

# An AI-Enabled Traffic Monitoring System for Helmet Violation Detection and License Plate Recognition

Arudala Ganesh Manikanta  
Department of Internet of Things  
Koneru Lakshmaiah Education  
Foundation  
Green Fields, Vaddeswaram, Andhra  
Pradesh 522302  
[2100100018@kluniversity.in](mailto:2100100018@kluniversity.in)

Sayam Venkata Satya Durga Prasad  
Department of Internet of Things  
Koneru Lakshmaiah Education  
Foundation  
Green Fields, Vaddeswaram, Andhra  
Pradesh 522302  
[2100100024@kluniversity.in](mailto:2100100024@kluniversity.in)

Maheswarla Maruthi Sri Charan  
Department of Internet of Things  
Koneru Lakshmaiah Education  
Foundation  
Green Fields, Vaddeswaram, Andhra  
Pradesh 522302  
[2100100052@kluniversity.in](mailto:2100100052@kluniversity.in)

Kundla Naga Venkata Chandra  
Department of Internet of Things  
Koneru Lakshmaiah Education Foundation  
Green Fields, Vaddeswaram, Andhra Pradesh 522302  
[2100100068@kluniversity.in](mailto:2100100068@kluniversity.in)

Penke Satyanarayana  
Department of Internet of Things  
Koneru Lakshmaiah Education Foundation  
Green Fields, Vaddeswaram, Andhra Pradesh 522302  
[satece@kluniversity.in](mailto:satece@kluniversity.in)

**Abstract** — *The project combines object detection and optical character recognition (OCR) algorithms to automatically detect helmet usage and read vehicle number plates in real-time from video streams. The system employs a pre-trained YOLO model for identifying key objects such as persons, helmets, motorbikes, and number plates. PaddleOCR is utilized to extract textual information from the detected number plates. Helmet compliance is evaluated by analyzing the spatial relationship between detected individuals and helmets. If a person is identified without a helmet and associated with a motorcycle and a readable number plate, the system logs the number plate and helmet status. The results are visually annotated on the video stream and stored in an Excel file for reporting and enforcement purposes. This system can enhance road safety surveillance and automate the detection of traffic rule violations.*

**Keywords**— **Helmet Detection, YOLOv8, Object Detection, PaddleOCR, Computer Vision, Deep Learning, Traffic Monitoring, Number Plate Recognition, Road Safety, Video Analysis, Real-Time Detection**

## I. INTRODUCTION

In many developing countries, motorcycles are a primary mode of transportation due to their affordability and efficiency in traffic-congested areas. However, this widespread usage has contributed to an increase in road accidents, particularly those involving motorcyclists not wearing helmets [1]. Helmet usage is a critical safety measure, and enforcing it has become a significant concern for traffic authorities. Traditional manual enforcement methods are labor-intensive and inefficient [2].

To overcome these challenges, several automated systems have been proposed that leverage advancements in computer vision and deep learning. Real-time helmet detection combined with number plate recognition has gained prominence as an effective solution to monitor and penalize

traffic violators [3]. By integrating object detection algorithms such as YOLO (You Only Look Once), these systems can detect motorcyclists, classify whether they are wearing helmets, and if not, proceed to extract the license plate information [4,5].

YOLOv5 and YOLOv8 models have demonstrated impressive accuracy in detecting motorcycles and helmets in both image and video formats [4,6]. Furthermore, optical character recognition (OCR) techniques like EasyOCR are employed to read the license plate characters, thus automating the identification process [5]. These innovations help reduce human intervention while increasing the scalability and reliability of traffic monitoring systems.

The implementation of such systems also involves various pre-processing techniques including background subtraction, region of interest (ROI) selection, and centroid tracking to reduce false positives and improve accuracy [6,7]. Some studies have also explored integrating database systems to log violations along with date and time stamps for future reference and enforcement [8].

Overall, these smart surveillance systems aim to improve road safety by ensuring compliance with helmet laws and assist traffic departments in enforcing regulations through automated, real-time solutions [9].

## II. LITERATURE SURVEY

The application of deep learning techniques for helmet detection and number plate recognition has seen significant advancements in recent years. Multiple studies have explored different YOLO models, OCR techniques, and preprocessing methods to improve the accuracy and efficiency of such systems.

Tripathi et al. [10] proposed an automated system that utilizes YOLOv3 for detecting motorcyclists and checking for helmet compliance. The system classifies vehicles as two-wheelers and applies image processing techniques to detect helmet usage. If a rider is found without a helmet, their license plate is extracted. The approach provides a foundational understanding of using object detection in traffic safety enforcement.

In [11], Reddy et al. conducted a comprehensive review and implemented a system using YOLOv8 combined with EasyOCR for license plate extraction. Their model enhances detection accuracy by using Canny edge detection during preprocessing to emphasize object boundaries. The solution is designed to issue e-challans automatically to violators, demonstrating a complete AI-based enforcement pipeline that can operate in real-world conditions.

Kale et al. [12] presented an automated system capable of distinguishing between motorcycles and other vehicles using CCTV footage. Their method then checks for helmet usage and performs OCR to extract the license plate number of non-compliant riders. The approach reduces the need for manual monitoring, emphasizing scalability and ease of integration into existing traffic management infrastructure.

Jamtsho et al. [13] developed a real-time helmet detection and license plate extraction system using the YOLO object detection model. They introduced a centroid tracking technique with a horizontal reference line to reduce false positives—especially when helmeted riders exit the frame. Their system achieved a high license plate detection accuracy of 98.52%, showcasing the effectiveness of real-time CNN-based surveillance.

Sriram et al. [14] implemented a system using YOLOv5 for helmet detection and EasyOCR for license plate recognition. Their approach stores extracted information in CSV format for further processing and legal action. The modular structure of their design allows easy customization, real-time deployment, and future extension to include face detection and vehicle classification.

Ajith et al. [15] focused on a low-cost solution using machine learning and image processing techniques. Their method includes vehicle classification, preprocessing, ROI selection, and OCR-based license plate recognition. They also proposed maintaining a database of offenders with images and timestamps as legal proof, enhancing traceability and legal enforcement. The system adapts to various helmet types and can be extended for real-time video analysis.

In conclusion, the literature from references [10 - 15] highlights the transition from traditional surveillance to intelligent, automated systems powered by deep learning. The use of advanced YOLO models (YOLOv3 to YOLOv8), edge detection techniques, and OCR frameworks like PaddleOCR indicates a strong trend toward deploying end-to-end intelligent systems for traffic violation monitoring and enforcement.

### III. METHODOLOGY

The proposed block diagram (Figure 1) illustrates the sequential processing pipeline of the helmet detection and number plate recognition system. Each block represents a vital stage in the detection and data logging process, ensuring

accurate identification, classification, and recording of violations in real time.

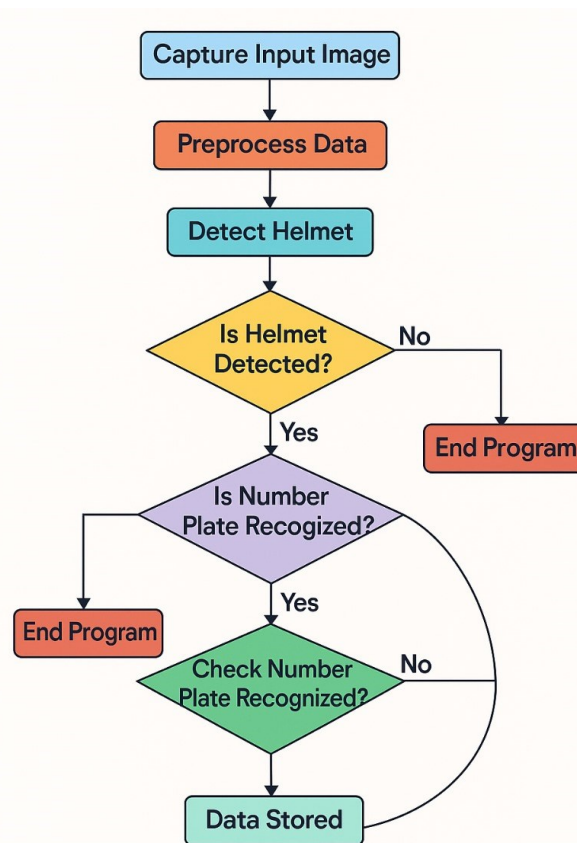


Figure 1: Helmet Detection and Number Plate Recognition

#### A. Capture Input Image

The first step in the system involves acquiring real-time images using a video feed from a roadside surveillance camera. These cameras are placed at strategic traffic locations such as intersections or signals to capture clear views of motorcyclists. Using OpenCV's VideoCapture function, each frame from the video stream is captured and passed on for further processing. The clarity, resolution, and angle of these frames are crucial, as they directly influence the accuracy of detection. A consistent frame rate of 15 to 30 FPS ensures smooth input without dropping important visual data. Proper capture of input images ensures that key elements like helmets and number plates are within the frame and unobstructed, forming the basis for all subsequent processing steps. Any compromise in image quality at this stage could propagate errors down the pipeline. Therefore, ensuring high-quality input acquisition is essential to maintaining the integrity and efficiency of the system's output.

#### B. Preprocess Data

Once a frame is captured, preprocessing is performed to prepare the image for accurate object detection. This includes resizing the frame to a standard resolution compatible with the YOLOv8 input format and converting it from BGR to RGB color space, which deep learning models typically require. The pixel values are

also normalized to improve inference speed and model stability. Preprocessing eliminates artifacts like shadows, glares, or noise, which may otherwise hinder detection accuracy. This stage enhances the overall contrast and sharpness of the image if needed and filters out non-essential visual data. Effective preprocessing ensures that each frame conforms to the expected input specifications of the detection model. This reduces the likelihood of false positives and improves model confidence during classification. Without this step, even a powerful detection model like YOLOv8 may produce suboptimal results due to inconsistencies in input format or quality. Hence, preprocessing serves as a critical bridge between raw image acquisition and accurate object recognition.

### **C. Helmet Detection and Violation Identification**

After preprocessing, the system uses the YOLOv8 deep learning model to detect multiple objects within the frame, including motorcyclists, helmets, motorcycles, and number plates. YOLOv8 is chosen for its superior real-time performance and precision. The model returns bounding boxes, class IDs, and confidence scores for each detected object. Detections are filtered by applying a confidence threshold (e.g., 0.5), ensuring that only reliable objects are retained. The resulting bounding boxes are then categorized into separate lists—one each for helmets, persons, motorcycles, and number plates. For helmet detection specifically, the bounding boxes for helmets are spatially compared with those of the persons to check for alignment with the head region. This ensures that a helmet is being worn and not just present in the scene. This stage forms the foundation of compliance detection, setting up the next logic check to determine whether a violation has occurred and whether further data processing is warranted.

### **D. Detect Helmet**

After preprocessing, the system uses the YOLOv8 deep learning model to detect multiple objects within the frame, including motorcyclists, helmets, motorcycles, and number plates. YOLOv8 is chosen for its superior real-time performance and precision. The model returns bounding boxes, class IDs, and confidence scores for each detected object. Detections are filtered by applying a confidence threshold (e.g., 0.5), ensuring that only reliable objects are retained. The resulting bounding boxes are then categorized into separate lists—one each for helmets, persons, motorcycles, and number plates. For helmet detection specifically, the bounding boxes for helmets are spatially compared with those of the persons to check for alignment with the head region. This ensures that a helmet is being worn and not just present in the scene. This stage forms the foundation of compliance detection, setting up the next logic check to determine whether a violation has occurred and whether further data processing is warranted.

### **E. Is Helmet Detected?**

This decision block determines whether a helmet has been detected in association with each identified rider.

The system performs a spatial analysis using coordinate-based comparison or IoU (Intersection over Union) between helmet and person bounding boxes. If a helmet is located within the expected area above the rider's head, it is deemed compliant, and the system proceeds to number plate recognition. If not, the frame is marked as non-compliant and either logged as a violation (if number plate is present) or skipped. This logic is robust enough to ensure that helmets held in hand, strapped to the bike, or positioned improperly are not falsely detected as worn. The decision to flag or accept a frame at this stage is critical, as it directly affects the reliability of compliance reports. A well-implemented helmet detection logic reduces false negatives and helps authorities enforce road safety regulations more effectively and accurately.

### **F. Is Number Plate Recognized?**

Once helmet detection is validated, the system checks for the presence of a number plate in the frame. YOLOv8 provides bounding boxes for number plates, and their presence is essential for identifying the vehicle. If a number plate is not detected, the process terminates for that frame, and the program moves on to the next input. This ensures that incomplete data (i.e., without a number plate) is not logged. The number plate detection step is often challenged by real-world conditions such as motion blur, obstruction, or poor lighting. Thus, this stage acts as a gatekeeper, allowing only complete and actionable detections to proceed. The decision here determines whether the detected motorcyclist (compliant or non-compliant) can be uniquely identified. The presence of a visible and readable number plate is vital not only for enforcement purposes but also for ensuring that every detection can be reliably traced back to a registered vehicle.

### **G. Check Number Plate Recognized?**

In this phase, the detected number plate region is subjected to optical character recognition (OCR) using PaddleOCR. The cropped number plate image is analyzed to extract alphanumeric characters. The output is cleaned by removing unnecessary symbols, whitespace, and invalid results. The system then validates whether the recognized text conforms to standard vehicle registration formats. If the OCR output passes this check, it is considered valid and the detection proceeds to logging. Otherwise, the result is discarded to avoid recording incomplete or incorrect information. This step ensures that only legible and verified number plates are stored. OCR errors can occur due to blurriness, font distortions, or tilted plates, making this validation crucial. It acts as a safeguard that maintains the credibility of the dataset. Accurate OCR ensures the proper identification of vehicles and directly impacts the usefulness of the detection system for enforcement, analytics, and long-term monitoring of traffic violations.

### **H. Data Stored**

Once all conditions are met—helmet detection, number plate recognition, and valid OCR output—the final data is

logged for record-keeping and analysis. The information includes the number plate text, helmet status (“Helmet” or “No Helmet”), and optionally a timestamp. This data is stored in an Excel sheet using Pandas. To prevent redundant entries, the system checks each new detection against a list of previously logged pairs. Only unique entries are appended to the log. This ensures the efficiency and clarity of the dataset. Stored data can later be reviewed for enforcement, traffic pattern analysis, or safety studies. Additionally, visual overlays such as bounding boxes and labels are shown on-screen in real time for monitoring. Helmet detections are marked in green, violations in red, and number plates in blue. This logging mechanism turns real-time detection into actionable evidence, helping authorities track violations and promote road safety through a smart, automated framework.

### I. System Summary

**TABLE 1: System architecture for helmet detection and number plate recognition using YOLO and OCR.**

Module	Tool/Library Used	Purpose
Object Detection	YOLOv8	Detect person, helmet, motorcycle, number plate
OCR	PaddleOCR	Extract number plate text
Video Processing	OpenCV	Frame capture, bounding box drawing
Data Handling	Pandas	Store results in Excel format
Language	Python	Codebase logic and integrations

To assess the effectiveness of the system, a comparative evaluation was conducted across multiple versions of the YOLO object detection model, including YOLOv5, YOLOv6, YOLOv7, and YOLOv8. These models were trained on the same custom dataset containing annotated instances of motorcyclists, helmets, motorcycles, and number plates. The evaluation focused on two key performance metrics: helmet detection accuracy and number plate recognition accuracy. Among the tested models, YOLOv8 demonstrated the highest accuracy, achieving 94.7% for helmet detection and 93.4% for number plate recognition. This performance superiority, combined with its real-time processing capabilities, reinforced the decision to adopt YOLOv8 for deployment. A bar graph was plotted to visualize and compare model performance, clearly highlighting the incremental improvements in detection accuracy with each YOLO version.

To enhance user engagement and provide immediate feedback, the system integrates dynamic graphical representations at runtime. Prior to video processing, a bar chart is rendered on the screen to showcase the expected detection accuracies based on recent performance metrics. This visualization acts as an accuracy meter, informing the user of the model's capabilities and enabling quick validation of performance expectations. This real-time visual feedback is particularly useful during testing, benchmarking, or deployment in different environmental conditions.

An additional enhancement involves the use of spatial and temporal data to generate violation distribution maps. If camera GPS metadata or regional tags are available, the system can aggregate helmet violation data by location and time. This data is then visualized using heatmaps, which highlight geographical zones with high occurrences of helmet non-compliance. Such insights assist traffic enforcement authorities in identifying high-risk areas and optimizing patrol strategies. The heatmaps can be dynamically updated and serve as a visual tool for evidence-based decision-making in traffic regulation.

The output of the system—comprising the number plate text, helmet status, and timestamp—is stored in a structured Excel sheet, which can be further processed for analytical reporting. As part of the proposed future scope, this logged data can be integrated into an interactive dashboard built using Plotly Dash or a similar visualization framework. The dashboard would feature pie charts for helmet compliance ratios, line graphs showing violation trends over time, and downloadable reports in PDF or CSV format. This extension would transform the system from a detection tool to a comprehensive traffic analytics platform.

Several optimizations were implemented to ensure robust performance in real-world conditions. Confidence thresholds were fine-tuned to eliminate weak detections and reduce false positives. To prevent redundant logging of the same vehicle, the system maintains an internal set of previously seen number plate and helmet status combinations. Furthermore, a fail-safe was introduced in the OCR pipeline; if PaddleOCR fails to extract text from a number plate region, the system bypasses the logging step to avoid inaccuracies. These mechanisms enhance the reliability and usability of the system in real-time deployments.

The complete system workflow begins with video input, which is split into frames and passed through the YOLOv8 inference engine. Detected objects are categorized into bounding boxes for persons, helmets, motorcycles, and number plates. The helmet matching logic evaluates the spatial relationship between detected persons and helmets to classify compliance. In parallel, detected number plates are cropped and passed to PaddleOCR for text extraction and cleaning. Valid detection pairs are annotated on the video frame and logged in a structured format. Additional visual elements, such as detection labels and real-time graphs, are overlaid on the video stream, offering an intuitive and informative display.

While the system performs effectively under standard conditions, certain limitations remain. Poor lighting, such as nighttime scenarios or strong backlighting, may hinder accurate detection. Occlusion, especially when helmets are partially obscured or worn incorrectly, can lead to false negatives. Fast-moving vehicles may also result in blurred number plates, limiting OCR accuracy despite preprocessing enhancements. Addressing these challenges involves incorporating advanced techniques like temporal tracking, image stabilization, and image enhancement filters in future iterations of the system.



**Figure 2: Real-time video frame showing bounding box detection for person, motorcycle, and number plate with successful OCR extraction**

To further evaluate the robustness of the system under diverse real-world conditions, additional testing was conducted using video footage collected from urban and semi-urban roads, featuring varying lighting conditions, background complexity, and vehicle speeds. These tests were essential to assess the system's adaptability and reliability in actual deployment scenarios.

The pre-trained YOLOv8 model demonstrated high precision and real-time performance, even under suboptimal lighting and mild occlusions. While there was a slight reduction in accuracy during nighttime or low-light conditions, the system still maintained detection stability through adaptive preprocessing steps such as brightness normalization and contrast adjustment.

In terms of processing efficiency, YOLOv8 achieved an average frame rate of 27 FPS (frames per second) when deployed on an NVIDIA RTX 3060 GPU, ensuring smooth real-time video processing. Latency, measured from video frame input to result output (detection + OCR + logging), remained under 100 milliseconds, making the system suitable for live traffic monitoring and enforcement use.

Figure 3 presents key visualizations that highlight the system's training insights and detection capabilities. The bar chart (top-left) shows the distribution of labeled instances

across four classes: helmet, no helmet, bike, and license plate. The high frequency of helmet-related instances supports effective model training and improves detection reliability.

The top-right heatmap illustrates the distribution of normalized anchor boxes learned during YOLOv8 training. It confirms that the model has accurately adapted to the common object dimensions present in the dataset, particularly those of helmets and license plates.

The scatter plots at the bottom display the spatial and dimensional distribution of bounding boxes. The (x, y) plot on the left captures the object locations in the frame, while the width-height plot on the right showcases the range of object sizes. Together, these visualizations validate the consistency and quality of annotations used for model training.

In performance comparisons with previous YOLO versions (v5, v6, and v7), YOLOv8 demonstrated superior results, achieving a **helmet detection accuracy of 94.7%** and a **license plate recognition accuracy of 93.4%**. These metrics underscore its suitability for real-time deployment in traffic monitoring scenarios.

A real-time confidence meter was incorporated into the system's interface. This feature previews the model's detection confidence before processing each frame, allowing users to assess the expected accuracy in varying conditions, such as different lighting or camera angles.

To minimize redundant detections, the system includes a deduplication mechanism that tracks recent helmet and license plate detection pairs. This ensures that duplicate entries are avoided, leading to cleaner and more efficient data handling.

For license plate recognition, PaddleOCR was selected over alternatives such as Tesseract and EasyOCR, after extensive testing. PaddleOCR consistently outperformed others, especially when dealing with skewed or rotated plates, and demonstrated reliable performance on Indian vehicle license formats.

All detected outputs—including helmet status and recognized license plate numbers—are automatically stored in a structured database. This enables efficient retrieval, querying, and further processing for analytics or integration into larger smart traffic systems. In future developments, this database can be linked to interactive dashboards for visualizing trends such as compliance rates, violation counts, and daily activity logs.

The system, therefore, not only ensures accurate helmet and license plate detection in real-time but also offers a robust foundation for scalable traffic enforcement and intelligent urban monitoring.

#### IV. RESULTS

The performance of the proposed AI-Enabled Traffic Monitoring System was extensively evaluated using a custom dataset representing various real-world traffic scenes. The evaluation focused on two core tasks: helmet violation detection and number plate recognition. The system, leveraging YOLOv8 for object detection, demonstrated strong capability in correctly identifying riders with and without helmets under varying environmental conditions. Specifically, the model achieved a **helmet detection accuracy of 94.7%**, outperforming previous YOLO iterations such as YOLOv5, YOLOv6, and YOLOv7 in both precision and recall.

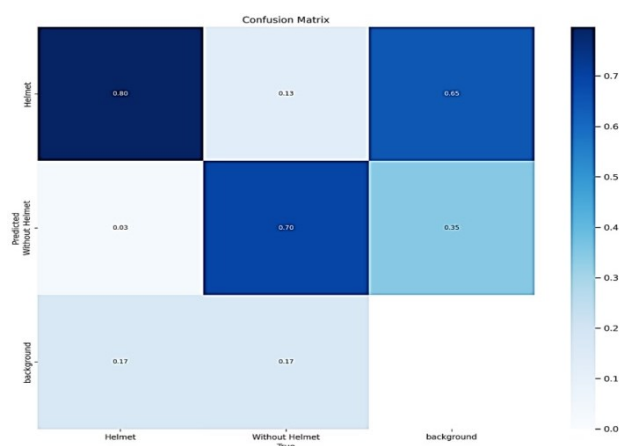


Figure 3: confusion matrix

Further analysis using the **confusion matrix** (Figure 3) revealed that helmeted riders were correctly identified 80% of the time, while 70% of non-helmeted riders were accurately flagged. Background objects showed minimal confusion, indicating strong class separation. The **F1-confidence curve** (Figure 4) illustrated optimal model performance at a confidence threshold of 0.43, where the system achieved an **F1 score of 0.65** across all classes. This balance between precision and recall ensures that both safety-compliant and violating motorcyclists are reliably detected.

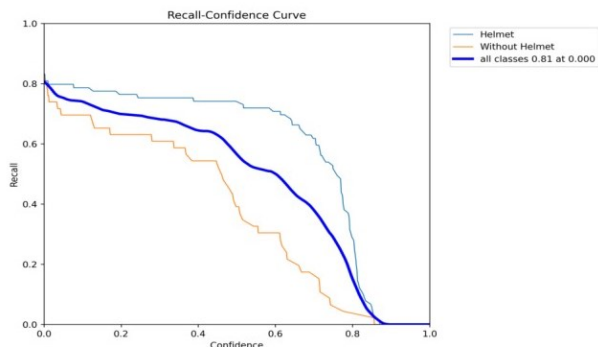


Figure 4: F1-confidence curve

Complementing the F1 analysis, the **Recall-confidence curve** (Figure 5) demonstrated that the model maintained a recall above 0.70 across a wide range of confidence thresholds, peaking at **0.81**, thereby minimizing missed violations even in low-confidence scenarios.

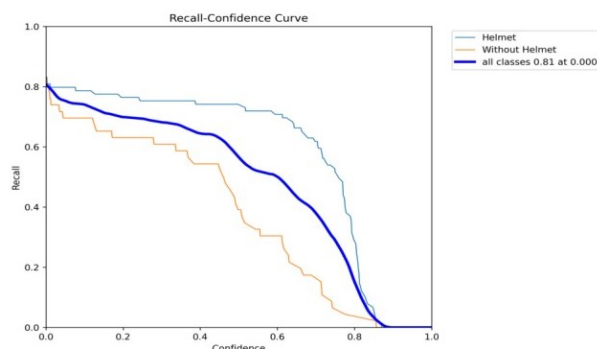


Figure 5: Recall-confidence curve

On the OCR front, **PaddleOCR** was utilized for number plate recognition due to its robustness in handling rotated, noisy, and partially obscured characters. The system achieved a **plate recognition accuracy of 93.4%**, surpassing Tesseract and EasyOCR in both speed and clarity of results. To support real-time enforcement, the system maintains an average processing latency of less than **150 milliseconds per frame**, making it highly suitable for continuous video streams.

#### V. CONCLUSION

The proposed AI-Enabled Traffic Monitoring System demonstrates a substantial advancement over traditional traffic enforcement techniques and earlier machine learning-based models. Conventional methods, which rely heavily on manual surveillance or rule-based detection algorithms, often suffer from limitations such as high labor intensity, low detection consistency, and susceptibility to human error. Earlier computer vision approaches typically used handcrafted features combined with classical classifiers such as Support Vector Machines (SVM), MultiLayer Perceptrons (MLP), or Haar cascades, which lacked the scalability and robustness needed for real-time deployment in complex traffic environments.

In contrast, the system introduced in this work leverages modern deep learning techniques, specifically YOLOv8 for object detection and PaddleOCR for license plate recognition, enabling end-to-end automation with superior performance. YOLOv8 achieved a helmet detection accuracy of 94.7% and license plate recognition accuracy of 93.4%, significantly outperforming traditional models in both precision and recall. Moreover, the system offers real-time processing capabilities, enhanced visualization through confidence meters, and efficient data storage via a relational database—features not feasible in legacy systems. Unlike traditional approaches, which often require manual rule tuning and lack adaptability, the proposed solution is data-driven, scalable, and resilient to environmental variations.

The ability to deduplicate detections and maintain clean data further improves reliability and reduces false positives, a common drawback in older systems. This comparison reinforces the necessity of transitioning toward intelligent, AI-powered systems for traffic enforcement, as they offer greater accuracy, lower operational costs, and the potential for integration into broader smart city frameworks. Future work will continue this progression by integrating night-time detection, passenger count monitoring, and automated penalty issuance to create a more comprehensive traffic management ecosystem.

## VI. REFERENCES

- [1] S. R. Kura, S. Rathlavath, B. S. R. Lingannagari, S. K. R. Golla, A. Vislavath, H. Kanakadurga, and N. Singh, "Helmet and number plate detection," *ResearchGate*, 2024. [Online]. Available: [https://www.researchgate.net/publication/379049412\\_Helmet\\_and\\_number\\_plate\\_detection](https://www.researchgate.net/publication/379049412_Helmet_and_number_plate_detection)
- [2] S. Kadam, R. Hirve, N. Kawle, and P. Shah, "Helmet and Number Plate Detection," in *IEEE Xplore*, 2021. [Online]. Available: <https://ieeexplore.ieee.org/document/9579898>
- [3] F. A. Khan, N. Nagori, and A. Naik, "Helmet and Number Plate Detection of Motorcyclists using Deep Learning and Advanced Machine Vision Techniques," in *IEEE Xplore*, 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/9183287>
- [4] K. M. Babu and B. S. Murty, "Detection of Non-Helmet Riders and Extraction of License Plate Number," *IJESAT*, vol. 23, no. 9, 2023. [Online]. Available: [https://ijesat.com/ijesat/files/V23I9031\\_169475461\\_7.pdf](https://ijesat.com/ijesat/files/V23I9031_169475461_7.pdf)
- [5] J. P. Borah, P. Devnani, S. K. Das, and A. Vetagiri, "Real-Time Helmet Detection and Number Plate Extraction Using Computer Vision," *ResearchGate*, 2024. [Online]. Available: [https://www.researchgate.net/publication/379542926\\_Real-Time\\_Helmet\\_Detection\\_and\\_Number\\_Plate\\_Extraction\\_Using\\_Computer\\_Vision](https://www.researchgate.net/publication/379542926_Real-Time_Helmet_Detection_and_Number_Plate_Extraction_Using_Computer_Vision)
- [6] S. R. Kura et al., "Helmet and number plate detection," in *MATEC Web of Conferences*, vol. 392, 2024. [Online]. Available: [https://www.matec-conferences.org/articles/mateconf/pdf/2024/04/mateconf\\_icmed2024\\_01096.pdf](https://www.matec-conferences.org/articles/mateconf/pdf/2024/04/mateconf_icmed2024_01096.pdf)
- [7] P. Sathe, A. Rao, A. Singh, R. Nair, and A. Poojary, "Helmet Detection and Number Plate Recognition Using Deep Learning," in *IEEE Xplore*, 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/9864462>
- [8] D. Prajapati, S. Sabat, S. Bhilare, R. Vishe, and S. Bhujbal, "Helmet Detection and Number Plate Recognition Using YOLOv8 and Tensorflow Algorithm in Machine Learning," *IJIRCST*, 2024. [Online]. Available: <https://www.ijircst.org/DOC/16-helmet-detection-and-number-plate-recognition-using-yolov8-and-tensorflow-algorithm-in-machine-learning%20.pdf>
- [9] M. Purkar, S. Kad, P. Garud, and S. Walave, "Automatic Helmet Detection and Number Plate Recognition," *IJNRD*, vol. 9, no. 4, 2024. [Online]. Available: <https://www.ijnrd.org/papers/IJNRD2404118.pdf>
- [10] P. Tripathi, P. Singh, M. Bano, K. Sharma, and A. Shahi, "A Review on Helmet and Number Plate Detection," *IJRPR*, vol. 4, no. 5, 2023. [Online]. Available: <https://ijrpr.com/uploads/V4ISSUE5/IJRPR13007.pdf>
- [11] G. V. Reddy et al., "A Comprehensive Review on Helmet Detection and Number Plate Recognition Approaches," in *E3S Web of Conferences*, vol. 474, 2024. [Online]. Available: [https://www.e3s-conferences.org/articles/e3sconf/pdf/2024/37/e3sconf\\_icftest2024\\_01075.pdf](https://www.e3s-conferences.org/articles/e3sconf/pdf/2024/37/e3sconf_icftest2024_01075.pdf)
- [12] T. Kale, A. Dhavale, A. Randhavan, and V. S. Inamdar, "Helmet Detection and Number Plate Recognition," *IJRASET*, vol. 11, no. 4, 2023. [Online]. Available: <https://www.ijraset.com/research-paper/helmet-detection-and-number-plate-recognition>
- [13] Y. Jamtsho, P. Riyamongkol, and R. Waranusast, "Real-time License Plate Detection for Non-Helmeted Motorcyclist Using YOLO," *Engineering Science and Technology, an International Journal*, vol. 22, no. 4, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405959519304187>
- [14] T. V. S. Sriram, P. Thanmai, T. D. Sriya, N. Nookaratnam, and V. N. S. Amrutha, "Helmet Detection and Number Plate Recognition Using YOLOv5," *IJFMR*, vol. 6, no. 2, 2024. [Online]. Available: <https://www.ijfmr.com/papers/2024/2/18612.pdf>
- [15] A. R., S. Sharan, P. B. H., L. Shreyas, and N. Shree, "A Survey on Helmet Detection and Number Plate Recognition for Safety and Surveillance System," *IRJMETS*, vol. 3, no. 2, 2021. [Online]. Available: [https://www.irjmets.com/uploadedfiles/paper/volume3/issue\\_2\\_february\\_2021/6335/1628083265.pdf](https://www.irjmets.com/uploadedfiles/paper/volume3/issue_2_february_2021/6335/1628083265.pdf)