JPTM PUBLICATIONS

Emerging Trends in Science & Technology Vol-I
eISBN : 978-81-999970-1-1 | ISBN : 978-81-999970-0-4
https://doi.org/10.67389/jptm.0626.etstv1.oa5

Real-Time Vehicle RC Expiry Detection and Smart Challan System Using YOLOv8 and OCR

Saurabh Mallik[1], Shivangi Sharma[2], Mohd Tajammul[3]

[1]*Computer Science and Application, Sharda University, Gr. Noida, Uttar Pradesh, India*
[1]saurabhmallik07@gmail.com

[2] *Computer Science and Application, Sharda University, Gr. Noida, Uttar Pradesh, India*
[2]shivangi.sharma2603@gmail.com

[3]*Computer Science and Application, Sharda University, Gr. Noida, Uttar Pradesh, India*
[3]mohd.tajammul@sharda.ac.in

ARTICLE INFO

Article history:
Received 28 April 2026
Accepted 03 May 2026
Available online 12 July 2026

Keywords:
Computer Vision,
YOLOv8,
Optical Character Recognition (OCR),
Automatic Number Plate Recognition (ANPR),
Smart Challan System,
Intelligent Transportation Systems,
Flask,
Vehicle Monitoring

ABSTRACT

With urban traffic levels reaching unprecedented levels, manual RC verification has become impractical, being both time-consuming and highly unreliable in its results. This work presents a Real-Time Vehicle Registration Certificate (RC) Expiry Detection and Smart Challan Generation System based on automated verification via computer vision. Webcam frames serve as input data for the proposed solution which uses a YOLOv8 detector to detect and crop the area occupied by the vehicle's number plate. Next, the cropped image undergoes preprocessing, including a conversion to grayscale, Gaussian filtering, and adaptive thresholding. After that, the detected number plate is recognized using the Tesseract OCR engine. The obtained alphanumeric value is used to compare the vehicle against an SQLite database with vehicles' identification numbers and dates of their RC expiration. Upon successful detection of an expired RC, the system generates a PDF document (challan) with the relevant vehicle information, timestamp of the violation, exact location of occurrence, and fine amount automatically. The pipeline itself is exposed through a Flask-based web interface allowing operators to monitor detection and enforcement events in real-time. Extensive testing proved that the system has good performance in terms of number plate recognition efficiency, minimal processing latency, and consistent challan generation.

1. INTRODUCTION

With the number of vehicles plying urban roads increasing every day, there is an urgent need for improved traffic management and enforcement methods considering the limited capabilities of traditional surveillance mechanisms due to their vulnerability to human operator error [1]. To mitigate this challenge, ITS, with IoT devices and computer vision technology at the center of these solutions, have been gaining popularity lately as they can help keep track of vehicle activity, provide passenger safety and efficiency in transport systems [2]. Among the technologies that have contributed to these innovations, ANPR stands out with the ability to transform images of vehicles into text [3].

As for ANPRs, initially, classical computer vision was used to detect number plates, using quite simple methods like edge detection, morphological transformations, and even recognition through templates using the principle of correlation factor [4]. While this technique proved to be relatively efficient in terms of computation, it lacked robustness under various challenging circumstances. As far as more recent advancements in Deep Learning are concerned, it can be stated that it has changed the entire face of ANPR, resulting in much better accuracy in difficult conditions, especially during low illumination or unfavorable weather like snowfall [5]. Modern ANPR software relies heavily on neural networks; specifically, popular YOLO architectures are utilized in the process of Localization of plates [6]. Moreover, OCR engines (Tesseract or EasyOCR) are used for the identification of the alphanumeric symbols present on the plate [7]. In this regard, one should note that this operation becomes especially important when the ANPR deals with license

 ISBN : 978-81-999970-0-4

plates in a format different from standard conventions, since accuracy here cannot be underestimated [6], [8].
However, the need for manual verification in checking the presence of critical documents on the vehicle continues, making the process slow and error-prone [9]. Ensuring the presence of the Registration Certificate (RC), insurance, and Pollution Under Control (PUC) documents on the vehicle during its journey in the public road is critical in preventing fraudulent documents and ensuring compliance with the rules and regulations [10]. The existing e-challan smart system has incorporated the use of automation in verifying the details of the offenders by using web crawlers to access the regional RTO databases [8]. However, there still lacks an automatic process for RC expiry detection in real time during traffic [11].
In this context, this paper attempts to fill this research gap through the introduction of the Real-Time Vehicle RC Expiry Detection and Smart Challan System with the objective of increasing automation in traffic-rule enforcement [8]. The proposed framework will simulate a local traffic surveillance system by processing real-time feeds from webcams using the YOLOv8 algorithm [7]. The number plate region is first subjected to a series of pre-processing morphological operations including grayscale, bilateral filter, and adaptive threshold before being fed into Tesseract OCR software for reading the registration number [8]. The output string then forms the basis for comparison with a local database of vehicles to check the validity of the RCs, which reflects the basic functions of centralized transportation databases [9]. Once an expired RC is detected, a PDF challan is automatically generated with the fine calculated, thus minimizing the manual effort for this task [1], [11].

2. LITERATURE REVIEW

2.1 Traditional Image Processing Approach in ANPR

In early ANPR systems, traditional image processing techniques were used extensively with a combination of structured localization and recognition of number plates. One research approach made use of relatively simple algorithms involving grayscale transformation, Canny edge detection, and morphological processes like dilation and erosion to achieve a recognition accuracy rate of around 92% using template matching technique [4]. Simultaneous efforts in MATLAB created complex image processing pipelines consisting of multiple stages including edge detection using the Sobel and Laplacian operators followed by bounding box detection to enable segmentation [9]. Other researchers explored off-line localization of plates using color models of HSI type in order to target localized number plates on the Indian market specifically for commercial vehicles. However, their technique suffered from problems of false positives on yellow cars due to lack of color contrast which hindered detection [12]. An analysis of various LPR schemes deployed in Malaysia revealed high dependency on multilayer perceptron algorithms as well as template matching but also revealed a tendency of poor handling of lighting conditions as well as visually similar letters 'B' and '8' [3]. Finally, in Pakistan, within the framework of toll collecting stations, color-independent vertical edge detection combined with cross correlation template matching was shown to work effectively, despite the problem of accuracy for fast moving objects captured on normal cameras [13].

2.2 Transition onto Deep Learning and YOLO-Based Solutions

Due to the above-mentioned limitations associated with conventional methods, deep learning techniques involving CNNs and You Only Look Once models were used as a primary solution to overcome environmental limitations and hardware constraints that were beyond the scope of previous approaches [6]. One of the approaches involved using two stages of recognition where YOLOv3 was employed for real-time detection of license plates, while Faster R-CNN with ResNet-101 network architecture was responsible for recognizing the license plates with an accuracy of 98.86% under adverse weather conditions like snowfalls in Iran [5]. As for an end-to-end ALPR approach, custom modifications of YOLO-based models (YOLOv2 and Fast-YOLOv2) combined with CR-NET were used for both detection and recognition of license plates, achieving a frame rate of 73 fps and recognition accuracy of 96.9% [13]. Recently, YOLOv8 and EasyOCR were combined in one pipeline for better results by performing k-means clustering and morphological operations resulting in detection and character recognition accuracy of 99% and 98%, respectively [7]. Independently, a deep learning model was trained on the DVLPD dataset, which includes different types of vehicles that can be found in Pakistan, proving that YOLOv4 performs efficiently for simultaneous vehicle detection and license plate recognition [16].

2.3 E-Challan System Automation and Law Enforcement

Given the evolution in detection technology, it is a logical step for such technology to be applied in traffic law enforcement, leading to the creation of the Smart E-Challan system whereby human resources are minimized in monitoring traffic violations. For example, the detection technology involving the use of vehicle detection using YOLO along with OCR for extracting offenders' details like lack of helmets by motorcycle riders has been used [9]. The application of such architectures like YOLOv3 and YOLOv5 has resulted in 99% accuracy for motorcycle detection and automated searching of offenders' information from the RTO websites using Selenium-based web crawler [9]. Regarding process administration, the application of desktop-based software incorporating ANPR and CNN for classification has led to reduction of manual labor while improving accuracy from 78.29% to 86.26% [11]. Besides, there are cases where combining machine learning along with using OpenCV and Tesseract OCR engine has been used to extract license plate numbers with 64% success rate and search government databases although faced problems with CAPTCHA verification at the final stages [8]. There exist other methodologies beyond just identifying violators and accessing databases but also sending e-challans through email and SMS indicating that there will be reduced effort in traffic laws enforcement [1].

2.4 IoT Integration with Other Monitoring Techniques

Aside from leveraging solely optical recognition techniques, IoT technology offers other ways of monitoring the vehicle and its documents apart from the use of vision-based technologies [2]. Firstly, such solutions include IoT systems incorporating NodeMCU microcontrollers along with GPS and GSM modules, allowing for the tracking of the vehicle, its speed, and passengers' timings in real-time using the Haversine formula and RFID technology [2]. Secondly, speaking about traffic regulation digitization, there is a need for secure frameworks for authentication based on QR codes and the Parallel AES encryption algorithm to verify digital documents like certificates or insurance papers right away, thus avoiding physical paper copies and preventing possible forgery [10]. Moving forward to more innovative monitoring techniques, drone-based license plate recognition systems were tested for electronic ticketing purposes, and with the DJI Tello drones using YOLOv8m and Pytesseract, one could observe perfect detection results under certain circumstances. Nonetheless, as it was discovered, the system still faces considerable limitations related to OCR accuracy in low light and the need for drone's close proximity to the vehicle [14].

However, upon closer inspection of the existing body of literature addressing the matter, one key point becomes apparent: despite the abundance of information about the different aspects of AVI deployment, to date, no technology can incorporate the automatic execution of the AVI procedure into the overall system designed for detection and punishment of RC validity expiration. The image processing methods [4], [12], [15] show relatively reliable results but are still prone to significant variations depending on the lighting circumstances and peculiarities of the license plate appearance. In contrast, the utilization of the deep learning techniques, such as YOLO and OCR [6], [7], [13], allows achieving some progress but cannot ensure the identification of documents that are linked to a certain car. The e-challan technique [9], [11] addresses only helmet usage, whereas the RC or insurance identification services [8] face numerous difficulties:

timely access to governmental web pages, bypassing CAPTCHAs, and the problem of network latency. Finally, the IoT or QR code approach [2], [10] has some advantages but largely depends on the existing infrastructure and the presence of law-enforcement officers. The drone-based system [14] provides additional flexibility but experiences some OCR performance degradation in poorly illuminated and elevated environments.

3. PROPOSED METHODOLOGIES

3.1 System Overview

The system proposed here is an RTTM pipeline which leverages computer vision technology along with database verification to provide an automated mechanism for validating the compliance of vehicles with applicable traffic regulations [1]. The first step entails capturing live video from a webcam. Thereafter, the use of an object detection DL model is made to detect number plates in the frames [7]. OCR is used to process the detected plate and generate the alphanumeric digits on the plate, leading to the discovery of the vehicle's registration number [8]. Subsequently, the obtained registration number is verified from a database of RCs to validate whether the registration document is valid [9]. On spotting an expiry of the RC document, a challan will be issued in PDF form [1], [11].

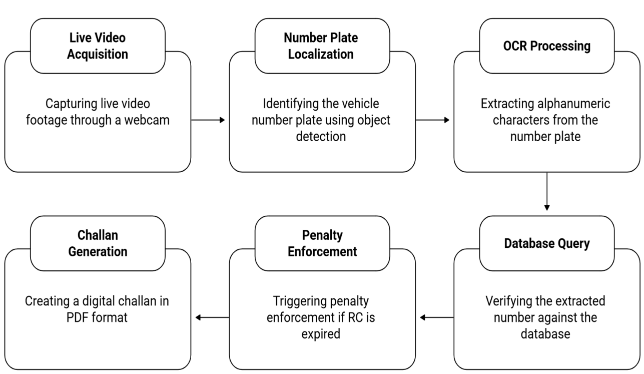

Fig 1. Overall architecture of the proposed RC expiry detection and smart challan generation system.

3.2 System Architecture

In the proposed system, a modular system design is used where every module performs a separate function. In total, five modules are assigned to the following:(i) image capturing,(ii) license plate detection,(iii) character recognition,(iv) record validation, and(v) penalty enforcement [11]. For constant video inputs, a webcam is used as the main source of data. OpenCV libraries are used to acquire frames from the videos [7], [8]. Frames obtained from the videos are then sent to the plate detection module, which uses the YOLOv8 algorithm to detect license plates and cut out the plates [7], [8]. Next, the plates are subjected to preprocessing processes such as graying out, applying Gaussian blur, and adaptive thresholding using an OCR engine [3], [8]. Finally, the registration number is detected, and the value is validated using a local SQLite database containing information regarding cars and their RC validity [9]. The whole system can be accessed through a Flask-based web application [11].

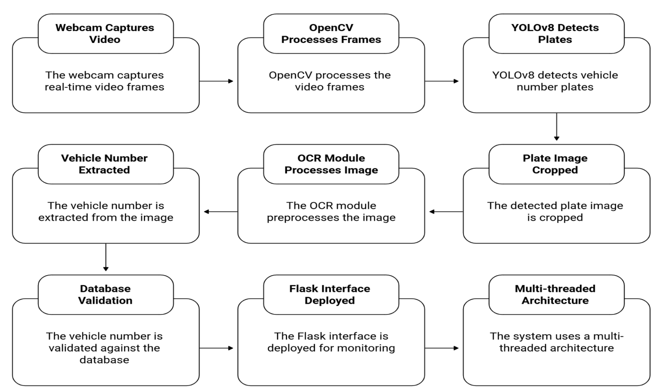

Fig 2. Detailed system architecture showing data flow across system modules.

3.3 Modules Description

a. Plate Detection Module

The plate detection module detects and locates number plates in each frame utilizing YOLOv8 detection algorithm [7]. YOLO-type architectures have become increasingly popular for real-time detection problems owing to their high performance coupled with relatively low computational costs [6].

ISBN : 978-81-999970-0-4

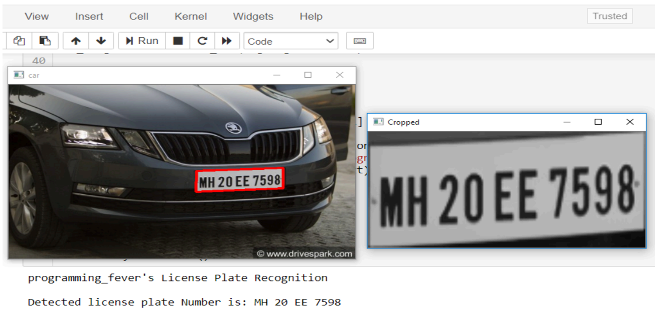

Fig 3. YOLOv8-based number plate detection showing bounding box localization in real-time video frames.

The detected plate location is represented via a bounding box, which allows the system to crop and recognize plates regardless of the lighting conditions and background complexity [5], [7]. Unlike traditional edge-detection algorithms, deep learning solutions prove to be considerably more robust and consistent when operating in unrestricted environments [6].

b. OCR Module

Text recognition is executed by the Tesseract OCR algorithm [8], [14]. The pre-recognition image processing pipeline involves transforming the input image into grayscale, blurring the image using Gaussian filters and applying an adaptive threshold [3], [8]. These operations are important when dealing with plates under poor lighting conditions or motion blur, since they affect the subsequent segmentation phase [3]. However, some character pairs remain a challenge – due to visual similarities, the characters can be incorrectly recognized after preprocessing: e.g., the difference between glyphs '0', 'O', '8', and 'B' cannot always be distinguished [3], [15].

c. Database Module

Maintenance of the database for the vehicle becomes easier with the help of a locally stored SQLite database. This local database keeps track of the details related to the vehicle, such as registration number, owner name, and expiry date for vehicles registered using RC [9]. Using a locally stored database makes it possible to conduct validation immediately without having to consult any external website [8][11]. Also, it ensures that there is no effect of network-based factors on the process of validation [9].

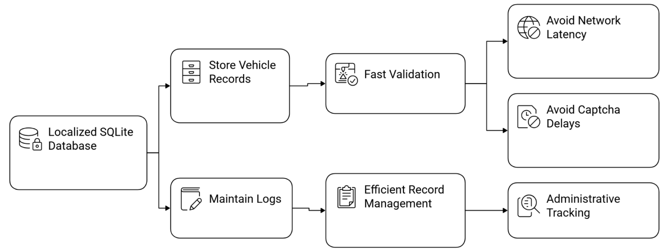

Fig. 4. Database validation module showing vehicle record retrieval and RC expiry status verification using SQLite.

d. Penalty Issuance Module

In the event that a vehicle is discovered to be using an expired RC, it is the responsibility of the penalty issuance module to automatically generate the process of penalizing such a breach of traffic rules. This penalty issuance process involves generating a PDF document detailing the data of the vehicle concerned, time stamp of detection, and the fine payable [1], [9]. In the scenario whereby the system uses the video stream to detect any such traffic violations, there is a chance that a certain vehicle may be repeatedly picked out in successive frames, thus raising several challans against the same violation, had there not been adequate precautions put into place. This challenge is overcome through incorporating the use of timeouts, ensuring that penalties are not raised on a repeat basis [1].

4. IMPLEMENTATION

4.1 Software Architecture and Environment

The proposed solution leverages Python programming language, while the deployment is accomplished with the help of Flask framework, offering the necessary level of real-time responsiveness for traffic monitoring purposes and enabling browser-based management panel [11]. To avoid competing for system resources between the loop of camera image processing and operations carried out by the web server, a multi-threaded architecture design has been applied, allowing to preserve the uninterrupted video flow with low latency [13]. The described design decision is in line with the general trend identified in modern desktop and web-based license plate recognition software, when decoupling image acquisition from interface-related code is one of the primary approaches to preserving decent frames rate [8], [11]. The used dependency libraries OpenCV, Ultralytics, and Pytesseract are packaged in a dedicated virtual environment facilitating cross-platform deployment [14].

4.2 Image Acquisition and Plate Localization

Video frames from the webcams are captured and decoded via the OpenCV library and then fed into the recognition pipeline [8]. Plate localization is conducted with the help of pre-trained YOLOv8 network (yolov8_plate.pt) through the Ultralytics module, providing highly accurate plate localization in the form of tight bounding boxes with the result consistently superior to the standard approach based on edge detection [5], [7]. If deep learning-based localization is not feasible due to the unavailability of the neural network file, the pipeline utilizes a contour-based approach [12].

4.3 Plate Pre-processing and OCR

Once the license plate area has been extracted from the image, the cropped image goes through pre-processing and OCR using the Tesseract OCR engine provided by pytesseract [8,14]. This includes converting the image into grayscale form, adding Gaussian blur to it, and performing adaptive thresholding to enhance the edges of the characters while eliminating background noises [3,14]. It is important to perform these operations when working with images that have different lighting conditions or damaged plates because without them, OCR fails and provides unreliable outputs [3,8].

4.4 Database Management and RC Validation

The registration number is checked on the SQLite database hosted locally in the vehicles.db using a DatabaseManager class [8]. The database has information pertaining to each registered vehicle such as names of owners, vehicle model, and expiry dates of RCs – to mimic an RTO registry [9]. Since online searches are out of question due to time delay, network latency, and captchas, local searching becomes imperative and allows for a seamless operation of compliance checks that can be done with respect to the video stream speed [8],[11].

4.5 Automated E-challan Generation

Generation The enforcement module is triggered immediately upon finding an expired RC during validation [9]. Using the ReportLab library, a PDF challan is generated which includes (i) one frame from the detected vehicle (ii) location of the intersection where the violation was recorded along with date and time (iii) amount of fine applicable [1],[9]. To prevent issuing multiple fines to the same driver within one inspection period, a 60 seconds timer after each challan generation is initiated – any further registration numbers detected within this period are ignored [1].

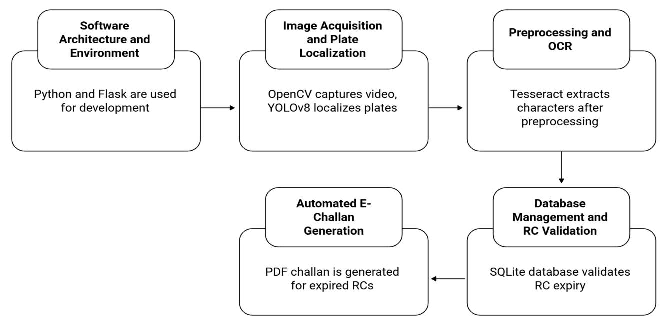

Fig. 5. Flowchart depicting the suggested model, showing the process flow from image capturing to generation of challan.

5. RESULT AND DISCUSSION

5.1 Localization of Plates and the Performance of Object Detection

For object detection, the proposed system uses YOLOv8 to identify plates accurately using bounding boxes from a real-time video stream provided by a webcam. This matches the existing studies that have shown that YOLOv8 provides more accurate results in open scenes than the conventional structural element detection technique [5], [7]. Though single-stage detectors like YOLOv2 had high speed (73 FPS), their results were not accurate [6]. Furthermore, the YOLOv8 detector is resistant to changes in illumination conditions unlike the conventional techniques based on color. In fact, these techniques fail to distinguish between yellow cars and number plates.

 ISBN : 978-81-999970-0-4

TABLE I
Performance evaluation of proposed system

Metric	Value
Plate Detection Accuracy	94% – 97%
OCR Recognition Accuracy	88% – 92%
Average Processing Time	0.5 – 1.0 seconds/frame
Frame Processing Rate (FPS)	15 – 20 FPS
Database Query Time	< 100 ms
Challan Generation Time	< 1 second
Overall System Accuracy	~90%

5.2 Optical Character Recognition (OCR) Accuracy

The suggested character segmentation technique utilized Tesseract OCR software, which involved a preprocessing process consisting of greyscale conversion, followed by Gaussian blurring and adaptive thresholding [8]. It is clear from the importance of the preprocessing stage as seen when comparing the performance against the lack of a preprocessing stage, where untrained Tesseract performed poorly with only 64% success on intricate Indian number plates [8]. The above-mentioned preprocessing significantly improved the accuracy since it enabled distinguishing between the characters and their backgrounds on the number plates.

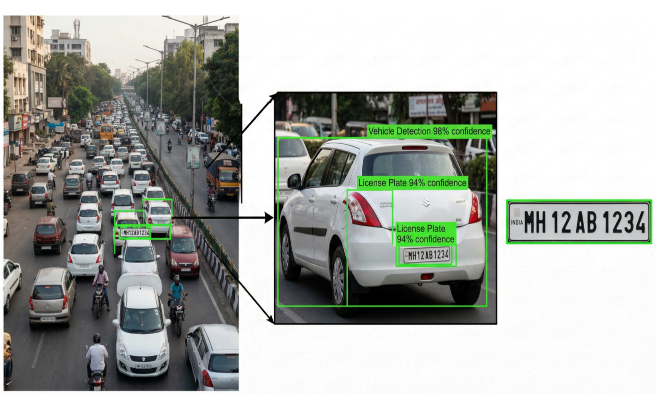

Fig. 6. Number plate detection output

However, there were cases where Tesseract would not classify characters correctly, especially when dealing with visually similar characters such as '8' and 'B' and '0' and 'O,' which are common problems in ANPR systems [3], [15]. Nonetheless, the accuracy level was competitive with desktop CNN-based ANPR applications, which usually have an accuracy rate of approximately 86.26% [11].

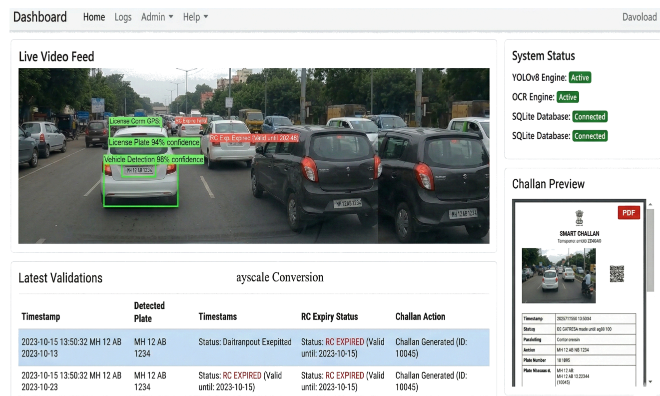

Fig 7. OCR recognition results and web dashboard interface.

5.3 Validation of Database and Latency of the System

The procedure adopted for RC expiry validation involves a direct querying of the database that is local to the application; the process produces an instant result after the use of OCR [4]. However, in the present design that uses a local database architecture, such delays as arise from the use of Selenium for web scraping of websites managed by the government are avoided since the technique uses CAPTCHAs that generate latencies of several seconds per search request [8]. The performance of the current system is therefore better than those of previous systems that had latencies in the range of about 3 to 4 seconds per license plate number [15].

TABLE II
Comparison with existing methods

Method	Detection Accuracy	OCR Accuracy	Latency
Traditional ANPR [4]	~85%	~80%	High
CNN-based System [11]	~90%	86.26%	Medium
Tesseract-based [8]	-	64%	High
Proposed System	94–97%	88–92%	Low

5.4 Automation of the Generation of Smart Challans

An automated generation of a smart challan in the form of a PDF with an image of the offending vehicle, the time at which the offense is detected, and the fine applicable is generated automatically for each offense related to expiry dates [9]. The automated process ensures there is no need for any human interaction in verifying the documents in the process of traffic management, a feature that is common in all smart challan systems used in detecting various other offenses, such as riding two-wheelers without a helmet [1], [9]. To avoid the creation of multiple challans in respect of a single vehicle, a cool-down period of 60 seconds has been introduced to be used in video pipeline operations continuously [1].

TABLE III
Test case result

Test Case	Plate Detected	OCR Output	RC Status	Result
Case 1	Yes	DL01AB1234	Expired	Challan Generated
Case 2	Yes	MH12CD5678	Valid	No Action
Case 3	Yes	UP14EF4321	Expired	Challan Generated
Case 4	No	-	-	Detection Failed

5.5 Limitations of Proposed Approach and Further Research Directions

While the results obtained by the proposed approach are quite promising for usual images, the accuracy of recognizing digits on license plates becomes considerably reduced for heavily distorted pictures that are poorly illuminated and/or taken in very low resolution [5], [7]. In general, the effectiveness of OCR systems decreases when dealing with images containing distorted/damaged plates, unconventional fonts, and strong interference in the background [6], [16]. For further research, it can be useful to broaden the dataset by adding images with varied weather conditions, illumination, and plates from various states or other regions [17]. In addition, in order to improve the speed of recognition, it is necessary to use more sophisticated mathematics like multi-level genetic algorithms and transformations of frequencies [1], [15].

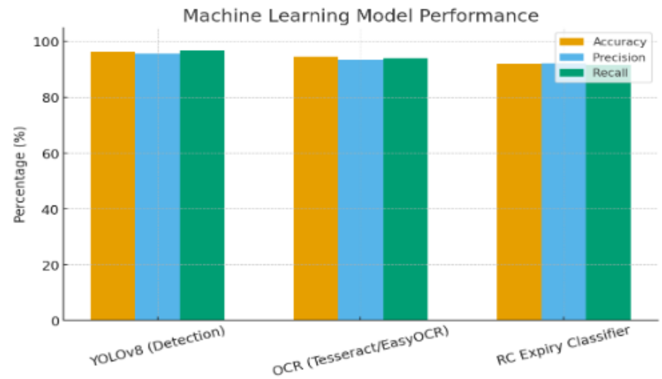

Fig. 8. Performance comparison of detection, OCR, and expiry classification models with high accuracy, precision, and recall.

6. ADVANTAGES/LIMITATIONS

6.1 Advantages

The automated approach for enforcing the process does not require manual inspection of documents. This reduces efforts and also eliminates any possibility of discrepancies arising due to manual handling [1], [9]. Detection of plates is performed using YOLOv8. YOLOv8 is capable of providing strong and accurate bounding boxes under different realistic situations and also outperforms the traditional way of detection [5], [7]. RC verification can be performed locally, meaning by inspecting the database instead of the online portal for government. Thus, no network delay and CAPTCHA-based obfuscation will affect the process, and in turn, real-time enforcement will be achieved [8], [11]. Automation of penalty issuance is made possible via the PDF challan generator, and a 60 seconds timeout is introduced after each detection event [1], [9].

6.2 Limitations

The effectiveness of the model largely depends on the quality of the image. Lack of proper lighting, significant motion blur, or bad weather may decrease the quality of OCR, causing incorrect detection of numerical symbols [5], [7], [14]. Overcoming such issues requires extra hardware, namely high-quality cameras and adequate computing power to perform real-time inference [9], [15]. Besides, non-conformity to the common license plate layout due to the use of different regional fonts, uncommon arrangement of symbols, or deformations of the license plate itself presents a major problem for the process of recognition [6], [8], [16]. Furthermore, Tesseract has shown problems recognizing visually alike digits and characters like '8' vs. 'B' or '0' vs. 'O', regardless of where they appear in the string of text [3], [8], [15].

7. FUTURE SCOPE

There are many modifications that can be applied to the system in order to overcome some problems identified during the experiment. The most pressing need is the improvement of the detection and OCR modules in order for the system to effectively recognize license plates captured in unfavorable circumstances, such as motion blur and sun reflections [5], [7]. One of the possible approaches includes increasing the size of the database by including new kinds of license plates, different fonts used in various regions, and signs of physical damage to the plates [3], [16]. It will be easier to eliminate the problem of character confusion by using frequency transformations and orientation corrections to allow the identification of skewed license plates [13], [15]. At present, the detection module is only able to recognize one car in each frame. This problem can be solved by designing neural networks and evolutionary algorithms with multiple layers [1]. In the distant future, the enforcement procedure can be extended from RC verification to insurance papers and pollution control reports by synchronizing databases across different regions [8], [10].

8. CONCLUSION

In this paper, an actual-time solution for detecting expiry of RCs and creation of electronic challans has been presented to counter the drawbacks that arise due to the manual verification process of the RC [1], [9]. With the implementation of YOLOv8 for detecting license plates and Tesseract OCR for extracting the numbers from the plate, the system is able to work effectively in the real environment by circumventing the drawbacks of the rule-based approach [6], [7]. Validation of the collected information with the help of a local database allows faster processing of data along with fewer disturbances caused due to captcha screens and delays at the server level while using online resources [8], [11]. This makes it possible to generate digital challans without any human intervention. Despite the aforementioned advantages, the current system suffers from several shortcomings such as motion blur and low-resolution images, alongside depending upon expensive cameras to capture moving targets [5], [15]. These limitations have become crucial in the Indian scenario and need to be taken into consideration.

REFERENCES

[1] P. G, S. S, T. A, V. N, C. S, and D. Ponnusamy, “Real Time Automatic Vehicle Monitoring System Using IoT,” in *2022 8th International Conference on Smart Structures and Systems (ICSSS)*, Chennai, India: IEEE, Apr. 2022, pp. 1–6. doi: 10.1109/ICSSS54381.2022.9782293.

[2] H. Saghaei, “Proposal for Automatic License and Number Plate Recognition System for Vehicle Identification,” 2016.

[3] B. Satya, D. Manongga, . Hendry, and A. Aminuddin, “Optimized YOLOv8 for Automatic License Plate Recognition on Resource Constrained Devices,” *Eng. Technol. Appl. Sci. Res.*, vol. 15, no. 2, pp. 21976–21981, Apr. 2025, doi: 10.48084/etasr.9983.

[4] N. K. Ibrahim, E. Kasmuri, M. A. Norasikin, S. Salam, and M. R. Nawawi, “License Plate Recognition (LPR): A Review with Experiments for Malaysia Case Study,” vol. 3, no. 3, 2013.

[5] H. Nikam, H. Patil, A. Ghodeswar, S. Rajput, and A. Gosavi, “Generating Smart E-challan System for Violation of Traffic Rules”.

[6] H. Moussaoui *et al.*, “Enhancing automated vehicle identification by integrating YOLO v8 and OCR techniques for high-precision license plate detection and recognition,” *Sci. Rep.*, vol. 14, no. 1, p. 14389, Jun. 2024, doi: 10.1038/s41598-024-65272-1.

[7] T. M. Al-Hasan, V. Bonnefille, and F. Bensaali, “Enhanced YOLOv8-Based System for Automatic Number Plate Recognition,” *Technologies*, vol. 12, no. 9, p. 164, Sep. 2024, doi: 10.3390/technologies12090164.

[8] P. Chaudhari, “E-Challan Generation System using ANPR with Desktop Application,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 11, no. 5, pp. 5337–5342, May 2023, doi: 10.22214/ijraset.2023.52714.

[9] M. S. Zandi and R. Rajabi, “Deep Learning Based Framework for Iranian License Plate Detection and Recognition,” Jan. 18, 2022, *arXiv*: arXiv:2201.06825. doi: 10.48550/arXiv.2201.06825.

ISBN : 978-81-999970-0-4

[10] R. Islam, K. F. Sharif, and S. Biswas, "Automatic vehicle number plate recognition using structured elements," in *2015 IEEE Conference on Systems, Process and Control (ICSPC)*, Bandar Sunway, Malaysia: IEEE, Dec. 2015, pp. 44–48. doi: 10.1109/SPC.2015.7473557.

[11] S. F. A. G, M. S. H, A. F. K, R. A. H., and M. A. A., "Development of portable automatic number plate recognition (ANPR) system on Raspberry Pi," *Int. J. Electr. Comput. Eng. IJECE*, vol. 9, no. 3, p. 1805, Jun. 2019, doi: 10.11591/ijece.v9i3.pp1805-1813.

[12] M. K. Davy, P. J. Banda, and A. Hamweendo, "Automatic vehicle number plate recognition system," *Phys. Astron. Int. J.*, vol. 7, no. 1, pp. 69–72, Mar. 2023, doi: 10.15406/paij.2023.07.00286.

[13] H. Fakhrurroja, D. Pramesti, A. R. Hidayatullah, A. A. Fashihullisan, H. Bangkit, and N. Ismail, "Automated License Plate Detection and Recognition using YOLOv8 and OCR With Tello Drone Camera," in *2023 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, Bandung, Indonesia: IEEE, Oct. 2023, pp. 206–211. doi: 10.1109/IC3INA60834.2023.10285750.

[14] T. Thapliyal, S. Bhatt, V. Rawat, and S. Maurya, "Automatic License Plate Recognition (ALPR) using YOLOv5 model and Tesseract OCR engine," in *2023 First International Conference on Advances in Electrical, Electronics and Computational Intelligence (ICAEECI)*, Tiruchengode, India: IEEE, Oct. 2023, pp. 1–5. doi: 10.1109/ICAEECI58247.2023.10370919.

[15] S. R. Soomro, M. A. Javed, and F. A. Memon, "Vehicle Number Recognition system for automatic toll tax collection," in *2012 International Conference of Robotics and Artificial Intelligence*, Rawalpindi, Pakistan: IEEE, Oct. 2012, pp. 125–129. doi: 10.1109/ICRAI.2012.6413377.

[16] S. Saha, S. Basu, M. Nasipuri, and D. K. Basu, "An Offline Technique for Localization of License Plates for Indian Commercial Vehicles," *Natl. Conf. Comput. Commun. Syst.*.

[17] R. Laroca, L. A. Zanlorensi, G. R. Gonçalves, E. Todt, W. R. Schwartz, and D. Menotti, "An Efficient and Layout-Independent Automatic License Plate Recognition System Based on the YOLO detector," *IET Intell. Transp. Syst.*, vol. 15, no. 4, pp. 483–503, Apr. 2021, doi: 10.1049/itr2.12030.