

Integrating YOLOv11 with Robotic Arms for Real-Time Shape Detection and Classification

Samuel Owoeye¹, Folasade Durodola¹, Micheal Adedokun¹, Mislal Erinkitola¹, Taiwo Akinsanya¹

¹Department of Mechatronics Engineering, Federal University of Agriculture, Abeokuta, Nigeria.
owoeyeso@funaab.edu.ng, durodolafo@funaab.edu.ng, michdokun@gmail.com,
mislaloluwatimilehin@gmail.com, akinsanyataiwo@gmail.com
Corresponding Author: owoeyeso@funaab.edu.ng

Received 10 February 2026; revised 15 April 2026; accepted 1 June 2026

Abstract

Traditional object detection models often struggle with challenges such as variable lighting, occlusions, and the presence of multiple overlapping objects, leading to inaccuracies in shape classification. This study investigates the integration of YOLOv11 with robotic arms for real-time shape classification, focusing on enhancing the operational capabilities of autonomous systems in dynamic environments. Utilising a dataset of 4,630 annotated images representing circles, squares, and triangles, the research evaluates YOLOv11's effectiveness in accurately detecting and classifying these shapes. The model achieved a mean Average Precision (mAP) of 0.99 at an Intersection over Union (IoU) threshold of 0.50 and 0.93 at IoU 0.50–0.95, demonstrating exceptional classification accuracy. Precision and recall rates were recorded at 0.97 and 0.95, respectively, indicating robust performance in object detection tasks. The integration process involved deploying the YOLOv11 model on a 3R robotic arm, enabling real-time processing of video feeds for shape detection and sorting. Detection times averaged around 0.8 seconds, while sorting times ranged from 2.6 to 5.1 seconds, showcasing the system's efficiency in operational settings. The study highlights the critical relationship between confidence thresholds and prediction accuracy, emphasising the need for careful calibration to mitigate misclassifications. Overall, the findings affirm YOLOv11's potential as a powerful tool for real-time object classification in robotics, suggesting avenues for future research to optimise performance and expand application domains in autonomous systems.

Keywords: Autonomous, Confidence threshold, Dataset, Precision, Recall, Sorting

1. Introduction

The field of autonomous robotics has witnessed remarkable progress, driven by the increasing demand for intelligent systems capable of operating in complex and dynamic environments. A fundamental requirement for such systems is the ability to accurately and efficiently perceive their surroundings, with real-time object detection and classification playing a pivotal role. This capability enables robots to perform tasks such as navigation, manipulation, and interaction with humans and other objects. This research delves into the application of YOLOv11, a state-of-the-art object detection model, for real-time shape classification in autonomous robotics, focusing on its performance and adaptability in dynamic environments. YOLOv11, the latest iteration in the You Only Look Once (YOLO) series, represents a significant advancement in object detection technology (He *et al.*, 2025). Building upon the successes of its predecessors, YOLOv11 introduces architectural refinements and training methodologies that enhance its accuracy, speed, and efficiency. The model's architecture is designed to optimise both speed and accuracy, building on advancements introduced in earlier YOLO versions (Hussain, 2023). Key innovations include the C3K2

block, SPFF (Spatial Pyramid Pooling Fast) module, and C2PSA (Cross Stage Partial with Spatial Attention) block, all of which contribute to improved spatial information processing while maintaining high-speed inference. YOLOv11's ability to perform object detection in a single pass makes it particularly well-suited for real-time applications in robotics (Simeonov *et al.*, 2025).

YOLOv11's architecture comprises three main sections: the backbone, neck, and head. The backbone extracts essential features from the input image, utilising advanced convolutional and bottleneck blocks (Li *et al.*, 2021). The neck combines features from different layers to improve the model's ability to detect objects at various scales. The head generates the final output, including bounding boxes, class labels, and confidence scores. YOLOv11 employs a modified CSP (Cross Stage Partial) architecture, enhancing feature learning while reducing computational overhead. The model also incorporates a loss function that combines classification, localisation, and confidence losses, optimised to improve detection accuracy while maintaining fast inference times (Su *et al.*, 2025). Accurate and real-time shape classification has broad implications for autonomous systems, enabling them to perform a wide range of tasks with greater efficiency and reliability.

A study by Khanam and Hussain in 2024 provides a detailed look into the architectural innovations of YOLOv11, highlighting its core components: the backbone, neck, and head. It emphasises the C3k2 block, which enhances computational efficiency, and the SPPF (Spatial Pyramid Pooling - Fast) module for robust feature representation. The study underscores YOLOv11's practical implications for computer vision tasks, including autonomous vehicles and surveillance systems. Also, Aravinda (2023) explored YOLOv11's capabilities in vehicle detection for intelligent transportation systems. It details the dataset used, comprising annotated images of various vehicle types under diverse conditions, and the data augmentation techniques employed to enhance model robustness. The study reveals that YOLOv11 outperforms previous versions in detection accuracy and real-time performance, making it suitable for autonomous driving and traffic monitoring applications.

Owoeye *et al.* (2024) introduced a weed detection system based on YOLOv5 for real-time classification and identification of weed species. The model achieved a mean Average Precision (mAP) of 91.4% and a detection latency of 87 milliseconds per frame. The trained model was integrated into a web-based tool to provide farmers with detailed information on weed species and removal techniques, supporting sustainable agricultural practices. Research by Badgujar *et al.* (2024) provides a broad overview of YOLO's applications in agriculture. It highlights YOLO's versatility in addressing complex challenges across fields such as medical imaging, autonomous vehicles, and agriculture. In each of these domains, YOLO's ability to process high-resolution images quickly and accurately has driven substantial improvements in detection efficiency and accuracy.

Deploying autonomous robots in dynamic environments presents unique challenges for object detection systems. These environments are characterised by moving objects, variable lighting conditions, occlusions, and complex backgrounds, which can significantly impact the performance of traditional object detection algorithms (Juel *et al.*, 2020). To address these challenges, robust object detection models must be capable of adapting to changing conditions and maintaining high accuracy and speed. This research aims to evaluate YOLOv11's ability to handle these dynamic factors and its potential for enhancing the performance of autonomous robots in real-world scenarios.

2. Methodology

This methodology outlines the steps for implementing YOLOv11 for real-time shape classification in an autonomous robotics context, specifically using a 3R robotic arm.

2.1 Data Collection

A comprehensive dataset was collected, containing 4630 images of the three shapes considered, with square images being the highest with 1620 images, circle with 1560 images and triangle with 1450, each image

was annotated with bounding boxes and class labels. The dataset was grouped into training (80%), testing (10%) and validation (10%).

2.2 Model Architecture

The YOLOv11 architecture, shown in Figure 1, was used, comprising a backbone for feature extraction, a neck for feature fusion, and a head for predictions (Xiao *et al.*, 2025). The model was initialised with pre-trained weights, and in order to optimise speed and reduce memory usage, the automatic mixed precision was enabled. The dataset was augmented using various augmentation techniques, including blurring, rotation, scaling, translation, flipping, median blurring and grayscale conversion to enhance model robustness. The training was conducted on Google Colab, utilising a Tesla T4 GPU for accelerated performance. The training process spanned 50 epochs to ensure optimal learning.

2.3 Model Evaluation

Upon finishing the training phase, the model was subjected to testing using the validation dataset to evaluate its performance. Essential metrics, including F1-score, recall, precision, and mean Average Precision (mAP), were calculated to assess the model's capability in distinguishing between ripe and unripe tomatoes. Confusion matrices were generated to depict the classification results, offering insights into the different errors made by the model.

2.4 Integration with the 3R Robotic Arm

The trained YOLOv11 model was integrated into the robotic arm's control system using the Raspberry Pi 5 microcontroller, shown in Figure 2. The model processes video feeds from a Raspberry Pi camera mounted on the arm in real-time, detecting and classifying shapes. Upon detecting a shape, the position data will be extracted, and the inverse kinematics calculations will be executed to compute the necessary joint angles. The robotic arm will then move to the target position, performing tasks of picking and placing the shapes in the appropriate positions.

2.5 Denavit-Hartenberg Parameters

Define the DH parameters for the 3R robotic arm. The parameters include:

- **Link Length (a):** Distance between the current joint and the next joint along the x-axis.
- **Link Twist (α):** Angle between the z-axes of consecutive links.
- **Link Offset (d):** Distance along the previous z-axis to the common normal.
- **Joint Angle (θ):** Angle about the previous z-axis from the old x-axis to the new x-axis.

The DH parameters were obtained as shown in Table 1. This was used to construct the transformation matrices for each joint, shown in equations 1-5

Table 1: DH Parameters

Joint	θ_i	d_i	a_i	α_i
Base(j_1)	θ_1	7.5cm	0	90°
Shoulder (j_2)	θ_2	0	11.7cm	0
Elbow (j_3)	θ_3	0	15.2cm	0
Wrist (j_4)	θ_4	0	7.5cm	0

Each link's transformation matrix T_i is given by:

$$T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i & 0 & a_i\sin\theta_i \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Base to Shoulder T_1

$$T_1^0 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 7.5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Shoulder to Elbow T_2

$$T_2^1 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & 11.7\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & 11.7\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Elbow to Gripper Base T_3

$$T_3^2 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & 15.2\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & 15.2\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Gripper Extension T_4

$$T_4^3 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & 7.5\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & 7.5\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

2.6 Block diagram

The block diagram, as shown in Figure 1, provides a high-level representation of the system architecture for the shape-sorting robotic arm with 4 Degrees of Freedom (DoF). It illustrates the interaction between the various components that make up the system. Each block in the diagram represents a key functional part of the project, and the connections between them show how signals, data, or power flow through the system. The block diagram illustrates a mechatronic system comprised of a Raspberry Pi camera, a Raspberry Pi 4, a servo motor, and a battery pack. The Raspberry Pi camera captures images, serving as the system's input device. These images are transmitted to the Raspberry Pi 4, which acts as the central processing unit, performing image processing to extract features or coordinates relevant to the task at hand. This processing is critical for determining how the servo motor should respond, enabling the system to interact with its environment effectively. Once the Raspberry Pi has calculated the necessary coordinates, it sends control signals to the servo motor, which executes the required movements, such as rotating or adjusting its position. The battery pack powers the entire setup, ensuring that the components function efficiently, particularly in mobile or remote applications. This configuration is versatile, making it applicable in fields like robotics, surveillance, and automated systems, where visual feedback and physical interaction are essential for operational success.

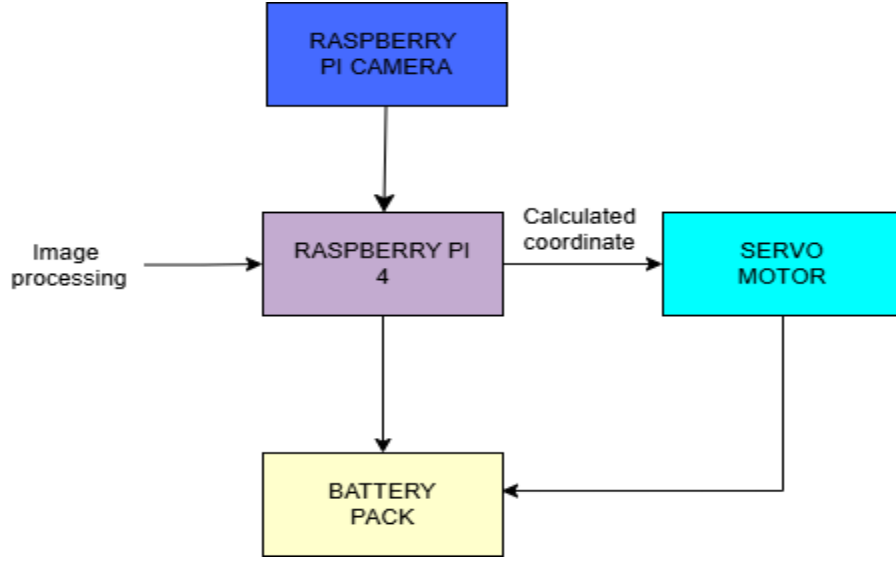


Fig. 1: Block diagram

3. Results and Discussion

3.1 Training and Validation

The training and validation performance curves for a YOLOv11-based model are shown in Figure 2, tracking different loss components and evaluation metrics across training epochs. In the training graphs, the box loss, classification loss, and distribution focal loss (dfl_loss) all show a clear decreasing trend as epochs progress, indicating that the model is progressively learning to localise objects more accurately, classify them correctly, and improve bounding box regression. Meanwhile, the precision and recall curves rise steadily, stabilising near 1.0 after about 30 epochs, suggesting that the model achieves both high correctness in predictions (precision) and high completeness in detecting objects (recall).

In the validation graphs, similar decreasing trends are observed for the box, classification, and dfl losses, demonstrating that the improvements generalise well to the validation dataset. Importantly, there are no signs of overfitting, as validation loss decreases consistently alongside training loss. The metrics curves for validation (mAP50 and mAP50–95) rise quickly within the first 10 epochs and plateau close to 0.95–1.0, reflecting excellent performance in object detection accuracy across different IoU thresholds. The trained YOLOv11 model demonstrated excellent results, with a precision of approximately 0.97, a recall of 0.95, mAP@50 of 0.99, and mAP@50–95 of 0.93 at convergence. These outcomes reflect a highly robust system, particularly the strong mAP@50–95 score, which indicates reliable detection performance across stricter IoU thresholds. When compared with existing literature, the results are competitive with state-of-the-art benchmarks. For instance, Reis *et al.* (2023) presented a refined YOLOv8 variant for flying object detection achieved mAP@50 of 99.1% and mAP@50–95 of 83.5%, though at real-time speeds of around 50 frames per second. Similarly, [8] in agricultural weed detection using YOLOv8m reported lower performance with a precision of 85%, a recall of 66%, mAP@50 of 77%, and mAP@50–95 of 44%. These comparisons suggest that the present model outperforms typical YOLOv8 deployments, particularly in terms of generalisation across varying IoU thresholds.

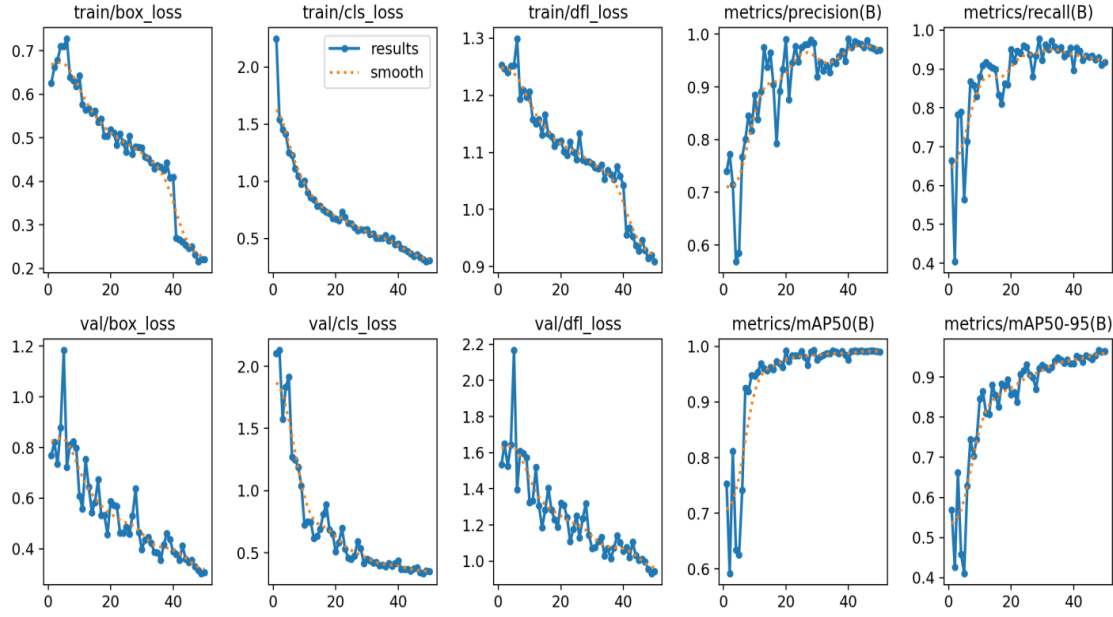


Fig. 2: Training and Validation Metrics

3.2 Precision Confidence Curve

Figure 3 presents a Precision-Confidence Curve that illustrates the relationship between precision and confidence levels for the three shapes. Notably, the circle class demonstrates the highest precision across a broad range of confidence levels, indicating that the model is highly reliable when identifying circles. In contrast, the square class shows moderate precision that increases with confidence but does not reach the levels seen with circles. The triangle class exhibits similar behaviour to squares, with overall lower precision, suggesting that the model struggles more with accurately classifying triangles compared to the other shapes. The model achieves perfect precision for all classes at a confidence threshold of 0.990. This implies that the model can reliably classify shapes when it is very confident in its predictions. This characteristic is especially important for real-world applications where high precision is necessary, such as in automated systems or safety-critical environments.

3.3 Recall-Confidence Curve

The Recall-Confidence Curve for three shape classes is shown in Figure 4. It was observed that the circle shape consistently maintains a high recall, indicating that the model effectively identifies nearly all actual circle instances across various confidence thresholds. Both the square and triangle classes exhibit lower recall values, especially at lower confidence levels. This suggests that the model is less effective at capturing true positives for these shapes, resulting in a higher likelihood of missing instances compared to circles.

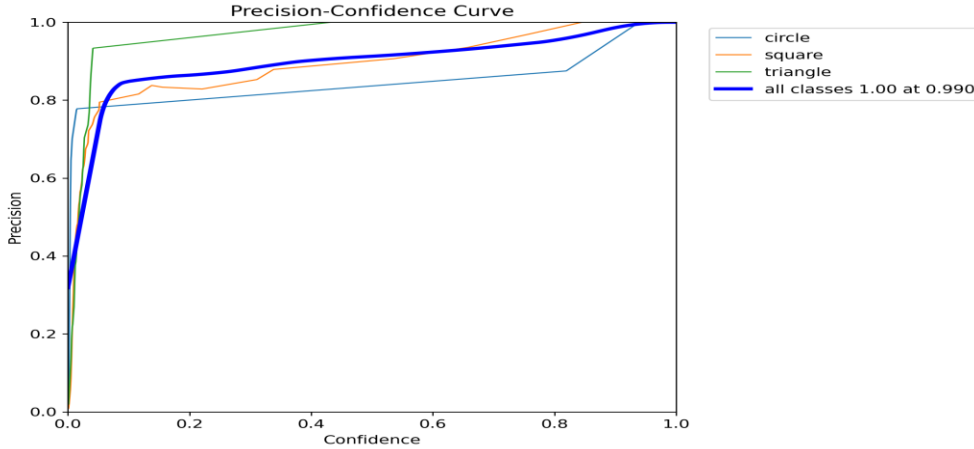


Fig. 3: Precision Confidence Curve

The model also achieves a perfect recall for all classes at a confidence threshold of 0.000. This implies that when the model is not selective about its predictions, it can identify all instances of each shape, but at the cost of precision. The sharp decline in recall for squares and triangles as confidence increases indicates that the model may be overfitting or misclassifying these shapes as it becomes more confident. Overall, this curve highlights the challenges faced in balancing recall and confidence in shape classification, particularly for shapes with less distinct features.

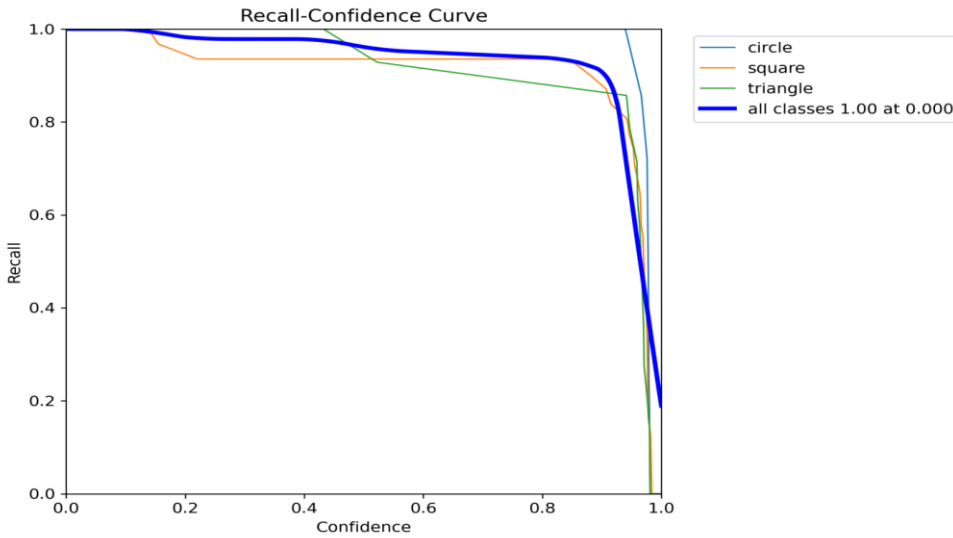


Fig. 4: Recall Confidence Curve

3.4 F1 Curve

The F1-Confidence Curve depicted in Figure 5 shows that the circle class exhibits the highest F1 score across a range of confidence levels, maintaining values close to 1.0. This indicates that the model performs exceptionally well in both precision and recall when identifying circles, while the square and triangle classes show lower F1 scores, particularly at lower confidence levels. This suggests that while the model is capable of recognising these shapes, it struggles to maintain a balance between precision and recall, leading to less effective overall performance.

The graph also highlights that all classes achieve an F1 score of 0.95 at a confidence threshold of 0.862, indicating a strong overall performance when the model is moderately confident. However, the variability in F1 scores for squares and triangles suggests that there may be challenges in distinguishing these shapes from one another or from other classes, especially at lower confidence levels. The relatively stable F1 scores for circles illustrate the model's reliability in classifying this shape, while the fluctuations for the other classes indicate areas that may require further refinement in model training or feature extraction.

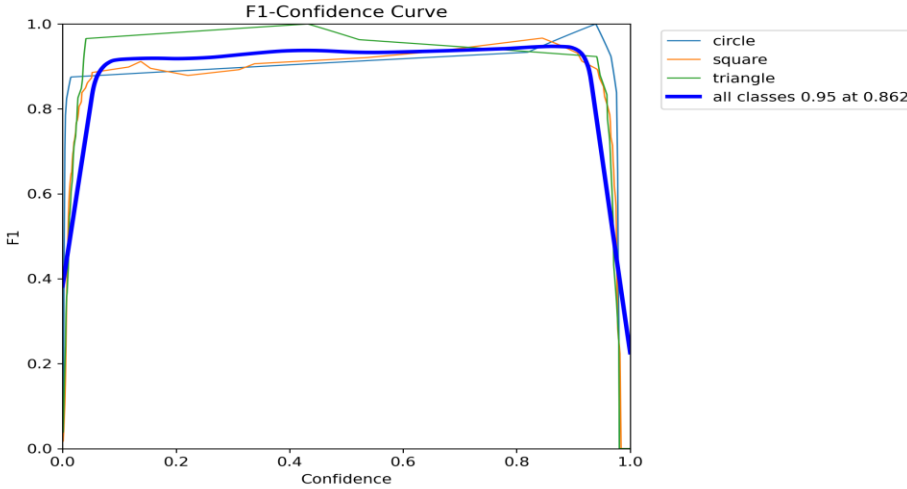


Fig. 5: F1 Curve

3.5 Performance Evaluation of Robotic Arm

3.5.1 Detection Time against Trial for Square Detection

The performance evaluation for square detection, as shown in Table 2, presents the results of ten experimental trials, focusing on detection time, sorting time, prediction accuracy, and confidence thresholds. The detection times across the trials are relatively consistent, ranging from 0.62 to 0.94 seconds, with most values clustering around 0.8 seconds. This indicates that the system is stable and reliable in recognising shapes within a short and predictable timeframe. Sorting times, on the other hand, show greater variability, ranging between 2.6 and 5.1 seconds. The shortest sorting time occurred in Trial 8 (2.6 seconds), while the longest was in Trial 10 (5.1 seconds), suggesting that the sorting mechanism may be influenced by operational or environmental factors that warrant optimisation.

In terms of prediction accuracy, the system correctly classified the square in eight out of ten trials, resulting in an overall accuracy rate of 80%. Misclassifications were observed in Trials 1 and 7, which coincided with relatively lower confidence threshold values of 0.85 and 0.72, respectively. This demonstrates a strong relationship between confidence levels and prediction accuracy, as higher confidence values (above 0.85) consistently corresponded with correct outcomes. Confidence thresholds overall ranged between 0.72 and 0.96, with the highest confidence (0.96 in Trial 4) yielding a correct classification.

Table 2: Performance Evaluation Table (Square)

Trial	Detection Time (s)	Sorting Time (s)	Prediction (True/False)	Confidence Threshold (%)
1	0.88	2.7	False	0.85
2	0.65	3.4	True	0.79
3	0.88	4.3	True	0.91
4	0.92	3.8	True	0.96
5	0.94	4.0	True	0.89
6	0.82	2.9	True	0.93

7	0.74	3.6	False	0.72
8	0.62	2.6	True	0.90
9	0.71	4.2	True	0.94
10	0.81	5.1	True	0.87

3.5.2 Detection Time against Trial for Triangle Detection

Table 3 shows the performance evaluation for triangle detection, which provides insights into the system's efficiency, accuracy, and confidence levels across ten trials. Detection times range from 0.32 to 0.89 seconds, with most values falling below 0.6 seconds, indicating that the system is generally faster in detecting triangular shapes compared to the square detection trials. The shortest detection time was observed in Trial 9 (0.32s), while the longest occurred in Trial 6 (0.89s), suggesting some variability but overall consistent speed in recognition. Sorting times vary between 2.8 and 5.4 seconds, with an average of about 3 to 4 seconds, showing moderate fluctuation similar to the square detection results, possibly influenced by the mechanical or computational aspects of sorting.

In terms of prediction outcomes, the system correctly identified triangles in 8 out of 10 trials, yielding an overall accuracy rate of 80%, identical to the square detection results. Misclassifications were observed in Trials 5 and 6, both of which had relatively higher detection times (0.78s and 0.89s) and moderate-to-high confidence thresholds (0.80 and 0.93). This shows that even with reasonably strong confidence levels, occasional errors can occur, particularly when detection time is longer than average.

Table 3: Performance Evaluation Table (Triangle)

Trial	Detection Time (s)	Sorting Time (s)	Prediction (True/False)	Confidence Threshold (%)
1	0.55	3.2	True	0.88
2	0.62	2.9	True	0.76
3	0.41	2.8	True	0.92
4	0.43	4.2	True	0.95
5	0.78	4.6	False	0.80
6	0.89	5.4	False	0.93
7	0.43	2.9	True	0.89
8	0.49	4.1	True	0.90
9	0.32	3.6	True	0.77
10	0.51	3.1	True	0.96

3.5.3 Detection Time against Trial for Circle Detection

The performance evaluation for circle detection, shown in Table 4, presents results from ten trials, assessing detection time, sorting time, prediction accuracy, and confidence thresholds. Detection times ranged between 0.22 and 0.87 seconds, with most trials clustering around 0.4–0.7 seconds. The fastest detection was recorded in Trial 4 (0.22s), while the slowest was in Trial 6 (0.87s), indicating that the system is generally quick and efficient in recognising circular shapes. Sorting times ranged from 2.1 to 3.5 seconds, with an average near 2.7 seconds, showing relatively lower variability compared to the square and triangle trials. This suggests that the sorting mechanism is more stable when processing circles.

In terms of prediction outcomes, the system achieved correct classifications in 9 out of 10 trials, resulting in a high accuracy rate of 90%, which is slightly better than the 80% accuracy observed in the square and triangle evaluations. The only misclassification occurred in Trial 6, despite a relatively high confidence threshold of 0.90, showing that even with high confidence, occasional errors can still occur. Nevertheless, this high success rate highlights the robustness of the system in detecting and classifying circular shapes.

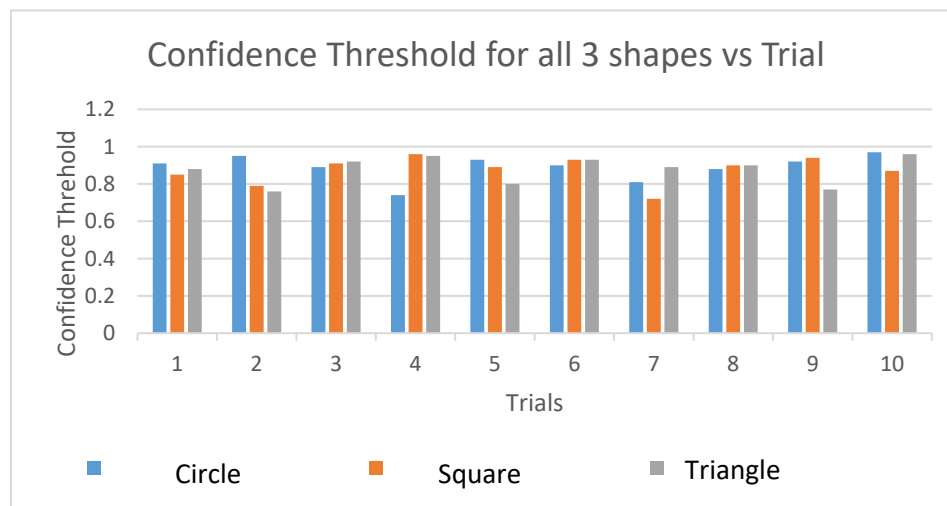
Table 4: Performance Evaluation Table (Circle)

Trial	Detection Time (s)	Sorting Time (s)	Prediction (True/False)	Confidence Threshold (%)
1	0.76	2.1	True	0.91
2	0.85	2.3	True	0.95
3	0.43	3.2	True	0.89
4	0.22	2.8	True	0.74
5	0.48	2.4	True	0.93
6	0.87	3.1	False	0.90
7	0.45	3.5	True	0.81
8	0.67	2.6	True	0.88
9	0.59	2.6	True	0.92
10	0.58	2.2	True	0.97

Figure 6 compares the confidence scores of the system across ten trials for three different shapes: circles, squares, and triangles. Overall, confidence thresholds for all shapes remain consistently high, mostly ranging between 0.75 and 0.97, which indicates the system generally predicts with strong certainty.

For circles, confidence levels are consistently above 0.8 in nearly all trials, with the highest values recorded in Trials 2, 5, and 10 (close to 0.95–0.98). This suggests that the system is particularly confident in circle detection, which aligns with earlier findings of higher classification accuracy for circles. The confidence scores fluctuate more for squares than circles, with the lowest value in Trial 7 (around 0.72) and the highest in Trial 4 (close to 0.96). Notably, misclassifications in the square trials were associated with lower confidence values, highlighting the link between prediction errors and confidence thresholds. For triangles, confidence values ranged between 0.76 and 0.96, showing a pattern similar to that of squares. While most trials achieve confidence above 0.85, dips are noticeable in Trials 2 and 9 (both around 0.76–0.77). Nonetheless, higher confidence values (above 0.9 in Trials 3, 4, 6, and 10) consistently align with correct classifications.

The chart demonstrates that the system generally produces high confidence scores across all shapes, with circles showing the most consistent high confidence, while squares and triangles exhibit slightly greater variability. The observed correlation between lower confidence values and misclassifications reinforces the importance of confidence thresholds as reliable indicators of prediction strength.

**Fig. 6:** Confidence Threshold against Trials

4. Conclusion

The integration of YOLOv11 with robotic arms for real-time object classification represents a significant advancement in the field of autonomous robotics, addressing critical challenges in dynamic environments. This study demonstrated that YOLOv11, with its sophisticated architecture and high accuracy, effectively enhances the perception capabilities of robotic systems. The model achieved a mean Average Precision of 0.99 at IoU 0.50 and 0.93 at IoU 0.50–0.95, underscoring its reliability in accurately detecting and classifying various shapes.

Furthermore, the practical implementation of the YOLOv11 model within a robotic arm facilitated rapid detection and sorting capabilities, with average detection times around 0.8 seconds. This efficiency is crucial for applications requiring swift and precise interactions in real-world scenarios. The study also identified the relationship between confidence thresholds and classification accuracy, revealing areas for potential refinement to reduce misclassifications.

Overall, the findings highlight the promise of YOLOv11 as a robust solution for real-time object classification in robotics, paving the way for future developments in intelligent systems. Continued research in optimising model performance and addressing operational challenges will further enhance the functionality and adaptability of autonomous robots in increasingly complex environments.

5. References

- Aravinda K. (2023). An embedded 3d object detection in real-time using deep learning techniques. *2023 4th International Conference on Smart Electronics and Communication (ICOSEC)*. <https://doi.org/10.1109/icosec58147.2023.10276263>
- Badgujar, C. M., Poullose, A., & Gan, H. (2024). Agricultural object detection with You Only Look Once (YOLO) Algorithm: A bibliometric and systematic literature review. *Computers and Electronics in Agriculture*, 223, 109090.
- He, L. H., Zhou, Y. Z., Liu, L., Cao, W., and Ma, J. H. (2025). Research on object detection and recognition in remote sensing images based on YOLOv11. *Scientific Reports*, 15(1), 14032.
- Hussain, M. (2023). Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection. *Machines*. <https://doi.org/10.3390/machines11070677>
- Juel, W., Haarslev, F., Krüger, N., & Bodenhagen, L. (2020). An integrated object detection and tracking framework for mobile robots. *Proceedings of the 17th International Conference on Informatics in Control, Automation and Robotics*. <https://doi.org/10.5220/0009888405130520>
- Khanam, R., & Hussain, M. (2024). Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*.
- Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*. <https://doi.org/10.1109/tnnls.2021.3084827>
- Owoeye S.O, Durodola F.O, Adetoro T.A, Adegbenro K.O, Bisiriyu B.M. (2024).Identification of Maize and Weed Using a Machine Learning Model Implemented Via YOLOv5. *FULafia Journal of Science and Technology*. 8(2); pp 49- 55.
- Reis, D., Kupec, J., Hong, J., & Daoudi, A. (2023). Real-time flying object detection with YOLOv8. *arXiv preprint arXiv:2305.09972*.
- Simeonov, M., Kamencay, P., & Dado, M. (2025). Evaluating yolov11's role in robust real-time object detection for autonomous driving. *2025 35th International Conference Radioelektronika (RADIOELEKTRONIKA)*.
- Su, Q., Mu, J., Xu, S., Wan, K., Qi, X., Zhang, Z., & Li, J. (2025). Drone object detection incorporating multi-head mixed self-attention and dynamic regression mapping loss function. *Journal of Real-Time Image Processing*. <https://doi.org/10.1007/s11554-025-01633-x>

Xiao, L., Yang, M., Li, Y., & He, Y. (2025). Rid_yolo: a lightweight and multi-object detection network for road inspection tasks. *2025 5th International Conference on Mechanical, Electronics and Electrical and Automation Control (METMS)*.
<https://www.semanticscholar.org/paper/b44d38d2890d3225b347187c0a3543c8a84a94c8>