International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048
Volume 9, Issue 6, pp 119-127, June-2025
https://doi.org/10.47001/IRJIET/2025.906014

Smart Traffic Monitoring and Violation Detection with YOLOv8 Algorithm

[1]Ajeet Singh Bhadoriya, [2]Dr. Raghvendra Upadhyay

[1]Student, Robotics and Artificial Intelligence (M. Tech), JSPM University, Wagholi, Pune, Maharashtra, India
[2]Professor, Robotics and Artificial Intelligence, JSPM University, Wagholi, Pune, Maharashtra, India

Abstract **- Smart Traffic Monitoring and Violation Detection with YOLOv8 Algorithm involves leveraging advanced deep learning techniques to automate the identification of motorcycle riders' helmet usage and vehicle license plates in real-time. This technology enhances traffic monitoring and safety enforcement by providing accurate and efficient detection capabilities. The project focuses on developing a robust system for detecting helmets and number plates using the YOLOv8 architecture, which is known for its high accuracy and speed in real-time applications. The system aims to address the increasing number of motorcycle accidents due to non-compliance with helmet usage, emphasizing the importance of safety gear in preventing head injuries. By utilizing YOLOv8, the detection process is optimized for various environmental conditions, ensuring reliable performance even in challenging scenarios such as poor lighting or occlusions. A comprehensive dataset of annotated images is created, featuring diverse scenarios to train the YOLOv8 model effectively. The model is fine-tuned using transfer learning techniques to enhance its detection capabilities for both helmets and number plates. Experimental results indicate that the YOLOv8 model achieves high accuracy rates in detecting helmets and number plates, making it suitable for deployment in intelligent transportation systems.**

Keywords: Deep Learning, Traffic Monitoring, YOLOv8 model, etc.

I. INTRODUCTION

The project titled "Smart Traffic Monitoring and Violation Detection with YOLOv8 Algorithm" employs advanced machine learning and computer vision techniques to address critical aspects of traffic management and safety enforcement. The system is developed using Python and leverages the powerful YOLOv8 (You Only Look Once, Version 8) object detection architecture for real-time identification of helmets and vehicle number plates. The front-end interface is built with HTML, CSS, and JavaScript, and is supported by the Flask web framework, providing a responsive and user-friendly environment for interacting with the detection system.

The YOLOv8 model, known for its speed and accuracy in object detection tasks, is trained on a diverse dataset to recognize helmets and number plates in various conditions and orientations. The model achieved a training accuracy of 88.00% and a validation accuracy of 79.00%, indicating a strong generalization capability to unseen data. These metrics reflect the model's proficiency in handling real-world variations and complexities in the input data.

The detection system operates in three distinct modes: 1) Image Mode: Allows users to upload and analyze static images to detect helmets and number plates. 2) Video Mode: Processes pre-recorded video files, extracting frames and applying the YOLOv8 model to detect and annotate the objects of interest throughout the video sequence. 3) Web Camera Mode: Utilizes a live feed from a web camera to perform real-time detection, making it suitable for dynamic and on-the-fly monitoring applications.

The integration of these modes provides versatility, catering to a range of use cases from stationary analysis to live traffic surveillance. The backend, powered by Flask, ensures smooth data handling and processing, while the front-end design focuses on ease of use and accessibility.

In summary, this project demonstrates the effective application of deep learning in enhancing road safety and regulatory compliance. By automating the detection of helmet usage and vehicle identification, it offers a practical solution for traffic authorities to monitor and enforce safety regulations efficiently. The high accuracy rates achieved during training and validation phases underscore the potential of YOLOv8 in real-world deployment scenarios. The daily competition between different malls as well as big malls is becoming more and more intense because of the rapid rise of international supermarkets and online shopping’s. Every mall or mart tries to provide personal and short-term donations or benefits to attract more and more customers on a daily basis, such as the sales price of everything which is usually predicted to be managed through different ways such as corporate asset management, logistics, and transportation service, etc. Current

© 2025 IRJIET All Rights Reserved

ISSN (online): 2581-3048
Volume 9, Issue 6, pp 119-127, June-2025
https://doi.org/10.47001/IRJIET/2025.906014

machine learning algorithms that are very complex and provide strategies for predicting or predicting long-term demand for a company's sales, which now also help in overcoming budget and computer programs.

II. LITERATURE REVIEW

Dasgupta et al. discussed about the important part of an effective traffic management system is the ability to constantly monitor the compliance of vehicles with traffic regulations. Due to the large number of people living in urban areas, motorcycles can be one of the most common modes of transportation in India. It was noted that most motorcyclists refrained from using primary insurance for city traffic or even street driving. In case of an accident, wearing a helmet reduces the risk of head and brain injuries, according to many studies. Most traffic and safety rules are currently enforced through a traffic video surveillance camera framework that allows the rules can be seen through today's break. This article presents a practical solution to enhance the movement of one or more motorcycle passengers - or "double passenger" as the authors call it. YOL is used to check the target whenever someone is coming, such as a motorcyclist at the start of a test. At the initial review, the occupational level used is YOL3. Another brain network design, Convolutional Network, is designed to recognize motorcyclists using a strategy called design coordination and edge detection. The results show that the predictions of the CNN model in the same traffic videos are more promising than the predictions of other models [1].

Nitin et al. was introduced with Accidents and injuries recently increased due to the increased use of motorcycles which made it difficult to keep the roads clear. The fact that the motorcyclist did not wear a helmet is one of the main reasons for this. Currently, to determine the location of motorcyclists required by law to wear helmet, or a physical search or video surveillance camera recording of a different intersection provided by the ministry must be carried out. The proposal includes a computer design that allows you to look at pictures of cruise ships and identify people wearing hats of individuals not wearing head protection, allowing for more accurate identifiable signs of customers riding mechanized cycles. Mainly the machine receives the objects in the light of the elements and then removes them. The YOL-Dark architecture provides convolutional network deep learning models for object recognition and computer vision that uses convolutional neural networks trained on regular objects the Cena. The mechanism is implemented as a sliding window, and the wavelet layers of the YOL classifier are modified to separate the three known classes [2].

Goel et al achieved an average accuracy of 81%, which gives a much more accurate picture of (to a greater extent) the extent of the map. This article I was introduced to is about predicting helmet-free requirements based on two-lap cycling data that doesn't use centralized authentication. In addition, it improves the user experience thanks to prescription fees. FIFO (first-in-first-out) or best-in-one-first-out (ABIO) methods are used to initiate vehicle identification and identify vehicle incidents in trapped traffic. The separation is then performed using a two-in-two field (TIR) or least-first-field (LFO) method. In OpenCV, it determines whether cyclists and passengers are wearing helmets after determining whether drivers and passengers are present. Digital imaging is used to scan and track motorcycles without a helmet to mark potential drivers, passengers or motorcyclists as authorized (OCR). The fine and all the information will be sent to the named person after receiving the registration number of the vehicle. The owner of the vehicle will also receive an email and text message. A user can be given access to an account which can be a website or an application. This account allows the user to pay court costs [3].

Roopa et al. represented that the people have certain deterministic tendencies, such as ignoring what is significant, ignoring what does not contribute to an event, and finding fault with things that do not exist, are all examples of cause and-effect relationships. For people who know most causes of death in motor vehicle accidents and they should stay at home, head restraints and other protective equipment are also available. Is important welfare protection patience can be considered insignificant for the reason that few or no people use it often, or maybe no one has tried to check its functionality before, avoiding having to stay inside limit points and growth potential; the observation should be completed so that the potential does not increase during observation or control work ability. As there is a clear link between human activity and traffic flows, we are generally considered to be the main drivers of this. When the police enforce traffic rules, it is physically impossible for them to control traffic. Successful implementation of large projects is possible only with a small number of people, and much more would be needed. In such a situation, the number on the helmet must be considered: two samples are likely to emerge from the flat pile. Unlike the other extreme, which is based on expediency and would lead to their rejection [4].

III. SYSTEM DESIGN

The proposed system for "Smart Traffic Monitoring and Violation Detection with YOLOv8 Algorithm" aims to enhance and expand the capabilities of the existing system by incorporating the latest advancements in deep learning and computer vision. This system is designed to detect both safety helmets and vehicle number plates, providing a more

© 2025 IRJIET All Rights Reserved

ISSN (online): 2581-3048
Volume 9, Issue 6, pp 119-127, June-2025
https://doi.org/10.47001/IRJIET/2025.906014

comprehensive approach to traffic and workplace safety monitoring.

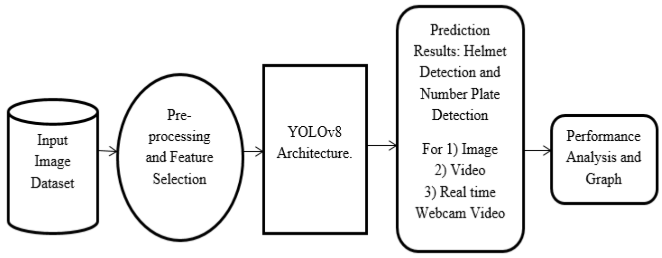


Figure 1: System Architecture

- YOLOv8 Architecture: The proposed system utilizes the YOLOv8 (You Only Look Once, Version 8) architecture, an improvement over the previous YOLOv5 model. YOLOv8 is known for its superior accuracy and efficiency in object detection tasks, capable of identifying multiple objects within a single frame more effectively.
- Python Implementation: The system is implemented in Python, leveraging its extensive machine learning and computer vision libraries such as TensorFlow and OpenCV. Python's versatility and ease of use facilitate the integration of complex algorithms and models.
- Dual Detection Capabilities: Unlike the existing system, which focused solely on helmet detection, the proposed system can detect both safety helmets and vehicle number plates. This dual capability extends the system's application to a broader range of safety and regulatory enforcement scenarios.
- Flexible Detection Modes: The system supports three modes of operation for detection:
- Image Mode: Allows users to upload and analyze static images to detect helmets and number plates.
- Video Mode: Processes pre-recorded video files, extracting frames and applying the YOLOv8 model to detect and annotate objects of interest throughout the video sequence.
- Web Camera Mode: Utilizes a live feed from a web camera for real-time detection, suitable for dynamic monitoring applications.
- Front-end Development: The user interface is developed using HTML, CSS, and JavaScript, ensuring a responsive and interactive experience. The front-end design prioritizes user-friendliness and accessibility, making it easy for users to interact with the detection system.
- Flask Web Framework: The system is integrated with the Flask web framework, which handles backend processes, including data handling, model inference, and communication between the user interface and the detection algorithms. Flask provides a lightweight yet powerful platform for deploying web applications.
- Training and Validation: The YOLOv8 model is trained on a diverse dataset encompassing various helmet types and vehicle number plates under different conditions. The training process aims to achieve high accuracy and robustness, ensuring reliable performance in real-world scenarios. The system achieved a training accuracy of 88.00% and a validation accuracy of 79.00%.
- Deployment and Scalability: The proposed system is designed for scalable deployment, capable of handling increased data loads and broader operational contexts. It is suitable for deployment in various environments, from individual workplaces to city-wide traffic monitoring systems.

In summary, the proposed system represents a significant enhancement over the existing one, leveraging advanced technologies to provide a comprehensive and versatile solution for helmet and number plate detection. It is built with the latest deep learning architectures, ensuring high performance and adaptability to various use cases.

IV. YOLO V8 ALGORITHM

A. Dataset Description

The Indian Helmet Detection Dataset contains 942 images of Indian Road Traffic including riders and their powered two-wheelers. The dataset has been further divided into train containing 800 images and valid including 142 images.

These images are captured in diverse locations across India, encompassing different lighting conditions, road infrastructure variations, and weather situations. Each image is accompanied by a corresponding annotation file in a standard format which specifies the following:

- 0: Number Plate.
- 1: Face with No Helmet.
- 2: Face with Good Helmet.
- 3: Face with Bad Helmet.
- 4: Rider.

The class distribution (number of images per category) is crucial information. An imbalanced dataset (unequal distribution of images across classes) might require specific strategies during training to ensure the model performs well for all helmet usage categories.

B. Architecture

YOLOv8 stands out for its efficient single-stage architecture, making it well-suited for real-time object detection tasks. Here’s a concise overview of its key components:

© 2025 IRJIET All Rights Reserved

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 6, pp 119-127, June-2025

https://doi.org/10.47001/IRJIET/2025.906014

Backbone Network: The foundation is a Convolutional Neural Network (CNN) backbone, often a modified Darknet variant (e.g., CSPDarknet53). This network extracts features from the input image at various levels of detail, providing a rich representation for object detection.

Neck Networks: Some YOLOv8 variants might include a neck network. This network refines the feature maps extracted by the backbone at different stages, combining them to create a more comprehensive image representation that incorporates information from various resolutions.

Head Network: The head network takes the processed feature maps (from the backbone and optionally the neck) and performs the final detections. It typically consists of several convolutional layers followed by fully connected layers. These layers predict bounding boxes for potential objects and classify them into predefined categories.

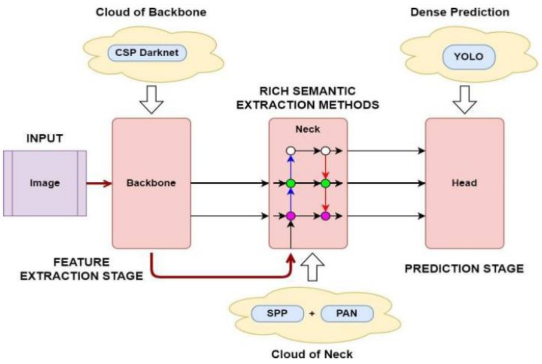


Figure 2: YOLO V8 Architecture

Thus, YOLOv8 takes an input image, feeds it through the backbone network for feature extraction optionally refines the features through the neck network, and finally utilizes the head network to predict bounding boxes and classify objects.

C. YOLOv8 Model Creation

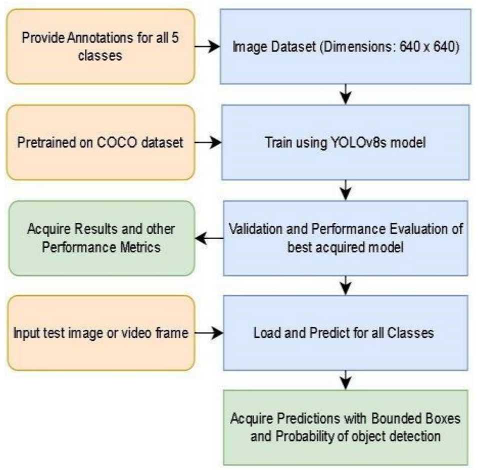


Figure 3: Process Flow Diagram

1. Training

From the described dataset, 800 images were used for training a YOLOv8s pre-trained model. The dataset includes motorcycle riders with and without helmets.

To increase model generalizability and prevent over-fitting, the image size was fixed. The model was trained with the parameters. The model was initially trained for 100 epochs but the best result was considered at the 48th iteration. Thus, to remove and redundancy and over-fitting introduced, the following train model was considered best: epochs=100: This parameter sets the number of cycles the entire training dataset will be trained and validated through the model. batch=16: This defines the number of images processed by the model in each training iteration. imgsz=640: This specifies the size (resolution) to which the training images will be resized before feeding them into the model.

optimizer=SGD: This indicates the optimizer algorithm used to update the model's weights during training. SGD (Stochastic Gradient Descent) is being used here.

lr=0.01: This is the initial learning rate, which controls the magnitude of updates to the model's weights.

momentum=0.937, weight decay=0.001: These are additional hyper-parameters used with the SGD optimizer to improve convergence and stability.

2. Testing

To evaluate the generalizability and real-world applicability of the trained YOLOv8 model, we conducted testing on unseen images not included in the training or validation sets. A single test image was processed, providing insights into the model's processing speed for each stage:

Pre-processing (0.6ms per image): This initial stage involves tasks like image resizing and normalization. The relatively fast preprocessing time indicates efficient handling of image data preparation.

Inference (16.3ms per image): This core step encompasses passing the pre-processed image through the YOLOv8 model to generate detections.

The inference speed is similar to the validation speed suggests potential consistency between controlled and real-world scenarios. Post-processing (98.3ms to under 2ms): This final stage varies depending on the image complexity and the specific post-processing tasks performed. It often involves decoding detections from the model's output format and applying non- max suppression (NMS) to remove redundant bounding boxes.

© 2025 IRJIET All Rights Reserved

ISSN (online): 2581-3048
Volume 9, Issue 6, pp 119-127, June-2025
https://doi.org/10.47001/IRJIET/2025.906014

V. IMPLEMENTATION

Modules Description

Data Collection:

- In the first module of the Helmet and number Plate detection using YOLOv8, we make the data collection process. This is the first real step towards the real development of a deep learning model, collecting data. This is a critical step that will cascade in how good the model will be, the more and better data that we get; the better our model will perform.
- There are several techniques to collect the data, like web scraping, manual interventions. The dataset is located in the model folder. The dataset is referred from the popular dataset repository called kaggle. The following is the link of the dataset:
- Kaggle Dataset Link: https://www.kaggle.com/datasets/jayaprakashpondy/helmet

Dataset:

- Organizing the data into train, val sets and Converting annotations to the format required by YOLOv8.
- Annotations typically include bounding boxes around the objects of interest (helmets and number plates) and their corresponding classes. YOLOv8 requires annotations in a specific format, such as YOLO format (class, x_center, y_center, width, height) and Total dataset size is 214 images 3 'With Helmet', 'Without Helmet', 'Number Plate'.

Data Preparation:

- Ensure that each set contains a representative sample of images to avoid overfitting. Convert annotations into the format required by YOLOv8, ensuring consistency between image paths and annotation paths.
- Resizing: Resize the images to a consistent size to ensure the model can process them efficiently.
- Normalization: Normalize the pixel values of the images to be within a specific range, usually between 0 and 1, to improve model performance.

Feature Extraction:

- In YOLOv8, feature extraction is handled by the model architecture itself, specifically by its convolutional layers. These layers extract relevant features from the input images, enabling the model to detect objects such as helmets and number plates.
- Use the YOLOv8 architecture, which is a state-of-the-art object detection model. YOLOv8 extracts features from the images using a backbone network and then predicts bounding boxes and class probabilities.

Splitting the dataset:

- Divide your dataset into training and validation to evaluate your model's performance. Typically, you might use an 80-20 split, but this can vary based on your dataset size and specific requirements.

Model Selection:

- The training module is responsible for training the deep learning models using the preprocessed data. It implements YOLOv8 architectures

YOLOv8:

- Selecting YOLOv8 as the object detection model for tasks like helmet and number plate detection is a strategic choice owing to its high accuracy and efficiency. YOLOv8, short for "You Only Look Once version 8," represents the latest iteration of the YOLO (You Only Look Once) family of object detection models, known for their real-time performance and strong detection capabilities.
- Here's a deeper exploration of why YOLOv8 is a compelling choice:
- State-of-the-art Performance: YOLOv8 builds upon the success of its predecessors and incorporates advancements in deep learning techniques and model architectures. It achieves state-of-the-art performance in terms of detection accuracy and speed, making it suitable for various real-world applications.
- Efficiency: YOLOv8 is designed to be highly efficient, striking a balance between accuracy and computational resources. Its architecture optimizes the use of hardware resources, allowing for fast inference speeds even on devices with limited computational power. This efficiency makes YOLOv8 particularly appealing for deployment in resource-constrained environments or for applications requiring real-time processing.
- Single-stage Detection: YOLOv8 follows the principle of single-stage detection, meaning it processes the entire image in a single forward pass through the network. This design choice eliminates the need for complex post-processing steps and significantly reduces inference time compared to multi-stage detection approaches.
- Multi-scale Feature Fusion: YOLOv8 incorporates multi-scale feature fusion techniques, enabling it to capture context and spatial information at different scales within the image. This enhances the model's ability to detect objects of varying sizes and aspect ratios with high accuracy.

© 2025 IRJIET All Rights Reserved

ISSN (online): 2581-3048
Volume 9, Issue 6, pp 119-127, June-2025
https://doi.org/10.47001/IRJIET/2025.906014

- Flexibility and Customization: YOLOv8 offers flexibility and customization options to adapt the model to specific use cases and datasets. Users can choose from different model variants (e.g., YOLOv8-s, YOLOv8-m, YOLOv8-l) based on their requirements for speed and accuracy. Additionally, the model can be fine-tuned on custom datasets to further improve performance on specific tasks.
- Open-source Implementation: YOLOv8 is often available as an open-source implementation, making it accessible to a wide range of developers and researchers. This fosters collaboration, experimentation, and innovation within the computer vision community.
- Overall, the selection of YOLOv8 as the object detection model for tasks like helmet and number plate detection reflects its reputation for achieving high accuracy and efficiency. By leveraging the strengths of YOLOv8, developers can build robust and effective detection systems capable of meeting the demands of various real-world applications.

Training the Model:

- Training loop: Train the YOLOv8 model using the training set. The model learns to predict bounding boxes and class probabilities for helmets and number plates.
- Loss function: Use a loss function such as mean average precision (MAP) to measure the model's performance during training.
- Optimizer: Use an optimizer such as stochastic gradient descent (SGD) or Adam to update the model's weights based on the loss.

Analyze and Prediction:

- Model evaluation: Evaluate the model's performance on the validation set during training to monitor its progress.
- Prediction: Use the trained model to predict bounding boxes and class probabilities for new, unseen images.

Accuracy on test set:

- Evaluate the final accuracy of the model on the test set to ensure its effectiveness in real-world scenarios. Calculate metrics like mAP, precision, and recall on the test set to quantify the model's performance objectively.

Saving the Trained Model:

- Save the trained YOLOv8 model for future use. YOLOv8 models can be saved in a format that allows easy reloading for inference and deployment in production environments.

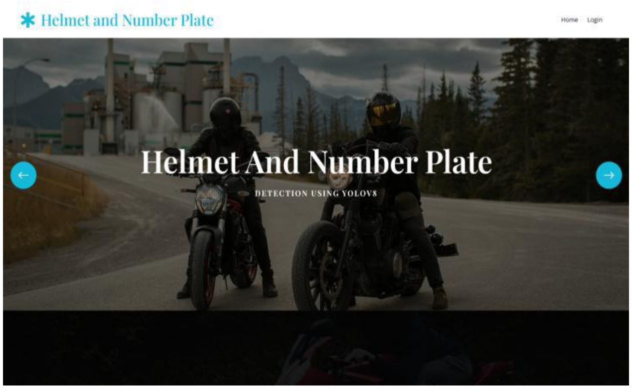
Prediction Module:

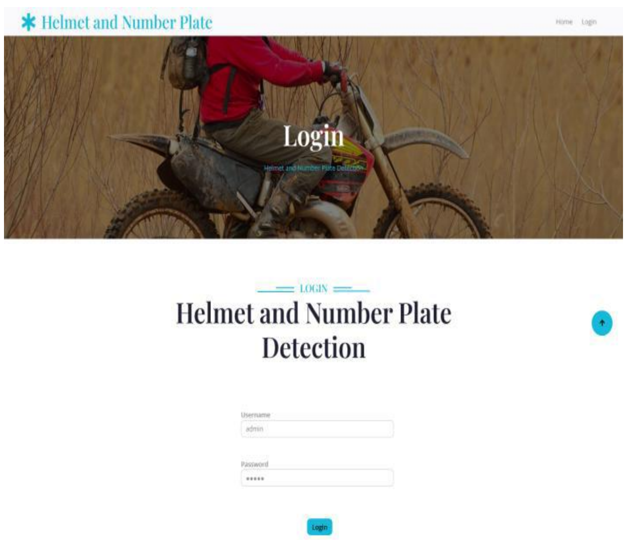
- Develop a prediction module that loads the saved YOLOv8 model and takes an input image or video stream. The module should be capable of processing the input data and outputting the detected 'With Helmet', 'Without Helmet', 'Number Plate' with bounding boxes, facilitating real-time detection.

Model Evaluation Module:

- This module evaluates the performance of the trained models using the testing dataset. It calculates accuracy metrics and other performance indicators to assess model effectiveness.
- Evaluate model accuracy, precision, recall, and mAP50.
- Generate confusion matrices for both models.
- Accuracy, precision, recall, and mAP50 are used to evaluate model performance.

VI. RESULTS





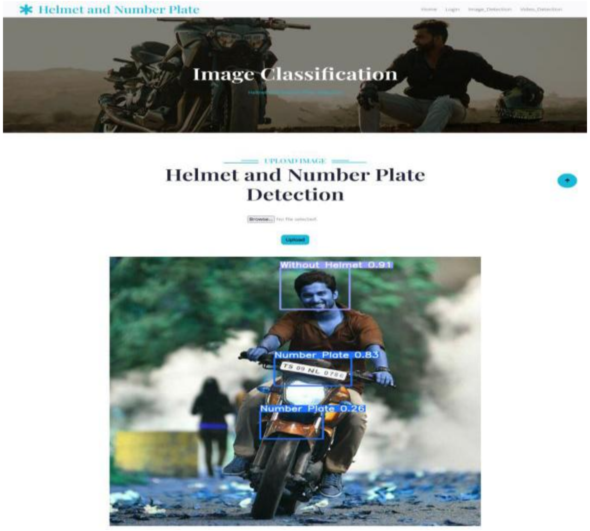
© 2025 IRJIET All Rights Reserved

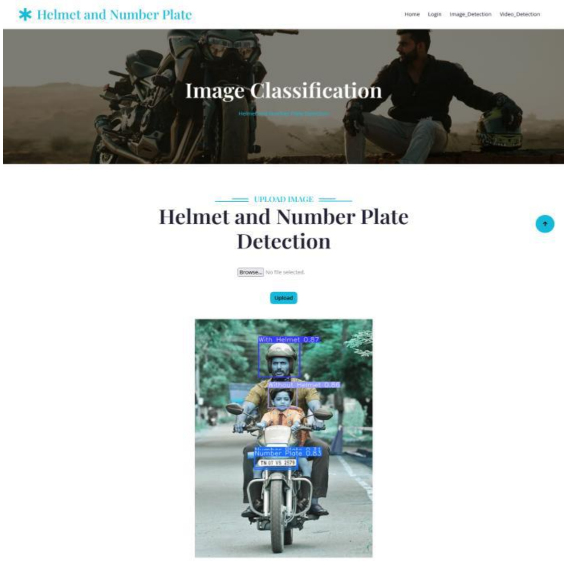
International Research Journal of Innovations in Engineering and Technology (IRJIET)

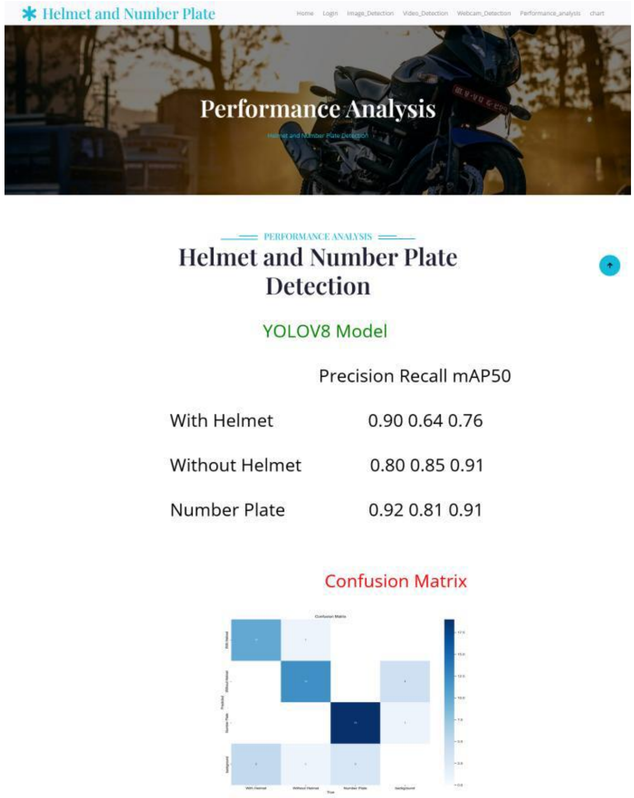
ISSN (online): 2581-3048

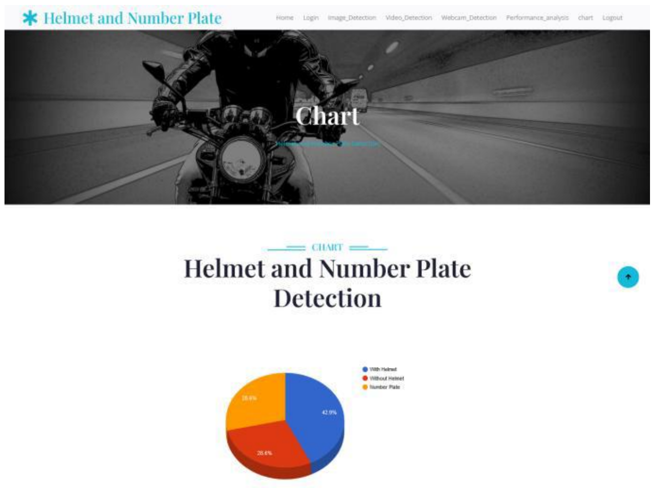
Volume 9, Issue 6, pp 119-127, June-2025

https://doi.org/10.47001/IRJIET/2025.906014









VII. CONCLUSION

The implementation of Smart Traffic Monitoring and Violation Detection with YOLOv8 Algorithm represents a significant advancement in the field of intelligent transportation systems and road safety monitoring. By leveraging the strengths of the YOLOv8 architecture, this system achieves high accuracy and real-time performance, making it an effective tool for enhancing compliance with safety regulations.

In summary, the YOLOv8 algorithm provides a powerful framework for developing intelligent detection systems that enhance road safety and traffic management. By automating

© 2025 IRJIET All Rights Reserved

ISSN (online): 2581-3048
Volume 9, Issue 6, pp 119-127, June-2025
https://doi.org/10.47001/IRJIET/2025.906014

the monitoring of helmet usage and vehicle compliance, this approach not only supports law enforcement efforts but also promotes a culture of safety among road users. Future work may involve further refining the model, expanding its capabilities, and integrating it with broader traffic management systems to maximize its impact on public safety.

REFERENCES

[1] M. Veldurthi and A. D. Dharmavaram, "Automatic Vehicle Identification and Recognition using CNN Implemented on PYNQ Board," 2022 6th International Conference on Electronics, Communication and Aerospace Technology, Coimbatore, India, 2022, pp. 1302-1306, doi: 10.1109/ICECA55336.2022.10009054.

[2] R. R. V. e. Silva, K. R. T. Aires and R. d. M. S. Veras, "Helmet Detection on Motorcyclists Using Image Descriptors and Classifiers," 2014 27th SIBGRAPI Conference on Graphics, Patterns and Images, Rio de Janeiro, 2014, pp. 141-148.

[3] P. Doungmala and K. Klubsuwan, "Helmet Wearing Detection in Thailand Using Haar Like Feature and Circle Hough Transform on Image Processing," 2016 IEEE International Conference on Computer and Information Technology (CIT), Nadi, 2016, pp. 611-614.

[4] Li, J., Liu, H., Wang, T., Jiang, M., Wang, S., Li, K., & Zhao, X. (2017, February). Safety helmet wearing detection based onimage processing and machine learning. In Advanced Computational Intelligence (ICACI), 2017 Ninth International Conference on (pp. 201-205). IEEE.

[5] K. Dahiya, D. Singh and C. K. Mohan, "Automatic detection of bike-riders without helmet using surveillance videos in realtime," 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, 2016, pp. 3046-3051.

[6] C. Vishnu, D. Singh, C. K. Mohan and S. Babu, "Detection of motorcyclists without helmet in videos using convolutional neural network," 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, 2017, pp. 3036-3041.

[7] J.Chiverton, "Helmet presence classification with motorcycle detection and tracking," in IET Intelligent Transport Systems, vol. 6, no. 3, pp. 259-269, September 2012.

[8] R. Silva, K. Aires, T. Santos, K. Abdala, R. Veras and A. Soares, "Automatic detection of motorcyclists without helmet," 2013 XXXIX Latin American Computing Conference (CLEI), Naiguata, 2013, pp. 1-7.

[9] R. R. V. e. Silva, K. R. T. Aires and R. d. M. S. Veras, "Helmet Detection on Motorcyclists Using Image Descriptors and Classifiers," 2014 27th SIBGRAPI Conference on Graphics, Patterns and Images, Rio de.

[10] P. Doungmala and K. Klubsuwan, "Helmet Wearing Detection in Thailand Using Haar Like Feature and Circle Hough Transform on Image Processing," 2016 IEEE International Conference on Computer and Information Technology (CIT), Nadi, 2016, pp. 611-614.

[11] K. Dahiya, D. Singh and C. K. Mohan, "Automatic detection of bike-riders without helmet using surveillance videos in real-time," 2016 International Joint Conference on Neural Networks (IJCNN)Detection of helmets on motorcyclists.

[12] Helmet Detection using Machine Learning and Automatic License Plate Recognition. By Lokesh Allamki, Manjunath Panchakshari, Ashish Sateesha, K S Pratheek (BNMIT, Bangalore).

[13] Detecting motorcycle helmet use with deep learning. By Felix Wilhelm Sieberta, Hanhe Lin (Department of Psychology and Ergonomics, Technische Universit¨at Berlin, Marchstraße 12, 10587 Berlin Germany).

[14] A Hybrid Approach for Helmet Detection for Riders Safety using Image Processing, Machine Learning, Artificial Intelligence. By M.Swapna, Tahniyath Wajeeh, Shaziya Jabeen.

[15] C. Vishnu, Dinesh Singh, C. Krishna Mohan and Sobhan Babu Visual Intelligence and Learning Group (VIGIL), Department of Computer Science and Engineering Indian Institute of Technology Hyderabad, Kandi, Sangareddy502285, India.

© 2025 IRJIET All Rights Reserved

ISSN (online): 2581-3048
Volume 9, Issue 6, pp 119-127, June-2025
https://doi.org/10.47001/IRJIET/2025.906014

Citation of this Article:

Ajeet Singh Bhadoriya, & Dr. Raghvendra Upadhyay. (2025). Smart Traffic Monitoring and Violation Detection with YOLOv8 Algorithm. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 9(6), 119-127. Article DOI https://doi.org/10.47001/IRJIET/2025.906014

****