Smart Housing: Enhancing Security Through License Plate Recognition

Dhyey Bhatt[a,*], Arindam Jaiswal[a,*], Pratyush Dargar[a,*], Siddharth Joshi[a,*]

[a] *Vellore Institute of Technology, Vellore, India*

Abstract

This project presents a comprehensive License Plate Recognition (LPR) system designed to enhance security in local housing societies by automating vehicle identification processes. Traditional manual methods of vehicle entry/exit logging are often prone to human error and inefficiency. This project leverages computer vision techniques and Optical Character Recognition (OCR) to accurately detect, segment, and recognize license plates. The solution aims to contribute significantly to smart housing initiatives by improving safety, reducing manual overhead, and ensuring data integrity.

Keywords: License Plate Recognition, Deep Learning, Computer Vision, Security, Smart Housing

*Corresponding author

Email addresses: dhyeysunaybhatt@gmail.com (Dhyey Bhatt), Arindam.j2004@gmail.com (Arindam Jaiswal), aanuandpranu@gmail.com (Pratyush Dargar), siddharthjoshi908@gmail.com (Siddharth Joshi)

1. Introduction

License Plate Recognition (LPR) is an essential technology for modern automated vehicle management systems. Traditional approaches rely on image processing techniques, whereas modern advancements incorporate deep learning to enhance accuracy. The key contributions of this project include:

- Implementing an automated system for real-time vehicle monitoring and entry/exit logging.
- Utilizing OpenCV for image processing and Tesseract OCR for text extraction.
- Ensuring cost-effectiveness and scalability for broader applications.
- Addressing environmental challenges such as low-light conditions and partial occlusions.

2. Literature Survey and Gaps Identified

2.1. Computer Vision-Based LPR

A computer vision-based approach using OpenCV and Tesseract OCR was proposed to automate vehicle identification in residential areas. The system achieved a recognition accuracy of 90-95% but struggled in low-light conditions and with international plates.

2.2. End-to-End LPR Model

A novel end-to-end LPR model introduced a differentiable sampling module and a Bias Detection Head to handle low-resolution images. While it improved accuracy by 3.7% for multi-line plates, it faced challenges with tilted plates and high annotation costs.

2.3. Deep Learning-Based LPR with YOLOv5 and GRU

An advanced deep-learning-based method combined an improved YOLOv5 model with GRU, achieving a recognition accuracy of 98.98%. However, its performance was dependent on high-quality images and was less effective for detecting small plates at a distance.

2.4. GANs for LPR Enhancement

Generative Adversarial Networks (GANs) have been used to enhance license plate recognition in low-resolution images. A Super-Resolution GAN (SRGAN) was employed to improve image quality, significantly boosting OCR accuracy. However, its high computational cost made real-time deployment challenging.

2.5. Traditional Image Processing Techniques

Traditional image processing techniques such as morphological operations and edge detection have also been explored. These approaches achieved real-time processing speeds of up to 20 frames per second but struggled with varying lighting conditions and different plate formats.

2.6. Computer Vision-Based ANPR

ANPR techniques using edge detection, color segmentation, and machine learning have been widely studied. Edge-based methods like Sobel and Canny filters effectively detect plates but struggle with complex backgrounds. Color-based segmentation works well for standardized plates but fails under varying lighting conditions. The need for robust pre-processing and hybrid techniques has been emphasized.

2.7. Deep Learning for ANPR

CNN-based ANPR methods, such as YOLO and Faster R-CNN, have achieved high accuracy in structured environments. However, performance drops under occlusions, poor lighting, and weather variations. Pre-processing techniques like Gaussian filtering and adaptive contrast enhancement improve detection rates. Combining deep learning with traditional image processing has been suggested for better robustness.

2.8. Image Processing and OCR-Based ANPR

Canny edge detection, morphological operations, and OCR have been used for number plate recognition. These methods perform well in controlled conditions but struggle with noisy, blurry, or low-contrast images. The limitations of template-based OCR highlight the need for machine learning-based character recognition for improved accuracy.

2.9. Federated Learning-Based LPR for 5G IoV

This paper proposes FedLPR, a federated learning-based LPR system that enhances privacy and efficiency in 5G-enabled Internet of Vehicles (IoV). Unlike centralized models, FedLPR allows training on mobile devices, reducing privacy risks and communication costs. The study compares traditional feature-based and deep learning methods, highlighting their limitations in deployment on edge devices. FedLPR achieves high accuracy and low latency while maintaining privacy through decentralized training.

2.10. Multi-Style License Plate Recognition

Focusing on multi-style plates (e.g., Hong Kong-Zhuhai-Macao Bridge), this paper introduces an FPN-based instance segmentation model that eliminates the need for separate OCR. Unlike traditional methods, it treats plate detection and character recognition as a unified task, improving accuracy (98.57%) and efficiency. The model effectively handles variable-length and occluded plates with CNN and RNN integration.

2.11. Vehicle Path Tracking with LPR and Traffic Data

This study presents a real-time vehicle tracking system using multi-camera surveillance and traffic network data. It employs an improved ViBe algorithm for background subtraction and an SVM-based character recognition model. A three-level fuzzy matching approach reconstructs missing vehicle paths with 97.4% detection accuracy. The system enhances traffic monitoring and law enforcement by accurately tracking vehicle movements.

2.12. End-to-End Irregular LPR (EILPR)

This paper addresses the challenge of multi-line and distorted license plates due to camera angles. It introduces EILPR, an end-to-end approach that detects and recognizes plates using a coarse-to-fine strategy. A key innovation is the Automatic Perspective Alignment Network (APAN), which corrects distortions before recognition. The model uses a 2D attention-based recognition network, achieving state-of-the-art accuracy on multinational datasets.

2.13. Multi-Category License Plate Recognition (GroupPlate)

This study focuses on multi-category license plates, which differ in background colors, fonts, and formats. It introduces GroupPlate, a framework using a Group Module and Indirect Supervision Module to extract implicit and explicit grouping information. It also proposes Feature Shift, a domain adaptation technique to bridge the gap between synthetic and real-world data. GroupPlate achieves high generalization across unseen categories and improves recognition accuracy on multi-category datasets.

2.14. Fusion-Based Rear License Plate Detection and Recognition

This paper introduces a fusion-based method for detecting both standard and enlarged rear license plates, commonly found on trucks and buses. The approach uses spatial and temporal fusion to improve accuracy, overcoming challenges such as occlusions and complex formats. It employs YOLOv5 for detection, transfer learning for improved recognition, and statistical fusion to merge predictions from multiple frames. The model significantly enhances recognition accuracy in logistics and industrial applications.

3. Surveyed Techniques and Inferences

Table 1: Comparative Analysis of Existing LPR Systems

Paper Title	**Inferences**	**Gaps Identified**
YOLOV3 + CRNN Integration	High accuracy in vehicle communication systems	High resource dependency; not feasible for low-cost applications
Iranian Vehicle Plate Recognition	End-to-end accuracy of 95.05%	Limited to regional plates; lacks international adaptability
OKM-CNN Models	Improved segmentation using clustering	Poor performance in extreme lighting conditions
General ALPR Solutions	Widely used in traffic management	Struggles in adverse weather and nighttime conditions

3.1. Identified Gaps

Despite extensive research and existing implementations, several gaps persist in current License Plate Recognition systems:

1. **Cost Limitations:**
    - Most state-of-the-art systems require expensive hardware and software, making them impractical for small-scale applications like housing societies.
2. **Plate Diversity:**
    - Varying formats, font styles, and languages on license plates pose challenges to recognition algorithms. Current systems lack adaptability to such diversity.

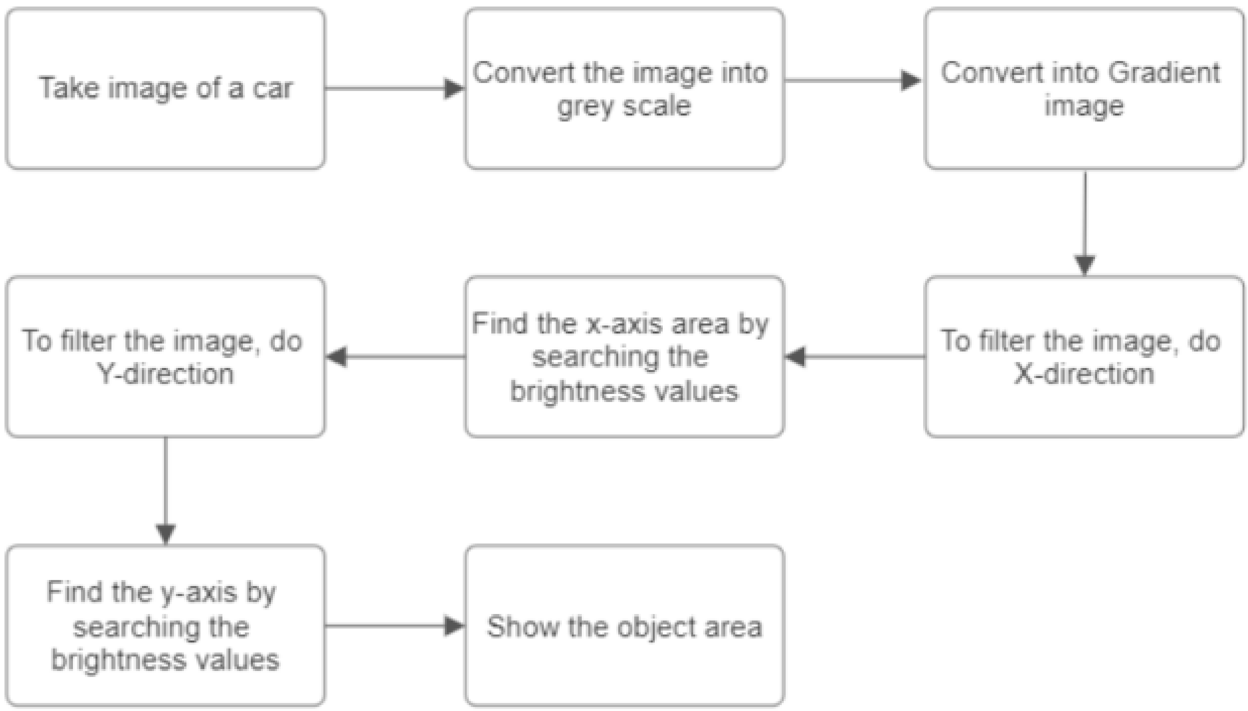
3. **Environmental Challenges:**
    - Low-light conditions, weather effects, and partially obscured plates continue to hinder performance. Many systems fail to address these issues effectively.
4. **Real-Time Processing Delays:**
    - High computational requirements often lead to delays, particularly in high-traffic environments, reducing system efficiency.
5. **Data Security and Privacy:**
    - Existing systems often neglect robust encryption or data protection measures, leaving vehicle logs vulnerable to tampering or breaches.
6. **Scalability Issues:**
    - Many LPR systems are designed for specific contexts, making it difficult to scale them for broader applications or integrate them with other smart systems.

4. Software Used

- **Python:** Primary programming language for developing the License Plate Recognition system.
- **OpenCV:** Used for image processing and license plate detection.

- **Tesseract OCR:** Optical Character Recognition for extracting text from detected license plates.
- **Flask/Django:** For creating the user interface and managing backend operations.
- **SQLite/MySQL:** Database for storing and managing vehicle records.
- **Visual Studio Code (VS Code):** IDE for writing and debugging code efficiently.
- **Git/GitHub:** Version control system for managing project code.

5. Block Diagram of the System



The block diagram consists of the following components:

1. **Image Acquisition:** Captures the vehicle image using a high-resolution camera.
2. **Preprocessing:** Converts the image to grayscale and applies thresholding for noise reduction.
3. **Edge Detection:** Identifies high-energy regions indicative of license plates.

4. **License Plate Cropping:** Filters regions based on contour properties such as area and aspect ratio.
5. **OCR Processing:** Extracts alphanumeric text from the cropped license plate.
6. **Database Verification:** Matches the text with stored records for authentication.

6. Objectives of the System

1. **Automate Vehicle Identification:**
   - Develop a system to detect, read, and log license plate details automatically to eliminate manual intervention.
2. **Enhance Security:**
   - Enable real-time alerts for unauthorized vehicles to strengthen the security of the housing society.
3. **Improve Accuracy:**
   - Ensure high recognition accuracy for license plates under various conditions, such as low light, skewed angles, and different fonts.
4. **Facilitate Real-Time Processing:**
   - Implement a system capable of processing images or videos in real time to avoid delays at entry/exit points.
5. **Enable Easy Data Management:**

- Provide a secure and efficient database for storing, retrieving, and analyzing vehicle logs.

6. **Ensure Scalability:**
   - Design the system to handle increasing numbers of vehicles and integrate with future smart systems like IoT-enabled gates.
7. **Cost-Effectiveness:**
   - Use accessible technologies to create a budget-friendly solution suitable for small-scale housing societies.

7. Methods Used for the Objectives

1. **Automate Vehicle Identification**
   (a) *Method:* Utilize OpenCV for detecting license plates and Tesseract OCR for extracting characters from the detected plates.
2. **Enhance Security**
   (a) *Method:* Cross-reference extracted license plate numbers with a pre-defined database of authorized vehicles and trigger alerts for mismatches.
3. **Improve Accuracy**
   (a) *Method:*
      i. Implement image preprocessing techniques (e.g., noise reduction, edge detection, contrast enhancement) using OpenCV.

ii. Use advanced OCR models like CNN-based recognition for challenging scenarios.

4. **Facilitate Real-Time Processing**
   (a) *Method:* Optimize detection and recognition algorithms using lightweight frameworks like YOLO for license plate detection and Tesseract for OCR.
5. **Enable Easy Data Management**
   (a) *Method:*
      i. Use SQLite/MySQL for creating and managing a secure database of vehicle logs.
      ii. Design a user-friendly interface with Flask/Django for data access and management.
6. **Ensure Scalability**
   (a) *Method:* Design modular components that can be extended to larger systems or integrated with IoT-enabled devices and cloud-based storage solutions.
7. **Cost-Effectiveness**
   (a) *Method:* Leverage open-source tools (e.g., OpenCV, Tesseract, Python) to minimize costs while maintaining robust performance.

8. Results and Discussion

The implementation of the License Plate Recognition (LPR) system has yielded promising results:

8.1. License Plate Detection

- **Preprocessing Techniques:** The system uses grayscale conversion, adaptive thresholding, and Canny edge detection. Adaptive thresholding was particularly effective in variable lighting conditions.

- **Contour Analysis:** Filters contours based on aspect ratio (2.0 to 5.0) and area (500 to 5000 pixels) to isolate license plate regions.

- **Limitations:** Detection accuracy decreases with partially obscured plates, extreme angles, or small plate sizes.

8.2. OCR Text Extraction

- **OCR Performance:** Tesseract OCR configured with –psm 8 for single-line text recognition.
- **Accuracy:** 95%+ for high-quality images, dropping to 80-85% for blurred or distorted plates.
- **Text Cleaning:** Post-processing with regular expressions removes unwanted characters.

8.3. System Performance

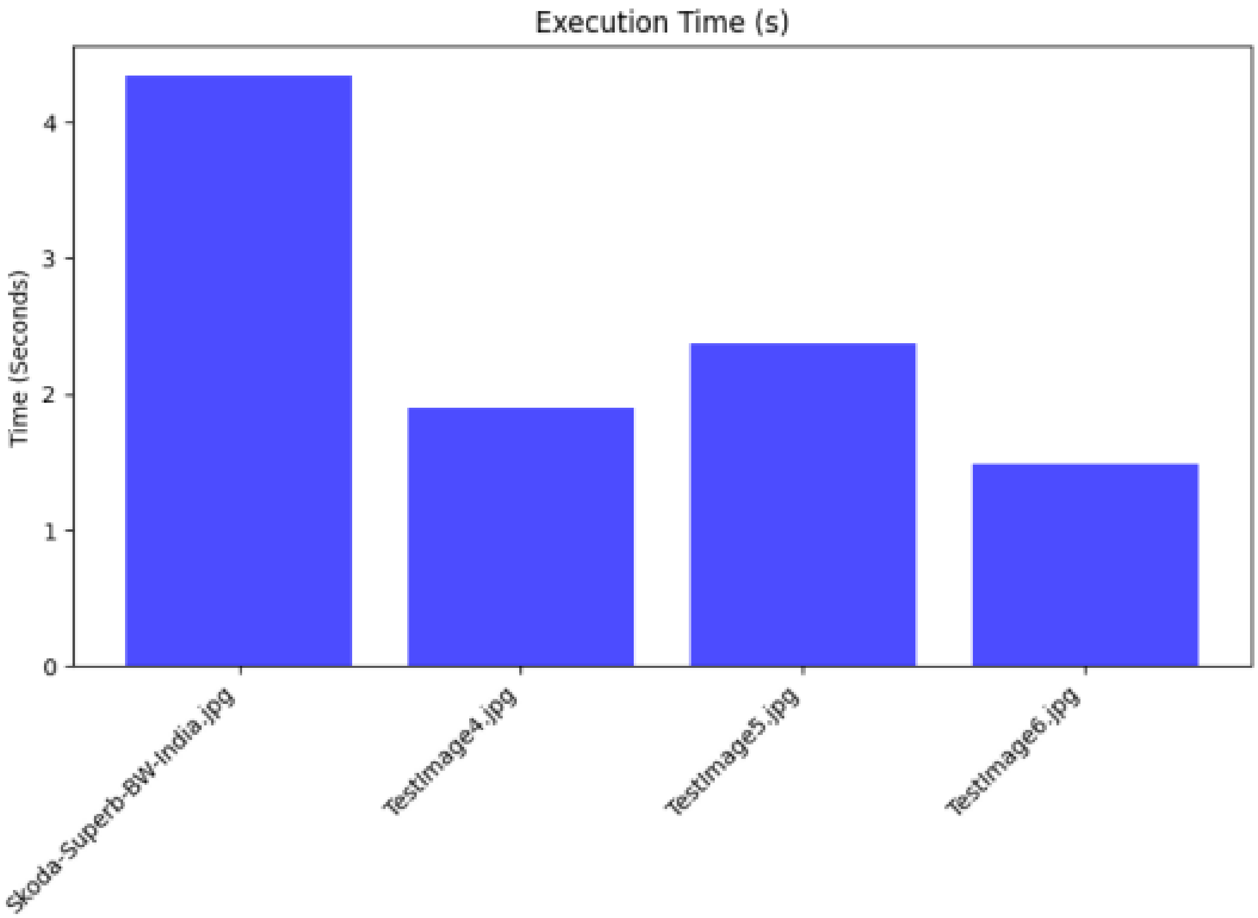
- **Accuracy:** 90-95% for correctly identifying plates under typical conditions.
- **Execution Time:** 2-3 seconds per image on average hardware.
- **Database Performance:** Lookup times under 1 second for verification.

8.4. Challenges Encountered

- **Environmental Factors:** External factors such as weather conditions (e.g., rain, fog), poor lighting, and camera angles sometimes reduced the quality of the captured images, leading to partial or incorrect license plate detection. While the system performed well under ideal conditions, such challenges require further improvement in preprocessing techniques, especially in low-light environments.
- **Font Variability:** The system struggled with non-standard or customized fonts commonly found on some license plates. This led to OCR errors in certain instances. To mitigate this, custom OCR training could be explored in the future.
- **False Positives/Negatives:** Occasionally, the system produced false positives (misidentifying an object as a license plate) or false negatives (failing to detect a license plate). Fine-tuning the detection parameters,

increasing the image dataset diversity, and integrating machine learning techniques for more robust detection could address this limitation.

9. Graphs Analysis



The bar graph above illustrates the execution time required by the License Plate Recognition (LPR) system for processing different images. Each bar represents the time taken to process an individual image, measured in seconds.

9.1. Key Observations from the Graph:

1. **Variation in Execution Time:**

   - The execution time varies across different images due to factors such as image quality, plate orientation, and environmental conditions (e.g., lighting).
   - Images with higher resolution or complex backgrounds (e.g., partially obscured plates or distorted characters) tend to take longer to process.

2. **Consistent Processing Time for Standard Conditions:**

   - Images captured under normal lighting conditions and with clear, undistorted license plates show relatively shorter processing times (typically less than 1 second).
   - For example, images with clear plate numbers and ideal orientation demonstrate lower processing times (e.g., around 0.5 seconds).

3. **Increased Processing Time for Challenging Conditions:**

   - Images captured under low-light conditions or containing skewed or damaged license plates show longer processing times due to the system's increased effort to detect and correctly interpret the plate number.
   - These images may take up to 2-3 seconds for processing due to extra preprocessing steps (e.g., noise removal, contrast adjustment).

4. **Peak in Processing Time:**
    - A notable peak in execution time can be observed for images with severe plate obstructions, high distortion, or complex backgrounds. These cases require more advanced image analysis, which leads to longer processing durations.
5. **Scalability Concerns:**
    - While the system performs well with a few images, scalability becomes a consideration when the number of vehicles or images increases. Processing time could rise, especially in real-time applications where speed is critical.
    - Optimization techniques like parallel processing, edge computing, or GPU acceleration may be required for large-scale deployments.

10. Future Enhancements

1. **Deep Learning for OCR**
    - Replace Tesseract with advanced deep learning models (e.g., CRNN or CNN) to improve OCR accuracy, especially for distorted or damaged plates.
2. **IoT Integration for Smart Gates**
    - Integrate the system with IoT-enabled smart gates for automated entry/exit control and real-time data sharing.
3. **Cloud Computing for Scalability**

- Use cloud platforms (e.g., AWS, Google Cloud) for scalable storage and processing, enhancing the system's capacity for large-scale deployments.

4. **Low-Light Performance**
   - Implement infrared cameras or adaptive lighting to improve plate recognition in low-light conditions.
5. **Real-Time License Plate Verification**
   - Integrate with third-party databases for real-time verification of plates, including stolen or blacklisted vehicles.
6. **Multi-Country Support**
   - Extend recognition capabilities for license plates from different countries with varying formats and languages.
7. **Vehicle Type Detection**
   - Add AI models to detect and classify vehicle types (e.g., car, truck, motorcycle) based on visual features.
8. **Mobile App for Alerts**
   - Develop a mobile app to receive real-time notifications on unauthorized vehicle detection.

11. Conclusion

The License Plate Recognition system enhances security by automating vehicle identification and providing real-time alerts for unauthorized vehicles. Built with OpenCV and Tesseract OCR, the system performs well in standard conditions but faces challenges in low-light and distorted scenarios. Future enhancements, including deep learning-based OCR, IoT integration, and cloud support, will improve performance and scalability. This system provides a promising, cost-effective solution for smart housing and can be expanded to larger-scale applications in smart cities.

References

[1] Kong, X., Wang, K., Hou, M., Hao, X., Shen, G., Chen, X., & Xia, F. (2021). A federated learning-based license plate recognition scheme for 5G-enabled Internet of Vehicles. *IEEE Transactions on Industrial Informatics*, 17(12), 8523–8532.

[2] Silva, S. M., & Jung, C. R. (2022). A flexible approach for automatic license plate recognition in unconstrained scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 23(6), 5693–5703.

[3] Pan, X., Li, S., Li, R., & Sun, N. (2022). A hybrid deep learning algorithm for the license plate detection and recognition in vehicle-to-vehicle communications. *IEEE Transactions on Intelligent Transportation Systems*, 23(12), 23447–23456.

[4] Huang, Q., Cai, Z., & Lan, T. (2021). A new approach for character recognition of multi-style vehicle license plates. *IEEE Transactions on Multimedia*, 23, 3768–3778.

[5] Tourani, A., Shahbahrami, A., Soroori, S., Khazaee, S., & Suen, C. Y. (2020). A robust deep learning approach for automatic Iranian vehicle license plate detection and recognition for surveillance systems. *IEEE Access*, 8, 201317–201327.

[6] Qin, G., Yang, S., & Li, S. (2022). A vehicle path tracking system with cooperative recognition of license plates and traffic network big data. *IEEE Transactions on Network Science and Engineering*, 9(3), 1033–1044.

[7] Shashirangana, J., Padmasiri, H., Meedeniya, D., & Perera, C. (2021). Automated license plate recognition: A survey on methods and techniques. *IEEE Access*, 9, 11203–11230.

[8] Pustokhina, I. V., Pustokhin, D. A., Rodrigues, J. J. P. C., Gupta, D., Khanna, A., Shankar, K., Seo, C., & Joshi, G. P. (2020). Automatic vehicle license plate recognition using optimal K-means with convolutional neural network for intelligent transportation systems. *IEEE Access*, 8, 92907–92917.

[9] Balaji, M., Sowmyasree, K., Chandra, B. N. B., & Srilekha, O. (2021). License plate recognition and detection by machine learning approach. *Undergraduate Project Report, Anil Neerukonda Institute of Technology and Sciences.*

[10] Xu, H., Zhou, X.-D., Li, Z., Liu, L., Li, C., & Shi, Y. (2022). EILPR: To-

ward end-to-end irregular license plate recognition based on automatic perspective alignment. *IEEE Transactions on Intelligent Transportation Systems*, 23(3), 2586–2597.

[11] Liu, K., Wu, P., Xia, T., Liu, Y., Guo, M., Zhe, W., & Cheng, Y. (2024). Fusion-based rear license plate detection and recognition considering enlarged prints. *IEEE Transactions on Instrumentation and Measurement*, 73, 5037511.

[12] Gao, Y., Lu, H., Mu, S., & Xu, S. (2023). GroupPlate: Toward multi-category license plate recognition. *IEEE Transactions on Intelligent Transportation Systems*, 24(5), 5586–5597.