License Plate Number Detection using Python

Anshah Hegde
221IT009
Information Technology
National Institute of Technology
Surathkal-575025
Karnataka
India

Sonali Kannojiya
221IT065
Information Technology
National Institute of Technology
Surathkal-575025
Karnataka
India

Alapati Harshini
221IT006
Information Technology
National Institute of Technology
Surathkal-575025
Karnataka
India

Abstract:

This project presents an innovative system for the automated detection of vehicle number plates, with the primary aim of enhancing vehicle monitoring and security applications. Vehicle number plate recognition plays a significant role in many areas.

In this paper, an efficient and an amazingly simple method is used to recognise the number plate. In the proposed method, OpenCV library along with python language is used for image processing using py tesseract.

Automatic license plate recognition is a technology that relies on optical character recognition methods used on images for reading license plate text. The input image is taken and converted into grayscale image and the processed image is filtered through bilateral filter to remove unwanted characters.

By automating the number plate detection process, this project contributes to improved efficiency and security in various domains, making it a valuable tool for modern transportation and surveillance systems.

Keywords: detection, surveillance, ocr, image processing etc.

I. Introduction

Our main goal is to automate the identification of vehicle number plates using cutting-edge technologies in **image processing and OCR.** By doing so, we aim to improve efficiency and security across various applications, making our system a valuable tool for transportation and surveillance.

We start by using OpenCV, a powerful tool for image processing in Python. The system takes an image and turns it into a grayscale version, making it easier to analyze. To refine the image, we apply a bilateral filter, which helps remove unnecessary elements and ensures a clear view of the number plate region.

To extract text, we've tried OCR techniques like EasyOCR and PyTesseract. In the proposed method we have used PyTesseract. They ensure that our system accurately captures the alphanumeric characters on the plate, ensuring reliable results.

In one of our reference paper implementations, template matching has been used.This technique helps us identify the license plate by comparing a template image (a representation of a typical number plate pattern) with different parts of the processed image.

Our innovative system combines the power of **OpenCV, and PyTesserac**t to create a straightforward and effective approach to license plate recognition.

Section II involves the related work and limitations, **section III** includes proposed methodology, and the methodology is elaborated. **Section IV** presents the comparison of results of different **methods. Section V** includes the results of final methodology. **Section VI** includes conclusion and **section VII** includes references.

II. Related Work

We have drawn references from the literature review in this section.

In paper [1], utilizing OpenCV and Tesseract in Python, the authors employ various image processing techniques, such as thresholding and contour finding, to optimize text extraction. The paper emphasizes adaptability to diverse image characteristics, highlighting its potential applications in traffic enforcement, housing societies, and malls.

In paper [2], the system uses Optical Character Recognition (OCR) and OpenCV for automatic license plate recognition, contributing to intelligent transportation systems. The project involves Digital Image Processing, license plate segmentation, and OCR for text extraction. The proposed system aims for efficiency, speed, and reliability, with a focus on machine learning algorithms and computer vision techniques.

In paper [3], the paper presents a system using image processing techniques, particularly from the OpenCV Python library. The system aims to detect and recognize license plate text from images of occupied parking spaces. Testing is performed on the 500 Image of the Rear View dataset, emphasizing speed and resource efficiency. Challenges include addressing colored license plates and optimizing Optical Character Recognition (OCR) processes.

In paper [4], the system utilizes image preprocessing techniques, including grayscale conversion and Gaussian filtering, for better analysis. The license plate is located through edge detection, morphological treatment, and color segmentation in the HSV model. Character segmentation involves corrosion, threshold filtering, and wavelet analysis, leading to accurate division. Convolutional Neural Network (CNN) is employed for character recognition.

In paper [5], the approach involves image preprocessing steps, such as grayscale conversion, bilateral filtering, and Canny edge detection. The algorithm includes contour detection and filtering, leading to the identification of license plate regions. Tesseract is utilized to extract text information from the license plate image. The software tools employed include OpenCV, Tesseract, and Python.

In paper [6], the paper proposes a vehicle license plate detection and recognition system using image processing techniques. The method involves preprocessing, license plate region extraction, segmentation, and character recognition, utilizing bounding box methods and template matching.

In paper [7], the referenced paper presents a systematic approach for license plate recognition using YOLOv5, involving stages of acquisition, detection, segmentation, and text recognition.

In paper [8], the paper introduces a robust license plate recognition method utilizing Hough transform and horizontal projection profile for efficient and accurate segmentation of characters. Through preprocessing, localization, segmentation, and recognition steps, the proposed approach demonstrates high accuracy in identifying license plate information.

In paper [9], the methodology includes gray-scale conversion, Sobel filtering, thresholding, morphological operations, contour extraction, and Support Vector Machine (SVM) for plate detection, followed by Optical Character Recognition (OCR) segmentation and Artificial Neural Network (ANN) for plate recognition.

In paper [10], the paper proposes an OCR-based system for automated number plate detection and extraction, utilizing edge detection and OCR techniques. It aims to simplify the process for end-users with low technical skills.

In paper [11], the system involves capturing vehicle images, applying grayscale conversion, Otsu's thresholding, noise removal, image cropping, and bounding box features. Optical Character Recognition (OCR) based on template matching is used for character analysis.

In paper [12], the paper presents a fast algorithm for automatic license plate detection tailored for Egyptian plates. The proposed system combines techniques like histogram equalization, Sobel edge detection, Haar-like features, PCA, and Artificial Neural Networks. It addresses challenges such as varying plate sizes and environmental conditions, emphasizing the importance of a high-quality detector for subsequent recognition tasks.

In paper [13], it focuses on improving image graying and binarization algorithms, employing artificial neural networks for feature extraction and prediction. The system achieves effective character segmentation and recognition, demonstrating superior performance compared to traditional approaches in handling diverse license plate conditions.

In paper [14], the paper presents a real-time (ALPR) system that eliminates the Region of Interest (ROI) setting step, improving computational efficiency without compromising recognition performance. The system utilizes YOLOv4 for license plate detection, addressing challenges in recognizing small and unclear license plates, especially in high-speed scenarios. The proposed approach aims for real-time recognition on lightweight embedded devices, specifically using the NVIDIA Jetson TX2 module.

In paper [15], the paper provides a detailed overview of the proposed zone-based license plate identification system, emphasizing the challenges specific to India.
The use of adaptive template matching and zone-based methods adds a novel aspect to the license plate recognition approach.

Although the existing works present efficient methods for license plate recognition, there are a few possible constraints or limitations as mentioned below:

- Many systems target large, clear license plates, but field data often contains small or unclear plates due to factors like high-speed vehicle movement and environmental noise.

- Multiple license plates in an image
- Physical obstructions on license plate
- Lighting conditions
- Plate character font
- Might not be robust when applied to diverse dataset
- Specific training might be required
- Assumption of single row plate

III. Proposed Methodology

INPUT: jpg/png license plate image/car image file
OUTPUT: Detected license plate text

1. **Image Loading and Preprocessing:** Loading the input image, resizing it for standardization, and converting it to grayscale. A bilateral filter is applied to reduce noise, and edge detection is performed using the Canny algorithm.

2. **Contours Detection:** Contours in the edged image are detected using the OpenCV function cv2.findContours. These contours represent potential shapes within the image.

3. **Visualizing Contours on Original Image**: The original image is copied, and the detected contours are drawn on it to aid visualization and understanding of the regions of interest.

4. **Displaying the Original Image Before Drawing Contours:** Another copy of the original image is displayed before drawing any contours to provide a visual reference.

5. **Sorting and Selecting Top Contours:** Contours are sorted based on their areas in descending order, and the top 30 contours are selected for further processing.

6. **Identifying License Plate Contour:** A loop iterates over the selected contours, approximating each contour's shape. If a contour has four corners, it is considered a candidate for a license plate. The script extracts the region of interest (ROI) around the license plate.

7. **Visualizing Contours on Original Image After Drawing:** The original image is again displayed, this time with the selected contours drawn on it, allowing visual confirmation of the identified license plate region.

8. **Displaying and OCR on the Cropped License Plate:** The cropped license plate region is displayed, and OCR is performed using Tesseract to extract text, representing the license plate number.

Libraries and methods used:

1. **OpenCV (cv2):** Open Source Computer Vision Library is a popular open-source computer vision and machine learning software library used for image processing tasks, including reading/writing images, image manipulation, contour detection, and drawing on images.

2. **Imutils:** Imutils is a convenience library built on top of OpenCV, providing a set of utility functions to simplify common tasks and used for resizing images (imutils.resize). It helps in standardizing image sizes.

3. **Pytesseract:** Pytesseract is a Python wrapper for Google's Tesseract-OCR Engine, which is an optical character recognition (OCR) tool.

4. **PIL (Python Imaging Library) / Pillow:** PIL is a library for opening, manipulating, and saving many different image file formats. Pillow is the modern and actively maintained fork of PIL. Used for loading and displaying images (Image module from PIL) as well as saving the cropped license plate image.

5. **Canny:** Canny is an edge detection algorithm that aims to find the edges in an image. Canny to perform edge detection on the grayscale image.

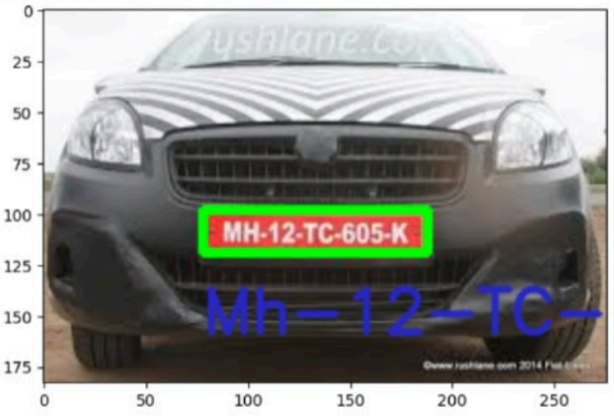
6. **cv2.imshow:** OpenCV function for displaying images in a window.

7. **cv2.imread:** OpenCV function for reading an image from a file.

IV Comparison of proposed method and reference paper implementation results:

1. Image:

Using easyocr[10]:



Using proposed method:

```
Number is: |- MH-12-TC-605-K
```

2. Image:

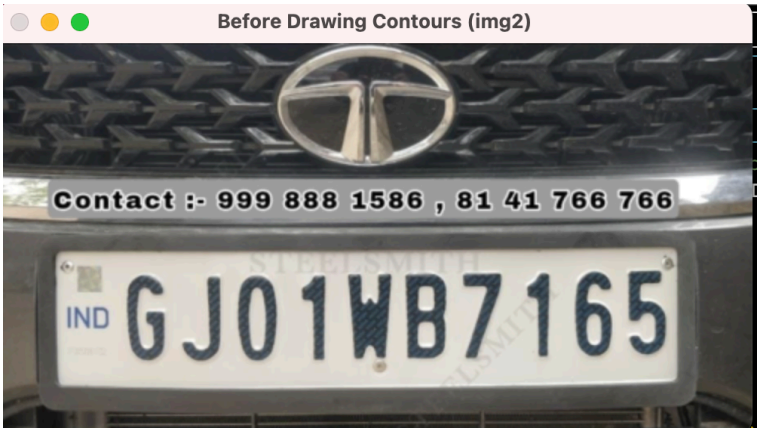
Using pytesseract[5]:

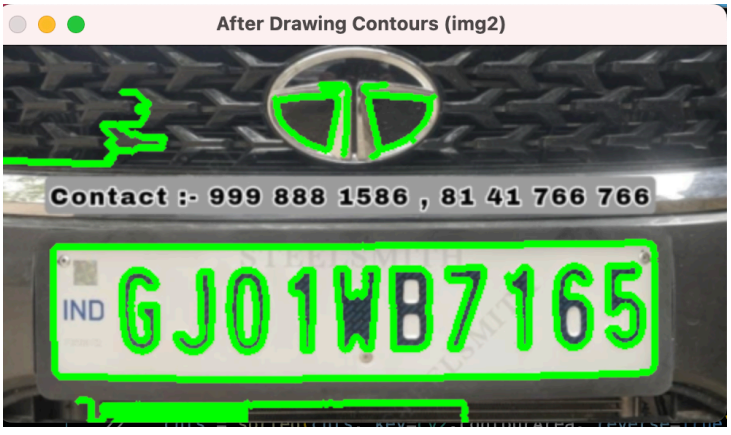
```
License Plate Text:

ROV RECD
```

Using proposed method:

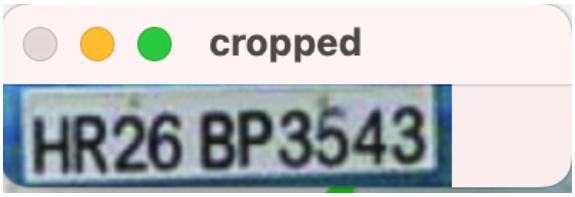
```
Number is: 'KA 03 AB 3289
```

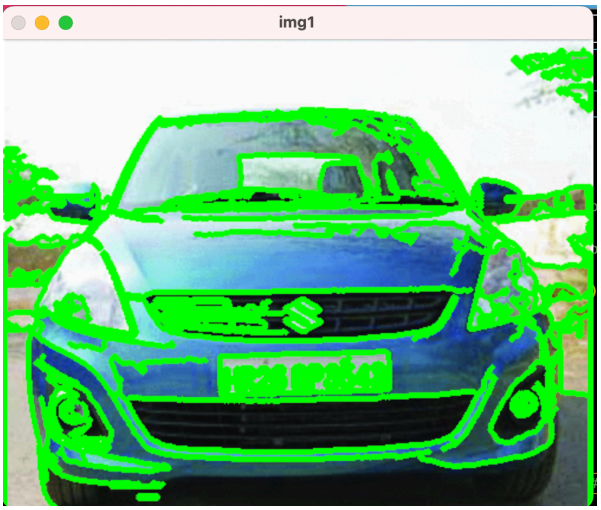
V. RESULTS:



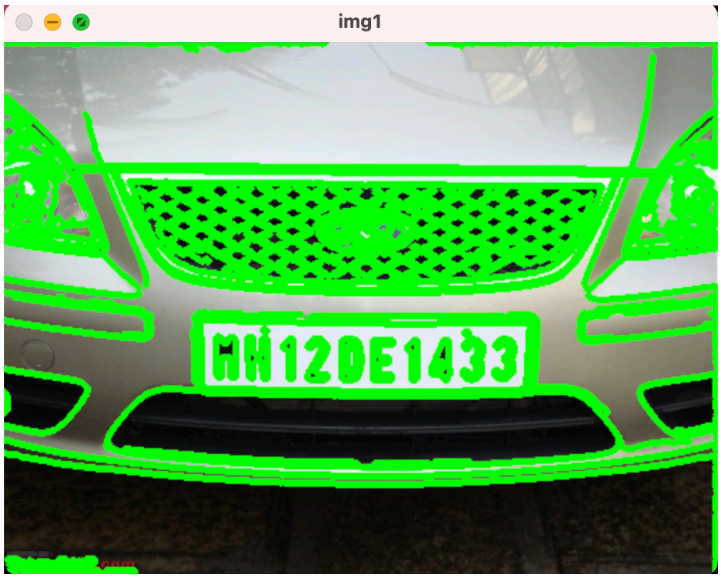


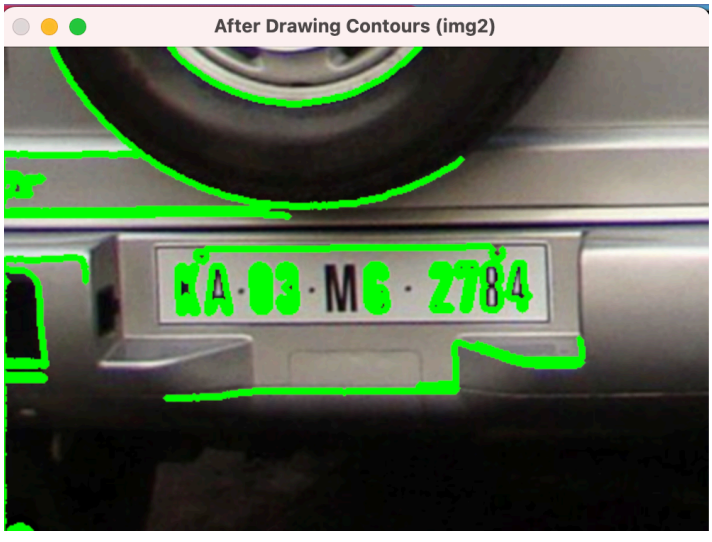

```
Number is: -GJ01WB7 165
```






```
Number is: HR26 BP3543}
```






```
Number is: MH12DE1433
```

```
Number is: 'KA-03-MG - 2784
```

VI Conclusion

The implementation of license plate detection has proven to be a successful venture, offering a technologically advanced solution to various societal challenges. The accuracy and efficiency demonstrated in recognizing license plates showcase the potential of this technology to revolutionize multiple sectors.

License plate detection plays a pivotal role in law enforcement by aiding in criminal investigations and traffic monitoring. It facilitates seamless entry and exit processes in parking facilities, ensuring efficient space utilization. Additionally, its integration into security and surveillance systems bolsters safety measures in various environments.

While celebrating the current success, there remains ample room for improvement and innovation in license plate detection technology. Future enhancements can focus on incorporating deep learning techniques for increased accuracy, achieving real-time processing capabilities, and ensuring robustness against environmental factors.

Expanding the scope to integrate with smart city initiatives, prioritizing privacy safeguards, and supporting multiple languages can further enhance the versatility and ethical considerations of license plate detection systems. Additionally, continuous training, updating, and integration with emerging technologies, such as autonomous vehicles, can solidify the role of license plate detection in shaping the future of transportation and security.

In conclusion, the success achieved in license plate detection is commendable, and the ongoing commitment to improvement and innovation ensures its continued relevance and positive impact on various aspects of modern life.

VII References

[1] R. R. Palekar, S. U. Parab, D. P. Parikh and V. N. Kamble, "Real time license plate detection using openCV and tesseract," 2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2017, pp. 2111-2115, doi: 10.1109/ICCSP.2017.8286778.

[2] K. Garg, K. Gautam, K. Singh and M. Gaur, "License Number Plate Recognition(LNPR)," 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022, pp. 1261-1267, doi: 10.1109/ICAC3N56670.2022.10074115.

[3] M. Kročka, P. Dakić and V. Vranić, "Automatic License Plate Recognition Using OpenCV," 2022 12th International Conference on Advanced Computer Information Technologies (ACIT), Ruzomberok, Slovakia, 2022, pp. 530-535, doi: 10.1109/ACIT54803.2022.9913168.

[4] L. Wang, Y. Huanga, C. Liang, J. Zhou and T. Zhu, "License plate recognition system based on image recognition," 2022 IEEE 21st International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS), Chongqing, China, 2022, pp. 275-279, doi: 10.1109/IUCC-CIT-DSCI-SmartCNS57392.2022.00051.

[5] A. S. Mohammed Shariff, R. Bhatia, R. Kuma and S. Jha, "Vehicle Number Plate Detection Using Python and Open CV," 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2021, pp. 525-529, doi: 10.1109/ICACITE51222.2021.9404556.

[6] K. M. Babu and M. V. Raghunadh, "Vehicle number plate detection and recognition using bounding box method," 2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), Ramanathapuram, India, 2016, pp. 106-110, doi: 10.1109/ICACCCT.2016.7831610.

[7] K. A, B. D. J, S. P. Gandu, J. Hinn T O, R. Kennedy C and A. Immanuel J, "Systematic Number Plate detection using improved YOLOv5 detector," 2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN), Vellore, India, 2023, pp. 1-6, doi: 10.1109/ViTECoN58111.2023.10157727.

[8] P. Prabhakar, P. Anupama and S. R. Resmi, "Automatic vehicle number plate detection and recognition," 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kanyakumari, India, 2014, pp. 185-190, doi: 10.1109/ICCICCT.2014.6992954.

[9] A. Menon and B. Omman, "Detection and Recognition of Multiple License Plate From Still Images," 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET), Kottayam, India, 2018, pp. 1-5, doi: 10.1109/ ICCSDET.2018.8821 138.

[10] S. Ranjan et al., "OCR based Automated Number Plate Text Detection and Extraction," 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2022, pp. 621-627, doi: 10.23919/ INDIACom54597.2022.9763248.

[11] K. Yogheedha, A. S. A. Nasir, H. Jaafar and S. M. Mamduh, "Automatic Vehicle License Plate Recognition System Based on Image Processing and Template Matching Approach," 2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA), Kuching, Malaysia, 2018, pp. 1-8, doi: 10.1109/ICASSDA.2018.8477639.

[12] A. E. Rashid, "A fast algorithm for license plate detection," 2013 International Conference on Signal Processing , Image Processing & Pattern Recognition, Coimbatore, India, 2013, pp. 44-48, doi: 10.1109/ ICSIPR.2013.6497956.

[13] Z. Shuaishuai and P. Chen, "Research on License Plate Recognition Algorithm Based on OpenCV," 2019 Chinese Automation Congress (CAC), Hangzhou, China, 2019, pp. 68-72, doi: 10.1109/ CAC48633.2019.8996599.

[14] J. -Y. Sung, S. -B. Yu and S. -h. P. Korea, "Real-time Automatic License Plate Recognition System using YOLOv4," 2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia), Seoul, Korea (South), 2020, pp. 1-3, doi: 10.1109/ICCE-Asia49877.2020.9277050.

[15] A. Choudhury and A. Negi, "A new zone based algorithm for detection of license plate from Indian vehicle," 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC), Waknaghat, India, 2016, pp. 370-374, doi: 10.1109/ PDGC.2016.7913178.