SMART LICENSE PLATE RECOGNITION SYSTEM

IMAGE PROCESSING

DEVELOPER:

YUSUF TAHA ÖNCÜ

TABLE OF CONTENTS

1. ABSTRACT

With the rapid increase of vehicles, automated license plate recognition (LPR) systems are critical for traffic management, security, and parking automation. This project implements a **Smart License Plate Recognition System** leveraging Python and OCR techniques to detect and extract license plate text from images. The system processes multiple images, detects license plates via computer vision, and reads plate numbers with EasyOCR. Results show effective recognition on clear images, demonstrating potential for practical applications in car-centric urban environments like Istanbul. The report outlines system design, implementation, evaluation, and future improvements for robustness and real-time operation. We aimed to address an issue that is inseparable from our daily lives. Especially in car-centered cities like Istanbul, it is crucial for such technologies to help maintain order. Our motivation is to create solutions that unleash our inner productivity and that is exactly what The Smart License Plate Recognition System is all about.

2. INTRODUCTION

Increasing urbanization and vehicle numbers create challenges in traffic control and law enforcement. Automated License Plate Recognition (LPR) systems enable fast, accurate vehicle identification, supporting parking systems, toll collection, and security monitoring. Istanbul, a densely populated city with heavy traffic, can greatly benefit from smart LPR technologies.

This project develops a basic but functional LPR system using **open-source Python tools**, focusing on image-based text detection via OCR. It demonstrates how computer vision and machine learning can automate plate reading, facilitating better traffic and parking management. The system reads multiple vehicle images, detects license plates with bounding boxes, and extracts plate texts for downstream uses.

3. DEFINITION OF THE PROBLEM, PURPOSE, AND GOALS

3.1 Problem Definition

Manual monitoring of vehicle license plates is inefficient and prone to errors. Automated systems are necessary to reliably identify and manage vehicles, yet commercial LPR solutions can be costly or complex. Our project addresses the need for an accessible, open-source approach that can recognize license plates from images effectively.

3.2 Purpose of the Project

To design and implement a license plate recognition system that:

- Processes multiple images of vehicles.
- Detects license plates using OCR.
- Draws bounding boxes on plates for visualization.
- Extracts and displays plate numbers in a readable format.

3.3 Goals Achieved

- Developed an OCR-based license plate text recognition model.
- Enabled batch processing of uploaded images.
- Applied computer vision to localize license plates visually.
- Provided clear output of recognized plate numbers.

4. METHODS AND TECHNOLOGIES USED

4.1 Programming Language

- Python 3.x: Main language for implementing all components.

4.2 Libraries and Frameworks

- **EasyOCR:** Optical character recognition for text detection.
- **OpenCV:** Image processing, bounding box drawing.
- **Matplotlib:** Image visualization.
- **Google Colab:** Development environment and user interface.

4.3 Computer Vision Techniques

- Text detection using OCR with confidence thresholding.
- Bounding box calculation and rendering around detected license plates.
- Support for various fonts and illumination conditions.

4.4 System Architecture

- User uploads one or more vehicle images.
- Images processed by EasyOCR to extract text elements.
- OpenCV draws bounding boxes on high-confidence text regions.
- Recognized plate numbers displayed alongside annotated images.

5. RESULTS OBTAINED AND COMMENTS

5.1 Functional Results

- Successfully detected license plates and extracted texts in most test images.
- Bounding boxes correctly localized the plates.
- OCR handled different fonts and environmental conditions with reasonable accuracy.

5.2 Technical Comments

5.2.1 Strengths

- Open-source implementation easily extensible.
- Batch image processing capability.
- Visual feedback through bounding boxes improves interpretability.

5.2.2 Limitations Identified

- Reduced accuracy under poor lighting, shadows, or skewed angles.
- Basic OCR limitations restrict performance on low-quality images.
- No real-time video processing yet; only static images supported.

5.2.3 Performance Metrics

- Average recognition confidence threshold set at 0.4.
- Processing time varies with image size and count but suitable for prototype use.

6. FUTURE WORKS AND IMPROVEMENTS

- **Image Preprocessing:** Apply grayscale, noise reduction, and contrast enhancement for better OCR input.
- **Custom OCR Training:** Fine-tune models on labeled license plate datasets for improved accuracy.
- **Real-Time Video Integration:** Enable live camera feed processing for real-world deployments.
- **Object Detection Integration:** Use models like YOLO to precisely locate plates before OCR.
- **Expanded Dataset Support:** Broaden system to handle multiple countries' plate formats.

7. CONTRIBUTION OF TEAM MEMBERS

Yusuf Taha ÖNCÜ is responsible for all the work.

8. CONCLUSION

This project presents a functional license plate recognition (LPR) system based on Optical Character Recognition (OCR), developed using Python and open-source libraries such as OpenCV and Tesseract. The system can process multiple images, detect license plates, and extract text with reasonable accuracy. Its design offers a solid starting point for applications in traffic management, security monitoring, and smart city technologies.

One of the key advantages of the system is its use of open-source tools, which makes it accessible, cost-effective, and easy to modify for future improvements. The results show that the system performs well with high-quality images taken in clear conditions, confirming its potential for real-life use cases.

However, its performance is still limited when dealing with poor lighting, complex angles, or low-resolution images. These limitations suggest that additional image preprocessing steps—such as contrast adjustment, noise reduction, or edge enhancement—are needed to improve reliability. Moreover, integrating advanced object detection models like YOLO or EfficientDet could significantly enhance the accuracy of license plate localization.

Future developments could focus on real-time video analysis, which would allow the system to work in dynamic environments such as live traffic feeds. Adding machine learning-based text recognition and support for different license plate formats could also improve flexibility and performance. In addition, deploying the system on mobile or web platforms could expand its range of practical applications.

In summary, this project highlights the potential of open-source software in addressing real-world problems. While it functions well as a basic prototype, it also lays the groundwork for future improvements toward creating a more powerful, accurate, and real-time license plate recognition system suitable for real-world deployment.

9. REFERENCES

• EasyOCR GitHub Repository: https://github.com/JaidedAI/EasyOCR
• OpenCV Documentation: https://docs.opencv.org/
• Google Colab Documentation: https://colab.research.google.com/notebooks/intro.ipynb
• Python Official Documentation: https://docs.python.org/3/