Nanotechnology Perceptions
ISSN 1660-6795
www.nano-*ntp.com*

Vehicle Number Plate Recognition Using Opencv: A Python-Based Solution

Vasim Vohra[1]*, Sohil Dabhi[2], Harikrushna Rathod[3] , Keyur Patel[4]

[1-4]*Department of Electronics and Communication, Dharmsinh Desai University, India*
Email: vasim.vohra.ec@ddu.ac.in

Automatic Number Plate Recognition (ANPR) helps in tracking vehicles, improving security, and enforcing traffic rules. This project presents a system that can automatically detect and read vehicle number plates using Python, OpenCV, and Tesseract OCR. It captures an image of a vehicle, enhances visibility, locates the number plate, and extracts its characters using Optical Character Recognition (OCR). By automating this process, the system reduces manual effort, improves accuracy, and speeds up vehicle monitoring. Tests show reliable results even with plates of different designs and backgrounds. Future improvements will focus on handling images taken at unusual angles and improving recognition in poor lighting conditions.

Keywords: Optical Character Recognition-OCR, Automatic Number Plate Recognition - ANPR, Open Source Computer Vision- OpenCV

I. INTRODUCTION

With the rapid expansion of cities and the increasing number of vehicles on roads, managing traffic efficiently and ensuring security has become a major challenge. Law enforcement agencies and administrative bodies often struggle with manually tracking vehicles, especially in high-traffic areas or during security operations. Traditional methods rely on human monitoring, which is time-consuming, prone to errors, and requires significant manpower.

To address these challenges, Automatic Number Plate Recognition (ANPR) systems have emerged as a powerful solution for traffic monitoring, security enforcement, and vehicle tracking ([6], [7]). ANPR technology automates the process of identifying vehicle number plates, reducing human involvement while enhancing accuracy and efficiency. From smart parking systems and toll collection to crime prevention and law enforcement, ANPR plays a crucial role in modern urban infrastructure. The global ANPR market, valued at USD 1.78 billion in 2016, is projected to reach USD 3.57 billion by 2023 ([5]).

1.1 The Need for Automation

Existing methods of vehicle identification often require officers to manually log license plate numbers, which becomes difficult during rush hours or high-speed vehicle chases. Additionally, monitoring illegally parked vehicles and tracking those involved in traffic violations remains a complex task without automation ([1], [4]). With the falling costs of high-

resolution surveillance systems and improvements in computer vision techniques, AI-driven number plate recognition offers a promising alternative to traditional approaches ([8]).

1.2 Challenges in Number Plate Detection

One of the biggest challenges in developing a universal ANPR system is the variation in number plate designs—different countries, and sometimes different states, use distinct fonts, sizes, colors, and alignments. This diversity complicates the development of a single recognition algorithm that works seamlessly across all cases ([12], [13]). Further obstacles include poor lighting conditions, occlusions, diagonal plate orientations, and background noise, all of which can hinder accurate recognition ([9], [11]).

1.3 Proposed Solution

This paper presents a Python-based ANPR system leveraging OpenCV for image processing and Tesseract OCR for character recognition. The system captures an image of a vehicle, enhances it using various filters, extracts the number plate region, and applies OCR to convert the plate characters into machine-readable text ([3], [5]). The extracted data can then be stored in a database, displayed on a screen, or used for further analysis.

The proposed approach integrates morphological processing, Sobel edge detection, and contour extraction to ensure accurate segmentation of characters. By automating the detection process, this solution reduces manual workload, lowers operational costs, and improves efficiency in vehicle monitoring applications ([14]).

1.4 Scope of the Study

The system is designed for security monitoring, traffic rule enforcement, smart parking systems, and database-driven vehicle identification ([10], [15]). It is adaptable for both standalone applications and integration into broader surveillance networks. Future enhancements will focus on improving detection accuracy in challenging conditions and incorporating deep learning techniques for even more precise recognition ([16]).

II. RELATED WORK

The current vehicle number plate recognition frameworks suffer from several limitations, including constant human involvement, high operational costs, and the need for significant manpower. The proposed system, developed using Python and OpenCV, addresses these drawbacks by offering automation, reduced labor requirements, and the ability to display, crop, and potentially store or process number plate data.

Earlier studies, such as by Anisha Goyal and Rekha Bhatia, highlighted traditional reliance on neural networks for License Plate Recognition (LPR) and proposed alternatives like Gabor filters, OCR, and Vision Assistant for faster and more efficient processing. Another study by Amr Badr and colleagues emphasized the rising necessity of ANPR due to increased vehicular traffic and proposed using morphological operations, histogram manipulation, and edge detection for plate localization, along with neural networks for character recognition.

The paper also includes a brief history of digital image processing, tracing its origins from early 20th-century cable image transmissions to modern applications in medical imaging and astronomy. Image processing involves various stages such as image acquisition, pre-

processing, segmentation, feature extraction, and character recognition. The evolution of this field has been closely tied to advances in computing and data storage technologies.

A feature extraction optical character recognition (OCR) algorithm [1] for number plate recognition, which captures images and retrieves vehicle details. However, it does not specifically address the use of Pytesseract or the Tesseract engine in Python. An OCR algorithm[2] for number plate recognition, emphasizing character segmentation and extraction. While it doesn't specifically mention Pytesseract or the Tesseract engine, it highlights the importance of image processing techniques for effective character recognition. The paper does not specifically mention using Pytesseract [3]or the Tesseract engine for OCR in number plate detection. It focuses on using the SSD model of TensorFlow and KNN algorithm for character recognition in license plates. The paper focuses [4]on automatic number plate extraction using image processing techniques, including noise removal, edge detection, and morphological processing, rather than OCR methods like Pytesseract or the Tesseract engine for number plate detection in Python.
The paper discusses an Optical Character Recognition (OCR)[5] scheme for number plate recognition, emphasizing its role in converting images of number plates into text. While it doesn't specifically mention Python libraries like Pytesseract, the OCR concept is applicable. The paper discusses Optical Character Recognition (OCR) for number plate detection, emphasizing its implementation through neural networks[6]. While it doesn't specifically mention Pytesseract or the Tesseract engine, it highlights the importance of character recognition in ANPR systems. The paper discusses using optical character recognition (OCR)[7] for license plate recognition, focusing on preprocessing and geometrical feature evaluation. While it doesn't specifically mention Pytesseract or the Tesseract engine, these tools can implement similar OCR techniques in Python. The paper utilizes Optical Character Recognition (OCR) to extract numbers from number plates, demonstrating its effectiveness in the ANPDR[8] system. While it doesn't specifically mention Pytesseract or the Tesseract engine, these tools can be integrated for similar tasks.The paper discusses an optical character recognition (OCR) [9] process for license plate recognition using image processing algorithms, but it does not specifically address the use of Pytesseract or the Tesseract engine in Python libraries for number plate detection.
The paper focuses on template matching and bounding box [10] methods for number plate recognition, rather than OCR techniques like Pytesseract. It achieves 80% accuracy in detecting Indonesian car license plates through early processing and segmentation. The paper does not specifically mention OCR [11]using Pytesseract or the Tesseract engine. It focuses on image processing techniques like CCA and various classifiers for number plate detection, particularly under challenging conditions like low light and overexposure. The paper focuses on an efficient algorithm for Indian vehicle number plate recognition[12], achieving 99.07% accuracy. While it does not specifically mention Pytesseract or the Tesseract engine, it addresses challenges in character recognition and scaling. The paper primarily discusses a neural network-based approach[13] for number plate detection and recognition, rather than Optical Character Recognition (OCR) using Pytesseract or the Tesseract engine. It emphasizes image processing and neural networks for character recognition. The paper does not discuss OCR using Pytesseract or the Tesseract engine[14]. It focuses on a character recognition

method using support vector machines and a character positioning algorithm for number plate detection. The paper utilizes an Optical Character Recognition (OCR) [15] module for character recognition from number plates, employing a neural network for high accuracy (98.75%). However, it does not specifically mention using Pytesseract or the Tesseract engine in its methodology.

III. PROPOSED METHODOLOGY

3.1 Problem Formulation

It is seen that the security forces and authorities face problems whenever security forces chase a vehicle or they can't catch a vehicle which broke traffic rules. Authorities find it very hectic on a busy day to log the vehicle numbers manually in a parking lot. So, in order to make the entire process autonomous, we can install this system so as to automatically detect the vehicle which breaks the traffic rules, take a picture of it and store the number in the database so as to fine the respective owner afterwards. The system can be used in parking so as to take the picture of the vehicle and log the vehicle number in the database (or the cloud, if connected to the internet). This technology reduces the unnecessary hectic manual work required on any busy day, saves the labour cost and is far more efficient than humans. The number of any vehicle once obtained as text, can be displayed, saved in the database or can be searched through the entire database for the details. This project is so versatile that it can be used as an entire application once converted to a software or can be used as a part of any big project.

3.2 Project Objective

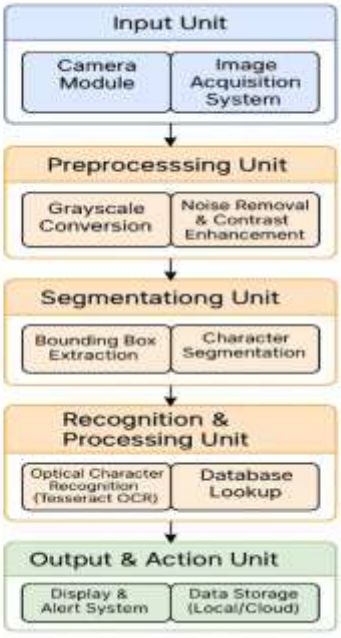

Fig.1 Block Diagram of Input Unit

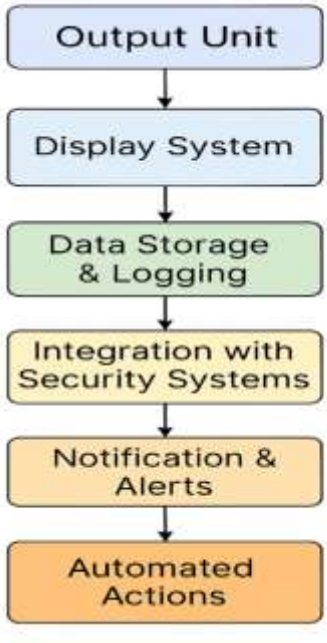

Fig.2 Block Diagram of Output Unit

The Automatic Number Plate Recognition (ANPR) system is designed to efficiently capture, process, and recognize vehicle license plates using a structured approach. As depicted in Fig. 1 (Input Unit) and Fig. 2 (Output Unit), the system follows a sequential workflow to achieve reliable recognition and storage.

In Fig. 1 (Input Unit), the process begins with image acquisition, where a preconfigured camera captures the vehicle's image when the subject (car) enters the frame. The camera is interfaced with a display device, such as a TFT monitor, and is activated via a manual switch or key press. Once the image is captured, it undergoes enhancement and preprocessing, including grayscale conversion to simplify analysis, noise reduction for clarity, and contrast adjustments to highlight the number plate region. The image is then converted to binary scale, allowing contour extraction techniques to efficiently segment the number plate from the background.

Following this, the number plate region is isolated using bounding box detection and morphological transformations, ensuring accurate extraction. The system then applies Optical Character Recognition (OCR) using Pytesseract, converting the plate characters into machine-readable text.

In Fig. 2 (Output Unit), the extracted plate number is processed and stored for further applications. The recognized text can be displayed on a monitor, saved in a local or cloud database, or trigger an actuator response based on predefined conditions. This ensures that the system can be used for vehicle identification, security enforcement, automated toll collection, and access control**.**

IV. METHODOLOGY

In vehicle number plate recognition, a digital image of the vehicle is captured and processed using Python and the OpenCV library, which is widely used for real-time computer vision and image processing tasks. Python's modules like NumPy, SciPy, and Matplotlib enhance its capability to handle large volumes of image data. To improve recognition accuracy, techniques like the median filter are used to reduce noise and emphasize high-frequency features such as edges, which are crucial for detecting the rectangular shape of number plates.

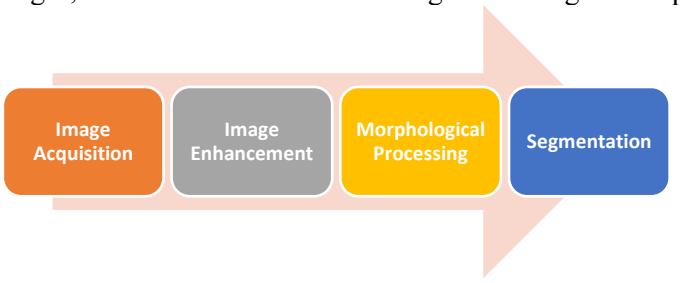

Fig.3 Block diagram of image processing workflow

As shown in Fig.3 The image processing workflow includes:

Image Acquisition – Capturing an image using a camera and converting it into digital form.

Image Enhancement – Improving image clarity by adjusting brightness, contrast, and other features to highlight important details.

Morphological Processing – Converting the image to binary and applying structural transformations to isolate features like number plates.
Segmentation – Dividing the image into distinct regions or objects to enable focused recognition of individual components such as characters on a plate.

These steps collectively enable accurate and efficient number plate detection and recognition.

4.1 Working of the Proposed Methodology

1. In this project, a prototype of Digital Image processing and OCR are executed using different python libraries such as OpenCV 4.2 and pytesseract respectively.
2. The primary camera of the computer is accessed and the image is clicked after any key is pressed, when the vehicle is in the frame. he input image is then fed in the system, for further processing.
3. The Morphed image and the input image and the morphed image is then displayed when any key is pressed. The Morphed image is obtained after morphological transformation.
4. After pressing one more key, the segmented plate is displayed from the morphed image in a new window which is performed using contour extraction on the morphed image.
5. The final step involves performing optical character recognition on the segmented plate using Tesseract engine and a library known as Pytesseract in python. The vehicle number is displayed on the terminal and the plate region is highlighted in a new image in a new window, after pressing one more key.
6. And Also the details related to Owner of vehicle saved in backend on local server is display on the web page.

4.2 Prerequisites of the system

Computer Specification:	Required Packages
Disk space: 1 GB	PYTHON
Operating systems: Windows 7 or later, MacOS, and Linux	OPEN CV2
Python versions: 3.7.5	PYTESSERACT
Included development tools: conda, conda-env, Jupyter Notebook (IPython)	DJANGO
Compatible tools: Microsoft Visual Studio, PyCharm	IMULTIS
Included Python packages: NumPy, SciPy, scikit-learn, pandas, Matplotlib,	
Processors: Intel Core i3 processor or later	
Camera set as primary camera	

4.3 Project Implementation:

1. In this project Digital Image processing and OCR are executed using different python libraries such as OpenCV 2 and pytesseract respectively.
2. And Web App is created using HTML, CSS, JAVA SCRIPT, Django.
3. The primary camera of the computer is accessed and the image is clicked after any key is pressed, when the vehicle is in the frame.
4. The input image is then fed in the system, for further processing.
5. The Morphed image and the input image and the morphed image is then displayed when any key is pressed. The Morphed image is obtained after morphological transformation.
6. After pressing one more key, the segmented plate is displayed from the morphed image in a new window which is performed using contour extraction on the morphed image.
7. The final step involves performing optical character recognition on the segmented plate using the Tesseract library known as pytesseract in python.
8. The details of the Owner are display on the screen.
9. The project is tested on various sets of test data and the system is found to with certain pictures

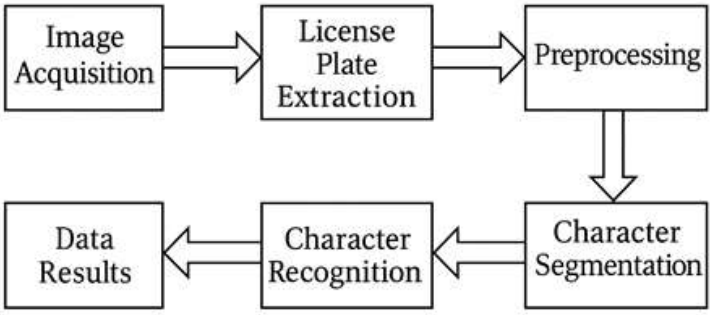

Fig.4 Block Diagram of Implementation Steps

V. RESULTS & DISCUSSION

The results of the ANPR system demonstrate the effectiveness of image processing and OCR-based recognition in extracting vehicle number plates under different conditions. Each step in the detection process contributes to improving clarity, accuracy, and segmentation reliability.

In Fig. 5 (Original Captured Image), the system captures the raw input using a preconfigured camera module. This image serves as the base for the processing pipeline, but due to variations in lighting, angle, and background noise, it requires multiple transformations to isolate the number plate effectively.

Moving to Fig. 6 (Grayscale Conversion), the image undergoes preprocessing to eliminate unnecessary color variations. This step, implemented using OpenCV, ensures that only intensity differences contribute to character extraction. Grayscale processing reduces

computational complexity, making it easier to identify sharp edges and structural boundaries within the number plate.

Fig. 7 (Edge Detection using Sobel/Canny Filter) enhances key contours by detecting sharp intensity transitions across the plate. This method, widely used in computer vision applications, focuses on identifying plate boundaries while removing non-essential details. By applying Sobel or Canny operators, the system successfully highlights structured elements of the image, aiding segmentation.

In Fig. 8 (Bounding Box & Contour Extraction), the system detects and isolates the number plate area using contour detection and morphological filtering. Bounding boxes are drawn around potential plate regions, ensuring precise localization. This approach significantly improves segmentation accuracy by eliminating background clutter and focusing only on relevant sections of the image.

Finally, Fig. 9 (Final Number Plate Recognition using OCR) demonstrates the successful extraction and recognition of characters through Tesseract OCR and Pytesseract integration. The segmented characters are processed and converted into machine-readable text, allowing storage and further analysis. Any distortions or misclassifications are mitigated using template matching, ensuring high recognition accuracy across various conditions.

Fig.5 Original Image

Fig.6 Gray Image

Fig.7 Edge Detection

Fig.8 Region of Interest

Fig.9 Detected Number Plate

VI. CONCLUSION AND FUTURE WORK

The implementation of an Automatic Number Plate Recognition (ANPR) system using OpenCV and Tesseract OCR has demonstrated significant efficiency in accurately detecting and recognizing vehicle license plates. Through a structured approach, incorporating grayscale conversion, edge detection, contour extraction, and optical character recognition, the system achieves high levels of precision in various environmental conditions. The results indicate that preprocessing techniques, such as Sobel and Canny edge detection, improve plate localization, while bounding box segmentation ensures reliable extraction of the number plate region. Furthermore, the integration of Pytesseract enables the system to convert extracted plate characters into machine-readable text, which can be stored or analyzed for enforcement purposes. Despite its successes, challenges such as poor lighting conditions, occlusions, and plate distortions remain areas for improvement.

Future work will focus on integrating deep-learning-based models, such as Convolutional Neural Networks (CNNs), to enhance plate detection accuracy, particularly under challenging imaging conditions. Additionally, optimizing real-time processing capabilities will enable rapid analysis and immediate recognition, making the system more suitable for smart surveillance applications. Expanding OCR capabilities to recognize multilingual number plates will also increase the system's usability across different regions. Further developments will include cloud-based ANPR systems that enable remote data storage and retrieval, making it more effective for law enforcement and automated toll collection. The integration of ANPR with IoT infrastructure will pave the way for smart city applications, facilitating traffic monitoring and vehicle identification with minimal manual intervention. With ongoing

advancements in machine learning and edge computing, the ANPR system is set to become a more intelligent, scalable, and adaptable solution for modern transportation and security needs.

REFERENCES

[1] Bhonsale, T. (2016). Number Plate Recognition and Document Verification using Feature Extraction OCR Algorithm. International Journal of Engineering and Computer Science. https://doi.org/10.18535/IJECS/V5I11.84[2] Bhonsale, T., Dhamal, O., Dhumal, R., Khedekar, P., & Patil, B. (2017). Number plate recognition and document verification using feature extraction OCR algorithm. International Conference Intelligent Computing and Control Systems, 1317–1320. https://doi.org/10.1109/ICCONS.2017.8250683

[3] Khaparde, D., Detroja, H., Shah, J., Dikey, R., & Thakare, B. S. (2018). Automatic Number Plate Recognition System. International Journal of Computer Applications, 179(49), 26–29. https://doi.org/10.5120/IJCA2018917277

[4] Acharya, P., Nandakumar, Y., & H.S, G. (2017). Automatic Number Plate Extraction. Journal of Emerging Technologies and Innovative Research, 4(4), 14–16. http://www.jetir.org/papers/JETIR1704004.pdf

[5] Kashyap, A., Suresh, B., Patil, A., Sharma, S., & Jaiswal, A. (2018). Automatic Number Plate Recognition. https://doi.org/10.1109/ICACCCN.2018.8748287

[6] Arora, M., Jain, A., Rustagi, S., & Yadav, T. (2019). Automatic Number Plate Recognition System Using Optical Character Recognition. 5(2), 986–992. https://doi.org/10.32628/CSEIT1952280

[7] Mani, A., & Panicker, N. J. (2019). An Efficient Method for License Plate Detection and Recognition using OCR. 5(3), 540–547. https://doi.org/10.32628/CSEIT1953155

[8] Shaikh, Z. A., Khan, U. A., Rajput, M. A., & Memon, A. W. (2016). Machine Learning based Number Plate Detection and Recognition. International Conference on Pattern Recognition Applications and Methods, 327–333. https://doi.org/10.5220/0005750203270333

[9] Saghaei, H. (2016). Proposal for Automatic License and Number Plate Recognition System for Vehicle Identification. arXiv: Computer Vision and Pattern Recognition. https://doi.org/10.1016/S0550-3213(01)00405-9

[10] Kusumadewi, I., Sari, C. A., Setiadi, D. R. I. M., & Rachmawanto, E. H. (2019). License Number Plate Recognition using Template Matching and Bounding Box Method. 1201(1), 012067. https://doi.org/10.1088/1742-6596/1201/1/012067

[11] Babbar, S., Kesarwani, S., Dewan, N., Shangle, K., & Patel, S. (2018). A New Approach for Vehicle Number Plate Detection. International Conference on Contemporary Computing, 1–6. https://doi.org/10.1109/IC3.2018.8530600

[12] Mishra, G., & Ahmed, M. B. (2016). Detection of Indian Vehicle Number Plate. Global Journal for Research Analysis, 5(6).

[13] Gala, R., Doshi, S., Jain, Y., Patidar, V., & Nadkarni, S. (2017). Vehicle Number Plate Detection and Recognition. International Journal of Advance Research, Ideas and Innovations in Technology, 3(2). https://www.ijariit.com/manuscripts/v3i2/V3I2-1457.pdf

[14] Singh, B., Kaur, M., Singh, D., & Singh, G. (2016). Automatic number plate recognition system by character position method. International Journal of Computational Vision and Robotics, 6(1), 94–112. https://doi.org/10.1504/IJCVR.2016.073761

[15] Shah, S., Rathod, N., Kaur Saini, P., Patel, V., Rajput, H., & Sheth, P. (2019). Automated Indian Vehicle Number Plate Detection (pp. 453–461). Springer, Singapore. https://doi.org/10.1007/978-981-13-0589-4_42

[16] Gonzalez, Rafael C. Digital image processing. Pearson education india, 2009.