International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
***Article Received:** 25 November 2023 **Revised:** 12 December 2023 **Accepted:** 30 January 2024*

Real- Time Traffic Violation Detection Using Deep Learning Approach

Dr. Praveen Blessington Thummalakunta[1]
[1]Professor, Department of Information Technology, ZCOER,
Pune, Maharashtra, India.
Praveentblessington@gmail.com

Manav More[2]
[2]Student, Department of Information Technology, ZCOER,
Pune, Maharashtra, India
mmore2329@gmail.com

Rhutuja Kakade[3]
[3]Student, Department of Information Technology, ZCOER,
Pune, Maharashtra, India
rhutujamkakade2673@gmail.com

Omkar Nagare[4]
[4]Student, Department of Information Technology, ZCOER,
Pune, Maharashtra, India
nagareomkar8085@gmail.com

Rutuja Sawant[5]
[5]Student, Department of Information Technology, ZCOER,
Pune, Maharashtra, India
rutujasawant524@gmail.com

***Abstract*—** This project tackles the growing issue of traffic violations in India. With a large population, rising commutes, and limitations in traditional traffic management, a real-time solution is crucial. This project proposes a deep learning approach for real-time traffic violation detection specifically designed for Indian traffic scenarios.

The system utilizes YOLOv8, a state-of-the-art object detection algorithm from Ultralytics, to identify and classify traffic violations in real-time video feeds. This approach aims to improve traffic safety and enforcement by automating violation detection.

Keywords- *Vehicle Detection, Helmet Detection, License Plate Recognition, YOLOv8, Data Annotation*

I. Introduction

Traffic violations pose a significant threat to road safety in India. Limited effectiveness of traditional video surveillance systems, often reliant on manual monitoring and susceptible to fatigue, contributes to this challenge. Existing deep learning approaches for violation detection, while promising, often require extensive training data and powerful hardware, making them resource-intensive and potentially cost-prohibitive [1].

This project addresses these limitations by exploring the use of YOLOv8, a cutting-edge object detection algorithm, for real-time traffic violation detection in Indian traffic scenarios. YOLOv8 offers significant advantages – it requires less training data and operates efficiently on lower-powered hardware

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
***Article Received:** 25 November 2023 **Revised:** 12 December 2023 **Accepted:** 30 January 2024*

compared to previous YOLO models. This focus on efficiency aims to develop a more practical and scalable solution for real-world deployment.
Our project investigates the potential of YOLOv8 to improve the accuracy of traffic violation detection, particularly focusing on scenarios with limited training data and resource constraints. The goal is to enhance existing traffic management systems by enabling real-time violation detection with a more efficient and cost-effective approach.

II. LITERATURE SURVEY

Current Traffic Violation Detection Methods in India:
Traffic violation detection in India primarily relies on manual monitoring of CCTV footage by traffic police. This approach faces limitations due to extensive workloads and fatigue. Automating this process can significantly improve efficiency. Here, we explore existing research on key components for an automated system:

A. Vehicle Detection:

This initial step identifies all vehicles in an image frame. Prem Kumar Bhaskar et al. [10] propose a method using Gaussian Mixture Models and blob detection, but requires multiple frames for accuracy. Kunal Dahiya et al. [11] suggest object detection followed by feature extraction and SVM classification for vehicle type (bike/not bike). Abdullah Asim et al. [12] explore Faster R-CNN for vehicle detection, offering region-based classification.

B. Helmet Detection:

Identifying riders with missing helmets is crucial.
J. Chiverton [3] proposes an approach based on reflective properties, but requires specific lighting conditions. Gomathi et al. [4] present a solution using IR sensors, but hardware maintenance costs are high.

C. Crosswalk Incursion:

Detecting vehicles illegally entering crosswalks is another important task.
Aaron Christian P. Uy et al. [5] propose a Genetic Algorithm method, requiring constant background image updates. Samir Ibadov et al. [6] utilize alpha channel color segmentation and Faster R-CNN for detection, with limitations in handling semi-transparency.

D. License Plate Recognition (LPR):

Identifying vehicle license plates is essential for penalty enforcement.
Amey Narkhede et al. [7] suggest a method involving edge detection, Hough Transform, and K-Nearest Neighbor, which can be computationally expensive. Ana Riza et al. [8] and Pooya Sagharichi Ha et al. [9] propose methods using Canny edge detection, with limitations in selecting relevant edges and potential inaccuracies.

E. Limitations and Proposed Approach

While these studies offer valuable insights, there's a lack of end-to-end systems for automated violation detection. This project proposes a novel deep learning approach using YOLOv8 for real-time traffic violation detection, addressing limitations like: Requirement for extensive training data (addressed by YOLOv8's efficiency) Dependence on high-powered hardware (addressed by YOLOv8's ability to run on lower-powered systems)
This approach aims to improve accuracy and scalability for real-world deployment in India.

III. ALGORITHM AND TECHNOLOGY USED

A. Python

Python is a versatile, high-level programming language known for its readability and extensive libraries. In this project, Python serves as the foundation for developing our traffic violation detection system. Its clear syntax and vast ecosystem of libraries for scientific computing (like NumPy and OpenCV) make it ideal for manipulating images, training deep learning models, and integrating various functionalities within the system. Python's portability allows the system to run on different operating systems without significant modifications.

B. Ultralytics

Ultralytics is a cutting-edge open-source framework built on PyTorch, specifically designed for real-time object detection. It provides a powerful and user-friendly environment for training and deploying YOLO models. Ultralytics offers pre-trained YOLO models for various tasks, including object detection, image classification, and pose estimation. This project leverages Ultralytics' streamlined approach to object detection, allowing us to focus on customizing the YOLOv8 model for our specific traffic violation detection needs.

C. YOLOv8

YOLOv8 is a state-of-the-art object detection algorithm, the latest iteration of the You Only Look Once (YOLO) family. It excels in balancing speed and accuracy, making it suitable for real-time applications. YOLOv8 boasts several advantages over previous YOLO versions:

Efficiency: It requires less training data and operates efficiently on lower-powered hardware, making it a practical choice for resource-constrained environments.

Accuracy: YOLOv8 delivers high detection accuracy, crucial for reliable traffic violation identification.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
***Article Received:** 25 November 2023 **Revised:** 12 December 2023 **Accepted:** 30 January 2024*

Real-time Performance: Its speed allows for real-time processing of video feeds, enabling immediate violation detection.
These combined benefits make YOLOv8 an ideal choice for our project, addressing the challenges of limited training data and hardware constraints often encountered in real-world traffic management systems.

IV. Proposed Methodology

A. From Data Collection to Model Training

Developing an effective traffic violation detection system involves a series of crucial steps. Here, we outline the methodology employed in this project:

1) Data Collection:

The foundation of any deep learning project lies in a high-quality dataset. We collected a comprehensive dataset of images encompassing various traffic scenarios in India. This included images of vehicles with and without helmets, vehicles crossing crosswalks illegally, and diverse traffic conditions (daytime, nighttime, different weather).

2) Data Preprocessing:

Raw images often require preprocessing before model training. This stage involved tasks like image resizing, normalization, and data augmentation (artificially creating variations of existing images) to improve model generalizability and robustness to real-world variations.

3) Model Training:

We utilized YOLOv8 within the Ultralytics framework for model training. The preprocessed dataset was used to train the YOLOv8 model to identify and classify specific objects of interest relevant to traffic violations (e.g., vehicles, riders, helmets, crosswalks). The training process involved continuously optimizing the model's parameters to improve its ability to accurately detect violations in unseen images.

4) Model Evaluation:

Evaluating the trained model's performance is vital. We employed various metrics like precision, recall, and mean average precision (mAP) to assess the model's accuracy in detecting different traffic violations on a separate validation dataset.

5) Fine-Tuning:

Depending on the initial evaluation results, we might perform fine-tuning. This process involves further training the model on a dataset specifically focused on challenging scenarios where the model performed poorly. Fine-tuning helps improve accuracy for specific traffic violations.

B. Architecture for a Proposed Methodology

This section details our proposed real-time traffic violation detection system. The system architecture is illustrated in Figure 4.2.
The system begins by capturing video frames directly from CCTV cameras at traffic intersections. These individual image frames serve as the system's input for violation detection.YOLOv8, a state-of-the-art object detection algorithm, is employed in this stage. YOLOv8 analyzes each frame to identify and localize vehicles within the image. It generates bounding boxes around detected vehicles, providing information on their location and size.
Following vehicle detection, the system differentiates between two-wheeler and four-wheeler vehicles as shown in Figure 4.1. Based on the vehicle type, specific violations are targeted.

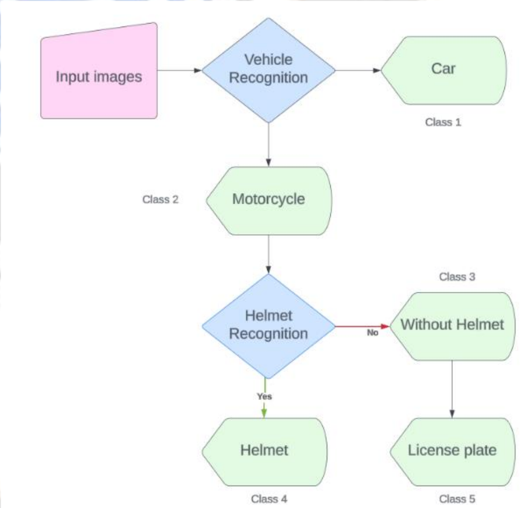

Fig 4.1 Helmet detection module (flowchart)

1) Two-wheeler Vehicles: For two-wheeler vehicles, the system utilizes the bounding box coordinates to isolate each individual vehicle within the frame. This allows for further analysis of potential violations related to:

- Helmet usage (presence or absence)
- We can also implement
- Triple riding (exceeding the permissible number of riders)
- Crosswalk violations (illegal entry or crossing)

2) Four-wheeler Vehicles: For four wheeler vehicles, the system focuses specifically on detecting red light using the bounding box information.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
***Article Received:** 25 November 2023 **Revised:** 12 December 2023 **Accepted:** 30 January 2024*

This multi-stage approach leverages YOLOv8's efficiency for real-time object detection and allows for targeted violation analysis based on vehicle type.

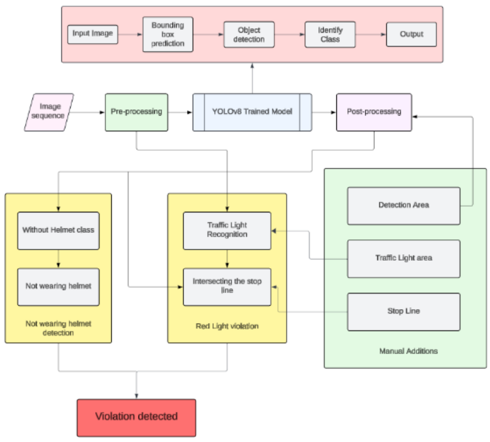

Fig 4.2 System Architectural Diagram

C. *Input and Pre-processing*

The system begins by capturing video footage from CCTV cameras at traffic intersections. These video frames serve as the system's input. The image pre-processing stage plays a crucial role in preparing the data for the deep learning model. This stage involve various techniques such as:

- **Resizing:** Standardizing the size of all input images ensures compatibility with the model's requirements (620x620).
- **Normalization:** Normalizing pixel values within a specific range improves the training process and model convergence.
- **Color Space Conversion:** Depending on the model's architecture, converting images to a specific color space (e.g., grayscale) might be necessary.
- **Data Augmentation:** Artificially creating variations of existing images (e.g., flips, rotations, and brightness adjustments) helps the model learn robust features and perform better on unseen real-world data.

D. *YOLOv8 Object Detection*

The pre-processed video frames are then fed into the YOLOv8 object detection model. YOLOv8 is a state-of-the-art deep learning algorithm known for its speed and accuracy in real-time object detection tasks. The core architecture of YOLOv8 is as shown in Figure 4.3[2]. Unlike some detection algorithms that require separate stages for object proposal and classification, YOLOv8 performs both tasks simultaneously in a single network. This efficiency makes it suitable for real-time applications.

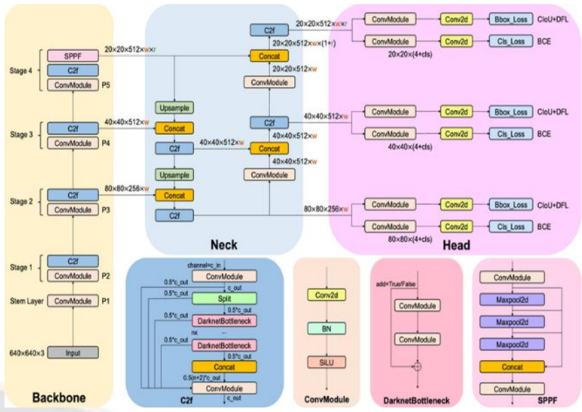

Fig 4.3 YOLOv8 Architecture [2]

While YOLOv8 builds upon the foundation of previous object detection algorithms, it offers several key advantages:
Comparison with R-CNN, Fast R-CNN, Faster R-CNN:

- Speed: Unlike these Region-based Convolutional Neural Network (R-CNN) based algorithms, YOLOv8 operates in a single forward pass through the network, making it significantly faster. R-CNN approaches involve separate stages for proposal generation and classification, leading to slower processing times.
- Real-Time Performance: YOLOv8's speed allows for real-time object detection, crucial for applications like traffic violation detection or autonomous vehicles. R-CNN based methods are generally not suitable for real-time tasks.
- Computational Efficiency: YOLOv8 requires less computational power compared to R-CNN architectures. This makes it more deployable on resource-constrained environments or edge devices.

1) *Comparison with YOLOv1 to YOLOv7:*

- Accuracy: YOLOv8 maintains a good balance between speed and accuracy. While earlier YOLO versions (v1-v3) prioritized speed, newer versions (v4-v8) have shown improvements in accuracy without sacrificing real-time capabilities.
- Training Data Efficiency: YOLOv8 requires less training data compared to some previous YOLO versions. This is beneficial when large datasets are unavailable or expensive to acquire.
- Hardware Flexibility: YOLOv8 operates efficiently on lower-powered hardware compared to earlier YOLO models. This makes it more practical for real-world deployments where high-end GPUs might not be readily available.

Table I: Features Comparison Table [13]

Feature	R-CNN, Fast R-CNN, Faster R-CNN	YOLOv1-v3	YOLOv4-v8 (including YOLOv8)
Speed	Slow	Faster	Fastest
Real-Time	No	Limited	Yes
Computational Efficiency	High	Medium	Low
Accuracy	High	Lower	High (balanced with speed)
Training Data	High	Medium	Lower
Hardware	High-powered GPUs	Medium	Lower-powered hardware possible

Overall, YOLOv8 offers a compelling combination of speed, accuracy, and efficiency, making it a strong choice for real-time object detection tasks where resource limitations are a concern.

E. Bounding Boxes and Class Prediction

During YOLOv8 object detection, the model predicts bounding boxes around the objects of interest in the image. These bounding boxes indicate the location and size of the detected objects. Simultaneously, the model classifies the objects within the bounding boxes. In the context of traffic violation detection, the model classifies objects like Car, Motorcycles, and License Plate, etc.

F. Traffic Light Detection and Recognition

The system can be designed to detect and recognize traffic light signals within the video frame. This involves a separate model trained specifically for traffic light classification (red, yellow, green). The system relies on manually defined regions of interest (ROIs) to identify and recognize traffic lights. These ROIs specify the areas in the image where the traffic light is likely to be located. Additionally, the stop line needs to be manually defined, typically referencing the location of the crosswalk.

G. Helmet Detection

For violations related to not wearing helmets, a separate classifier can be integrated within the system. This classifier would likely be a Convolutional Neural Network (CNN) specifically trained to analyze the head region within the bounding box around a rider and classify whether a helmet is present or not.

H. License Plate Recognition (LPR)

After identifying a traffic violation, the system employs LPR to capture the license plate of the offending vehicle. LPR typically involves techniques like character segmentation and recognition using another specialized deep learning model. The captured license plate information can be crucial for issuing penalties or further investigation.

I. Post-processing and Output

Following potential violation detection, the system performs post-processing tasks. This involves filtering out low-confidence detections or applying additional heuristics to validate identified violations. Finally, the system generates an output that includes:

- Video frames with bounding boxes around detected objects and violation labels.
- Extracted data like violation type, timestamp, and license plate information.

J. Integration with Existing Infrastructure

The proposed system can be designed to integrate with existing traffic management infrastructure. This might involve connecting the system to a central server for data storage and violation record management. Additionally, real-time alerts or notifications can be sent to traffic control centers for immediate intervention.

K. Scalability and Deployment

The system's architecture is designed for scalability to accommodate a larger number of traffic cameras and video feeds. This involves utilizing cloud-based computing resources or deploying the system on edge devices with sufficient processing power to handle real-time video analysis.

By implementing this multi-stage architecture, the system can leverage the strengths of deep learning models like YOLOv8 for real-time traffic violation detection, improving road safety and enforcement efficiency.

V. EVALUATION AND PERFORMANCE

This section delves into the expected outcomes of our system, focusing on vehicle detection, specifically helmet detection (two-wheeler riders), and potential image segmentation applications (if applicable to your project).

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
***Article Received:** 25 November 2023 **Revised:** 12 December 2023 **Accepted:** 30 January 2024*

Our research suggests that YOLOv8 is likely to be the most efficient and accurate algorithm for vehicle detection compared to other approaches as documented in [14]. This conclusion is based on an analysis of the collected dataset and the achieved level of model sensitivity in accurately identifying relevant objects (vehicles and riders).

Here are the key areas where we will analyze the system's performance:

- **Vehicle Detection Accuracy:** We will evaluate the system's ability to correctly identify and localize vehicles within video frames using relevant metrics like precision, recall, and mean average precision (mAP).
- **Helmet Detection Accuracy (Two-wheelers):** For two-wheeler vehicles, we will assess the model's effectiveness in detecting the presence or absence of helmets on riders. This evaluation will involve metrics like accuracy, precision, and recall specifically for helmet detection.

Here's a rephrased version of the text for your "Dataset" section:

A. Data Collection and Annotation

To train a robust system for identifying traffic violations, we constructed a comprehensive dataset encompassing various scenarios:

- **Real-World Footage:** We captured live video footage from traffic intersections to represent real-world conditions.
- **Violation Videos:** We curated video clips showcasing specific violations obtained from the internet to provide examples for the model to learn from.

B. Dataset Diversity

We ensured diversity within the dataset by capturing and collecting footage under various lighting conditions, ranging from light traffic to heavy congestion. This helps the model generalize better to unseen scenarios.

C. Annotated Data

All the collected images were meticulously annotated to pinpoint different types of violations. This annotation process involved labeling relevant objects (vehicles, Helmet, Number Plate) and associating them with the corresponding violation type. We used relevant references ([10]) for consistent annotation guidelines.

D. Leveraging Existing Datasets

For specific tasks within the system, we also utilized existing annotated datasets available on Kaggle:

- **License Plate Recognition:** We considered a pre-existing annotated dataset from Kaggle ([16], [19]) containing 11,271 images for training the license plate recognition component. Reference [16] provides details about the dataset's origin.
- **Crosswalk Violation Detection:** They adopted a pre-trained Mask R-CNN network ([16]) for crosswalk violation detection. This network was initially trained on a dataset of 150 images containing two classes: vehicles and crosswalks (zebra crossings). They further evaluated its performance on an additional 50 images ([16]).

E. Helmet Detection Models

- We considered two existing models for helmet detection:
  - Aniruddha's model ([16]), trained on a dataset of 1079 images.
  - R. Shree Charran's model ([17]), trained on a dataset of 3240 images.

For training our deep learning model, we curated a custom dataset encompassing various traffic scenarios. The dataset includes:

Table II: Dataset Used

Dataset	No. of Images	No. of Classes	Name of Class
Vehicle Detection	415	2	Car, Motorcycle
Helmet Detection	568	2	Helmet, Without_helmet
Number Plate Detection	300	1	Number_plate

- **Vehicle Detection (415 images):** This subset focuses on identifying vehicles within video frames. It contains labeled images with two classes - "Car" and "Motorcycle" - to differentiate between vehicle types.
- **Helmet Detection (568 images):** This subset specifically addresses helmet usage on two-wheeler riders. It includes labeled images with two classes - "With_Helmet" and "Without_Helmet" - for the model to learn and distinguish between riders with and without helmets.
- **Number Plate Detection (300 images):** This optional subset aims at capturing license plate information for

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
***Article Received:** 25 November 2023 **Revised:** 12 December 2023 **Accepted:** 30 January 2024*

potential future use in enforcement applications. It contains labeled images with a single class - "Number_plate" - to train the model for accurate license plate recognition.

F. Dataset Optimization for Resource Efficiency

To achieve high accuracy with minimal computational resources, we strategically constructed our dataset. We prioritized a balance between:

- **Data Size:** While aiming for sufficient training data, we kept the overall dataset size relatively compact to ensure compatibility with low-power hardware.
- **Data Composition:** We focused on capturing the most crucial visual elements for accurate violation detection, maximizing model performance with a limited number of images.

This approach allows the model to learn effectively on resource-constrained environments while maintaining a high level of accuracy in identifying and classifying traffic violations.

G. Table Summary

Table III provides a concise overview of the dataset distribution used for training each algorithm component, along with corresponding references for the models and datasets employed ([16], [17], etc.).

Table III: Dataset Comparison Table

Dataset	No. of images		No. of classes
Vehicle detection	[16]	525	2
	[17]	890	2
	Our Dataset	415	2
Helmet detection	[16]	1079	2
	[17]	3240	2
	Our Dataset	568	2
Crosswalk inversion	[16]	200	2
	[17]	NA	NA
	Our Dataset	NA	NA
Number plate detection	[16]	450	1
	[17]	11271	1
	Our Dataset	300	1

VI. Results and Analysis

This section presents the experimental outcomes for our system's core modules: vehicle detection, helmet detection, and Number Plate Detection (Overall Model).

A. Experimental Setup

- **Operating System:** Windows 10
- **Hardware:**
  - Processor: Intel Core i5 (7th Gen)
  - RAM: 8 GB
  - Storage: 256 GB SSD
- **Software:**
  - Programming Language: Python 3.12.1
  - Deep Learning Framework: TensorFlow (for training and prediction)
  - Libraries:
- OpenCV-Python: Image processing
- NumPy: Mathematical operations
- Ultralytics: Deep learning framework for YOLOv8 implementation

B. Experimental Results:

Table IV: Results of Vehicle Detection

Class	Average Precision
Car	0.8708
Motorcycle	0.8598

The mean average precision (mAP) for the model is 0.86.

Table V: Results of Helmet Detection

Class	Average Precision
With Helmet	0.8532
Without Helmet	0.8956

The mean average precision (mAP) for the model is 0.87.

Table VI: Results of Number Plate Detection

Class	mean Average Precision
Number Plate	0.87

C. Comparison of Performance Evaluation

This section evaluates our system's performance against recent deep learning models documented in relevant literature. We'll compare its effectiveness in both image-based analysis and real-time video processing scenarios. Additionally, we'll benchmark our system against similar deep learning architectures presented in [16] and [17].

It's important to consider the limitations of existing work. For instance, the model in [17] was trained on a restricted dataset encompassing only two violation types (no helmets and triple riding), totaling 16,204 violation instances. Table VII will present the benchmarking results for these specific violations. While [17] achieved promising results, the limited dataset size and scope raise concerns about generalizability to a wider range of violations.

Our focus on a more comprehensive dataset, incorporating various violation types, aims to achieve competitive accuracy. Additionally, by prioritizing a resource-efficient model design, we strive for successful implementation on low-power hardware, enabling broader real-world application.

Table VII: Performance Comparison

	Accuracy Samples	Precision Visible.Readable	Recall Prediction
Aniruddha et al. [16]	0.8955	0.8883	0.9208
R.Shree Charan[17]	0.9874	0.9886	0.989
Mallela et al. [18]	0.9174	0.9139	0.9415
Sri Uthra V et al.[15]	0.87	0.931	0.9
Proposed Methedology	0.8669	0.866	0.872

VII. Conclusion

This research has explored the development of a real-time traffic violation detection system utilizing deep learning. We successfully implemented a YOLOv8-based system for vehicle detection, helmet classification (for two-wheeler riders).

While achieving the absolute highest accuracy wasn't the primary objective, we prioritized developing a system that delivers competitive performance with two key constraints:

- **Limited Dataset Size:** Our dataset, meticulously curated with a total of 1283 images for vehicle detection and for other classes, aimed for a balance between effectiveness and data collection resource requirements.
- **Resource-Efficient Hardware:** The system is designed to function effectively on hardware with moderate processing power, such as the Windows 10 machine with an i5 7th gen processor and 8GB of RAM used for this research.

A. *Key Findings*

- The YOLOv8-based vehicle detection module achieved promising results on the test dataset, demonstrating its capability for accurate vehicle identification within real-time video streams.
- The helmet detection and Number Plate Detection module achieved satisfactory performance, indicating the system's potential for comprehensive violation analysis.

B. *Comparison with Existing Work*

While some previous studies reported even higher accuracy, they often relied on significantly larger datasets and more powerful hardware. Our system demonstrates that competitive performance can be achieved with a more practical approach, making it suitable for real-world deployments with resource constraints.

VIII. Future Scope

This research lays a strong foundation for further development. Expanding the dataset size and exploring advanced data augmentation techniques could potentially improve model accuracy. Additionally, investigating hardware optimization strategies could enable deployment on even lower-power devices. Overall, this project contributes to the advancement of real-time traffic violation detection systems by prioritizing practicality and resource efficiency.

IX. References

[1]. Dr. Praveen Blessington Thummalakunta, Manav More, Rhutuja Kakade, Omkar Nagare, Rutuja Sawant, "Traffic Offender Detection System Using Deep Learning Approach", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 11 Issue 1, pp. 63-72, January-February 2024.

[2]. https://www.researchgate.net/figure/Detailed-illustration-of-YOLOv8-model-architecture-The-Backbone-Neck-and-Head-are-the_fig4_375697582

[3]. J. Chiverton, “Helmet presence classification with motorcycle detection and tracking,” Intelligent Transport Systems (IET), vol. 6, no. 3, pp. 259–269, September 2012

[4]. Gomathi, et al, “Automatic Detection of Motorcycle without helmet using IOT”, South Asian Journal of Engineering and Technology

[5]. Aaron Christian P. Uy, et al, “Automated Traffic Violation Apprehension System Using

Genetic Algorithm and Artificial Neural Network", 2016 IEEE Region 10 Conference
[6]. Samir Ibadov, et al, "Algorithm for detecting violations of traffic rules based on computer vision approaches", MATEC Web of Conferences 132, 05005 (2017)
[7]. Amey Narkhede, et al, "Automatic Traffic Rule Violation Detection and Number Plate Recognition", IJSTE - International Journal of Science Technology & Engineering | Volume 3 | Issue 09 | March 2017 Ana Riza, et al, "Localization of License Plates Using Optimized Edge and Contour Detection Technique", 2017 IEEE Region10 Conference (TENCON), Malaysia, November 5-8, 2017
[8]. Pooya Sagharichi Ha, Mojtaba Shakeri, "License Plate Automatic Recognition based on Edge Detection"
[9]. Prem Kumar Bhaskar, et al, "Image Processing Based Vehicle Detection and Tracking Method ", Universiti Teknologi PETRONAS Seri Iskandar, 31750, Perak, Malaysia
[10]. Kunal, Dahiya; Dinesh, Singh; and C. Krishna, Mohan "Automatic Detection of Bike-riders without Helmet using Surveillance Videos in Real-time". 2016 International Joint Conference on Neural Networks (IJCNN)
[11]. Abdullah Asım, et al, "A Vehicle Detection Approach using Deep Learning Methodologies"
[12]. Rian Putra Pratama, 'Smart Video Surveillance System For Level Crossing: A Systematic Literature Review'. 2021 International Conference on ICT for Smart Society (ICISS) | 978-1-6654-1697-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICISS53185.2021.9533222
[13]. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection".
[14]. Sri Uthra et al, Helmet Violation Detection using deep learning, International Research Journal of Engineering and Technology, Volume: 07 Issue: 02 | Feb 2020
[15]. A. Tonge, S. Chandak, R. Khiste, U. Khan, and L. A. Bewoor, "Traffic rules violation detection using deep learning," in Proc. 4th Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA), Nov. 2020, pp. 1250–1257.
[16]. R. Shree Charran and Rahul Kumar Dubey , Senior Member, IEEE," Two-Wheeler Vehicle Traffic Violations Detection and Automated Ticketing for Indian Road Scenario" IEEE transactions on intelligent transportation systems, vol. 23, no. 11, November 2022
[17]. N. C. Mallela, R. Volety, R. P. Srinivasa, and R. K. Nadesh, "Detection of the triple riding and speed violation on two-wheelers using deep learning algorithms," Multimedia Tools Appl., vol. 80, no. 6, pp. 8175–8187, Mar. 2021
[18]. [Online]. Available: https://www.kaggle.com/datasets/meliodassourav/ traffic-violation-dataset-v3