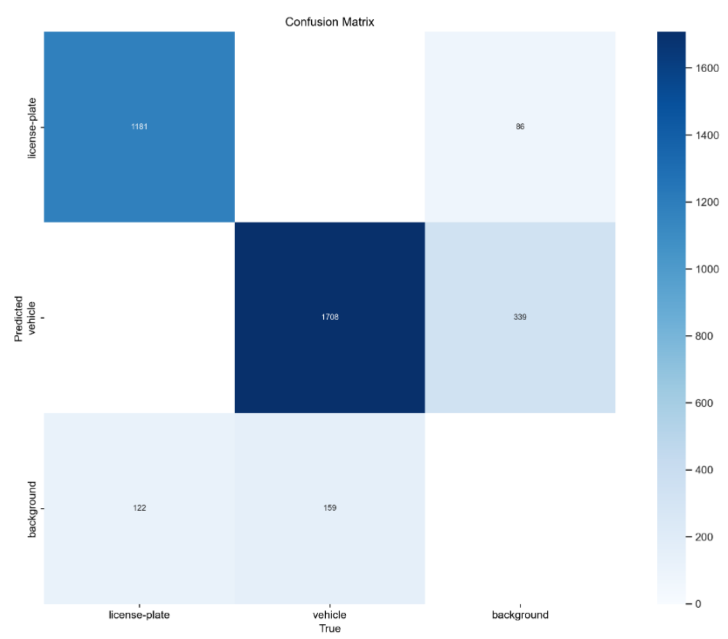
University Car Tracker Software with YOLOv8-Based License Plate Detection and OCR for Improved Violation Management

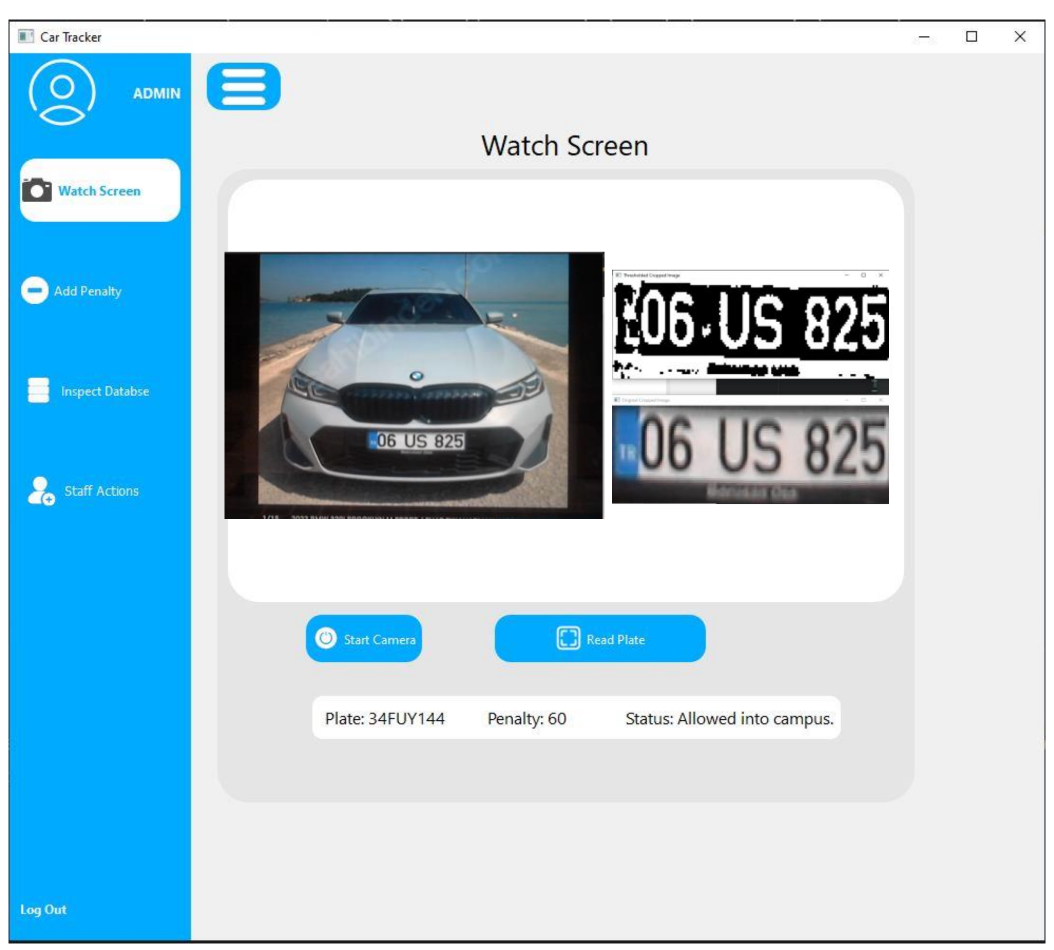
FEYZİYE SCHOOLS FOUNDATION
IŞIK UNIVERSITY

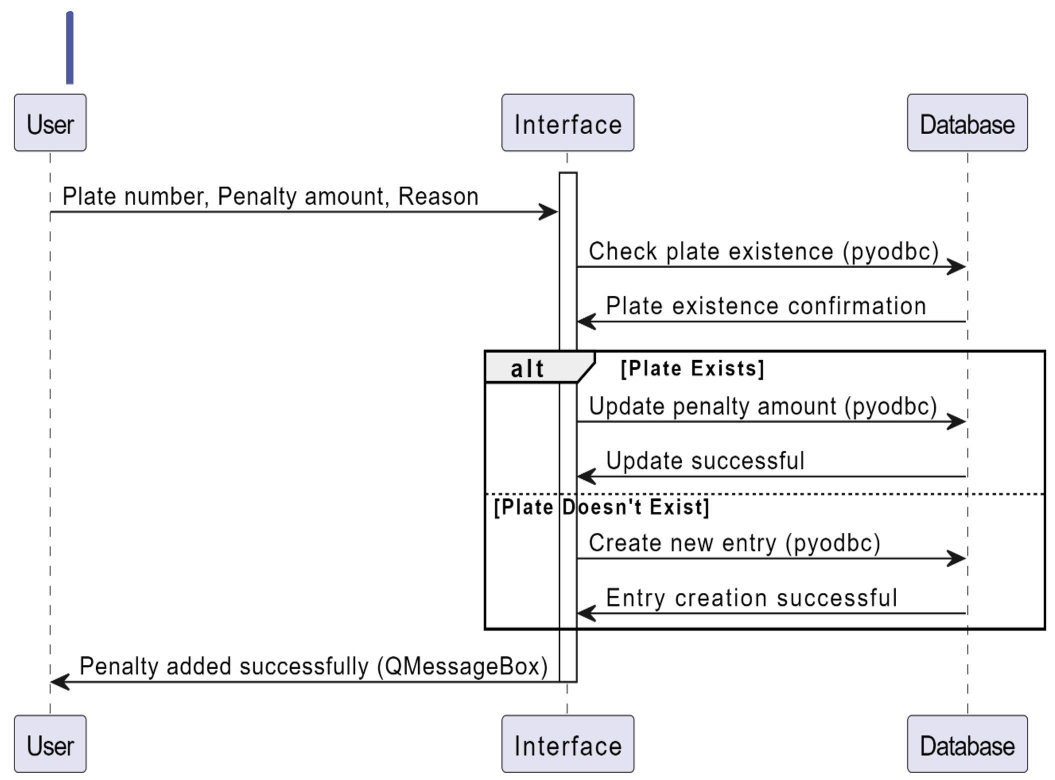
Hüseyin Ateş Öztürk
Thesis Advisor: Dr. Şahin Aydın
Işık University, Management Information Systems

Abstract

This paper proposes a university car tracker software that utilizes deep learning for automated license plate recognition and penalty management. Traditional systems rely on security personnel to manually identify and record violating vehicles, a process prone to errors and inefficiencies. This software addresses these limitations by employing the YOLOv8 deep learning model for real-time car plate detection in video footage captured from campus cameras. EasyOCR then extracts the license plate characters for further processing. The software stores car plate information, penalties, and user data in a secure relational database. It offers functionalities for staff to inspect the database, add penalties, and manage user accounts. The YOLOv8 model was trained on a car plate dataset from Roboflow to ensure it can handle diverse plate formats and lighting conditions. Challenges encountered during development included improving accuracy for specific scenarios and mitigating potential misinterpretations by the OCR engine due to similar-looking characters. To address these, adaptive thresholding and a custom function for character correction were implemented. Overall, the system offers significant advantages over manual approaches by enhancing efficiency, reducing errors, and enabling real-time enforcement of parking regulations. Future enhancements include expanding vehicle type classification, incorporating multi-lingual OCR, and implementing advanced reporting and analytics functionalities.



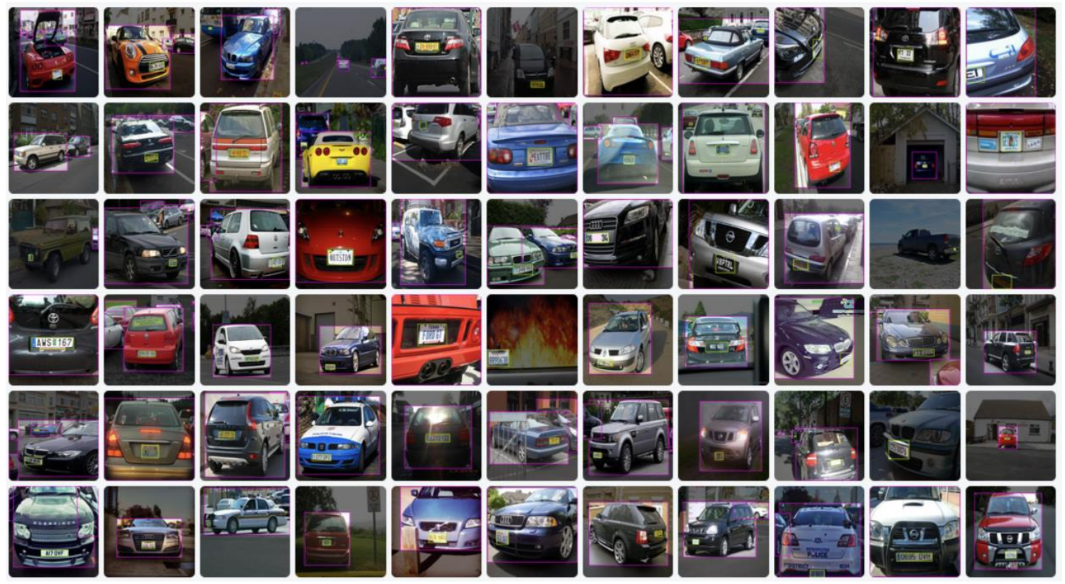




Introduction

Universities often implement license plate penalty systems to ensure safety, security, and a smooth flow of traffic within their campuses. These systems deter violations like illegal parking in designated no-parking zones (e.g., dormitory areas or the road leading to the social center), exceeding speed limits, and excessive exhaust emissions, all of which can pose risks or inconveniences to students, faculty, and staff.

Traditionally, enforcing these penalties relied on security personnel manually identifying and recording license plates of violating vehicles. This process, however, is prone to errors, time-consuming, and inefficient, especially when dealing with a large volume of vehicles. In our university, security guards relied on a written list of fined license plates, which they had to cross-check with parked vehicles one by one – a tedious and error-prone task. Recognizing the limitations of this manual approach, we turned to deep learning to automate the process. Utilizing the YOLOv8 model, we developed a software solution that streamlines license plate penalty enforcement, significantly reducing manual workload and improving overall efficiency.



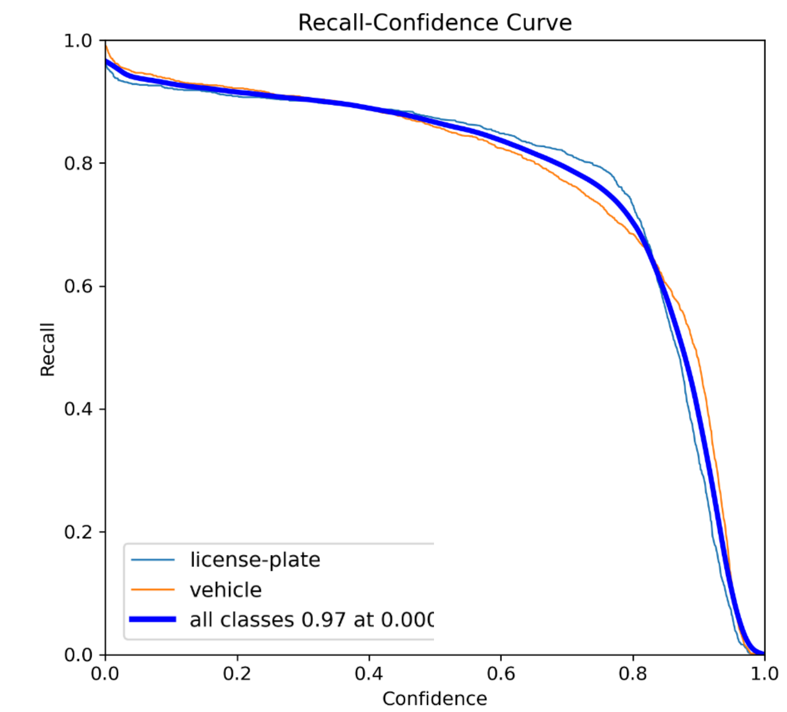
Methodology

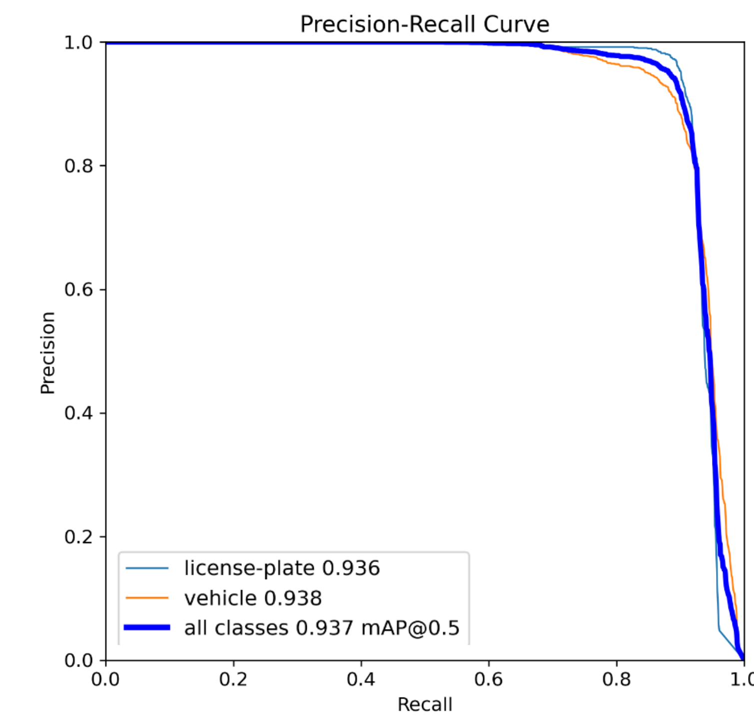
The university car tracker software is designed with a modular architecture, ensuring clear separation of functionalities for ease of development and maintenance. The core components include user login, license plate reading, database inspection, penalty management, and staff actions. Each component is implemented as a separate module, with well-defined roles and responsibilities. The user login module verifies user credentials against a database, providing secure access to the system. The license plate reading module captures video frames using OpenCV, applies the YOLOv8 model for object detection, and employs EasyOCR for character recognition. Detected license plates are then cross-referenced with the database to retrieve and display relevant information, such as penalty status.

The database inspection module allows authorized users to query and view records stored in the plates_penalties and user_data tables. This module supports functionalities like searching, filtering, and refreshing the displayed data to ensure up-to-date information. The penalty management module enables manual entry of penalties by allowing staff to input plate numbers, penalty amounts, and reasons. This module updates existing records or creates new entries in the database, providing feedback on successful operations. The staff actions module, restricted to admin users, manages staff accounts by adding new users and updating their status.

This module ensures proper access control and user management within the system.To integrate YOLOv8 and Roboflow for car plate detection, the training process was carefully planned and executed. Data acquisition involved utilizing a diverse dataset from Roboflow, ensuring a wide representation of license plate styles and lighting conditions. Each image in the dataset was meticulously annotated to create a robust ground truth for the model. The annotated data was then exported in a format compatible with YOLOv8 training, which included labeled images and corresponding text files specifying bounding box coordinates and class labels. The training parameters were defined within a Python script, and the model was trained, validated, and evaluated to ensure high detection accuracy.

Challenges encountered during development, such as accuracy improvement and similar character recognition, were addressed with specific solutions. Adaptive thresholding was applied to cropped plate images to enhance character distinction, improving OCR accuracy. A custom function was developed to correct potential character misinterpretations based on visual similarities, tailored to the Turkish plate format. These solutions were critical in refining the system's performance and ensuring reliable license plate detection and recognition. Overall, the methodology reflects a comprehensive approach to developing a sophisticated, efficient, and accurate university car tracker software.





Conclusion

The university car tracker software presents a robust solution for automated license plate recognition and penalty management, leveraging the YOLOv8 deep learning model to accurately detect and process car plates in real-time. By integrating a secure relational database, the system ensures efficient storage and retrieval of car information, penalties, and user data, while the user management features streamline administrative tasks. This automation significantly enhances accuracy, reduces human error, and improves overall operational efficiency compared to traditional manual systems. With future enhancements like advanced vehicle type classification, multi-lingual OCR, and sophisticated reporting and analytics capabilities, the software promises to provide even greater utility and adaptability in managing campus parking and traffic regulations.

Looking ahead, there are several avenues for enhancing the university car tracker software. Firstly, expanding the system's detection capabilities to include various vehicle types, such as motorcycles and scooters, can provide more comprehensive parking management. Additionally, incorporating a multi-lingual OCR engine would allow for accurate recognition of international license plates, accommodating the diverse university community. On the security front, implementing data encryption and two-factor authentication for user logins would significantly bolster the system's security measures. Lastly, developing advanced reporting and analytics features, such as heatmap generation for frequent violation areas and predictive analytics for identifying peak violation times, can offer valuable insights and help optimize campus traffic management. A user-friendly data visualization dashboard could further enhance decision-making processes by presenting these insights in an accessible manner. These enhancements will ensure the software remains adaptable and effective in meeting the evolving needs of the university.

Acknowledgements

We would like to express our sincere gratitude to everyone who contributed to the successful development of the university car tracker software. Special thanks to our academic advisors for their invaluable guidance and support throughout the project. We are also deeply grateful to the university's IT department for providing the necessary infrastructure and resources. Our appreciation extends to the contributors of the YOLOv8 and EasyOCR projects, whose open-source tools were instrumental in the realization of our system. Additionally, we acknowledge Roboflow for their comprehensive dataset and annotation tools that greatly facilitated our model training. Finally, we thank our colleagues and testers who provided critical feedback during the development and testing phases, ensuring the software's robustness and reliability. This project would not have been possible without the collective effort and dedication of all involved.