ISSN 3032-3533
Vol 2, No.1 (2025)

Design and Build a Smart Door Lock Home Security System with the Face Recognition Method Based on ESP32 CAM

1st Mutiara Widasari Sitopu
dept. Electrical Engineering
Politeknik Negeri Medan
Medan, Indonesia
mutiarasitopu@polmed.ac.id

2nd Charla Tri Selda Manik
dept. Electrical Engineering
Politeknik Negeri Medan
Medan, Indonesia
charlatriselda@polmed.ac.id

3rd Ummu Handasah
dept. Electrical Engineering
Politeknik Negeri Medan
Medan, Indonesia
ummuhandasah@polmed.ac.id

4th Muhammad Sukri Habibi Daulay
dept. Electrical Engineering
Politeknik Negeri Medan
Medan, Indonesia
mdaulayhabibi@polmed.ac.id

***Abstract*— This study aims to design and develop a home security system based on facial recognition technology, using the ESP32-CAM device to unlock the door through the activation of the solenoid. This system utilizes the Haar Cascade algorithm in detecting facial features such as hair, forehead, eyes, eyebrows, nose, and lips. This tool is designed to improve the level of security with a more modern and precise technological approach. The test results showed that the accuracy of the system at a distance of 20 cm was 12%, 40 cm reached 80%, and 60 cm was 16%. Meanwhile, tests with a variety of accessories showed an accuracy of 68% for faces without accessories, 6% for faces with mouth masks, 66% for glasses faces, and 14% for faces wearing headbands. Testing of sabotage attempts using photos showed a detection failure rate of 100%, proving that the system is resistant to such manipulation. The designed system contributes to the development of home security technology that is more effective and adaptive to user needs.**

Keywords—ESP32_Cam, Solenoid, Face Detection component.

I. Introduction

Modern life that is increasingly developing along with technological advances provides convenience and efficiency in various fields, including in security systems. Crimes such as theft, extortion, and identity forgery are still serious threats in society (Maruli, 2021). A report by the Central Statistics Agency of Lampung Province shows a significant increase in the crime rate, with 10,191 cases in 2020 (Central Statistics Agency, 2020). This indicates the need to improve the security system, especially in the home environment.

One effective way to strengthen home security is to utilize modern technology. Padlocks that used to be the main tool are now replaced by automatic sensor technology such as RFID and biometrics, including Face ID. This facial recognition technology works by detecting the user's characteristics, which are then converted into digital data for identification (Sarwoko, 2006). Biometric technology, which includes fingerprints, retina, DNA, and facial recognition, offers a high level of accuracy and ease of implementation (Wardoyo et al., 2019). Several studies, such as those conducted by Nusri and Alimuddin (2022) and Hayati (2022), show the success of the use of biometrics in improving security systems.

The Face ID-based security system with ESP32-CAM is now a more economical alternative compared to the Raspberry Pi 3. The system utilizes cameras to detect the user's face and match it to existing data. After face verification, the system will unlock the solenoid for a few seconds, then lock it again. The system is also equipped with an LCD to display the detection results in real-time, and can be accessed through a web server for easy monitoring. This research aims to develop a smart door system based on Face ID using ESP32-CAM which is affordable and effective in improving home security.

1. Purpose

The objectives of this research are as follows:

1. Design and implement a home security system with smart door technology based on facial recognition using ESP32-CAM.
2. Measures the accuracy of face detection at various distances of use.
3. Test the device's performance in recognizing faces with additional accessories.
4. Evaluate the device's ability to detect facial recognition, such as the use of images or photos.

2. Problem Limitations

The system's facial database is only capable of recognizing faces that have been pre-registered, without any adaptive learning features or the ability to automatically register new faces by the device.

II. Literature Study

A. Related Research

This study developed a smart door system based on face detection technology with ESP32-CAM, which has been discussed in several previous studies. The system uses a relay connected to a microcontroller and a solenoid as a door locking mechanism. If a forced access attempt is detected, the microcontroller will lock the door and send an SMS

notification. The results of previous research produced a prototype of the system (Yanto et al., 2022).

The development of smart door systems is carried out in two main stages: hardware and software design. Hardware design involves assembling electronic components until the device functions to specifications, while software design involves setting up a microcontroller using an Arduino IDE with programming language C. Program code is uploaded via a USB cable to activate the hardware according to instructions. Figure 2.1 shows the overall design of the system that includes both aspects.

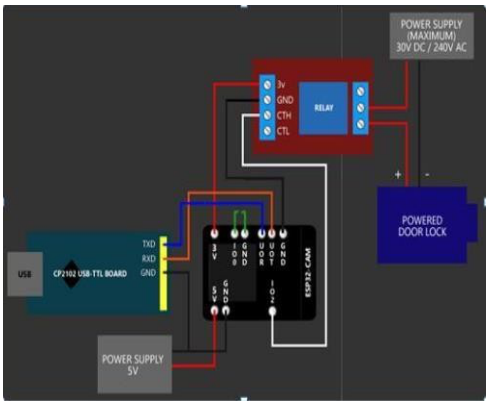


Figure 1. Overall design of the smart door prototype (Yanto et al., 2022).

In the face recognition-based security system using ESP32-CAM, programming is done with C through the Arduino IDE. The test was performed by connecting the ESP32-CAM to a USB 232 FTDI device. The first step in the creation of the software is to create a BOT in the Telegram application to obtain the BOT ID used in the integration system. The system uses solenoid lock doors as an actuator to lock the door, with additional components such as buzzers and LCDs to enhance functionality, making implementation easier (Koroy et al., 2020).

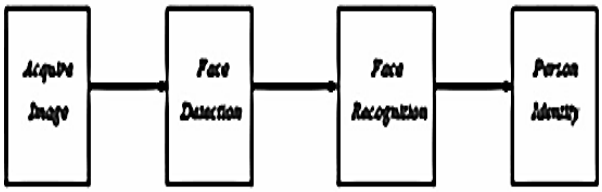
The safe-based security system developed with the Raspberry Pi 3 starts with the initialization of the face using a camera module connected to the Raspberry Pi 3. Facial data is stored and used for matching. The system uses Python as a programming language and is equipped with a buzzer for alarms, as well as an IRF 540 as a relay to activate the solenoid that locks the safe door. HDMI LCD is also used to make it easier to adjust the position of the face during detection (Putri Hayati, 2022).

Research by Ibrahim et al. (2020) developed a house door security system using the triangular face method. Facial images obtained through the webcam are processed by the Raspberry Pi 3 for matching with existing data. After that, the results are sent to the user via Telegram to control the actuator. The relay is used to activate the solenoid lock door as the main door lock.

B. Face Recognition

Biometric technology, especially facial recognition, has come a long way in recent years (Suryansah et al., 2020) [1], [2]. This technology uses computer algorithms to recognize individuals through the unique characteristics of faces, such as shapes, textures, and other distinctive elements, resembling how human vision works (Alfauzan et al., 2017) .

The facial recognition process utilizes visual data in the form of pixels displayed in formats such as greyscale or true color. Factors such as lighting, contrast, contour, and texture



greatly affect the quality of data processing results (Bambang Yuwono et al., 2019) [3], [4], [5], [6], [7]. In general, this technology consists of several stages of the process which can be seen in Figure 2.2.

Figure 2. Stages of facial recognition system

The first step in facial recognition is *Acquire Image*, which is the process of capturing an image of a face or object. Furthermore, the image is processed in the *Face Detection* stage to recognize the face before undergoing the *Face Recognition* algorithm. The last stage is *Person Identity*, where the system can recognize the identity of the user or object.

The facial recognition method is divided into two, namely 2D and 3D-based. The 3D method uses approaches such as facepoint analysis, half-face measurement, and 3D geometric dimensions. Meanwhile, the 2D method uses two-dimensional images as data input (Gürel, 2012).

Popular algorithms in facial recognition include *Principal Component Analysis* (PCA) and *Linear Discriminant Analysis* (LDA). PCA functions to reduce the dimension of the image with *the eigenface* technique, which utilizes *the eigenvector* as a representation of the main characteristics of the face image (Budi & Maulana, 2016). LDA, on the other hand, is used for data classification by looking for the best attributes, minimizing variance between classes, and projecting data into lower-dimensional spaces (Rahman, 2015).

C. Hair Cascade Algorithm

The Haar Cascade algorithm is a method designed forFeatures such as Haar , y Integral Images I Have At the detection stage, images of irrelevant areas will be filtered first, so that only potential areas are further analyzed. This process makes the algorithm efficient in recognizing faces

The implementation begins with building a detector model using a specific software library (Indriawan, 2022). Feature analysis was carried out by calculating the difference in the intensity of light and dark areas to determine the presence of faces in the image (Yulina, 2021).

Overall, Haar Cascade is known to be accurate and is often used in a wide range of face detection applications (Bradley et al., 2007).

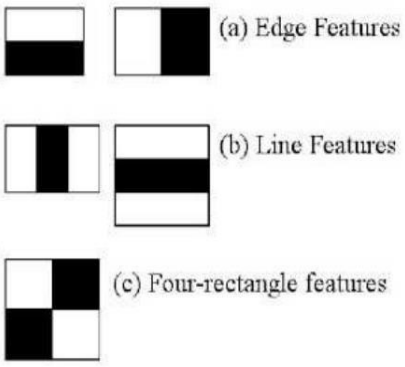


Figure 3. Haar Features

The real-time face detection process involves calculating rectangular features using an image representation called *an integral image* . An integral image at a location (x, y) is the result of summing the pixel values in a given area with the formula:

$$ii(x,y) = \sum i(x',y')\ x' \leq x, y' \leq y$$

Information:
II: Integral Images
i: Pixel value on the original image

If the equation is applied in a square plane, as shown in Figure 2.3, then this formula can be further simplified into a new equation to calculate a specific value in that field.

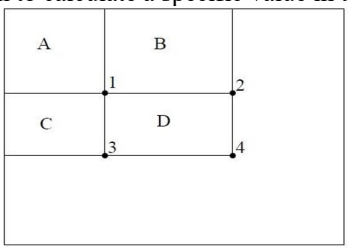


Figure 4. Pixels in a rectangle

$$s(x, y) = s(x, y - 1) + i(x, y) \quad (2.2)$$
$$ii(x, y) = ii(x - 1, y) + s(x, y) \quad (2.3)$$

Information:
s(x, y) = cumulative number of rows
ii(x, y) = integral value of the image

Based on Figure 2.4, the integral value of the image at point 1 is the number of pixels in area A. The value at point 2 is the number of pixels in areas A and B, at point 3 is the number of pixels in areas A and C, while at point 4 is the number of pixels in areas A, B, C, and D. With this explanation, the following equation is obtained.

Point 1 = A, Point 2 = A, Point 3 = A+B+C+D (2.4)

Point 1+Point 4 – Point 3 = A+A+B+C+D-A-B-A-C = D (2.5)

Equation 2.2 illustrates how a feature value s(x,y)s(x,y,y)s(x,y) is calculated based on previous data in an image matrix. Here, s(x,y)s(x,y)s(x,y) represents the value at the position x,yx,yx,y, while s(x,y−1)s(x,y-1)s(x,y−1) is the previous value in the same line. The value of i(x,y)i(x, y)i(x,y) is the contribution given by the position of x,yx, yx,y in the matrix (Prathivi & Kurniawati, 2020).
In general, this equation describes how the value of i(x,y)i(x, y)i(x,y) is accumulated along the line yyy to calculate the value of s(x,y)s(x,y)s(x,y) at each position x,yx, yx,y. This process serves as a filter or data processing operation that is often used in image processing.
In addition, ii(x,y)ii(x,y)ii(x,y) refers to the *integral image*, which is calculated from the number of pixel values in the area detected by the Haar feature. To detect faces, the system uses the Haar feature by scanning images using threshold values. This threshold serves to convert grayscale images into binary images. The Haar feature consists of two parts: the black part and the white part. The threshold value is obtained from the sum of the black and white parts, according to equation 2.6.

$$Putih = \frac{Jumlah\ Keseluruhan\ Piksel - Jumlah\ Hitam}{Dimensi\ Gambar} \quad (2.6)$$

$$Hitam = \frac{Jumlah\ Keseluruhan\ Piksel - Jumlah\ Putih}{Dimensi\ Gambar} \quad (2.7)$$

$$threshold_{fitur} = Putih + Hitam \quad (2.8)$$

The threshold in equation 2.8 is an equation of one type of haar feature. To calculate the entire threshold the following equation is used.

$$threshold_{total} = \frac{threshold_1 + threshold_2 + \ldots fitur_n}{} \quad (2.9)$$

Algoritma Haar Cascade terdiri dari empat tahap pemrosesan gambar meliputi pemilihan fitur Haar, pembuatan integral gambar, pelatihan AdaBoost dan pengklasifikasi berjenjang seperti yang tersaji pada Gambar **2.5.** Algoritma ini merupakan pendekatan berbasis *machine learning*.

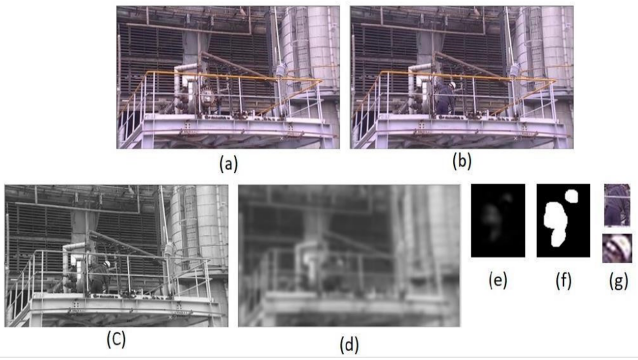


Figure 5. Example of image processing (a) the first background image; (b) images

In the face detection process, the image taken by the camera will be converted into a grayscale image as seen in Figure 2.5. In grayscale conditions, the ESP32-CAM will detect facial features using the Haar Cascade algorithm. Images that have been processed through this algorithm will result in face detection, which is shown in Figure 2.6. In the use of the Haar Cascade algorithm, images are categorized into two types: positive imagery and negative imagery. Positive imagery contains objects that need to be detected (e.g. faces), while negative images do not contain objects that need to be recognized (Phuc et al., 2019).

The detection process involves several stages, including image conversion to grayscale, Gaussian blur implementation,

and frame reduction using methods such as cv2.absdiff. The results of these steps are further processed through threshold techniques to produce images that can be used for further object detection.

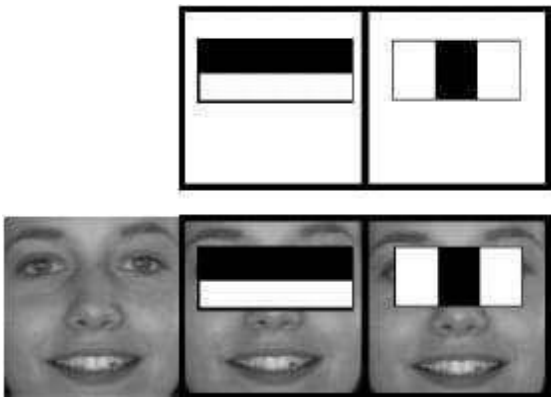


Figure 6. Haar feature on facial objects

The face detection procedure generally involves three main steps. First, the Haar cascade classifier algorithm uses various Haar features to detect and recognize important elements in the input image. Second, this algorithm introduces the concept of integral drawing, which allows for quick calculation of features. Furthermore, using AdaBoost's learning algorithm, only a small number of distinctive visual features are selected from the many potential features that exist. Finally, the classifier is structured in the form of a cascade, which allows for quick removal of the background of the image and focus on areas that are more likely to contain the face.

C. Internet of Things (IoT)

The Internet of Things (IoT) is a concept in which everyday objects such as electronic devices, vehicles, and buildings are equipped with sensors, software, and internet connectivity to collect and share data. IoT allows these objects to communicate and interact autonomously, supporting advanced data monitoring, control, and analysis capabilities. Some of the important aspects of IoT include concepts and architectures that involve networks of interconnected devices with data sharing capabilities, IoT applications in various sectors, as well as security and privacy issues, such as data protection through encryption, access authorization, and protection from cyberattacks. Additionally, IoT leverages a variety of networks such as Wi-Fi, Bluetooth, Zigbee, LoRa, and NB IoT, as well as protocols such as MQTT, CoAP, and HTTP (Popli et al., 2019).

III. METHOD

The research method used in this study is qualitative descriptive, which consists of several main stages as follows:

A. Data Collection

At this stage, data is collected to support the development of facial recognition-based smart door security system software. Data collection techniques include:

1. Literature review: Understand relevant algorithms and protocols, such as Haar Cascade, MQTT, and integration with ESP32-CAM.
2. Study of the technical documents: Identify the ESP32-CAM specifications and required software libraries.

B. Software Needs Analysis and Design

The objective of this stage is to establish the functional and non-functional needs of the developed software, which includes:

1. Functional needs: Face detection, face database management, device control (solenoids, LEDs), and notification delivery via the Blynk app.
2. Non-functional needs: Process speed, data security, and memory usage efficiency.
3. Software design: Design system workflows, facial detection flowcharts, device controls, and data communication with Blynk applications, and define software architecture.

C. Software Development

At this stage, the implementation of the design that has been made is carried out, which includes:

1. Implementation of the Haar Cascade algorithm for face detection on the ESP32-CAM.
2. Use of software libraries such as OpenCV for image processing and PubSubClient for the MQTT protocol.
3. Integration with Blynk apps for device control and notification delivery.
4. Software optimization for memory efficiency and processing speed.

D. Authors and Affiliations

This stage aims to test the performance of the software, with the following test methods:

1. Functional testing: Ensures features such as face detection, device controls, and notification delivery are up to specification.
2. Performance testing: Measures the speed of face detection and the system's response to Blynk application commands.
3. Fault testing: Identifies and fixes weaknesses, such as face detection in low lighting or unstable Wi-Fi connections.

E. Working Steps

In the software design stage, the first step is to create a flowchart that describes the process of facial recognition, door lock control, and notification to users. ESP32- CAM serves as the main processor that handles the facial recognition process and organizes other devices. Notifications related to access status will be sent via Blynk and Telegram applications to the user's smartphone,

A flowchart is a diagram that uses specific symbols to show the sequence of steps in a process, as well as the relationships between those processes. Using flowcharts, a series of procedures can be described clearly and easily understand, as well as facilitate the visualization of the steps in the program. A flowchart of this system can be seen in the following image.

A flowchart is a diagram that uses specific symbols to show the sequence of steps in a process, as well as the relationships between those processes. Using flowcharts, a series of procedures can be described clearly and easily understand, as well as facilitate the visualization of the steps in the program. A flowchart of this system can be seen in the following image.

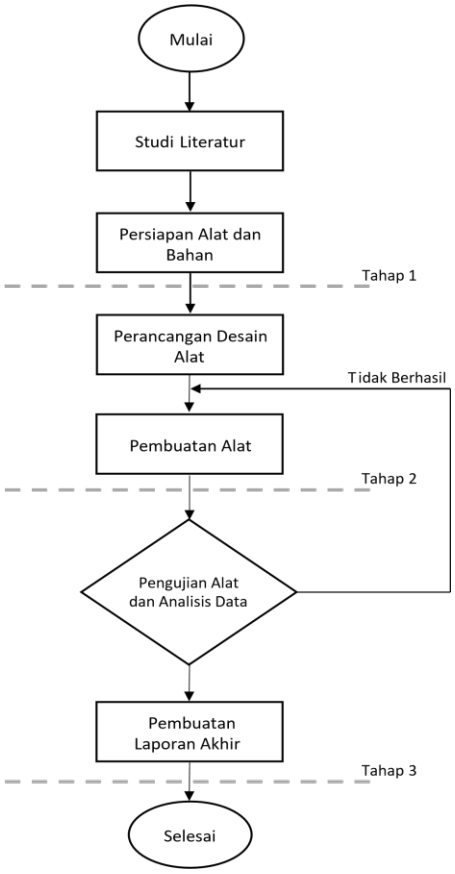


Figure 1. Flowchart Flowchart

1. Initial Preparation

a) Install the Arduino IDE and add the ESP32 board.
b) Download the required library, including for the ESP32 camera, Wi-Fi, and the Blynk app.
c) Test the ESP32-CAM connection with the computer to make sure the device is working properly.

2. Build a Face Detection System

a) Use the Haar Cascade algorithm to recognize faces
b) Set the camera resolution to 320x240 for faster and more efficient processes.
c) The logic program compares faces with stored databases.

3. Storing Face Data
   a) Create a simple database to store legitimate face data, such as using arrays or small files in ESP32 memory.
   b) Implement face verification logic so that only the matched are given access.
4. Integration with Blynk App
   a) Create an account in Blynk and design dashboards for device controls such as solenoids and notifications.
   b) The ESP32-CAM connection to Blynk uses the provided token.
   c) Test the connection and data communication between devices and applications.
5. Device Control (Solenoid and LED)
   a) The ESP32-CAM program to open the solenoid when the face is known.
   b) Use LEDs as indicators: green for familiar faces, red for unknown faces.
6. Wi-Fi Connection Management
   a) Connect the ESP32-CAM to a stable Wi-Fi network.
   b) Add an alert feature to Blynk if the Wi-Fi connection drops.
7. System Testing
   a) Face detection test under various lighting conditions and distances (20–60 cm).
   b) Verify the control function of the device (solenoid and LED) according to the condition of the face.
   c) Make sure the Blynk app provides notifications when faces are not recognized.
8. Improvement
   a) Bug fixes based on test results.
   b) Optimal programs to improve performance, such as reducing ESP32- CAM memory usage.
9. Final Implementation
   a) Upload the final software to the ESP32-CAM.
   b) Test the system in real conditions to ensure optimal functionality.

RESULTS AND DISCUSSION

Testing of the "Smart Door" security system based on "face recognition" with the ESP32-CAM was carried out to realize functionality and performance. The test included real-time detection and facial recognition using the Haar Cascade method, taking into account distance factors, facial manipulation, and the use of photos as inputs.

1. Testing at a distance of 20 cm and 40 cm

At a distance of 20 cm, the system can detect and recognize faces well, indicating a high level of accuracy. However, at a distance of 40 cm, although the face is still detectable, recognition becomes slightly less accurate, due to the limited resolution of the image.

a) Distance 20 cm : Optimal face detection and recognition (✓).
b) Distance 40 cm : Face detection still works (✓), but sometimes recognition fails (✎).

2. Testing with Facial Manipulation

Facial manipulation, such as the use of glasses or hats, affects the accuracy of recognition. The system can still detect faces, but it has difficulty in recognizing modified faces.

   a) Results : Faces are often detected (✓), but recognition fails (□) when there are significant physical changes.

3. Testing with Sample Face Photos

When using photos as input, the system can detect faces, but the recognition rate is increased compared to real face recognition. This shows that the Haar Cascade method cannot always accurately distinguish real faces from statistical images.

   a) Results : Faces from photos are often detected (✓), but recognition tends to fail (✓).

4. Webserver Notifications

The system provides feedback through the webserver in the form of the following notification:

   a) Face Detecting : Faces are detected and recognized correctly, accompanied by a "Door Open" message and username.

b) Undetectable Face : Faces go undetected, usually due to improper lighting conditions or positioning.
c) Unrecognized Face : Faces are detected but not recognized in the user's database.

From the test results, the system showed good performance in face detection at optimal distances and adequate lighting. However, facial acquaintances experience a decrease in accuracy when physical manipulation occurs or when using photos as input, indicating limitations in the accuracy system in handling such variations.

CONCLUSION

The "Smart Door" system based on "face recognition" with the ESP32-CAM shows good performance in detection and recognition at close distances with optimal lighting. However, facial recognition decreases in accuracy over longer distances, facial manipulation such as the use of accessories, and input in the form of facial photos. This indicates that although the system is effective under ideal conditions, there are still limitations in dealing with physical variations and different types of facial input. Furthermore, improving the quality of facial recognition can be done by improving image resolution and detection algorithms.

REFERENCES

[1] K. Kimani, V. Oduol, and K. Langat, “Cyber security challenges for IoT-based smart grid networks,” *International journal of critical infrastructure protection*, vol. 25, pp. 36–49, 2019.

[2] V. Kiran, S. Hooda, S. Dahiya, Y. P. S. Berwal, and R. Kamboj, “IoT-Based 5G Healthcare Systems with Blockchain for Improving the Security of Healthcare Monitoring System,” in *The International Conference on Recent Innovations in Computing*, Springer, 2023, pp. 809–826.

[3] H. Nasir, W. B. W. Aziz, F. Ali, K. Kadir, and S. Khan, “The implementation of IoT based smart refrigerator system,” in *2018 2nd International Conference on Smart Sensors and Application (ICSSA)*, IEEE, 2018, pp. 48–52.

[4] A. D. Dwivedi, G. Srivastava, S. Dhar, and R. Singh, “A decentralized privacy-preserving healthcare blockchain for IoT,” *Sensors (Switzerland)*, vol. 19, no. 2, pp. 1–17, 2019, doi: 10.3390/s19020326.

[5] S. Wahyuni, A. Khaliq, H. M. Z. N. Amrul, and A. Akbar, “Innovation of The Sipemang Application Using Qr Code For Monitoring And Preserving Mangrove Ecosystems In Pari City Village,” *Journal of Information Technology, Computer Science and Electrical Engineering*, vol. 1, no. 3, pp. 172–180, 2024.

[6] S. Wahyuni, A. Khaliq, H. M. Z. N. Amrul, and A. Akbar, “Innovation of The Sipemang Application Using Qr Code For Monitoring And Preserving Mangrove Ecosystems In Pari City Village,” *Journal of Information Technology, computer science and Electrical Engineering*, vol. 1, no. 3, pp. 172–180, 2024.

[7] S. Wahyuni, A. Khaliq, H. M. Z. N. Amrul, and A. Akbar, “Designing a Website-Based Kota Pari Village Mangrove Application with the Agile Scrumban Method,” in *International Conference on Artificial Intelligence, Navigation, Engineering, and Aviation Technology*, 2024, pp. 415–419.

[8] Patel, A., et al. (2022). "Face Detection Using Haar Cascade." *International Journal of Computer Vision* .

[9] Gomes, R., et al. (2021). "ESP32 Integration with IoT Applications." *IoT Journal.*

[10] Turki, M., & Pentland, A. (1991). "Eigenface for Introduction." Journal of Cognitive Neuroscience .