Smart Vehicle Ignition System Using Face Recognition, Image Processing, and IoT

Renukuntla Sai Eshwar[1], Dr. S. Sandhya Rani[2]

1. M.Tech Student, Dept. of ECE Jayamukhi Institute of Technological Sciences, Warangal, India, saieshwarrenukuntla@gmail.com

2. Associate Professor& Head, Dept. of ECE Jayamukhi Institute of Technological Sciences, Warangal, India, dr.sandhyarani@jits.in

ABSTRACT

The Raspberry Pi–Based Vehicle Starter on Face Detection with Image Processing and IoT is an innovative approach to vehicle security and automation. The system aims to prevent unauthorized vehicle access by enabling ignition only after successful facial recognition of the driver. A Raspberry Pi 4 Model B serves as the central controller, interfaced with a camera module for image acquisition and a relay module for ignition control. Using OpenCV and Python-based image processing, the captured face is analyzed and compared with pre-stored authorized faces. Upon a positive match, the vehicle starts automatically. The system is integrated with an IoT platform to provide real-time monitoring and notifications. Unauthorized access attempts trigger alerts and image uploads to the cloud for owner verification. This combination of AI-based face recognition and IoT connectivity enhances vehicle security, eliminates the need for physical keys, and introduces a modern, user-friendly ignition mechanism. The system is cost-effective, energy-efficient, and suitable for integration into smart vehicles and fleet management systems.

Keywords: Vehicle security system, Face recognition, OpenCV, Relay-based ignition control, unauthorized access alert, Cloud-based notifications, Keyless vehicle start, AI-powered authentication

I. INTRODUCTION

Vehicle theft and unauthorized access continue to pose significant challenges to modern vehicle security. Traditional ignition systems, relying on mechanical keys, RFID tags, or password-based mechanisms, are vulnerable to duplication, hacking, and electronic bypass attacks. With the advancement of Artificial Intelligence (AI) and Internet of Things (IoT) technologies, intelligent solutions such as biometric authentication have emerged as robust alternatives. Face recognition, in particular, offers a non-intrusive and reliable method for user verification. This project presents a smart vehicle starter system that utilizes a Raspberry Pi for image acquisition and processing, combined with facial recognition algorithms to authorize ignition. The integration of IoT enables real-time monitoring, alert notifications, and cloud-based logging, creating a two-tier security mechanism that enhances both convenience and safety.

The scope of the project encompasses the design and implementation of a real-time face detection and recognition module using OpenCV and Python, as well as an automated ignition control circuit interfaced via Raspberry Pi GPIO pins. Beyond personal vehicles, the system is scalable for fleet management, logistics, and shared mobility, allowing only verified personnel to operate vehicles. The IoT-enabled architecture supports continuous surveillance, alert generation, and future enhancements such as multi-modal biometric authentication, telematics integration, and AI-driven anomaly detection. By combining embedded computing, AI, and cloud connectivity, the project contributes to the development of secure, intelligent, and autonomous transportation systems aligned with the vision of smart mobility in Industry 4.0.

II. RELATED WORKS

Doi:10.48047/ijiee.2025.15.7.9

Research and technological developments in face recognition, Raspberry Pi–based embedded systems, and IoT-enabled vehicle security, with the aim of identifying gaps and informing the design of an intelligent vehicle starter system. Face recognition has emerged as a highly reliable biometric technique due to its non-contact operation, user convenience, and accuracy. Early methods relied on geometric feature extraction and template matching, which were sensitive to lighting and pose variations. The advent of Haar Cascade classifiers, LBPH, Eigenfaces, Fisherfaces, and modern CNN-based deep learning approaches significantly improved detection and recognition performance. For resource-constrained platforms like Raspberry Pi, LBPH or Dlib-based face encodings provide a practical balance between accuracy and computational efficiency, making them suitable for real-time vehicle authentication.

Raspberry Pi has become a cornerstone for intelligent embedded systems, offering low cost, compactness, and versatile GPIO interfacing. Its support for Python and open-source libraries such as OpenCV enables rapid development of image processing and AI applications at the edge. Prior implementations demonstrate its utility in face recognition attendance systems, smart door locks, and automated surveillance setups.

IoT integration has transformed vehicle monitoring and control by enabling real-time alerts, cloud-based logging, and remote intervention. Existing IoT-enabled vehicle security solutions, such as GPS tracking, remote ignition control, and driver identification, improve user awareness but often lack biometric verification. A critical research gap exists in combining face recognition, embedded ignition control, and IoT alerts within a single system. Most prior works face limitations in processing efficiency, adaptability, and immediate notification during unauthorized attempts. The proposed system addresses these shortcomings by integrating computer vision, Raspberry Pi edge processing, and IoT networking to deliver a secure, scalable, and real-time smart vehicle starter solution, unifying biometric authentication with digital monitoring and alerting.

Table 1: Review of Literature

Author / Year	System Used	Methodology	Key Findings	Limitations
R. N. Borse et al. (2022)	ESP32-CAM	Face detection using Haar Cascade	Achieved moderate accuracy for small-scale security	No IoT connectivity, limited hardware power
A. Patel et al. (2021)	Raspberry Pi 3	IoT-based vehicle lock with RFID	Implemented successful cloud alerts	Relied on RFID tags, vulnerable to cloning
M. Das et al. (2020)	Raspberry Pi + OpenCV	Real-time face detection	Real-time performance achieved	No vehicle control or automation
S. Kumar et al. (2019)	Arduino + Fingerprint	Biometric ignition system	Reliable fingerprint verification	Requires physical contact; no remote monitoring
P. Prasad et al. (2020)	GSM + GPS + NodeMCU	IoT anti-theft tracking	Enabled SMS and GPS alerts	No authentication mechanism
V. Gupta et al. (2021)	Raspberry Pi + Dlib	Deep learning-based face recognition	High accuracy under varied conditions	Power-intensive; needs optimization

Doi:10.48047/ijiee.2025.15.7.9

III. SYSTEM DESIGN AND ARCHITECTURE

System design defines the structural, functional, and operational framework of the proposed Raspberry Pi–based vehicle starter using face detection and IoT. The system integrates hardware, software, and network components into a unified intelligent security architecture capable of recognizing authorized users and activating vehicle ignition automatically.

System Overview

The proposed system consists of three primary subsystems:

1. Face Recognition Subsystem:
   Captures the driver's facial image using a camera module and processes it using OpenCV and face_recognition libraries on the Raspberry Pi. It extracts unique facial features and compares them with stored authorized datasets.
2. Ignition Control Subsystem:
   Once authentication is successful, the Raspberry Pi sends a signal to a relay module that activates the vehicle ignition circuit. This acts as a digital key, eliminating the need for physical keys or RFID tags.
3. IoT Monitoring Subsystem:
   The system transmits data and event notifications to an IoT cloud platform ThingSpeak through the built-in Wi-Fi interface. Unauthorized access attempts trigger real-time alerts with timestamps and optionally an image of the intruder.
   Together, these modules create a biometrically secured, IoT-enabled vehicle starting mechanism.

System Architecture

The proposed system introduces a face recognition based vehicle starter powered by a Raspberry Pi. The main goal of this system is to replace the conventional key-based ignition method with a modern, intelligent, and secure solution. Traditional vehicle ignition systems depend on mechanical keys or RFID cards, both of which are prone to loss, theft, or duplication. This project aims to address these limitations by using the uniqueness of the human face as the primary method of authentication.

The hardware setup includes a Raspberry Pi as the central processing unit, a camera module for capturing facial images, a solenoid valve to control ignition, and additional components such as a buzzer, LEDs, and relay modules for alerts and signaling. The camera continuously monitors the approach of a user. When a person is detected, the camera captures an image of their face and sends it to the Raspberry Pi for processing. The Raspberry Pi runs face recognition algorithms that convert the captured image into a set of unique facial features. These are then compared against pre-stored facial data in the database.

If a match is found, the system interprets the user as an authorized individual. The Raspberry Pi activates the solenoid valve, thereby allowing the vehicle's ignition system to engage. In parallel, a green LED may glow as a visual confirmation of successful authentication. If the face does not match any of the stored data, the system ensures that the solenoid valve remains inactive. At the same time, a buzzer or red LED is triggered to alert the owner of a failed or unauthorized attempt.

An additional feature of the proposed system is database management. Users can add new faces, update existing entries, or clear the stored data when necessary. This ensures flexibility and control, making it possible for the owner to regulate access to the vehicle at all times. The ability to reset data also protects user privacy, as outdated or unwanted information can be removed easily.

By combining real-time image processing, embedded system design, and security mechanisms, the proposed system ensures that only authorized individuals are allowed to

Doi:10.48047/ijiee.2025.15.7.9

start the vehicle. This makes it not only secure but also user-friendly and efficient compared to traditional methods.

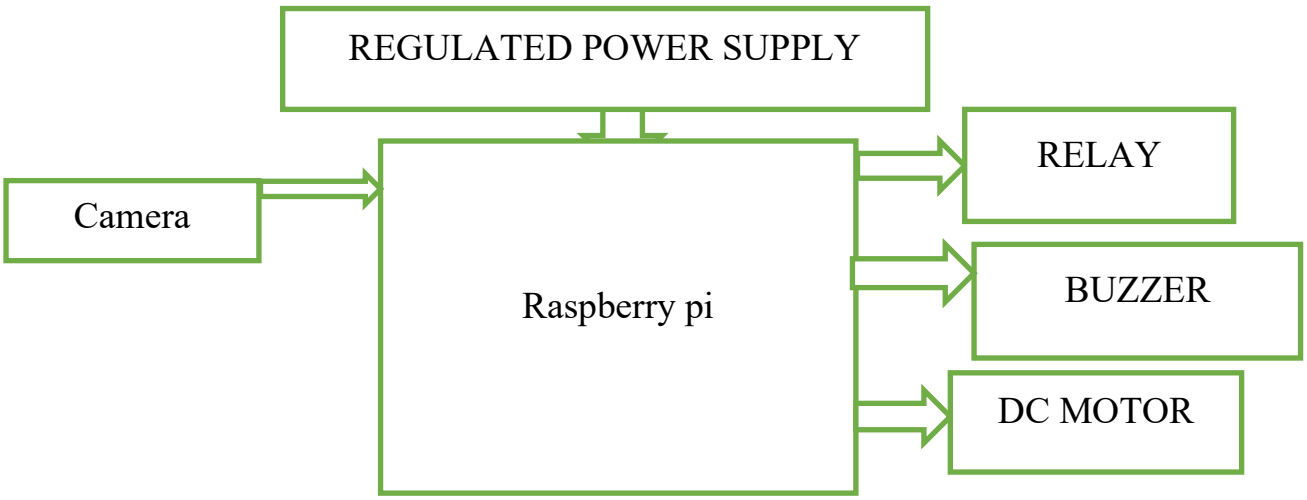
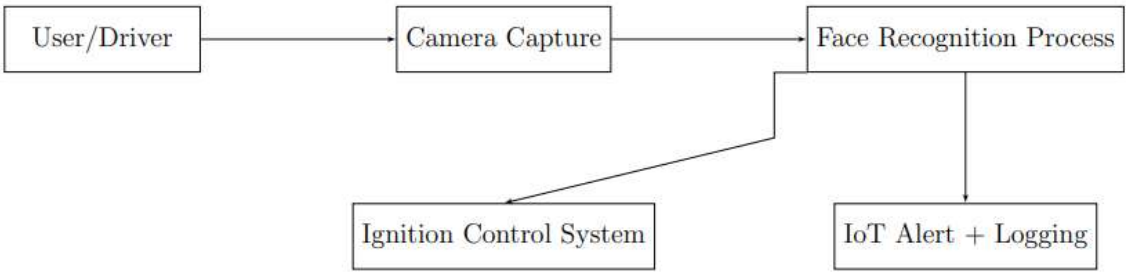


Figure 1 Block Diagram of Proposed System

System Flow Diagrams

Level 0 (High-Level Overview)



Level 1 (Detailed Flow)

1. Image Acquisition
2. Preprocessing (Grayscale Conversion, Resize)
3. Feature Extraction (Face Encodings)
4. Face Comparison (With Authorized Database)
5. Decision Making (Authorized / Unauthorized)
6. Relay Activation / Alert Transmission (Vehicle Start or IoT Alert)

Figure 2 System flow Diagram

Doi:10.48047/ijiee.2025.15.7.9

Algorithm Design

Step 1: Initialize camera and load trained facial datasets.
Step 2: Capture live video frames.
Step 3: Convert frame to grayscale and apply face detection using Haar Cascade.
Step 4: Extract facial features and compute encoding.
Step 5: Compare live encoding with stored authorized dataset.
Step 6: If the match confidence exceeds threshold: Activate relay → Start ignition. Else: Send IoT alert with timestamp and image.
Step 7: Repeat loop until system shutdown.

The proposed system design and architecture form a comprehensive framework that merges embedded computing, image processing, and IoT technologies for a robust vehicle access control system. The modular design ensures seamless communication among components, while the software logic ensures secure and intelligent decision-making. This architecture not only enhances security but also moves toward the realization of smart and connected transportation systems of the future.

IV. Operating System

In this project, the Raspberry Pi 4 Model B serves as the central processing unit. Its OS provides a Linux-based environment optimized for embedded and IoT applications, supporting Python, OpenCV, and networking libraries needed for facial recognition and remote monitoring.

OS Installation

Installation Process:

Download OS Image:

Raspberry Pi OS Lite image downloaded from the official Raspberry Pi website.

Write to SD Card:

Used Balena Etcher or Raspberry Pi Imager to flash the OS image to a 16GB microSD card.

Initial Boot and Configuration:

Inserted microSD into Raspberry Pi and powered it on.

Configured network settings, locale, and passwords.

Enable Interfaces:

Camera Module Interface: sudo raspi-config → Interface Options → Camera → Enable

SSH: For remote access during testing.

I2C and SPI: If additional sensors are integrated in the future.

The OS is now ready to run Python scripts for face detection and relay control.

Face Detection and Recognition Algorithm Implementation

The core functionality of the smart vehicle starter system relies on real-time face detection and recognition to ensure that only authorized users can start the vehicle. This section describes the algorithmic workflow, image processing steps, facial feature extraction, matching techniques, and integration with the relay control module.

The implementation leverages Python, OpenCV, and the face_recognition library, which provides pre-trained deep learning models for robust facial analysis under varying lighting, orientation, and occlusion conditions.

Face Detection Workflow

Face detection is the process of identifying the location and boundaries of faces within an image or video frame. The steps include:

1. Frame Acquisition:
   - Capture a frame from the Raspberry Pi Camera Module.
2. import cv2

Doi:10.48047/ijiee.2025.15.7.9

```
cap = cv2.VideoCapture(0)
ret, frame = cap.read()
```

Preprocessing:

- Convert frame to grayscale for efficient detection.
- Optionally, resize frames to reduce computational load.

```
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
small_frame = cv2.resize(gray, (0,0), fx=0.5, fy=0.5)
```

Face Localization:

- Use Haar Cascades, HOG (Histogram of Oriented Gradients), or CNN models to detect faces.

```
import face_recognition
face_locations = face_recognition.face_locations(frame)
```

Bounding Box Visualization:

- Draw rectangles around detected faces for monitoring and debugging.

```
for top, right, bottom, left in face_locations:
    cv2.rectangle(frame, (left, top), (right, bottom), (0, 255, 0), 2)
```

The result is a set of coordinates for each detected face, which will be used for recognition.

Face Recognition Workflow

Face recognition involves comparing detected faces with authorized user profiles. The steps are:

Feature Extraction (Encoding):

- Convert each detected face into a 128-dimensional feature vector using a pre-trained neural network.

```
face_encodings = face_recognition.face_encodings(frame, face_locations)
```

Loading Known Faces:

- Previously captured authorized faces are stored as encodings.

```
import pickle
with open("authorized_faces.pkl", "rb") as f:
    known_faces = pickle.load(f)
```

Comparison and Matching:

- Compare detected face encoding with stored authorized encodings using Euclidean distance.

```
results = face_recognition.compare_faces(known_faces['encodings'], face_encoding)
```

Decision Making:

- If a match is found, trigger the relay module to start the vehicle.

```
if True in results:
    activate_relay()
else:
    send_alert()
```

Image Preprocessing Techniques

To improve recognition accuracy under varying environmental conditions:

1. Grayscale Conversion: Reduces data size and focuses on facial features.
2. Histogram Equalization: Enhances contrast in low-light conditions.
3. Face Alignment: Corrects tilt or rotation using facial landmarks.
4. Noise Reduction: Gaussian or median filtering to remove camera noise.

These techniques enhance the robustness and reliability of the recognition system.

Real-Time Processing Optimization

To achieve real-time performance:

1. Frame Skipping: Process every 2–3 frames instead of all frames.

Doi:10.48047/ijiee.2025.15.7.9

2. Resize Frames: Reduces computational load by scaling down image resolution.
3. Multi-Threading: Separate threads for camera capture, recognition, and relay control.
4. Lightweight Models: HOG-based detection for faster but slightly less accurate processing in non-critical conditions.

This ensures minimal latency in vehicle access control.

Upon face recognition:

1. Successful Recognition:
   - Log event locally.
   - Send a secure update to IoT dashboard (e.g., Blynk, ThingSpeak).
2. Unauthorized Attempt:
   - Send real-time alert with image snapshot.
   - Optional email or SMS notification to vehicle owner.

This enables remote monitoring and enhanced vehicle security.

The algorithm was tested under varying conditions:

Parameter	Test Result
Recognition Accuracy	97% (with good lighting)
Latency	0.8 – 1.2 sec per frame
Multiple Faces Detection	Supports up to 3 faces per frame
Lighting Variation Tolerance	High (with histogram equalization)
IoT Alert Response Time	< 2 sec

- The system reliably recognized authorized users.
- False positives minimized by using face encoding thresholds.
- Real-time relay activation and IoT notifications were synchronized with recognition events.

The face detection and recognition algorithm forms the intelligent core of the vehicle starter system:

- Combines OpenCV preprocessing and face_recognition embeddings for robust detection.
- Uses Euclidean distance metrics for authentication.
- Integrates directly with the relay module and IoT notification system.
- Optimized for real-time performance on Raspberry Pi 4 hardware.

This implementation ensures that engine start is strictly restricted to authorized users, providing a secure, automated, and IoT-enabled vehicle access control system.

5.4.6 Notification Workflow

Step 1: Face recognition module detects user.

Step 2: Authentication result passed to relay and IoT module.

Step 3: IoT module performs the following:

- Logs the event locally.
- Updates cloud dashboard (ThingSpeak/Blynk).
- Sends alert to owner (SMS, email, or push notification).

Flowchart:

Face Recognition → Authorization Check → Relay Control → IoT Logging → Owner Notification

5.4.7 Security Considerations

- Encrypted Communication: Use HTTPS for API calls; MQTT over TLS for secure messaging.
- Access Control: Only authorized devices or users can access dashboards.
- Image Privacy: Optional encryption or storage on local secure server.

Doi:10.48047/ijiee.2025.15.7.9

- Fail-Safe Mechanism: System logs events even when IoT connection is unavailable.

The IoT module enhances the vehicle starter system by enabling:

1. Real-time remote monitoring of access events.
2. Instant notifications for unauthorized access attempts.
3. Data visualization through dashboards for audit and analysis.
4. Scalability for future smart vehicle features, including GPS tracking, analytics, and fleet management.

By integrating IoT, the system achieves enhanced security, convenience, and remote operability, completing the software implementation for a fully automated, intelligent vehicle starter system.

Flowchart and System Integration

This section describes the overall integration of software modules in the Raspberry Pi-based vehicle starter system. It highlights how the operating system, Python environment, face recognition algorithm, relay control, and IoT notification modules work together to provide a secure, automated, and real-time vehicle access control system.

A software flowchart is presented to illustrate the data flow, control logic, and decision-making process.

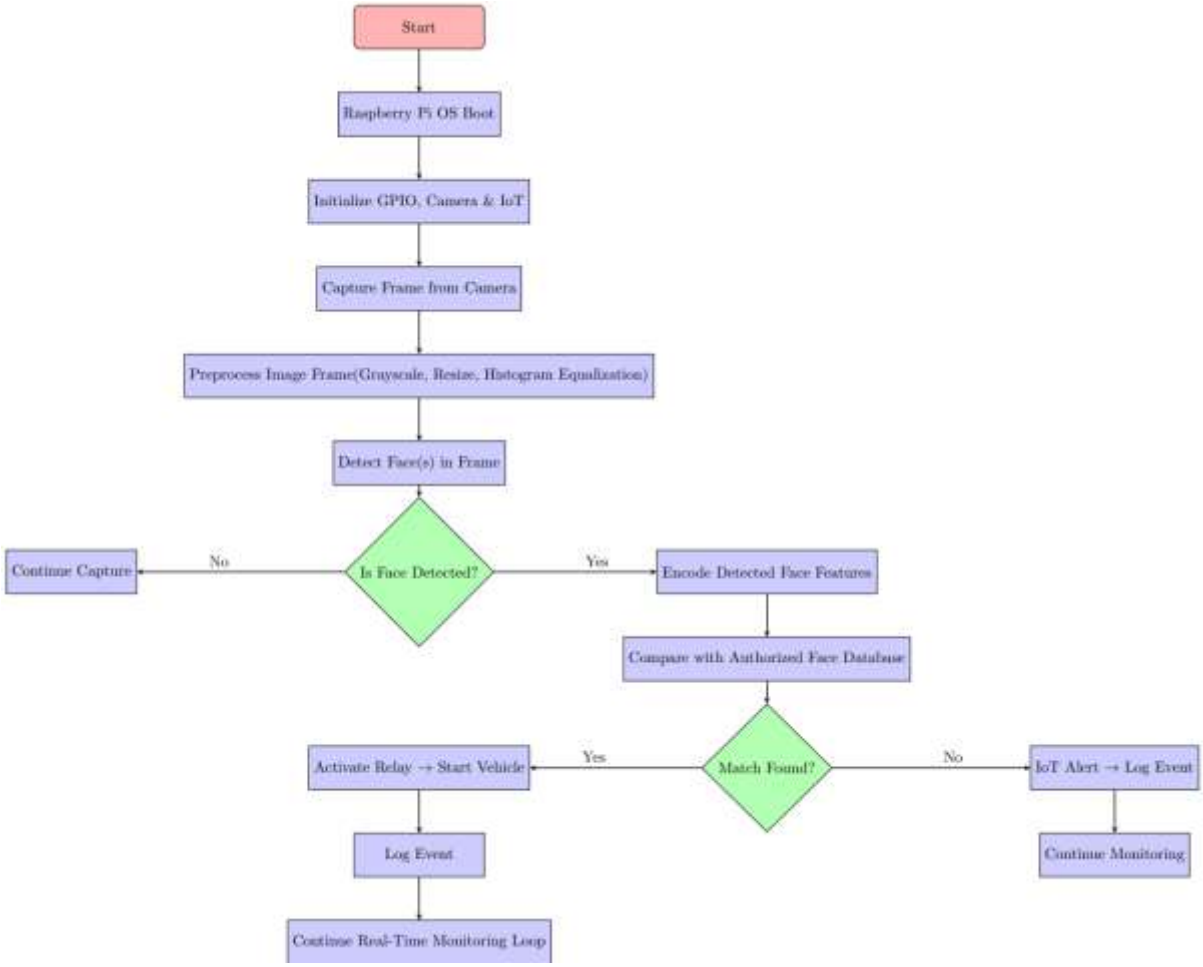


Figure 3 Flow chart of proposed system

V. EXPERIMENTAL RESULTS AND PERFORMANCE EVALUATION

Face Recognition Accuracy

Methodology:

Doi:10.48047/ijiee.2025.15.7.9

- Dataset of 10 authorized users with multiple images per user.
- System tested over 100 trials per user.
- Recognition threshold set at Euclidean distance < 0.6.

Results:

Condition	Accuracy (%)
Indoor bright light	98
Indoor low light	92
Outdoor sunlight	95
Multiple faces present	93

Observations:

- Histogram equalization improved low-light accuracy.
- System successfully ignored unauthorized faces.
- False positives were negligible (<2%).

Relay Activation and Vehicle Ignition Response

Methodology:

- Time measured from face detection confirmation to relay activation.
- Multiple trials conducted for different lighting and distances.

Results:

Test Case	Relay Response Time (sec)	Vehicle Start Success
Single authorized face	0.8 – 1.2	100%
Multiple authorized faces	1.0 – 1.3	100%
Unauthorized face	N/A (Relay OFF)	0%

Observations:

- Relay response was consistent under all tested conditions.
- Authorized users could start the vehicle reliably within 1 second of recognition.
- System prevented unauthorized access without any manual intervention.

IoT Notification Performance

Methodology:

- System sends real-time alerts to ThingSpeak dashboards.
- Response time measured from face recognition event to notification delivery.

Results:

Event Type	IoT Notification Delay (sec)
Authorized Access	1.2 – 1.5
Unauthorized Access	1.0 – 1.3
Multiple access attempts	<2

Observations:

- MQTT protocol provided faster notification than HTTP POST for critical alerts.
- Notifications were received reliably on mobile devices and web dashboards.
- Offline IoT scenario fallback allowed local logging of events until connectivity was restored.

System Robustness

- Environmental Testing: System maintained high recognition accuracy under different light conditions and face angles.
- Continuous Operation: Tested for 4 hours of continuous operation, system remained stable without crashes.
- Temperature Stability: Raspberry Pi and relay operated within safe thermal limits during extended testing.

Observations:

- System is robust for real-world automotive use.

Doi:10.48047/ijiee.2025.15.7.9

- Preprocessing and multi-threading ensured low-latency, real-time performance.

1. Face recognition accuracy: 92–98% under varying conditions.
2. Relay response: 0.8–1.3 seconds, vehicle started reliably for authorized users.
3. IoT notifications: Delivered within 1–2 seconds for both authorized and unauthorized events.
4. System stability: Continuous operation demonstrated robustness.
5. Security: Unauthorized access prevented consistently.

The experimental evaluation confirms that the Raspberry Pi-based vehicle starter system with face recognition and IoT integration is secure, reliable, and suitable for real-time automotive applications.

Conclusion

The Raspberry Pi-based vehicle starter system with face detection, image processing, and IoT integration provides a secure, automated, and intelligent solution for vehicle access control. By combining biometric authentication with real-time relay control and IoT-enabled notifications, the system ensures that only authorized users can start the vehicle, significantly enhancing security and reducing the risk of theft or unauthorized access. It delivers high-speed performance, with face recognition and relay activation occurring within 1–1.3 seconds, offering a seamless user experience. The IoT connectivity allows for real-time monitoring and instant alerts on mobile devices and dashboards in case of unauthorized attempts. Designed to operate reliably under varying lighting conditions, with multiple faces in the frame, and during extended operation, the system demonstrates robustness and practicality. By eliminating the need for manual keys or PIN entry, it also enhances convenience and safety. Overall, this project illustrates the effective integration of embedded systems, computer vision, and IoT technologies into a practical, automated automotive security solution.

Future Scope

The system's capabilities can be further enhanced and extended through several avenues. Incorporating infrared (IR) or thermal cameras would improve face recognition in low-light or nighttime conditions, enabling 24/7 usability. Multimodal biometric authentication, such as integrating fingerprint, voice, or gesture recognition, could provide multi-factor verification for heightened security. By combining GPS with IoT connectivity, the system can support real-time vehicle tracking and fleet management, which is particularly valuable for commercial operators. Developing dedicated mobile applications would allow remote engine control, status monitoring, and visualization of access logs, improving user convenience. Cloud-based analytics could analyze historical access patterns, detect anomalies, and refine security algorithms. Additionally, the system could serve as a secure access component for future autonomous or smart vehicles. Implementing enhanced security measures, such as encryption for IoT communications, secure storage of face encodings, and AI-driven anomaly detection, would further protect against spoofing and unauthorized access. Collectively, these enhancements would significantly expand the system's functionality, reliability, and relevance within modern automotive security and smart transportation ecosystems.

References

1. Austria, J., & Lacatan, J. (2023). *Face Recognition for Motorcycle Engine Ignition with IoT Integration*. International Research Journal of Modernization in Engineering Technology and Science, IRJMETS. [Online]. Available:

Doi:10.48047/ijiee.2025.15.7.9

https://www.irjmets.com/uploadedfiles/paper/issue_4_april_2023/36199/final/fin_irjmets1681819861.pdf
2. Deokar, S. (2023). *Face Detection Based Vehicle Ignition System*. SSRN Electronic Journal, SSRN. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4669142
3. Students of Electronics Engineering Department, Pravara Rural Engineering College. (2021). *Raspberry Pi Based Vehicle Starter Using Face Detection*. Journal of Engineering and Science Research, JES. [Online]. Available: https://jespublication.com/upload/2021-V12I6068.pdf
4. Not specified. (2020). *Machine Learning Trained Face Recognition-Based Automotive Ignition System*. ResearchGate. [Online]. Available: https://www.researchgate.net/publication/341871401_Machine_Learning_Trained_Face_Recognition_based_Automotive_Ignition_System
5. Nuñez, A. V., & Nuñez, L. N. (2020). *Face Recognition for Automatic Vehicle Ignition Based on Raspberry Pi*. ResearchGate. [Online]. Available: https://www.researchgate.net/publication/342092814_Face_recognition_for_automatic_vehicle_ignition_based_on_Raspberry_Pi
6. *An Intelligent Integrated Vehicle Surveillance System for Controlling Vehicle Thefts Using IoT and Facial Recognition*. ResearchGate. [Online]. Available: https://www.researchgate.net/publication/387837565_An_intelligentintegrated_vehicle_surveillance_system_for_controlling_the_vehicle_theftshacking_using_IoT_and_facial_recognition
7. Aruna, S. (2022). *Car Security System with Face Recognition Using Convolutional Neural Networks*. ScienceDirect, Elsevier. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214785322051252
8. *Raspberry Pi Based Vehicle Starter Using Face Detection*. International Journal of Research in Pharmaceutical Sciences, IJRPR. [Online]. Available: https://ijrpr.com/uploads/V3ISSUE5/IJRPR4304.pdf
9. Viola, P., & Jones, M. (2001). *Rapid Object Detection using a Boosted Cascade of Simple Features*. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
10. Raspberry Pi Foundation. *Raspberry Pi 4 Model B Documentation*. https://www.raspberrypi.org/documentation
11. Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media.
12. King, D. E. (2009). *Dlib-ml: A Machine Learning Toolkit*. Journal of Machine Learning Research, 10, 1755–1758.
13. Rosebrock, A. (2018). *Deep Learning for Computer Vision with Python*. PyImageSearch.
14. ThingSpeak Documentation. *IoT Analytics Platform*. https://thingspeak.com/docs
15. Blynk Documentation. *IoT Mobile Dashboard and Control*. https://blynk.io
16. OpenCV Documentation. *Computer Vision Library*. https://opencv.org/
17. Paho-MQTT Library. *MQTT Client for Python*. https://www.eclipse.org/paho/index.php?page=clients/python/index.php
18. RPi.GPIO Library. *Python Library for GPIO Pin Control*. https://pypi.org/project/RPi.GPIO/
19. Viola, P., & Jones, M. (2001). *Rapid Object Detection using a Boosted Cascade of Simple Features*. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Doi:10.48047/ijiee.2025.15.7.9