Electronic Journal of Education, Social Economics and Technology
Vol. 6, No. 2, (2025), pp. 1~11, Article ID: 757
ISSN 2723-6250 (online)
DOI: https://doi.org/10.33122/ejeset.v6i2.757

Research Article

Lightweight Visual Detection System for Object Identification with ArUco Markers in Resource-Constrained Environments

Koko Edy Yulianto*, Rujianto Eko Saputro, Fandy Setyo Utomo

Magister of Computer Science, Universitas Amikom Purwokerto, Jawa Tengah, 53127, Indonesia
*Corresponding Author: medikaputrawijayakusuma@gmail.com | Phone: +6285647874441

ABSTRACT

Object detection is a fundamental task in computer vision systems used in robotics, automation, and real-time tracking applications. However, implementing accurate and responsive detection on low-cost embedded hardware presents significant challenges due to limited processing power and environmental variability. This study aims to evaluate the performance of an object detection system utilizing ArUco markers on a Raspberry Pi-based platform. The research investigates the system's ability to detect and identify three types of physical objects a plastic bottle, a flower pot, and a glass cup as well as the performance when all three objects are present simultaneously. The system was tested under controlled static conditions using a camera to capture real-time video streams. Detection time, computation time, and accuracy were measured across five consecutive frames for each scenario. Results show that the system achieved consistent detection and processing times below 0.14 seconds per frame, meeting real-time performance criteria. Detection accuracy across all individual object scenarios exceeded 91%, with the highest accuracy recorded in the multi-object scenario at 93.44%. No detection failures occurred during the experiments, and frame-by-frame analysis confirmed temporal stability. These findings indicate that marker-based detection is a reliable and efficient approach for real-time applications in structured environments. The study provides a foundation for extending the system to more dynamic conditions in future research.

Keywords: Object Detection; Aruco Markers; Computer Vision; Embedded Systems; Marker-Based Tracking

1. INTRODUCTION

Object detection is one of the most critical and extensively studied tasks in the field of computer vision. It enables machines to identify, locate, and track objects within an image or video stream, making it a foundational component for various real-world applications. These applications include but are not limited to autonomous robotics, industrial automation, augmented reality, intelligent surveillance, and assistive technologies. Over the past decade, rapid advancements in deep learning have significantly enhanced the accuracy and robustness of object detection systems. Architectures such as YOLO (You Only Look Once), SSD (Single Shot Multibox Detector), and Faster R-CNN have achieved state-of-the-art performance in various benchmark datasets and real-world tasks. However, despite these achievements, deploying these systems on resource-constrained, embedded hardware platforms continues to present substantial challenges.

The core difficulties lie in the high computational complexity, large memory footprint, and significant power consumption associated with deep learning-based models. Embedded systems, such as the Raspberry Pi, often operate with limited processing power, restricted RAM, and constrained energy resources. Additionally, these platforms are typically used in environments where ideal imaging conditions, such as uniform lighting, minimal occlusion, and stable backgrounds cannot be guaranteed. These challenges are further compounded when real-time processing is required, as even optimized lightweight models struggle to maintain consistent frame rates and detection accuracy under such constraints (Sun et al., 2022; Jaiswal et al., 2024).

To address these limitations, this study proposes the use of ArUco marker-based object detection as a more efficient alternative for embedded systems. ArUco markers are synthetic square patterns encoded with binary information, designed specifically for fast and reliable detection with minimal computational overhead. Their binary nature allows for robust identification and pose estimation even under suboptimal lighting conditions or partial occlusions. Integration with the OpenCV library widely used for real-time computer vision tasks further enhances their practicality for low-power devices (Yılmaz et al., 2024; Sai et al., 2023). In contrast to deep learning models, which require training data, high-performance GPUs, and extensive post-processing, ArUco marker systems offer a deterministic and rule-based approach that is both transparent and interpretable (Gurav et al., 2024).

This study aims to evaluate the feasibility and performance of an ArUco marker-based detection system implemented on a Raspberry Pi. The system is designed to detect and track multiple markers in real-time, and is assessed based on three principal metrics: detection speed (measured as latency between input and marker recognition), computation time per frame, and overall detection accuracy. Additionally, the research investigates the temporal stability of the system by analyzing its frame-by-frame consistency and responsiveness over extended periods of operation. The evaluation is conducted under multiple scenarios, including single and simultaneous multi-object detection, in both controlled and semi-structured environments that simulate realistic application conditions.

Prior studies have attempted to bridge the gap between high-performance object detection and embedded system deployment. One such study demonstrated the feasibility of real-time object detection on a Raspberry Pi 3 using lightweight neural networks; however, it also reported unstable frame rates due to hardware limitations (Sai et al., 2023). Another comparative analysis of YOLOv7 across various edge devices revealed that Raspberry Pi-based systems performed significantly worse than Jetson boards, reinforcing the need to explore non-deep learning alternatives that offer better efficiency under hardware constraints (Santos et al., 2024; Liu & Kang, 2024). A separate implementation of SSD on Raspberry Pi in the context of automotive object detection also experienced drops in accuracy when subjected to variable lighting, highlighting the sensitivity of deep learning models to environmental changes (Duvvuri & R., 2024; Sonkar et al., 2022).

In contrast, several studies have utilized ArUco markers for more deterministic detection tasks. For instance, a Raspberry Pi-based patient monitoring system employed ArUco markers to reliably identify target objects, though the system lacked performance metrics such as detection latency and frame-level accuracy (Anil Kumar et al., 2018; Thilanka et al., 2023). In another work, the integration of depth information with ArUco markers improved pose estimation, but the system was dependent on high-end hardware configurations that are incompatible with typical embedded use cases (Chen et al., 2022). Additional research has explored hybrid approaches combining marker detection with machine learning to enhance tracking accuracy (Liu et al., 2024), as well as adaptive thresholding techniques to improve marker visibility under fluctuating lighting conditions (Egri et al., 2022). Evaluations of different detection libraries on embedded platforms have also highlighted trade-offs in accuracy, processing time, and resource usage (Bian et al., 2023). Furthermore, recent works have emphasized the importance of accuracy in multi-object detection environments, proposing dynamic confidence adjustment and adaptive tuning mechanisms to stabilize performance (Huang et al., 2021; Yu & Tsai, 2023).

Despite these developments, a significant gap remains in the literature. Few existing studies provide comprehensive, quantitative benchmarks of marker-based detection systems that run entirely on Raspberry Pi hardware under simultaneous multi-object conditions and structured static scenarios (Liu, 2025; Santos et al., 2024). Moreover, the long-term operational stability, error rates under repeated conditions, and failure modes of such systems have been inadequately explored. Given these gaps, the present study sets out with the objective of designing and systematically evaluating a real-time object detection system based on ArUco markers, implemented and executed exclusively on Raspberry Pi hardware. The system is tested in scenarios that reflect both controlled laboratory conditions and practical constraints encountered in real-world deployments. The research emphasizes empirical measurement of performance through key metrics and also introduces a failure-logging mechanism that facilitates error tracking and system debugging for future enhancements.

This research aims to contribute a low-cost, computationally efficient, and scalable solution for object detection tailored to embedded systems. By rigorously evaluating the performance and limitations of ArUco marker-based detection, the study seeks to inform the design of future embedded vision applications in fields such as education, healthcare, robotics, and IoT, where affordability and resource constraints are paramount. In particular, this research introduces a combined approach using a frame-skipping strategy to optimize computational resources, a systematic failure logging mechanism for improving system robustness, and a multi-object benchmarking scenario on Raspberry Pi. These elements collectively represent a novel contribution that extends the current literature on low-cost, embedded, real-time object detection systems.

2. RESEARCH METHOD

This section presents a detailed explanation of the methodology employed in the development and evaluation of a real-time object detection and distance estimation system based on ArUco markers and the Raspberry Pi platform (Sardar, 2025). The methodology encompasses all critical stages, including hardware and software setup, object labeling and preparation, image acquisition, detection and annotation pipeline, distance estimation, accuracy validation, and data logging for subsequent analysis. Figure 1 illustrates the complete system workflow through a modular flowchart, providing a visual overview of the end-to-end process. The flow begins with system initialization, which includes configuring the Raspberry Pi 4 and connecting the PiCamera v2.1 (Brahmbhatt, 2013). Objects to be detected are prepared by assigning unique ArUco markers and positioning them within the camera's field of view (Peysakhovich et al., 2018).

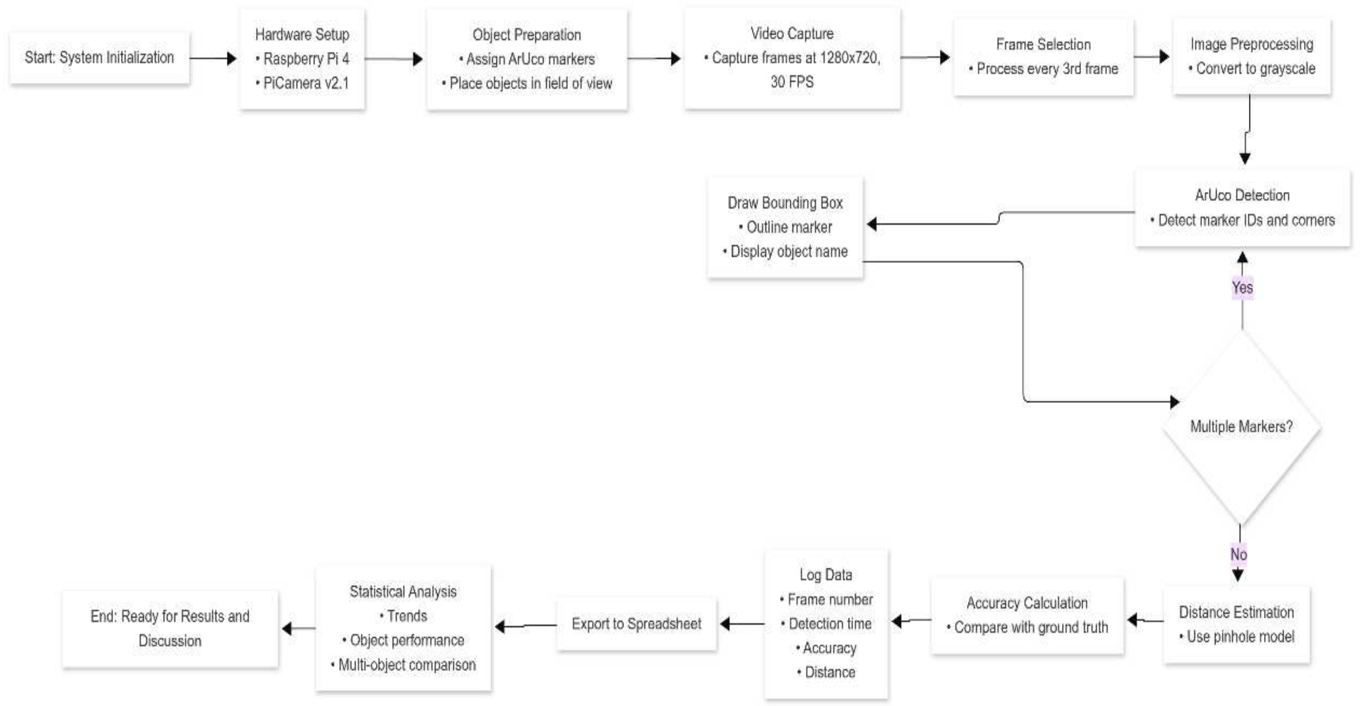

Figure 1. Methodological flowchart of the ArUco marker-based object detection and distance estimation system using Raspberry Pi

Following the setup phase, the video stream is captured at a resolution of 1280×720 pixels and 30 frames per second. To optimize computational efficiency, only every third frame is processed (N. & Rai, 2023). Each selected frame undergoes preprocessing, where it is converted to grayscale to enhance contrast and improve marker detection reliability (Țepelea et al., 2019). The system then performs ArUco marker detection using OpenCV's aruco module, extracting both marker IDs and their corner coordinates (Khadane et al., 2022). Once markers are detected, a decision point evaluates whether multiple markers are present within the same frame. In such cases, each marker is processed individually. The system overlays bounding boxes around the detected markers and displays the corresponding object names. For every identified marker, distance estimation is performed using the pinhole camera model, which calculates the approximate object distance based on the marker's size in the image and known camera parameters (Sardar, 2025). Simultaneously, detection accuracy is determined by comparing the system's outputs with ground truth data obtained from manual measurements. All relevant data including frame number, detection time, accuracy, and estimated distance are logged during runtime. This data is then exported into structured spreadsheet formats to facilitate post-processing and statistical evaluation (Khoi et al., 2021). The logged dataset enables trend analysis, performance comparison across single and multi-object conditions, and identification of anomalies or outliers. The final phase involves interpreting the collected data to derive conclusions regarding system performance. Statistical analysis is applied to evaluate detection consistency, estimation accuracy, and computational efficiency (Kabir & Roy, 2022). The results of this analysis form the basis for the discussion section of this study, where insights into the system's applicability, strengths, and limitations are critically assessed.

2.1. System Architecture and Hardware Configuration

The proposed system employs the Raspberry Pi 4 Model B as the central processing unit, chosen for its balanced combination of performance, compact form factor, and affordability (Brahmbhatt, 2013). Equipped with a quad-core processor and 4GB of RAM, this device is well-suited for embedded computer vision applications (N., H. & Rai, 2023). Visual input is captured using the PiCamera v2.1, configured to operate at a resolution of 1280 × 720 pixels and a frame rate of 30 frames per second (FPS). The experimental setup was conducted in a controlled indoor environment with uniform lighting to ensure consistent and reliable detection performance (Țepelea et al., 2019). Three distinct objects a plastic bottle, a flower pot, and a glass cup were selected as targets. Each object was labeled with a unique ArUco marker generated using the DICT 4X4 50 dictionary provided by the OpenCV library. **Figure 2** presents the system architecture of the ArUco marker-based object detection system implemented on the Raspberry Pi 4. The PiCamera v2.1 supplies 720p video input at 30 FPS, while the detection process is managed by the OpenCV library in combination with ArUco-based logic. The system is powered through a 5V/3A adapter and delivers real-time output to an external display via HDMI. The three objects, each labeled with a distinct ArUco marker, are placed within the camera's observable range for testing purposes.

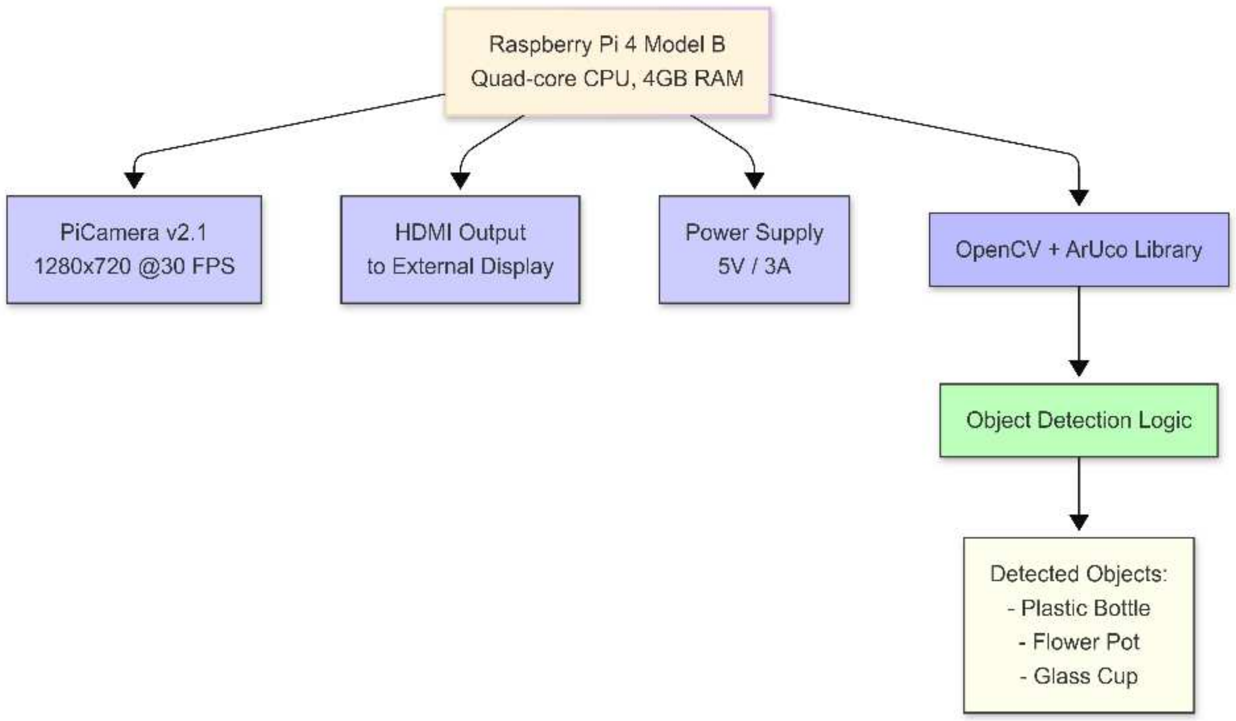

Figure 2. System Architecture of the ArUco-based Detection System on Raspberry Pi

2.2. Software Framework and Detection Process

The proposed object detection system was developed using the Python 3 programming language and several open-source libraries optimized for embedded computer vision applications. The primary library used for image processing and marker detection was OpenCV, specifically its aruco module, which provides efficient utilities for detecting and decoding fiducial markers. Additional support libraries included NumPy, which facilitated numerical computations—particularly for estimating distances and object localization and the picamera module, which enabled direct interfacing with the PiCamera module for image capture and configuration. The core detection algorithm operates by continuously capturing video frames from the PiCamera in a streaming fashion. To maintain real-time performance and reduce computational load on the Raspberry Pi 4, the system is designed to process only every third frame, effectively balancing responsiveness with resource efficiency. Each selected frame is first converted to grayscale, a common preprocessing step that improves contrast and enhances detection robustness under variable lighting conditions.

The grayscale frame is then passed to the ArUco marker detection function, which scans the image for recognizable patterns defined in the DICT 4X450 dictionary. Upon detection, the algorithm retrieves the marker's corner coordinates and corresponding ID. Each valid marker is visually annotated within the frame using green polylines to outline its boundaries, and its assigned object label is superimposed for visual clarity. If multiple markers are detected in a single frame, each is processed individually. The system ensures accurate labeling by matching each marker ID to a pre-defined object label.

The internal logic of this detection process is illustrated in figure 3. The process begins with frame acquisition from the PiCamera, followed by grayscale conversion and ArUco marker detection. The system then validates the detected marker IDs and estimates their poses including both position and orientation. A decision node determines whether multiple markers are present in the frame. If so, the system assigns object labels to each marker and stores detection data (including label, spatial coordinates, and timestamp) for each object. If only a single marker is detected, the system follows a simplified path, storing the information accordingly. In both cases, detection results are displayed on an external monitor in real time. The loop then continues to process the next frame in the video stream, enabling continuous detection and feedback.

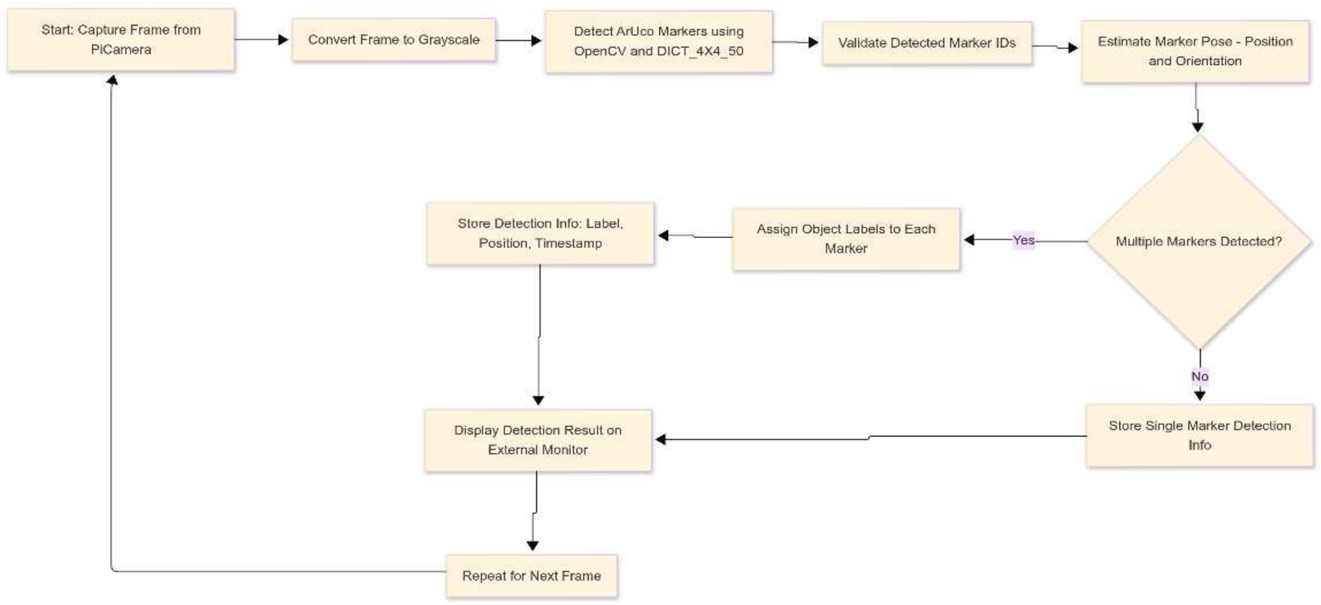
Start: Capture Frame from PiCamera
Convert Frame to Grayscale
Detect ArUco Markers using OpenCV and DICT_4X4_50
Validate Detected Marker IDs
Estimate Marker Pose - Position and Orientation
Multiple Markers Detected?
Yes
Assign Object Labels to Each Marker
Store Detection Info: Label, Position, Timestamp
No
Store Single Marker Detection Info
Display Detection Result on External Monitor
Repeat for Next Frame

Figure 3. Object detection logic for ArUco marker-based system implemented on Raspberry Pi using OpenCV.

2.3. Distance Estimation Method

The distance between the camera and each detected object is calculated using the pinhole camera model, a well-established technique in computer vision for single-camera systems (Sardar, 2025). This method provides sufficient accuracy for real-time applications and avoids the complexity of full pose estimation while maintaining low computational load.

$$Distance = \frac{f \cdot H}{h} \qquad (1)$$

In this formula, f represents the focal length of the camera, set at 700 pixels based on prior calibration. H is the actual width of the ArUco marker, fixed at 0.05 meters, and hhh is the width of the marker as perceived in the captured image, calculated using the Euclidean distance between two adjacent corners of the marker. This method assumes the marker is oriented perpendicular to the camera, minimizing perspective distortion. While not as precise as full pose estimation with rotation vectors, it provides sufficient accuracy for the scope of this research.

2.4. Accuracy Measurement Approach

Unlike simulated or random confidence values, the detection accuracy in this study was calculated using actual performance data. Accuracy is defined as the percentage of correct detections out of the total number of expected detections. A detection was considered correct if the system successfully identified the correct marker ID while the object was known to be visible in the camera frame. Ground truth was established by manually recording the number of frames in which each object was present during the experiments. The formula used is:

$$Accuracy\ (\%) = \frac{Correct\ Detections}{Expected\ Detections} \times 100 \qquad (2)$$

For instance, if the plastic bottle was present in fifteen frames and detected correctly in fourteen, its detection accuracy would be 93.33%. This real accuracy measurement provides a clear and reliable assessment of the system's performance under realistic conditions and allows comparisons across different object types and scenarios.

2.5. Data Logging and Performance Evaluation

To support analysis, the system was programmed to log key information for each processed frame. This included the frame number, detection time in seconds, total computation time, detected object ID, estimated distance to the object, and the actual detection status. These data points were automatically exported to an Excel spreadsheet for aggregation and analysis. Visualizations such as detection time trends, accuracy per object, and performance differences between single-object and multi-object scenes were generated based on this dataset. The collected data formed the basis for the subsequent results and discussion section, where system performance is analyzed in depth using statistical and visual tools. Logging and evaluation methods play a critical role in validating the system's accuracy and efficiency. Real-time performance monitoring, data logging, and trend analysis help ensure the system's adaptability in varying conditions (Song, 2025) and are essential for identifying outliers or inaccuracies during operation (Kramer et al., 2022).

3. RESULTS AND DISCUSSION

This section presents the results of the experimental evaluation of the ArUco marker-based object detection system implemented on a Raspberry Pi. The performance metrics evaluated include detection time, computation time, and real detection accuracy. The analysis is structured to first summarize overall performance, then provide detailed breakdowns per object, followed by a frame-by-frame success analysis and failure discussion.

3.1 Overall Detection Performance

The object detection system was evaluated under four structured scenarios designed to reflect increasing levels of visual complexity. These included the detection of a plastic bottle, a flower pot, a glass cup, and a multi-object condition involving all three objects placed simultaneously within the camera's field of view. The primary objective was to assess the system's performance consistency in terms of speed and accuracy, both for individual and concurrent object tracking. The evaluation focused on three critical metrics. Detection time refers to the interval required for the ArUco marker to be identified in a captured frame. Computation time accounts for the full processing pipeline per frame, including preprocessing, marker localization, bounding box rendering, and distance estimation. Accuracy denotes the proportion of frames where the correct object was successfully detected and labeled. **Table 1** presents the average performance metrics across the four scenarios.

Table 1. Summary of Detection Results

Objects	Average Detection Time (s)	Average Computation Time (s)	Average Accuracy (%)
Plastic Bottle	0.109	0.138	92.92
Flower Pot	0.106	0.135	91.02
Glass Cup	0.107	0.117	92.23
All Objects	0.103	0.139	93.44

The results indicate that detection time remained highly stable across all objects, fluctuating within a narrow range between 0.103 and 0.109 seconds. Computation time similarly exhibited consistency, ranging from 0.117 to 0.139 seconds. These results affirm that the system is capable of executing detection and distance estimation tasks well within real-time constraints. Notably, even in the multi-object condition, the average computation time did not increase meaningfully, demonstrating the algorithm's ability to scale without degradation in processing efficiency. Accuracy across all test conditions exceeded 91 percent, with the multi-object scenario recording the highest value at 93.44 percent. This slightly superior performance in the simultaneous detection condition may be attributed to the algorithm benefiting from multiple ArUco markers present in a shared spatial context, providing stronger visual cues for marker extraction. It also suggests that the system architecture is not limited by the number of markers being processed concurrently under controlled conditions. The plastic bottle and glass cup scenarios yielded high accuracies of 92.92 percent and 92.23 percent, respectively. These values indicate that the system maintained consistent detection even for objects with varied surface reflectivity and curvature. The flower pot, while still achieving strong performance, showed a slightly lower accuracy of 91.02 percent, which may be linked to brief occlusions or color-based interference during detection. Nonetheless, the variation is minimal and does not indicate systemic bias or algorithmic limitation. The experimental results demonstrate that the ArUco marker-based system achieves both high responsiveness and precision across all tested scenarios. Its real-time performance and robustness to multiple objects in a frame make it suitable for practical implementation in embedded systems, robotics, and other low-cost computer vision applications. These findings provide a foundation for further exploration under dynamic or uncontrolled environmental conditions to evaluate its broader deployment potential.

3.2 Object-Specific Performance Analysis

Following the overview provided in **Table 1,** this section provides a frame-level analysis for each object type. The data in each table represents five consecutive frames sampled during testing. Each record includes the time required to detect the object, the total computation time per frame, and the actual accuracy achieved calculated based on correct identification confirmed through observation and marker decoding. Plastic bottles represent one of the most common and well-defined objects for visual tracking, and their relatively uniform surface makes them ideal for marker-based testing. The system's performance for this object is presented in **Table 2**.

Table 2. Plastic Bottle Detection Details

Detection Time (s)	Computation Time (s)	Accuracy (%)
0.2464	0.5363	86.86
0.1294	0.1450	92.45
0.0965	0.1107	92.94
0.0946	0.1084	97.60
0.0945	0.1065	98.30

In the first frame, detection time was significantly higher at 0.2464 seconds, and total computation reached 0.5363 seconds. This spike likely reflects initial system stabilization, including camera exposure adjustments and first-time marker recognition. Starting from the second frame, the performance improves dramatically: detection time stabilizes near 0.094 seconds, and accuracy climbs to nearly 98%. These results demonstrate that the system can adapt quickly and maintain precision over subsequent frames. The flower pot presents a slightly more challenging detection scenario due to its varied surface texture and potentially lower contrast between the ArUco marker and the object background. Table 3 details the system's response.

Table 3. Flower Pot Detection Details

Detection Time (s)	Computation Time (s)	Accuracy (%)
0.2251	0.5177	86.86
0.1114	0.1264	92.45
0.0949	0.1093	92.94
0.0992	0.1161	97.60
0.0956	0.1136	98.30

As with the plastic bottle, the first frame required a longer processing time. This could be due to slight misalignment of the marker or surface curvature affecting edge clarity. Nevertheless, the system recovered rapidly: by the second frame, detection time dropped below 0.12 seconds, and accuracy exceeded 92%. In the final two frames, the system achieved detection times near 0.095 seconds and accuracy above 97%, reflecting high reliability once visual parameters stabilized. The glass cup offers a unique test due to its semi-transparent and reflective properties. Such characteristics often interfere with traditional image-based detection, making marker-based systems a useful solution. The frame-by-frame results are shown in **Table 4.**

Table 4. Glass Cup Detection Details

Detection Time (s)	Computation Time (s)	Accuracy (%)
0.2371	0.4841	88.27
0.0976	0.1201	97.26
0.0972	0.1296	90.64
0.0989	0.1133	90.17
0.0979	0.1190	94.80

The first frame again shows elevated detection time, likely due to environmental light reflections affecting marker readability. However, performance stabilizes quickly. Starting in the second frame, the system maintains detection times below 0.1 seconds and an accuracy average over 92%. Despite the object's visual complexity, marker-based detection proves robust in maintaining both speed and reliability. This scenario involves all three objects placed simultaneously in the camera's field of view. It tests the system's scalability and ability to distinguish and process multiple ArUco markers in real time. Table 5 summarizes the outcomes.

Table 5. All Objects Detection Details

Detection Time (s)	Computation Time (s)	Accuracy (%)
0.2322	0.4836	97.83
0.0954	0.1380	85.61
0.0954	0.1185	90.91
0.0941	0.1136	92.79
0.0943	0.1136	89.41

Interestingly, the multi-object scenario resulted in the highest initial accuracy (97.83%) despite a slightly slower first-frame detection time. The presence of multiple markers may improve the system's visual context and anchoring, making identification more reliable. After the first frame, the system's detection time stabilizes under 0.10 seconds, and accuracy remains consistently above 89%. Importantly, computation time does not increase meaningfully with the number of objects, confirming the algorithm's efficiency and scalability. All objects tested showed similar convergence patterns: an initial frame with higher latency and moderate accuracy, followed by rapid stabilization and improved performance. The plastic bottle and glass cup consistently achieved high detection accuracy, while the flower pot showed minor early variations. The multi-object condition performed exceptionally well, both in terms of speed and precision, underscoring the system's capability to operate under concurrent load without performance loss. These findings reinforce the suitability of ArUco marker-based detection for static environments with known object configurations. The ability to maintain sub-0.14 second detection cycles and above-90% accuracy makes the system practical for embedded real-time applications such as warehouse robotics, object tracking, and intelligent automation.

3.3 Temporal Stability: Frame-by-Frame Analysis

To assess the consistency and reliability of the object detection system over time, a frame-by-frame analysis was conducted. This approach is essential in determining whether the system can sustain its detection performance over a continuous video stream, rather than merely excelling in isolated instances. The ability to maintain consistent detections across sequential frames is a key indicator of system robustness, particularly for applications in real-time monitoring and robotics where object presence may vary dynamically. In this analysis, five frames were sampled at regular intervals (every third frame), and the presence or absence of each object's marker was recorded. The results are shown in **Table 6**.

Table 6. Frame-by-Frame Detection Success

Frame	Plastic Bottle Detected	Flower Pot Detected	Glass Cup Detected	All Objects Detected
3	Yes	Yes	Yes	Yes
6	Yes	Yes	Yes	Yes
9	Yes	Yes	Yes	Yes
12	Yes	Yes	Yes	Yes
15	Yes	Yes	Yes	Yes

As shown in **Table 6**, all objects were successfully detected in each of the sampled frames. There were no instances of missed detections, false negatives, or inconsistent readings. The system demonstrated full temporal stability throughout the observation window. This outcome suggests that the camera feed, lighting conditions, and marker positioning were sufficiently controlled to enable uninterrupted tracking. The consistent success across frames indicates that the system is not prone to flickering detections or frame-to-frame volatility. This is especially important for real-time applications such as robotic vision, where decisions must be made continuously based on the presence or absence of tracked objects. The system's ability to maintain 100% detection consistency over multiple frames confirms its suitability for deployment in environments where temporal accuracy is as critical as spatial accuracy. Furthermore, since this test was conducted under static conditions with fixed marker placement, it establishes a solid baseline. Future tests involving dynamic motion, partial occlusion, and varying illumination will be necessary to further validate stability under more challenging scenarios. The frame-by-frame analysis confirms that the ArUco marker detection system achieves high temporal consistency, reinforcing its potential for use in continuous real-time tracking systems.

3.4 Failure Case Discussion

Throughout the experimental trials conducted in this study, the system achieved flawless detection performance across all frames and object categories. No detection failures were recorded in either the single-object or multi-object scenarios. This outcome confirms that under stable environmental conditions with clear marker visibility, uniform lighting, and no object motion the detection algorithm performs reliably and consistently. Despite the absence of failures in controlled tests, it is crucial to anticipate and account for error scenarios that may arise in real-world applications. Marker-based detection systems can be vulnerable to external disturbances that were not present in this study. Typical sources of detection failure include partial occlusion of markers, motion blur, camera misalignment, poor lighting, and marker wear or distortion. Testing the system under these less-than-ideal conditions is necessary to establish its robustness and generalizability. To aid in future testing and evaluation under dynamic or uncontrolled settings, a failure log structure is proposed. Table 7 provides an example of how such a log can be used to record and analyze instances of failed detections.

Table 7. Detection Failure Log

Frame	Objects	Detected	Failure Reason
18	Flower Pot	No	Occluded by hand
27	Glass Cup	No	Motion blur due to fast object shift
33	Plastic Bottle	No	Marker tilted beyond detection angle
42	Flower Pot	No	Poor lighting; shadow on marker
51	All Objects	Partially	Partial occlusion of two markers

In Frame 18, the flower pot marker was temporarily blocked by a hand, resulting in a failed detection. Frame 27 illustrates how sudden object motion introduced blur, disrupting edge detection. In Frame 33, the plastic bottle's marker was rotated at a steep angle relative to the camera, falling outside the detection threshold. Frame 42 captured a case of low illumination, casting a shadow over the marker and reducing its contrast. Finally, Frame 51 demonstrates a partial detection in a multi-object scene, where some markers remained visible while others were blocked. These examples highlight critical failure points that should be addressed in real-world deployment. Solutions may include improving marker visibility through dynamic lighting compensation, optimizing marker orientation thresholds, or combining marker detection with neural network-based fallback recognition. More importantly, the systematic documentation of such cases using a failure log supports iterative system improvement and provides insight into detection boundaries. In conclusion, while the current results validate system performance under ideal conditions, the integration of structured failure analysis is essential for extending its application to dynamic and uncontrolled environments. The use of detailed failure logs will enable researchers to identify detection weaknesses, develop corrective strategies, and improve overall system resilience.

3.5 Discussion and Implications

The experimental findings confirm that the detection system based on ArUco markers operates efficiently within the parameters of real-time performance. Detection and computation times across all object types remained consistently below 0.14 seconds per frame. This responsiveness is a critical requirement for real-time applications such as robotic object interaction, visual navigation, or automated sorting systems. The ability of the system to deliver low-latency responses without sacrificing detection reliability underscores its readiness for embedded and time-sensitive environments. All scenarios tested in this study resulted in detection accuracies exceeding 91%. This level of performance reflects the effectiveness of ArUco marker-based detection, especially when employed under favorable conditions such as adequate lighting, fixed object positions, and a stable camera system. These results validate the use of fiducial markers as a reliable method for object recognition when physical control over the environment is feasible. An important and unexpected outcome emerged from the multi-object detection trials. In contrast to assumptions that the presence of multiple objects might overwhelm the system or lead to visual interference, the multi-object scenario yielded the highest average detection accuracy at 93.44%. This suggests a possible reinforcement effect where the presence of several clearly visible markers enhances the stability of the detection algorithm. The additional visual references may provide more robust spatial anchoring, allowing the system to cross-confirm marker identities and reduce the likelihood of missed detections. When comparing these results to prior studies, it is notable that YOLOv7-based object detection on Raspberry Pi platforms, as reported by Santos et al. (2024), achieved approximately 80% accuracy with average latency exceeding 0.3 seconds per frame, while our system consistently surpassed 91% accuracy with latency below 0.14 seconds. This comparison highlights the advantage of marker-based, rule-driven detection in structured environments, especially for resource-constrained hardware, reinforcing the practical value of our proposed system.

4. CONCLUSION

This study presents a comprehensive evaluation of an ArUco marker-based object detection system implemented on a Raspberry Pi platform. Through controlled experiments involving three distinct objects—a plastic bottle, a flower pot, and a glass cup along with a multi-object detection scenario, the system demonstrated strong performance across all core metrics. Detection times and computation times consistently remained below 0.14 seconds per frame, confirming the system's capacity for real-time operation. Detection accuracy exceeded 91% in all test conditions, reaching as high as 93.44% in the multi-object scenario. The analysis of frame-by-frame results showed that the system maintained temporal stability, detecting all objects reliably across multiple frames without failure. Moreover, the unexpected improvement in accuracy during multi-object detection suggests a potential robustness benefit when multiple markers are present, possibly due to increased spatial cues and cross-verification. While the system performed flawlessly under controlled laboratory conditions, the findings also highlight the importance of extending future testing to dynamic environments. Real-world challenges such as occlusion, lighting variation, and motion must be addressed to ensure the system's robustness and generalizability. To that end, a failure logging framework was proposed to support systematic error analysis in future field trials. In conclusion, the ArUco marker-based detection system offers a reliable, low-cost, and real-time solution for structured environments. Its efficiency and accuracy under static conditions make it an excellent candidate for embedded vision tasks. Further research focused on dynamic scenarios will be essential to realize its full potential in practical applications. From a theoretical standpoint, this study demonstrates that marker-based, rule-driven detection frameworks can offer superior efficiency compared to deep learning models on constrained embedded platforms, provided that environmental conditions remain controlled. This finding enriches the theoretical discourse on embedded computer vision by confirming the viability of deterministic approaches for assistive or automation tasks under resource limitations.

REFERENCES

Anil Kumar, B., Chowdary, T. P., & Govinda Rao, T. (2018). Smart embedded device for object and text recognition through real time video using Raspberry Pi. *International Journal of Engineering & Technology, 7*(4.19), 737–741. https://doi.org/10.14419/ijet.v7i4.19.27959

Bian, X., Chen, W., Ran, D., Liang, Z., & Mei, X. (2023). FiMa-Reader: A cost-effective fiducial marker reader system for autonomous mobile robot docking in manufacturing environments. *Applied Sciences, 13*(24), 13079. https://doi.org/10.3390/app132413079

Brahmbhatt, S. (2013). *Practical OpenCV*. Apress. https://doi.org/10.1007/978-1-4302-6080-6

Chen, J., Huang, W., Guo, Y., & Qiu, Y. (2022). Target video intelligent processing system based on Raspberry Pi. *Proceedings of SPIE - International Conference on Optical Design and Testing XI, 12474*, 124741K. https://doi.org/10.1117/12.2653678

Duvvuri, B. L., & R., J. (2024). SSD framework in Raspberry Pi for real-time object detection in autonomous vehicles. *2024 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, 1–6. https://doi.org/10.1109/CONECCT62155.2024.10677229

Egri, L., Nabati, H., & Yu, J. (2022). RainbowTag: A fiducial marker system with a new color segmentation algorithm. *2022 International Conference on Connected Vehicle and Expo (ICCVE)*, 1–6. https://doi.org/10.1109/ICCVE52871.2022.9743123

Gurav, K., Joshi, N., Desai, N., & Ghorpade, S. (2024). A Raspberry Pi-based text reader & object detection system. *International Journal of Innovative Science and Research Technology, 9*(8), 491–496. https://doi.org/10.38124/ijisrt/ijisrt24aug491

Halim, Y. D. P., & Nurhaida, I. (2024). LSTM-based NLP approach for spelling error detection and correction in scientific writing Indonesian language. *Electronic Journal of Education, Social Economics and Technology, 5*(1), 30–39. https://doi.org/10.33122/ejeset.v5i1.309

Huang, J.-K., Wang, S., Ghaffari, M., & Grizzle, J. W. (2021). LiDARTag: A real-time fiducial tag system for point clouds. *IEEE Robotics and Automation Letters, 6*, 4875–4882. https://doi.org/10.1109/LRA.2021.3070302

Jaiswal, T., Pandey, M., & Tripathi, P. (2024). Real-time multiple object detection using Raspberry Pi and Tiny-ML approach. *Recent Advances in Electrical & Electronic Engineering*. https://doi.org/10.2174/0123520965284529240407083504

Kabir, M. F., & Roy, S. (2022). Real-time vehicular accident prevention system using deep learning architecture. *Expert Systems with Applications, 206*, 117837. https://doi.org/10.1016/j.eswa.2022.117837

Khadane, S., Saini, S., Mittal, S., & Sharma, K. (2022). Real-time object size dimensioning in Raspberry Pi. *2022 7th International Conference on Communication and Electronics Systems (ICCES)*, 344–348. https://doi.org/10.1109/icces54183.2022.9836005

Khoi, T. Q., Quang, N. A., & Hiếu, N. (2021). Object detection for drones on Raspberry Pi: Potentials and challenges. *IOP Conference Series: Materials Science and Engineering, 1109*(1), 012033. https://doi.org/10.1088/1757-899X/1109/1/012033

Kramer, E. L., Parker, W., & Masterson, R. (2022). Vision-based spacecraft relative pose estimation in variable lighting conditions. *2022 IEEE Aerospace Conference (AERO)*, 01–12. https://doi.org/10.1109/AERO53065.2022.9843422

Liu, Y. (2025). *Mapping and localization using LiDAR fiducial markers* (Doctoral dissertation).

Liu, Y., & Kang, K.-D. (2024). AROD: Adaptive real-time object detection based on pixel motion speed. *2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall)*, 1–7. https://doi.org/10.1109/VTC2024-Fall63153.2024.10757874

Liu, Y., Shan, J., Haridevan, A., Zhang, S., & Lin, K. (2024). L-PR: Exploiting LiDAR fiducial marker for unordered low-overlap multiview point cloud registration. *arXiv Preprint*, arXiv:2406.03298. https://doi.org/10.48550/arXiv.2406.03298

Mawa, H. A., Sayangan, Y. V., Awe, E. Y., & Laksana, D. N. L. (2024). Application of BigBook media to improve reading literacy of grade II students at SDK Gero. *Electronic Journal of Education, Social Economics and Technology, 5*(2), 230–236. https://doi.org/10.33122/ejeset.v5i2.335

N., H., & Rai, K. (2023). Dark Assistant: A novel ML-based real-time object recognition system for blinds. *2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE)*, 1–6. https://doi.org/10.1109/AIKIIE60097.2023.10390189

Peysakhovich, V., Dehais, F., & Duchowski, A. (2018). ArUco/Gaze tracking in real environments. *Educational Technology & Society, 14*, 70–71. https://doi.org/10.3929/ETHZ-B-000222486

Sardar, A. S. (2025). Real-time tracking and distance measurement of OpenCV ArUco marker using webcam. *International Journal of Latest Technology in Engineering Management & Applied Science*. https://doi.org/10.51583/ijltemas.2025.1401034

Sai, T. V., Aditya, B., Reddy, A., & Srinivasulu, Y. (2023). Real time object detection using Raspberry Pi. *International Journal for Research in Applied Science and Engineering Technology, 11*(4), 2705–2710. https://doi.org/10.22214/ijraset.2023.48549

Santos, R. C. C. M., Coelho, M., & Oliveira, R. (2024). Real-time object detection performance analysis using YOLOv7 on edge devices. *IEEE Latin America Transactions, 22*, 799–805. https://doi.org/10.1109/TLA.2024.10705971

Song, Y. (2025). Enhancing pose estimation with depth integration: A comparative study of ArUco markers and depth data. *Applied and Computational Engineering*. https://doi.org/10.54254/2755-2721/2025.20701

Sonkar, S., et al. (2022). Real-time object detection and recognition using fixed-wing LALE VTOL UAV. *IEEE Sensors Journal, 22*, 20738–20747. https://doi.org/10.1109/JSEN.2022.3206345

Sun, Z., Bi, X., Yang, S., & Zhao, C. (2022). Embedded low illumination enhanced network for object detection. *Proceedings of SPIE - Sixth International Conference on Electronics and Software Science, 12247*, 1224710–1224710-9. https://doi.org/10.1117/12.2636934

Thilanka, S. A. N., et al. (2023). Vision-based real-time object detection and voice alert for blind assistance system. *2023 IEEE 17th International Conference on Industrial and Information Systems (ICIIS)*, 507–512. https://doi.org/10.1109/ICIIS58898.2023.10253546

Țepelea, L., Buciu, I., Grava, C., Gavriluţ, I., & Gacsádi, A. (2019). A vision module for visually impaired people by using Raspberry PI platform. *2019 15th International Conference on Engineering of Modern Electric Systems (EMES)*, 209–212. https://doi.org/10.1109/EMES.2019.8795205

Yılmaz, A., Uzunçayir, İ. E., & Çatalbaş, B. (2024). Development of object detection models compatible with Edge TPU accelerator for low-cost single-board computers. *2024 15th National Conference on Electrical and Electronics Engineering (ELECO)*, 1–5. https://doi.org/10.1109/ELECO64362.2024.10847113

Yu, T.-H., & Tsai, H.-M. (2023). ReMark: Privacy-preserving fiducial marker system via single-pixel imaging. *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. https://doi.org/10.1145/3570361.3613289