Journal of Engineering, Electrical and Informatics

E-ISSN: 2809-8706
P-ISSN: 2810-0557

Research Article

Effectiveness of Face Recognition-Based Security System on CCTV with Raspberry Pi and Esp32-Cam Using Face Recognition Method

Frencis Matheos Sarimolle [1*], Satria Wira Yudha [2], Sutisna Sutisna [3], Ahas Eko Septianto [4]

[1] Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika (Stikom Cki) Jakarta
[2] Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika (Stikom Cki) Jakarta
[3] Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika (Stikom Cki) Jakarta
[4] Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika (Stikom Cki) Jakarta
* Corresponding Author: e-mail: frencis@stikomcki.ac.id

Abstract: Current technological advances, such as the internet of things (IOT), have a very broad scope. Especially in the security sector. The fact is that there are many robots that have been made by humans to do jobs that can help humans beyond their abilities. CCTV is very important to protect the house from various types of threats, such as burglary and other hazards. However, a security system that only uses CCTV cameras is no longer secure enough because someone is needed to monitor activities in the CCTV area for 24 hours. As for CCTV that provides facial recognition features, the price is arguably quite expensive. Therefore, we need home security with a more modern, affordable, and effective version of CCTV that utilizes the technology that has been developed to date. In this context, I propose a prototype of a sophisticated, low-cost, Raspberry-PI-based home security system that is integrated with a mobile real-time application. This intelligent robot can monitor the surrounding area by detecting people who are within the range of the camera. notification if a stranger enters the area and is not recognized by the robot to a mobile application that can be installed. The author uses Raspberry Pi hardware as the main control center, OpenCV to perform motion detection and facial recognition, a webserver to make it easier for users to access data and control the system remotely, mobile applications as notification recipients, and real-time monitoring of CCTV. In the tests carried out, the developed IOT-based security system has succeeded in detecting motion and facial recognition with good accuracy and is able to send notifications to smart phones in a short time when suspicious events occur in the house. Thus, this IOT-based home security system can help improve security and comfort by integrating technology and providing more effective and efficient solutions for protecting homes and buildings from various types of threats.

Keywords: Computer Vision; Face Recognition; Home Security; Internet of Things (IoT); Raspberry Pi.

Received: April 15, 2022
Revised: Mey 12, 2022
Accepted: Mey 25, 2022
Published: June 16, 2022
Curr. Ver.: June 30, 2022

1. Introduction

The rapid development of technology has significantly transformed various aspects of human life, particularly in the fields of Internet of Things (IoT) and Computer Vision (CV) [1]. Internet of Things refers to a technology that enables interconnected devices to communicate through networks, allowing users to remotely monitor and control electronic systems. Meanwhile, Computer Vision is a branch of computer science that focuses on image processing and visual data analysis through computational algorithms. The integration of IoT and Computer Vision technologies has created opportunities for the development of intelligent systems in various sectors, including home security systems [2], [3].

In modern society, home security has become an essential concern due to the increasing number of valuable assets stored within residential environments. Most homeowners spend a considerable amount of time outside their homes for work and daily activities, resulting in minimal supervision of their properties. Conventional home security systems such as manual door locks, Radio Frequency Identification (RFID) door locks, and traditional Closed Circuit Television (CCTV) systems are still vulnerable to manipulation and physical damage [4]. Although several modern CCTV systems have been introduced, many of them are relatively expensive and do not provide automatic notifications when unknown individuals enter the surveillance area [5].

Therefore, a more intelligent and affordable security system is needed to improve home safety and provide convenience for homeowners. The utilization of IoT, Computer Vision, and Deep Learning technologies can be implemented to develop a modern security system capable of detecting and recognizing human faces automatically [3], [6]. Facial recognition technology can identify whether an individual captured by a surveillance camera is an authorized resident or an unknown person who may potentially pose a security threat [7], [8].

Face detection and recognition processes require algorithms capable of processing visual image data captured by cameras. Several algorithms commonly used for facial recognition include the Histogram of Oriented Gradient (HOG) and Convolutional Neural Network (CNN) algorithms provided by the Dlib library [9]. These two algorithms have fundamental differences in terms of performance and computational requirements. The HOG algorithm is known for its faster image processing capability; however, it generally produces lower recognition accuracy. In contrast, the CNN algorithm provides higher accuracy in facial recognition but requires more advanced hardware resources for processing. Due to hardware limitations and efficiency considerations, the HOG algorithm is considered more suitable for implementation on standard computing devices [10].

To support the implementation of facial recognition systems, a microcontroller-based server capable of processing image data is required. One of the widely used mini-computer platforms is the Raspberry Pi, which functions as the central server for processing visual data and transmitting information to mobile applications [11], [12]. By integrating Raspberry Pi with ESP32-CAM technology, the system can perform real-time facial recognition and monitoring processes efficiently [13].

Based on these conditions, this study aims to analyze the effectiveness of implementing facial recognition technology in a Raspberry Pi-based home security system integrated with a smartphone application. The proposed system is expected to recognize faces with high accuracy and provide real-time notifications through mobile applications when unidentified individuals are detected within the camera coverage area [14], [15]. Therefore, this research is entitled "The Effectiveness of a Face Recognition-Based Security System on CCTV Using Raspberry Pi and ESP32-CAM with the Face Recognition Method."

This study is expected to contribute to the development of affordable modern security systems that improve residential safety and user convenience. Furthermore, the implementation of facial recognition technology integrated with mobile applications is expected to provide flexibility for users in monitoring their homes remotely and in real time [16], [17]. In addition, this research also contributes to the enhancement of students' technical competencies in programming, networking, and operating systems, thereby improving graduates' competitiveness in the professional field.

2. Literature Review

Research Methodology Survey

The survey methodology in this study was developed based on the *PICOC* (Population, Intervention, Comparison, Outcomes, and Context) framework. The PICOC approach was used to identify information requirements from previous studies related to the development of *IoT*-based home security systems using facial recognition technology [1], [12].

The framework is presented in Table 1.

Table 1. PICOC Framework for the Research Review.

PICOC Element	Description
Population	Homeowners who require a modern and affordable home security system.
Intervention	Implementation of an *IoT*-based home security system using Raspberry Pi facial recognition technology integrated with a real-time mobile application.
Comparison	Comparison with traditional home security systems and modern *CCTV* systems.
Outcomes	Improving home security by utilizing *IoT* and facial recognition technologies to identify unknown individuals.
Context	Private residential environments.

Survey Protocol

The survey protocol was conducted to collect scientific journal articles relevant to the proposed research topic and the PICOC framework described previously. The protocol focused on selecting studies related to IoT-based home security systems, facial recognition technologies, embedded systems, and mobile application integration [2], [8]. The detailed survey protocol is shown in Table 2.

Table 2. Survey Protocol Review.

Criteria	Description
Publication Year	2018 – 2023 (Last 5 Years)
Publication Type	Journal Articles
Search String	("home security system" OR "smart home") AND ("face recognition" OR "face detection") AND ("Raspberry Pi" OR "embedded system") AND ("mobile application" OR "Android" OR "iOS")
Final Selected Studies	20 Journal Articles

The literature search process was conducted using several scientific databases, including ScienceDirect and Google Scholar. The study selection process involved several stages to ensure that the selected articles were relevant to the research objectives. Initially, a pilot search was performed using predefined search strings. If most primary studies relevant to the topic were not identified, the search strings were refined and re-evaluated. After obtaining relevant references, the studies were screened based on titles and abstracts, followed by a full-text review process [1], [2].

A total of 43 and 985 articles were initially identified from the selected databases. After screening based on titles and abstracts, 24 articles remained for further evaluation. Subsequently, 20 journal articles were selected after the full-text assessment process.

The overall study selection process is illustrated in Figure 1.

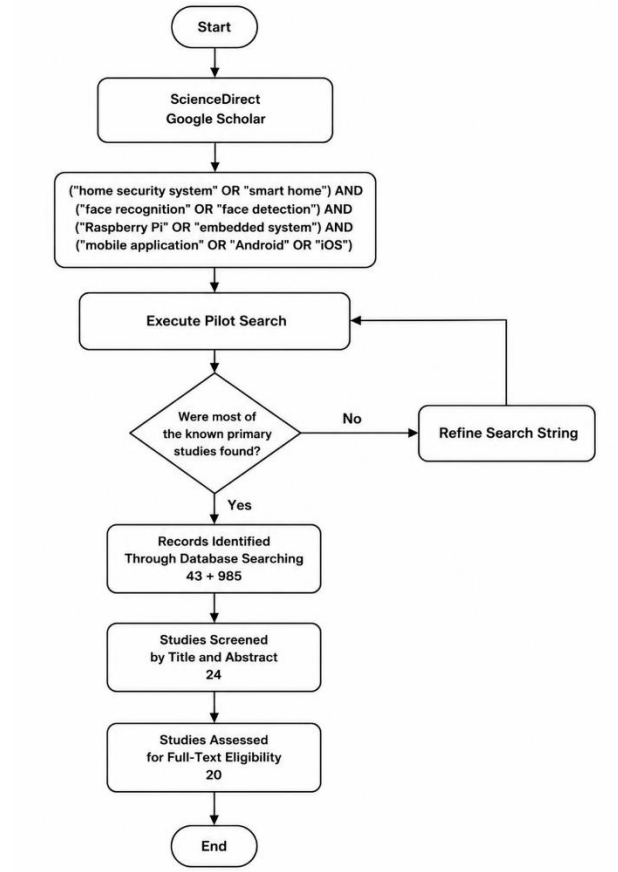

Figure 1. Studies Selection Strategy.

Fundamental Concepts

System

A system can be defined as a collection of interconnected elements or components that interact with one another to achieve specific objectives. In a broader context, a system consists of several related factors operating within a particular environment. Each component within a system has its own role and function; however, the overall objectives can only be achieved when all components operate in an integrated manner [1], [13].

Various systems can be found in everyday life, including transportation systems, educational systems, and healthcare systems. In the context of this research, the proposed home security system consists of interconnected hardware and software components designed to provide real-time monitoring and facial recognition functionalities [4], [5].

Internet of Things (IoT)

The Internet of Things (IoT) refers to a concept in which physical devices are interconnected through internet networks, enabling communication and data exchange among devices [1], [18]. Through IoT technology, devices such as sensors, cameras, and controllers can interact automatically and perform useful actions based on collected data.

IoT allows users to remotely monitor and control connected devices through internet-based applications. One common implementation of IoT technology is in smart home systems, where devices such as lighting systems, security cameras, and door locks can be controlled through smartphones. In this research, *IoT* technology is utilized to develop a home security system integrated with mobile applications for real-time monitoring and notification services [12], [15].

Computer Vision (CV)

Computer Vision (CV) is a branch of computer science related to image and video processing, analysis, and interpretation by computers. The primary objective of Computer Vision is to enable computers to understand visual information similarly to human vision [9], [10].

Computer Vision technologies involve various techniques such as object detection, facial recognition, object tracking, image segmentation, and three-dimensional reconstruction. These processes require algorithms capable of extracting important features from visual data for further analysis and decision-making. In this study, Computer Vision is implemented for facial detection and recognition processes within the home security system [3], [11].

Raspberry Pi

Raspberry Pi is a miniature computer platform widely used in electronic and do-it-yourself (DIY) projects. The platform was designed to provide an affordable and flexible computing environment for various applications, including smart home systems, robotics, and IoT-based developments [12], [13].

Figure 2. Rasberry Pi.

Raspberry Pi offers several advantages, including compact size, low power consumption, affordable cost, and compatibility with Linux-based operating systems. In addition, Raspberry Pi can be integrated with external components such as cameras, sensors, and actuators. In this research, Raspberry Pi functions as the main processing server responsible for executing facial recognition algorithms and transmitting data to mobile applications [11], [14].

ESP32-CAM

ESP32-CAM is a camera module based on the ESP32 microcontroller developed by Espressif Systems. The module is designed to support image acquisition and wireless communication in *IoT*-based applications efficiently [12], [13].

Figure 3. Esp32CAM.

The ESP32-CAM module is equipped with a 2-megapixel camera, WiFi and Bluetooth connectivity, and several input/output interfaces for connecting external devices such as sensors and relays. Due to its low power consumption and efficient processing capability, ESP32-CAM is widely used in smart surveillance and monitoring applications [4], [11].

In this study, ESP32-CAM is utilized to capture visual data in the form of images and videos, which are then transmitted to the Raspberry Pi server for facial recognition processing [14], [17].

Dlib

Dlib is an open-source software library commonly used in image processing, pattern recognition, and Machine Learning applications. The library was developed by Davis King and supports programming languages such as C++ and Python [9], [10].

Dlib provides various algorithms and functions for image analysis, including object detection, facial recognition, image segmentation, and Machine Learning model implementation. One of the main advantages of Dlib is its support for Deep Learning architectures, allowing developers to implement efficient neural network models for visual recognition tasks. In this research, Dlib is used as the primary library for implementing facial recognition algorithms [3], [6].

Histogram of Oriented Gradients (HOG)

Histogram of Oriented Gradients (HOG) is a feature extraction method commonly used in object detection and image recognition tasks. The method is widely applied in facial recognition, pedestrian detection, and vehicle detection systems [9], [10].

The basic principle of HOG involves calculating the orientation gradients within image regions to represent object shapes and textures. The method is effective for detecting objects with varying textures and is relatively robust to scale variations. Compared to more computationally intensive methods such as Convolutional Neural Networks (CNN), *HOG* offers faster processing performance and lower hardware requirements, making it suitable for implementation on Raspberry Pi devices [3], [12].

React Native

React Native is a mobile application development framework developed by Facebook. The framework enables developers to create cross-platform mobile applications using JavaScript and React-based components [15], [16].

One of the primary advantages of React Native is its capability to develop applications for both Android and iOS platforms using a single codebase. React Native adopts a native rendering approach, where React components are converted into native user interface components on the target platform. This approach provides good performance while maintaining access to platform-specific features and APIs [8], [17].

3. Materials and Method

Research Data

The data collection process in this study utilized references obtained from several Systematic Literature Reviews (SLR), documentation from algorithm libraries, and supporting datasets required for prototype development. To support the implementation of the proposed home security system, additional data from family members and individuals associated with the homeowners were collected for computational purposes.

The collected data consisted of personal information such as names and facial image datasets. These facial images were used as reference data for the facial recognition process implemented within the system. The datasets enabled the proposed system to identify authorized individuals and distinguish them from unknown persons detected within the CCTV surveillance area.

Furthermore, the collected datasets were processed using facial recognition algorithms integrated into the Raspberry Pi-based system architecture. The data played an important role in supporting real-time facial verification and improving the effectiveness of the proposed IoT-based home security system.

Method Implementation

The methodology implemented in this study focuses on facial detection and recognition using Deep Learning-based algorithms provided by the Dlib library. The entire processing mechanism is executed on a Raspberry Pi mini-computer integrated with the ESP32-CAM module. The system is designed to automatically send notifications to the homeowner whenever an unidentified person is detected within the CCTV monitoring area.

Architecture Design

The architecture design refers to the process of planning and organizing the overall structure of the proposed security system. The system architecture consists of hardware and software components integrated to support real-time facial recognition and monitoring processes.

The proposed architecture includes an ESP32-CAM module for image acquisition, a Raspberry Pi server for image processing and facial recognition computation, and a mobile application developed using React Native for monitoring and notification services. Communication between devices is performed through internet connectivity using *IoT*-based communication mechanisms.

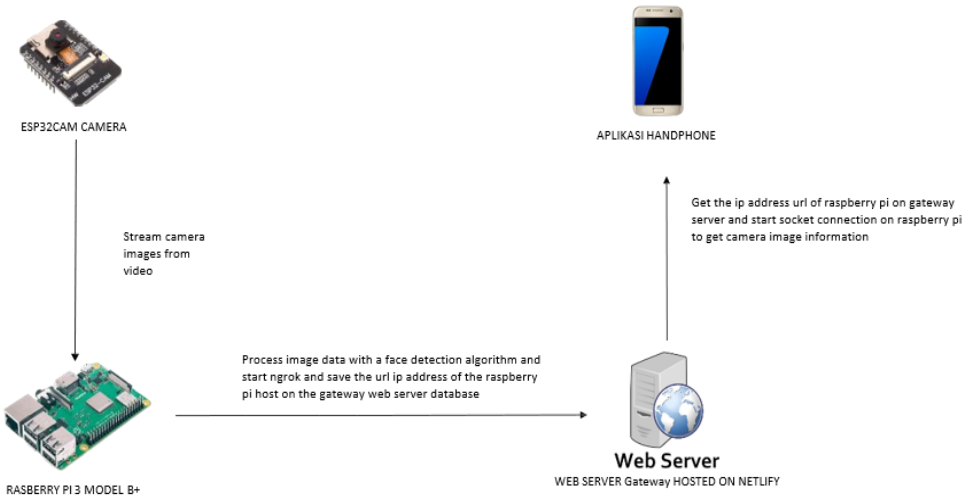

Figure 3. Architectural Design.

Face Recognition Method (Dlib and HOG)

The facial recognition method implemented in this study combines the *Histogram of Oriented Gradients* (*HOG*) algorithm with the Dlib library to develop an accurate and reliable facial recognition system through image processing and *Machine Learning* techniques. The process begins with facial detection using Dlib, which identifies facial regions from image frames captured by the camera module. After the facial regions are successfully detected, the system performs face pose alignment to normalize facial orientation because faces captured by surveillance cameras may appear in different positions and angles. During this process, facial landmarks such as the eyes, nose, and ears are identified to extract important facial feature information. Subsequently, the detected facial images are converted into numerical representations through a face encoding process. Using *Deep Learning* mechanisms provided by Dlib, the system generates 128-dimensional facial embeddings that represent unique facial characteristics derived from grayscale intensity values and facial landmark positions. These facial encoding vectors are then utilized to compare and verify facial similarities between detected individuals and registered datasets within the system.

Push Notification and Real-Time CCTV Method

To provide immediate information to homeowners when abnormal conditions occur, the system implements an automatic push notification mechanism. Notifications are generated when the system successfully detects an unknown individual within the CCTV monitoring area.

The Raspberry Pi server sends push notifications to the mobile application through Google Firebase Cloud Messaging services. This mechanism allows users to receive warning notifications in real time through their smartphones.

In addition to push notification services, the Raspberry Pi server also provides socket-based communication for real-time image transmission. Captured image frames are converted into Base64 text format and periodically transmitted to the React Native mobile application. These image streams are then reconstructed into real-time video monitoring accessible through the smartphone application.

System Flowmap

The operational flow of the proposed home security system consists of several sequential processes, beginning with image acquisition from the ESP32-CAM module, followed by facial detection and recognition processing on the Raspberry Pi server. If the detected face matches the registered dataset, the system categorizes the individual as an authorized user. Otherwise, the system generates a warning notification and sends it to the homeowner's mobile application.

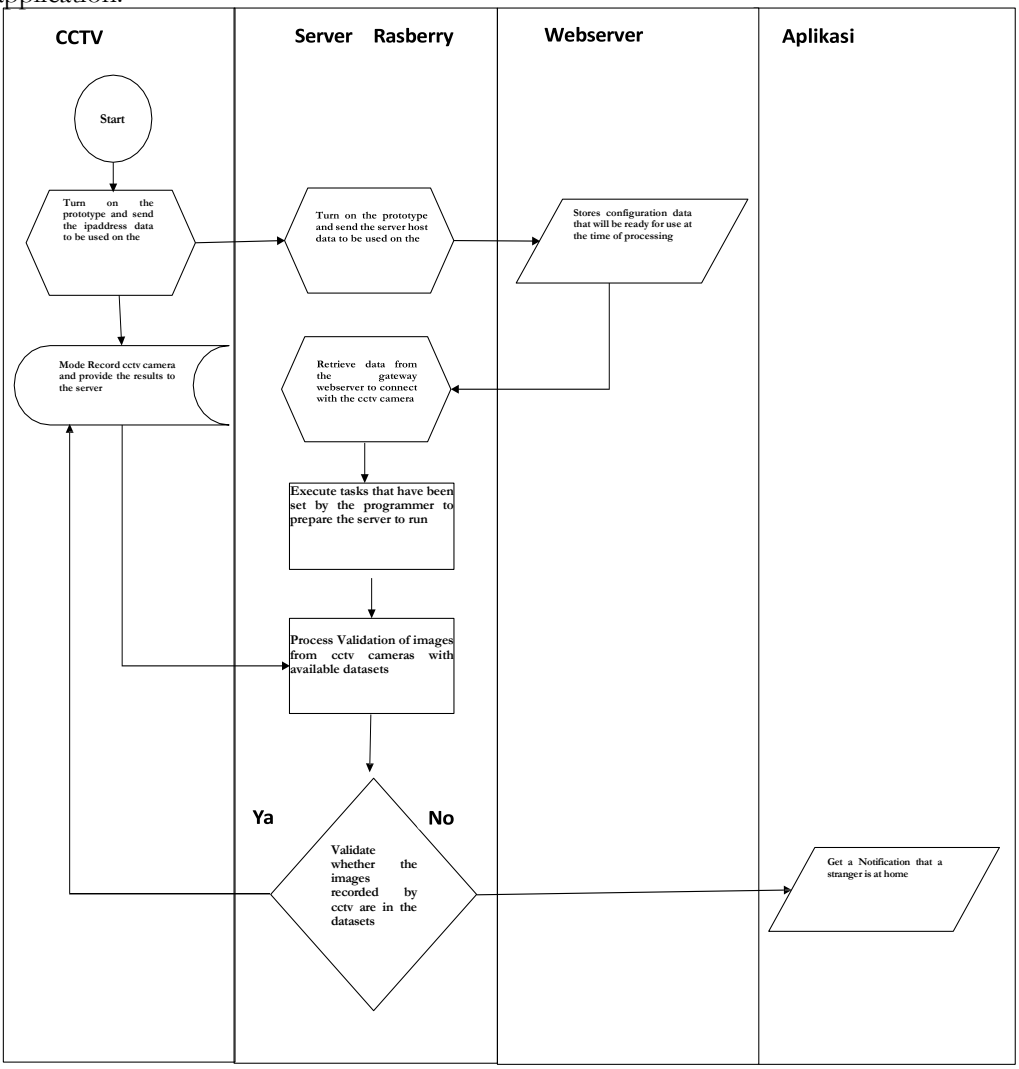

Figure 4. Flowmap Diagram of Running System.

The overall process enables users to monitor home conditions remotely and receive immediate alerts when suspicious individuals are detected within the surveillance area.

Testing Design

The testing process was conducted directly by the researcher using several predefined scenarios to evaluate the effectiveness of the proposed security system. The testing focused on the system's capability to recognize registered individuals and detect unknown persons accurately.

Table 3. Testing Scenario Results.

Testing Scenario	Test Case	Expected Result	Testing Result
Detecting registered family members within camera coverage	Individual walks toward and faces the camera	The server does not send warning notifications to the mobile application	Appropriate

Detecting unknown individuals within camera coverage	Unregistered individual walks toward and faces the camera	The server sends warning notifications indicating the presence of an unknown person	Appropriate

The testing results indicate that the proposed system successfully distinguishes between registered and unregistered individuals within the CCTV monitoring area. Furthermore, the push notification mechanism operates properly and delivers real-time alerts to users through the mobile application.

4. Results and Discussion

Research Tools

The software needed to run the proposed system is:

Table 4. Software Technology Requirements.

Software	Specification
Nano Terminal	Version 2.3.40
Firebase	Version 9.0.0
PuTTY	Version 0.76
Database	MongoDB Atlas (*NoSQL*)

Table 5. Hardware Technology Requirements.

Hardware	Specification
Raspberry Pi	Processor: BCM2837 Quad-Core A53 (ARM v8) 64-bit up to 1.2 GHz, Memory: 1 GB LPDDR2 SDRAM
ESP32-CAM	Ultra-small 802.11 b/g/n Wi-Fi + BT/BLE, Low-power Dual-Core 32-bit CPU, 520 KB SRAM, External 4 MB PSRAM
Smartphone	Minimum Android 6.0 (*Marshmallow*), Quad-Core Snapdragon 400 Series Processor, 2/16 GB Storage, HD Display Resolution (720p) or higher

Methodology Implementation

System Implementation

The implementation of the proposed home security system was developed based on the architectural design described in Chapter 3. The system operation begins when the IP address information of the ESP32-CAM and Raspberry Pi hardware devices is stored in the database. Afterward, communication between the hardware components is established to enable the integrated system to operate according to the intended functionalities. Through this communication mechanism, the system can perform image acquisition, facial recognition processing, and real-time monitoring effectively.

Database Implementation

The proposed system architecture utilizes a NoSQL database management system. One of the online and freely accessible NoSQL database service providers used in this research is MongoDB Atlas. By utilizing MongoDB Atlas, the system does not require local database installation on the Raspberry Pi device because the database services are already available through cloud-based infrastructure. This cloud implementation simplifies database access, management, and real-time data synchronization within the proposed *IoT*-based security system.

Data Communication Implementation

Based on the architectural design described in Materials and Method, each hardware component used in the proposed system was designed dynamically, allowing the devices to operate without requiring manual configuration during the initial setup because the configuration process is automatically managed by the system. The data communication process within this project utilizes three different communication techniques, namely Local

Tunneling, Application Programming Interface (API), and WebSocket. Local Tunneling enables servers or services running on local networks, particularly on the Raspberry Pi device, to be accessed through public networks, thereby facilitating remote communication and monitoring. Furthermore, the system implements an API-based communication mechanism using the HTTP protocol through a Node.js server running on the Raspberry Pi. The API is utilized for storing and exchanging hardware information such as the IP addresses of the ESP32-CAM and Raspberry Pi devices, enabling communication between hardware components within the same network environment. In addition, WebSocket technology is implemented to support bidirectional real-time communication between the client and server. Compared to conventional HTTP request-response mechanisms, WebSocket enables continuous and interactive data transmission without requiring repeated requests from the client. This communication method is particularly important for real-time CCTV monitoring because image data captured from the facial recognition process on the Raspberry Pi can be continuously transmitted through the Node.js server to the mobile application developed using React Native and integrated with Socket.IO technology.

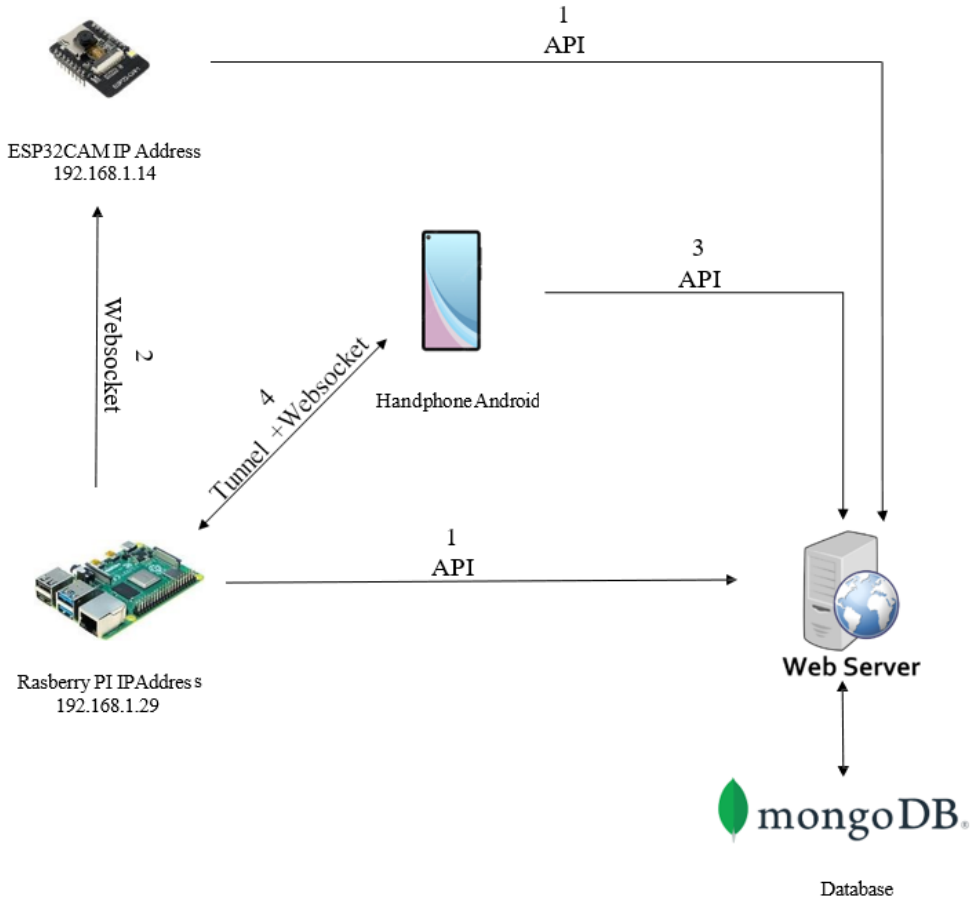

Figure 5. Data Communication System Sequence.

The numbering shown in the communication flow diagram represents the sequence of interactions performed by the system components during operation. Several stages must be completed to ensure that all hardware and software components communicate properly within the proposed security system. Initially, when each hardware device is powered on, the system performs an automatic registration process to store the IP address information of each device into the MongoDB database. This process enables both hardware and software components to identify communication destinations and establish connectivity among devices. After all devices successfully obtain their communication addresses, the Raspberry Pi begins interacting with the ESP32-CAM module by requesting captured image data through the WebSocket protocol. Once the image data are received, the Raspberry Pi executes the facial recognition algorithm and performs a tunneling process to make the server accessible through public networks. Subsequently, the mobile application installed on the

user's smartphone communicates directly with the system by retrieving the tunnel domain information stored in the MongoDB database, which is provided through the NGROK tunneling service. Finally, the system performs real-time CCTV streaming by transmitting the facial recognition video results through the WebSocket protocol using the established tunneling connection.

Face Recognition Technique Implementation

The implementation of the facial recognition technique in this study was conducted through several stages based on the system design described in Chapter 3. These stages include facial object detection from various face poses, image modification and resizing to accelerate data processing performance, and the utilization of Deep Learning methods to generate unique facial codes based on important facial characteristics such as eye position, nose shape, and other facial landmarks. Subsequently, the generated unique facial encoding data obtained from CCTV images are compared and verified with all facial datasets stored within the system database.

a.) Face Object Detection from Various Poses

Several algorithms for object detection have been developed, including the Viola-Jones algorithm with Haar Feature Classifier (2001) and the Histogram of Oriented Gradients (HOG) algorithm introduced by Navneet Dalal and Bill Triggs (2005). In this study, the HOG algorithm was selected to detect facial positions within images because it provides faster processing performance compared to the Haar Cascade method. Furthermore, this algorithm has also been widely implemented in large-scale applications, including Facebook's facial recognition and friend-tagging features.

Figure 6. HOG Implementation in Drawing.

b.) Face Alignment and Deep Learning Facial Encoding

At this stage, the system performs facial alignment by adjusting the facial position obtained from the Histogram of Oriented Gradients (HOG) face detection process. This procedure is necessary to identify the central facial landmark points, such as eye positions and other important facial features, relative to the center point of the detected face. Through this alignment process, the generated unique facial encoding data can represent facial characteristics more consistently based on standardized facial positions within the image frame. This approach simplifies the Deep Learning process in comparing unique facial encoding vectors during the subsequent facial verification stage.

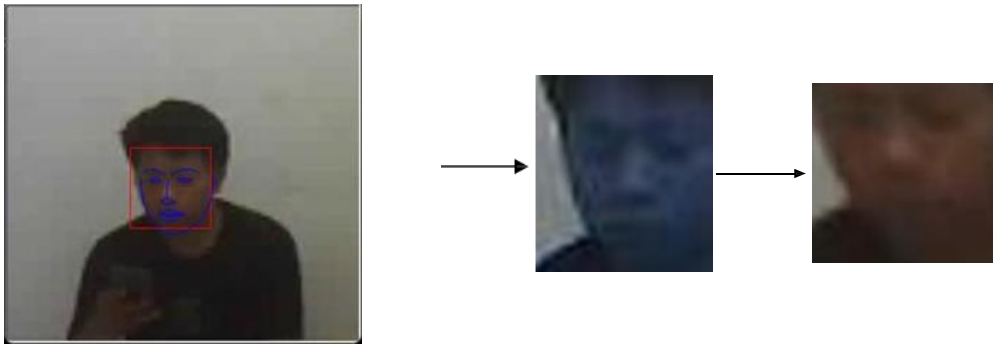

Figure 7. Intermediate Face Object Modifiers.

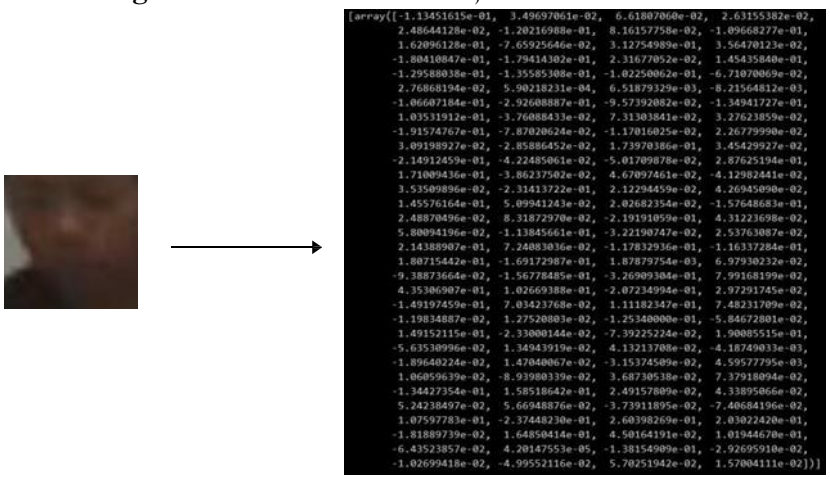

Figure 8. Unique Deep Learning Result Code.

c.) Face Recognition Process

At the final stage, the system performs an analysis using Deep Learning techniques to determine the similarity between the facial encoding vectors captured by the camera and the facial datasets stored on the Raspberry Pi device. The data processing mechanism is carried out through complex Deep Learning algorithms that enable the computer to analyze and compare unique facial characteristics automatically. Therefore, adequate hardware specifications are required to achieve faster and more optimal processing performance. In this study, the researcher utilized a Raspberry Pi 3 Model B to execute the facial recognition process, which required approximately 0.5 to 1.5 seconds to complete the analysis and produce sufficiently accurate recognition results.

Figure 9. Results of Face Encoding on Unique Code.

User Mobile Application Interface Implementation

The implementation of the user mobile application interface focuses on providing a real-time CCTV monitoring display that can be accessed directly through the mobile application. In addition to real-time video monitoring features, the application also provides several supporting functionalities, including push notifications, a list of unidentified faces, a list of recognized individuals, and other monitoring features designed to improve user convenience and enhance the effectiveness of the proposed home security system.

a.) User Mobile Application Login Interface

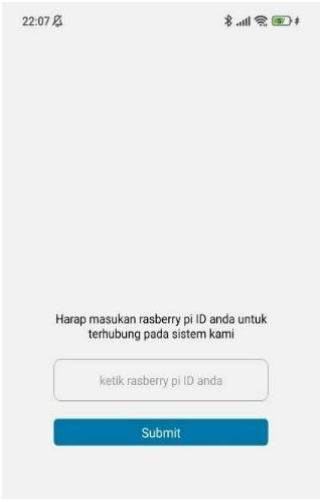

Figure 10. Mobile Application Login Interface.

When opening the mobile application for the first time, users are required to enter the Raspberry Pi ID displayed on the Raspberry Pi device label. Since the Raspberry Pi information has been previously registered within the database, the login form cannot be filled arbitrarily. If the user enters an incorrect Raspberry Pi ID, the system displays an error message stating that the "Raspberry Pi is not registered." However, if the entered Raspberry Pi ID matches the registered ID stored in the database, the user is redirected to the homepage while the system simultaneously initiates a *WebSocket* connection to receive real-time CCTV image data.

b.) User Mobile Application Homepage Interface

Figure 11. Mobile Application Homepage Interface.

After successfully logging into the application, users are directed to the homepage interface. This page provides a live CCTV monitoring feature that has already undergone the facial recognition process. In addition, the homepage displays summary information regarding

the number of unidentified faces detected by the system, allowing users to quickly monitor security conditions without navigating to additional pages. The interface also provides a logout feature that enables users to return to the login page.

c.) User Mobile Application Dataset Interface

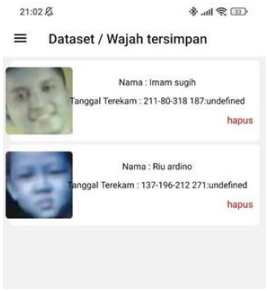

Figure 12. Stored Dataset List Interface.

This interface displays a list of facial datasets that have been previously stored within the system. The facial recognition algorithm validates detected faces by comparing them with the facial datasets available on this page. If an individual captured by the CCTV camera matches one of the stored datasets, the application will not generate warning notifications. Furthermore, users are able to delete stored datasets through this interface. When a dataset is deleted, the system sends a signal through the *WebSocket* connection to remove the corresponding dataset from the Raspberry Pi server.

d.) User Mobile Application Unknown Face Interface

Figure 13. Unrecognized Face Interface.

This page displays a list of unidentified faces or faces whose facial encoding vectors do not match any stored datasets. Through this interface, users can convert selected unidentified faces into registered datasets. After the selected face is successfully stored within the dataset list, the system will subsequently recognize the individual whenever the same face appears within the CCTV monitoring area. During this process, users are required to input the individual's name after selecting the "I Know This Person" button.

e.) Notification Alert Interface

This interface displays notification alerts generated when an individual is detected within the CCTV surveillance area. Users can receive these notifications even when the mobile application is not actively opened. The notification delivery mechanism utilizes Firebase

Cloud Messaging services by sending push notifications through the Firebase Console using the user smartphone token that has been previously stored in the database.

Figure 14. Notification Alert Interface.

Final Testing Results

Based on the testing design described in Materials and Method, direct experiments were conducted to evaluate the effectiveness of the Raspberry Pi and ESP32-CAM-based home security system. Several important findings and conclusions were successfully obtained from the testing process.

Facial Recognition Accuracy

Through direct implementation testing of the facial recognition prototype, it was identified that several factors significantly influence the accuracy of facial recognition processing within image data. These factors include hardware specifications, the facial recognition algorithm used, and image quality obtained from the camera device.

Table 6. Factors Affecting Facial Recognition Accuracy.

Determining Factor	Current Implementation	Contribution
Hardware (*Mini Computer*) Raspberry Pi Version	Raspberry Pi 3 Model B+ (Quad-Core A53 ARM v8 64-bit, 1 GB LPDDR2 SDRAM)	50%
Facial Recognition Algorithm	Dlib Facial Recognition Algorithm (*HOG*) with 2x Iteration	35%
Camera Image Quality (Contrast, Resolution, etc.)	ESP32-CAM (JPEG, BMP, 2 Megapixel 1600×1200 Resolution, Grayscale)	15%

Based on these factors and the hardware configuration utilized in this project, the following processing speed and facial recognition accuracy results were obtained.

Table 7. Obtained Facial Recognition Accuracy Results.

HOG Facial Recognition Algorithm with Facial Encoding Comparison Repeated (N) Times	Processing Time Required for Facial Recognition per Image	Recognition Result from 5 Image Frames (Close Object)	Recognition Result from 5 Image Frames (Distant Object)
1x	0.5 Seconds	4 / 5	1 / 5
2x	1 Second	5 / 5	3 / 5
3x	1.5 Seconds	5 / 5	4 / 5

The testing results indicate that increasing the number of algorithm iterations improves facial recognition accuracy, particularly for objects located farther from the camera. However, additional iterations also increase the processing time required by the Raspberry Pi device.

Advantages and Limitations

The testing process also identified several advantages and limitations of the proposed home security system prototype. These findings are summarized in Table 8.

Table 8. Advantages and Limitations of the Proposed System.

Advantages	Limitations
Minimal initial configuration is required because the system is designed to connect automatically	Requires adequate lighting conditions
Can be integrated with other *IoT*-based systems such as smart home devices and workplace automation	Image processing speed depends on hardware specifications
Dynamic features integrated with a mobile application accessible from anywhere	-

Overall, the proposed system demonstrated effective performance in implementing facial recognition-based home security monitoring while maintaining flexibility and real-time accessibility through mobile applications.

5. Conclusion

Based on the experimental results obtained from the development of a facial recognition-based home security system using Raspberry Pi and ESP32-CAM, this study successfully implemented an IoT-based security system capable of improving residential security through facial recognition technology. The proposed system was able to identify individuals detected within the CCTV surveillance area and distinguish between registered and unregistered users effectively. The testing results demonstrated that the Raspberry Pi mini-computer and ESP32-CAM module were capable of communicating properly and executing facial recognition algorithms successfully to determine the identity status of individuals captured by the surveillance camera. Furthermore, the implementation results indicated that facial recognition processing performance highly depends on hardware specifications. The Raspberry Pi 3 Model B+ showed relatively good performance in detecting facial positions and comparing facial encoding vectors generated through Deep Learning processes, with image processing repeated three times within approximately 1.5 seconds. In addition, the integration of communication technologies such as WebSocket, Local Tunneling, and API-based communication contributed significantly to the effectiveness of real-time data transmission within the system. Moreover, the push notification feature implemented on users' smartphones proved effective in providing immediate alerts when unknown individuals were detected within the CCTV monitoring area, thereby offering additional convenience and security for homeowners who frequently perform activities outside their homes with limited direct supervision.

References

[1] P. L. Chong, Y. Y. Than, S. Ganesan, and P. Ravi, "An Overview of IoT Based Smart Home Surveillance and Control System: Challenges and Prospects," *Malaysian J. Sci. Adv. Technol.*, pp. 54–66, 2023.

[2] H. H. Ali, J. R. Naif, and W. R. Humood, "A New Smart Home Intruder Detection System Based on Deep Learning," *Al-Mustansiriyah J. Sci.*, vol. 34, no. 2, pp. 60–69, 2023, [Online]. Available: https://mjs.uomustansiriyah.edu.iq/index.php/MJS/article/view/1267

[3] A. Rahim, Y. Zhong, and T. Ahmad, "A Deep Learning-Based Intelligent Face Recognition Method in the Internet of Home Things for Security Applications," *J. Hunan Univ. Nat. Sci.*, vol. 49, no. 10, pp. 39–52, 2022.

[4] R. A. Nadafa, S. M. Hatturea, V. M. Bonala, and S. P. Naikb, "Home Security Against Human Intrusion Using Raspberry Pi," *Procedia Comput. Sci.*, vol. 167, pp. 1811–1820, 2020.

[5] M. H. Khairuddin, S. Shahbudin, and M. Kassim, "A Smart Building Security System with Intelligent Face Detection and Recognition," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1176, no. 1, p. 12030, 2021.

[6] P. K. Malpe, "A Face Recognition Method in the Internet of Things for Security in Smart Recognition Places," *Int. J. Res. Appl.*

Sci. Eng. Technol., vol. 10, no. 1, pp. 687–690, 2022.

[7] S. Yedulapuram, R. Arabelli, K. Mahender, and C. Sidhardha, "Automatic Door Lock System by Face Recognition," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 981, no. 3, p. 32036, 2020.

[8] Y. X. Tok, N. Katuk, and A. S. Che Mohamed Arif, "Smart Home Multi-Factor Authentication Using Face Recognition and One-Time Password on Smartphone," *Int. J. Interact. Mob. Technol.*, vol. 15, no. 24, pp. 32–48, 2021, [Online]. Available: https://online-journals.org/index.php/i-jim/article/view/25393

[9] S. Suwarno and K. Kevin, "Analysis of Face Recognition Algorithm: Dlib and OpenCV," *J. Informatics Telecommun. Eng.*, vol. 4, no. 1, pp. 173–184, 2020.

[10] Z. Zhu and Y. Cheng, "Application of Attitude Tracking Algorithm for Face Recognition Based on OpenCV in the Intelligent Door Lock," *Comput. Commun.*, vol. 154, pp. 390–397, 2020.

[11] H. Meddeb, Z. Abdellaoui, and F. Houaidi, "Development of Surveillance Robot Based on Face Recognition Using Raspberry-PI and IoT," *Microprocess. Microsyst.*, vol. 96, p. 104728, 2023.

[12] S. A. Radzi, M. K. M. F. Alif, Y. N. Athirah, A. S. Jaafar, A. H. Norihan, and M. S. Saleha, "IoT Based Facial Recognition Door Access Control Home Security System Using Raspberry Pi," *Int. J. Power Electron. Drive Syst.*, vol. 11, no. 1, p. 417, 2020.

[13] Andreas, C. R. Aldawira, H. W. Putra, N. Hanafiah, S. Surjarwo, and A. Wibisurya, "Door Security System for Home Monitoring Based on ESP32," *Procedia Comput. Sci.*, vol. 157, pp. 673–682, 2019.

[14] M. Zuma, P. A. Owolawi, V. Malele, K. Odeyemi, G. Aiyetoro, and J. S. Ojo, "Intrusion Detection System Using Raspberry Pi and Telegram Integration," in *Proceedings of the International Conference on Artificial Intelligence and its Applications*, New York, NY, USA: ACM, 2021, pp. 1–7.

[15] K. M. Mohi Uddin, S. Afrin Shahela, N. Rahman, R. Mostafiz, and M. M. Rahman, "Smart Home Security Using Facial Authentication and Mobile Application," *Int. J. Wirel. Microw. Technol.*, vol. 12, no. 2, pp. 40–50, 2022.

[16] V. S. Reddy, S. Cheerla, S. Inthiyaz, V. V. N. Chakravarthy, and V. G. Ram, "Face Recognition and Home Automation Using Telegram Bot," in *Proceedings*, 2021, p. 20004.

[17] S. Chitti, P. R. Rao, J. T. Kumar, and S. Merugu, "Implementation of Integrated Home IoT and CCTV Face Recognition Technology," in *Proceedings*, 2022, p. 30034.

[18] G. Rajeshkumar, M. Braveen, R. Venkatesh, P. Josephin Shermila, B. Ganesh Prabu, and B. Veerasamy, "Smart Office Automation via Faster R-CNN Based Face Recognition and Internet of Things," *Meas. Sensors*, vol. 27, p. 100719, 2023.