Two-Wheeler Vehicle Traffic Violations Detection and Automated Ticketing for Indian Road Scenario

R. Shree Charran and Rahul Kumar Dubey, *Senior Member, IEEE*

***Abstract*— Traffic violation monitoring and control is a major concern in India due to excess crowd, increasing commuters, bad traffic signal management, and rider mentality. It is obvious that physical traffic police-based monitoring alone is insufficient to monitor such large traffic volumes and simultaneously track violations. This has led to many violators going unnoticed. The violators, in turn, cause more serious mishaps on the road resulting in danger to their own life as well as to other's life. Thus, there is a need for incorporating Artificial Intelligence (AI)-based techniques to eliminate manual intervention for the detection and catching of violators. In this paper, we propose a system to automatically detect two-wheeler violations like not wearing a helmet, usage of a phone while riding, triple riding, wheeling, and illegal parking for Indian road scenarios and eventually automating the ticketing process by capturing the violations and corresponding vehicle number in a database. We propose using a custom trained Yolo-v4 + DeepSORT for violation detection and tracking and Yolo-v4 + Tesseract for number plate detection and extraction. This implementation obtained a mean average precision (mAP) of 98.09% for violation detection and an accuracy of 99.41% for number plate detection on the test data. Further, the system detected 77 out of 93 violations with zero false positives in real-life scenarios. Thus,showing that the traffic violation system developed can be used to automate traffic violation ticketing. The developed system would be particularly useful in deriving various safety-related policies and will help to enforce strong regulation of traffic rules and build towards a smart city ecosystem via the automated AI-based traffic violation and ticketing system.**

***Index Terms*— Artificial intelligence (AI), detection, smart city, traffic violation.**

I. Introduction

TRAFFIC monitoring and traffic violation control is a major concern for the Indian Government, due to the excess crowd, increasing commuters, bad traffic system designs, people mentality. Most India's urban traffic management systems today are still manually monitored. This leads to large traffic congestion and human errors. More than one-third (37%) of those killed in road accidents in 2019 were two-wheeler riders, as per the Ministry of Road Transport and Highways' report [1]–[3]. Further, not wearing helmets resulted in 44,666 deaths (30,148 drivers and 14,518 pillion passengers) i.e., 29.82% of total road accident fatalities during 2019. The World Health Organization's (WHO) findings have shown that just correct helmet use could reduce the risk of fatal injuries by 42% and head injuries by 69% [2]. Additionally, WHO reported that drivers using mobile phones are four times more likely to be involved in a crash compared to drivers not using a mobile phone while driving. Using a phone while driving slows reaction times and causes the inevitable. These statistics are for only two of the several types of traffic violations noted for two-wheelers.

Manuscript received 28 August 2021; revised 6 March 2022 and 29 May 2022; accepted 22 June 2022. Date of publication 12 July 2022; date of current version 7 November 2022. The Associate Editor for this article was N. Bekiaris-Liberis. *(Corresponding author: Rahul Kumar Dubey.)*

R. Shree Charran is with the Department of Management and Studies, Indian Institute of Science, Bengaluru, Karnataka 560012, India (e-mail: shreer@alum.iisc.ac.in).

Rahul Kumar Dubey is with the Robert Bosch Engineering and Business Solutions Private Ltd., Bengaluru, Karnataka 560095, India (e-mail: rahulkumar.dubey@in.bosch.com).

Digital Object Identifier 10.1109/TITS.2022.3186679

The total capacity of traffic police in India was a little over 72,000 personnel while the number of vehicles currently on Indian roads is upward of 200 million, according to the report of the Bureau of Police Research and Development published in 2017 [4]. Traffic Police monitoring alone is not enough to handle the large volume and variety of vehicles. Furthermore, many existing solutions lack the flexibility to recognize, analyze, and track the large variety of vehicle types, license formats, ever-changing traffic patterns, and street layouts [5]–[8]. Many cities still have outdated control systems that simply cannot be scaled up efficiently to handle the volume and variety of traffic. To address these varieties of challenges modern technologies, need to be implemented to monitor traffic and automated enforcement. Thus, enabling smarter traffic control systems.

Effective enforcement includes establishing, regularly updating, and enforcing laws at the national, municipal, and local levels that address the mentioned risk factors. It includes levying appropriate penalties. If traffic laws are not enforced efficiently, then they cannot bring about the expected reduction in road traffic fatalities and injuries related to specific behaviors. Thus, if the traffic laws are not enforced or are perceived as not being enforced it is likely they will not be followed properly and will have negative influencing behavior. Thus, video-based artificial intelligence systems are becoming a vital tool for governments to automate traffic control, detect violations, and even dispense tickets.

With this as motivation, in this paper, we propose a computer vision-based system to help resolve the traffic violation detection challenges in all the dimensions i.e., to minimize human interference and cost, to bring high accuracy, automated data retrieval, and ticketing system.

II. Background

A. Vehicle and Violation Detection

Aniruddha et al. [9], proposed a framework to detect helmet violations and crosswalk violations. The framework consists of models running in sequential order, beginning with detecting all vehicles in the frame using YOLO, and then each vehicle being cross-checked for helmet violations. The Helmet violation is detected using a CNN-based classifier and the crosswalk violation is detected using instance segmentation by mask R-CNN architecture. The main drawback of the suggested framework is that it requires high computational resources as it requires running sequentially of multiple modules of object detection, classification, and image segmentation and is not feasible in real-time.

Chiverton [10], proposed an approach to classify and track motorcycle riders with and without helmets. The author proposes the use of a Support Vector Machine (SVM) trained on histograms derived from cropped images of helmets of riders for classification. For Detection, the motorcycle riders are segmented from data using background subtraction and the heads of the riders are isolated and then classified using a trained classifier. The major demerit observed was that the background subtraction stage is unable to accurately segment the entire head of the rider due to varying sizes captured and background noise. Additionally, this approach tries to locate the helmet in the computationally expensive full-frame, and it may often confuse other similar shaped objects as the helmet.

Dahiya et al. [11], proposed a similar approach for helmet detection by first detecting bike riders using background subtraction and object segmentation. Then to determine whether the bike rider is using a helmet or not using visual features and a binary classifier. *Wang et al. [12]*, improved the background-updating algorithm by using wavelet transform on the dynamic background and then tracking moving vehicles by feature-based tracking method for more fast and accurate detection and tracking. *Mallela et al. [13]*, proposed a deep learning framework using a convolutional neural network-based YOLO algorithm for the detection of the number of persons riding a bike to classify violations. *Reddy et al. [14]*, proposed using YOLO to detect persons and motorcycles. Further, to detect the helmet present in the image they used HOG features and an SVM classifier trained on color histogram features of helmets.

Mandal et al. [15], performed a detailed survey of object detection and tracking algorithms for vehicle counting. The authors compare the current state-of-the-art object detection models like YOLOv3, EfficientDet along with several object-tracking techniques to find the most accurate algorithm for vehicle tracking and counting. The algorithm was tested in daylight, night light, and rainy conditions. The counting results show that the best performing model combinations were YOLOv3 and tracker Deep SORT across all types of vehicles.

B. Number Plate Recognition

Kumari et al. [16] proposed a system to automatically detect number plates with initial pre-processing done with gray scaling, thresholding, and an unwanted line elimination algorithm to remove unwanted lines from the image. Then a Sobel operator is used to extract the extracts edges from the image, and then the plate region is extracted using the aspect ratio. Hough transformation is used for feature extraction and Artificial Neural Network (ANN) is used for character recognition. Gaussian and median filters are proposed to deal with different illumination conditions.

Kocer et al. [17] proposed a similar approach with canny edge detection being used for localization of the plate. Contrast extension and median filtering are used to enhance the image for segmentation. And finally, for character recognition, they used ANNs. Singh *et al.* [18] proposed a deep learning-based approach, where a Faster RCNN - InceptionV2 which is trained on the Coco dataset for detection of number plate. After basic pre-processing, Tesseract LSTM-OCR engine was used for segmentation and recognizing text within a cropped number plate. Authors have used Yolo-v2 for vehicle detection and the Fast-YOLO model for Number Plate detection. A character segmentation CNN and a character recognition CNN are trained for final predictions.

Table-I gives a comparative study of the literature. The following are the observations/issues that we have tried to tackle in this paper:

- The majority of the work done is for helmet detection and counting of passengers and no other violations.
- Number plate detection is almost done exclusively with no other additions as the process is pre-processing intensive.
- Most of the Literature concentrates on the detection from pre-processed images but not live feed which requires real-time processing and preparation of frames and real-time predictions.
- There is not much literature that combines violation detection with number plate detection and almost none on live videos and live ticketing for multiple violations combination as this involves sequential detections making it computationally expensive.
- No Literature to effectively describe an architecture to handle real world scenarios or issues.

TABLE I
COMPARATIVE ASSESSMENT WITH EXISTING APPROACHES

	[9]	[10]	[11]	[12]	[13]	[14]]	[16]	[17]	[18]	Purposed
Helmet Violations	Yes	Yes	Yes	Yes	Yes	Yes	NA	NA	NA	Yes
No. of Riders	NA	NA	NA	NA	Yes	NA	NA	NA	NA	Yes
2-Wheeler Detection	NA	NA	NA	NA	NA	Yes	NA	NA	NA	Yes
Number Plate Detection	NA	NA	NA	NA	NA	NA	Yes	Yes	Yes	Yes
Triple Riding	NA	NA	NA	NA	NA	NA	NA	NA	NA	Yes
Wheeling	NA	NA	NA	NA	NA	NA	NA	NA	NA	Yes
Mobile Usage	NA	NA	NA	NA	NA	NA	NA	NA	NA	Yes
No Parking	NA	NA	NA	NA	NA	NA	NA	NA	NA	Yes
Ticketing	NA	NA	NA	NA	NA	NA	NA	NA	NA	Yes

With these observations as our motivation, we propose a methodology that can detect multiple violations with negligible added computational cost using a State-Of-The-Art object detector to completely cover all reasonable violations for a two-wheeler which was previously not done. Furthermore, we propose direct use of object detectors to detect violations rather than extracting persons and then the violations as a two-phased approach as proposed in most literature. Secondly, we propose the use of an object tracker to help remove duplicates of count of violations and capture only high probability frames from live videos to be processed for number plate detection. Finally, we strategically use the same detector module used in violation detection to localize the number plates preventing additional computation of pure image processing or feature extraction approaches saving a whole step, and finally store the detections in a database to make a full-fledged ticketing system.

III. PROPOSED METHODOLOGY

In this section, we will explain the proposed end-to-end system to automate traffic violation detection and ticketing. The system consists of four major components namely: vehicle detection, vehicle tracking, number plate detection and ticketing.

- We first obtain the frames from the live footage as input to perform violation detection using an **Object detection module**. The violations covered for our study are *not wearing helmets, usage of phones while riding, Triple riding, wheeling, and no parking* in Indian scenario. The object detection is carried out using YOLO-v4 [1] object detector to detect one or more violations in each image frame.
- **Object tracking model:** The detected object is then tracked using the object tracking model to avoid repetitive counting and to ensure capturing the vehicles across multiple frames.
- **Licence plate detection:** After detecting and tracking the violations, each violating vehicle is cropped out using the coordinates obtained from bounding boxes given by object detection module of vehicle detector module. YOLO-V4 is used for the detection of the number plate of a vehicle from cropped vehicle images. Tesseract [8] is used for OCR (Optical Character Recognition) is used to extract the license numbers from the number plate and store them in a database.
- **Ticketing:** Vehicle users are notified of associated violations from the database. The database can be used further for statistical analysis on traffic rules violations.

The proposed system for Two-Wheeler Traffic Violation Detecting and ticketing is summarised in Figure 1 and explained further below.

A. Object Detector Module

Object detection is concerned with the identification and localization of objects, in our case the violations. We have used YOLOv4 [1] with pre-trained weights, as this is a pre-trained setup capable of identifying 9000 classes of objects and can be fine-tuned to capture

TABLE II

SUMMARY OF BOF AND BOS USED IN YOLOV4 FOR ENHANCED DETECTION

BoF for Backbone	BoS for Backbone	BoF for Detector	BoS for Detector
Cut-Mix and Mosaic Data Augmentation.	Mish Activation.	CIoU Loss.	Mish Activation.
Drop Block Regulation.	Cross Stage Partial Connections.	CmBN.	Modified SPP Block.
Class Label Smoothing.	Multi Input Weighted Residual Connections.	Drop Block Regulation.	Modified SAM Block.
		Mosaic Data Augmentation.	Modified PAN Path Aggregation Block.
		Self Adversarial Training.	DIoU NMS.
		Eliminate Grid Sensitivity.	
		Multiple Anchors.	
		Cosine Annealing Scheduler.	
		Optimal Hyper Parameter.	
		Random Training Shape.	

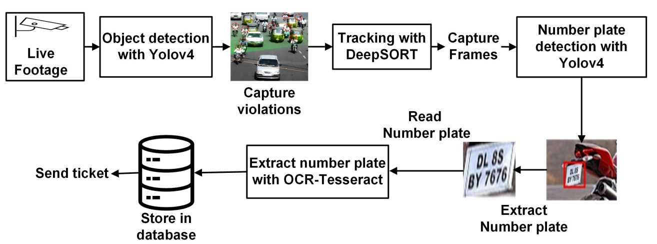

Fig. 1. Proposed architecture for two-wheeler traffic violation detecting and ticketing.

our select objects. The reason for the choice of Yolov4 is that it's twice as faster and more accurate in comparison to most other object detectors [1]. In addition, Yolov4's Average Precision of detections and Frames Per Second processing exceeds the 10% and 12% in absolute terms when compared to YOLOv3 which is its predecessor and closest State-of-the-Art competitor. The summary of Yolov4 components is as below:

- **Backbone** - This network is responsible for feature mapping using pre-trained extractors from input images. YoloV4 uses the CSPDarknet53 neural network as the feature extractor. Yolov4 additonally uses two methods to improve the object detector's accuracy i.e. Bag of freebies (BoF): that can make the object detector receive better accuracy without increasing the inference cost and Bag of specials (BoS): post-processing methods that only increase the inference cost by a small amount but can significantly improve the accuracy of object detection. Table-II summarises the BoS and BoF methods adopted to enhance detection in Yolov4.
- **Neck**- is the sub-set of the backbone, which helps enhance the feature discriminability and robustness using SPP (Spatial pyramid pooling) and PANet (Path Aggregation Network). SPP significantly increases the receptive field, separates most of the significant context features, and causes almost no reduction of the network operation speed. The PANet (Path Aggregation Network) is used as a method of parameter aggregation from different backbone levels for different detector levels.
- **Head**- handles the detection part (classification and regression) of bounding boxes. A single output has 4 values describing the predicted bounding box (x, y, h, w) and the probability of k classes + 1 (one extra for background).

B. Object Tracking Module

Object tracking involves the process of tracking an object across a series of frames. We start with all possible object detections in a frame and give them an ID. In subsequent frames, we try to carry forward an object's ID. If the object has moved away from the frame then that ID is dropped. If a new object appears then they start with a fresh ID. This is important for us to remove duplication of violations count across frames. For tracking objects, we are using DeepSORT [6], which is an improved variant of kalman filter tracker [7]. Deep sort uses position-velocity-measured Kalman filter tracking to effectively handle Multiple Object Tracking issues like occlusion, distortion etc. DeepSORT consists of the following three steps:

- Yolo-v4 (Object detector) is used to perform the initial object detections per frame.
- Kalman filter-based estimation of the tracks of existing objects in the current frame. This uses the state of each track as a vector of eight quantities, that is, box center (x, y), box scale (s), box aspect ratio (a), and their derivatives with time as velocities. If there is no detection of the tracking object for a threshold for consecutive frames, it is considered to be out of frame or lost. Thus for every freshly detected box, a new track begins. Kalman filter further helps the problem of sudden occlusions.
- Finally, an association is made from the predicted states from Kalman filtering for the new detection with old object tracks in the previous frame using the Hungarian algorithm on bipartite graph matching. This is made robust by setting the weights of the matching with distance formulation.

C. License Plate Detection Module

The process of Licence Plate capturing consists of three important steps;

- **Detection:** An object detector, in our case Yolov4 is used to identify and localize the number plate and provide coordinates of the bounding box of the number plate in the image. The exact bounding box is cropped by image processing.
- **Pre-processing:** The main reason for pre-processing is to enhance the quality and readability of the cropped number plate characters before undergoing segmentation and recognition.
  1) RGB to greyscale conversion
  2) Gaussian blur for noise removal
  3) Binarization of image to improve detection quality. The adaptive Thresholding method is used and found to give better results as we calculate the threshold of the smaller region in the image rather than the global threshold.
- **Segmentation and Recognition:** Tesseract [8] is used for segmentation and character recognition. In the first step, the outlines of components are gathered together, purely by nesting, into Blobs by the Connected Component Analysis. Blobs are organized into text lines, and the lines and regions are analyzed for fixed pitch or proportional text. The lines are broken into words differently based on the kind of character spacing. Fixed pitch text is chopped immediately by character cells. Proportional text is broken into words using definite spaces or fuzzy spaces. Recognition proceeds as a two-pass process. During the

Fig. 2. Summary of the number plate detection module.

Fig. 3. Example of the process of extraction from a localised number plate.

TABLE III

VIOLATION DISTRIBUTION OF THE TRAIN DATASET

Datatype/ Class	No of Images containing the Violation	No of Instances of Violation across all Images
Not Wearing Helmet(Class 0)	3240	3888
Usage Of Phone While Riding (Class 1)	2145	2252
Triple Riding (Class 2)	1120	1344
Wheeling (Class 3)	305	306
No Parking (Class 4)	240	288

first pass, an attempt is made to recognize each word. The words that are correctly recognized are passed to an adaptive classifier as training data. As a result, the adaptive classifier gets a chance at improving results among text lower down on the page. The final Phase resolves the fuzzy spaces and checks alternative hypotheses for the x-height to locate small-cap text.

Figure 2 summarises the entire number plate detection pipeline and Figure 3 provides a visual example of the systematic process of the number plate detection after a number plate is localised, i.e. grey scaling, blurring, binarisation, segmentation, and recognition of characters from an input number plate image.

IV. EXPERIMENTAL SETUP

A. Dataset

For Violation detection, a custom dataset is prepared from live footage captured and footage of violation from various online sources. The images extracted from the footages consists of both light and dense traffic conditions, to include diversity in data. The images extracted are manually annotated for various types of violations. A total of 5800(train) and 1210 (test) images are labeled across the following violation categories *Not wearing a helmet, Usage of a phone while riding, Triple riding, Wheeling, and No parking*. Table-III depicts the distribution of the dataset and the violations distribution across the training dataset.

For Number plate detection, a relevant annotated dataset is obtained from Kaggle [5] which consists of 11271 images are used from the dataset. Further, Data Augmentation techniques like horizontal flipping, cropping, etc. are used to boost the data size.

Fig. 4. Sample from annotated dataset of violations.

From Figure 4, it can be seen that (i) There are multi violators with similar violations like No helmet in the Upper middle image. (ii) There can be a single Violator with Multi violations like upper left and lower middle and left images. Or, a single type of violations per image like the upper left and lower right images. Every image is annotated to capture all violations arising in each frame of image.

B. Evaluation Tasks

We evaluate the system across following tasks:

1) Violations Detections across all possible Violations on the test dataset.

2) Number Plate Detection across all detected violations frame obtained from the violation module.

3) The overall pipeline is tested on 8 video feeds with light and dense traffic conditions to test its capability and robustness.

4) Benchmark test with state of the art algorithms.

V. RESULTS AND ANALYSIS

The experimental results were conducted on a machine with a memory of 16GB with an Intel core i5-8250U CPU@2.60GHZ X 8 processor with an INTEL UHD Graphics 620 with GNOME 3.28.2 and 64-bit OS with a disk of 320 GB. The programs are written in Python3.7.3 with the help of various libraries like OpenCV, NumPy and Darknet framework. The setting for the DeepSORT is set as the following: resize images to- 416; iou threshold - 0.45; confidence threshold- 0.50; and all other settings is per the original paper defaults [6]. The preprocessor is set to capture only 20 frames per second by frame skipping. The video for the live testing is captured using *Canon XF605 UHD 4K HDR Pro Camcorder*, which can record Ultra HD 4k resolution videos at 60 frames per second. The camera setup is made strategically to effectively capture live footage without hindrance at 12 feet from ground level and relay the same to the computer for real time predictions.

A. Violation Detection & Tracking

Violation detection and tracking were performed using the YOLO-v4 + DeepSORT algorithm, which was trained on 29000 Images (5800 images with data augmentation) consisting of all five violation classes. This module is evaluated with 1210 test images. The results containing correct prediction and wrong prediction for each type of violation are presented in Table-IV. It is noted that the mean average precision (mAP) for violations detection is 98.09% for all classes. Further, it can be seen that there were 0 violations missed out for Triple-riding and Wheeling violations. Overall, the predictions show good accuracy for all classes. Figure 5 shows ROC curve across all the classes of detection. Further:

TABLE IV
RESULT OF VIOLATION DETECTION ON TEST DATASET

Violation	No Of Actual Violations	True Positive	False Positive	Precision
Not Wearing Helmet	1045	1041	13	0.987
Usage Of Phone While Riding	802	788	22	0.973
Triple Riding	165	165	1	0.994
Wheeling	45	45	0	1.000
No Parking	21	19	1	0.950
Average				0.981

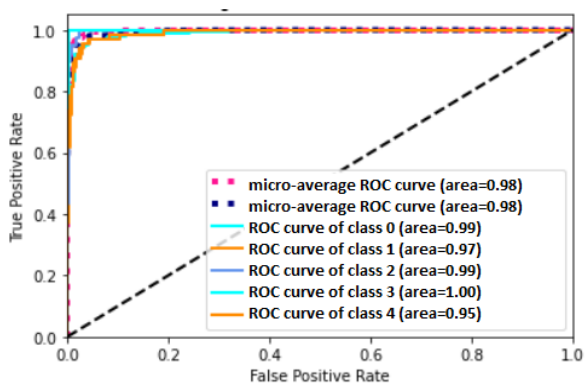

Fig. 5. ROC curve for class detections.

- The performance is relatively good for no Helmet and usage of phone while riding as YOLO is pretrained mobile and helmet images and helps capture these helmets and mobiles with high accuracy.
- The performance is high for Triple riding detection as Yolo-v4 is an excellent object detector of persons and vehicles. Thus 3 persons and a vehicle in single bounding box is easy to capture for the model.
- The performance is best for wheeling violation as its a very distinct type of offence with only a wheel on the ground.
- The performance is relatively less on No parking detection as it trained on comparatively lesser images and the model doesn't detect violations when the vehicle is 5+ feet away from a sign board as it exceeds a particular bounding box size.

The performance is better than most of literature performance [9]–[15] as Yolo-v4 is a faster and better object detector in contrast to the ones previously used for violation detection. There is no direct comparison available as no literature had conducted test for all of the above classes.

B. Number Plate Detection

The 2058 predicted True violations images from the violation detection module are passed to the Number plate detection module. Not all the violators' vehicle number plates are visible or readable, mainly due to camera angles and occlusions. The results of the Number plate detection module are given in Table-V. It can be seen that out of 1853 readable number plates, 1842 was correctly detected and recognized which gives an accuracy of 99.41%. Thus showing the effectiveness of using an object detector like Yolo-v4 for the detection of number plates. The performance is very similar to literature performance [16]–[18] as its a pure single class detection and processing.

C. Live Video

Eight high-resolution videos of roughly 1 minute each of various frames-per-second are used. Each video has varying traffic conditions.

TABLE V
RESULT OF NUMBER PLATE DETECTION ON TEST DATASET

	Number samples	Number plates visible/readable	Correct Prediction	Success Rate (%)
Detection	2058	1853	1849	0.9978
Segmentation	1849	1845	1843	0.9989
Recognition	1843	1843	1842	0.9995

TABLE VI
RESULTS OF VIOLATION AND NUMBER PLATE DETECTION ON VIDEO FEED

Video	Frames per second	Actual Viola-tions	Detected Viola-tions	Violators-Number plate detected	Error in violation detection %	Error in number plate detection %
Video 1	24	5	5	5	0	0.00
Video 2	24	13	11	10	15.38	9.09
Video 3	24	22	20	18	9.09	10.00
Video 4	36	1	1	1	0	0.00
Video 5	34	3	3	3	0	0.00
Video 6	44	13	11	11	15.38	0.00
Video 7	43	14	14	13	0	7.14
Video 8	56	22	18	13	13.64	11.11
Total		**93**	**83**	**79**		

The results for the video feed are presented in Table-VI. It is noted that 83 out of 93 violations were accurately captured resulting in a accuracy of 90.32%. In addition, it should be noted that the proposed system gave no false positives which meant wrong violations was tagged. On analysing the false negatives it can be noted that most of the non detections are mainly due to usage of caps or fancy head gear which the model tags as helmets and hence doesn't capture. Also most violations of usage of mobiles go undetected as the angle of capture is not able to provide a label of high accuracy.

Further, 79 out of 83 number plates was accurately captured giving the accuracy of the Number plate detection module to be 94.04%. This is very much in line with other State-of-the-Art-Models [16]–[18]. This is because the same benchmark dataset is used for training and most of literature's detector models work on similar deep learning networks. Finally, 79 out of 93 violations was captured making the system 84.94% accurate which is very high when considering there is no human intervention until this stage and 0% false positive triggers. There is no State-of-the-Art-Model for comparison of the pipeline as there are no similar work for two wheeler violation and ticketing in real-time. Also the video relay was optimally processed using frame skipping technique and image compression before processing hence, its not logical to compare with other State-of-the-Art-Models which haven't performed the same.

D. Benchmark Analysis

Finally, we test our system against State-of-the-Art models in literature both in image and real time scenarios. We first test our proposed system against similar deep learning frameworks ([9], [13]). We use a marginal dataset containing only 2 types of violations (no helmets and triple riding) [19]. The dataset has 16204 instances of violations. The results for benchmarking of violations are presented in Table-VII. It is noted that the proposed model out performs [9] and [13]. Upon analyzing, it can be noted that [9] fails majorly in segmenting the images with multiple violations. Plus, 3 algorithms running simultaneously makes this framework computationally expensive. The proposed system beats [13] which is a similar framework model as Yolov4 incorporates BoS, BoF SPP etc. for enhanced detection.

We further integrate the violation detector of [13] and the number plate detection of [18] to build a competitor system for our proposed model. The reason for use of [18] is the fact it records the best

TABLE VII

VIOLATION DETECTION- BENCHMARK ANALYSIS

	Accuracy samples	Precision visible/readable	Recall Prediction	Time per detection Rate (%)
Aniruddha et al.[9]	0.8955	0.8883	0.9208	1.08 sec
Mallela et al.[13]	0.9174	0.9139	0.9415	.22 sec
Proposed Methodology	0.9874	0.9886	0.9890	0.09 sec

TABLE VIII

VIOLATION DETECTION- BENCHMARK ANALYSIS

	Total Violations	Violations detected	Total number plates detected	Success Rate (%)	Total Time
Mallela et al.[13] + Singh et al.[18]	23	14	6	0.2609	345 seconds
Proposed Methodology	23	21	17	0.7391	198 seconds

performance in literature and has a similar framework to our proposed framework. The results for a live video of 188 seconds at 28 frames per second and 640*320 resolution are presented in Table-VIII. It can be clearly seen that the proposed frameworks have both computational and performance superiority. The speed gain is due to: (I) The violations and number plate detector uses the same detector. (ii) Efficient frame skipping adopted. (iii) The proposed preprocessor auto crops the violator's vehicle and passed the violation for detection, which enables more precision in segmentation and detection of number plate. Additionally the [13] fails in real-time applications as it does not incorporate an object tracker and hence re-counts most detections.

VI. CONCLUSION

In this paper, we present an end-to-end system to automate two-wheeler violation detection to have zero or Minimum human intervention for ticketing. We propose the use of trained Yolo-v4 + DeepSORT for violation detection and tracking and Yolo-v4 + Tesseract for number plate detection and recognition. This implementation obtained a mean average precision (mAP) of 98.09% for violation detection and accuracy of 99.41% for number plate detection on test data. Further when tested, on eight live feeds for real-life scenarios, the system detected 77 out of 93 violations with zero False Positive detection. The system was able to detect violations, capture the number plate of the violators efficiently, and write the same to a database with all relevant data to enable automatic ticketing of violations. Thus, strict regulations of traffic rule violations can be implemented using AI-based systems to improve road safety standards and bring awareness among vehicle users.

There are several challenges in building a very robust violation model. Custom model training requires a large amount of labelled data. The task involves creating, gathering, and labelling many proprietary data for the task, which is very time consuming and human resource intensive. In addition, there are issues of good quality images, as most CCTV footages are of low resolution making most data useless. Another challenge observed: inconsistent number plates with fancy fonts, local language number plates, and at times no number plates present in vehicles. These are a few of the challenges in addition to the large computational resource required for efficiently running the system.

For future work, the system can be improved by including other types of road vehicles and detecting corresponding vehicle-related traffic violations to make the system, more robust and complete. Also, architectures can be built to handle multiple camera views and to tackle more violation types. The OCR for license plate detection can be further trained to capture number plates having non-traditional fonts and designs. Additionally, newer object detection models can be experimented with to benchmark models for accuracy and speed.

ACKNOWLEDGMENT

Project repository: https://github.com/ShreeCharranR/Two-Wheeler-Vehicle-Traffic-Violations-Detection-and-Automated-Ticketing-for-Indian-Road-Scenario.

REFERENCES

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

[2] [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

[3] [Online]. Available: https://morth.nic.in/sites/default/files/RA_Uploading.pdf

[4] [Online]. Available: https://epaper.timesgroup.com/Olive/ODN/TimesOfIndia/shared/ShowArti cle.aspx?doc=TOIDE%2F2019%2F09%2F08&entity=Ar01219&sk=00765E9C&mode=text#

[5] [Online]. Available: https://www.kaggle.com/rajaman/vehicle-licence-plate

[6] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649, doi: 10.1109/ICIP.2017.8296962.

[7] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3464–3468, doi: 10.1109/ICIP.2016.7533003.

[8] R. Smith, "An overview of the Tesseract OCR engine," in *Proc. 9th Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2007, pp. 629–633, doi: 10.1109/ICDAR.2007.4376991.

[9] A. Tonge, S. Chandak, R. Khiste, U. Khan, and L. A. Bewoor, "Traffic rules violation detection using deep learning," in *Proc. 4th Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA)*, Nov. 2020, pp. 1250–1257.

[10] J. Chiverton, "Helmet presence classification with motorcycle detection and tracking," *IET Intell. Transp. Syst.*, vol. 6, no. 3, pp. 259–269, 2012.

[11] K. Dahiya, D. Singh, and C. K. Mohan, "Automatic detection of bike-riders without helmet using surveillance videos in real-time," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 3046–3051.

[12] X. Wang, L.-M. Meng, B. Zhang, J. Lu, and K.-L. Du, "A video-based traffic violation detection system," in *Proc. Int. Conf. Mech. Sci., Electr. Eng. Comput. (MEC)*, Dec. 2013, pp. 1191–1194.

[13] N. C. Mallela, R. Volety, R. P. Srinivasa, and R. K. Nadesh, "Detection of the triple riding and speed violation on two-wheelers using deep learning algorithms," *Multimedia Tools Appl.*, vol. 80, no. 6, pp. 8175–8187, Mar. 2021.

[14] B. Y. Reddy and M. Budka, "Detection of motor bicyclist violating traffic rules using computational neural networks," Tech. Rep.

[15] V. Mandal and Y. Adu-Gyamfi, "Object detection and tracking algorithms for vehicle counting: A comparative analysis," *J. Big Data Anal. Transp.*, vol. 2, pp. 251–261, Nov. 2020.

[16] S. Kumari, D. K. Gupta, and R. M. Singh, "A novel methodology for vehicle number plate recognition using artificial neural network," in *Proc. 3rd Int. Symp. Comput. Vis. Internet*, Sep. 2016, pp. 110–114.

[17] H. Erdinc Kocer and K. Kursat Cevik, "Artificial neural networks based vehicle license plate recognition," *Proc. Comput. Sci.*, vol. 3, pp. 1033–1037, Jan. 2011.

[18] J. Singh and B. Bhushan, "Real time Indian license plate detection using deep neural networks and optical character recognition using LSTM tesseract," in *Proc. Int. Conf. Comput., Commun., Intell. Syst. (ICCCIS)*, Oct. 2019, pp. 347–352.

[19] [Online]. Available: https://www.kaggle.com/datasets/meliodassourav/traffic-violation-dataset-v3