International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
SSN: 3068-272X www.ijdim.com Original Research Paper

AI-Based Smart Traffic Violation Detection System

[1]Mr. k. Ramesh Babu,[2]Pittu Siva Chaitanya,[3]Sajja Mohan,[4]Papisetty Poojitha

[1]Assistant Professor, Dept of Computer Science and Engineering, St. Ann's College of Engineering and Technology, Chirala-523187, India.

[2,3,4]U. G Student, Dept of Computer Science and Engineering, St. Ann's College of Engineering and Technology, Chirala-523187, India.

ABSTRACT- ***The AI Traffic Violation Detection System is a web-based application that automates traffic monitoring using AI and Computer Vision. It uses the YOLO deep learning model to detect vehicles, motorcycles, pedestrians, helmets, and no-helmet violations from images, videos, and live webcam streams. Built with Python, Flask, OpenCV, HTML, and CSS, it features secure login, an interactive dashboard, and supports image, video, and real-time detection. The system calculates vehicle counts, traffic density, helmet usage, no-helmet violations, and detection accuracy to provide valuable insights for traffic authorities. By automating monitoring, it reduces manual effort and helps improve road safety by promoting helmet compliance and efficient traffic management. This intelligent tool supports law enforcement and contributes to safer, more efficient road transportation.***

Keywords: Artificial Intelligence, YOLOv8 Object Detection, Traffic violation Detection, Helmet and No-Helmet Detection, Traffic Density Analysis, Real-Time Webcam Monitoring

INRODUCTION

The rapid increase in vehicles and urban traffic has created major challenges for road safety and management. Traditional monitoring depends on police and CCTV operators manually spotting violations, which is time-consuming and often ineffective in crowded areas. To improve this, the AI Traffic Violation Detection System uses Artificial Intelligence and Computer Vision with the YOLO deep learning model to detect vehicles, motorcycles, people, helmets, and no-helmet violations from images, videos, and live streams. It also counts vehicles and

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
SSN: 3068-272X www.ijdim.com Original Research Paper

classifies traffic density as low, medium, high, or traffic jam. Built with Python, Flask, OpenCV, YOLO, HTML, CSS, and SQLite, the system offers user registration, a dashboard, image and video detection, live monitoring, and accuracy reports. After analyzing videos, it provides detailed data on vehicle counts, traffic density, and detection accuracy. This system reduces manual monitoring efforts and enhances road safety. Future updates could include number plate recognition, automated fines, traffic signal violation detection, and real-time alerts.

LITERATURE SURVEY

Traditional traffic monitoring relies on CCTV and police to spot violations, but it is labor-intensive, time-consuming, and prone to errors, especially in busy areas. Operators struggle to monitor multiple feeds, and human fatigue reduces accuracy. These systems also lack real-time analysis, delaying response to violations and traffic congestion. Manual methods don't provide reliable vehicle counts or traffic reports for planning. To solve these issues, AI, Computer Vision, and Deep Learning—especially models like YOLO—are used for automatic detection of vehicles, people, helmets, and violations from images, videos, and live streams. These systems improve accuracy, reduce human effort, offer real-time monitoring, and generate detailed reports, making AI-based traffic monitoring an effective solution for safer, smarter roads.

RELATED WORK

Research on intelligent traffic monitoring using AI, Computer Vision, and deep learning has advanced significantly. Traditional systems depend on CCTV and manual observation, which require much manpower and can miss violations in heavy traffic. Earlier automated methods used image-processing techniques like background subtraction and motion tracking but faced issues with shadows, poor lighting, and complex scenes. Deep learning models such as YOLO have improved detection speed and accuracy for multiple objects including vehicles and people. YOLOv8 offers even better real-time performance. Detecting helmet violations is a key focus, with models trained to differentiate helmeted from non-helmeted riders. Traffic density is estimated by counting vehicles and categorizing traffic levels. The proposed AI-Based Smart Traffic Violation Detection System combines vehicle and helmet detection,

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
SSN: 3068-272X www.ijdim.com Original Research Paper

triple-riding detection, and traffic density classification into a Flask web app for image, video, and live monitoring.

EXISTING SYSTEM

The existing traffic monitoring system relies mainly on CCTV cameras, traffic signals, and manual observation by traffic police. Cameras placed at key locations capture vehicle movement, and officers watch live feeds or recorded videos to detect violations like helmetless riding, triple riding, signal jumping, and congestion. This process requires constant human attention, making it time-consuming and prone to missed or delayed detections, especially during heavy traffic. Some automated systems use basic image-processing techniques such as motion detection and background subtraction to identify moving vehicles, but they often struggle with poor lighting, weather conditions, vehicle overlap, and low-quality video. These systems typically offer only basic vehicle counting and monitoring and do not integrate helmet violation detection, triple-riding detection, traffic density classification, or support image, video, and live webcam monitoring in a single web application.

PROPOSED SYSTEM

The proposed AI-Based Smart Traffic Violation Detection System automates traffic monitoring to enhance road safety using AI, Computer Vision, and YOLO deep learning models. It detects vehicles—including cars, buses, trucks, and motorcycles—along with pedestrians, helmets, and no-helmet violations from images, videos, and live webcam feeds. The system analyzes motorcycle riders to check helmet use and identify potential triple-riding by counting riders. Traffic density is calculated by counting vehicles and classifying road conditions from No Traffic to Traffic Jam, enabling quick congestion assessment. Developed with Python, Flask, OpenCV, HTML, CSS, SQLite, and YOLOv8, it features secure user registration, a dashboard, image and video detection, live monitoring, traffic density results, and accuracy reports. By automating monitoring, the system reduces manual effort and delivers faster, reliable, real-time traffic analysis.

SYSTEM ARCHITECTURE

The AI-Based Smart Traffic Violation Detection System architecture consists of image and video uploads, live webcam input, YOLO detection models, traffic analysis modules, a Flask web application,

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
SSN: 3068-272X www.ijdim.com Original Research Paper

an SQLite database, and result display components. Users register and log in through the Flask app, with their data stored in SQLite. The system processes images, videos, and webcam frames using YOLO to detect vehicles (cars, buses, trucks, motorcycles), persons, helmets, and no-helmet violations. The traffic analysis module counts vehicles, calculates traffic density, and detects possible triple-riding violations. Results, including vehicle labels, bounding boxes, traffic density, no-helmet counts, triple-riding status, and detection accuracy, are then displayed to the user.

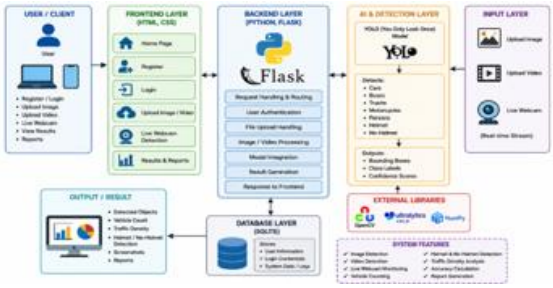

Fig 1: System Architecture

METHODLOGY DESCRIPTION

The AI-Based Smart Traffic Violation Detection System automates traffic monitoring by processing uploaded images, videos, and live webcam feeds. Using OpenCV for preprocessing and the YOLOv8 deep learning model for real-time detection, it identifies vehicles, pedestrians, helmets, and no-helmet violations. The system detects helmet use and triple-riding by analyzing motorcycle riders and counts vehicles to estimate traffic density. Built with Flask, it offers secure user registration and an interactive dashboard for uploading media and viewing live streams. Detection results are displayed with labeled bounding boxes, traffic density, violation counts, and accuracy reports. Data is securely stored in an SQLite database. This approach reduces manual effort while providing fast, reliable, and real-time traffic violation detection to improve road safety.

RESULTS AND DISCUSSION

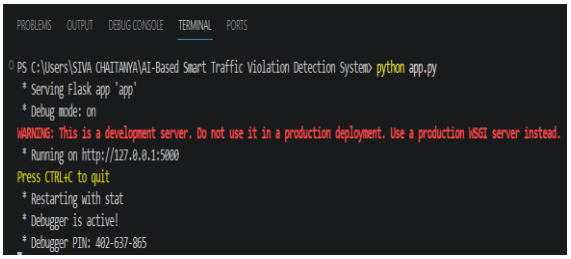

Fig 1: running app.py

The AI Traffic Violation Detection System is executed by running the python app.py command in the Visual Studio Code terminal. The Flask development server starts successfully and hosts the application on http://127.0.0.1:5000. This confirms that the application is ready for image, video, and live traffic violation detection.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
SSN: 3068-272X www.ijdim.com Original Research Paper

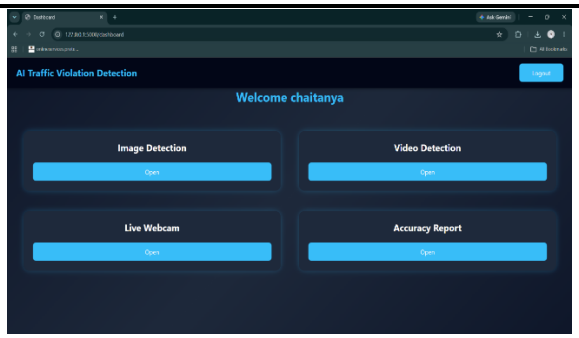

Fig 2: Dashboard page

The dashboard serves as the main interface of the AI Traffic Violation Detection System after user login. It provides access to the Image Detection, Video Detection, Live Webcam, and Accuracy Report modules through a user-friendly interface. This enables users to easily perform traffic violation detection and view analysis results from a single dashboard.

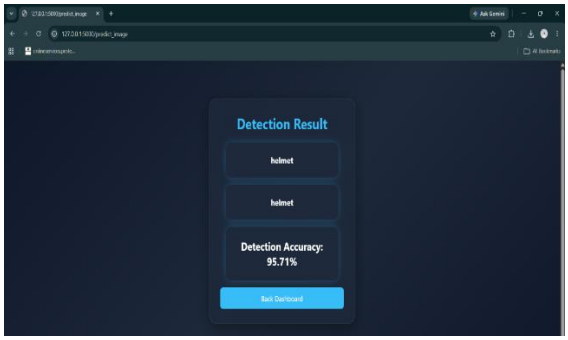

Fig 3: Image Detection Result

The image detection module analyzes the uploaded image using the YOLO model and identifies the presence of a helmet. The system displays the detected object along with the detection accuracy score of 85.71%. This confirms that the image has been successfully processed and the detection results are generated accurately.

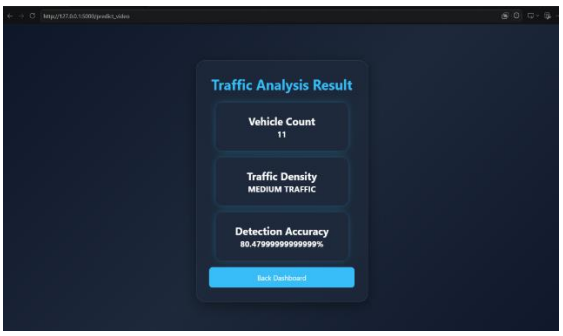

Fig 4: Video Detection Result

The video detection module analyzes the uploaded traffic video using the YOLO model and provides detailed traffic analysis results. It displays the vehicle count, traffic density level, and detection accuracy after processing the video. This enables efficient traffic monitoring and supports automated traffic management by providing accurate analytical information.

Fig 5: Live Traffic Monitoring

The live webcam monitoring module captures real-time video and detects vehicles using the YOLO model. It displays

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
SSN: 3068-272X www.ijdim.com Original Research Paper

the traffic density, helmet violation count, and triple-riding status while continuously analyzing the live video stream. This enables real-time traffic surveillance and helps improve road safety through automated violation detection.

CONCLUSION

The AI-Based Traffic Violation Detection and Analysis System uses YOLO-based object detection to monitor traffic and detect violations from images, videos, and live streams. It identifies vehicles, motorcycles, and persons, reducing manual monitoring and enabling faster analysis. The system calculates vehicle count, traffic density, and detection confidence while detecting helmet use, no-helmet violations, and possible triple-riding. Built with Flask, Python, OpenCV, and SQLite, it offers secure login, registration, live monitoring, and result display via a web platform. Its modular design supports easy maintenance and future upgrades like number plate recognition and automatic fine generation.

FUTURE SCOPE

Future improvements can use larger deep learning models trained on diverse Indian road traffic datasets to improve detection in low light, rain, crowded roads, different camera angles, and moving vehicles. Automatic Number Plate Recognition can be added to identify vehicle registration numbers and generate violation reports or challans. Future versions can detect signal jumping, wrong-side driving, illegal parking, lane violations, seat-belt violations, mobile phone usage while driving, and overspeeding. The system can also be integrated with cloud storage and a mobile application for remote monitoring, report access, and detection history. SMS, email, and mobile alerts can notify traffic authorities about major violations in real time.

REFERENCES

[1] D. P. Harini, “Electronic Health Record Stage Identification using Principal Component Analysis,” *Machine Intelligence Research,* vol. 18, no. 01, pp. 864- 873, 17 8 2024.

[2] D. P. Harini, “Two Level Intrusion Detection For Detecting Intruders in Multitier Web Applications,” *International Journal of Engineering & Science Research,* vol. 3, no. 9, pp. 472-478, 2013.

[3] D. P. Harini, “Analyze and Predict of Human Cyber Attackers using Artificial Neural Network,” *Journal of Basic Science and Engineering,* vol. 21, no. 1, pp. 1529 - 1536, 8 7 2024.

[4] OpenCV, “Open Source Computer Vision Library Documentation,” OpenCV

Foundation. [Online]. Available: OpenCV Documentation.

[5] Pallets Projects, "Flask Web Framework Documentation," Flask Documentation. [Online]. Available: Flask Documentation.

[6] Python Software Foundation, "Python Programming Language Documentation," Python Documentation. [Online]. Available: Python Documentation.

[7] PyTorch Foundation, "PyTorch Deep Learning Framework Documentation," PyTorch Documentation. [Online]. Available: PyTorch Documentation.

[8] R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed. Cham, Switzerland: Springer, 2022.

[9] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. Boston, MA, USA: Pearson, 2018.

[10] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Hoboken, NJ, USA: Pearson, 2020.

[11] SQLite Consortium, "SQLite Database Engine Documentation," SQLite Documentation. [Online]. Available: SQLite Documentation.

[12] GitHub, "GitHub Documentation," GitHub Documentation. [Online]. Available: GitHub Documentation.

[13] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.

[14] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 7464–7475.

[15] Ultralytics, "Ultralytics YOLO: Object Detection and Image Segmentation Framework," GitHub Repository. [Online]. Available: GitHub.

[16] Shashank, A. (2025). Self-Healing Data Pipelines for Enhanced Reliability: A Paradigm Shift in Enterprise Data Management. Journal of Computer Science and Technology Studies, 7(8), 1097-1104.