Helmet Detection and Biometric Vehicle Security System for Two-Wheelers

Thouti Meghana, M Sridhar
Department of Electronics and Communication Engineering
Jayamukhi Institute of Technological Sciences, Narsampet, Warangal, Telangana, India

Abstract
Road safety and vehicle security are important concerns in the widespread use of two-wheelers. Conventional ignition systems primarily authenticate access to a vehicle, whereas helmet compliance is commonly enforced through rider behaviour or external monitoring. This paper presents an embedded system that integrates machine-learning-based helmet detection with fingerprint-based rider authentication to control two-wheeler ignition. A Pi Camera captures an RGB image of the rider, and a YOLO-based model running on a Raspberry Pi 4 determines whether a helmet is present. A fingerprint sensor then verifies the rider against an enrolled user template. The two decisions are combined using an AND-based control rule: ignition is enabled only when helmet detection and biometric authentication are both successful. A single-channel relay interfaces with a low-voltage ignition simulator, while a buzzer and Wi-Fi-based IoT notification provide local and remote indication of failed attempts. The implementation configuration specifies 640 × 480 image input, ten authorized users, and 100 test attempts distributed across four helmet/authentication conditions. The supplied experimental tables label the reported values as tentative; they indicate 95.00% helmet-detection accuracy, 97.00% fingerprint-authentication accuracy, 98.00% ignition-decision accuracy, and an average end-to-end response time of approximately 1.646 s. These values are therefore presented as preliminary prototype figures rather than independently verified performance. The integrated architecture offers a practical basis for combining rider-safety enforcement and vehicle-access control on an embedded platform.

Keywords: Helmet detection; YOLO; biometric authentication; fingerprint recognition; Raspberry Pi 4; two-wheeler security; ignition control; relay; Internet of Things.

1. Introduction

Two-wheelers provide accessible and economical transportation, but riders are more exposed to injury because they lack the protective structure available in enclosed vehicles. Road traffic injuries remain a major public-health concern. The World Health Organization identifies correct helmet use as an important intervention for reducing motorcycle-related fatalities and brain injuries [1], [6]. Despite legal requirements, helmet compliance may still depend on individual behaviour and enforcement after a violation has occurred.

A vehicle-start interlock offers a preventive alternative by checking helmet use before ignition is enabled. Camera-based machine learning can analyze the rider's head region and estimate whether a helmet is present. In parallel, biometric authentication can restrict vehicle operation to enrolled users. Combining these functions makes vehicle operation conditional on both a safety requirement and an identity requirement.

Conventional two-wheeler security mechanisms generally rely on keys or other access controls and do not automatically verify helmet use. Conversely, traffic-surveillance systems can identify helmet violations but are not necessarily connected to the ignition circuit of an individual vehicle. Biometric vehicle-security systems can restrict unauthorized access but may not ensure that the rider is wearing a helmet. The problem addressed in this work is the absence of a unified embedded mechanism that verifies both conditions before permitting vehicle operation.

The proposed system uses a camera and a machine-learning model to produce a binary helmet decision, H, and a fingerprint sensor to produce an authorization decision, A. The Raspberry Pi evaluates both decisions and controls a relay. Ignition is enabled only when H=1 and A=1. If either decision is negative, ignition remains OFF and the system activates a buzzer and can generate an IoT alert.

The objectives of the paper are

- Develop a camera-based helmet-detection module using a machine-learning model.
- Implement fingerprint-based authentication for enrolled vehicle users.
- Integrate helmet and biometric decisions using an AND-based ignition-control rule.
- Use Raspberry Pi as the central embedded processing and control platform.
- Control a relay-based ignition simulator and generate a buzzer warning for invalid attempts.
- Provide Wi-Fi connectivity for event notification and logging.
- Evaluate helmet detection, fingerprint authentication, ignition decisions, and response time under the specified test conditions.

The system-level contribution is the integration of two independent verification functions into a single

Doi:10.48047/ijiee.2026.16.2.68

ignition interlock. The helmet detector supplies the rider-safety condition, while fingerprint matching supplies the access-control condition. The Raspberry Pi coordinates image inference, biometric input, decision logic, relay switching, and alert generation. Unlike a standalone helmet detector or standalone biometric lock, the integrated architecture prevents ignition when either safety or authorization is not satisfied.

2. Related work

Helmet detection has progressed from manually engineered image descriptors to deep-learning-based detection. Chiverton investigated helmet-presence classification in conjunction with motorcycle detection and tracking [7]. Silva et al. combined Circle Hough Transform, Histogram of Oriented Gradients, and Local Binary Patterns with a Random Forest classifier [8]. These methods demonstrated the feasibility of automatic helmet recognition, although handcrafted features can be sensitive to changes in viewpoint, lighting, and background.

Deep convolutional networks have enabled systems to learn visual features directly from training data. Siebert and Lin applied deep learning to motorcycle helmet-use detection in traffic video and highlighted the challenge of generalizing across observation sites [10]. YOLO-family detectors and attention-based models have subsequently been considered for real-time helmet recognition, including work on YOLOv7 and enhanced YOLOv8 approaches [11], [14]. Lightweight architectures remain important for embedded deployment because inference must be performed within limited processing and power budgets.

Biometric systems use physiological or behavioural characteristics to verify identity. Fingerprint sensors are widely used in embedded access-control applications because a compact sensor can capture a fingerprint and compare it with an enrolled template. Suthar et al. described a multilayer vehicle-access mechanism using fingerprint and face identification with Raspberry Pi [16]. Austria et al. investigated face recognition for motorcycle engine ignition with messaging functionality [17], while Vinothkumar et al. developed a Raspberry Pi-based vehicle-security system using face recognition [18].

More recent work has explored multimodal biometric authentication, GPS tracking, and IoT notifications for vehicle security [19]. Such systems show that biometric access control can be combined with remote event reporting. Fingerprint authentication nevertheless has practical limitations: wet, dirty, or damaged fingers may affect acquisition, and false acceptance or rejection must be measured under representative conditions.

The literature includes helmet-detection systems primarily intended for traffic surveillance and violation monitoring, and biometric systems primarily intended to prevent unauthorized vehicle access. These functions are often treated separately. The proposed system uses the helmet result as an ignition interlock condition and combines it with fingerprint authentication. This design connects a computer-vision safety decision directly to embedded vehicle-control logic.

Research direction	Typical technique	Limitation addressed by this work
Helmet recognition	HOG/LBP, CNN, YOLO, attention-based detectors	Detection result used as an ignition prerequisite
Biometric vehicle access	Fingerprint or face recognition	Authentication combined with helmet compliance
IoT vehicle security	Wi-Fi/GSM/GPS notifications	Failed helmet or fingerprint checks trigger local and remote alerts
Proposed architecture	YOLO helmet detection + fingerprint + Raspberry Pi + relay	Unified safety and authorization decision

3. Proposed system architecture

The system comprises a Pi Camera, Raspberry Pi 4, YOLO-based helmet-detection model, fingerprint sensor, single-channel relay, buzzer, Wi-Fi interface, and a low-voltage DC motor or ignition simulator for laboratory evaluation. The camera supplies RGB images at 640 × 480 resolution. The Raspberry Pi performs image processing and model inference, receives the fingerprint result, and executes the decision logic. The relay controls the test load, while the buzzer and IoT interface indicate unsuccessful attempts.

Doi:10.48047/ijiee.2026.16.2.68

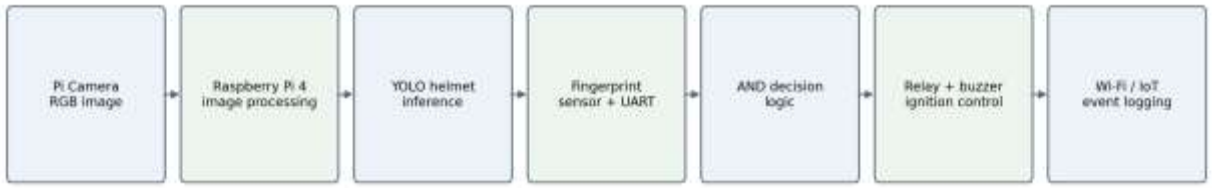


Fig. 1. Integrated architecture of helmet detection, biometric authentication and ignition control.

3.1 Functional modules

- Image acquisition: Pi Camera captures the rider image.
- Helmet inference: YOLO model estimates helmet presence from the image.
- Biometric verification: fingerprint sensor compares the input with an enrolled template.
- Decision logic: Raspberry Pi combines the binary outputs using a logical AND operation.
- Ignition control: relay enables the test ignition/load only for a valid combined decision.
- Alert and logging: buzzer provides local warning and Wi-Fi transmits an event notification.

3.2 Hardware configuration

Component	**Specified function**
Raspberry Pi 4	Central processing, YOLO inference, authentication coordination and GPIO control
Pi Camera	RGB image acquisition at 640 × 480 pixels
Fingerprint sensor	Capture and matching of enrolled fingerprint templates
Single-channel relay	Switching interface for the low-voltage ignition simulator
Buzzer	Local indication of helmet or authentication failure
Wi-Fi	IoT alert transmission and event logging
DC motor / ignition simulator	Laboratory load representing the vehicle ignition state
Power supply	Supplies the embedded platform and peripheral modules

The relay provides an electrical switching interface between the Raspberry Pi control output and the low-voltage test load. The source implementation explicitly cautions against connecting a Raspberry Pi GPIO pin directly to a vehicle ignition circuit. A properly rated automotive interface, isolation, protection, and qualified electrical installation are required for any real vehicle integration.

3.3 Software environment and implementation configuration

The implementation uses Python on Raspberry Pi, OpenCV for image handling, Picamera2 for camera acquisition, the Ultralytics YOLO interface for helmet inference, RPi.GPIO for relay and buzzer control, and serial communication for the fingerprint sensor. The supplied code specifies relay GPIO 17, buzzer GPIO 27, fingerprint serial port /dev/serial0 at 57600 baud, model file helmet_model.pt, and a helmet confidence threshold of 0.50. The IoT endpoint is represented by a configurable local API URL and must be replaced with the actual server endpoint.

Parameter	**Implementation value**
Processing platform	Raspberry Pi 4
Image format and resolution	RGB, 640 × 480
Helmet model interface	Ultralytics YOLO
Model file	helmet_model.pt
Detection confidence threshold	0.50
Relay / buzzer GPIO	BCM 17 / BCM 27
Fingerprint interface	/dev/serial0, 57600 baud
Communication	Wi-Fi
Test load	Low-voltage DC motor / ignition simulator

Doi:10.48047/ijiee.2026.16.2.68

3.4 Overall operating sequence

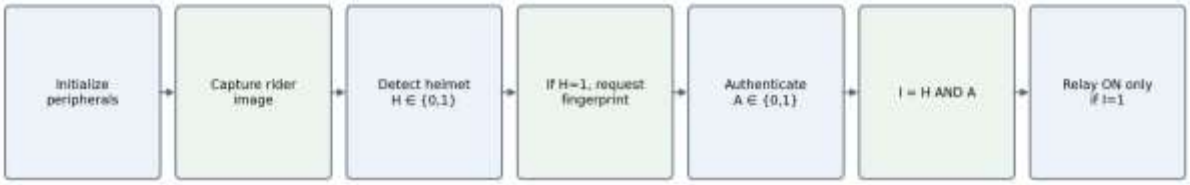


Fig. 2. Operating sequence from image capture to conditional ignition activation.

At startup, the Raspberry Pi initializes the camera, fingerprint interface, relay, buzzer, and Wi-Fi connection. It then captures the rider image and executes helmet inference. If no helmet is detected, the relay remains OFF and an alert is generated. If a helmet is detected, the system requests a fingerprint. A successful match activates the relay; an unsuccessful match keeps the relay disabled and triggers an alert.

4. Methodology

4.1 Image acquisition and preprocessing

The Pi Camera captures an RGB image I(x,y) of the rider. The acquired image is passed to the computer-vision pipeline. Preprocessing may include resizing, pixel normalization, noise reduction, region-of-interest extraction, and conversion to the input tensor format required by the trained model. The objective is to maintain a consistent image representation between model training and embedded inference.

4.2 Machine-learning-based helmet detection

The helmet-detection model receives the preprocessed image and generates a detection result. The project implementation uses the Ultralytics YOLO interface and loads the trained weights from helmet_model.pt. A confidence threshold of 0.50 is specified in the supplied code. The model output is converted to a binary helmet variable H:

$$H = f\theta(I), \quad H \in \{0,1\}$$

where I is the input image, fθ represents the trained model with parameters θ, and H=1 denotes helmet detected while H=0 denotes helmet not detected. If the model produces a helmet confidence score PH, the threshold decision is:

$$H = 1, \text{ if } PH \geq TH; \text{ otherwise } H = 0$$

The selected confidence threshold affects the trade-off between false helmet detections and missed helmet detections. Its suitability should be evaluated using a separate validation set rather than selected solely from test results.

4.3 Fingerprint registration

Authorized users are enrolled before vehicle operation. During registration, a fingerprint is captured, processed by the biometric sensor, and stored as a reference template. The system configuration identifies ten authorized users. The biometric template is subsequently used during an ignition attempt to compare the newly captured fingerprint with the enrolled reference.

4.4 Fingerprint authentication

When helmet detection is positive, the system requests a fingerprint sample Fi. The fingerprint sensor compares the input with the registered template Fr and returns a match result or matching score SF. The authentication decision is represented as:

$$A = 1, \text{ if } SF \geq TF; \text{ otherwise } A = 0$$

where TF is the authentication threshold, A=1 indicates an authorized user, and A=0 indicates an unsuccessful authentication. The supplied implementation uses a serial fingerprint interface and optionally imports the PyFingerprint library to communicate with the sensor.

4.5 Ignition decision and relay control

The central decision rule is the logical AND of helmet detection and fingerprint authentication:

$$I = H \wedge A$$

The ignition output I is set to 1 only when both H and A equal 1. In every other case, I=0. This prevents a helmeted but unauthorized rider and an authorized but non-helmeted rider from starting the test vehicle.

Doi:10.48047/ijiee.2026.16.2.68

4.6 Decision table and alert generation

Helmet H	Authentication A	Ignition I	System action
0	0	0	Ignition OFF; buzzer and IoT alert
0	1	0	Ignition OFF; buzzer and IoT alert
1	0	0	Ignition OFF; buzzer and IoT alert
1	1	1	Ignition ON; buzzer OFF

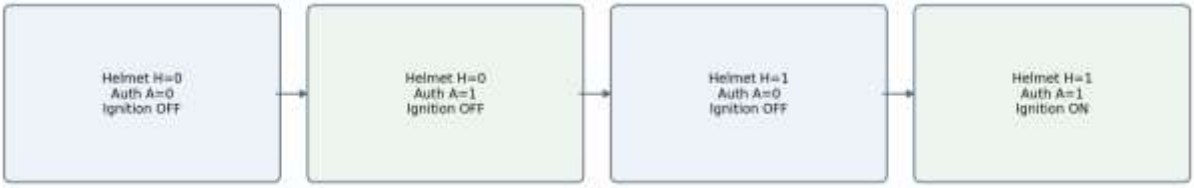


Fig. 3. Complete binary decision space for the ignition-control output.

When H=1 and A=1, the Raspberry Pi drives the relay-control output to enable the low-voltage ignition simulator. If either condition fails, the relay remains in the OFF state. The buzzer provides immediate local feedback, and the Wi-Fi interface can transmit an event for remote logging. A communication failure should not alter the safe default state of the ignition interlock.

4.7 Algorithm

1. Start the Raspberry Pi and initialize the camera, fingerprint sensor, relay, buzzer, and Wi-Fi interface.
2. Capture the rider image using the Pi Camera.
3. Preprocess the image and execute the trained YOLO helmet-detection model.
4. Convert the model output into helmet status H using the specified confidence threshold.
5. If H=0, keep ignition OFF, activate the buzzer, generate an IoT alert, and restart the verification cycle.
6. If H=1, request and capture the rider fingerprint.
7. Compare the fingerprint with the enrolled template and determine authentication status A.
8. If A=0, keep ignition OFF, activate the buzzer, generate an IoT alert, and restart the verification cycle.
9. If A=1, activate the relay and enable the low-voltage ignition simulator.
10. Record the event and repeat the process for the next vehicle-start attempt.

The ignition output should default to OFF during boot, sensor failure, missing camera frames, failed fingerprint communication, software exceptions, or loss of reliable decision data. The embedded controller must not energize a relay based on stale or incomplete results. A hardware interlock and appropriately rated automotive relay/driver should be used for any real vehicle integration. The supplied code itself specifies laboratory use of a low-voltage DC motor/load and warns against direct connection of Raspberry Pi GPIO to vehicle ignition wiring.

5. Experimental setup and evaluation

5.1 Experimental configuration

The supplied experimental configuration identifies a Raspberry Pi 4, Pi Camera, RGB input at 640 × 480 pixels, fingerprint authentication, single-channel relay, DC motor/ignition simulator, buzzer, Wi-Fi, and AND-based decision logic. Ten authorized users and 100 test attempts are specified. The four test conditions cover the complete combination of helmet status and user authorization.

Parameter	Value
Processing platform	Raspberry Pi 4
Camera input	RGB, 640 × 480
Helmet classes	Helmet / No Helmet
Biometric modality	Fingerprint
Authorized users	10
Test attempts	100
Trials per test case	25
Relay and load	Single-channel relay; DC motor/ignition simulator
Alerts	Buzzer and IoT notification
Communication	Wi-Fi

Doi:10.48047/ijiee.2026.16.2.68

5.2 Test-case distribution

Test case	Helmet status	User status	Trials
TC1	Helmet	Authorized	25
TC2	Helmet	Unauthorized	25
TC3	No helmet	Authorized	25
TC4	No helmet	Unauthorized	25
Total	—	—	100

The four cases evaluate both individual verification functions and the combined decision. TC1 is the only condition in which ignition should be enabled. TC2, TC3, and TC4 should result in ignition OFF, buzzer activation, and an IoT alert.

5.3 Helmet-detection results

Actual condition	Correct	Incorrect	Accuracy
Helmet	48	2	96.00%
No helmet	47	3	94.00%
Overall	95	5	95.00%

The table gives an overall helmet-detection accuracy of 95.00% across 100 test images, comprising 50 helmet and 50 no-helmet examples. The source also lists precision of 94.12%, recall of 96.00%, and F1-score of 95.05%. The accompanying source text explicitly characterizes these metrics as tentative classification results rather than verified measured performance.

5.4 Fingerprint authentication results

Authentication condition	Correct	Incorrect	Accuracy
Authorized fingerprint	49	1	98.00%
Unauthorized fingerprint	48	2	96.00%
Overall	97	3	97.00%

The fingerprint results list 97.00% overall accuracy, with 98.00% for authorized fingerprint trials and 96.00% for unauthorized fingerprint trials.

5.5 Ignition-control results

Test case	Helmet	Authentication	Expected	Observed	Buzzer	IoT alert
TC1	Detected	Authorized	ON	ON	OFF	No
TC2	Detected	Unauthorized	OFF	OFF	ON	Yes
TC3	Not detected	Authorized	OFF	OFF	ON	Yes
TC4	Not detected	Unauthorized	OFF	OFF	ON	Yes

The project material lists 100 total ignition decisions, 98 correct decisions, and two incorrect decisions, corresponding to a tentative ignition-decision accuracy of 98.00%. It also lists 24/25 successful authorized ignition cases and 74/75 correct ignition-prevention cases.

5.6 Relay response and end-to-end latency

Relay trial	Response time (ms)
1	118
2	125
3	121
4	127
5	119
6	123
7	126
8	120
9	124
10	122
Average	122.5

Processing stage	Average time
Image acquisition	210 ms

Doi:10.48047/ijiee.2026.16.2.68

Image preprocessing	85 ms
YOLO helmet inference	420 ms
Fingerprint acquisition	610 ms
Biometric verification	180 ms
Decision processing	18 ms
Relay activation	123 ms
Total average response time	1.646 s

The average relay response is listed as 122.5 ms. The end-to-end processing table gives an average of 1.646 s, dominated by fingerprint acquisition (610 ms) and helmet inference (420 ms).

5.7 Alert-generation results

Condition	Expected buzzer	Observed buzzer	IoT alert
Helmet + authorized	OFF	OFF	No
Helmet + unauthorized	ON	ON	Yes
No helmet + authorized	ON	ON	Yes
No helmet + unauthorized	ON	ON	Yes

The supplied functional table records the expected buzzer state and IoT alert outcome for each of the four decision conditions. The overall performance summary lists alert-generation success as 99.00%

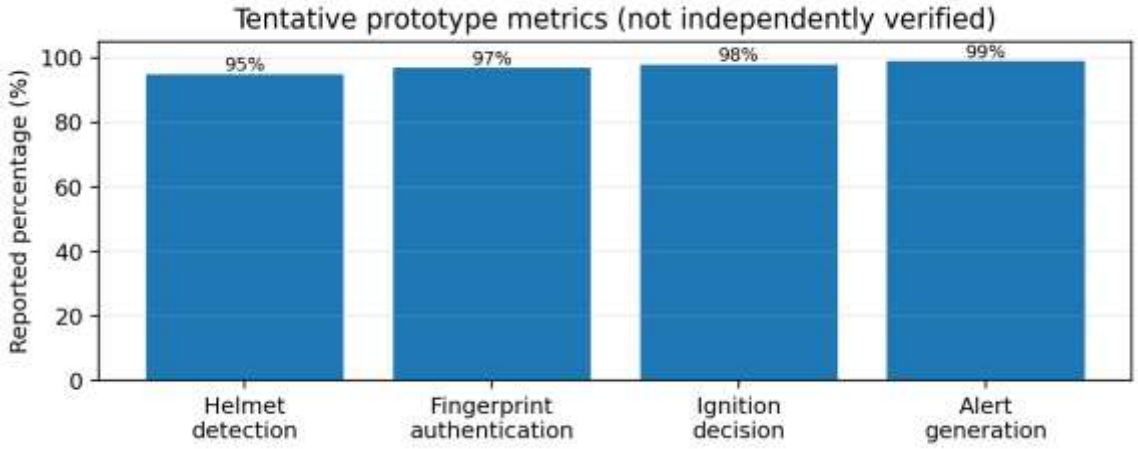


Fig. 4. Preliminary performance values reproduced from the supplied project material; all values are marked tentative.

6. Discussion

The system combines two complementary functions: computer-vision-based helmet recognition and fingerprint-based user authentication. The helmet detector determines whether the safety condition is satisfied, while the fingerprint subsystem checks whether the rider belongs to the enrolled user group. The AND-based control rule is straightforward and interpretable: neither condition alone is sufficient to activate ignition. This approach directly connects a visual safety decision with vehicle access control.

The preliminary values in the supplied material indicate helmet-detection accuracy of 95.00%, fingerprint-authentication accuracy of 97.00%, and ignition-decision accuracy of 98.00%. The reported average system response time is approximately 1.646 s, with an average relay response of 122.5 ms

.

6.1 Error sources and operational challenges

Error category	Possible cause	System consequence
False helmet detection	Poor lighting, unusual viewing angle	Unsafe start may be permitted
Missed helmet detection	Occlusion, blur, small helmet region	Authorized rider may be blocked
Fingerprint rejection	Wet, dirty or damaged finger	Authorized rider rejected
False fingerprint acceptance	Sensor or template-matching limitation	Unauthorized access risk

Doi:10.48047/ijiee.2026.16.2.68

Camera delay	Processing load or frame acquisition	Longer start-up time
IoT failure	Wi-Fi interruption or server unavailable	Remote alert may not arrive
Relay delay	Switching and driver characteristics	Ignition response delay

6.3 Practical implications

A vehicle-start interlock can encourage helmet use by preventing ignition when helmet presence is not detected. Fingerprint authentication can reduce unauthorized vehicle operation, while local and remote alerts provide feedback about failed attempts. The same architecture may be useful in delivery fleets, shared two-wheeler services, and institutional vehicle pools, provided that privacy, reliability, and operational safety are addressed.

8. Conclusion

A machine-learning-based helmet detection and biometric vehicle-security system for two-wheelers has been presented. The architecture integrates a Pi Camera, YOLO-based helmet detection, fingerprint authentication, Raspberry Pi 4 processing, AND-based decision logic, relay-controlled ignition simulation, buzzer indication, and Wi-Fi-based IoT alerting. Ignition is enabled only when the rider is detected wearing a helmet and fingerprint authentication succeeds. Failure of either condition keeps the relay OFF and generates a local warning and a remote event notification.

The prototype configuration specifies 100 test attempts across four operating conditions and lists preliminary helmet-detection, fingerprint-authentication, ignition-decision, and timing results. Since the source explicitly identifies the classification metrics as tentative and does not provide the raw records needed for independent verification, these values should not be interpreted as validated field performance. Further work should focus on reproducible datasets, independent evaluation, biometric error-rate analysis, embedded optimization, secure IoT communication, and safe automotive integration.

References

[1] World Health Organization, “Road Traffic Injuries,” WHO Fact Sheet, 20 July 2026.
[2] National Highway Traffic Safety Administration, “Motorcycle Safety,” U.S. Department of Transportation.
[3] P. J. Phillips, A. F. Martin, C. L. Wilson, and M. A. Przybocki, “An Introduction to Evaluating Biometric Systems,” NIST, 2000.
[4] P. J. Phillips, J. R. Beveridge, G. H. Givens, B. A. Draper, Y. M. Lui, and D. Bolme, “Introduction to Face Recognition and Evaluation of Algorithm Performance,” Computational Statistics, 2013.
[5] Raspberry Pi Ltd., “Camera Software Documentation,” Raspberry Pi Documentation.
[6] World Health Organization, Helmets: A Road Safety Manual for Decision-Makers and Practitioners, 2nd ed., WHO, 2023.
[7] J. Chiverton, “Helmet presence classification with motorcycle detection and tracking,” IET Intelligent Transport Systems, 2012.
[8] C. Silva et al., “Automatic motorcycle helmet identification using computer vision and machine learning,” 2013.
[9] “Automated visual helmet identification based on deep convolutional neural networks,” Computer Aided Chemical Engineering, vol. 44, pp. 2299–2304, 2018.
[10] F. W. Siebert and H. Lin, “Detecting motorcycle helmet use with deep learning,” Accident Analysis & Prevention, vol. 134, Art. 105319, 2020, doi: 10.1016/j.aap.2019.105319.
[11] “Deep Learning-Based Automatic Helmet Recognition for Two-Wheeled Road Safety,” Transportation Research Procedia, vol. 74, pp. 1171–1178, 2023, doi: 10.1016/j.trpro.2023.11.258.
[12] “AI-Based Helmet Violation Detection for Traffic Management System,” Computer Modeling in Engineering & Sciences, vol. 141, no. 1, pp. 733–749, 2024, doi: 10.32604/cmes.2024.052369.
[13] “Comprehensive study on the development of an automatic helmet violator detection system (AHVDS) using advanced machine learning techniques,” Computers & Electrical Engineering, vol. 118, Art. 109289, 2024, doi: 10.1016/j.compeleceng.2024.109289.
[14] “An enhanced YOLOv8-Based approach for accurate and efficient helmet-wearing detection in complex environments,” Digital Signal Processing, vol. 168, Art. 105681, 2026.
[15] H. Zhang, J. Li, Z. Li, L. Kong, and X. Zhao, “Real-time safety helmet detection using multi-scale feature-gated fusion,” Automation in Construction, vol. 186, Art. 106891, 2026, doi: 10.1016/j.autcon.2026.106891.
[16] K. Suthar, H. Pandya, M. Jivani, and H. Parikh, “Multilayer Authentication System to Access Vehicle Using Fingerprint and Face Identification with Raspberry Pi,” vol. 14, no. 2, 2022.
[17] Y. D. Austria et al., “Face Recognition for Motorcycle Engine Ignition with Messaging System,” 2019.

Doi:10.48047/ijiee.2026.16.2.68

[18] B. Vinothkumar et al., "Real time vehicle security system through face recognition," International Journal of Intellectual Advancements and Research in Engineering Computations, vol. 10, no. 2, pp. 177–182, 2022, doi: 10.61096/ijiarec.v10.iss2.2022.177-182.
[19] "Multimodal vehicle security system based on Internet of Things: integration of fingerprint authentication, face recognition, and GPS tracking," Jurnal Polimesin, 2026.
[20] R. Sai Eshwar and S. Sandhya Rani, "Smart Vehicle Ignition System Using Face Recognition, Image Processing, and IoT," International Journal of Information and Electronics Engineering, vol. 15, no. 7, 2025, doi: 10.48047/vsqpax30.
[21] J. A. J. Alsayaydeh et al., "Enhancing Electric Vehicle Security with Face Recognition: Implementation Using Raspberry Pi," International Journal of Advanced Computer Science and Applications, vol. 16, no. 4, 2025, doi: 10.14569/IJACSA.2025.0160472.

Doi:10.48047/ijiee.2026.16.2.68