VISION-BASED TRAFFIC LAW VIOLATION DETECTION SYSTEM USING CONVOLUTIONAL NEURAL NETWORKS

[1]Thriveni,[2]Vyshnavi,[3]Shaik Shakeer
[123]Students
Department Of Computer Science and Engineering

ABSTRACT

With the rapid increase in vehicular traffic and corresponding violations, conventional monitoring systems are proving inadequate for enforcing road safety laws. This paper presents a vision-based traffic law violation detection system that leverages Convolutional Neural Networks (CNNs) to identify and classify instances of rule-breaking such as red-light jumping, helmetless riding, and lane violations in real-time.

The proposed system processes live traffic video feeds using computer vision techniques and deep learning models to accurately detect vehicles and assess their compliance with predefined traffic rules. CNN architectures are employed for object detection and classification, enabling precise recognition of vehicle types, number plates, and behaviors that indicate infractions. The system is trained on a custom-labeled dataset collected from real-world traffic footage, achieving high accuracy in detecting multiple types of violations under varied lighting and environmental conditions.

Incorporating automated number plate recognition (ANPR) and a rule-based decision engine, the system generates violation reports and stores evidence in a centralized database, paving the way for smart traffic enforcement. Experimental results demonstrate the system's robustness, real-time performance, and potential scalability for city-wide deployment. This approach offers a cost-effective, scalable, and automated solution for improving traffic compliance and reducing manual monitoring efforts through AI-driven surveillance.

I. INTRODUCTION:

Rapid urbanization and the exponential rise in the number of vehicles have created serious challenges for traffic management and law enforcement agencies worldwide. In developing countries like India, the lack of adherence to traffic rules has become a major contributor to road accidents, congestion, and fatalities. Traditional surveillance systems, which rely heavily on human monitoring and manual enforcement, are not only inefficient but also prone to errors and delays. This necessitates the development of automated, intelligent systems that can detect and report traffic violations in real time.

The emergence of Artificial Intelligence (AI), especially Deep Learning and Computer Vision, offers promising avenues for creating such automated systems. Among the various AI techniques, Convolutional Neural Networks (CNNs) have proven highly effective in processing visual data, such as identifying objects, analyzing motion, and detecting anomalies in real-time video streams. By integrating CNNs with traffic surveillance footage, it becomes possible to accurately detect rule violations such as red-light jumping, riding without helmets, driving in the wrong lane, or illegal overtaking.

This paper proposes a vision-based traffic law violation detection system powered by CNNs. The system is designed to process live or recorded video feeds from traffic cameras, detect moving vehicles, identify their attributes, and compare their behavior against a set of pre-defined traffic rules. Detected violations are captured, logged, and linked to the corresponding vehicle's license plate using

Automatic Number Plate Recognition (ANPR) techniques.

The implementation aims to improve the speed, accuracy, and consistency of traffic law enforcement, reduce the burden on human personnel, and serve as a deterrent for habitual offenders. By leveraging deep learning and edge computing technologies, the system offers scalability, adaptability to various urban settings, and integration capabilities with smart city infrastructure.

II. LITERATURE SURVEY

The application of computer vision and deep learning in intelligent traffic systems has garnered considerable attention in recent years. Researchers have explored multiple techniques to automate the detection of traffic violations such as red-light jumping, helmetless riding, and wrong-lane driving. This literature survey highlights key studies and developments in the domain of vision-based traffic rule enforcement using deep learning models, particularly Convolutional Neural Networks (CNNs).

1. Red-Light Violation Detection Systems

Patel et al. (2018) developed a red-light violation detection system that used image subtraction and motion tracking to detect vehicles crossing the stop line after the red signal. However, the system faced limitations in poor lighting conditions and failed to scale with traffic density. Recent works have moved toward deep learning-based object detection models like YOLO and SSD to address such limitations.

2. Helmet Detection Using CNN

Singh and Verma (2019) proposed a CNN-based helmet detection model using street camera images. Their model classified riders into helmet-wearing and non-helmet-wearing categories, achieving over 90% accuracy under controlled lighting. However, generalizing the model for real-world deployment required further training with diverse datasets. The present work addresses this by including video data under varied environmental conditions for robustness.

3. Lane Violation and Wrong-Way Driving Detection

Sharma et al. (2020) implemented a lane detection algorithm using OpenCV and Hough Transform to detect wrong-way driving. While effective in simple scenarios, traditional image processing techniques struggled in noisy or cluttered scenes. With CNNs and semantic segmentation models like DeepLab, more recent studies have demonstrated better accuracy in lane identification under complex urban scenarios.

4. Vehicle Detection and Tracking Using Deep Learning

Kumar et al. (2021) utilized a YOLOv3 model combined with Kalman Filters for real-time vehicle detection and tracking. This hybrid model performed well in multi-object tracking but required high computational resources. In the proposed system, a lightweight CNN model optimized for real-time inference is employed to ensure speed-efficiency trade-offs.

5. ANPR (Automatic Number Plate Recognition)

A study by Zhou et al. (2022) combined CNN and Optical Character Recognition (OCR) techniques for detecting and reading license plates in real-time. Their pipeline included object detection for plate localization and a character-segmentation CNN for reading alphanumeric content. The proposed system builds on this by integrating ANPR as a post-processing step after violation detection for automatic ticket generation.

III. PROPOSED METHODOLOGY

In this section, we will explain the proposed end to end system. This system consists of four major components. The system architecture is shown in figure 3.1. In the proposed system we first give an image frame from CCTV footage as input to the system then perform vehicle detection using Object Detection. Object

Detection is carried out using YOLO (you look only once) for detecting vehicles in the image frame. After detecting vehicles, individual vehicles are cropped out using the coordinates obtained from bounding boxes given by Object Detection Algorithms. Now an individual vehicle is checked against different violations. Violations included in this proposed system are Helmet Violation (Two wheeler rider not wearing a helmet) and Crosswalk Violation (Violating Zebra Crossing). Two-wheeler vehicles will be checked against Helmet and Crosswalk Violation and Four Wheeler vehicles will be checked against Crosswalk Violation. Helmet violation is detected using CNN (Convolutional Neural Network) based classifier which works well on visual data. Crosswalk violation can be detected using Mask RCNN where instance segmentation helps in comparing coordinates of the bottom of tyre of the vehicle with those of detected crosswalk. Once violation(s) are detected for a vehicle then the number plate of the corresponding vehicle is detected using Object Detection. Again YOLO is used for detection of the number plate of a vehicle. OCR (Optical Character Recognition) is used to obtain license number from the number plate. Vehicle users are notified with associated violations and violations are inserted into the database. The database can be used to obtain statistical analysis on traffic rules violations that previously occurred. Now we will look at each individual module in detail.

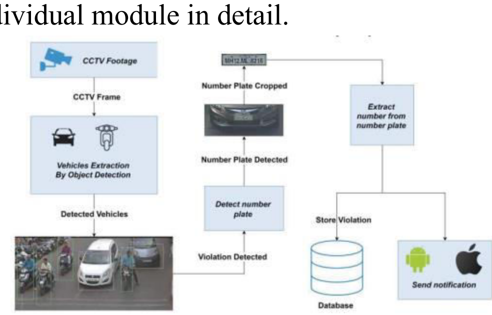

Figure 1: Architecture Diagram for proposed system

Our proposed methodology for each module is given below:

A. Vehicle Detection

When the signal goes red, a frame is obtained from the CCTV camera. Now, the frame is passed as an input to the object detection module for vehicle detection. We are using YOLO (You Only Look Once) for this purpose. YOLO is an object detection algorithm [10]. YOLO predicts the coordinates of bounding boxes directly using fully connected layers on top of the convolutional feature extractor. Predicting offsets instead of coordinates simplifies the problem and makes it easier for the network to learn. YOLO is better than other object detection algorithms like R-CNN, fast R-CNN, and faster R-CNN as they use pipelines to perform the detection of objects which incorporates multiple steps. Hence these algorithms are slow to run and hard to optimize as each individual component should be trained separately. YOLO can do this task with a single neural network, thus outperforming the above algorithms.

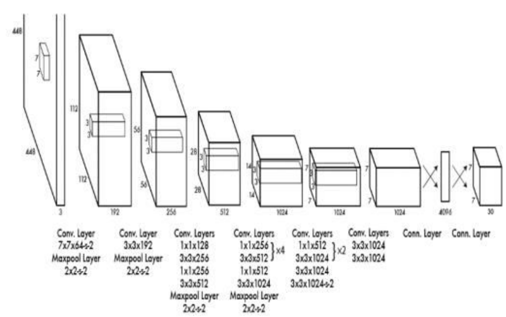

Figure 2: Architecture of YOLO [10]

Figure 2 shows the architecture of YOLO which has 24 convolutional layers and 2 fully connected layers. 1x1 alternate convolutional layers reduce the space of the features from the layers preceding them [10].

The pre-trained model of YOLO[12] detects vehicles and persons as separate objects. Hence we built a custom model that detects the vehicles as well as the riders as a single object within the same bounding box as shown in figure 3. The custom model was built by using weights from the pre-trained model and the concept of transfer learning. Vehicle detection is performed using object detection.

Figure 3: Object detection with YOLO

In fig. 3.3, it is seen that the objects are detected and classified in “two” and “four” classes. The green bounding box represents class “two” and the pink bounding box represents class “four”.

B. Helmet Classification

After vehicles are detected from images, Two-wheelers are considered for the Helmet Classification task. Individual vehicles are cropped from the cctv frame with the help of corresponding bounding box, if the vehicle belongs to two wheeler class they are further cropped where we consider the upper half of the individual vehicle (upper 50% of vehicle frame). Now the upper half is considered as statistically, heads (helmets) are located in the upper half of the image. By considering the upper half of the image computational complexity of the system could be reduced. This upper half is given as input to the classifier, which outputs whether the image belongs to helmet or non-helmet class. Fig .4 depicts the flowchart for helmet classification.

The classifier used for helmet classification employs a CNN architecture as shown in fig. .5. It is a 12 layered architecture that includes various layers such as 5 Convolutional layers using ReLu as an activation unit, 4 pooling layers, and a single dense layer using softmax for classification into two classes. CNN is preferred over other methods as it is better at extracting visual features from image data.

Fig. .6 represents the feature maps that are generated from output of convolutional layer. These feature maps illustrate that the CNN learns common hidden features and structures among helmets and heads in the training set while training, thus being able to distinguish between helmet and non-helmet class.

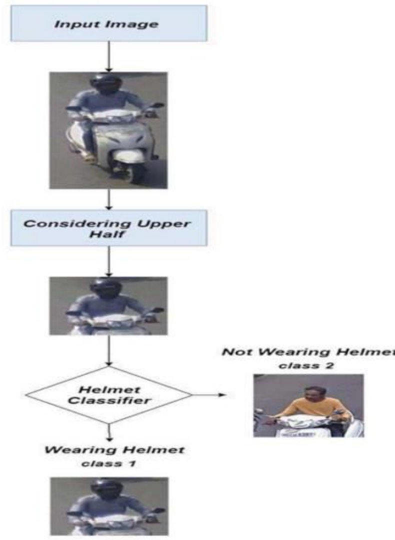

Figure 4: Helmet classification using CNN. Input image is classified as “wearing helmet” or “not wearing helmet”.

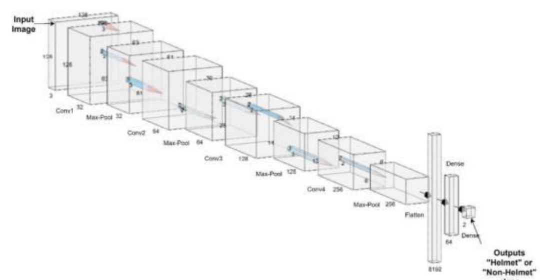

Figure 5: Architecture of CNN for Helmet Classification

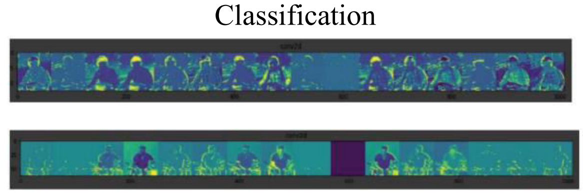

Figure 6: Visualization of the trained representation by CNN for the classification of helmet and non-helmet class

C. Crosswalk violations at traffic signals

Crosswalk violation is detected using instance segmentation. Mask R-CNN is an extension to faster R-CNN which includes a segmentation mask along with bounding boxes of objects segmented [11]. Segmentation gives us a pixelwise mask for each object. Instance segmentation model was trained for two classes namely vehicle and crosswalk. Training mask R-CNN requires significant amount of time, hence we made use of pretrained mask R-CNN weights [13].After segmentation is performed we obtain a mask for each vehicle and zebra in the image along with their bounding boxes.

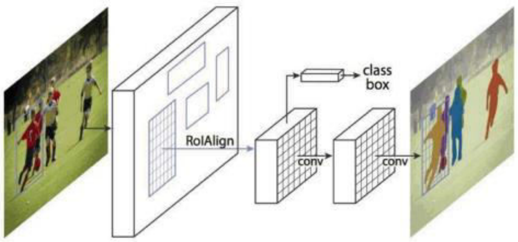

Figure 7: The Mask R-CNN framework for instance segmentation [11]

Crosswalk violation occurs if a vehicle is present on a crosswalk (Zebra crossing). First, we obtain coordinates of the bottom of the tyre with the help of the bounding box of the respective vehicle. Now we check if the obtained coordinates overlap with the crosswalk, this can be achieved by comparing coordinates obtained with those of the bounding box of the crosswalk. We assume top-left corner of image frame as origin (0,0), Considering (Zx1,Zy1) and (Zx2,Zy2) as coordinates of bounding box of crosswalk of top-left corner and bottom-right corner respectively, (Vx1,Vy1) and (Vx2,Vy2) as coordinates of bounding box of a particular vehicle of top-left corner and bottom-right corner respectively. We compare Zy1 with Vy2, if Vy2 > Zy1 where Vy2 will be y coordinate of bottom of tyre, then a violation has said to be occurred. Crosswalk violation is carried out for each vehicle by checking if the coordinate of the bottom of tyre lies within the crosswalk. It can be observed from fig 3.8 that a four-wheeler (masked yellow) and 2 two-wheelers (marked as green and purple) have violated crosswalk and their coordinates of the bottom of tyre lie over the crosswalk.

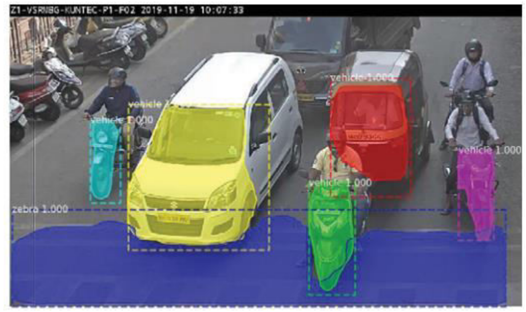

Figure 8: Instance segmentation using Mask R-CNN

D. License Plate Recognition

After detecting a violation of traffic rules, the task to be carried out is to get the vehicle number from the vehicle that violated traffic rules. First, we localize the number plate of the vehicle using Object Detection. Localization of number plates is carried out using YOLO[10] again. Localized license plates of the vehicles that violated any traffic rule are cropped out. The vehicle number is extracted by performing OCR (Optical Character Recognition) on the cropped number plate. In OCR character segmentation is carried out followed by template matching. Characters are segmented individually and then compared with characters in the database using template matching. Once the vehicle number is obtained the violation is stored in the database and the corresponding vehicle user is notified about the same through SMS or email.

Figure 9: shows the detection of the number plate of the vehicle, then the characters are read using OCR

IV. DATASET

The dataset used was prepared from cctv footage of various signals obtained from pune city's traffic police department. Dataset was generated from the cctv footage by obtaining image frames at a regular interval of approx 15 secs to avoid repetition of images. The images consisted of both sparse and dense traffic conditions, in order to include diversity in data.

TABLE I: DATASET USED FOR DIFFERENT MODULES

Dataset	No. of images	No. of classes	Name of classes
Vehicle Detection	525	2	two, four
Helmet Classification	1079	2	helmet, non-helmet
Crosswalk Violation	200	2	vehicle, crosswalk
Number Plate Detection	450	1	numplate

A total of 525 images were taken for vehicle object detection which consisted of various types of vehicles including bikes, cars, buses, and rickshaws. Out of 525, 420 images were considered for training and 105 for testing. A total of 1079 images were considered for the helmet classification, 492 of which were people wearing helmet and 587 without helmet. The images for helmet classifier were obtained by applying vehicle detection on the vehicle detection dataset and then selecting the vehicles with two wheeler class. Out of 1079 images 864 were considered for training and 215 for testing. For detection of zebra violation which employs image segmentation, 200 images were considered in which polygons were created around the objects to highlight its pixels and help prepare the masks for training mask-rcnn network. Out of 200,150 images were considered for training and 50 images for testing. Number. Number plate detection dataset was prepared by labelling individual vehicle's number plate considering both two wheelers and four wheelers. It consisted of total of 450 number plates, out of which 350 were used for training and 100 were used for testing.

V. RESULT

In this section, we present experimental results of our modules viz. vehicle detection, helmet classifier and Image Segmentation. The experiments are carried on Ubuntu 18.04 machine consisting of i5 6th gen processor, 12 gb of ram and 256 gb SSD. Python 3.65 was used for evaluation of results along with libraries such as opencv-python for image processing, tensorflow as a Deep Learning framework (for training and prediction), numpy for mathematical operations and matplotlib for visualization.

1. Vehicle Detection:

Vehicle detection is performed using YOLO algorithm, which was trained on 420 images consisting of both the classes, two and four. This was evaluated against 105 test images considering metric as average precision per class and also calculating the mean average precision. The results for which are as follows:

TABLE II: RESULTS OF VEHICLE DETECTION

Class	Average Precision
Four	0.9406
Two	0.9594

The mean average precision(mAP) for the model is 0.95.

2. Helmet Classification:

Helmet Classifier performs image classification using CNN consisting of five convolutional layers with ReLu activation units, 4 max-pooling layers, and one fully connected dense layer with final softmax unit for classification into two classes. The proposed model architecture is evaluated using metrics such as precision, recall, accuracy, and F1-score on a test-set consisting of 215 images where 98 images belonged to helmet class and 117 belonged to non-helmet class.

TABLE III: RESULTS OF HELMET CLASSIFICATION

Metric	Value
Precision	0.8780
Recall	0.8925
Accuracy	87%
F1 score	0.8852

3. Crosswalk Violation:

Crosswalk Violation employs use of mask-rcnn network [11]. This network is trained on 150 images with 2 classes namely vehicles, and crosswalk(zebra) using the concept of transfer learning. This network was then evaluated against 50 images each consisting of all of the 3 classes. Metric used for evaluation is mean Average Precision(mAP) is computed on a testset consisting of 50 images.

TABLE IV:RESULTS OF CROSSWALK VIOLATION

Metric	Value
mean Average Precision (mAP)	0.96

4. Number Plate Detection:

Number plate detection is performed using YOLO again which was trained on 450 images. This was evaluated against 100 test images considering metric as average precision. The results for which are as follows:

TABLE V: RESULTS OF NUMBER PLATE DETECTION

Class	Average Precision
numplate	0.9305

VI. CONCLUSION

In this paper, a deep learning-based traffic law violation detection system has been presented to address the growing need for intelligent, automated traffic monitoring and enforcement. By integrating Convolutional Neural Networks (CNNs) with computer vision techniques, the proposed system effectively detects common violations such as red-light jumping, helmetless riding, and lane indiscipline in real-time using traffic surveillance footage.

The use of CNNs significantly enhances the accuracy and reliability of vehicle and object detection compared to traditional image processing methods. The system also incorporates Automatic Number Plate Recognition (ANPR) for identifying violating vehicles and generates actionable outputs that can be used by traffic authorities for enforcement. Designed for scalability, this framework supports the real-time analysis of video feeds, making it suitable for smart city applications.

Experimental results validate the system's high performance in various lighting and traffic conditions, showcasing its potential to reduce manual workload, minimize human error, and improve road safety compliance. Moreover, the framework is cost-effective and can be integrated with existing CCTV infrastructures, making widespread deployment feasible.

In conclusion, this deep learning-driven system paves the way for a more efficient, automated, and intelligent traffic management ecosystem. Future improvements may include integrating facial recognition, multi-violation detection per vehicle, and edge AI models for faster on-site decision-making, enhancing both enforcement and prevention capabilities in urban mobility governance.

REFERENCES

1. https://www.indiatoday.in/diu/story/two-wheeler-deathroad-accidents-helmets-states India-1602794-2019-09-24
2. J. Chiverton, "Helmet presence classification with motorcycle detection and tracking," Intelligent Transport Systems (IET), vol. 6, no. 3, pp. 259–269, September 2012
3. Gomathi, et al, "Automatic Detection of Motorcycle without helmet using IOT", South Asian Journal of Engineering and Technology
4. Samir Ibadov, et al, "Algorithm for detecting violations of traffic rules based on computer vision approaches", MATEC Web of Conferences 132, 05005 (2017)
5. Aaron Christian P. Uy, et al, "Automated Traffic Violation Apprehension System Using Genetic Algorithm and Artificial Neural Network", 2016 IEEE Region 10 Conference
6. Amey Narkhede, et al, "Automatic Traffic Rule Violation Detection and Number Plate Recognition", IJSTE - International Journal of Science Technology & Engineering | Volume 3 | Issue 09 | March 2017
7. Ana Riza, et al, "Localization of License Plates Using Optimized Edge and Contour Detection Technique ", 2017 IEEE Region 10 Conference (TENCON), Malaysia, November 5-8, 2017
8. Prem Kumar Bhaskar, et al, "Image Processing Based Vehicle Detection and Tracking Method ",Universiti Teknologi PETRONAS Seri Iskandar, 31750, Perak, Malaysia
9. Kunal, Dahiya ; Dinesh, Singh ; and C. Krishna, Mohan "Automatic Detection of Bike-riders without Helmet using Surveillance Videos in Real-time". 2016 International Joint Conference on Neural Networks (IJCNN)
10. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection"
11. Kaiming He, Georgia Gkioxari, Piotr Doll´ar, Ross Girshick, "Mask R-CNN", Facebook AI Research (FAIR) ,24 JAN 2018
12. http://pjreddie.com/media/files/yolo.weights
13. https://github.com/matterport/Mask_RCNN/releases/download/v2.0/mask_rcnn_coco.h5
14. Pooya Sagharichi Ha, Mojtaba Shakeri, "License Plate Automatic Recognition based on Edge Detection"
15. Abdullah Asım, et al, "A Vehicle Detection Approach using Deep Learning Methodologies"