PROJECT REPORT

Prepared By: Ikramullah Khan, Sabtain khan
Registration Id: B22F0256AI097, B22F0827AI069

Project: Automatic Number Plate Recognition System Project Analysis

Introduction

This report analyzes an Automatic Number Plate Recognition (ANPR) system implemented in Python. The application uses computer vision techniques to detect license plates in images through a multi-stage pipeline that includes image preprocessing, edge detection, contour analysis, and plate extraction. The system features a user-friendly graphical interface built with Tkinter and visualization capabilities through Matplotlib. While the implementation successfully identifies license plate regions, it does not include optical character recognition (OCR) for reading the actual plate numbers.

System Architecture

The ANPR application follows a clear, modular structure with distinct processing stages:

1. **Image Acquisition**: Users can upload images through a file dialog interface
2. **Image Preprocessing**: Enhances image quality through grayscale conversion, bilateral filtering, and contrast improvement
3. **Edge Detection**: Combines multiple edge detection methods (Sobel, Canny) to identify boundaries
4. **Contour Analysis**: Identifies potential license plate regions based on geometric features
5. **Plate Extraction**: Performs perspective transformation to obtain normalized plate images
6. **Result Visualization**: Displays processing steps and detected plates in the GUI

Implementation

Image Preprocessing

The implementation uses several effective image enhancement techniques such as the bilateral filter effectively removes noise while preserving edges, and the CLAHE (Contrast Limited Adaptive Histogram Equalization) enhances contrast in local regions, making plate features more distinguishable.

Edge Detection

The system employs a robust approach to edge detection by combining multiple methods:

- Sobel operators in x and y directions to detect gradients
- Canny edge detection with multiple threshold sets
- Morphological operations to close gaps in the edges

License Plate Localization

The plate detection algorithm uses a sophisticated filtering mechanism based on multiple geometric features:

1. **Shape approximation**: Identifies contours with 4-8 points (reasonable for rectangular plates)
2. **Aspect ratio**: Filters for width-to-height ratios between 1.0 and 8.0
3. **Extent**: Ensures the contour area fills a significant portion of its bounding rectangle
4. **Solidity**: Verifies the contour is mostly convex

Each potential plate is scored using a weighted combination of these features:

score = 0.6 * extent + 0.4 * solidity

Plate Region Extraction

For detected plate regions, the system performs perspective transformation to obtain a normalized view:

M = cv2.getPerspectiveTransform(src_pts, dst_pts)

warped = cv2.warpPerspective(original_img, M, (width_rect, height_rect))

This transformation is crucial for handling tilted or skewed license plates in the original image.

User Interface

The application provides a clean and functional GUI with:

- Image upload functionality
- Processing controls
- Status indicators
- Visualization of processing steps using Matplotlib integration

Strengths and Limitations

Strengths

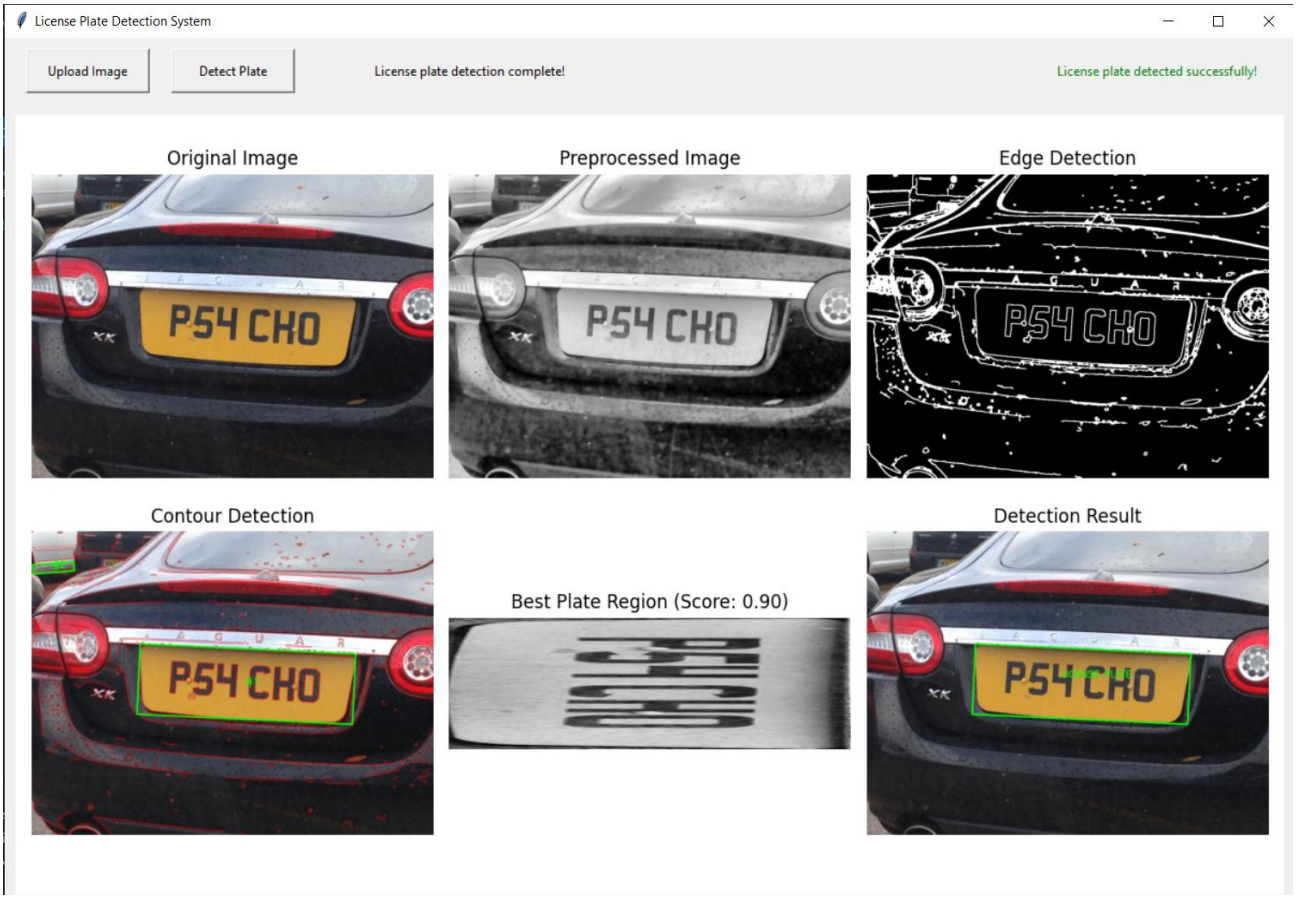
1. **Robust Preprocessing**: The multi-stage preprocessing pipeline effectively handles varying image qualities
2. **Advanced Contour Analysis**: The system uses multiple geometric criteria to accurately identify plate regions

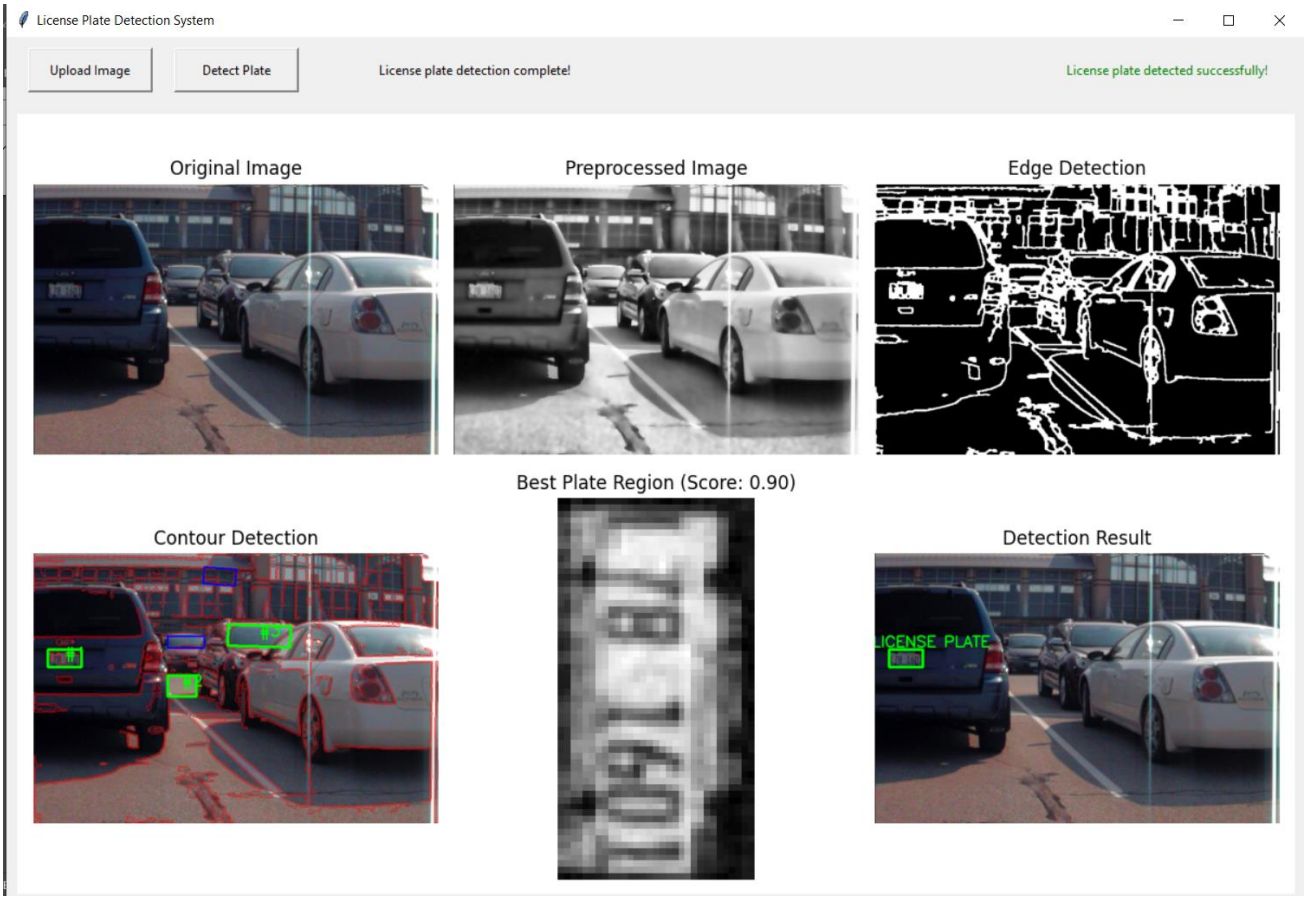
3. **Visual Feedback**: Comprehensive visualization of processing steps aids in understanding and debugging
4. **User-Friendly Interface**: Simple controls make the application accessible to non-technical users

Limitations

1. **No OCR Implementation**: The system detects plate regions but does not extract the actual alphanumeric characters
2. **Limited Real-Time Processing**: The current implementation processes static images only, not video streams
3. **Single-Plate Focus**: The system prioritizes the highest-scoring plate candidate but may struggle with multiple plates
4. **Dependency on Geometric Assumptions**: Relies heavily on rectangular shape assumptions which may not hold for all plate styles

Output:





Conclusion

The ANPR application demonstrates a well-structured approach to license plate detection using classical computer vision techniques. The implementation shows good understanding of image processing fundamentals and provides a solid foundation for a complete ANPR system. With the addition of OCR capabilities and performance optimizations, this project could be developed into a more comprehensive solution for practical license plate recognition applications.