YOLOv3-Based Traffic Infraction Recognition With OCR Precision

Mrs. Gowri Vidhya N[1], Mr. Vishwa R[2], Mr. Viswanathan N[3]
[1]Assistant Professor, Dept of Information Technology
[2, 3]Dept of Information Technology
[1, 2, 3] St. Joseph's Institute of Technology, OMR, Chennai-600019

Abstract- *An increasing number of applications for traffic violations have been made in several countries in recent years. By examining vehicle speeds close to road violation detection stations on major metropolitan road-ways, the current study aims to ascertain the effect of road traffic violation detection on vehicle speeds. Using the camera method, traffic was captured on a sample segment of an urban highway. Vehicle speeds were measured by the estimate approach prior to, during, and following a road traffic control zone. Using statistical methodology and SPSS software, speed data were cate-gorised and processed to create probabilistic density models of vehicle speeds prior to, during, and following a traffic violation checkpoint.The results demonstrate that the trafficenforcement area's maximum and average speeds are significantly slower than those in the areas before and after it. 80.2% of cars were speeding outside the checkpoint violation area, while 70.1% of cars were speeding before the infraction was noticed.At the same time, speeding in the checkpoint decreased to 15.9%. When vehicles pass through a checkpoint for road traffic violations, the speed of the vehicle decreases and in-creases. The detection of violations in road traffic at its active location can effectively regulate driving behavior and reduce speed, but these effects are limited to the de-tection of traffic violations. Vehicle speed distributions can be calculated from vehicles the potential velocity of the density model. It allows, the identification of traffic signal violations in real time. To make it simple for the user to operate the system, monitor traffic, and take en-forcement action against breaches of traffic rules, a user-friendly graphical interface is included.*

Keywords- Computer Vision, OCR Precision, Object Detection, YOLOv3, License Plate Recognition, Deep Learning, Real-time Recognition.

I. INTRODUCTION

The flow of cars on a road is referred to as traffic. Its volume, density, and speedcan all change. Reducing travel times and minimising congestion require an efficient flow of traffic. Land use planning and urban development are influenced by transportation infrastructure, which includes road networks, public transportation systems, and traffic management. To build livable, effective, and sustainable cities, well-thought-out traffic systems are necessary.City planners and traffic authorities can more effectively manage traffic flow with the use of traffic detection systems. To make decisions about the timing of traffic signals and lane management in real-time, they can keep an eye on the number of vehicles on the road, their speed, and other pertinent data. This can lessen traffic jams, enhance traffic flow, and cut down on commuter delays. Compared to manual traffic management, these systems have a number of important advantages. When a traffic control and management system is manual, it means that human operators are used to guide and control traffic. This method is predicated on the use of flaggers or traffic police officers who manually regulate traffic and pedestrian flow at crosswalks, intersections, construction zones, and other locations where traffic control is necessary. which depends on human operators is prone to error or the omission of crucial information, whereas automated systems are intended to be more consistent and dependable when gathering data and making decisions.

II. RELATED WORKS

For a variety of uses, the system combined OCR (Optical Character Recognition) and YOLOv3 (You Only Look Once version 3). Currently, there is a lot of work going on to detect traffic parameters in the research field.

[1] Inductive Loop Sensors are electromagnetic sensors set into the pavement at intersections and all along the length of roads. By monitoring changes in the loop's inductance upon a vehicle passing over it, they are able to detect the presence of moving vehicles. Common applications for inductive loop sensors include counting vehicles and controlling traffic signals.

[2] Video-Based Detection: Real-time video footage is captured by cameras positioned at intersections and along roads; this footage can be analysed to track and identify vehicles, pedestrians, and traffic flow. In video-based systems, sophisticated machine learning and

computer vision algorithms are frequently utilised for tracking and detecting vehicles.

[3] Radar-Based Systems: Radio waves are used by radar sensors to determine the location and velocity of moving vehicles. They are frequently employed for adaptive traffic signal control and traffic monitoring. Radar sensors can operate in various weather conditions and provide reliable data.

[4] Lidar-Based Systems: Lidar (Light Detection and Ranging) sensors measure distances and produce a sophisticated three-dimensional map of their immediate environment by using laser light. They are employed in many different contexts, such as autonomous car collision avoidance and traffic monitoring.

[5] Ultrasonic Sensors: These devices produce sound waves and time how long it takes for the waves to return after striking objects like automobiles. These sensors have been used in toll booths and parking lots to detect the presence of vehicles.

[6] Licence Plate Recognition (LPR) Systems: OCR technology is used by LPR systems to read and identify licence plates on automobiles. They are frequently employed in the fields of law enforcement, toll collection, and parking management.

[7] Bluetooth and Wi-Fi-Based Detection: Some systems use Bluetooth or Wi-Fi signals emitted by mobile devices in vehicles to track traffic flow and travel times. This technology is often used for monitoring traffic on highways and urban roads.

III. METHODOLOGY

A. Vehicle Classification - Finding and identifying cars within an image or video stream is the task of vehicle detection in computer vision. Vehicle detection can be accomplished in a variety of ways. One well-liked technique is the application of deep learning models, like YOLO (You Only Look Once). Because of its high accuracy, scalability, and real-time object detection capabilities, YOLOv3 is an excellent choice for vehicle classification in traffic systems.In the classic YOLOv3, nine prior sizes are clustered out based on all the actual sample data in the label using the K-means clustering technique. These clusters are then used for subsequent regression reasoning. The input picture (e.g., RGB, grayscale, etc.) is divided by the network into a S×S grid (e.g., 13×13). A given grid is in charge of anticipating the item if the foreground target's ground truth lies on it. In the output layer, every grid in the picture will produce a certain number of predicted bounding boxes that match the given item, together with their confidence ratings and class probabilities.Videos are offered for the categorization of vehicles. The following factors are involved in processing videos:

1) *The angle and placement of the camera:*Make sure that the video's camera is positioned correctly to capture desired traffic scenes. Take into account variables like field of view, height, and camera angle to maximise the detection model's performance.

2) *Lighting:*Similar to other computer vision models, Yolov3 is susceptible to variations in lighting. Make sure there's enough lighting to take distinct and clear pictures of cars, and that the video quality is high.

3) *The Variability of Traffic:*There are many different types of traffic scenarios, such as clogged roads, diverse car types, and fluctuating speeds. It is advantageous to practise or hone your YOLOv3 model using a dataset that reflects the precise traffic circumstances you anticipate.

4) *Custom Training:*Depending on your specific requirements, you may need to train YOLOv3 on a custom dataset that includes images or video frames from your traffic system. This helps the model adapt to the unique characteristics of your environment.

How YOLOv3 Operates:

5) *Grid Organisation:*The input image is divided into a grid by YOLOv3, and each grid cell is in charge of making an object prediction. YOLOv3 is able to make predictions at various scales thanks to this grid system.

6) *Bounding Box Predictions:*YOLOv3forecasts bounding boxes, confidence scores, and class probabilities for every grid cell. This indicates that each grid cell is searching for and locating objects, and that a single object may be detected by more than one cell.

7) *Different Scales:*Within the network, YOLOv3 functions on multiple scales, enabling it to detect objects of different sizes. Utilising several detection layers at various scales allows for this.

8) *Anchor Cases:* Anchor boxes are used by YOLOv3 to enhance bounding box predictions. Bounding box sizes are present for anchor boxes.

This is accomplished by using the vehicle's center x and y coordinates to create a mini-bounding box that, when the vehicle approaches the camera, always stays in the center of the vehicle and does not represent any empty space around it.

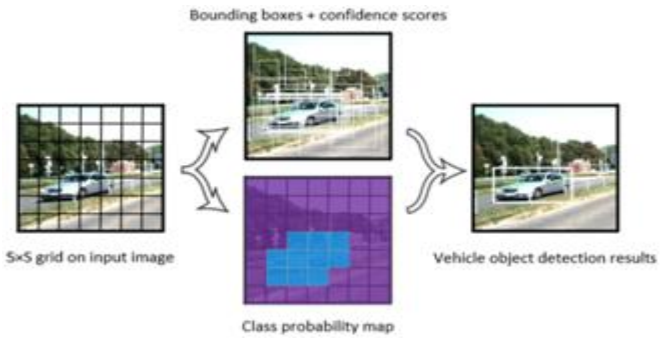

Fig. 1: Schematic diagram of the original YOLOv3 detection method

***B. Violation Detection* -** The YOLOv3 model is used to detect the cars. The automobiles are found, and then the violation cases are examined. In the user-provided video footage preview, a traffic line is depicted across the road. The line indicates that it is a red traffic light. Any car that crosses the traffic line in a red state is in violation. Creating of those 2 reference points/lines and append their coordinates points are as follows:

```
"poly_co_finish = [] #line 88
blanked = np.zeros((1024,2048), dtype=np.uint8)
pts_finish = np.array(([634,800], [1279,638], [1330,641],
[722,851]))
cv2.fillPoly(blanked, np.int32([pts_finish]), 255)
#print(np.where(blanked == 255))
x_cord_finish = np.where(blanked == 255)[1]
y_cord_finish = np.where(blanked == 255)[0]
for q in range(0, len(x_cord_finish)):
poly_co_finish.append((x_cord_finish[q],y_cord_finish[q]))"
```

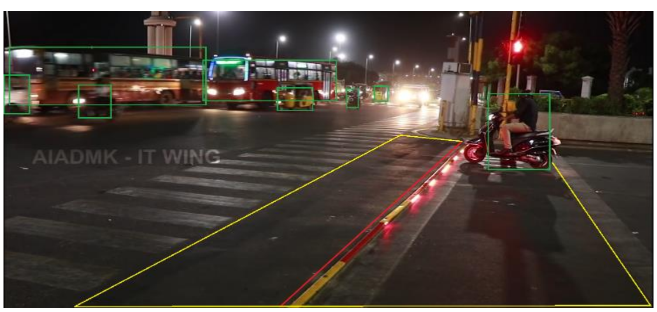

Fig. 2: Violation line estimator and violation region

Initially, I made two empty arrays called poly_co_start and poly_co_finish to which I will append the pixel coordinates of the two reference points. Here, I've chosen a stripe of pixel coordinates as my reference line rather than a single line since, in some cases, too thin reference lines prevent the intersection from happening. As a result, there is no intersection as the vehicle crosses the reference line. This System consists of two components. They are:

1) *Red Light Violation :*The red light violation detection module's main goal is to find and document instances of cars disobeying red traffic signals at intersections. Any car that runs a red traffic light is in violation. The bounding box surrounding the car becomes red when a violation is detected.

Procedure Applied:

a. The YOLOv3 model is used to detect the vehicles.
b. The vehicles are found, and then the violation cases are examined.
c. In the user-provided video footage preview, a traffic queue is depicted across the road.
d. The line indicates that it is a red traffic light.
e. Any vehicle that crosses the traffic line in a red state is in violation.
f. The bounding box of the detected objects is green.
g. Any vehicle that runs a red traffic light is in violation.
h. The bounding box surrounding the car turns red when a violation is detected.

If a vehicle crosses the zebra line while the traffic light is red, a violation must be detected. So the first step is to save the coordinates of the zebra crossings and traffic signals and draw a bounding box with their names as labels. The next phase is to check for traffic light infractions whenever a vehicle crosses the zebra line while the traffic signal is red. It was difficult to avoid the vehicles visible on video on the other side of the zebra crossing throughout the violation detection process.

Fig. 3: Before Violation

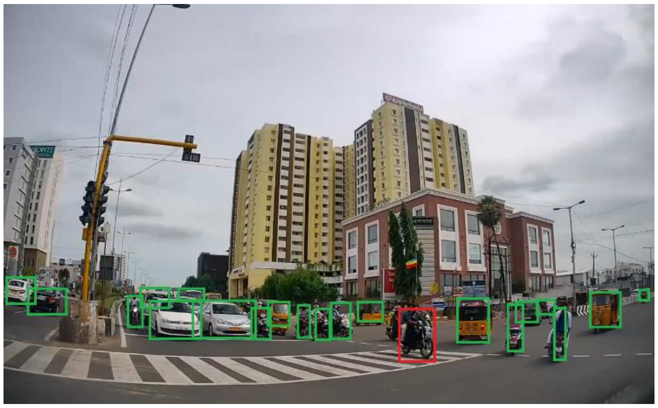

Fig. 4: After Violation

Processing Time Evaluation:

The following table presents the average processing time for the metrics mentioned above following the testing of the input videos:

Measurement	Average Value (Seconds)
Traffic Light Localization	0.267
Violation Line Approximation	1.933
Tracking and Violating Detection per Frame	0.294

2) *Speed Violation :*The Speed Violation Detection module aims to identify vehicles that exceed the posted speed limits on roads and highways, promoting safe driving practices. The module calculates vehicle speeds by measuring the time it takes for a vehicle to travel between two points. It then compares these speeds to the posted limits and flags violations.Our planned application will make it easier to abide by traffic laws and regulations, particularly those pertaining to instances of speeding. It lowers the number of incidents of speeding. These days, a lot of accidents occur as a result of drivers exceeding the speed limit. This causes many skidding incidents and accidents on roads, in steep valleys, at abrupt turns, and in mountains.Here, we're utilizing a variety of Python packages to estimate the speed of the car. Specifically, we are employing the dlib, haarcascade, tesseract, and openCv frameworks. In order to predict speed using the formulas we have defined in our code, we must first determine the width and height of the road in this system. Using dlib, which assigns a unique id to each car so we can identify them individually, we can determine the speeds of many vehicles that are present in the frame.

Fig. 5: Speed Detection

3) *Number Plate Detection:* It focuses on locating and recognising number plates inside the cars that are found. The identification and collection of number plate data from moving vehicles is the primary function of the Number Plate Detection module. We have given a specific co-ordinate and images so that it can extract that specified coordinated image of plate number.This module detects and recognises licence plates on automobiles using image processing and optical character recognition (OCR) techniques.Utilising OCR algorithms facilitates the identification of characters contained in the divided areas of the licence plate. OCR software recognises and decodes characters using machine learning and pattern recognition techniques.

Fig. 6: Number Plate Detection using OCR

C. User Interface - It gives system administrators and operators a graphical user interface (GUI) through which they can communicate with the traffic signal violation detection system. The standard programming language Python GUI (Graphical User Interface) toolkit is called Tkinter. It is frequently employed in the design and creation of graphical desktop applications. An intuitive user interface for a traffic management system can be created with Tkinter's array of widgets and customization options.

IV. CONCLUSION

Given that traffic laws and the quantity of vehicles on the road vary depending on the location and time of day, it might be difficult to identify infractions using video surveillance. The YOLOv3 algorithm is suggested to be appropriate for traffic infraction identification in this research. The findings demonstrate that it is possible to identify several traffic infractions from a single input source. The system's accuracy in detecting vehicle counts is 97.67%, and its accuracy in detecting vehicle speeds is 89.24%. For high-density traffic flows, the detection time is shorter. Thus, the traffic density affects the system's operating speed.Consequently, this can not only help the Traffic Police Department but also result in significant cost savings on human resources. This money might be put toward improving people's lives or increasing the number of cameras on the road to increase security. People would start adhering to traffic laws and wearing helmets out of fear of being punished if the current manual detection system were to be replaced with automatic detection.

Future studies on the implementation of the developed algorithm for additional sophisticated image processing methods. Since this could shorten the system's program runtime by ignoring other pointless steps carried out by the background difference approach. Alternatively, a computer vision algorithm could be implemented to enhance the system's intelligence. In order to strengthen this system, we intend to add more real life features, privacy protection, mobile app integration in the future.

REFERENCES

[1] B. Srilekha, K. V. D. Kiran and VenkataVaraPrasad Padyala, "Detection of License Plate Numbers and Identification using Yolo v2 and OCR Method", 2022 International Conference on Electronics and Renewable Systems (ICEARS), 2022.

[2] ShelkeNilesh, JadhavSamiksha, DoifodeMitali, UmateYamini, PatilRinki and HarinkhedeNilam, A Review on Identification of Number Plate and Wrong Way Vehicles Detection, 19 February 2023.

[3] Wang Liang, Li Lingmin, Wang Hao, Zhu Shaohua, ZhaiZhiqiang and Zhu Zhongxiang, Real-time vehicle-identification using improved YOLO v4 , March 2022.

[4] R. J. Franklin and Mohana, "Traffic Signal Violation Detection using Artificial Intelligence and Deep Learning", 2022 5th International Conference on Communication and Electronics Systems (ICCES), pp. 839-844, 2022.

[5] AljelawyQudes Mb and M. Salman Tariq, License plate-recognition in slow motion vehicles, August 2023.

[6] SwetaKumari, Leeza Gupta and PrernaGupta, "Automatic license plate recognition using open cv and neural network", International Journal of Computer Science Trends and Technology (IJCST), vol. 5, no. 3, 2023.

[7] SheerazMemon et al., "A video based vehicle detection counting and classification system", International Journal of Image Graphics and Signal Processing, vol. 10, no. 9, pp. 34-41, September 2022.

[8] Huang B, "An improved YOLOv3-tiny algorithm for vehicle detection in natural scenes", IET Cyber-Syst. Robot., vol. 3, no. 6, pp. 256-264.

[9] KarthikSrivathsa and D Sand Kamalraj R, "Vehicle detection and counting of a vehicle using opencv", International Research Journal of Modernization in Engineering Technology and Science, vol. 03, no. 05, May 2023.

[10] M. B. Natafgi, M. Osman, A. S. Haidar and L. Hamandi, "Smart traffic light system using machine learning", 2022 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET), pp. 1-6, 2022.

[11] D. Snegireva and G. Kataev, "Vehicle Classification Application on Video Using Yolov5 Architecture", 2021 International Russian Automation Conference (RusAutoCon), pp. 1008-1013, 2021.

[12] M. Udoy, A. Rahman and M. Chowdhury, Real-Time Traffic Vehicle Detection in Bangladesh Using Yolo, 2021.

[13] M.-A. Andrei, C.-A. Boiangiu, N. Tarba and M.-L. Voncil, "Robust Lane Detection and Tracking Algorithm for Steering Assist Systems", Machines 2022, vol. 10, no. 10, 2022.

[14] F. A. Arnob, A. Fuad, A. T. Nizam, S. Barua, A. A. Choudhury and M. Islam, "An Intelligent Traffic System for Detecting Lane Based Rule Violation", 2019 International Conference on Advances in the Emerging Computing Technologies (AECT), pp. 1-6, 2020.

[15] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh and I.-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN", 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Jun. 2020.

[16] K. Liu, H. Tang, S. He, Q. Yu, Y. Xiong and N. Wang, "Performance Validation of Yolo Variants for Object Detection", Proceedings of the 2021 International Conference on Bioinformatics and Intelligent Computing (BIC 2021), pp. 239-243, 2021.

[17] M. M. Bachtiar, A. Mawardi and A. R. A. Besari, "Vehicle Classification and Violation Detection on Traffic Light Area using BLOB and Mean-Shift Tracking

Method", 2020 International Conference on Applied Science and Technology (iCAST), pp. 94-98, 2020.

[18] Zillur Rahman, Amit Mazumder and Md. A. Ullah, "A real time wrong way vehicle detection based on YOLO and centroid tracking", IEEE Region 10 Symposium (TENSYMP), pp. 5-7, June 2020.

[19] J. Shashirangana, H. Padmasiri, D. Meedeniya and C. Perera, "Automated License Plate Recognition: A Survey on Methods and Techniques", IEEE Access, vol. 9, pp. 11203-11225, 2021.

[20] S. Alghyaline, "Real-time Jordanian license plate recognition using deep learning", Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 6, pp. 2601-2609, 2022.

[21] R. Shashidhar, A. S. Manjunath, R. S. Kumar, M. Roopa and S. B. Puneeth, "Vehicle Number Plate Detection and Recognition using YOLO-V3 and OCR Method", 2021 IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC), pp. 1-5, December 2021.

[22] G.-W. Chen, C. Yang and T.-U. Ik, "Real-time license plate recognition and vehicle tracking system based on deep learning", 2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS), pp. 378-381, 2021.

[23] C. Henry, S. Y. Ahn and S.-W. Lee, "Multinational license plate recognition using generalized character sequence detection", IEEE Access, vol. 8, pp. 35 185-35 199, 2020.

[24] Dr. Yeresime Suresh, Ankitha R, ChillaraAnusha, Dharani C and Aniketh K, "Traffic Rules Violation Detection System", September 2022.

[25] S. Bose, C. D. Ramesh and M. H. Kolekar, "Vehicle classification and counting for traffic video monitoring using YOLO-v3", Proc. Int. Conf. Connected Syst. Intell. (CSI), pp. 1-8, Aug. 2022.

[26] A. Soni and A. P. Singh, "Automatic motorcyclist helmet rule violation detection using tensorflow&Keras in OpenCV", Proc. IEEE Int. Students’ Conf. Electr. Electron. Comput. Sci. (SCEECS), pp. 1-5, Feb. 2020.

[27] K. Lavingia, M. Vaja, P. Chaturvedi and A. Lavingia, "Machine learning based approach for traffic rule violation detection", Proc. IEEE 7th Int. Conf. Recent Adv. Innov. Eng. (ICRAIE), vol. 7, pp. 244-249, Dec. 2022.