Journal of Applied Engineering and Technological Science
Vol 6(1) 2024: 69-85

AN ENHANCED IMAGE SEGMENTATION TECHNIQUE-BASED ON MOTION DETECTION ALGORITHM

Zaid Sh. Bakr[1], Hamzah M. Marhoon[2*], Ammar Alaythawy[3]
Department of Automation Engineering and Artificial Intelligence, College of Information Engineering, Al-Nahrain University, Jadriya, Baghdad, Iraq[1,2,3]
zaid.shafeeq@nahrainuniv.edu.iq[1], hamzah.marhoon@nahrainuniv.edu.iq[2], ammar.alaythawy@nahrainuniv.edu.iq[3]

Received: 09 September 2024, Revised: 17 November 2024, Accepted: 19 November 2024
**Corresponding Author*

ABSTRACT
This paper presents a prototype for an intelligent, self-contained theft detection system designed for small-scale applications. Utilizing a Raspberry Pi 3 as the core processing unit, the system employs a motion-detecting camera to monitor a defined area, recording and securely archiving video data on a cloud server upon detecting movement. This cloud-based repository supports real-time analysis, ensuring that data remains available for future reference. Battery-powered configuration enhances the system's portability, making it adaptable across various environments, such as healthcare for patient monitoring or wildlife tracking for behavioural studies. The design aligns with IoT principles, featuring autonomous operation and cloud connectivity, offering a scalable, flexible solution capable of integration into larger IoT ecosystems for diverse surveillance applications.
Keywords: *IoT, Motion detection, Raspberry Pi 3, Firebase Cloud*.

1. Introduction

Image segmentation is a fundamental process in the field of computer vision, enabling the partitioning of an image into meaningful segments based on shared characteristics such as color, texture, and intensity (Amiriebrahimabadi et al., 2024). The segmented regions facilitate applications in various domains, including medical imaging, where segmentation assists in isolating organs and tissues; remote sensing, which benefits from distinguishing different land types or vegetation; and autonomous vehicles, where segmentation aids in recognizing roads, obstacles, and pedestrians (Lei et al., 2024). As technology advances, the demand for highly accurate and efficient segmentation methods continues to grow, particularly as applications expand to real-time systems, such as video surveillance and dynamic scene analysis. Conventional segmentation techniques include a variety of algorithmic approaches, such as thresholding, clustering, edge detection, and region-based methods, each tailored to specific use cases and yielding optimal results in static environments (Mahmood, 2024). For instance, thresholding methods like Otsu's algorithm apply a set intensity threshold to convert an image into binary form, useful for distinguishing objects with high contrast from their background (Basil et al., 2022; Pang et al., 2024). Clustering methods, such as K-means and Gaussian Mixture Models (GMM), group pixels based on shared features like colour or texture, often yielding excellent segmentation results in multimodal images with varied intensities (Mishra et al., 2024). Edge detection algorithms like the Canny and Sobel operators identify the boundaries between regions by focusing on abrupt changes in intensity, which proves effective in delineating shapes within an image (You et al., 2020). Additionally, region-based techniques, including Region Growing and Watershed algorithms, use seed points that expand by incorporating neighbouring pixels that meet certain homogeneity conditions, making these methods suitable for applications requiring precise boundary definitions (Basil & Marhoon, 2023; Garea & Das, 2024). These algorithms represent the backbone of conventional image segmentation approaches, excelling in applications where the scene remains largely static. However, static segmentation methods often struggle in dynamic environments where the scene involves motion or rapid changes in visual content. This limitation has driven researchers to explore more advanced approaches that integrate motion detection to accommodate the challenges presented by dynamic scenes. Motion detection algorithms, which distinguish moving objects from static backgrounds, provide valuable support for applications such as real-time video surveillance, traffic monitoring, and automated video analysis (Rayed et al., 2024). Integrating motion detection with segmentation allows for isolating regions of interest by focusing only on areas where activity occurs, reducing computational complexity and improving system efficiency. This combined approach is particularly promising for environments requiring adaptive, real-time responses and has the potential to advance segmentation accuracy in dynamic scenes. In recent years, real-time video assessment through motion algorithms has shown significant promise, especially in applications where stationary backgrounds and moving entities need to be distinguished for enhanced segmentation. By combining motion detection with image segmentation, researchers can target key regions of interest, thereby increasing object identification, tracking, and analytics capabilities (Basil, Marhoon, et al., 2024; Hassan et al., 2024). This synergy holds immense potential for applications such as high-speed road monitoring, where vehicles need to be accurately tracked; wildlife monitoring, where animal behaviour can be observed in dynamic natural environments; and even healthcare, where patient activity must be carefully monitored (Safaldin et al., 2024). With these applications in mind, motion detection serves as a bridge between traditional segmentation methods and the specific requirements of dynamic environments, especially those requiring continuous, real-time data processing. In real-time applications such as video surveillance and traffic monitoring, accurate segmentation is crucial, yet achieving it remains challenging due to the unpredictable nature of these environments. Conventional methods, such as edge detection and threshold-based segmentation, lack robustness in the presence of motion and struggle to maintain accuracy when applied to scenes with continuous movement or shifting backgrounds. For example, thresholding techniques are highly sensitive to changes in illumination, which can lead to false segmentation, while clustering methods often fail to accurately segment objects in complex, dynamic scenes due to their static nature. Moreover, edge-detection techniques, although effective in highlighting boundaries, can produce fragmented results in dynamic contexts, particularly when motion blurs the contours of objects. Given these challenges, there is a clear need for advanced segmentation techniques that can operate effectively in dynamic environments while minimizing resource demands. The current literature lacks a comprehensive solution that combines segmentation and motion detection within a lightweight, adaptable framework suitable for real-time applications. This study addresses this gap by proposing an integrated system that uses dynamic thresholding in conjunction with motion detection algorithms, aiming to enhance segmentation accuracy and reduce false detections. By addressing the limitations of conventional approaches, this research contributes to the development of robust, IoT-compatible systems capable of efficient and accurate image segmentation in dynamic, real-time environments (Chapel & Bouwmans, 2020; Mohammed, Basil, et al., 2024).

Despite advancements in segmentation techniques, achieving accurate results in dynamic scenes remains challenging. Traditional segmentation algorithms are typically designed for static environments, limiting their effectiveness in applications involving motion, such as video surveillance or traffic monitoring (Kabir et al., 2025). These algorithms often fail to adapt to changes in the scene, leading to decreased segmentation accuracy and higher computational demands. In real-time applications, this limitation presents significant issues: real-time surveillance systems, for instance, rely on the accurate detection and tracking of moving objects to ensure timely and appropriate responses. However, conventional segmentation methods, when used alone, are generally ill-suited for such tasks, as they cannot effectively account for continuous motion in a scene without additional processing layers. Consequently, there is a need for a solution that merges segmentation and motion detection to create a system capable of handling the challenges posed by dynamic environments (Amosa et al., 2023; Basil & Marhoon, 2024).

Moreover, systems that solely rely on motion detection often generate redundant data by recording entire scenes, regardless of the relevance of the activity within them. This approach increases storage, and processing demands and can lead to data overload, making it challenging to isolate meaningful information. Therefore, a targeted approach that focuses on segmenting relevant, motion-detected areas is essential for optimizing both processing efficiency and storage management. Addressing this gap is crucial for improving the performance of surveillance systems and ensuring that resources are allocated efficiently. In summary, the main problem driving this research is the lack of an adaptable, efficient segmentation solution for dynamic environments that minimizes computational load while maximizing accuracy and responsiveness (Garcia-Garcia et al., 2020).

Existing literature on image segmentation and motion detection highlights significant advancements but also reveals critical gaps, especially concerning the application of these techniques in dynamic environments. Traditional image segmentation algorithms, such as thresholding, clustering, and region-based methods, have proven effective for static scenes but often lack the adaptability needed for real-time applications involving motion. When applied in dynamic settings, these methods may experience reduced accuracy and efficiency due to their inability to distinguish between static and moving objects reliably. Consequently, these conventional approaches struggle to meet the demands of applications such as real-time video surveillance, autonomous systems, and traffic monitoring, where rapid and accurate object identification is essential (Mohammed, Marhoon, et al., 2024; Xu et al., 2024). However, while motion detection algorithms such as background subtraction, optical flow, and temporal differencing have been instrumental in tracking moving objects, they often generate excessive data by capturing entire scenes, even those without meaningful activity. This results in data redundancy, increased storage demands, and slower processing speeds. Furthermore, combining motion detection with segmentation remains underexplored in the context of creating a streamlined, autonomous system capable of processing only relevant areas within a dynamic scene. Most existing studies either focus on segmentation in static scenes or motion detection in isolation, leaving a gap in literature regarding integrated systems designed specifically for dynamic, real-time environments (Basil, Sabbar, et al., 2024). Another limitation lies in the applicability of current solutions in resource-constrained environments. Many segmentation and motion detection systems require significant computational power and are not portable, limiting their feasibility for small-scale applications or deployment in remote locations where resources are limited. Although recent studies have examined IoT-based surveillance, few have addressed the need for a compact, IoT-compatible solution that offers autonomous operation, portability, and real-time data access. These constraints point to a gap in the literature regarding affordable, scalable, and flexible surveillance systems that combine the strengths of motion detection and segmentation to optimize performance in dynamic settings (Mukalaf et al., 2023; Wang et al., 2023).

1.1 Literature review

Recent studies have explored the integration of segmentation with motion detection to address these challenges. For example, the work of (Ding et al., 2023)demonstrated a hybrid approach that combines optical flow with edge-detection algorithms to enhance segmentation accuracy in dynamic scenes. Another study by (Thakur & Mishra, 2024) proposed an adaptive background model that responds to scene changes, which showed promise in improving segmentation stability in challenging lighting conditions. Moreover, advances in IoT-based surveillance systems have led to the development of portable, self-contained units capable of capturing and processing video data on-site, making them highly suitable for applications requiring mobility and autonomy. However, while these developments have improved segmentation accuracy and real-time adaptability, they often lack a comprehensive, modular approach that allows for easy expansion into broader IoT ecosystems, which limits their scalability and applicability in diverse environments. Recent studies have attempted to address these challenges through the integration of motion detection algorithms, such as background subtraction and optical flow, with segmentation. However, these approaches also have limitations. For example, background subtraction relies on a stable model and is easily disrupted by lighting changes, while optical flow methods are computationally demanding, limiting their practicality in real-time scenarios (Zamalieva & Yilmaz, 2014).Furthermore, current motion detection techniques often produce high false positive rates in complex scenes, where background elements or minor movements are misinterpreted as significant motion. As a result, these methods often fail to maintain segmentation accuracy across varied and unpredictable conditions, making them unsuitable for resource-limited, real-time applications (Yazdi & Bouwmans, 2018).

However, while integrating motion detection with segmentation has improved performance in dynamic environments, these solutions often demand substantial computational resources, limiting their feasibility in small-scale, real-time applications. Most of these methods have been designed for environments with access to high-performance processing units and stable power supplies, which restricts their adaptability for remote or resource-constrained locations. As IoT technology advances, IoT-based surveillance systems have become more prevalent, with research showing that self-contained, portable systems capable of on-site processing are crucial for achieving efficient and adaptable surveillance in remote or decentralized environments. Recent IoT-compatible designs often employ embedded systems like the Raspberry Pi, which provide cost-effective processing without compromising functionality. However, while IoT-based approaches show promise, few studies have explored their integration with motion detection and image segmentation in a cohesive, modular design suitable for small-scale applications. The literature reveals a gap in solutions that effectively manage data to reduce redundancy and ensure efficient storage. As motion detection systems often record entire scenes, including static background areas, they generate large amounts of redundant data, leading to high storage requirements and data overload. Recent studies have examined cloud integration for data storage, emphasizing the advantages of remote data access, but many existing systems lack the modularity to selectively archive relevant data, instead storing entire video streams regardless of motion relevance. This oversight has implications for both data management and processing efficiency, particularly in resource-limited settings. A significant advancement in this area is seen in the work of (Wan et al., 2022), who proposed a cloud-based model for selective data storage based on motion relevance, reducing storage needs by archiving only footage with detected motion. This selective approach not only optimizes storage but also ensures that data retrieval and analysis focus on relevant areas, a feature that is particularly beneficial in surveillance contexts where continuous data capture can lead to data overload. Although promising, the implementation of such selective storage solutions remains limited in IoT-compatible designs, highlighting an area where further development is needed.

In the field of computer vision, moving object detection has become an essential task, applied in many diversified fields related to Closed-Circuit Television (CCTV) systems, high-speed road monitoring, and even sign language translation into text (Wahab et al., 2022). In applications with fixed cameras, several different techniques are applied at the first stage, separating the moving objects from the background and all essentially building a static scene model, often called the background, to detect the foreground or moving objects (Gowda et al., 2024).

1.2. The study contributions

This study proposes a novel approach to address these challenges by developing a prototype system that integrates motion detection with image segmentation to create an adaptable, real-time theft detection solution suitable for small-scale applications. The proposed system leverages a Raspberry Pi 3 as the primary processing unit, a compact, affordable, and highly portable option that enables deployment in various settings without extensive setup requirements. The system is equipped with a motion-detecting camera capable of monitoring a defined area and recording activity only when motion is detected. This setup minimizes redundant data capture, focusing resources on relevant activities and reducing computational complexity. Upon detecting motion, the system captures video data and securely uploads it to a cloud server, allowing for real-time analysis and future reference. The cloud-based repository not only facilitates secure archiving but also enables seamless data access, analysis, and scalability. This design also aligns with IoT principles, incorporating portability and modularity into the system's configuration. A battery-powered setup enhances the system's independence from external power sources, allowing it to operate autonomously in various environments, including remote or mobile locations. This adaptability is crucial for surveillance applications that require portability and flexibility, such as wildlife tracking, healthcare monitoring, and other contexts where traditional infrastructure is unavailable. The novelty of this approach lies in the integration of motion detection with segmentation within a flexible, self-contained, and IoT-compatible system. By focusing on active areas within the scene, the system increases segmentation accuracy and minimizes processing time, making it suitable for real-time applications. This research provides valuable insights into the design of efficient IoT surveillance systems, illustrating a framework that combines real-time motion detection, segmentation, and cloud-based data management. The system's modularity allows for future expansion, enabling integration into larger IoT ecosystems or adaptation for more complex surveillance needs.

The purpose of this study is to develop a practical, cost-effective, and adaptable theft detection system that can operate independently and securely archive data in real-time for small-scale applications. This research addresses the need for a flexible and portable surveillance solution that leverages IoT technologies, making it ideal for various environments where traditional security systems may be impractical. By designing a prototype based on the Raspberry Pi 3 with motion-detection capabilities and cloud storage, the study aims to provide a reliable option for remote monitoring, enhancing security and enabling real-time data access and analysis across diverse applications, such as patient monitoring in healthcare and wildlife tracking.

In this study, we adopted a design-based research approach to develop and evaluate a prototype theft detection system. The methodology involved configuring a Raspberry Pi 3 as the main processing unit, integrating it with a motion-detecting camera, and designing a custom application for detecting and recording motion within the monitored area. On detecting movement, the system automatically captures video data and securely transfers it to a cloud server, enabling real-time access and analysis. To enhance portability, we implemented a battery-powered configuration, allowing the system to operate independently of external power sources. This iterative design approach, combined with extensive testing in various environments, enabled us to optimize the system for diverse applications, ensuring flexibility, reliability, and alignment with IoT principles.

The value of this study lies in its contribution to both theory and practice in the field of IoT-based surveillance and security systems.

- Theoretical Contributions: This research advances the understanding of how small-scale, self-contained IoT surveillance systems can function autonomously while ensuring secure, real-time data transfer to cloud-based repositories. By integrating motion detection, cloud connectivity, and portability into a cohesive system, the study contributes to existing IoT frameworks, illustrating how these elements can be optimized for scalability and versatility in surveillance applications. This framework serves as a model for future studies seeking to develop adaptable and independent IoT solutions for real-time monitoring.
- Practical Contributions: Practically, this study offers an affordable, flexible, and portable surveillance solution that is ideal for small-scale applications in varied settings, from healthcare to wildlife tracking. The design principles demonstrated in this study can guide the development of cost-effective, IoT-compatible surveillance systems for real-time monitoring in environments where traditional infrastructure is lacking or impractical. By showcasing a modular, scalable system, this research provides a blueprint that can inform the development of future IoT-based security solutions tailored to specific monitoring needs.

This study aims to bridge the identified gaps in current literature by developing an innovative, integrated theft detection prototype that combines motion detection with image segmentation for real-time, small-scale surveillance applications. Using a Raspberry Pi 3 as the primary processing unit and a motion-detecting camera, the system captures activity within a monitored area and securely stores the data in a cloud-based repository, providing accessible and real-time data storage. Optimized for dynamic environments, the system improves segmentation accuracy and processing efficiency by focusing solely on active areas, thereby addressing the need for real-time adaptability. By targeting only motion-detected areas, the system also reduces data redundancy and storage demands, overcoming challenges related to data overload in existing solutions. Furthermore, as a battery-powered, IoT-compatible system, it ensures portability and autonomy, making it viable for deployment in diverse and even resource-constrained locations.

2. Proposed System Hardware and Software

This section will provide a comprehensive definition and explanation for the used hardware and software components such as Raspberry Pi 3 and Firebase cloud storage.

Raspberry Pi 3

The Raspberry Pi Foundation has created the Raspberry Pi 3, a single-board computer designed to support a variety of electronic projects and advance computer science education (Ariza & Baez, 2022). Compared to its predecessors, it offers improved performance thanks to a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor, 1 GB of RAM, integrated Bluetooth 4.1 and Wi-Fi, as exhibited in Fig. 2. This small, affordable and incredibly flexible gadget can be used for a wide range of tasks, from simple educational resources to complex IoT systems (Aldahoud et al., 2024). The Raspberry Pi 3 is often used in academic and research environments to deploy small embedded systems. and prototyping, allowing professionals and students to work on projects such as robotics, networked sensors, and home automation. Due to its open-source nature, strong community support, and compatibility with various operating systems, it has been widely adopted and is an invaluable tool for learning computer science principles and real-world engineering solutions (Mathe et al., 2024).

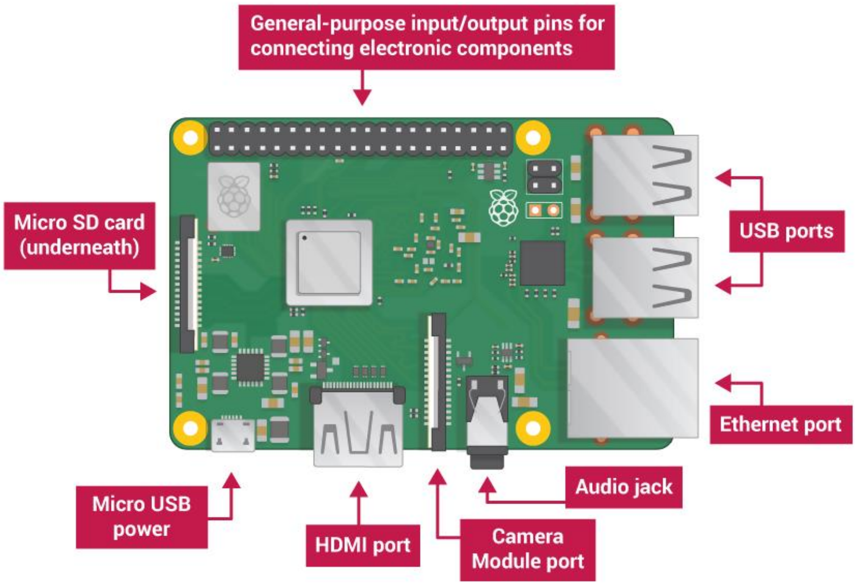

Fig. 1. Raspberry Pi 3 structure.

Firebase Cloud

Google developed Firebase Cloud, a product that includes a variety of independent and connected tools for building and improving applications. It has features that are a good support infrastructure for the app development process, like cloud real-time database, cloud storage, cloud messaging, analytics, Cloud Firestore, authentication, and more (Albertengo et al., 2020). Data housing and, to a greater extent, instantaneous user chaining are made possible by the combination of Firebase, which includes a real-time database, and Cloud Firestore. Google-powered Firebase Hosting is a web hosting service that offers quick and inexpensive web hosting. Even users of mobile devices can use their own identities, and Firebase Authentication supports a variety of user authentication methods. Furthermore, user-generated documents can be kept in cloud storage, and cloud messaging is the service that ensures productive user communication. Another service that guarantees proper user traffic is Firebase Cloud Messaging. With Cloud Functions, programmers can create backend code that can be expanded based on actions taken within the application. Because of these features, Firebase Cloud can be used in a variety of contexts, including the development of websites and mobile applications, IoT applications, and enterprise solutions. It also simplifies tasks by providing ready-to-use components for commonly used tasks rather than requiring the development to start from scratch (Singh et al., 2024).

3. Comprehensive System Architecture and Implementation Strategy

This study adopts a design-based research approach to develop and evaluate a theft detection system that integrates motion detection with image segmentation, specifically tailored for real-time, small-scale surveillance applications. The chosen methodology is grounded in practical considerations: it allows for iterative testing and refinement of the system, ensuring that it meets the unique demands of dynamic environments, where both accuracy and efficiency in detecting motion and segmenting images are critical.

The prototype system is built on a Raspberry Pi 3, which was selected for its balance between affordability, portability, and sufficient processing power to support the required image processing tasks. The Raspberry Pi serves as the primary processing unit, managing data acquisition and processing tasks while keeping overall costs low and enabling deployment in a range of environments. This choice reflects the system’s need for versatility and portability, particularly in remote or resource-constrained locations.

A motion-detecting camera is integrated into the system to capture visual data within a defined monitored area. The camera continuously scans for movement and triggers image capture when motion is detected. This feature is essential for reducing data redundancy and focusing the system’s processing resources on relevant segments, thereby increasing both efficiency and accuracy. The captured images provide visual information on detected activity, including the location, size, and movement patterns of objects in motion. From the displayed images, users can determine not only the presence of potential intruders or unauthorized movement but also the precise time and nature of the detected activity. This information is critical for applications that require reliable surveillance in real-time and the ability to store relevant video data for future reference or analysis.

The system is designed to upload detected motion images and videos to a cloud-based storage solution, chosen to ensure secure data archiving, real-time access, and the potential for future expansion. Cloud integration also facilitates data accessibility across various platforms, allowing for analysis even when the system operates autonomously in a remote location. This structure aligns with IoT principles, making the system both scalable and suitable for integration with larger IoT-based surveillance ecosystems.

The following paper considers the application of motion detection techniques in CCTV systems, which present a special case of constraints and challenges in the framework of background subtraction. These challenges are addressed through the following methodological considerations:

- **Autonomy and Adaptability:** The system shall run autonomously over a long period without any human intervention. It shall adapt to gradual or even sudden scene changes resulting from variations in illumination or the introduction of new static objects. In this respect, temporally adaptive background modelling shall be a requirement of the system so that there is a constant update of the background.
- **Filtering Natural Events:** It should be capable of filtering out natural events like the movement of tree leaves in the wind or the falling of raindrops, and compensate for camera instability while tracking moving objects. This requires algorithms that tell between relevant motion and environmental noise.
- **Resource Efficiency:** Real-time functionality in CCTV applications calls for making the system resource-efficient with respect to mainly minimizing power consumption, CPU load, and RAM usage. Keeping these considerations in check keeps the system viable for continuous operation.
- **AI-Powered Indoor Monitoring with Cloud Storage:** The system moves beyond simple motion detection and is equipped with an AI-powered monitoring system to secure the indoor environment against illegal access. On detecting unauthorized access, the video recording of the event will be captured and stored in Firebase Storage, securely storing the footage and hence ensuring its availability for evaluation at a later stage.\

- **Real-Time Alarm System and User Interface:** The system shall also provide a comprehensive solution for real-time monitoring and alarm services in its entirety. A user-friendly panel is developed to alert about the detected events of motion, send alarms, and allow control and download of the recorded footage from the Firebase Database. Such an interface would ensure that users are able to act against a probable threat within a very short time and manage their security footage efficiently.

It applies these cutting-edge techniques in a system using AI and cloud-based solutions to ensure robust monitoring capabilities with high efficiency for modern surveillance applications.

Proposed Architecture

This paper presents a security camera system developed to monitor a storage room. The system is fitted with motion detection capability to automatically trigger a video recording upon motion detection. The video is recorded for 14 seconds, from the moment when the motion is first detected. Subsequently, the recording gets uploaded into Firebase Cloud Storage, hence very secure. There will also be a user panel dashboard where the recorded videos can be managed, viewed, and downloaded by the users after authentication of the authorized entity with the proper permissions. This dashboard provides a user-friendly interface for efficiently dealing with the stored footage. Fig. 2 exhibits the detailed design of the proposed system, its different components, and its functionalities.

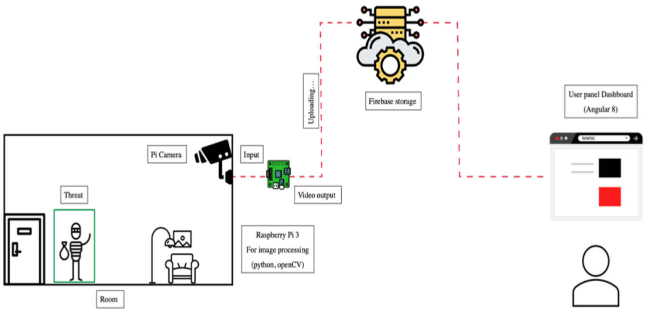

Fig. 2. The overall proposed system design

Building user panel

This section will integrate all components of the proposed system, from motion detection to displaying the moment of theft, into a simple, user-friendly interface. The user interface consists of three main Angular components and a module that unifies them. In more detail, the overall Angular application will include the following:

- The main component – app.component
- Sub-components:
- Authentication component – login.component
- User profile managing component – user.component
- Recordings component – recordings.component
- Modules:
- Main app module – app.module
- Routing module – app-routing.module
- Services:
- Authentication service – auth.service
- User managing service – user.service

The main part of the app will have HTML links for the login page and the recordings display, acting as the main place for other parts of the app. Also, this main part will handle the navigation, letting users move between different parts of the app. This is done by connecting the navigation settings to the main part (app.component), as shown in the example code below, which displays the app.component.ts file and how the navigation settings are integrated. The panels have been created as shown in Fig. 3.

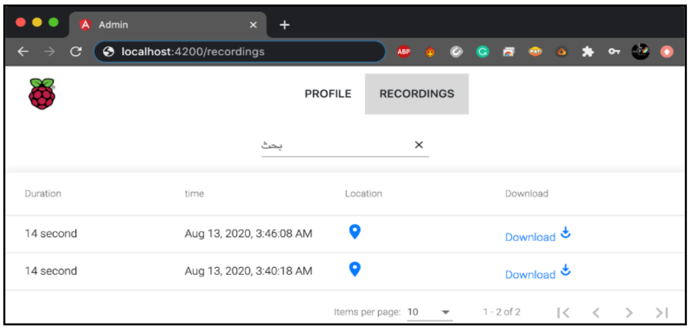

Fig. 3. Recordings management webpage

The previously presented Fig. 3 illustrates three fundamental columns, the function of each one can be interpreted as follows, the first shows the duration of the recording, the second column shows the time of recording this video and the third column shows the last location of the device when a motion occurred.

This is how the recordings component will look to manage and download the recordings, which is the most essential component in the panel. The next component is the Login component which contains the authentication system. In this article, the FireAuth module is used to create a smooth link between the Angular framework and the Firebase authentication system. The FireAuth module allows us to add different ways for users to sign in, like using Gmail or Facebook accounts, based on the app's needs. For this specific project, we decided to use the basic email and password method to manage access to the app. We chose to keep the login process simple and safe for users. The article includes details about how we set up the login page and the steps for authenticating users, which are shown in Fig. 4. The Fig. shows how users enter their login information, which Firebase checks to let them use the app's features.

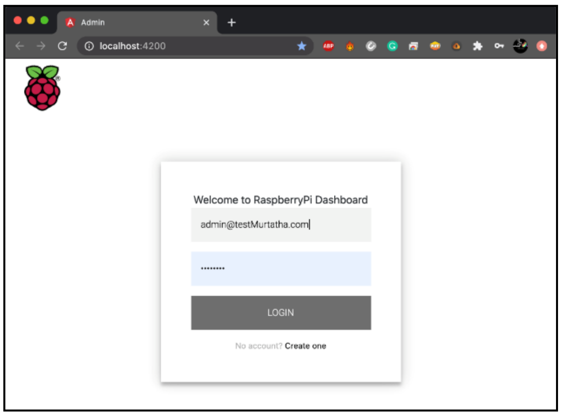

Fig. 4. The Authentication page

Motion detection Implementation

The method presented in this article was designed to improve efficiency and quickness, fitting the Raspberry Pi-3 CPU's processing power, which is presented in Fig. 5. This method aims to boost the frame rate to at least 26 frames per second (fps), which is needed for smooth video display and instant processing.

Fig. 5. The utilised Raspberry Pi-3 with the compatible camera

Furthermore, the method seeks to sharpen the difference between images, vital for precise motion tracking and examination. To reach these goals, we combined image division with a sophisticated thresholding method, specifically Otsu's thresholding. This mix has proven to yield better outcomes than other methods we tested, like the Image Segmentation & Image Subtraction Algorithm and the Moving Edge Detection Algorithm. The flexible threshold value in this approach allows for better adaptability and dependability, making it quicker and more effective. This approach involves finding the T value, which stands for the overall threshold, right after looking at the shades of Gray in the picture. In the next part, these shades of Gray are called F (for the brighter parts) and B (for the darker parts). Let us assume that the image is represented using L grey levels $\{0, 1, \dots, L-1\}$, denote by n_k number of pixels with grey level K and put $N = \sum_k n_k$. Then we can describe the probability distribution of grey levels as $\{k, p_k\}_{k=0}^{L-1}$ with $p_k = n_k/N$ and to calculate for a fixed value T, for $0 < T < L$, probability that a pixel belongs to foreground (F) or background (B) class as:

$$\pi_F = \sum_{k=0}^{T} p_k \quad \text{and} \quad \pi_B = \sum_{k=T+1}^{L-1} p_k \tag{1}$$

At this point, we can Fig. out the average and how much the numbers vary for the shades of gray in both the main subject (foreground) and the area around it (background), as well as for the whole picture, using the formula given below:

$$m_F = \sum_{k=0}^{T} kp_k\,, m_B = \sum_{k=T+1}^{L-1} kp_k \quad \text{and} \quad m = \pi_F m_F + \pi_B v m_B \tag{2}$$

$$\sigma_F^2 = \sum_{k=0}^{T} (k - m_F)^2\, p_k \quad \text{and} \quad \sigma_B^2 = \sum_{k=T+1}^{L-1} (k - m_B)^2 p_k \tag{3}$$

From here we get for a fixed value of T which are between and within variances, as presented in the following equation:

$$\sigma^2_{\text{between}}(T) = \pi_F(m_F - m)^2 + \pi_B(m_B - m)^2 \quad \text{and} \quad \sigma^2_{\text{within}}(T) = \pi_F\sigma_F^2 + \pi_B\sigma_B^2 \tag{4}$$

Finally, the value of T is determined via the maximization process. This chooses the value that guarantees an optimal result with regard to a certain criterion, such as maximizing accuracy, minimizing error, or optimizing some specific measure of performance. The value of the chosen T identifies this point at which the maximum of the criterion takes place and therefore stands for the most effective threshold regarding a given application or analysis, as follows:

$$\eta(T) = \frac{\sigma^2{}_{within}(T)}{\sigma^2{}_{within}(T)} \qquad i.e. T = \underset{0<T<L}{\arg max}\, \eta(T) \tag{5}$$

The advantage of this approach is that the global threshold value T is continuously updated with every background adjustment, hence adaptable to almost all light conditions. Image subtraction and segmentation techniques are usually considered two of the best methods for motion detection. Isolating the foreground from the rest of the stationary objects in the background is one of the prime goals. For such a purpose, the application is required to hold a part of the non-moving video to accurately recognise the background and then separate the foreground out of it. This background should be updated periodically in order to keep the scene active and real as well. It should specifically focus on the region of interest, which is actually the moving object, usually a small part of the frame.

Image Segmentation & Image Subtraction Algorithm

The segmentation and subtraction algorithm form an integral part of effective motion detection, bringing out moving objects from the background. These techniques segment the image so as to extract the region of interest-that is, usually a moving object-from the static background. The algorithm continuously updates the background to adapt to changing lighting conditions and other environmental changes. Sample results of the above approach are shown in Figs. 6 and 7 below. From these figures, can clearly identify how the algorithm segregates the moving objects of the scene and puts them in relief through the green bounding boxes. In the top image, the original frame indicates the moving object in the red circle. The bottom image shows the result of subtracting images, which suppresses the background and makes the moving object highly distinguishable. These results highlight the efficiency of image segmentation and subtraction techniques that help to improve the accuracy of motion detection, particularly under dynamic and real-time settings. This algorithm continuously updates the global threshold and background, therefore remaining adaptive under different conditions.

Fig. 6. Algorithm results highlighting the false positive detection areas.

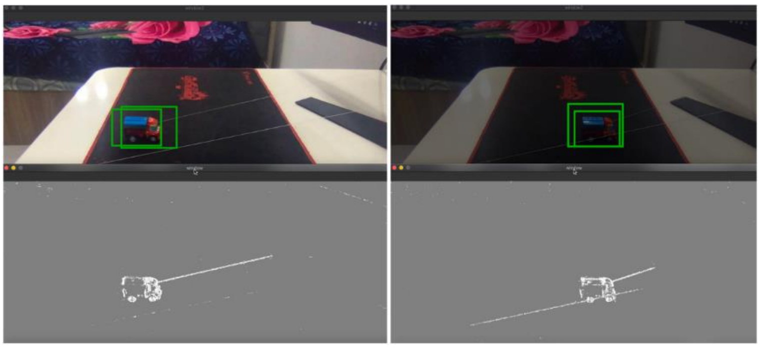

Fig. 7. The impact of different lighting conditions on the similar experimental demo video.

Comparing the results, the improvement in accuracy, speed, and overall performance is huge. Though the false positive alarms still occur, the percentage of accurately identifiable frames, i.e., those recognized as movement, rose to 96.31% of the total frames. This shows that the exact position of movements can be pretty accurately identified for different lighting conditions. What impresses me most is how the algorithm can work effectively with different lighting conditions. This is very useful when the light condition is low in places that have minimum lighting, such as storage rooms. When using an infrared camera or night vision camera in poorly lit areas, it is also expected to go well with the system. As presented, low light conditions also provide clear motion recognition and segmentation due to dynamic thresholding. The edges of the detected moving objects are distinct and well-defined. Fig. 8 below is the flowchart of the whole proposed algorithm.

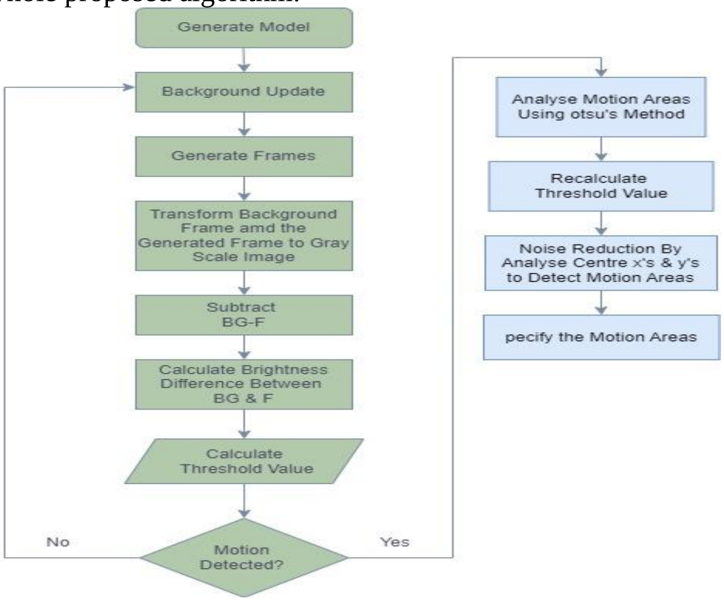

Fig. 8. Flowchart for the overall proposed algorithm.

4. Discussion

In this section, we evaluate the system's performance by comparing the actual results to the anticipated outcomes, focusing on accuracy, error rates, and adaptability. A key measure used in this analysis is "identifiable frames," defined as frames where motion is detected with a certainty of at least 20%. This threshold is implemented through a recognition filter applied to the image segmentation and subtraction algorithm described in previously. This algorithm employs spatial update methods and grey-level thresholding, functioning similarly to direct thresholding by classifying only frames with more than 20% certainty as motion. The 20% threshold was necessary because initial tests showed a high occurrence of false positives, where frames without motion were mistakenly identified as containing motion. Additionally, a temporal recognition filter was added to reduce false positives further, ensuring that a green rectangle (indicating detected motion) appears only if more than 12 consecutive frames display positive motion results. Without this filter, the system might mistakenly classify non-motion frames, like those affected by lighting changes or small, insignificant movements, as motion events.

The moving edge detection algorithm, specifically the Canny edge detector, was also evaluated with temporal and spatial update methods. While this algorithm naturally reduced false positives due to its focus on identifying changes in edges, it resulted in more false negatives where actual motion events were missed. Based on the results, we chose to disable both recognition filters for this engine to assess its raw performance. A comparative analysis between the two algorithms basic image segmentation and Canny edge detection revealed distinct strengths and weaknesses. The analysis involved comparing the number of identifiable frames, the percentage of these frames relative to the total, and the occurrence of false positives and negatives across the algorithms. The error rate was calculated by dividing the sum of false positives and negatives by the total number of frames. This approach follows recommendations from previous research, which emphasizes assigning greater weight to false positives, as minimizing false alarms is critical in motion detection systems. Previous studies, such as those by (Baruwal Chhetri et al., 2024) and (Mustafa et al., 2023), similarly highlighted the importance of prioritizing the reduction of false positives to avoid unnecessary alerts and enhance system reliability.

The performance comparison of the algorithms is summarized in Table 1, showing the identifiable frames, percentage accuracy, false positives, false negatives, and overall error rates. The standard image segmentation approach yielded an accuracy rate of 51.6% with an error rate of 0.35, while the Canny edge detection algorithm achieved a 48.4% accuracy with an error rate of 0.37. The enhanced image segmentation algorithm, which incorporated a dynamic threshold value, demonstrated superior performance, reaching an accuracy of 86.6% and an error rate of only 0.1. This improvement in the enhanced segmentation algorithm aligns with findings from (Kim et al., 2024), who noted that adaptive models adjusting to light changes and scene dynamics significantly reduce false detections. The dynamic threshold mechanism in our enhanced algorithm proved particularly effective across varying lighting conditions and rapid movements, thereby improving the reliability of the application in diverse environments.

Despite the advancements achieved with the enhanced segmentation method, limitations remain, as with any segmentation and motion detection technique. All tested algorithms produced false negatives in cases of non-critical motion, such as a flickering television or a moving bush outside a window. This limitation is consistent with previous research findings by (Atakishiyev et al., 2024), which discuss the challenges of discerning critical from non-critical motion in real-time systems. The importance of reducing false alarms is emphasized by researchers like (Kang et al., 2022), who argue that missing a motion event may be less critical since subsequent frames may capture the event, while repeated false alarms can undermine system reliability. The low error rate of the enhanced image segmentation algorithm supports its efficacy in distinguishing relevant motion, with fewer false positives and a high degree of accuracy across different testing scenarios.

Some researchers suggest assigning greater weight to false positives when evaluating performance, as avoiding false alarms in motion detection is critical. Missing a motion event is less critical because subsequent frames may capture the motion. The developed algorithm, which relies on image segmentation, produced the most accurate results among the three approaches tested. Implementing a dynamic threshold value enhanced the application's reliability across varying light conditions and rapid movements. However, a limitation of this approach, as well as the others, is that it may produce false negatives in scenarios involving non-critical motion, such as a flickering TV or a moving bush outside a window. The results of the comparison are shown as follows:

Table 1 - Comparison of speed, accuracy and performance

Algorithm	Identifiable frames	Percentage	False positives	False negatives	Performance
Image segmentat-ion	16	$16/31 \times 100\% = 51.6\%$	10	1	$(10+1)/31 = 0.35\ error\ rate$
Canny edge	30	$30/62 \times 100\% = 48.4\%$	14	3	$(14+3)/62 = 0.37\ error\ rate$
Enhanced image segmentat-ion	26	$26/30 \times 100\% = 86.6\%$	3	0	$(3+0)/30 = 0.1\ error\ rate$

The results of this study demonstrated that the developed theft detection system successfully fulfils its intended functions across various scenarios. The prototype effectively detected motion captured video data, and securely uploaded it to a cloud server for real-time analysis. Testing indicated that the system operated reliably in different environments, maintaining its functionality even in remote, battery-powered configurations. The integration of cloud storage provided secure archiving, enabling seamless access to recorded data for future reference. Additionally, the system’s modular design and compatibility with IoT frameworks allow for scalability and adaptability, making it suitable for broader applications in healthcare, wildlife monitoring, and other surveillance needs. This flexibility confirms the potential of the system as a foundational element for more complex IoT ecosystems.

5. Conclusions

In this paper, this study demonstrates that integrating dynamic thresholding with background subtraction provides an effective and efficient solution for real-time motion detection and segmentation, particularly suited to small-scale, security-focused applications. The enhanced segmentation approach, utilizing a dynamic threshold based on Otsu's algorithm, significantly improves detection accuracy by adjusting to varying lighting and environmental conditions, thus minimizing false positives and reducing the need for additional tracking algorithms. This adaptability allows the system to accurately capture even subtle motion events without excessive data processing or storage requirements, making it a powerful tool for IoT-compatible surveillance in resource-constrained environments. While some challenges with false negatives in non-critical motion scenarios remain, the algorithm reliably detects essential motion events and preserves sharpness in object representation. These advantages underscore the potential of the developed system to improve surveillance efficiency, particularly in applications where real-time accuracy and minimized resource use are essential. This study has significant practical and theoretical implications, particularly for the development of real-time surveillance systems in dynamic environments. Practically, the enhanced image segmentation and motion detection system offers a scalable, cost-effective solution for small-scale, IoT-based surveillance applications. By utilizing dynamic thresholding and selective data storage, the system minimizes resource consumption, making it ideal for deployment in resource-constrained settings, such as remote or confined spaces, where traditional surveillance solutions are less feasible. This approach can be especially beneficial in fields like healthcare, wildlife monitoring, and facility security, where efficient and accurate motion detection is essential for both safety and operational efficacy. Additionally, the system's ability to detect motion without constant human supervision highlights its potential for integration in broader IoT ecosystems, enabling automated, autonomous monitoring that could transform surveillance practices. Theoretically, this research contributes to the evolving body of knowledge on adaptive image segmentation within IoT-compatible frameworks. By demonstrating the effectiveness of dynamic thresholding in real-time environments, the study supports existing theories of adaptability in image processing and introduces new considerations for developing lightweight, efficient motion detection systems. The findings suggest that incorporating dynamic threshold mechanisms can bridge the gap between high accuracy and low resource demand, addressing a critical challenge in IoT surveillance systems. Moreover, this research lays the groundwork for future studies that may further optimize motion detection through machine learning or neural network integration, advancing the capabilities of IoT-based surveillance in increasingly complex scenarios.

References

Albertengo, G., Debele, F. G., Hassan, W., & Stramandino, D. (2020). On the performance of web services, google cloud messaging and firebase cloud messaging. *Digital Communications and Networks*, *6*(1), 31–37.

Aldahoud, A., Fezari, M., Al-Dahoud, A., & Aqel, D. (2024). A Systematic Review on Applying Machine Learning and Deep Learning on SBCs. *WSEAS Transactions on Information Science and Applications*, *21*, 272–281.

Amiriebrahimabadi, M., Rouhi, Z., & Mansouri, N. (2024). A Comprehensive Survey of Multi-Level Thresholding Segmentation Methods for Image Processing. *Archives of Computational Methods in Engineering*, *31*(6), 3647–3697. https://doi.org/10.1007/s11831-024-10093-8

Amosa, T. I., Sebastian, P., Izhar, L. I., Ibrahim, O., Ayinla, L. S., Bahashwan, A. A., Bala, A., & Samaila, Y. A. (2023). Multi-camera multi-object tracking: A review of current trends and future advances. *Neurocomputing*, *552*, 126558. https://doi.org/https://doi.org/10.1016/j.neucom.2023.126558

Ariza, J. A., & Baez, H. (2022). Understanding the role of single-board computers in engineering and computer science education: A systematic literature review. *Computer Applications in Engineering Education*, *30*(1), 304–329.

Atakishiyev, S., Salameh, M., Yao, H., & Goebel, R. (2024). Explainable artificial intelligence for autonomous driving: A comprehensive overview and field guide for future research directions. *IEEE Access*.

Baruwal Chhetri, M., Tariq, S., Singh, R., Jalalvand, F., Paris, C., & Nepal, S. (2024). Towards Human-AI Teaming to Mitigate Alert Fatigue in Security Operations Centres. *ACM Transactions on Internet Technology*, *24*(3), 1–22.

Basil, N., & Marhoon, H. M. (2023). Towards evaluation of the PID criteria based UAVs observation and tracking head within resizable selection by COA algorithm. *Results in Control and Optimization*, *12*, 100279.

Basil, N., & Marhoon, H. M. (2024). Correction to: selection and evaluation of FOPID criteria for the X-15 adaptive flight control system (AFCS) via Lyapunov candidates: Optimizing trade-offs and critical values using optimization algorithms. *E-Prime-Advances in Electrical Engineering, Electronics and Energy*, *8*, 100589.

Basil, N., Marhoon, H. M., Gokulakrishnan, S., & Buddhi, D. (2022). Jaya optimization algorithm implemented on a new novel design of 6-DOF AUV body: a case study. *Multimedia Tools and Applications*, 1–26.

Basil, N., Marhoon, H. M., Hayal, M. R., Elsayed, E. E., Nurhidayat, I., & Shah, M. A. (2024). Black-hole optimisation algorithm with FOPID-based automation intelligence photovoltaic system for voltage and power issues. *Australian Journal of Electrical and Electronics Engineering*, *21*(2), 115–127.

Basil, N., Sabbar, B. M., Marhoon, H. M., Mohammed, A. F., & Ma'arif, A. (2024). Systematic Review of Unmanned Aerial Vehicles Control: Challenges, Solutions, and Meta-Heuristic Optimization. *International Journal of Robotics and Control Systems*, *4*(4), 1794–1818.

Chapel, M.-N., & Bouwmans, T. (2020). Moving objects detection with a moving camera: A comprehensive review. *Computer Science Review*, *38*, 100310.

Ding, J., Zhang, Z., Yu, X., Zhao, X., & Yan, Z. (2023). A novel moving object detection algorithm based on robust image feature threshold segmentation with improved optical flow estimation. *Applied Sciences*, *13*(8), 4854.

Garcia-Garcia, B., Bouwmans, T., & Silva, A. J. R. (2020). Background subtraction in real applications: Challenges, current models and future directions. *Computer Science Review*, *35*, 100204.

Garea, S. A., & Das, S. (2024). Image Segmentation Methods: Overview, Challenges, and Future Directions. *2024 Seventh International Women in Data Science Conference at Prince Sultan University (WiDS PSU)*, 56–61. https://doi.org/10.1109/WiDS-PSU61003.2024.00026

Gowda, V. B., Gopalakrishna, M. T., Megha, J., & Mohankumar, S. (2024). Foreground segmentation network using transposed convolutional neural networks and up sampling for multiscale feature encoding. *Neural Networks*, *170*, 167–175.

Hassan, S., Mujtaba, G., Rajput, A., & Fatima, N. (2024). Multi-object tracking: a systematic literature review. *Multimedia Tools and Applications*, *83*(14), 43439–43492.

Kabir, M. M., Jim, J. R., & Istenes, Z. (2025). Terrain detection and segmentation for autonomous vehicle navigation: A state-of-the-art systematic review. *Information Fusion*, *113*, 102644.

Kang, H. J., Aw, K. L., & Lo, D. (2022). Detecting false alarms from automatic static analysis tools: How far are we? *Proceedings of the 44th International Conference on Software Engineering*, 698–709.

Kim, M.-J., Kim, S., Lee, B., & Kim, J. (2024). Enhancing Deep Learning-Based Segmentation Accuracy through Intensity Rendering and 3D Point Interpolation Techniques to Mitigate Sensor Variability. *Sensors*, *24*(14), 4475.

Lei, L., Yang, Q., Yang, L., Shen, T., Wang, R., & Fu, C. (2024). Deep learning implementation of image segmentation in agricultural applications: a comprehensive review. *Artificial Intelligence Review*, *57*(6), 149. https://doi.org/10.1007/s10462-024-10775-6

Mahmood, Z. (2024). Digital Image Processing: Advanced Technologies and Applications. In *Applied Sciences (Switzerland)* (Vol. 14, Issue 14). Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/app14146051

Mathe, S. E., Kondaveeti, H. K., Vappangi, S., Vanambathina, S. D., & Kumaravelu, N. K. (2024). A comprehensive review on applications of Raspberry Pi. *Computer Science Review*, *52*, 100636.

Mishra, S., Suman, S. K., & Roy, L. B. (2024). Automated Road Crack Classification Using A Novel Forest Optimization Algorithm For Otsu Thresholding And Hybrid Feature Extraction. *International Journal Of Advanced Technology And Engineering Exploration*, *11*(111), 219–242.

Mohammed, A. F., Basil, N., Abdulmaged, R. B., Marhoon, H. M., Ridha, H. M., Ma'arif, A., & Suwarno, I. (2024). Selection and Evaluation of Robotic Arm based Conveyor Belts (RACBs) Motions: NARMA (L2)-FO (ANFIS) PD-I based Jaya Optimization Algorithm. *International Journal of Robotics & Control Systems*, *4*(1).

Mohammed, A. F., Marhoon, H. M., Basil, N., & Ma'arif, A. (2024). A New Hybrid Intelligent Fractional Order Proportional Double Derivative+ Integral (FOPDD+ I) Controller with ANFIS Simulated on Automatic Voltage Regulator System. *International Journal of Robotics & Control Systems*, *4*(2).

Mukalaf, A. H., Marhoon, H. M., Suwarno, I., & Ma'arif, A. (2023). Design and Manufacturing of Smart Car Security System with IoT-Based Real-Time Tracking. *International Journal of Intelligent Systems and Applications in Engineering*, *11*(6s), 745–752.

Mustafa, F. E., Ahmed, I., Basit, A., Malik, S. H., Mahmood, A., & Ali, P. R. (2023). A review on effective alarm management systems for industrial process control: barriers and opportunities. *International Journal of Critical Infrastructure Protection*, *41*, 100599.

Pang, S., Thio, T. H. G., Siaw, F. L., Chen, M., & Xia, Y. (2024). Research on Improved Image Segmentation Algorithm Based on GrabCut. *Electronics (Switzerland)*, *13*(20). https://doi.org/10.3390/electronics13204068

Rayed, Md. E., Islam, S. M. S., Niha, S. I., Jim, J. R., Kabir, M. M., & Mridha, M. F. (2024). Deep learning for medical image segmentation: State-of-the-art advancements and challenges. *Informatics in Medicine Unlocked*, *47*, 101504. https://doi.org/https://doi.org/10.1016/j.imu.2024.101504

Safaldin, M., Zaghden, N., & Mejdoub, M. (2024). An Improved YOLOv8 to Detect Moving Objects. *IEEE Access*, *12*, 59782–59806. https://doi.org/10.1109/ACCESS.2024.3393835

Singh, K. U., Varshney, N., Gupta, P., Kumar, G., Singh, T., & Dogiwal, S. R. (2024). Mobile Application Control with Firebase Cloud Messaging. *International Conference On Innovative Computing And Communication*, 527–535.

Thakur, A., & Mishra, S. K. (2024). An in-depth evaluation of deep learning-enabled adaptive approaches for detecting obstacles using sensor-fused data in autonomous vehicles. *Engineering Applications of Artificial Intelligence*, *133*, 108550.

Wahab, F., Ullah, I., Shah, A., Khan, R. A., Choi, A., & Anwar, M. S. (2022). Design and implementation of real-time object detection system based on single-shoot detector and OpenCV. *Frontiers in Psychology*, *13*, 1039645.

Wan, S., Ding, S., & Chen, C. (2022). Edge computing enabled video segmentation for real-time traffic monitoring in internet of vehicles. *Pattern Recognition*, *121*, 108146.

Wang, B., Liu, J., Zhu, S., Xu, F., & Liu, C. (2023). A dual-input moving object detection method in remote sensing image sequences via temporal semantics. *Remote Sensing*, *15*(9), 2230.

Xu, Y., Quan, R., Xu, W., Huang, Y., Chen, X., & Liu, F. (2024). Advances in Medical Image Segmentation: A Comprehensive Review of Traditional, Deep Learning and Hybrid Approaches. *Bioengineering*, *11*(10), 1034.

Yazdi, M., & Bouwmans, T. (2018). New trends on moving object detection in video images captured by a moving camera: A survey. *Computer Science Review*, *28*, 157–177.

You, Y., Cao, J., & Zhou, W. (2020). A survey of change detection methods based on remote sensing images for multi-source and multi-objective scenarios. *Remote Sensing*, *12*(15), 2460.

Zamalieva, D., & Yilmaz, A. (2014). Background subtraction for the moving camera: A geometric approach. *Computer Vision and Image Understanding*, *127*, 73–85.