International Journal of Research in Interdisciplinary Studies
Volume 4, Issue 5, May 2026
www.ijris.com | E-ISSN: 2584-1017

# Real Time Helmet Violation Detection Using Deep Learning

K. Elakia[1], R. G. Suresh Kumar[2*], M. Sree Haran[3], R. Vishva Moorthi[3], P. Dinesh Kumar[3], I. Mohamed Syed[3]
*[1]Assistant Professor, Department of Computer Science and Engineering, Rajiv Gandhi College of Engineering and Technology, Puducherry, India*
*[2]Professor & HoD, Department of Computer Science and Engineering, Rajiv Gandhi College of Engineering and Technology, Puducherry, India*
*[3]B.Tech. Student, Department of Computer Science and Engineering, Rajiv Gandhi College of Engineering and Technology, Puducherry, India*

Abstract—**Artificial Intelligence (AI) has emerged as a transformative technology for addressing real-world societal challenges by enabling machines to perform tasks that traditionally require human intelligence. Deep learning has gained significant importance in computer vision, supporting automatic recognition, classification, and analysis of visual data. These capabilities have been effectively utilized in traffic surveillance systems to enhance monitoring, decision-making, and law enforcement for improved road safety. Existing research proposes an AI-based multitier framework with a lightweight classifier for detecting helmetless motorbike riders. This two-stage approach initially identifies riders from surveillance footage and subsequently classifies helmet usage, achieving reduced computational complexity and faster processing with acceptable accuracy. However, such approaches are limited in scope, focusing primarily on helmet detection and lacking robustness under challenging environmental conditions. To address these limitations, the proposed system employs the YOLOv12 algorithm for comprehensive and real-time traffic violation detection. The system extends its capabilities to identify multiple violations, including triple riding and mobile phone usage, while maintaining performance under diverse conditions such as low light and adverse weather. Additionally, it incorporates an automated alert mechanism to notify authorities and warn riders. This integrated approach enhances detection accuracy, enables real-time enforcement, and contributes to the development of safer and more intelligent traffic management systems.**

*Index Terms*—**Artificial Intelligence, Deep Learning, Computer Vision, YOLOv12, Traffic Surveillance, Helmet Detection, Violation Detection, Road Safety.**

## 1. Introduction

Artificial Intelligence (AI) has become a powerful technology for solving real-world problems by enabling machines to perform tasks that typically require human intelligence.

Among its branches, deep learning has gained significant importance, especially in computer vision, where it enables automatic detection, classification, and analysis of images and videos.

These advancements have played a crucial role in improving traffic surveillance systems by allowing efficient monitoring of road activities and assisting authorities in maintaining public safety. With the rapid increase in vehicle numbers and traffic congestion, traditional manual monitoring methods have become inefficient, time-consuming, and prone to human error.

As a result, there is a growing demand for automated and intelligent traffic management solutions that can operate in real time. Computer vision–based techniques have been widely used for applications such as vehicle detection, license plate recognition, and helmet detection for motorbike riders, which is essential for reducing accident-related fatalities. Existing approaches often use multi-stage frameworks and lightweight models to achieve faster processing with reduced computational cost.

However, many systems are limited in scope and struggle to perform effectively under varying conditions such as low lighting, occlusion, and adverse weather. Therefore, more robust and scalable solutions are required to improve efficiency and ensure reliable traffic monitoring.

### *A. Architecture of Yolov12*

The YOLOv12 architecture is designed as a single-stage, end-to-end object detection framework optimized for real-time traffic violation detection. It consists of three main components: the backbone, neck, and detection head. The backbone extracts rich spatial features from input images using convolutional layers and residual connections, ensuring effective feature representation even in complex environments. The neck aggregates multi-scale features through feature pyramid structures, enhancing the model's ability to detect objects of varying sizes such as riders, helmets, and mobile phones.

Fig. 1. Architecture diagram of the YOLOv12

---

*Corresponding author: aargeek@gmail.com

The detection head performs bounding box prediction, object classification, and confidence scoring in a single pass, enabling fast and accurate detection. This streamlined architecture reduces computational overhead while maintaining high accuracy, making it suitable for real-time surveillance applications under diverse conditions.

### *B. Architecture Diagram*

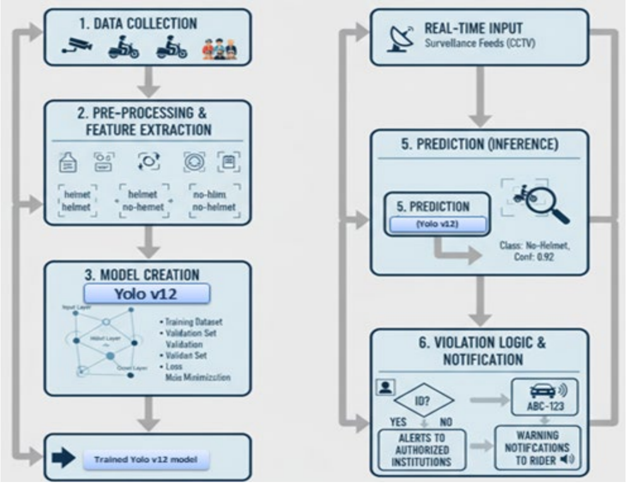

Fig. 2. Architecture diagram of the proposed system

The architecture diagram illustrates a structured pipeline for real-time traffic violation detection using the YOLOv12 model. The process begins with data collection, where traffic images and videos are gathered, followed by preprocessing and feature extraction to prepare labelled datasets such as helmet and non-helmet classes. The model creation phase involves training the YOLOv12 network using training, validation, and testing datasets to optimize detection performance.

In deployment, real-time input is obtained from surveillance systems such as CCTV cameras. The trained model performs inference to detect violations and classify objects with confidence scores. Finally, a violation logic module verifies detected offenses and triggers automated notifications, sending alerts to authorities and warnings to riders, ensuring efficient and intelligent traffic monitoring.

## 2. Literature Survey

*[1] Felix IL helm Siebert, Christoffer Riis, Kira, Oguzhan Gül, Frederik.* Bicycle helmets are a vital factor in reducing the severity of injuries during traffic accidents, making helmet usage an important metric in transport safety research. Traditionally, helmet-use data is collected through manual observation, which is expensive, time-consuming, and limited in scale. To address these challenges, the study proposes an automated computer vision–based approach for detecting bicycle helmet usage from traffic video footage. Video data was collected from two observation sites in Copenhagen, Denmark, and a dataset consisting of 4,000 cyclists was manually annotated to indicate helmet use. A state-of-the-art object detection model was trained to identify cyclists and classify whether they were wearing helmets. The developed model demonstrated strong performance in detecting active cyclists and estimating helmet usage. On the test dataset, helmet usage was only slightly underestimated by 0.52%, showing high accuracy and reliability.

*[2] Md. Ameer Raza, J. Bhavyasri Tanmaya Moukthika*Automated detection of motorcycle helmet usage from surveillance videos plays a critical role in strengthening road safety enforcement and awareness programs. However, many existing approaches face notable limitations, such as the inability to reliably track individual motorcycles across consecutive video frames and difficulty in distinguishing between drivers and passengers when evaluating helmet compliance. Additionally, earlier studies often depend on limited or homogeneous datasets, reducing robustness under varying traffic densities and real-world conditions. To overcome these challenges, the proposed study introduces a convolutional neural network (CNN)–based multi-task learning (MTL) framework that simultaneously performs motorcycle detection, tracking, rider identification, and helmet usage classification. By jointly learning visual similarity features for tracking and helmet classification within a single unified model, the system effectively links helmet status to specific riders over time.

*[3] Fatih Aslan and Yaşar Becerril.* Wearing personal protective equipment (PPE), especially helmets, is essential on construction sites to prevent severe head injuries and to comply with safety regulations. Helmets are particularly important because they are highly visible in surveillance footage, making them suitable for automated worker detection and monitoring.

However, manual inspection of helmet usage and worker identity is inefficient, labour-intensive, and impractical for large or busy construction sites. Both models achieved high precision and recall, with YOLOv5 showing slightly better performance for real-time symbol identification. The system was effective at identifying workers from distances of up to 10 meters. Additionally, a left-to-right, location-based symbol-ordering algorithm ensured consistent and reliable interpretation of symbols during inference.

*[4] Md. Ameer Raza, J. Bhavyasri Tanmaya Moukthika Dr. Varsha M, Niveditha MG, Priyadarshini P, Punya Shree B S, Sahana M. K.* Automated helmet detection systems play a crucial role in improving road safety in India, where two-wheelers are the most used mode of transportation. A significant number of road accidents and fatalities occur due to helmet non-compliance, even though traffic safety laws are clearly defined.

To overcome limitations in manual enforcement, automated helmet violation detection systems based on CCTV surveillance and machine learning have been developed. These systems use deep learning–based object detection techniques and are enhanced with Class Activation Mapping (CAM) to improve detection performance, particularly under low-light conditions. The proposed approach achieves an accuracy of 85.3%, making it suitable for large-scale deployment in urban traffic environments.

*[5] Sri Uthra V, Sariga Devi V, Vaishali K S, Padma Priya S. Two*-wheelers are among the most used modes of transportation, particularly in densely populated regions, but

they are also associated with a high rate of road accidents. A major contributor to fatal injuries is helmet non-compliance, as head injuries account for a large proportion of two-wheeler deaths. To address this safety concern, the proposed system introduces an integrated and automated framework for detecting helmet rule violations and identifying offenders, thereby reducing reliance on manual traffic monitoring. The system combines multiple computer vision and machine learning techniques into a single pipeline. First, motorcycles are detected and classified from traffic surveillance footage using a Support Vector Machine (SVM)–based vehicle classification approach. Helmet detection is then performed using Convolutional Neural Networks (CNNs) to extract visual features from rider images, followed by an SVM classifier to determine the presence or absence of a helmet.

## 3. Proposed System

The proposed system overcomes the limitations of existing helmet violation detection methods by adopting a fully automated deep learning–based framework built on YOLOv12, which offers improved accuracy, faster inference, and better generalization across real-world traffic conditions. YOLOv12 is utilized to precisely detect helmet usage by learning complex visual features, enabling reliable differentiation between helmeted and non-helmeted riders even in challenging scenarios such as varying lighting conditions, different helmet styles, dense traffic, and diverse camera angles. The system is trained using robust and diverse datasets to ensure consistent performance in both urban roads and highways. Upload video processed in real time, allowing instant identification of helmet violations without the need for continuous human supervision. When a rider without a helmet is detected, the system automatically triggers alert mechanisms by notifying authorized institutions such as traffic police departments and issuing warning notifications to riders. This immediate response supports proactive enforcement and increases rider awareness, encouraging compliance with helmet regulations. The automated workflow significantly reduces human error, monitoring effort, and enforcement delays, making the system suitable for large-scale and continuous deployment. By combining high-accuracy detection, real-time processing, and automated alert generation, the proposed system ensures efficient enforcement and practical implementation.

### *A. Data Collection*

Data collection is the foundational step in developing an effective deep learning model for traffic violation detection. In this system, datasets are obtained from open-source platforms such as Kaggle, which provides a wide variety of annotated image and video datasets related to traffic surveillance. These datasets typically include images of motorcyclists, vehicles, and road scenes under different environmental conditions. For this project, relevant datasets are selected that contain instances of helmet usage and helmetless riders. The availability of labelled data, such as bounding box annotations and class labels, is crucial for supervised learning tasks like object detection. Kaggle datasets often come in formats like COCO or Pascal VOC, which are compatible with modern object detection frameworks. Additionally, data diversity is ensured by including images captured under different lighting conditions, weather scenarios (rain, fog, night), and camera angles to improve the model's generalization ability. In some cases, multiple datasets may be combined to create a more comprehensive training set. Data augmentation techniques such as flipping, rotation, and scaling can also be applied at this stage to artificially increase dataset size and variability. Proper data collection ensures that the model learns from real-world scenarios, ultimately improving its robustness and accuracy in detecting helmet violations in practical environments**.**

### *B. Pre-Processing*

Pre-processing is a critical step that prepares the collected data for efficient and accurate model training. Raw datasets obtained from Kaggle may contain inconsistencies such as varying image sizes, noise, irrelevant objects, or incomplete annotations. Therefore, preprocessing involves cleaning and standardizing the data to ensure uniformity. First, images are resized to a fixed resolution compatible with the YOLOv12 model, enabling faster computation and consistent input dimensions. Noise reduction techniques and normalization are applied to enhance image quality and stabilize pixel value distributions. Annotation files are also verified and converted into the required format (such as YOLO format), where bounding box coordinates are normalized relative to image dimensions. Data augmentation techniques, including rotation, flipping, brightness adjustment, and cropping, are applied to improve model robustness and prevent overfitting. These transformations help the model learn variations in object orientation, lighting, and scale. Additionally, dataset splitting is performed during preprocessing to divide the data into training, validation, and testing sets. This ensures unbiased model evaluation and prevents data leakage.

Proper preprocessing significantly enhances the learning capability of the model by reducing noise, improving data consistency, and increasing variability. As a result, it plays a crucial role in improving the overall performance, accuracy, and generalization of the YOLOv12-based helmet detection system.

### *C. Feature Extraction*

Feature extraction is the process of identifying and representing important visual patterns from input images that help in detecting and classifying objects. In the YOLOv12 framework, feature extraction is performed automatically using a deep convolutional neural network (CNN) backbone. The backbone processes input images through multiple convolutional layers, each designed to capture specific levels of information. Early layers focus on low-level features such as edges, corners, and textures, while deeper layers capture high-level semantic features like object shapes and contextual relationships. This hierarchical feature learning enables the model to understand complex patterns required for detecting helmet usage.

YOLOv12 enhances feature extraction by using an optimized backbone that balances computational efficiency and accuracy. Additionally, feature maps generated at different stages are passed to the neck (such as FPN or PAN), which combines multi-scale features to improve detection of both small and large objects. This is particularly important in traffic scenarios where helmets may appear small or partially occluded. Effective feature extraction ensures that the model can distinguish between helmet and no-helmet classes. By learning meaningful and discriminative features, the system achieves improved detection accuracy and robustness in diverse real-world conditions.

### *D. Model Creation Using YOLOv12*

Model creation involves designing, configuring, and training the YOLOv12 object detection model for helmet violation detection. The YOLOv12 model is initialized with a predefined architecture consisting of backbone, neck, and detection head components. During this stage, custom classes such as "helmet" and "no helmet" are defined based on the problem requirements. The pre-processed dataset is then fed into the model for training. Transfer learning can be applied by initializing the model with pretrained weights, which helps improve performance and reduce training time. The training process involves optimizing model parameters using techniques such as stochastic gradient descent (SGD) or Adam optimizer, along with appropriate loss functions that account for bounding box regression, classification, and object confidence. Hyperparameters such as learning rate, batch size, and number of epochs are carefully selected to ensure stable and efficient training. During training, the model learns to detect riders and classify helmet usage accurately by minimizing prediction errors. Validation is performed periodically to monitor performance and avoid overfitting. The lightweight nature of YOLOv12 allows for faster training and real-time inference capabilities.

### *E. Test Data*

Test data is used to evaluate the performance and generalization capability of the trained YOLOv12 model. It consists of a separate subset of the dataset that is not used during training or validation, ensuring an unbiased assessment of the model's effectiveness. The test dataset includes images representing real-world traffic scenarios with varying conditions such as different lighting, weather, and traffic density. Proper selection of test data is crucial, as it should reflect the diversity and complexity of practical environments. During testing, the trained model processes each image and generates predictions, including bounding boxes, class labels, and confidence scores. These predictions are then compared with ground truth annotations to evaluate performance metrics such as precision, recall, F1-score, and mean Average Precision (map). High precision indicates fewer false positives, while high recall indicates effective detection of actual violations.

Testing also helps identify limitations of the model, such as difficulty in detecting small objects or performance degradation under poor lighting conditions. Visualization of detection results is often performed to qualitatively assess model behaviour. By analysing test results, further improvements can be made through hyperparameter tuning, data augmentation, or model optimization.

### *F. Prediction*

Prediction is the final stage where the trained YOLOv12 model is deployed to detect helmet violations in real-time or from stored images and videos. In this phase, input data such as live camera feeds or recorded footage is passed through the trained model. The model processes each frame and outputs predictions in the form of bounding boxes, class labels, and confidence scores. These predictions indicate the presence of riders and whether they are wearing helmets or not. YOLOv12 performs prediction in a single forward pass, ensuring high-speed processing suitable for real-time applications. Non-Maximum Suppression (NMS) is applied to eliminate duplicate detections and retain the most accurate bounding boxes. The system can also be integrated with alert mechanisms that trigger notifications or warnings when violations are detected. For example, alerts can be sent to traffic authorities or displayed as caution messages. Visualization techniques such as overlaying bounding boxes on video frames help in understanding detection results. The prediction stage demonstrates the practical utility of the system, enabling automated monitoring and enforcement of helmet safety rules.

## 4. Result and Discussion

The results and discussion of the proposed YOLOv12-based helmet detection system demonstrate its effectiveness in accurately identifying helmet usage in real-world traffic scenarios. The model was evaluated using a separate test dataset consisting of diverse images captured under varying environmental conditions, including different lighting levels, weather situations, and traffic densities. The system achieved high performance in terms of precision, recall, and mean Average Precision (map), indicating its ability to correctly detect riders wearing helmets as well as those without helmets while minimizing false detections. The use of a deep learning-based single-stage detector like YOLOv12 enabled fast processing speeds, making the system suitable for real-time applications. Experimental results show that the model performs well even when helmets are partially occluded or when riders appear at different distances and angles. Compared to traditional methods and earlier lightweight models, the proposed approach offers improved detection accuracy and better generalization due to enhanced feature extraction and multi-scale detection capabilities. However, certain challenges were observed during testing, such as slight performance degradation in extremely low-light conditions or when image quality is poor. Despite these limitations, the system maintains consistent and reliable performance across most scenarios. The integration of real-time prediction and alert mechanisms further enhances its practical usability for traffic monitoring and enforcement.

### *A. Input Layer*

The Input Layer in the proposed YOLOv12-based helmet detection system plays a crucial role in preparing traffic surveillance images for efficient processing and accurate detection. In this system, input data is obtained from CCTV cameras or Kaggle datasets containing images of motorbike riders. These images are first resized to a fixed resolution (such as 640×640) to ensure uniformity and compatibility with the YOLOv12 architecture. This resizing helps maintain consistent spatial dimensions, enabling faster batch processing and real-time performance. After resizing, the pixel values of the images are normalized to a range between 0 and 1 to stabilize training and improve convergence of the model. The normalization process can be represented as:

$$x' = \frac{x}{255}$$

Where, x is the original pixel value and x′ is the normalized value. The input image is then structured into a tensor of dimensions H, times, times, where H and W represent height and width, and C=3 corresponds to RGB channels. Furthermore, the image is conceptually divided into a grid of size S, where each grid cell is responsible for detecting objects, whose centres fall within that region. This grid-based representation allows the model to perform detection in a single forward pass. In the context of the proposed system, the input layer ensures that diverse traffic images—captured under different lighting and environmental conditions—are standardized and efficiently processed, thereby enabling accurate detection of helmet usage in real-time traffic monitoring applications.

### *B. Convolutional Layers*

The Convolutional Layers in the proposed YOLOv12-based helmet detection system are the core components responsible for extracting meaningful features from input images. After preprocessing in the input layer, the image is passed through multiple convolutional layers, where filters (kernels) slide over the image to detect important visual patterns such as edges, textures, shapes, and object boundaries. In early layers, simple features like edges and corners are captured, while deeper layers learn complex patterns such as the structure of helmets and rider shapes. Each convolutional layer applies multiple filters to generate feature maps, which highlight different aspects of the image. These layers are typically followed by activation functions (such as REL or Shilu) and batch normalization to improve learning efficiency and stability. The convolution operation can be mathematically expressed as:

$$Y(i,j) = \sum_m \sum_n X(i+m, j+n) \cdot K(m,n)$$

where X represents the input image, K is the convolution kernel (filter), and Y(i,j) is the output feature map at position (i,j). This operation enables the model to focus on relevant spatial features while reducing noise and redundant information. In the proposed system, convolutional layers play a vital role in distinguishing between helmet and no-helmet classes by learning discriminative features from traffic images. Their ability to automatically extract hierarchical features significantly improves detection accuracy and robustness, making them essential for real-time helmet detection in diverse environmental conditions.

### *C. Batch Normalization Layers*

The Batch Normalization Layers in the proposed YOLOv12-based helmet detection system play a crucial role in improving training stability, convergence speed, and overall model performance. During deep neural network training, the distribution of inputs to each layer can change as the model updates its weights, a phenomenon known as internal covariate shift. Batch normalization addresses this issue by normalizing the activations of each layer for every mini-batch, ensuring that the data fed into subsequent layers maintains a consistent distribution. This normalization allows the model to train faster and reduces sensitivity to initialization and learning rates. In the context of the proposed system, batch normalization is applied after convolutional layers to stabilize feature extraction and improve the detection of helmets under varying lighting and environmental conditions. It also acts as a regularize, reducing the need for additional techniques like dropout and helping prevent overfitting. The batch normalization process can be mathematically expressed as:

$$\hat{x} = \frac{x-\mu}{\sqrt{\sigma^2+\epsilon}}$$

Where, x is the input activation, $\mu$ is the mean of the batch, $\sigma^2$ is the variance, and $\epsilon$ is a small constant added for numerical stability. The normalized output x^ is then scaled and shifted using learnable parameters. By maintaining stable input distributions across layers, batch normalization enhances the robustness and efficiency of the YOLOv12 model, leading to improved accuracy in helmet detection and better generalization to real-world traffic scenarios.

### *D. Accuracy*

Accuracy is a fundamental evaluation metric used to measure the overall performance of the proposed YOLOv12-based helmet detection system. It represents the proportion of correctly predicted instances out of the total number of predictions made by the model. In this system, accuracy reflects how effectively the model can classify riders as wearing a helmet or not wearing a helmet. A high accuracy indicates that the model can correctly identify most of the instances, thereby demonstrating its reliability in real-world traffic monitoring scenarios. However, accuracy alone may not always provide a complete picture, especially in cases where the dataset is imbalanced, such as having more helmet-wearing instances than non-helmet cases. Despite this limitation, it remains an important baseline metric for evaluating model performance during testing. The mathematical representation of accuracy is given as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

Where, TP (True Positives) represents correctly detected helmet cases, TN (True Negatives) represents correctly detected no-helmet cases, FP (False Positives) represents incorrect detections where the model predicts a helmet but none is present, and FN (False Negatives) represents missed detections where the model fails to identify the absence of a helmet. In the proposed system, achieving high accuracy ensures that the model can reliably distinguish between helmet and no-helmet conditions. This contributes to effective enforcement of safety regulations and enhances the practical deployment of the system in intelligent traffic management applications.

### *E. Loss*

The loss function in the proposed YOLOv12-based helmet detection system is a critical component used to measure the difference between the model's predictions and the actual ground truth values during training. It guides the learning process by indicating how well the model is performing and how its parameters should be updated to improve accuracy. In object detection tasks like helmet detection, the loss function is typically a combination of multiple components, including classification loss (for identifying helmet or no helmet), localization loss (for predicting accurate bounding box coordinates), and confidence loss (for determining whether an object is present). By minimizing the total loss, the model learns to make more precise predictions over time. A commonly used representation of total loss in YOLO-based models is:

$$Loss = L_{cls} + L_{loc} + L_{conf}$$

Where, LLC's is the classification loss, Looc is the localization (bounding box regression) loss, and Lacon is the confidence loss. During training, optimization algorithms such as stochastic gradient descent or Adam adjust the model weights to reduce this loss value iteratively. In the proposed system, an effective loss function ensures accurate detection of helmets by improving both object classification and localization. As the loss decreases over epochs, the model becomes more reliable and capable of handling real-world traffic conditions efficiently.

### *F. Precision*

Precision is an important performance metric used to evaluate the effectiveness of the proposed YOLOv12-based helmet detection system, particularly in terms of how accurate the positive predictions are. It measures the proportion of correctly predicted positive instances (helmet or no-helmet detections) out of all instances that the model has predicted as positive. In the context of this system, precision indicates how many of the detected violations (such as identifying a rider without a helmet) are correct. A high precision value means that the model produces very few false alarms, which is crucial in real-world traffic monitoring systems to avoid unnecessary penalties or incorrect alerts. Precision becomes especially important when the cost of false positives is high, such as wrongly identifying a rider as violating safety rules. The mathematical formula for precision is given as:

$$Precision = \frac{TP}{TP+FP}$$

Where, TP (True Positives) represents the number of correctly detected positive cases (e.g., correctly identifying a rider without a helmet), and FP (False Positives) represents the number of incorrect positive predictions (e.g., identifying a helmet when none exists or vice versa). In the proposed YOLOv12 system, achieving high precision ensures that most of the detected helmet violations are accurate and reliable. This improves the credibility of the system when deployed in real-time traffic environments. Although precision focuses on correctness of positive predictions, it is often analysed along with recall to provide a complete understanding of model performance.

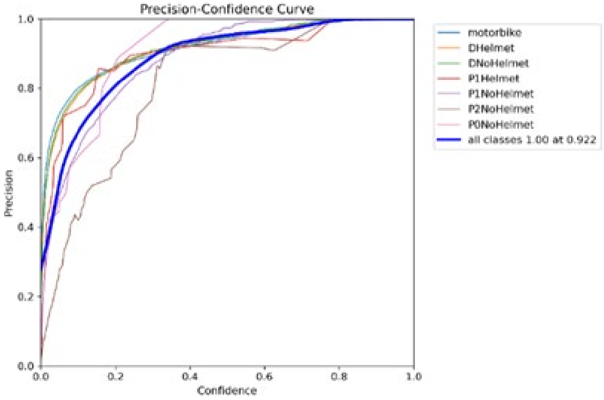

Fig. 3.

A precision graph in the proposed YOLOv12-based helmet detection system visually represents how precision varies with different confidence thresholds or detection conditions. Typically, precision is plotted on the y-axis, while the confidence threshold or recall is plotted on the x-axis. As the confidence threshold increases, the model becomes more selective, resulting in fewer false positives and higher precision. However, this may reduce the number of detected instances. The graph helps in identifying the optimal threshold where the model achieves a good balance between accuracy and detection coverage. In the proposed system, the precision graph demonstrates that the model maintains high precision across most thresholds, indicating reliable and consistent helmet detection performance in real-world scenarios.

### *G. Recall*

Recall is an important evaluation metric used in the proposed YOLOv12-based helmet detection system to measure the model's ability to correctly identify all actual positive cases. It represents the proportion of true positive detections out of all real positive instances present in the dataset. In this system, recall indicates how effectively the model detects riders without helmets without missing any violations. A high recall value means the model successfully captures most of the actual helmet violations, reducing the chances of missed detections. The mathematical formula for recall is given as:

$$Recall = \frac{TP}{TP+FN}$$

Where, TP represents correctly detected positive cases and FN represents missed detections.

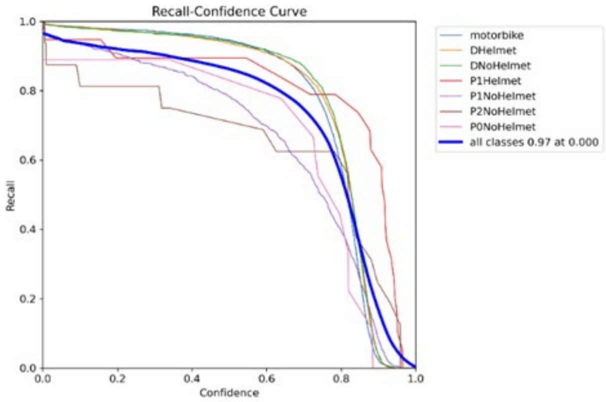

Fig. 4.

A recall graph in the proposed YOLOv12-based helmet detection system illustrates how effectively the model detects all actual positive instances (helmet or no-helmet cases) across different confidence thresholds. In this graph, recall is plotted on the y-axis, while the confidence threshold or sometimes precision is plotted on the x-axis. As the confidence threshold decreases, the model becomes less strict in its predictions, resulting in more detections and higher recall. This means the system can capture most of the actual helmet violations, reducing missed detections (false negatives). However, increasing recall may also introduce more false positives. Conversely, at higher confidence thresholds, recall tends to decrease because the model only considers highly confident predictions, potentially missing some true cases. The recall graph helps in analysing how well the system balances detection completeness under varying conditions. In the proposed system, the recall graph typically shows high recall values across moderate thresholds, indicating that the model successfully identifies most riders without helmets even in complex traffic scenarios. This is particularly important for safety enforcement, as missing violations can reduce system effectiveness.

### *H. F1 Score*

The F1-score is an important evaluation metric used in the proposed YOLOv12-based helmet detection system to provide a balanced measure of both precision and recall. It is especially useful when there is a need to consider both false positives and false negatives together. The F1-score represents the harmonic mean of precision and recall, ensuring that both metrics are equally weighted. A high F1-score indicates that the model performs well in correctly detecting helmet violations while also minimizing missed detections. The mathematical formula for F1-score is given as:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

In the proposed system, a high F1-score reflects overall strong detection performance.

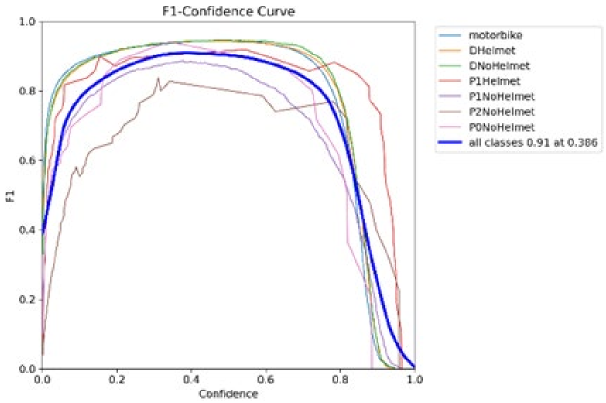

Fig. 5.

An F1-score graph in the proposed YOLOv12-based helmet detection system illustrates the balance between precision and recall across different confidence thresholds. In this graph, the F1-score is plotted on the y-axis, while the confidence threshold is shown on the x-axis. The curve typically rises to a peak at an optimal threshold where both precision and recall are well balanced, resulting in the highest F1-score. At very low thresholds, recall may be high but precision drops, reducing the F1-score. Conversely, at very high thresholds, precision increases but recall decreases, again lowering the F1-score.

### *1) Comparison Graph*

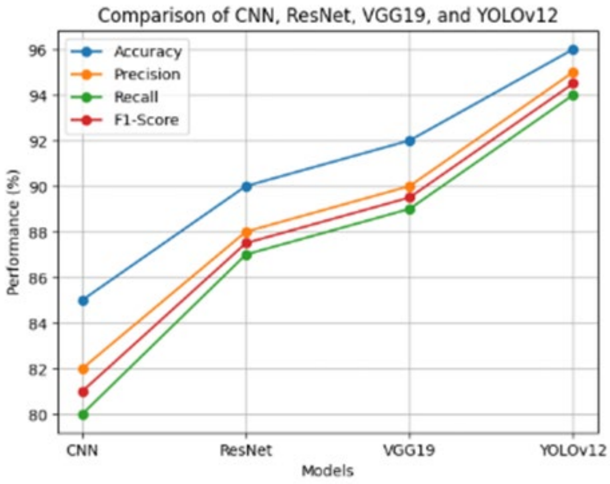

Fig. 6.

The comparison graph illustrates the performance differences between CNN, Reset, VGG19, and YOLOv12 based on key evaluation metrics such as accuracy, precision, recall, and F1-score. In the graph, each model is represented along the x-axis, while performance values are shown on the y-axis. Traditional CNN models demonstrate moderate performance, as they are effective in basic feature extraction but lack depth for complex pattern recognition.

Reset shows improved results due to its deep architecture and residual connections, which help in better feature learning and reduced vanishing gradient problems. VGG19 further enhances performance with deeper layers and uniform architecture, leading to higher accuracy and better classification capability. However, YOLOv12 outperforms all other models across all metrics, as it is specifically designed for real-time object detection with optimized feature extraction and multi-scale detection capabilities. The graph clearly indicates that YOLOv12 achieves the highest values, demonstrating superior efficiency, accuracy, and robustness for helmet detection in traffic surveillance systems.

Table 1
Comparison table

| **Model** | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F1-Score (%)** |
|---|---|---|---|---|
| CNN | 85 | 82 | 80 | 81 |
| Reset | 90 | 88 | 87 | 87.5 |
| VGG19 | 92 | 90 | 89 | 89.5 |
| YOLOv12 | 96 | 95 | 94 | 94.5 |

## 5. Conclusion

In conclusion, the proposed system demonstrates the significant potential of Artificial Intelligence and deep learning techniques in enhancing traffic surveillance and road safety. By leveraging the capabilities of computer vision, the system effectively automates the detection and analysis of traffic violations from real-time visual data. While existing approaches focused mainly on helmetless rider detection using lightweight models, they were limited in scope and did not address broader challenges such as multiple violations and varying environmental conditions. To overcome these limitations, the proposed system utilizes the YOLOv12 algorithm, which provides efficient and accurate real-time object detection. The system not only identifies helmet violations but also extends its functionality to detect additional violations such as triple riding and mobile phone usage, thereby offering a more comprehensive solution. Moreover, its robustness under challenging conditions like low light and adverse weather enhances its practical applicability in real-world scenarios. The integration of an automated alert mechanism further strengthens the system by enabling timely notifications to authorities and warning messages to riders, promoting proactive enforcement of traffic rules. Future enhancements of the proposed YOLOv12-based helmet detection system can focus on improving scalability, accuracy, and real-world integration. The system can be extended to support additional traffic violations such as lane discipline, signal jumping, and vehicle overloading for a more comprehensive solution. Incorporating advanced techniques like attention mechanisms and transformer-based models can further enhance detection accuracy under complex conditions. Integration with IoT devices and smart city infrastructure can enable automated fine generation and centralized monitoring.

## References

[1] İ. Kılıç and G. Aydin, "Turkish vehicle license plate recognition using deep learning," in *Proc. Int. Conf. Artificial Intelligence and Data Processing (IDAP)*, 2018, pp. 1–5.

[2] V. Gnanaprakash, "Automatic number plate recognition using deep learning," *IOP Conference Series: Materials Science and Engineering*, vol. 1084, Art. no. 012027, 2021.

[3] R. Shashidhar, A. S. Manjunath, R. Santhosh Kumar, M. Roopa, and S. B. Puneeth, "Vehicle number plate detection and recognition using YOLO-V3 and OCR method," in *Proc. IEEE Int. Conf. Mobile Networks and Wireless Communications (ICMNWC)*, 2021, pp. 1–5.

[4] D. R. Vedalias, R. Sudhan, G. Saranya, M. Safa, and D. Arun, "Comparative analysis of Easier and Tesseract for automatic license plate recognition using deep learning algorithm," in *Proc. 6th Int. Conf. Electronics, Communication and Aerospace Technology (ICECA)*, 2022, pp. 966–971.

[5] M. Al-Mheiri, O. Kais, and T. Bonny, "Car plate recognition using machine learning," in *Proc. Advances in Science and Engineering Technology International Conferences (ASET)*, 2022, pp. 1–6.

[6] N. Saleem, H. Muazzam, H. M. Tahir, and U. Farooq, "Automatic license plate recognition using extracted features," in *Proc. 4th Int. Symp. Computational and Business Intelligence*, 2016, pp. 221–225.

[7] R. Islam, K. F. Sharif, and S. Biswas, "Automatic vehicle number plate recognition using structured elements," in *Proc. IEEE Int. Conf. Systems, Process and Control*, 2015, pp. 44–48.

[8] P. Prabhakar, P. Anupama, and S. R. Resmi, "Automatic vehicle number plate detection and recognition," in *Proc. IEEE Int. Conf. Control, Instrumentation, Communication and Computational Technologies*, 2014, pp. 185–190.

[9] J. Chong, C. Tianshui, and J. Linhao, "License plate recognition based on edge detection algorithm," in *Proc. 9th IEEE Int. Conf. Intelligent Information Hiding and Multimedia Signal Processing*, 2013, pp. 395–398.

[10] K. M. Hung and C. T. Hsieh, "A real-time mobile vehicle license plate detection and recognition," *Tamkang Journal of Science and Engineering*, vol. 13, no. 4, pp. 433–442, 2010.

[11] A. Puranic, K. T. Deepak, and V. Umadevi, "Vehicle number plate recognition system: A literature review and implementation using template matching," *International Journal of Computer Applications*, vol. 134, no. 1, pp. 12–16, 2016.

[12] P. Sai Krishna, "Automatic number plate recognition by using MATLAB," *International Journal of Innovative Research in Electronics and Communications*, vol. 2, no. 4, pp. 1–7, 2015.

[13] C. Arth, C. Leistner, and H. Bischof, "Tricam: An embedded platform for remote traffic surveillance," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*.

[14] N. K. Bansal and R. Gupta, "Real-time helmet detection using convolutional neural networks."

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2012, pp. 1097–1105.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[17] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[18] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[19] G. Jocher *et al.*, "YOLOv5," 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[20] Ultralytics Team, "YOLOv8: Next generation object detection model," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520.

[22] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Machine Learning (ICML)*, 2019, pp. 6105–6114.

[23] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2015.