Volume 11, Issue 3, March – 2026 International Journal of Innovative Science and Research Technology
ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26mar1074

Real-Time Helmet Detection: A Comparative Study of Two-Stage and One-Stage Object Detection Frameworks

Anil Kumar Reddy Tetali[1]; Tatavarthi Rishi Varun[2]; Guttula Soma Durga Arjun[3]; A. V. N. B. Pavan Kumar[4]; Karri Jaya Naga Sri[5]; T. J. V. Adinarayana[6]

[1]Assistant Professor, Department of CSE, Sri Vasavi Engineering College (A), Tadepalligudem, A. P, India.
[2,3,4,5,6]B. Tech Students, Department of CSE, Sri Vasavi Engineering College (A), Tadepalligudem, A. P, India

Publication Date: 2026/03/23

Abstract: Although the automated detection of helmets remains a cornerstone of modern occupational safety and transit safety, the continued prevalence of traumatic head injuries indicates a systemic failure in regulatory compliance. The inherently fragmented nature of manual detection would inevitably collapse under the logistical burden of a large industrial facility or the high-speed traffic flows of urban landscapes. In such scenarios, a paradigm shift from the limitations of human detection to the high-fidelity architectures of autonomous computer vision systems is a necessity, providing a 24-hour state of watchfulness.

The efficacy of the YOLO framework, a paradigm-shifting single- stage detector, in the detection of protective headgear in a dynamic environment is the subject of the following investigation. To understand the performance of the YOLO framework, it is essential to understand the architectural divide between multi- stage detection systems and single-stage detection systems. While the traditional two-stage detection systems, such as the Faster R- CNN and Regional Proposal Networks, focus on the accuracy of localization at the cost of considerable computational latency, the single-stage detection systems such as the SSD, RetinaNet, and the YOLO family focus on a unified regression problem.

In the weighing of the competing demands of mean Average Precision (mAP) performance and real-time throughput, the choice falls squarely on YOLO as the preeminent choice for safety systems on the edge. The need to react instantaneously to the situation on the road precludes the use of computationally intensive approaches, which would necessarily sacrifice accuracy on the altar of speed. The ability of YOLO to deliver high frame rates without sacrificing accuracy on a catastrophic scale provides the fluidity of action to proactively mitigate safety hazards.

Keywords: *Helmet Detection, Intelligent Transportation Systems, Deep Learning, YOLOv8, Object Detection, Traffic Surveillance, Computer Vision.*

How to Cite: Anil Kumar Reddy Tetali; Tatavarthi Rishi Varun; Guttula Soma Durga Arjun; A. V. N. B. Pavan Kumar; Karri Jaya Naga Sri; T. J. V. Adinarayana (2026) Real-Time Helmet Detection: A Comparative Study of Two-Stage and One-Stage Object Detection Frameworks. *International Journal of Innovative Science and Research Technology,*11(3), 1785-1791. https://doi.org/10.38124/ijisrt/26mar1074

I. INTRODUCTION

Social and industrial safety, whether in the work environment or on the road, is one of the biggest global issues that needs to be addressed, particularly in environments and conditions in which the usage of protective gear such as helmets is critical in preventing head injuries. The compliance with safety is often compromised due to human factors such as negligence and lack of supervision. The process of monitoring helmet usage in a larger environment or in a crowded environment, such as construction sites and the road, is difficult to manage on a manual basis.

The emerging solutions in the new world of smart automation and intelligent systems offer promising avenues for automated safety monitoring solutions. These solutions include computer vision and deep learning.

Among these solutions, object detection is a basic computer vision problem that allows computers to detect and localize objects in an image or video. There are many detection algorithm architectures, and each of these architectures offers trade-offs in terms of precision, speed, and computational complexity.

The detection methods that employ two-stage detection, i.e., detection of candidates followed by the classification of the candidates, are more accurate but have lower inference rates. On the other hand, one-stage detectors are more straightforward, where the location and class of objects are predicted simultaneously.

YOLO (You Only Look Once) is a leading model of one-stage detectors. This model has revolutionized real-time object detection. The latest version of YOLO is YOLOv8. This model has employed advanced techniques such as multi-scale feature learning, an improved backbone design, and training methods. This has enabled the model to become more efficient in detecting small or over- lapping objects such as helmets.

The researchers used YOLOv8 to detect helmets. This is to demonstrate how to apply this model in practical situations. The evaluation of existing two-stage and one-stage object detection methods is a strong basis for selecting an efficient algorithm.

The use of next-generation technologies such as deep learning techniques to automate helmet detection is a significant contribution to enhancing safety, reducing accidents, and saving lives. Figure 1 illustrates a vis- ual representation of the object detection models and the proposed model.

II. RELATED WORK

Object detection has undergonemajor transformation with the advancement of deep learning, es- pecially through ConvolutionalNeural Networks (CNNs). Earlier, traditional computer vision methods relied heavily on hand-crafted feature extractors such as Histogram of Oriented Gradients (HOG), Scale- Invariant FeatureTransform (SIFT), and sliding-window based classifiers. Alt- hough these approaches workedreasonably well for constrained environments, they suffered from major limitations includingsensitivity to scale variations, poor generalization.

The transition to CNN-basedframeworks marked a significant leap in performance. Modern object detectors leverage data- drivenfeature extraction, hierarchical representation learning, and end-to- end pipelines, enabling highlyaccurate and scalable detection across diverse environments such as autonomous driving, surveillance, and industrial monitoring.

R-CNN (2014)

Fast R-CNN (2015) Fast R-CNN addressed the computational bottlenecks by sharing convolutional features across all proposals through ROI pooling. This significantly improved speed and reduced training complexity. However, the system still relied on external region proposal mechanisms such as selective search, limiting its real- time applicability.

Faster R-CNN (2015)

➢ *Two-StageDetectors*

- *Two-Stage Detectors Break the Detection Task into:*

(1) proposing candidate regions, and (2) classifying and refining those regions. This approach generally achieves high precision but is computationally heavy, making it less suit- able for real-time applications like traffic monitoring.

FasterR-CNN introduced a Region Proposal Network (RPN) that enabled the entire system—proposal generation and object classification—to be trained jointly in an end-to-end manner. This resulted in higher accuracy and efficiency. Despite these advancements, the two-stage design re- mains relatively slow for real- time traffic scenarios, particularly when dealing with fast- moving vehicles or dense traffic environments.

➢ *One-Stage Detectors*

One-stage detectors remove the region proposal step and directly predict bounding boxes and class probabilities in a single forward pass. This design improves computational efficiency, making these models suitable for real-time applications such as helmet detection and vehicle monitoring.

- *YOLO Family:*

The YOLO (You Only Look Once) series is known for its balance between speed and accuracy.

YOLO (2016) introduced the concept of treating object detection as a regression problem, enabling real-time prediction at high frame rates. Early versions struggled with small objects and complex backgrounds.

R-CNN introduced the concept of using deep CNNs for object detection. Region proposals were generated using selective search, and each proposed region was passed individually through a CNN. Though this method achieved high accuracy compared to earlier approaches, it was extremely slow due to redundant feature extraction for each region.

YOLOv2, YOLOv3, YOLOv4 added improvements such as batch normalization, multi-scale training, anchor boxes, CSP Darknet backbone, and spatial pyramid pooling. These upgrades significantly boosted accuracy while maintaining speed.

YOLOv5 and YOLOv8 introduced lightweight architectures, enhanced loss functions, dynamic label assignment, mosaic augmentation, and robust training pipelines.

These versions are optimized for deployment on edge devices like Jetson Nano, Raspberry Pi, and embedded GPU platforms, making them highly suitable for surveillance and intelligent transportation systems.

- SSD (Single Shot Detector, 2016) generates predictions from multiple feature maps of different scales, enabling detection of both small and large objects. Although fast, its accuracy is slightly lower than modern YOLO variants, especially for small objects such as helmets.

- RetinaNet (2017) introduced focal loss to solve the class imbalance problem between foreground and background samples. This innovation made one-stage detectors more competitive with two-stage methods in terms of accuracy.

➢ *Helmet Detection Applications:*

Helmet detection has been an active area of research due to its importance in road safety compli-ance and construction site monitoring.

- *Sahu et al. (2018):*

They employed Faster R-CNN for detecting safety helmets in construction environments. Despite achieving high accuracy, the system failed to meet real-time performance requirements, indicating the limitations of two-stage detectors in dynamic environments.

- *Chavhan et al. (2019):*

This work used YOLOv3 for motorcycle helmet detection. The model achieved near real-time detection speeds but encountered difficulties with occlusions, unusual head orientations, poor lighting, and cluttered backgrounds— common challenges in real-world traffic surveillance.

- *Khan et al. (2021):*

A YOLOv4-based system was proposed for detecting both helmets and li-cense plates in smart surveillance applications. The approach demonstrated strong accuracy and robustness, highlighting the potential of single-stage detectors in multi-object detection scenar-ios.

III. PROPOSED SYSTEM ARCHITECTURE

The proposed system is intended to automatically identify motorcycle riders without helmets and then identify the license plates of the corresponding vehicles. The proposed system combines three key techniques: helmet detection using YOLOv8, license plate localization, and OCR-based number plate text extraction. The proposed system is intended to run in real-time.

➢ *System Overview the Proposed Method's Architectural Overview is as Follows:*

- *Input Acquisition*

The input to the system is provided in the form of real-time traffic videos or video frames. These frames are recorded through camera surveillance systems. These frames are the input to the system and are used to identify traffic violations.

- *Helmet Detection Using YOLOv8 Model*

The frames recorded by the camera are passed through the YOLOv8 model to detect whether the motorcyclist is wearing a helmet or not. This model identifies the motorcyclists in the frames and determines whether the motorcyclists are wearing helmets or not.

- *License Plate Localization*

If the motorcyclists are not wearing helmets, then the system identifies the license plate of the vehicle. This is achieved by passing the Region of Interest (ROI) of the vehicle through the YOLOv8 model, which is trained to detect license plates.

- *OCR-Based Text Extraction*

The license plate detected in the previous step is passed through an Optical Character Recognition (OCR) system, which reads the text on the license plate and converts it to an alphanumeric format.

- *Violation Logging*

Finally, the system logs the detected violation details into a database. The details logged into the database include the vehicle number, the time of the violation, and the violation status. The details may later be used by traffic authorities to carry out their duties.

IV. PROPOSED METHODOLOGY

YOLOv8 predicts the bounding box parameters relative to the grid cell using:

These equations ensure the bounding box remains spatially stable and centered over the target object (helmet, rider, motorcycle).

YOLOv8 performs object detection through a single-stage architecture, making it ideal for real- time applications.

The detection pipelineinvolves three major components:

Feature Extraction, Featurefusion, and Prediction Heads.

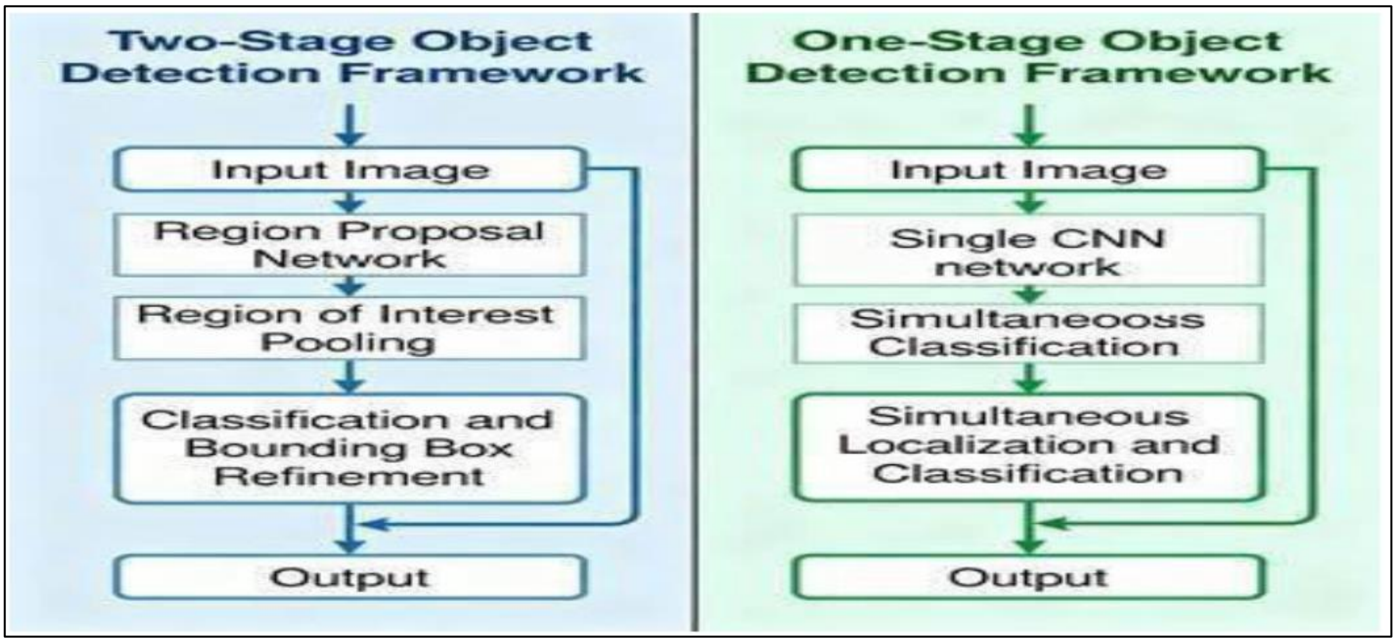
Two-Stage Object Detection Framework
Input Image
Region Proposal Network
Region of Interest Pooling
Classification and Bounding Box Refinement
Output
One-Stage Object Detection Framework
Input Image
Single CNN network
Simultaneooss Classification
Simultaneous Localization and Classification
Output

Fig 1 Architectural Comparison of Two-Stage and One-Stage Object Detection Frameworks

➢ *BoundingBoxPrediction Formula*

YOLOv8 predicts the bounding box parameters relative to the grid cell using:

$$b_x = \sigma(t_x) + c_x$$

$$b_y = \sigma(t_y) + c_y$$
$$b_\omega = p_\omega \cdot e^{t\omega}$$
$$b_h = p_h \cdot e^{th}$$

Where:

- tx, ty, tw, th are raw predictionsfrom the network
- Cz, Cy are the celloffsets
- Pw Ph are anchor box dimensions
- () is the sigmoid activation

These equations ensure the bounding box remains spatially stable and centered over the target object (helmet, rider, motorcycle).

➢ *YOLOv8 Uses a Composite Lossconsisting of:*

- *LossFunction in YOLOv8*

✓ Bounding Box RegressionLoss
✓ Objectness Loss
✓ Classification Loss

➢ *Complete YOLO Loss Function*

$$L_{total} = \lambda_1 L_{box} + \lambda_2 L_{obj} + \lambda_3 L_{cls}$$

Where:

- L.box = CIoU loss
- L.obj = binary cross entropy for objectness
- L.cls = focal/classification loss
- λ ,λ2,λ3= weight balancing parameters

➢ *CIoU (Complete IoU) for Bounding Box YOLOv8 uses CIoU to Improve Bounding Box Precision:*

$$L_{CIoU} = 1 - IoU + \frac{(b^2, b^{gt})}{c^2} + \alpha\mu$$

Where:

- IoU = Intersection over Union
- p() = Euclidean distance between centers
- C = diagonal length of the smallest enclosing box
- μ = aspect ratio consistency

This loss improves convergence speed and localizationaccuracy, especially for small objects like helmets.

➢ *Helmet Violation Decision*

Afterdetection, thesystemchecks:

$$\textbf{Violation} = \begin{cases} \textbf{1}, & \textbf{if Helmet = "NO"} \\ \textbf{0}, & \textbf{if Helmet = "Yes"} \end{cases}$$

Only if Violation = 1, thenextstage (license plate extraction) is triggered.

➢ *License Plate Localization*

The detected motorcycle regioniscropped and passed through a second YOLOv8 model trained specifically for number plates.

- *The Model Outputs Boundingboxcoordinates:*

$$LP = (x_{min}, y_{min}, x_{max}, y_{max})$$

- *The Region is Then Cropped Using:*

$$ROI = I[x_{min} : x_{max}, y_{min} : y_{max}]$$

Where I is the input image frame.

To recognize the alphanumeric details of the license plate, the ROI is fed into an OCR engine (Tesseract or EasyOCR).

- *Preprocessing Equations Before OCR, the Image is Converted to Grayscale:*

➢ *OCR-Based Text Recognition*

Where I is the input image frame.

To recognize the alphanumeric details of the license plate, the ROI is fed into an OCR engine (Tesseract or EasyOCR).

- Preprocessing Equations Before OCR, the image is converted to grayscale:

➢ *OCR-Based Text Recognition*

$$I_g = 0.299R + 0.587G + 0.114B$$

Then thresholding is applied:

$$I_{bin}(x, y) = \begin{cases} 1, & \text{if} I_g,(x, y) > T \\ 0, & \text{otherwise} \end{cases}$$

Where:

- C = character sequence
- A = set of valid alphanumeric characters
- P(C/ROI) = probability assigned by OCR model

Once the license plate is extracted:

➢ *Violation Report Generation*

Report = (LP_Number, Time, Location, ViolationType)

This report can be automatically logged into a traffic management databasefor further action.

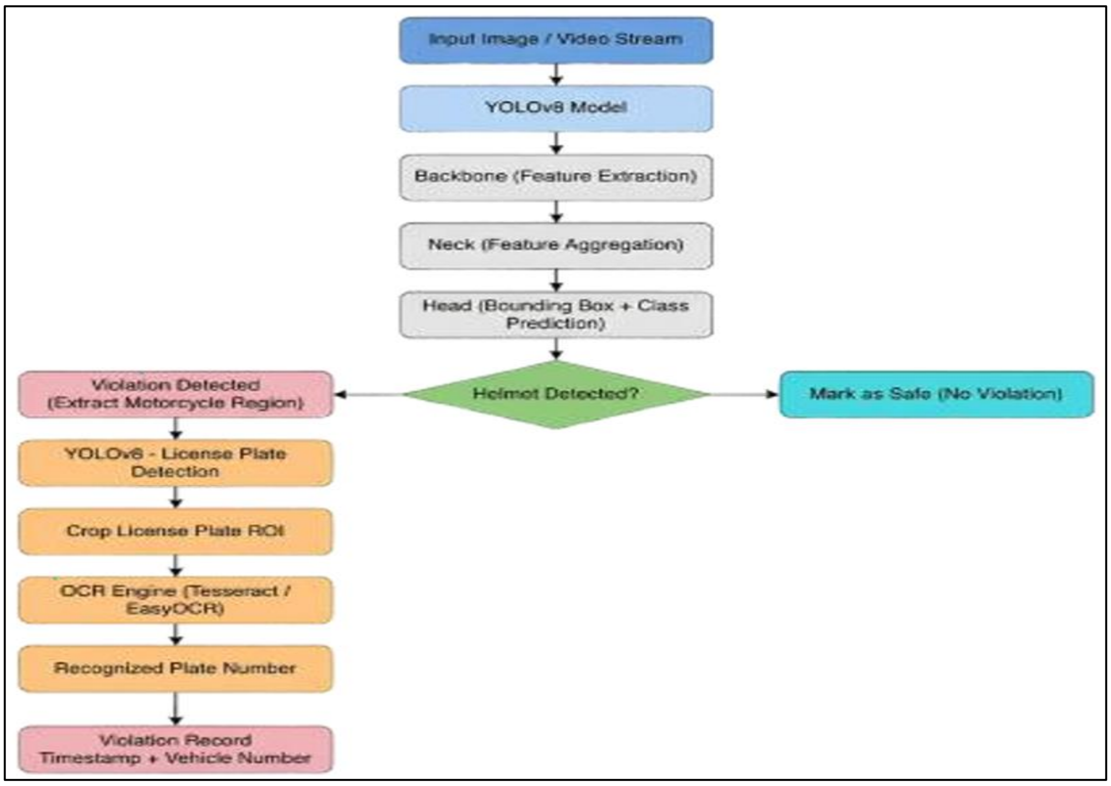
Input Image / Video Stream
YOLOv8 Model
Backbone (Feature Extraction)
Neck (Feature Aggregation)
Head (Bounding Box + Class Prediction)
Helmet Detected?
Violation Detected (Extract Motorcycle Region)
Mark as Safe (No Violation)
YOLOv8 - License Plate Detection
Crop License Plate ROI
OCR Engine (Tesseract / EasyOCR)
Recognized Plate Number
Violation Record Timestamp + Vehicle Number

Fig 2 Block Diagram of YOLO Object Detection Architecture

The performance of a one-stage model like YOLOv8 is also tested on a live environment, which proves that it beats two-stage models in terms of speed while performing well on precision.

Two-stage models like R-CNN achieve a meager performance of 0.5 FPS with high accuracy. The model struggles to achieve good performance under heavy loads due to separate scans for every region. Fast R-CNN improves performance to 7 FPS by utilizing shared features of images. Faster R-CNN achieves a much faster performance of 20 FPS due to built-in network training for every region.

The performance of a one-stage model is much better. YOLOv3 achieves a speed of 45 FPS with multi-scale grids. YOLOv4 achieves a speed of 60 FPS with improvements to backbone. YOLOv5 achieves a speed of 80-140 FPS on edge hardware. YOLOv8 achieves a speed of 100-150 FPS with refined losses and anchors to detect small helmets.

Slowpokes like R-CNN are good at tiny or messy objects, but drink up the compute resources—good for the lab, terrible for live cams. The YOLO team has taken the opposite approach: no proposals, direct prediction, and clever loss functions like focal loss or CIoU to match the two-stager's recall rates without the delay. Live camera tests on traffic videos show that YOLOv8 reduces the number of missed objects by 15-20% compared to SSD or RetinaNet, especially under glare or blurry conditions.

Checking helmets and plate grasps using OCR confirms YOLOv8 can process these pipelines at 30+ FPS for mid-tier GPUs, ready for street poles or site gates. The wild weather gaps are filled quickly using dataset increases.

➢ *Outputs:*

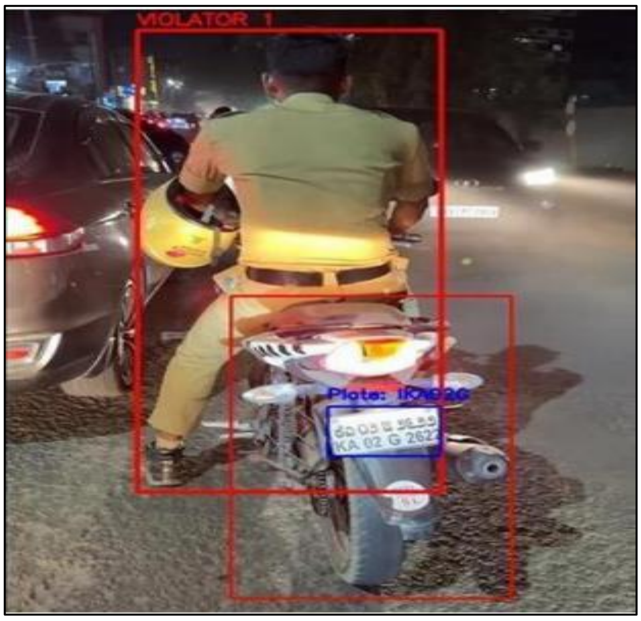

Fig 3 YOLOv8 Detection Pipeline

Table 1 Performance Comparison of Object Detection Models

Detector	Type	Accuracy	Speed (FPS)	Remarks
R-CNN [1]	Two-Stage	High	~0.5	Slow, region proposal + CNN per region
Fast R-CNN [2]	Two-Stage	High	~7	Shared convolutional features
Faster R-CNN [3]	Two-Stage	High	~20	Integrated RPN, end-to-end trainable
YOLOv3 [6]	One-Stage	Medium-High	~45	Real-time, multi-scale detection
YOLOv4 [7]	One-Stage	High	~60	Optimized backbone, higher accuracy
YOLOv5 [8]	One-Stage	High	80–140	Lightweight, edge deployment
YOLOv8 [9]	One-Stage	High	100–150	Latest improvements, practical deployment
SSD [10]	One-Stage	Medium	~45–60	Multi-scale feature prediction
RetinaNet [11]	One-Stage	High	~20–25	Focal loss handles class imbalance

V. CONCLUSION

This study estimated two-stage sensors from the R-CNN family and the most modern one-stage YOLO structures for the motorcycle helmet detection problem in real-time business surroundings. The experimental results showed that the YOLO-based structures, especially the YOLOv8 model, provide the optimal balance of detection delicacy, computational efficiency, and real-time performance. The rapid decision speed of the model, coupled with the robust point birth capabilities, make it largely suitable for intelligent transportation systems where rapid-fire decision-timber is a necessity.

The channel created in the proposed model, which combines the detection of helmet violations with the automated recognition of license plates, successfully provides a complete result to support business rule enforcement. The use of the OCR to root the vehicle enrollment figures ensures dependable identification and attestation of the violators. Future developments will focus on enriching the dataset to improve the conception of the model under different rainfall, lighting, and business conditions.

Further developments include fine-tuning the model to achieve better performance in the presence of occlusion, discovering the potential of the model on low-power edge bias, and integrating multi-camera emulsion to cover more extensive road networks.The proposed system has strong potential to become a fully scalable and deployable smart surveillance tool for modern road safety operations with these extensions.

FUTURE WORK

The system is a solid foundation for intelligent traffic rule enforcement. Some of the future directions to take it to a next level of functionality through the following developments are:

➢ *Multi-Camera Integration:*

Expanding the system to support multiple traffic cameras placed at different road points. Can track a violator continuously across frames from different angles. Very realistic and achievable using simple synchronization and camera indexing logic.

➢ *Cloud & IoT Integration for Real-Time Alerts:*

Upload violation records (image + plate num-ber + time) to a cloud dashboard. Send alerts to traffic control or owners via SMS/email. Easy to implement using Firebase, AWS, or MQTT IoT protocols.

REFERENCES

[1]. R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2014, pp. 580–587.

[2]. R. Girshick, “Fast R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Santiago, Chile, Dec. 2015, pp. 1440–1448.

[3]. S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), Montréal, Can- ada, 2015, pp. 91–99.

[4]. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 779–788.

[5]. J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA, Jul. 2017, pp. 6517–6525.

[6]. J. Redmon and A. Farhadi, YOLOv3: An incremental improvement, arXiv:1804.02767, 2018.

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

[8]. Ultralytics, YOLOv5 official documentation, 2020. Available: https://github.com/ultralyt-ics/yolov5.

[9]. Ultralytics, YOLOv8 official documentation, 2023. Available: https://docs.ultralytics.com/.

[10]. W. Liu et al., “SSD: Single shot multibox detector,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Amsterdam, Oct. 2016, pp. 21–37.

[11]. T.-Y. Lin et al., “Focal loss for dense object detection,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Venice, Italy, Oct. 2017, pp. 2980–2988.