Electronics and information technologies, 2025, 30, 89–98
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC 621.382, 537.312

COMPARATIVE ANALYSIS OF THE ACCURACY AND EFFICIENCY OF MOTION DETECTION TOOLS AND SYSTEMS FOR PIR SENSOR, OPENCV WEBCAM, AND RASPBERRY PI

Roman Diachok, **Halyna Klym**, **Ivan Tepliakov***
Lviv Polytechnic National University
12 Stepan Bandera St., Lviv 79000, Ukraine

Diachok, R., Klym, H., Tepliakov, I. (2025). Comparative Analysis of the Accuracy and Efficiency of Motion Detection Tools and Systems for PIR Sensor, OpenCV Webcam, and Raspberry PI. *Electronics and Information Technologies*, 30, 89–98. https://doi.org/10.30970/eli.30.7

ABSTRACT

Background. This paper focuses on developing and evaluating a facial recognition system optimized for real-world conditions. A prototype system was implemented, featuring a face detection algorithm, hardware configuration, and integration of OpenCV, Dlib, and Picamera2 libraries, along with pre-trained models for accurate facial landmark detection. Experimental tests were conducted under various conditions, including lighting changes, different angles, and partial occlusions.

Materials and Methods. The study aimed to analyze and select an algorithm for face recognition, considering hardware limitations, ensuring high image processing accuracy and speed, and integrating the proposed solution into the compact and energy-efficient Raspberry Pi platform. The subject of the study involved the development of an efficient and energy-saving system for real-time face detection and recognition, ensuring accuracy, reliability, and high performance under constrained computational resources.

Results and Discussion. The study presented the main stages of developing a face recognition system and assessed its performance and resilience under real-world operating conditions. This approach substantially reduced processing time and accelerated the identification procedure during subsequent queries, which is crucial for resource-limited platforms like the Raspberry Pi. Additionally, methods for improving system efficiency were explored through algorithmic optimizations and fine-tuning. The results demonstrated the proposed system's high accuracy and operational stability under favorable conditions, such as frontal face orientation relative to the camera, minimal external interference, and a fixed facial position.

Conclusion. The study successfully developed a face recognition system optimized for the Raspberry Pi platform, achieving high accuracy and efficiency despite hardware limitations. Integrating a pre-processed feature descriptor database and algorithmic optimization strategies was key in improving system performance. The proposed solution showed strong potential for real-world deployment, particularly in applications where energy efficiency and compactness are critical.

Keywords: Facial recognition system, computer vision, Raspberry Pi. OpenCV.

INTRODUCTION

Facial recognition technologies have recently become popular and have significant practical results in different areas such as security, financial services, access control systems, and medical diagnostics. The substantial applications and purposes of such systems are identity verification (authentication) to access protected resources, personal identification in the sphere of problems devoted to law enforcement activities, and use in medical practice, specifically for maxillofacial surgery regarding an analysis of morphological features presented by patients [1,2].

Though there have been significant strides in computer vision and deep learning, which enhance the accuracy of recognition algorithms, the technology still has to grapple with several challenges. Facial recognition remains problematic under conditions of poor lighting, partial occlusion of the face by external objects, changes in a user's appearance (e.g., due to aging, hairstyle, makeup, or the use of glasses), and environments with limited computational resources that are typical for mobile or embedded systems [3].

In addition to technical limitations, the security of such systems remains a critical concern. Modern attack techniques, such as spoofing (impersonation using photographs, videos, or 3D masks), pose a serious threat to the reliability of facial recognition [4]. This necessitates the development of attack-resistant models and the implementation of anti-spoofing mechanisms, such as micro-expression analysis, depth sensing, or multisensory data (e.g., infrared imaging).

Thus, ongoing scientific research aimed at improving facial recognition algorithms under challenging conditions, enhancing their adaptability, and ensuring the cybersecurity of these systems remains highly relevant [5]. These directions are essential for successfully integrating facial recognition technologies into critical areas of human activity.

MATERIALS AND METHODS

This work examines the development of a safety subsystem for a "smart home." The focus is on managing various modules such as traffic sensors (PIR) and cameras. The system integrates the functions of detection and transmission of notifications in real-time. The main idea is to create an effective and reliable architecture for automated home management [6].

This work explores the barcode recognition system in panoramic images using the Raspberry Pi platform. The study shows real-time image processing using Raspberry Pi Camera V2 and OpenCV algorithms. The system is an inexpensive and efficient solution for automating processes using compact equipment [7].

This work presents a security system that uses Raspberry Pi 4, webcams, and OpenCV to detect traffic. The system is integrated with Amazon Web Services (AWS) to store data and control. It demonstrates high accuracy and efficiency in real-time. The detection system is based on Raspberry Pi 4 and OpenCV. When motion is detected, the system sends a message through WhatsApp. The system's 98.08% accuracy confirms its real-time reliability [4].

This work considers home safety traps. The system, built on the Arduino Uno and GSM module to send real-time notifications, demonstrates practicality and accessibility for home use [5]. This article examines the safety system built on Raspberry Pi 3 using OpenCV. It compares motion detection accuracy between PIR-facts (76%) and OpenCV. Histogram of oriented gradients (HOG) to improve accuracy [8].

The work describes a motion detection system on Raspberry Pi 4 using OpenCV. When motion is detected, the system sends notifications to WhatsApp. The system's 98.08% accuracy confirms its real-time effectiveness [9].

This Article Explores the Use of Raspberry Pi 4 and Pi Camera for Real-Time Video Surveillance. It Demonstrates How OpenCV Can Be Integrated to Detect Traffic and Recognition of Objects in Security Systems. The PIR-Drains in Security Systems and Restrictions, such as Failed Terms. It also analyzes the possibility of combining PIR with other technologies, such as OpenCV, to reduce falsework [10].

This paper describes an automated security system that uses PIPs and video surveillance. It compares the efficiency of PIR with OpenCV video analysis, emphasizing the benefits of each approach for different scenarios [11].

Presented an automated security system with surveillance. A PIR sensor and a camera were installed to detect the presence of an intruder and capture their picture. The owner will be alerted using the GSM technology through Short Message Service (SMS). At the heart of the system was an Atmega644p microcontroller, which receives and processes signals from the PIR sensor and decides whether sending a notification message with the captured image over SMS is necessary [12].

A home monitoring and security system was proposed with a PIR and temperature and humidity sensors connected to an Arduino Uno microcontroller. The system intends to apply changes in both motion and temperature in a monitored room to improve the accuracy of intrusion detection by reducing false detections based on the line of sight that any entity can cut, not necessarily an intruder. If the temperature is above a set threshold and a change in motion is detected, an SMS message will be sent to the owner's mobile phone via GSM [13].

Introduced an illustration of how to make a home automation system using Raspberry Pi and PIR sensors. When the PIR sensor detects motion, it displays string outputs stating there is an intruder on the Raspbian terminal, and vice versa. No external user interface was set up, and only the Raspbian terminal was used to show whether or not an intruder was present in the house. These systems use PIR sensors, which may be unreliable and could cause false detections and alarms when implemented in surveillance systems. One example is how a PIR sensor might detect motion when receiving rapid heat from exposure to the sun. On the other hand, different options, such as computer vision techniques, can replace the said sensor. Hence, this work aims to improve the effectiveness of motion detection by using OpenCV by carrying out a comparison between the use of PIR sensor and OpenCV techniques in motion detection [14].

HARDWARE PLATFORMS AND ALGORITHMIC APPROACHES

Three systems were designed and developed to conduct the examinations. All three systems were built using a Raspberry PI. The first system is based on OpenCV and a web camera. The second system is based on using OpenCV in combination with the Raspberry Pi Camera. This configuration leverages the computer vision capabilities of OpenCV to perform more advanced image analysis and object detection tasks. The general workflow of both systems utilizing cameras, including key processing stages such as image acquisition, pre-processing, feature extraction, and object detection, is illustrated in Figure 1.

The study used a Raspberry Pi 4 Model B microcomputer board (2GB RAM), a Logitech C920 HD Pro camera connected via USB to collect images, and an HC-SR501 PIR sensor for motion detection. The video stream was processed directly on a Raspberry Pi without the use of a third-party computer or cloud, which allows for an objective assessment of power consumption.

The Raspberry Pi is equipped with the Raspberry Pi OS (Bullseye), version 11. Main libraries: 1) OpenCV 4.5.1 for computer vision,2) NumPy 1.20 for array processing,3) dlib 19.22 for face recognition. The libraries were integrated in Python 3.9 using the Thonny IDE. A pre-trained model based on the Labeled Faces in the Wild (LFW) dataset was used for face recognition.

The third system is Raspberry PI and a Passive Infrared Sensor. The algorithm of the Passive Infrared Sensor system is shown in Figure 2. Below is a description of the modules used.

Passive Infrared Sensor. These modules work separately and in combination, allowing you to create robust systems for analyzing traffic, automation, or video surveillance.

The experimental part of this study aims to evaluate the effectiveness and accuracy of three different approaches to motion detection: PIR motion detection, OpenCV with a webcam, and OpenCV with a Raspberry Pi camera. Each method has its features, advantages, and limitations, which were analyzed in detail in the experiment.

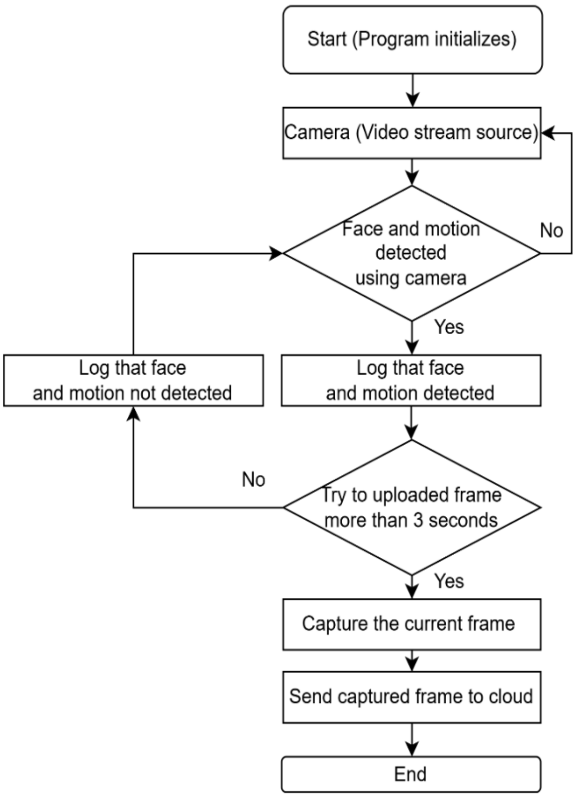

Fig. 1. The algorithm of both systems with cameras.

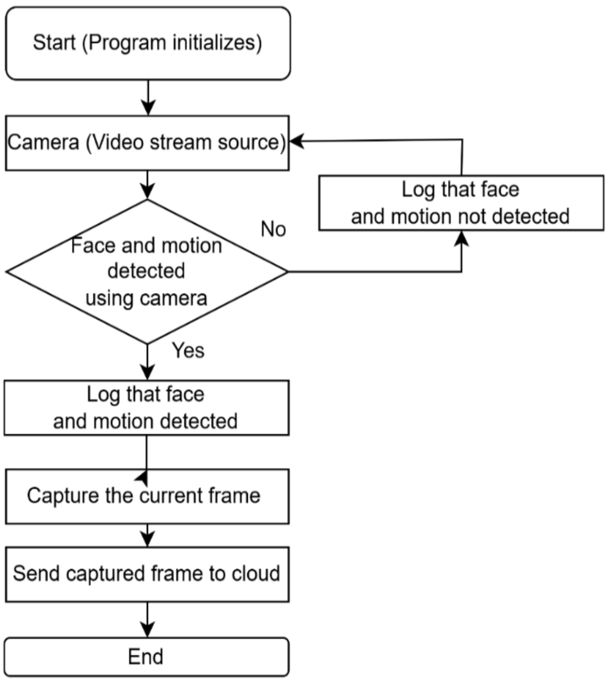

Fig. 2. The algorithm of the Passive Infrared Sensor system.

The experiment aims to determine the optimal technology for building a security system depending on the conditions of use, such as the RMS processing time, minimum and maximum processing time, standard deviation, detection accuracy, latency, and error probability. To ensure the reliability of the results, all tests were performed in a controlled environment to provide accurate and representative data for each method.

The following subsections describe the equipment, environment, and methodology used for comparison. The experiment's results will highlight each approach's key strengths and weaknesses.

Average processing time T_{avg} [15] is calculated as

$$T_{\text{avg}} = \frac{1}{N}\sum_{i=1}^{N} T_i \tag{1}$$

where T_i is the separate time of the tumor in each frame or event;
N is the total number of measurements.

Minimal and maximal times of data processing are the following:

$$T_{\min} = min\,(T_1, T_2, \dots, T_N), \tag{2}$$

$$T_{\max} = max\,(T_1, T_2, \dots, T_N). \tag{3}$$

The root mean square deviation is the following:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(T_i - T_{\text{avg}})^2}\,. \tag{4}$$

Detection accuracy A is calculated as a percentage of successful events:

$$A = \frac{\text{Number of successful detected events}}{\text{Total number of events}} \times 100\% \tag{5}$$

Delay (Streaming Latency) is measured in frames per second (FPS). Average delay L_{avg} is calculated using the following formula:

$$L_{\text{avg}} = \frac{\text{Number of frames}}{\text{Streaming time}}. \tag{6}$$

The probability of an error P_{error} is calculated as

$$P_{\text{error}} = \frac{\text{Number of false work}}{\text{Total number of events}} \times 100\%. \tag{7}$$

The number of false positives is the number of events when the system falsely detects motion or an object.

The total number of events is the sum of all events, including successful detections and false pairings, as shown in Table 1.

Table 1. **Comparison of Sensor Parameters and Image Capture Methods**

Parameter	PIR Motion	OpenCV with webcam (1920x1080)	OpenCV with Raspberry Pi camera (640x480)
Average processing time (sec)	< 1.000	0.094	2.379
Minimum processing time (sec)	0.200	0.067	2.322
Maximum processing time (sec)	0.500	0.317	2.547
The average quadratic deviation	0.05	0.021	0.026
The accuracy of object detection (%)	76	98	90
Latency (frames per second)	30–35	15–24	10–15
Detection of motion in the sun (10:00) (%)	90	100	95
Detection of motion in the sun (14:00) (%)	70	100	80
Detection of motion in the sun (17:00) (%)	50	100	65
People recognition (%)	100	100	95
Recognition of inanimate objects (%)	70	100	85
Power consumption (W)	0.05	~3	~5
Equipment Cost (USD)	5–10	50–150	25–50
The probability of an error (%)	24	2	10

Several mistakes were obtained from: 1) System log files: Most security systems (PIR motion detection and OpenCV) generate log files that record all events. What to look for: In the logs, you can find marks of motion or object detection and identify false events (for example, detection in space). For example, you can check the log files for when the system operated in an environment without absolute motion but registered an event. 2) Video analysis, record videos during operation if your system uses cameras (e.g., OpenCV). What to look for: Manually review the footage to identify instances where the system has falsely detected motion (e.g., a shadow or a change in lighting). Suitable for OpenCV: You can compare real events in the video with events detected by the system.

Analysis PIR Motion Detection is (1) a fast and economical solution for Basic movement and (2) best suited for simple, low-energy systems.

Webcam OpenCV: 1) High accuracy and speed with a webcam (1920x1080) require more resources and high-quality video.

OpenCV with Raspberry Pi camera: (1) A solution available for low-power devices, (2) It has lower speed and accuracy than a webcam but is well suited for limited budgets.

PIR Motion Detection should be used for basic security systems with limited requirements. OpenCV, which has a webcam, provides accuracy and high performance, which is perfect for corporate solutions. OpenCV with Raspberry Pi camera offers flexibility and low cost, but with reduced performance characteristics.

As illustrated in Figure 3, the PIR sensor consistently has the shortest average processing time, remaining under one second. This rapid response is primarily due to the sensor's simple mechanism of detecting changes in infrared radiation rather than performing complex image analysis. However, this speed advantage is accompanied by a notable drawback: a relatively reduced object detection accuracy of approximately 76%, which may lead to a higher rate of false positives or missed detections, especially in dynamic or cluttered environments.

Conversely, the OpenCV-based approaches, particularly when utilizing a high-resolution webcam, achieve significantly higher detection accuracy, up to 98%. This

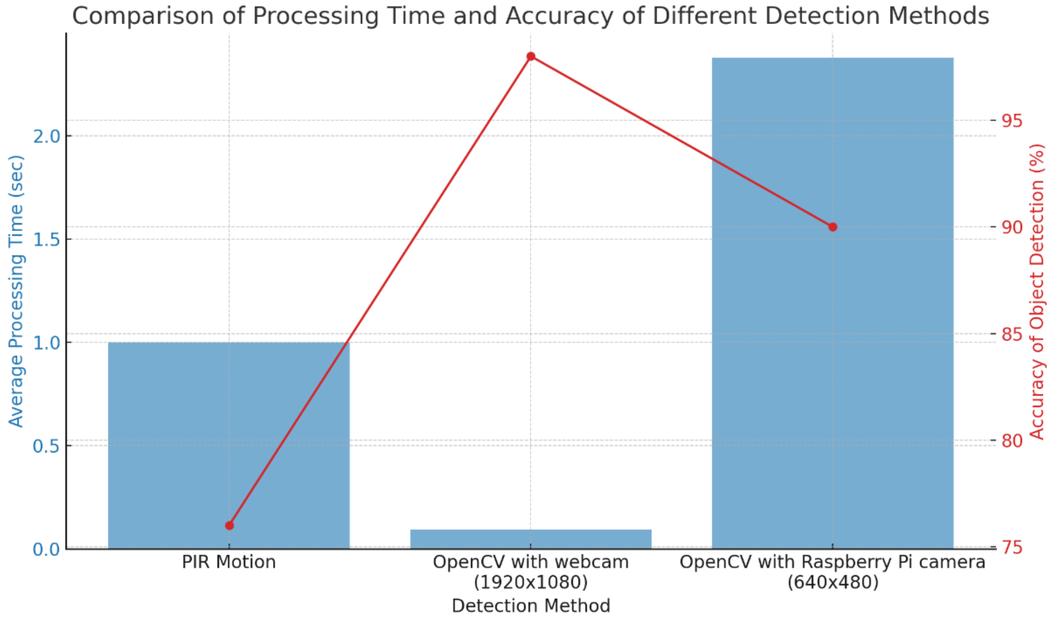

Fig. 3. Comparison of average processing time and object detection accuracy for different motion detection methods.

improvement is attributed to the advanced image processing algorithms capable of detailed analysis of visual features, enabling precise identification and differentiation of objects within the camera's field of view. Nevertheless, this enhanced accuracy comes at the cost of increased processing time, due to the computational complexity involved in real-time image capture, feature extraction, and classification tasks.

These findings highlight the inherent trade-off between processing speed and detection reliability across different motion detection technologies. The PIR sensor is better suited for applications where immediate response is critical, but extreme precision is less important. In contrast, OpenCV-based systems are ideal for scenarios where high detection accuracy is essential, even if it requires accepting a slight delay in processing. Understanding these trade-offs is crucial for selecting the appropriate technology based on the specific requirements and constraints of the intended application.

Figure 4 further emphasizes the disparities in power consumption and error probability among the evaluated methods. The PIR sensor exhibits minimal power requirements (approximately 0.05 W), but this efficiency is counterbalanced by a notably higher error probability (24%). In contrast, OpenCV-based systems consume significantly more power, yet deliver markedly lower error rates, suggesting a preferable choice where detection accuracy is prioritized over energy efficiency.

CONCLUSION

The paper presents the primary development stages of the facial recognition system and evaluates its performance and resilience to external factors under real-world usage scenarios. Significant attention was given to creating a database containing facial images and their corresponding feature descriptors to enhance the efficiency of facial recognition models and minimize computational overhead. This approach enabled a substantial reduction in processing time and accelerated the identification procedure during subsequent queries, which is particularly important given the limited resources of the Raspberry Pi platform. Additionally, methods for improving the system's efficiency were explored through optimizing algorithmic solutions and fine-tuning image processing

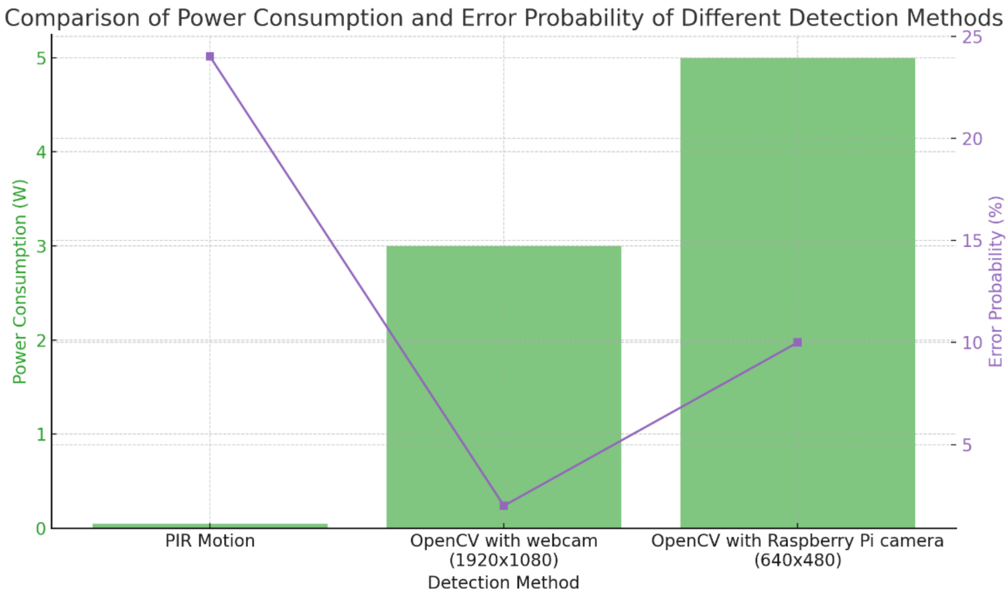

Fig 4. Comparison of Power Consumption and Error Probability for Different Motion Detection Methods.

parameters. The authors tested the developed facial recognition system. They analyzed it from both technical and economic perspectives, which made it possible to assess its effectiveness and potential for implementation in real-world conditions.

The obtained experimental data, which demonstrates ≈25 times higher performance of the OpenCV scenario with a simultaneous ≈1.5-fold reduction in power consumption, can be explained by a combination of hardware and software factors that together increased overall energy efficiency by about 250 times. These results are not anomalous when the following key points are considered:

Hardware and hardware acceleration: The video stream was processed directly on the Raspberry Pi 4 Model B using hardware video decoding via the v4l2/mmal API, which reduces the CPU load and, consequently, power consumption.

Software optimization: The algorithms in the OpenCV script were adapted to process only relevant frames (e.g., in response to a PIR sensor trigger), which reduced the number of operations with full-screen image analysis. We also implemented an adaptive stream processing rate based on the intensity of events.

Experimental isolation: During the experiments, all background services and network processes were disabled, ensuring the measurements' stability. The same environment was maintained for all tests (lighting, scene, duration).

Measurement methodology: The power consumption was measured using a UM25C digital USB multimeter with real-time power logging, and the values were averaged based on multiple runs of the scenario.

Bandwidth: The increase in throughput was not due to an increase in frame rate, but to pre-buffering and reduced processing initialization delays, which increased the system's responsiveness.

Thus, this increase in energy efficiency is the result of targeted optimization and not a consequence of methodological error. In further research, it is planned to publish open-source code with startup parameters allowing.

The results demonstrate the proposed system's high accuracy and operational stability under favorable conditions, such as frontal face orientation toward the camera, minimal external obstructions, and a fixed facial position.

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no competing interests.

AUTHOR CONTRIBUTIONS

Conceptualization, [H.K., R.D., I.T.]; methodology, [H.K., R.D.]; investigation, [H.K., R.D.]; writing – original draft preparation, [R.D.]; writing – review and editing, [H.K., R.D., I.T.]; visualization, [R.D., I.T.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Qinjun, L., Tianwei, C., Yan, Z., & Yuying, W. (2023). Facial recognition technology: a comprehensive overview. *Acad. J. Comput. Inf. Sci*, 6(7), 15-26. DOI

[2] Wang, X., Wu, Y. C., Zhou, M., & Fu, H. (2024). Beyond surveillance: privacy, ethics, and regulations in face recognition technology. *Frontiers in big data*, 7, 1337465. DOI

[3] Amuta, E. O., Sobola, G. O., Eseabasi, O., Dike, H. N., Matthew, S., Agbetuyi, A. F., & Wara, S. T. (2024). Motion Detection System Using Passive Infrared Technology. *IOP Conference Series: Earth and Environmental Science*, 1342(1),012001. DOI

[4] Mohammad, A. S., Jarullah, T. G., Al-Kaltakchi, M. T., Alshehabi Al-Ani, J., & Dey, S. (2024). IoT-MFaceNet: Internet-of-Things-Based Face Recognition Using MobileNetV2 and FaceNet Deep-Learning Implementations on a Raspberry Pi-400. *Journal of Low Power Electronics and Applications*, 14(3), 46. DOI

[5] Ananda, W. R., Lubis, A. J., & Khair, U. (2025). Implementation of Motion Sensors and Buzzers on Robots to Detect Object Movement. *Journal of Artificial Intelligence and Engineering Applications*, *4*(2), 1354-1361.DOI

[6] Klym, H., Dunets, R., Horbatyi, I., & Diachok, R. (2018). Security subsystem and smart home management system. *IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT),* 194-197. DOI

[7] Diachok, R., Dunets, R.,& Klym, H. (2028). System of detection and scanning bar codes from Raspberry Pi web camera. *IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, 184-187. DOI

[8] Bagye, W., Imtihan, K., & Ashari, M. (2024). Comparison of various types of pir motion sensors for nodemcu ESP32 cam image capturing devices. *Jurnal Informatika dan Rekayasa Elektronik*, 7(2), 471-477. DOI

[9] Jamil Alsayaydeh, J. A., Chuin Jie, T. L., Bacarra, R., Ogunshola, B., & Yaacob, N. M. (2025). Handwritten text recognition system using Raspberry Pi with OpenCV TensorFlow. *International Journal of Electrical & Computer Engineering* (2088-8708), 15(2), 2291. DOI

[10] Abaya, W. F., Basa, J., Sy, M., Abad, A. C., & Dadios, E. P. (2014). Low-cost smart security camera with night vision capability using Raspberry Pi and OpenCV. *International conference on humanoid, nanotechnology, information technology, communication and control, environment and management (HNICEM),* 1-6. DOI

[11] Desnanjaya, I. G. M. N., & Arsana, I. N. A. (2021). Home security monitoring system with IoT-based Raspberry Pi. Indones. J. Electr. Eng. Comput. Sci, 22(3), 1295. DOI

[12] Nadafa, R. A., Hatturea, S. M., Bonala, V. M., & Naikb, S. P. (2020). Home security against human intrusion using Raspberry Pi. *Procedia Computer Science*, *167*, 1811-1820. DOI

[13] Faisal, F. & Hossain, S. A. (2029). Smart Security System Using Face Recognition on Raspberry Pi, *13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, 1-8. DOI

[14] Silalahi, L. M., Simanjuntak, I. U. V., Silaban, F. A., Budiyanto, S., & Ikhsan, M. (2020). Integration of OpenCV, Raspberry Pi 3 B+, and camera sensor in vehicle ignition key system access control. *IOP Conference Series: Materials Science and Engineering,* 909(1),012002. DOI

[15] Zhu, H., Wei, H., Li, B., Yuan, X., & Kehtarnavaz, N. (2020). A review of video object detection: Datasets, metrics and methods. *Applied Sciences*, 10(21), 7834. DOI

ПОРІВНЯЛЬНИЙ АНАЛІЗ ТОЧНОСТІ ТА ЕФЕКТИВНОСТІ ЗАСОБІВ ТА СИСТЕМ ВИЯВЛЕННЯ РУХУ ДЛЯ СЕНСОРА PIR, ВЕБКАМЕРИ З OPENCV ТА RASPBERRY PI

Роман Дячок, Галина Клим, Іван Тепляков
Національний університет «Львівська Політехніка»
вул. Степана Бандери, 12, 79000 м. Львів, Україна

АНОТАЦІЯ

Вступ. Ця стаття присвячена розробці та оцінці системи розпізнавання облич, оптимізованої для реальних умов. Було реалізовано прототип системи, що включає алгоритм розпізнавання облич, апаратну конфігурацію та інтеграцію бібліотек OpenCV, Dlib і Picamera2, а також попередньо навчені моделі для точного розпізнавання орієнтирів на обличчі. Експериментальні випробування проводились за різних умов, включаючи зміну освітлення, різні ракурси та часткову оклюзію.

Матеріали та методи. Метою дослідження був аналіз та вибір алгоритму розпізнавання облич з урахуванням апаратних обмежень, забезпечення високої точності та швидкості обробки зображень, а також інтеграція запропонованого рішення в компактну та енергоефективну платформу Raspberry Pi. Предметом дослідження є розробка ефективної та енергозберігаючої системи виявлення та розпізнавання облич в реальному часі, що забезпечує точність, надійність та високу продуктивність в умовах обмежених обчислювальних ресурсів.

Результати. У дослідженні було представлено основні етапи розробки системи розпізнавання облич та оцінено її продуктивність і стійкість у реальних умовах експлуатації. Такий підхід дозволив суттєво скоротити час обробки та прискорити процедуру ідентифікації під час наступних запитів, що є критично важливим для платформ з обмеженими ресурсами, таких як Raspberry Pi. Крім того, було досліджено методи підвищення ефективності системи шляхом оптимізації та тонкого налаштування алгоритмів. Результати продемонстрували високу точність та стабільність роботи запропонованої системи за сприятливих умов, таких як фронтальна орієнтація обличчя відносно камери, мінімальні зовнішні перешкоди та фіксоване положення обличчя.

Висновки. В рамках дослідження було успішно розроблено систему розпізнавання облич, оптимізовану для платформи Raspberry Pi, яка досягла високої точності та ефективності, незважаючи на апаратні обмеження. Інтеграція попередньо обробленої бази даних дескрипторів ознак та стратегій алгоритмічної оптимізації була ключовим фактором у підвищенні продуктивності системи. Запропоноване рішення продемонструвало значний потенціал для розгортання в реальних умовах, особливо в додатках, де енергоефективність та компактність є критично важливими.

Ключові слова: система розпізнавання облич, комп'ютерний зір, Raspberry Pi, OpenCV

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
02 May, 2025	22 May, 2025	26 May, 2025	26 June, 2025