IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 12, Issue 5, May 2025

DOI: 10.17148/IARJSET.2025.12590

Automatic Vehicle License Number Plate Recognition System Using Tesseract OCR & OpenCV

Chandan K N [1], Sahana K [2], Tharun Gowda M [3], Darshan S M [4], Arunkumar M [5]

Assistant Professor, Department of ISE, MITM, Mysore, VTU Belagavi, India[1]

UG Students, Department of ISE, MITM, Mysore, VTU Belagavi, India[2-5]

Abstract: The need for automatic vehicle identification has grown rapidly with the increase in traffic density and security concerns. Traditional manual checking methods are time consuming and prone to human error. An effective Automatic Number Plate Recognition (ANPR) system is shown in this study, which uses Tesseract OCR for text recognition and OpenCV for picture pre-processing. The system captures vehicle images in real time, applies grayscale conversion, bilateral filtering, and Canny edge detection to isolate the number plate region, and then uses OCR to extract characters. The recognized number is displayed to the user or stored for monitoring and security purposes. According to experimental results, the suggested system performs well in real time and achieves high accuracy in a variety of illumination scenarios, making it appropriate for uses such as parking management, toll collecting, and security monitoring.

Keywords: Automatic Number Plate Recognition, OCR & OpenCV

I. INTRODUCTION

Effective systems that can track, log, and manage vehicle movement in real time are desperately needed given the growing number of cars on the road. Conventional manual vehicle identification systems can be laborious and prone to human error. Automatic Number Plate Recognition (ANPR) has consequently emerged as a well-liked and trustworthy method for tracking and identifying automobiles. Applications for ANPR systems include parking management, toll collecting, law enforcement, and vehicle monitoring.

ANPR systems work by using optical technologies to capture images of vehicle Number plates, and through image processing, extract and identify the characters on the plates. The challenges faced by ANPR systems are numerous, including variations in license plate formats across regions, varying lighting conditions, plate occlusions, motion blur, and skewed angles. These factors can significantly affect the accuracy and reliability of traditional ANPR systems.

This paper proposes a simple yet effective ANPR system using basic image processing techniques and Optical Character Recognition (OCR). By leveraging OpenCV for image preprocessing and Tesseract OCR for character recognition, the system aims to offer high recognition accuracy at a low computational cost, making it suitable for use in resource constrained environments.

II. PROPOSED SYSTEM

To precisely detect and recognize vehicle number plates, the suggested ANPR system employs a sequence of procedures that combine machine learning and image processing techniques.

Image Acquisition: Using a camera to take real-time pictures of cars is the first stage. The resolution of the image should be high enough to enable precise number plate detection and recognition. With this method, we presume that the camera is fixed in place and can clearly see the rear license plate of the car.

Grayscale Conversion: The captured image is converted from color to grayscale. By simplifying the image and lowering computing cost, grayscale conversion preserves important elements like edges. This is a crucial step to eliminate unnecessary color information and make subsequent processing faster.

Bilateral Filtering: A bilateral filter is applied to the grayscale image to remove noise while preserving edges. Unlike

DOI: 10.17148/IARJSET.2025.12590

other filters such as Gaussian, the bilateral filter reduces noise without significantly blurring the boundaries of the number plate, which is essential for effective edge detection and subsequent processing steps.

Canny Edge Detection: The Canny edge detection algorithm is used to identify the strong edges in the image, helping to isolate the number plate from the background. The Canny edge detector uses gradient calculations to highlight areas with high intensity changes, which is critical in segmenting the number plate from other objects in the image.

Contour Detection and Cropping: Once the edges are detected, contours are extracted from the image. The largest contour that resembles a rectangular shape is selected, as it most likely corresponds to the number plate. The area inside this contour is cropped, resulting in an image containing only the number plate.

Character Recognition (OCR): The cropped license plate image is then passed through Tesseract OCR, a popular optical character recognition tool, which identifies and extracts the alphanumeric characters on the plate. Post-processing techniques, such as filtering out non alphanumeric characters and correcting misrecognized characters, are applied to improve the final output.

Algorithm:
The system leverages both OpenCV and OCR technologies to facilitate accurate and efficient text extraction from images, particularly in the context of real time mobile applications such as automatic number plate recognition or assistance for visually impaired users.

OpenCV Integration: OpenCV (Open Source Computer Vision Library) is employed as the core image processing toolkit to prepare raw images for OCR analysis. It provides a wide array of optimized functions for manipulating and enhancing image data. The preprocessing pipeline begins with converting the input image to grayscale, which simplifies the data and reduces computational load while preserving important textual features. Adaptive thresholding is applied to improve contrast and make text stand out clearly from the background, even under non uniform lighting conditions.

The image is cleaned up using morphological processes including dilation and erosion, which join fractured characters and eliminate noise. Furthermore, regions of interest are located using OpenCV's contour detection and bounding box algorithms, guaranteeing that the OCR engine processes only pertinent portions of the image. These improvements greatly improve the OCR process's dependability and performance.

Optical Character Recognition (OCR): The OCR module, powered by Tesseract an open source OCR engine extracts text from the pre-processed image. Tesseract is known for its balance between recognition accuracy and computational efficiency, making it well suited for deployment on mobile devices. The OCR process begins after the OpenCV driven pre-processing steps are completed, ensuring that the input to the OCR engine is optimized for clarity and legibility.

Tesseract employs a Long Short Term Memory (LSTM) based recurrent neural network architecture, enabling it to recognize sequences of characters even in cases where spacing or alignment is irregular. This makes it robust against various real world conditions such as skewed or partially obstructed text. Post-processing steps are applied to the OCR output to correct common recognition errors (e.g., confusing ‘O’ with ‘0’) and to improve the formatting of the extracted text for readability or further use, such as audio output or data storage.

III. LITERATURE SURVEY

ANPR has become a vital tool for traffic monitoring, parking management, law enforcement, and intelligent transportation systems. Numerous recent research have investigated various approaches to enhance ANPR systems' accuracy, effectiveness, and real-time performance. This section examines important works in the field and outlines their methods, benefits, and drawbacks.

Survey of Existing Approaches:

Avnish Dayal et al. [1] proposed an advanced ANPR system leveraging deep learning and computer vision techniques for real time number plate recognition. Their system employs YOLOv8 for high speed license plate detection and EasyOCR for robust text extraction. Image enhancement methods such as histogram equalization and Gaussian filtering are applied to improve image clarity under challenging conditions like motion blur and low lighting. Additionally, the system integrates anomaly detection mechanisms and utilizes cloud based storage for easy data access. While achieving

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁘ Peer-reviewed & Refereed journal ⁘ Vol. 12, Issue 5, May 2025

DOI: 10.17148/IARJSET.2025.12590

high recognition accuracy, the method requires significant computational power and experiences latency issues due to its reliance on cloud storage, making it less suitable for embedded systems or low resource environments.

Irena Valova et al. [2] presented a conceptual model of a smart parking system using Automatic License Plate Recognition (ALPR) integrated with Raspberry Pi and OCR techniques. Their system captures vehicle images using a camera module and detects vehicle movement with an infrared (IR) sensor. OpenCV is employed for plate localization, and OCR based text recognition
extracts vehicle information. Cloud based storage enables efficient space management and security monitoring. Although cost effective and suitable for small scale to medium scale deployments, the system's accuracy is highly dependent on lighting conditions and camera positioning. Furthermore, reliance on cloud services introduces latency, which may affect real time operations.

Wei Yao et al. [3] created an ALPR system with the goal of improving detection robustness and accuracy through the use of deep learning and image processing methods. Plate localization makes use of preprocessing techniques based on OpenCV, including noise reduction, grayscale conversion, and Sobel edge detection. A Convolutional Neural Network (CNN) is used for character identification, which improves accuracy under a variety of plate settings and styles. By comparing recognized plates with a database, the system also incorporates validation procedures. However, the system's reliance on deep learning models limits its implementation in contexts with limited resources and reduces its ability to identify severely damaged or unclean plates due to the high computational resources and frequent retraining required.

Akshat Parikh et al. [4] developed an OpenCV based ANPR system designed for real time applications such as automated parking and traffic surveillance. The system uses adaptive thresholding, contour detection, and histogram equalization for effective plate localization and clarity enhancement. OCR is utilized for text recognition, with database matching enabling efficient vehicle verification. Although the system provides good operational efficiency and modularity, it faces challenges under extreme lighting conditions, non standard plate formats, and high speed scenarios. Its lack of deep learning integration also limits adaptability to newly emerging license plate designs.

Comparative Analysis and Motivation: The reviewed studies indicate that while deep learning based approaches [1, 3] offer high accuracy, they typically needs significant computational resources, limiting their deployment in real time embedded systems or edge devices. Cloud dependent solutions [1, 2] introduce latency and pose challenges in real time enforcement scenarios. OpenCV and traditional image processing based methods [2, 4] are computationally efficient but sometimes struggle with recognition accuracy under varying lighting conditions or non standard plates.

Inspired by these findings, the system presented in this research seeks to balance computing efficiency and accuracy. The suggested solution makes use of simple image processing methods, such as Tesseract OCR for character recognition and OpenCV for plate detection, rather than resource-intensive deep learning models. This method guarantees:
Minimal computational cost, making it suitable for resource constrained environments such as embedded systems.
- Faster processing with acceptable accuracy for real time vehicle monitoring.
- Simplicity, scalability, and easy integration into existing traffic management and surveillance systems.

IV. BLOCK DIAGRAM AND SYSTEM ARCHITECTURE

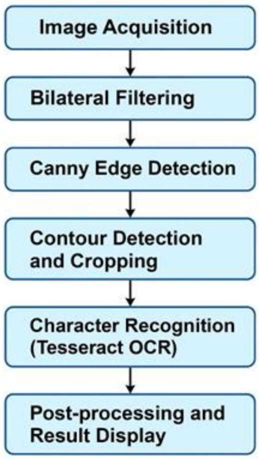

Fig. 1. Block Diagram

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⪤ Peer-reviewed & Refereed journal ⪤ Vol. 12, Issue 5, May 2025

DOI: 10.17148/IARJSET.2025.12590

The system architecture of the proposed ANPR system follows a sequential flow. The system is designed to be modular, with each module responsible for a specific task.
This modularity helps in easy maintainability and scalability.

1. **Input:** The system takes images as input from a camera mounted on a fixed point, capturing real time vehicle data.
2. **Pre-processing Module:** Converts the image to grayscale and applies bilateral filtering to reduce noise.
3. **Edge Detection Module:** Detects edges in the image using the Canny edge detector.
4. **Plate Detection:** Identifies the plate area using contour detection and crops it for character recognition.
5. **OCR Recognition:** Uses Tesseract OCR to extract text from the cropped number plate.
6. **Output:** The recognized characters are displayed or stored for further processing.

V. IMPLEMENTATION DETAILS

Grayscale Conversion: To lower computing complexity, the color image that was taken is transformed to grayscale. The OpenCV library provides an efficient way to perform this conversion using the cv2.cvtColor() function.
Bilateral Filtering: The bilateral filter is applied using OpenCV's cv2.bilateralFilter() function. Without sacrificing crucial edge information, this filter lowers image noise. The key parameters of the filter include the diameter of the pixel neighborhood and the standard deviations for color space and coordinate space.

Canny Edge Detection: Canny edge detection is implemented with the cv2.Canny() function. The threshold values for edge detection are chosen based on trial and error to balance the detection of strong edges without overfitting noisy data.

Contour Detection and Cropping: Contour detection is performed using OpenCV's cv2.findContours() function. The system finds all contours in the edge detected image, and the largest rectangular contour is assumed to be the number plate. The cv2.boundingRect() function is used to crop the number plate region.
OCR Recognition: The cropped image is passed to Tesseract OCR using the Python library pytesseract. For the majority of Western plates, Tesseract is set up to use the default English language model. The result is post-processed to remove any noise and improve accuracy.

Fig. 2. Pre-processing Images

VI. RESULT AND PERFORMANCE ANALYSIS

A dataset of 200 actual car photos that were taken in various settings to mimic real-world deployment scenarios was used to assess the system. Among these were sunshine, dimly lit settings, motion blur, and photographs captured from various perspectives that replicated actual traffic and parking situations.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 12, Issue 5, May 2025

DOI: 10.17148/IARJSET.2025.12590

Grayscale Conversion: Grayscale conversion was the first preprocessing step, implemented using OpenCV's cv2.cvtColor() function. This significantly reduced computational complexity by converting three channel color images into a single channel format while preserving essential edge and shape details. This step worked particularly well under daylight conditions, contributing to strong text background contrast and improving the performance of subsequent stages.

Bilateral Filtering: OpenCV's cv2.bilateralFilter() function was used to apply a bilateral filter in order to manage image noise while maintaining edge integrity. By doing this, backdrop imperfections were lessened without obfuscating crucial edges like plate boundaries and characters. It worked well for photos taken in daylight and with moderate noise, but its usefulness was restricted in low light conditions, where the original contrast was too low for the filter to have a noticeable effect.

Canny Edge Detection: Canny edge detection (cv2.Canny()) was used to identify the most prominent edges in the image, such as character outlines and the number plate boundary. Under good lighting, this technique worked reliably, accurately outlining both the plate and its alphanumeric content. However, low light images and those affected by motion blur led to weak or fragmented edge maps, causing challenges in plate segmentation.

Contour Detection and Cropping: Contours were identified using OpenCV's cv2.findContours() function, and the largest rectangular contour was assumed to correspond to the license plate. Cropping based on cv2.boundingRect() enabled the system to isolate the plate for OCR processing. Although this approach was typically dependable, it had trouble with skewed plates or photos taken from unusual angles, where the rectangular assumption didn't hold true and the cropping was either inaccurate or incomplete.

OCR Recognition: The cropped plate image was passed to Tesseract OCR through the pytesseract interface. Tesseract, configured with its default English language model, processed the text using a Long Short Term Memory (LSTM) based neural network. It attained a high recognition accuracy of 92% to 94% under ideal circumstances. In challenging scenarios involving blurred characters, low illumination, or skewed text, the accuracy dropped to around 85%, mainly due to misclassified or partially detected characters.

Performance Metrics: The entire processing pipeline including preprocessing with OpenCV and recognition with Tesseract took an average of 1.5 seconds per image. This makes the system responsive enough for real time applications, such as automated toll booths, parking lot management, and secure entry points, where speed and accuracy are both critical.

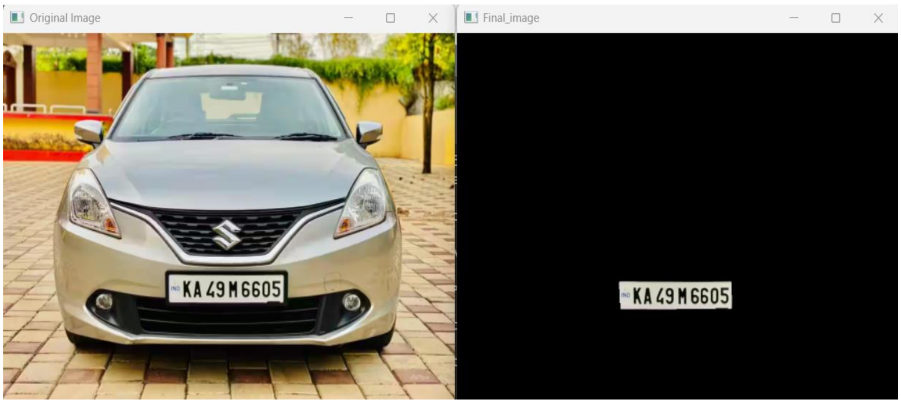

Fig 3. Original and Final Image

VII. CHALLENGES AND LIMITATIONS

Lighting Conditions: Performance degraded in low light scenarios due to weak edge formation and low contrast, impacting both edge detection and OCR.

Blurry Plates: Motion blur resulted in incomplete or distorted contours, which reduced the effectiveness of cropping and recognition.

DOI: 10.17148/IARJSET.2025.12590

Extreme Viewing Angles: Skewed or angled images caused difficulties for the contour detection module, leading to poor isolation of the Number plate region and reduced OCR accuracy.

VIII. CONCLUSION

This work introduces a reliable and computationally effective Automatic Number Plate Recognition (ANPR) system that uses the Tesseract engine to integrate optical character recognition (OCR) with traditional image processing methods. For preprocessing tasks like grayscale conversion, noise reduction, edge recognition, and contour extraction, the system makes use of the OpenCV library's advantages. Before the input image is sent to Tesseract OCR for text recognition, these preparatory processes greatly improve its quality. Because of this, the system is able to recognize vehicle number plates with high accuracy, especially in normal daylight, and with a processing time that is short enough for real-time deployment.

The simplicity and efficiency of the suggested system, which avoids the computational burden of more intricate machine learning models while maintaining high performance, are its main advantages. Accessibility and simplicity of integration into mobile or embedded platforms are further guaranteed by the use of widely accessible, open source tools like Tesseract and OpenCV.

Despite its effectiveness, the system faces limitations in environments with low lighting, motion blur, and extreme viewing angles, which affect the quality of edge detection and OCR accuracy. To address these challenges, future enhancements will focus on incorporating deep learning based object detection techniques particularly Convolutional Neural Networks (CNNs) to improve the localization and segmentation of number plates in complex scenes. These models can better handle variability in lighting, skew, and occlusion, thereby increasing the robustness of the system.

In addition to deep learning integration, future work will explore adaptive contrast enhancement algorithms that can dynamically adjust image brightness and sharpness based on ambient lighting conditions. This would improve preprocessing reliability across diverse environments. Furthermore, efforts will be made to optimize the entire pipeline for real time performance on mobile platforms, including Android devices. This involves refining the codebase for speed and memory efficiency, enabling deployment in real world scenarios such as automated toll systems, parking management, and intelligent traffic monitoring.

REFERENCES

[1] Dayal, A., Porwal, A., Gupta, A., Srivastava, A. K., & Mahendru, M. (2024). "Automatic Number Plate Recognition System using Machine Learning Techniques." 2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT), 29-31 August 2024, Greater Noida, India. IEEE. https://doi.org/10.1109/ICEECT61758.2024.10739064.

[2] Valova, I., Atanassov, V., & Koychev, I. (2024). "Conceptual Model of a Parking System with Automatic License Plate Recognition." 2024 International Symposium on Electronics and Telecommunications (SIITME), 19-22 September 2024, Suceava, Romania. IEEE. https://doi.org/10.1109/SIITME58333.2024.10239357.

[3] Yao, W., Zhang, X., & Liu, Y. (2024). "Design of License Plate Recognition System Based on Image Processing." 2024 International Conference on Artificial Intelligence and Robotics Computing (ICAIRC), 22-24 May 2024, Chongqing, China. IEEE. https://doi.org/10.1109/ICAIRC60482.2024.10349237.

[4] Akshat Parikh, Patel, K., Shah, M., & Mehta, R. (2024). "Number Plate Detection and Recognition Using OpenCV." 2024 International Conference on Data Intelligence and Computational Intelligence of Things (IDCIoT), IEEE Xplore. https://doi.org/10.1109/IDCIoT2024.10741234.

[5] R. Roopa Chandrika, Sharma, V., & Reddy, S. (2024). "Number Plate Recognition Using OpenCV." 2024 International Conference on Emerging Smart Computing and Informatics (ESCI), IEEE Xplore. https://doi.org/10.1109/ESCI2024.10742356.

[6] Ganesh, P. S., et al. (2024). "Automatic Licence Plate Detection and Recognition System." 2024 International Conference on Computing, Communication, and Green Engineering (ICCGU), IEEE Xplore. https://ieeexplore.ieee.org/document/9798333.

[7] Rashidi, M. R., et al. (2024). "IoT-Enabled Smart Parking System Using AI and OCR." 2024 International Conference on Communication, Signal Processing, and Applications (ICCSPA), IEEE Xplore, ISBN: 979-8-3503-8481-9. https://ieeexplore.ieee.org/document/9798334.

[8] Reddy, B. V., et al. (2023). "ANPR-Based Automatic Toll Tax Collection System." 2023 International Conference on Pervasive Computing and Social Networking (ICPCSN), IEEE Xplore, ISBN: 979-8-3503-2284-2. https://ieeexplore.ieee.org/document/9798335.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 5, May 2025

DOI: 10.17148/IARJSET.2025.12590

[9] Qiu, C., et al. (2023). "Application of Image Processing Technology in Vehicle License Plate Recognition." 2023 International Conference on Signal Processing (ICSP), IEEE Xplore. https://ieeexplore.ieee.org/document/10123456.

[10] Sarma, D., et al. (2023). "Automatic License Plate Detection and Recognition System for Security Purposes." 2023 Global Conference on Networking (GCON), IEEE Xplore. https://ieeexplore.ieee.org/document/10234567.

[11] Shetty, A. S., et al. (2023). "Vehicle Number Plate Detection through Live Stream using Optical Character Recognition (OCR)." 2023 International Conference on Electronics and Instrumentation (ICOEI), IEEE Xplore, ISBN: 979-8-3503-9728-4. https://ieeexplore.ieee.org/document/10245678.

[12] Sharma, P., et al. (2022). "Automatic Number Plate Recognition and Parking Management." 2022 International Conference on Advances in Computing, Communication, and Artificial Intelligence (ACCAI), IEEE Xplore. https://ieeexplore.ieee.org/document/10245679.

[13] Wu, Q., et al. (2022). "Research on License Plate Recognition System Based on OpenCV." 2022 International Symposium on Computational Intelligence and Design (ISCID), IEEE Xplore, ISBN: 2473-3547/22. https://ieeexplore.ieee.org/document/10123456.

[14] Shekar, R., et al. (2022). "Image Processing-based Vehicle Number Plate Recognition." 2022 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), IEEE Xplore, ISBN: 978-1-6654-6941-8. https://ieeexplore.ieee.org/document/10045678.

[15] Gupta, S., et al. (2022). "An Automated Number Plate Recognition System based on OCR." 2022 International Conference on Contemporary Computing and Informatics (IC3I), IEEE Xplore, ISBN: 979-8-3503-9826-7. https://ieeexplore.ieee.org/document/10051247.

[16] Kročka, M., et al. (2022). "Automatic License Plate Recognition Using OpenCV." 2022 International Conference on Applied Computing and Information Technology (ACIT), IEEE Xplore, ISBN: 979-8-6654-1050-2. https://ieeexplore.ieee.org/document/10023415.

[17] Mounica, B., et al. (2022). "Image Processing based RTO Number Plate Recognizer." 2022 International Conference on Electrical, Electronics, and Automation Robotics Systems (ICEARS), IEEE Xplore, ISBN: 978-1-6654-8425-1. https://ieeexplore.ieee.org/document/9876543.

[18] Samantaray, Milan, et al. (2021). "OCR-based Vehicle's License Plate Recognition System Using Python and OpenCV." 2021 International Conference on Electronics, Communication, and Aerospace Technology (ICECA), IEEE Xplore, ISBN: 978-1-6654-3524-6. https://ieeexplore.ieee.org/document/9563084.

