Journal on Electronic and Automation Engineering
Vol: 4(4), December 2025
REST Publisher; ISSN: 2583-6951 (Online)
Website: https://restpublisher.com/journals/jeae/
DOI: https://doi.org/10.46632/jeae/4/4/4

An Innovation Approach for Detection of Helmet and Number Plate

C. Prakash Narayanan, S. Chandra Sekaran, D. Sugandhi Mariyal, * Kalaiyarasi R
P.S.V College of Engineering and Technology, Krishnagiri, Tamil Nadu, India.
Corresponding Author Email: kalaiyarasik925@gmail.com

Abstract: *A major cause of fatalities in motorcycle accidents is riders not wearing helmets. Conventionally, helmet compliance is monitored by traffic police at intersections or through manual analysis of CCTV footage, both of which demand considerable human effort. The proposed system overcomes these limitations by automatically analysing CCTV videos to detect motorcyclists who are not wearing helmets and extracting their license plate details. First, moving objects are classified as motorcycles or other vehicles. The system then checks whether motorcycle riders are wearing helmets. For riders without helmets, OCR is used to read the vehicle's license plate number, which also supports theft detection. Using the YOLOv5 algorithm, the system accurately detects helmet usage and license plates in real time. It can also cross-check vehicles against a database to identify stolen ones. When a violation is detected, a digital fine and notification are sent to the vehicle owner, while stolen vehicles trigger instant alerts to both the owner and law enforcement, making the system effective for traffic enforcement and vehicle security.*

1. INTRODUCTION

The growing emphasis on road safety, coupled with rapid progress in intelligent transportation, highlights a critical need for advanced technology to improve adherence to traffic laws. This study responds by investigating deep learning methods for real-time detection of helmet compliance and vehicle license plate recognition. To combat the persistent issues of traffic accidents and violations, applying machine learning to safety enforcement presents a viable solution. The core aim is to create a robust system that precisely identifies whether a driver is wearing a helmet and reads license plate information. This would aid law enforcement and safety campaigns, offering a tool with considerable potential for widespread use. Such implementation would enable more effective traffic monitoring and control, ultimately enhancing safety for all road users.

2. LITERATURE REVIEW

This system employs deep learning, specifically Convolutional Neural Networks (CNNs), to create a robust framework for traffic safety enforcement. Its primary function is twofold: first, to serve as an AI-powered safety tool by detecting whether a motorcycle rider is wearing a helmet, and second, to accurately recognize vehicle license plates. The process involves capturing real-time video or images via cameras, which are then analyzed using advanced image processing. For helmet detection, the YOLO (You Only Look Once) model is utilized to identify compliance instantly. Simultaneously, Optical Character Recognition (OCR) techniques are applied to extract alphanumeric characters from license plates. The resulting data—helmet status and vehicle identification details—is logged in a database and made accessible through a user interface for monitoring and regulatory actions, such as issuing violations.

Given that helmet non-compliance significantly increases the risk of serious injury in motorcycle accidents, consistent automated oversight is crucial to prevent repeat offenses and enhance road safety. The proposed solution integrates YOLO-based object detection with OCR to fully automate the identification of riders without helmets and the reading of their license plates. This automation enhances accuracy, boosts operational efficiency, and minimizes manual effort. Tailored to work under specific operational conditions, the system processes video feed in real time, delivering high-speed performance and offering a unified approach to helmet verification and license plate recognition.

3. METHODOLOGY

Existing System: The existing approach to traffic violation monitoring relies largely on manual inspection of CCTV footage. Traffic police personnel must observe the recorded videos and, when a rider is found without a helmet, manually zoom in to capture the vehicle's license plate number. With the growing number of two-wheeler users and the high frequency of traffic violations, this process demands substantial time and manpower. An automated system that can detect violations such as riding without a helmet and simultaneously extract the vehicle's license plate number would significantly improve efficiency. Although recent studies have addressed this problem using techniques such as CNN, R-CNN, LBP, HOG, and Hear features, many of these approaches still face challenges related to detection speed, accuracy, and overall performance.

Advantage: This system, powered by deep learning, delivers multiple key advantages by automatically identifying motorcycle riders without helmets. It enhances roadway safety, fosters greater adherence to traffic laws, and helps prevent severe head injuries. The system supports law enforcement by identifying and recording violations in real time, lowering the dependence on manual monitoring and reducing human error. Furthermore, license plate recognition aids in tracking stolen or suspicious vehicles, supporting crime prevention and investigation efforts. The system can also gather useful traffic data, including vehicle flow, helmet usage rates, and movement patterns, which can assist in urban traffic planning and management. Overall, this approach offers a cost-effective, efficient, and accurate solution for enhancing public safety and strengthening traffic rule enforcement.

Proposed System: The proposed system is designed to transform current infrastructure into a comprehensive and unified platform for traffic safety and surveillance. It integrates advanced functionalities including multi-object detection to recognize pedestrians, vehicles, and road conditions; traffic flow analysis for real-time congestion assessment; automated identification of violations; and smooth interoperability with established traffic management systems. Additional enhancements comprise license plate recognition, cloud-based data processing, and a dedicated mobile application for authorized personnel. Improved dependability is ensured through data analytics, robust privacy safeguards, and adaptability to diverse environmental settings. An intuitive control interface, coupled with a systematic maintenance strategy, supports consistent and reliable performance. This results in a flexible and efficient solution for enhancing road safety and traffic oversight. Within this framework, helmet and license plate detection are developed using Python, leveraging computer vision and deep learning methods. Pre-trained detection models like YOLO or SSD form the technical foundation, with YOLOv8 specifically applied for helmet identification and Optical Character Recognition (OCR) for extracting license plate data. These models are fine-tuned to achieve real-time processing, high precision, and reliable performance across different lighting and environmental scenarios.

Limitations: Although helmet and license plate detection systems are highly effective, they do have some limitations. Their performance can be compromised in low-light conditions, adverse weather, or when vehicles are moving at high speeds. Accuracy may also decrease if helmets or license plates are partially obscured, dirty, or damaged. Continuous video surveillance can raise privacy concerns. The cost of implementing and maintaining high-resolution camera infrastructure is another challenge. Furthermore, false positives or negatives may occur, resulting in incorrect fines or missed violations. To maintain optimal performance and adapt to changing operational requirements, regular system updates and maintenance are necessary.

4. TESTING

Unit Testing: Unit testing involves checking the correctness of individual code components, usually at the function level. In object-oriented programming, this typically applies to classes, including their constructors and destructors. These tests are generally written by developers during implementation using a white-box approach to ensure each function behaves as intended. Multiple test cases may be created to cover different logic paths or complex conditions. While unit testing cannot validate the entire system, it ensures that core components operate independently and correctly. Overall, unit testing helps reduce development risks, time, and cost by identifying defects early.

Integration Testing: Integration testing, based on the system design, is intended to validate the interactions and interfaces between combined software components. Components can be integrated either incrementally or all at once, with incremental integration being preferred because it enables early identification and resolution of issues. The primary goal of this testing is to uncover defects that arise from the interaction between modules. Testing continues progressively as additional modules are integrated, ensuring that the system functions correctly as a whole once all components are combined.

Functional Testing: Functional testing evaluates whether specific features or functions of the software perform as required. These tests are derived from requirement specifications, use cases, or user stories. The goal is to confirm that users can perform intended actions and that system features work correctly. Functional testing provides formal validation that the software meets business and technical requirements as documented in specifications and user manuals.

Data Flow Testing: Data flow testing evaluates a program by tracing how data moves through it, specifically tracking variables from their points of assignment to their points of use. This technique focuses on the order of control flow and the operations that modify data values. By examining the assignment and usage of variables, it helps identify errors linked to incorrect data handling and uncover logical defects in the program's design.

Control Flow Testing: Control flow testing is a white-box technique that verifies the order in which program statements are executed through different control structures. Test cases are created based on the program's control logic, with specific code sections selected to define execution paths. It is frequently applied during unit testing. A control flow graph—made up of nodes, edges, junctions, and endpoints—visually represents all potential execution routes.

Branch Coverage Testing: Branch coverage is a white-box testing method that ensures each branch in a control flow graph is taken at least once. It validates both true and false outcomes of every conditional statement. Though similar to decision coverage, branch coverage specifically guarantees that every possible branch in the code is executed, whereas decision coverage focuses on evaluating all possible results of decision points.

Statement Coverage Testing: Statement coverage is a common white-box testing approach that ensures every executable statement in the source code is executed at least once. It measures the percentage of executed statements out of the total statements in the program. This technique aids in designing test cases based on the internal code structure and helps ensure that no part of the code remains untested.

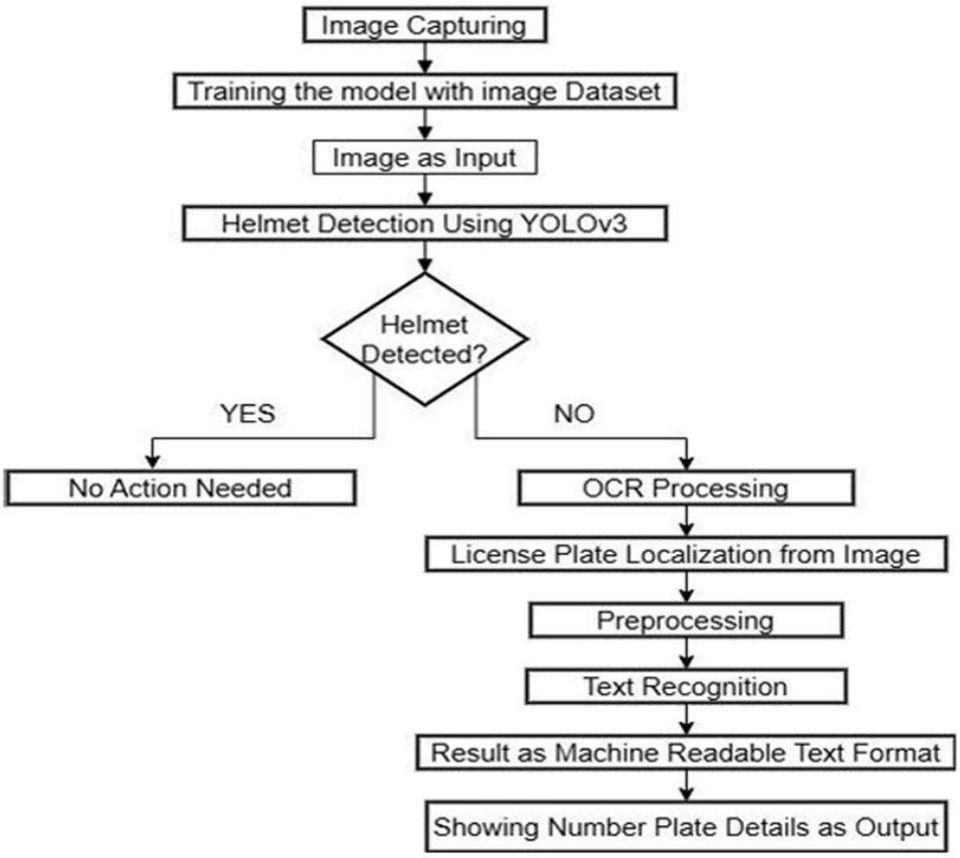

FIGURE 5. Data Flow Diagram

5. IMPLEMENTATION

Number Plate Recognition using OCR: Once a vehicle is detected, the system isolates the license plate area and employs OCR-based character recognition to extract the registration number. The process begins with detecting the plate using a YOLO-based Region Proposal Network (RPN), followed by character extraction using Tesseract OCR. The recognized text is then refined and formatted to match the standard Indian license plate format (e.g., MH-12-AB-1234).

Helmet Detection using YOLOv8: Helmet detection is carried out using the YOLOv8 object detection model, selected for its real-time performance and high accuracy. The model is trained to identify riders wearing helmets, riders without helmets, and multiple motorcyclists in the same frame. Bounding box coordinates are extracted for each detected object, and classification is performed using pre-trained weights and fine-tuned parameters.

Image Acquisition Module: The image acquisition module forms the foundation of the system, capturing real-time visual data from sources such as CCTV cameras, IP cameras, or road surveillance setups. Frames or images are continuously collected, converted into a suitable format, timestamped, and forwarded to the preprocessing stage. This module also manages frame rate, resolution, and initial noise filtering to ensure consistent, high-quality input for accurate detection.

Image Preprocessing Module: The pre-processing module standardizes captured video frames to ensure they are optimally prepared for analysis by deep learning models. Raw input images frequently differ in lighting, contain noise, vary in resolution, and may be inconsistently oriented. To address these variations, techniques such as resizing, normalization, noise reduction, contrast adjustment, and color correction are applied. This stage transforms the input into a uniform format, enhancing both detection accuracy and overall model reliability.

Helmet Detection Module: The helmet detection module analyzes image frames to classify whether a motorcycle rider is wearing protective headgear. It employs a YOLO (You Only Look Once) convolutional neural network model to identify relevant objects within the scene—specifically riders, motorcycles, and helmets. For each detected object, the model outputs bounding boxes along with confidence scores. To determine compliance, the system applies spatial logic: it checks whether the region of the rider's head overlaps significantly with a detected helmet bounding box. If sufficient overlap is confirmed, the rider is classified as "Helmet Detected". If no helmet overlaps with the rider's head region, the status is recorded as "Helmet Not Detected". This method allows for automated, real-time monitoring of helmet usage.

License Plate Detection Module: The license plate detection module accurately identifies and extracts the vehicle's license plate from images or video frames. Using deep learning object detection models like YOLO or SSD, it can detect plates under varying sizes, orientations, and lighting conditions. The preprocessed frame is analyzed to generate bounding boxes with confidence scores around potential plate regions. The identified plate is cropped and passed to the OCR module for character recognition. Optimized for real-time performance and high accuracy, this module can effectively detect even small or partially obscured license plates.

6. RESULT

This research follows a structured methodology to develop a robust system for helmet and license plate detection. The process begins with assembling a varied dataset of real-world traffic images. This data is then standardized through pre-processing techniques—including resizing, normalization, and augmentation—to improve model generalization. A deep learning object detection architecture is carefully chosen, prioritizing an optimal trade-off between accuracy and processing speed. The core of the approach utilizes transfer learning, where a pre-trained model is finely adjusted using the custom dataset to specialize in identifying helmets and license plates. The training phase involves strategic dataset splitting, hyper parameter tuning, and cross-validation to mitigate overfitting. Performance is rigorously assessed using standard metrics like precision, recall, and F1-score. Post-processing methods are applied to refine the output. The optimized model is subsequently deployed within a real-time processing framework, engineered for high-speed operation and extensively tested under diverse environmental conditions. The entire network—comprising the backbone, regional proposal network (RPN), and detection head—is refined during fine-tuning. The detection head finalizes the process by classifying proposed regions into helmet or license plate categories and precisely adjusting the corresponding bounding box coordinates. This iterative development cycle, supported by thorough documentation, ensures a reproducible and scalable solution, contributing to the advancement of intelligent transportation and improved road safety enforcement.

7. CONCLUSION

This project presents a comprehensive framework for automatically detecting motorcyclists not wearing helmets in CCTV footage and extracting their vehicle license plate numbers using CNNs and transfer learning. Beyond detection, the system stores recognized license plate information, enabling transport authorities to access vehicle records and issue fines to violators. The system can also alert nearby police stations upon identifying stolen vehicles, improving the efficiency of vehicle recovery. Some current limitations include reduced accuracy in mouse right-click operations and less smooth drag-and-drop functionality; efforts are ongoing to address these issues. Future enhancements include the integration of voice assistant features to improve human–computer interaction, making the system more intuitive and effective.

Future Work: The proposed system can be integrated with existing traffic surveillance cameras and, with minimal adjustments, deployed for real-time helmet detection. By incorporating an automated license plate recognition module, it can be extended to automatically identify riders not wearing helmets and issue penalty notices to them.

REFERENCES

[1]. L Allamki, M Panchakshari, A Sateesh, and K S Pratheek, "Helmet Detection using Machine Learning and Automatic License Plate Recognition", Int. Research J of Technology and Research (IRJET), Vol: 6, Issue: 12, Dec 2019.
[2]. C. Vishnu, D Singh, C. K Mohan, and S Babu, "Detection of Motorcyclists without Helmet in Videos using Convolutional Neural Network", Institute of Electrical and Electronics Engineers (IEEE), July 2017.
[3]. Madhuchhanda Dasgupta, Oishila Bandyopadhyay, Sanjay Chatterji, "Automated Helmet Detection for Multiple Motorcycle Riders using CNN", Institute of Electrical and Electronics Engineers(IEEE) Dec 2019.
[4]. Balasubramanian, P., Jaganathan, R., & Kumar, V. A. (2019). Real-time helmet detection and warning system for two-wheelers. International Journal of Engineering and Advanced Technology, 8(5), 110-114.
[5]. Bhardwaj, P., & Tiwari, S. (2020). Automatic number plate recognition using deep learning techniques: a survey. International Journal of Machine Learning and Networking, 3(1), 1-11.
[6]. Chakraborty, S., Ghosh, A., & Das, A. (2020). Efficient detection of helmet and two wheeler riding using deep learning. International Journal of Computer Applications, 179(32), 12-16.
[7]. Chen, X., Li, Y., & Chen, L. (2018). A real-time vehicle detection and recognition system for intelligent transportation systems. Sensors, 18(11), 3796.
[8]. Fan, H., & Yang, C. (2020). A review of vehicle detection and tracking systems. Journal of Advanced Transportation, 2020, 1-14. 65.