ADVANCES IN CYBER-PHYSICAL SYSTEMS
Vol. 10, No. 1, 2025

EFFICIENCY AND ACCURACY: COMPARISON OF PIR, OPENCV WITH A WEBCAM, AND RASPBERRY PI

Roman Diachok[1], Ivan Tepliako[1], Myroslava Lapko[1], Anatoli I. Popov[2]

[1] *Lviv Polytechnic National University, 12, S. Bandery str., Lviv, 79013, Ukraine,*
[2] *Institute of Solid State Physics University of Latvia, 8, Kengaraga, Riga, LV-1063, Latvia*
Authors' e-mails: roman.v.diachok@lpnu.ua, ivan.y.tepliakov@lpnu.ua, myroslava.lapko.mkisk.2023@lpnu.ua, popov@latnet.lv

https://doi.org/10.23939/acps2025.01.077

Submitted on 10.03.2025

***Abstract*: This paper is dedicated to developing and evaluating the facial recognition system, focusing on its effectiveness and operational reliability under real-world conditions. The choice of the Raspberry Pi hardware platform for implementing the system has been justified by its capability to process video streams in real time, as well as its compatibility with the high-quality Raspberry Pi Camera V2, which enables the acquisition of images with sufficient resolution for the proper functioning of computer vision algorithms. The implementation of a prototype of the studied system has been presented, encompassing the development of a face detection algorithm, hardware configuration, integration of the OpenCV, Dlib, and Picamera2 libraries, as well as the use of pre-trained models for accurate detection of facial key points. The developed algorithm has been adapted and optimized to consider the limited hardware resources of the Raspberry Pi platform, ensuring high accuracy and real-time image processing performance.**

***Index terms*: facial recognition system, computer vision, Raspberry Pi. OpenCV.**

I. INTRODUCTION

Facial recognition technologies have become widely adopted in the modern world and demonstrate significant practical value across numerous domains, including security, financial services, access control systems, and medical diagnostics [1]. The primary applications of such systems include identity verification (authentication) for granting access to protected resources, personal identification in law enforcement activities [2], and usage in medical practice, particularly in maxillofacial surgery, for analyzing the morphological features of patients' faces.

Despite significant advances in computer vision and deep learning, which have helped improve the accuracy of recognition algorithms, this technology still faces 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.

II. LITERATURE REVIEW AND PROBLEM STATEMENT

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 [1].

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 inexpensive and efficient for automating processes using compact equipment [2].

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 examines the implementation of home safety traps, focusing on an automated alert system designed for enhanced security and quick response. The proposed solution is built on the Arduino Uno platform, combined with a GSM module to send real-time notifications when an intrusion or hazardous event is detected. The system integrates sensors that monitor environmental changes, such as motion detection, door / window openings, or gas leaks. Upon detecting a potential threat, the GSM module triggers an immediate alert to the user's mobile device, ensuring rapid awareness and action. This approach enhances remote monitoring capabilities, making it a practical and accessible solution for everyday home security needs. [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 [6].

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. The system utilizes computer vision techniques and real-time processing to ensure efficient motion detection. Implemented on Raspberry Pi 4, it leverages OpenCV for image processing, allowing it to detect movement with high precision. Upon detecting motion, the system triggers an automated notification process, instantly informing users via WhatsApp. This functionality enhances security and monitoring applications, making it a reliable solution for real-time surveillance [7].

The project aims to create an autonomous car based on Raspberry Pi, using OpenCV for computer vision and NVIDIA's neural network to analyze the traffic situation. The camera captures the environment, OpenCV processes the image, and CNN predicts the movement. Raspberry Pi controls the motors, responding to route changes in real time. This approach allows for developing an effective automatic driving system on affordable hardware [8].

This paper presents an automated security system integrating Passive Infrared Sensors (PIRs) and video surveillance. The study compares the efficiency of PIR-based motion detection with OpenCV-powered video analysis, highlighting the advantages and limitations of each approach in various operational environments. PIR sensors offer a fast and energy-efficient solution, making them ideal for low-power security systems. However, their primary drawback lies in limited accuracy, particularly in distinguishing between human movement and environmental changes, such as temperature fluctuations [9].

An automated security system with surveillance is presented. 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 [10].

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 [11].

An illustration of how to make a home automation system using a Raspberry Pi and PIR sensors is introduced. 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 comparing the use of PIR sensor and OpenCV techniques in motion detection [12].

III. SCOPE OF WORK AND OBJECTIVES

The primary objective of this study is to conduct a comprehensive comparative analysis of the efficiency and accuracy of three motion and facial recognition systems: A Passive Infrared (PIR) sensor, an OpenCV-based solution utilizing a conventional webcam, and a Raspberry Pi platform integrated with a facial recognition algorithm. The research aims to evaluate the strengths and limitations of each system in terms of detection accuracy, processing latency, computational resource usage, and adaptability to dynamic environmental conditions. A significant part of the study is devoted to assessing the performance of these technologies under diverse real-world scenarios, including variations in ambient lighting, facial occlusions, different angles of observation, and the presence of accessories such as glasses or masks.

Furthermore, the research explores developing and integrating a facial image database with precomputed feature descriptors to enhance recognition speed and system responsiveness, particularly on resource-constrained platforms like Raspberry Pi. The study also investigates algorithmic optimizations and image preprocessing techniques to improve detection robustness and reduce false positive and false negative rates.

Ultimately, this research provides a data-driven foundation for selecting and deploying suitable recognition technologies in practical applications such as intelligent surveillance, automated access control, and low-cost embedded security systems. By analyzing both the technical and economic feasibility of each approach, the study contributes to informed decision-making in designing and implementing efficient, reliable, and scalable recognition systems.

IV. RESULTS AND DISCUSSIONS

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 OpenCV and a Raspberry PI camera. The algorithm of both systems with cameras is shown in Fig. 1.

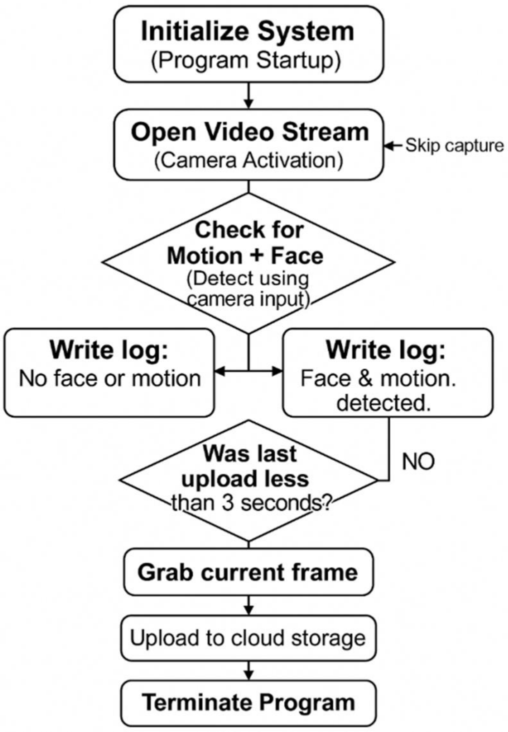

Fig. 1. The algorithm of both systems uses cameras

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

OpenCV contains over 2500 optimized algorithms to detect movement and recognize objects, faces, text, and more. Features: 1) real-time video flow processing; 2) algorithms for the detection of objects (people, transport, animals); and 3) background subtraction to detect motion; 4) support for recognition of faces and gestures; 5) modules for depth, optical flow, and image segmentation analysis. Programming languages: Python, C++, Java, Matlab. Advantages: 1) open code and a large community of developers; 2) high performance; 3) easy integration with other modules (for example, Raspberry Pi). Application scope: 1) CCTV system; 2) robotics; 3) analysis of sports events; 4) medicine (image analysis).

Raspberry Pi is a compact one-board computer designed for educational and project purposes. It supports additional modules such as cameras, sensors, and many others. Specifications (Raspberry Pi 3/4): 1) processor: ARM Cortex-A53; 2) RAM: 1–8 GB (depending on the model); 3) interfaces: USB, HDMI, CSI (for cameras), GPIO (for sensors); 4) operating system: Raspberry Pi OS (Linux). Advantages: 1) compact size; 2) support for various peripheral devices; 3) relatively low cost. Application Scope: 1) automation systems; 2) home media servers; 3) educational projects.

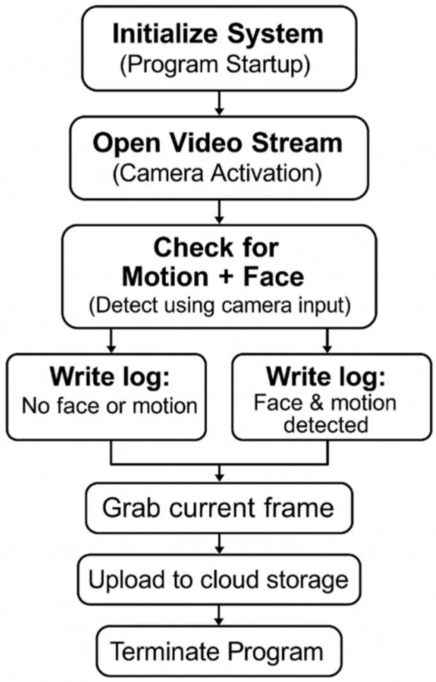

Fig. 2. The algorithm of the passive infrared sensor system

Pi Camera V2 is a compact camera designed for Raspberry Pi use. It provides high-quality images and videos. Specifications: 1) resolution: 8 MP; 2) sensor: Sony IMX219; 3) video support: full HD (1080p) at 30 frames per second; 4) connection interface: CSI (Camera Serial Interface). Advantages: 1) ease of connection to Raspberry Pi; 2) compact and lightweight design; 3) high image quality for its size. Disadvantages: 1) dependence on Raspberry Pi; and 2) performance sensitivity to lighting conditions. Application areas: 10 CCTV systems: 1) reasonable home; 2) machine training (movement detection, face recognition).

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

Description: A Passive Infrared Sensor is an electronic sensor that detects the movement of objects by changing infrared radiation. It is called passive because it does not radiate its energy but only receives infrared radiation. Specifications: 1) work range: 5–10 meters; 2) inspection angle: ~ 120°; 3) energy consumption: ~ 50 MW; 4) supply voltage: 3.3–5 V. Advantages: 1) low cost and energy consumption; 2) simplicity; 3) sensitivity to thermal changes. Disadvantages: 1) the tendency to false operation due to heating or temperature changes, and 2) the inability to distinguish the types of objects.

Application: 1) alarm system; 2) automatic lighting; 3) robotics.

Manuscripts should be divided into ordered and numbered sections. These sections should contain sufficient details to allow others to replicate and build on published results. New methods and protocols should be described in detail, while well-established methods can be briefly described and appropriately cited.

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. 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: to calculate the average processing time T_{avg}.[15] is calculated as

$$T_{avg} = \frac{1}{N}\sum_{k=1}^{N} t_k, \quad (1)$$

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

Minimal and maximal time of data processing is the following:

$$T_{\min} = \min(t_j)_{1\le j\le N}, \quad (2)$$

$$T_{\max} = \max(t_j)_{1\le j\le N}. \quad (3)$$

The average quadratic deviation (standard deviation, σ) is the following:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(t_i - T_{avg})^2}. \quad (4)$$

Detection accuracy (accuracy, a) is calculated as a percentage of successful events by the following:

$$A = \frac{E_{correct}}{E_{total}} \times 100\,\%. \quad (5)$$

Delay (streaming latency) is measured in frames per second (FPS). To calculate the average delay, we use the following formula:

$$L_{avg} = \frac{F}{T_{stream}}. \quad (6)$$

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

$$P_{err} = \left(\frac{E_{false}}{E_{total}}\right) \times 100\,\%. \quad (7)$$

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.

Comparison of sensor parameters and image capture methods

Parameter	PIR Motion	OpenCV with webcam (1920×1080)	OpenCV with Raspberry Pi camera (640×480)
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 Movement 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
The probability of an error, p_{error}, %	24	2	10

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

Several false 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 video 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.

Webcam OpenCV: high accuracy and speed with a webcam (1920×1080) 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.

V. CONCLUSION

The paper presented the primary development stages of the facial recognition system and evaluated 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 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. Experimental tests were conducted using a control set of scenarios simulating various operating conditions, including lighting changes, viewing angle variations, accessories' presence, and partial facial occlusions by external objects. The results demonstrated 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.

References

[1] Klym, H., Dunets, R., Horbatyi, I., & Diachok, R. (2018). Security subsystem and smart home management system. *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT),* 194–197. IEEE. DOI: https://doi.org/10.1109/DESSERT.2018.8409126.

[2] R. Diachok, R. Dunets and H. Klym (2018). System of detection and scanning bar codes from Raspberry Pi web camera, *IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, 2018, 184–187, DOI: https://doi.org/10.1109/DESSERT.2018.8409124

[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. In *IOP Conference Series: Earth and Environmental Science*. IOP Publishing. DOI: https://doi.org/10.1088/1755-1315/1342/1/012001

[4] Luna, J. I. V., Rangel, F. J. S., Francisco, J., Aceves, C., Guzmán, G. S., & Vargas, V. N. T. (2020). Motion detection using a Raspberry Pi 4 and OpenCV. DOI: https://doi.org/10.1088/1755-1315/1342/1/012001

[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 (JAIEA)*, *4*(2), 1354-1361. DOI:https://doi.org/10.59934/jaiea.v4i2.907.

[6] Bagye, Wire, Khairul Imtihan, and Maulana Ashari (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: https://doi.org/10.36595/jire.v7i2.1344.

[7] Jamil Alsayaydeh, Jamil Abedalrahim (2025). Handwritten text recognition system using Raspberry Pi with OpenCV TensorFlow. *International Journal of Electrical & Computer Engineering*, 2291–2303. DOI: https://doi.org/10.11591/ijece.v15i2.pp2291-2303.

[8] Park, Y. (2025). Development of Raspberry Pi Autonomous Car using OpenCV and NVIDIA CNN Model. *The Journal of the Convergence on Culture Technology,* 11(2), 367–378. DOI: https://doi.org/10.17703/JCCT.2025.11.2.367

[9] Banerjee, P., Datta, P., Pal, S., Chakraborty, S., Roy, A., Poddar, S., Ghosh, A. (2022). Home Security System Using Raspberry Pi. In *Advanced Energy and Control Systems: Select Proceedings of 3rd International Conference, ESDA 2020,* 167–176. DOI: https://doi.org/10.1007/978-981-16-7274-3_14

[10] 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: https://doi.org/10.1016/j.procs.2020.03.200

[11] F. Faisal and S. A. Hossain (2019), Smart Security System Using Face Recognition on Raspberry Pi, *13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, 1–8. DOI: https://doi.org/10.1109/SKIMA47702.2019.8982466

[12] Silalahi, L. M., Simanjuntak, I. U. V., Silaban, F. A., Budiyanto, S., & Ikhsan, M. (2020, December). Integration of opencv raspberry pi 3b+ and camera sensor in access control of vehicle ignition key system. *IOP Conference Series: Materials Science and Engineering,* 909(1), 012002. DOI: https://doi.org/10.1088/1757-899X/909/1/012002.

Roman Diachok is an assistant at the Department of Specialized Computer Systems, Lviv Polytechnic National University, Ukra ine. He received a PhD at Lviv Polytechnic National University in 2023. He has scientific, academic, and practical experience in computer system research and design. He has made a proven contribution to the methodology of FPGA design and high-performance algorithms for computer systems. He has also designed information and measurement systems with multi-sensor configurations.

Ivan Tepliakov is an assistant at the Department of Automated Control Systems, Lviv Polytechnic National University, Ukraine. He received a PhD in the field of radio engineering and telecommunications in 2020. He has scientific, academic, and practical experience researching and developing computer systems, software, and hardware products.

Myroslava Lapko received the M. Sc. degree in Computer Engineering at Lviv Polytechnic National University, Ukraine, in 2024. She has experience in scientific research, academic work, and practical development of computer vision systems using advanced methods and state-of-the-art technologies, with deployment on various hardware platforms.

Anatoli I. Popov is a Doctor of Physics, Senior scientist at the Institute of Solid State Physics, University of Latvia, one of the world's leading experts in the field of solid-state radiation physics, sensor materials for cyber-physical systems, a board member of Crystal Clear Collaboration at CERN, board member of Enabling Research Projects on Materials, EUROfusion, is an author and co-author of more than 250 articles (Scopus), including articles from the first quartile Q1 (Scientific Reports, Ceramics International, Surfaces and Interfaces, Journal of Materials Chemistry C, Nanomaterials, Symmetry, Journal of Materials Research and Technology, Journal of Nuclear Materials). In total, over the past 5 years, according to SCOPUS, 104 articles have been published.