it has an ESP32-S processor with visual processing capabilities and integrated Wi-Fi connectivity [10], [11]. The classification data is transmitted from the ESP32-CAM to the NodeMCU using the MQTT protocol. MQTT was chosen because it is a lightweight, efficient publish–subscribe communication protocol that supports real-communication[2],[12],[13].

Prior to image acquisition and dataset annotation, the spatial boundaries of all physical samples were verified using a standardized precision measuring ruler. The baseline dataset compiled for this study comprises a total of 106 high-resolution images, structurally divided into 43 large box samples (≥ 5 × 5 × 5 cm) and 63 small box samples (< 5 × 5 × 5 cm). To ensure rigorous training, the dataset was partitioned using an 80:20 ratio, allocating 85 images for training and 21 images for validation. To mitigate overfitting risks associated with this compact baseline dataset, online data augmentation was implemented via Roboflow [14].

The YOLOv8 model was chosen because it has real-time detection performance with high accuracy [15]. Training was conducted in Google Colab with batch size parameters of 16, a learning rate of 0.001, and a number of epochs of up to 100. Model performance was evaluated using precision, recall, and mAP metrics, as widely used in the evaluation of classification and object detection systems [16], [17], [18].

This servo failure indicates that the actuator's performance is at its theoretical limit. Additionally, the increase in current to 73.3 mA under heavy load shows that the servo attempts to maintain torque by increasing electrical power consumption. This is consistent with the servo characteristics in the literature by [9], which states that the servo will lose precision if the load exceeds its torque capacity.

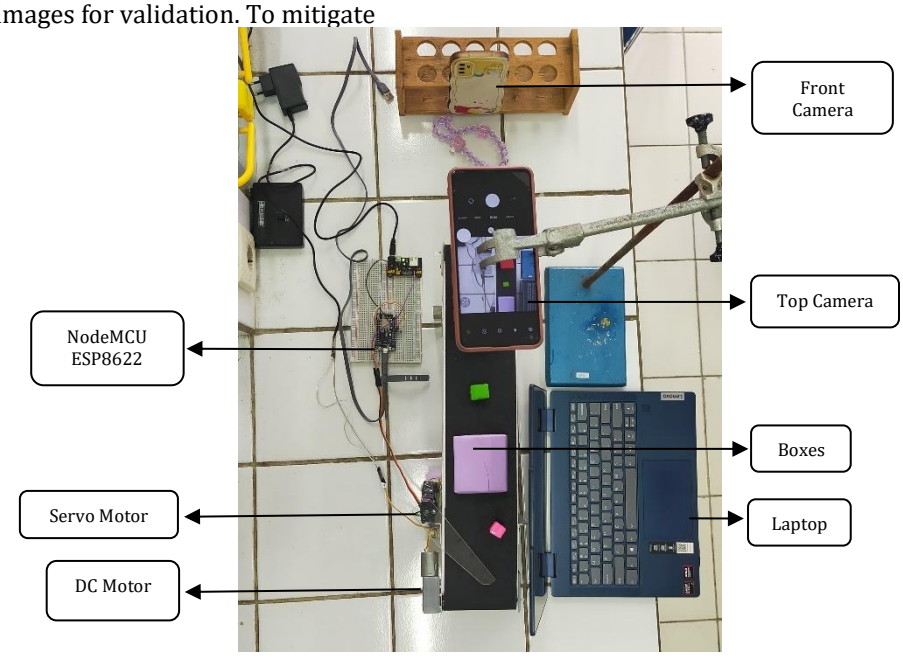


Fig. 1: The physical form of the tool design

Table 1. Camera testing

No.	Camera	Object Detected	Preprocess Time (ms)	Inference Time (ms)	Postprocess Time (ms)	Frame Rate (FPS)	Data Rate (Mbps)
1	Top	big box	1.0	29.6	1.0	21.65	1077.62
		small box	2.0	27.6	1.0	21.00	1045.31
		no detection	2.5	28.6	0.0	23.18	1153.76
2	Front	big box	1.0	28.4	0.0	23.75	1181.82
		small box	2.0	26.1	1.0	18.46	918.75
		no detection	1.5	29.6	0.0	24.95	1241.57

3. Results and Discussion

3.1 Camera Testing

This test was conducted to evaluate the performance of the two cameras, namely the top and front cameras, in detecting objects by measuring the processing time, frame rate, and data rate generated according to the Table 1.

The difference in accuracy between the rear camera and front camera shows that the angle of image capture affects the stability of object detection. The top camera produces a more stable confidence level (92–96%) than the front camera (83–95%). This can be analyzed using the concept of camera field of view projection, where the camera position affects the capture area and perspective distortion [19].

To ensure system reliability, the visual data acquisition is modeled mathematically. The image quality metric Q captured by the ESP32-CAM is governed by Equation (3):

$$Q = k\frac{1}{d \cdot \cos\theta} \quad (3)$$

Where Q is the image quality acquisition index, k is the empirical ambient illumination calibration constant, d is the orthogonal distance between the lens plane and the conveyor surface (cm), and θ is the projection angle relative to the surface normal vector. These results reinforce the findings of [20] The camera position determines the accuracy of computer vision-based classification systems.

3.2 Servo and Effective Load Testing

The servo motor was tested at angles of 0° and 45°. Effective load tested on conveyor systems and servo motors to ensure optimal performance and prevent damage due to overload can be seen in the Table 2.

The results demonstrate that the sorting servo actuator can consistently maintain its angular position under payload conditions of ≤ 700 g, but exhibits structural routing failure when subjected to a heavy load of 1 kg. This dynamic phenomenon can be analyzed systematically using the mechanical torque formulation in Equation (2). Given an experimental mass of m=1 kg, gravitational acceleration of g = 9.81 m/s^2, and an effective actuator torque arm radius of r = 0.03 m, the calculated mechanical torque demand equals τ=0.294 Nm. While this demand remains lower than the manufacturer-specified absolute stall torque of the authentic MG996R metal-gear servo (τ=0.94 N.m at 4.8V), it significantly exceeds the continuous operational dynamic torque threshold under the limited current output of the microcontroller environment. Consequently, this threshold breach triggers a severe actuator failure, driving the servo current into a complete overload state (OL). This empirical loss of sorting precision under excessive rotational resistance aligns perfectly with the electric actuator profiles documented by Rahman et al. (2020), which confirm that servo systems experience immediate spatial degradation when continuous operational load overhead compromises internal gear tracking loops.

Table 2. Servo and effective load testing

No	Weight (kg)	t_{conv} (s)	Speed of Conveyor (m/s)	Servo Current (mA)	Torque (N. m)	Operating Status
1	0.1	4.46	0.112107623	12.72	0.058413676	Normal
2	0.2	4.72	0.105932203	14.07	0.116827352	Normal
3	0.3	5.04	0.099206349	21.12	0.175241028	Normal
4	0.4	5.31	0.094161959	39.70	0.233654704	Normal
5	0.5	5.49	0.091074681	40.02	0.29206838	Normal
6	0.6	6.11	0.081833061	43.04	0.350482056	Normal
7	0.7	6.74	0.074183976	47.70	0.408895732	Normal
8	0.8	7.27	0.068775791	62.40	0.467309408	Suboptimal
9	0.9	7.72	0.064766839	73.30	0.525723084	Suboptimal
10	1	8.97	0.05574136	OL	0.584136761	Overload

Note: OL indicates an Overload operating state where the structural payload exceeds the torque limitations of the conveyor motor

3.3 DC Motor Testing

The DC motor was tested at voltages of 1-5 V This can be seen in Table 3. The results showed a linear relationship between voltage and conveyor speed. At 1 V, a speed of 7.12 cm/s was obtained, while at 5 V, it increased to 26.59 cm/s.

The physical translation of input voltage (V) to conveyor linear velocity (v) depends on the electromechanical coupling of the DC motor. The torque (τ) generated by the electromagnetic Lorentz force is formulated in Equation (4):

$$\tau = I.B.L.r_p.\cos\alpha \quad (4)$$

where I is the motor current, B is the magnetic flux density, L is the coil length, r_p is the pulley radius, and α is the angular displacement. Given that current is defined by Equation (5):

$$I = \frac{V - E_b}{R} \quad (5)$$

where E_b is the back-electromotive force (back-EMF) proportional to angular velocity (ω) and R is the internal armature resistance, the mechanical output power equals the electrical power ($P = VI = Fv$) neglecting frictional overhead. This model yields a direct linear relationship where an increase in operating voltage proportionally accelerates the conveyor belt velocity.

Table 3. DC motor testing

No	Voltage (V)	Object Time (s)	Distance (cm)	Velocity (cm/s)
1	1	7.02	50	7.12
2	1.5	6.25	50	8.00
3	2	5.88	50	8.50
4	2.5	5.40	50	9.25
5	3	4.70	50	10.63
6	3.5	3.67	50	13.62
7	4	3.20	50	15.62
8	4.5	2.80	50	17.85
9	5	1.88	50	26.59

The speed increases linearly with voltage. The voltage-speed relationship graph shows $R^2 >$ 0.95 confirming consistency with electromagnetic

theory [8]. A small deviation from linearity occurs due to mechanical losses (belt friction, bearing friction), in accordance with the results of Wardoyo et al. (2016).

3.4 MQTT Communication Testing

The communication test results show that classification data was successfully transmitted from ESP32-CAM to NodeMCU via the MQTT broker without packet loss and with low latency. Theoretically, the advantage of MQTT over other protocols is its efficiency and low overhead because it uses a publish–subscribe mechanism [12].

These findings are consistent with the research by Kashyap et al. (2018) and Liu et al. (2020), who reported that MQTT is capable of maintaining real-time communication stability in IoT systems, even under limited network conditions.

The software network metrics exhibited high performance during the deployment trials. The MQTT broker maintained a quantitative mean transmission latency of 42.5 ms with zero packet loss under local Wi-Fi conditions. The total end-to-end latency from structural image acquisition to mechanical servo actuation averaged approximately 185 ms, validating the real-time operational claims of the system.

The computational parameters have been updated in the experimental setup section. The YOLOv8 model optimization was trained locally using an AMD Ryzen 7 8845HS processor equipped with integrated Radeon 780M Graphics and 16.0 GB of system RAM on a 64-bit architecture. The training execution for 120 epochs was completed in approximately 42 minutes.

The comparison of epoch values is derived from the number of epochs to be tested, ranging from 40 to 130 epochs, in conjunction with accuracy, precision (P), recall (R), and mean average precision (mAP) values. This can be observed in Table 4.

The YOLOv8 model showed the best performance at epoch 80 with a precision of 0.992, recall of 0.999, and mAP50-95 of 0.886. After epoch 100, performance declined due to overfitting. Precision and recall are calculated as follows:

$$\text{Precision} = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN} \quad (6)$$

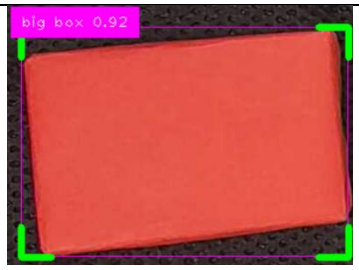
With the test data, TP = 54,FP = 1,FN = 0 → precision = 0.982 and recall = 1.0. These values are close to the training results, proving that the model is capable of generalization. The confusion matrix shows only two classification errors out of 106 test samples.

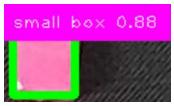
These results support the research of Ali & Zhang (2024) and Cheng et al. (2024), who state that YOLOv8 excels in real-time object detection but is sensitive to overfitting on limited datasets. Thus, the YOLOv8 model used in this study can be considered valid for integration into an automatic sorting system.

Table 4. Epoch value comparison

No	Epoch	Accuracy	Class	Precision	Recall	mAP50	mAP50-95
1	40	95.50%	All	0.985	0.979	0.989	0.872
			Big Box	1.000	0.957	0.994	0.911
			Small Box	0.969	1.000	0.983	0.832
2	50	97.25%	All	0.988	0.999	0.995	0.874
			Big Box	1.000	0.998	0.995	0.905
			Small Box	0.976	1.000	0.995	0.843
3	60	100%	All	0.998	1.000	0.995	0.885
			Big Box	0.997	1.000	0.995	0.927
			Small Box	1.000	1.000	0.995	0.843
4	70	97.25%	All	0.996	0.990	0.995	0.888
			Big Box	1.000	0.979	0.995	0.926
			Small Box	0.991	1.000	0.995	0.851
5	80	98.15%	All	0.992	0.999	0.995	0.886
			Big Box	1.000	0.998	0.995	0.922
			Small Box	0.985	1.000	0.995	0.850
6	90	94.64%	All	0.992	0.967	0.994	0.896
			Big Box	1.000	0.937	0.994	0.932
			Small Box	0.984	0.998	0.993	0.861
7	100	95%	All	0.974	0.999	0.995	0.893
			Big Box	1.000	0.999	0.995	0.945
			Small Box	0.948	1.000	0.994	0.842
8	110	97.22%	All	0.977	0.994	0.992	0.891
			Big Box	0.988	0.995	0.995	0.942
			Small Box	0.955	1.000	0.989	0.839
9	120	98.15%	All	0.998	1.000	0.995	0.901
			Big Box	0.999	1.000	0.995	0.941
			Small Box	0.996	1.000	0.995	0.862
10	130	93.64%	All	0.974	0.990	0.992	0.899
			Big Box	1.000	0.980	0.995	0.942
			Small Box	0.947	1.000	0.989	0.855

Table 5. System result

No.	Top Camera	Front Camera	Confi-dent Level	Servo Respon	Result
1.			Top: 92% Front: 85%	Rotate 0°	Works
2.			Top: 94% Front: 96%	Rotate 45°	Works
3.			Top: 91% Front: 85%	Rotate 0°	Works
4.			Top: 90% Front: 84%	Rotate 45°	Works
5.			Top: 92% Front: 92%	Rotate 0°	Works
6.			Top: 95% Front: 86%	Rotate 45°	Works
7.			Top: 90% Front: 85%	Rotate 45°	Works

8.			Top: 94% Front: 89%	Rotate 0°	Works
9.			Top: 96% Front: 83%	Rotate 0°	Works
10.			Top: 89% Front: 88%	Rotate 45°	Works

3.6 System-wide Result

The following table 5 is the result of system testing and the data collection process to evaluate the performance of the system that has been tested.

Based on Table 5, out of 108 box samples tested, the system successfully classified 106 boxes correctly and misclassified 2 boxes, resulting in an accuracy rate of 98.15%. This value indicates that the system works with high reliability, with an error rate of only 1.9% of the total test.

Classification errors within the system can be critically analyzed as a direct consequence of unstable ambient lighting conditions and non-ideal object positioning on the running conveyor belt. In specific instances, physical boxes were either not fully captured within the camera's field of view or were partially obscured by transient shadows. Consequently, the localized image features processed by the YOLOv8 model failed to perfectly match the pre-trained structural patterns. This phenomenon aligns with the fundamental characteristics of deep learning-based object detection frameworks, which exhibit acute sensitivity to input image variations and illumination shifts [20]. In highly dynamic industrial environments, intense solar glare or deep shadows could further degrade YOLOv8 bounding-box regression due to local pixel saturation. To combat this limitation in real-world factory deployment, the automated system must be retrofitted with a localized LED light shroud to maintain a constant, controlled lux profile independent of unpredictable factory floor lighting shifts.

When evaluated against prior literature, the classification reliability achieved by this architecture demonstrates a superior performance threshold. Wardoyo et al. (2016) reported an operational sorting accuracy of approximately 91% using a basic digital image processing algorithm, which suffered from rigid constraints under varying light intensities [21]. Similarly, Pratama (2022) developed a light-dependent sensor-based sorting mechanism achieving 95% accuracy; however, that system exhibited low structural flexibility when adapting to dynamic variations in object size and shape complexity. In contrast, the results of this study—sustaining a synchronized system accuracy of 98.15%—confirm that integrating the lightweight YOLOv8 nano algorithm with a distributed edge-IoT topology can significantly improve automated system benchmarks.

Thus, the overall integrated system developed in this work is not only theoretically valid but also structurally superior to conventional low-cost automation approaches. The high classification accuracy and rapid end-to-end response runtime (<1 second) verify that this architecture is highly suitable for localized small-scale fulfillment applications. To completely eliminate the two classification errors observed during operational testing, future iterations will focus on expanding the training dataset with diverse environmental artifacts and engineering the aforementioned physical lighting enclosures to ensure the edge deployment remains highly adaptive to ambient industrial changes.

4. Conclusion

This study successfully demonstrated the electromechanical and computational viability of an automated sorting system integrated with IoT communication and computer vision frameworks. The empirical results verified that spatial topology significantly dictates detection reliability; a top-down camera perspective yielded superior bounding-box stability and higher confidence scores than a frontal orientation by minimizing perspective distortion. Physically, the conveyor mechanics adhere strictly to electromagnetic coupling principles, exhibiting a robust linear relationship between the applied DC operational voltage and the linear velocity of the belt. However, the system operates under a clear mechanical threshold, with the conveyor capacity constrained to a maximum payload of 700 g due to electromechanical friction overhead, while the MG996R servo motor functions within theoretical torque boundaries but compromises routing precision if overloaded.

On the computational layer, data propagation via the asynchronous MQTT protocol sustained a stable,

real-time edge architecture with zero packet loss. The optimized YOLOv8 nano deployment achieved definitive convergence at the 80th epoch, yielding a precision of 0.992, a recall of 0.999, and a mAP50–95 of 0.886. The core scientific contribution of this research lies in validating a highly cost-efficient, decoupled multi-microcontroller framework capable of matching heavy industrial sorting accuracy at a fraction of standard hardware costs, making it immediately applicable for localized small-and-medium enterprise (SME) logistics. To scale this technology for demanding industrial environments, future advancements must focus on upgrading the actuator torque, expanding the physical conveyor frame payload capacity, and integrating multi-camera sensor fusion to eliminate blind spots under uncalibrated illumination fields.

Acknowledgment

The author would like to thank the supervisor, the Physics Laboratory of Andalas University, and all those who have provided assistance, support and facilities during the implementation of this research.

References

[1] R. Krishnan, E. Perumal, M. Govindaraj, and L. Kandasamy, "Enhancing logistics operations through technological advancements for superior service efficiency," in *Innovative Technologies for Increasing Service Productivity*, IGI Global, 2024, pp. 61–82. doi: 10.4018/979-8-3693-2019-8.ch004.

[2] Z. B. Abilovani, W. Yahya, and F. A. Bakhtiar, "Implementasi Protokol MQTT Untuk Sistem Monitoring Perangkat IoT," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 12, pp. 7521–7527, 2018, [Online]. Available: http://j-ptiik.ub.ac.id

[3] D. S. Pratama, "Rancang Bangun Conveyor Penyortir Mur Berbasis Raspberry Pi Menggunakan Metode Contour," *J. Tek. Elektro*, vol. 11 Nomor 02, pp. 246–254, 2022.

[4] M. L. Ali and Z. Zhang, "The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection," *Computers*, vol. 13, no. 12, 2024, doi: 10.3390/computers13120336.

[5] Amna *et al.*, "Machine vision-based automatic fruit quality detection and grading," *Front. Agric. Sci. Eng.*, vol. 0, no. 0, p. 0, 2023, doi: 10.15302/j-fase-2023532.

[6] Z. Duan, W. Liu, S. Zeng, C. Zhu, L. Chen, and W. Cui, "System for Fresh-Cut Flowers," 2024.

[7] V. N. Pham, Q. H. Do Ba, D. A. Tran Le, Q. M. Nguyen, D. Do Van, and L. Nguyen, "A Low-Cost Deep-Learning-Based System for Grading Cashew Nuts," *Computers*, vol. 13, no. 3, pp. 1–22, 2024, doi: 10.3390/computers13030071.

[8] L. Petru and G. Mazen, "PWM control of a DC motor used to drive a conveyor belt," in *Procedia Engineering*, Elsevier Ltd, 2015, pp. 299–304. doi: 10.1016/j.proeng.2015.01.371.

[9] F. Rahman, F. Faridah, A. I. Nur, and A. N. Makkaraka, "Rancang Bangun Prototipe Manipulator Lengan Robot Menggunakan Motor Servo Berbasis Mikrokontroler," *ILTEK J. Teknol.*, vol. 15, no. 01, pp. 42–46, 2020, doi: 10.47398/iltek.v15i01.11.

[10] S. Mahmood, S. Alani, F. Hasan, and M. Mustafa, "ESP 8266 Node MCU Based Weather Monitoring System," European Alliance for Innovation n.o., Sep. 2020. doi: 10.4108/eai.28-6-2020.2298609.

[11] M. Wijayanti, "PROTOTYPE SMART HOME DENGAN NODEMCU ESP8266 BERBASIS IOT," *J. Ilm. Tek.*, vol. 1, no. 2, pp. 101–107, 2022.

[12] R. A. Light, "Mosquitto: server and client implementation of the MQTT protocol," *J. Open Source Softw.*, vol. 2, no. 13, p. 265, 2017, doi: 10.21105/joss.00265.

[13] X. Liu, T. Zhang, N. Hu, P. Zhang, and Y. Zhang, "The method of Internet of Things access and network communication based on MQTT," *Comput. Commun.*, vol. 153, no. December 2019, pp. 169–176, 2020, doi: 10.1016/j.comcom.2020.01.044.

[14] S. Van Der Walt *et al.*, "Scikit-image: Image processing in python," *PeerJ*, vol. 2014, no. 1, 2014, doi: 10.7717/peerj.453.

[15] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "YOLO-World: Real-Time Open-Vocabulary Object Detection," pp. 16901–16911, 2024, [Online]. Available: http://arxiv.org/abs/2401.17270

[16] M. Glučina, N. Anđelić, I. Lorencin, and Z. Car, "Detection and Classification of Printed Circuit Boards Using YOLO Algorithm," *Electron.*, vol. 12, no. 3, 2023, doi: 10.3390/electronics12030667.

[17] M. Anwar, Y. Kristian, and E. Setyati, "Classification Of Chili Plant Diseases Equipped With Leaf and Fruit Image Segmentation Using Yolo V7," *J. Inf. Technol. Comput. Sci.*, vol. 6, no. 1, 2023.

[18] B. R. Prabhu, A., K, A. K., Abhiram, A., & Pushpa, "Mango Fruit Classification using Computer Vision System," *4th Int. Conf. Inven. Res. Comput. Appl.*, pp. 1797–1802, 2022, doi: https://doi.org/10.1109/ICIRCA54612.2022.9985773.

[19] A. Siregar, B., Pradaning, R., & Hizriadi, "Cocoa Ripeness Level Sorting System Using Integrated Computer Vision Technology On Conveyor Belt," *8th Int. Conf. Electr. Electron. Inf. Eng.*, pp. 1–6, 2023, doi: https://doi.org/10.1109/ICEEIE59078.2023.10334634.

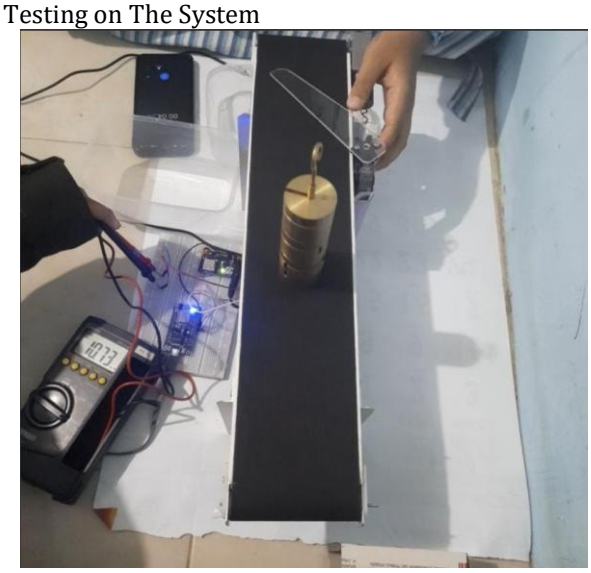
[20] I. Aryeni, H. M. Maulidiah, H. Toar, M. J. Wimbang, and I. Gunawan, "Application of Computer Vision for Real-Time Detection of Fruit Color and Size in Fruit Sorter," *J. Appl. Electr. Eng.*, vol. 7, 2023.

[21] S. Wardoyo, J. Saepul, and A. S. P. Suryo Pramudyo, "Rancang Bangun Alat Uji Karakteristik Motor DC Servo, Battery, dan Regulator untuk Aplikasi Robot Berkaki," *Setrum Sist. Kendali-Tenaga-elektronika-telekomunikasi-komputer*, vol. 2, no. 2, p. 111, 2016, doi: 10.36055/setrum.v2i2.490.

Appendix

Appendix 1. Overall Program and Box Dataset https://github.com/Stovia27/YOLOv8-Box-Classification-System.git

Appendix 2. Documentation: Maximum Load Limit Testing on The System



Appendix 3. Documentation: Training the Model