

Indian Institute of Information Technology Vadodara

Computer Vision Project

License Plate Detection

Glen Quadros

Indian Institute of Information Technology Vadodara
Email: 202051211@iiitvadodara.ac.in

Nishant

Indian Institute of Information Technology Vadodara
Email: 202051127@iiitvadodara.ac.in

Rishi Raj Sachan

Indian Institute of Information Technology Vadodara
Email: 202051158@iiitvadodara.ac.in

Deepak

Indian Institute of Information Technology Vadodara
Email: 202052312@iiitvadodara.ac.in

Abstract—The abstract provides a brief overview of the license plate recognition system implemented using computer vision techniques and the EasyOCR library.

I. INTRODUCTION

The introduction section introduces the purpose of the project and gives an overview of the code components. It emphasizes the significance of license plate recognition in various applications, such as traffic management, security, and automation.

II. MOTIVATION

The motivation section provides a deeper insight into the motivation behind implementing the license plate recognition system. It discusses real-world scenarios where such a system can be beneficial, such as automated toll collection, parking management, and law enforcement.

III. RELATED WORK

This section briefly reviews existing literature and projects related to license plate recognition. It discusses different approaches, algorithms, and technologies used in similar systems, highlighting the advancements and challenges in the field.

IV. CODE COMPONENTS

- 1) **Image Preprocessing:** The input image is read and a grayscale version is created. A bilateral filter is applied to enhance the quality of the image.
- 2) **Edge Detection:** Canny edge detection is applied to identify edges in the image, crucial for subsequent contour detection.
- 3) **Contour Detection:** Contours are detected using the `cv2.findContours` function with a tree hierarchy.

This section explains the importance of contour detection in isolating objects within an image.

- 4) **Polygonal Approximation:** The Douglas-Peucker algorithm is applied to approximate the contour as a polygon, simplifying the shape for efficient processing.
- 5) **Masking and Cropping:** A mask is created to isolate the identified license plate region. This step is crucial for focusing on the region of interest.
- 6) **Text Recognition using EasyOCR:** The cropped license plate region is fed into EasyOCR for text recognition. This section discusses the advantages of using EasyOCR for accurate optical character recognition.
- 7) **Output:** The recognized license plate number is printed, providing the final result of the system.

V. ALGORITHM SIGNIFICANCE

This section delves deeper into the significance of each algorithm used in the code:

- **Bilateral Filter:** Preserves edges while reducing noise, enhancing the quality of the image for subsequent processing.
- **Canny Edge Detection:** Identifies edges in the image, crucial for contour detection and localization.
- **Contour Detection:** Hierarchical contour extraction facilitates the identification of objects and regions within an image.
- **Polygonal Approximation:** Simplifies the contour to a polygon, reducing the number of vertices for efficient processing.
- **EasyOCR:** Efficient and accurate optical character recognition, enabling the extraction of text from images.

VI. IMAGES

A. Original Image (Img1)



Fig. 1. The original image before license plate detection.

B. Result after Detection (Img2)



Fig. 2. The result after license plate detection.

VII. RESULTS AND EVALUATION

This section presents the outcomes of the license plate recognition system, providing an in-depth analysis of its performance and accuracy. The quantitative evaluation incorporates metrics such as precision, recall, and F1 score to assess the efficacy of the implemented system. Despite variations in accuracy that fall below the desired benchmark, there is potential for improvement. Integration of a Convolutional Neural Network (CNN) model, coupled with rigorous training, holds promise for enhancing accuracy. Furthermore, consideration is given to the implementation of a YOLO (You Only Look Once) model, which is anticipated to yield optimal results.

VIII. CONCLUSION

In conclusion, the implemented license plate recognition system showcases commendable performance in accurately identifying and extracting license plate information. The system's robustness and adaptability are evident in various real-world scenarios. While the current implementation achieves its objectives effectively, continuous refinement and optimization remain essential for addressing potential challenges and improving overall efficiency.

IX. FUTURE WORK

The future work section outlines strategic directions for enhancing the license plate recognition system.

We envision a comprehensive training regimen for a Convolutional Neural Network (CNN) model, leveraging an extensive dataset to optimize the system's performance. This approach aims to elevate accuracy levels and address potential limitations observed in the current implementation. Rigorous training over diverse scenarios is anticipated to enhance the system's adaptability and robustness.

Furthermore, exploring the implementation of such models beyond license plate recognition is considered. Integrating these advanced models into systems such as traffic signal management holds potential for broader applications and increased operational efficiency. This exploration aligns with our commitment to continuous improvement and innovation in computer vision technologies.

X. REFERENCES

REFERENCES

- [1] OpenCV Documentation. <https://docs.opencv.org/>
- [2] EasyOCR Documentation. <https://github.com/JaidedAI/EasyOCR>
- [3] imutils Documentation. <https://github.com/jrosebr1/imutils>
- [4] LaTeX Project. <https://www.latex-project.org/>
- [5] IEEE Citation Style Guide. <https://ieeeauthorcenter.ieee.org/author-resources/ieee-citation-reference/>
- [6] NumPy Documentation. <https://numpy.org/doc/stable/>
- [7] Matplotlib Documentation. <https://matplotlib.org/stable/contents.html>
- [8] Python Official Documentation. <https://docs.python.org/3/>
- [9] OpenCV Python Tutorials. https://docs.opencv.org/master/d6/d00/tutorial_py_oop.html