ISSN 2277-2685
IJESR/April-June. 2025/ Vol-15/Issue-3s/839-847
Sangireddy Abhinaya *et. al.*, / International Journal of Engineering & Science Research

EnhanceLPR

Mohd Basit Mohiuddin[1], Sangireddy Abhinaya[2], Tokalla Akshaya Reddy[3], I Meghana[4], N Tareni[5]
[1]Assistant Professor, Department of CSE (AI&ML), Bhoj Reddy Engineering College for Women, India
[2,3,4,5]B.Tech Students, Department of CSE (AI&ML), Bhoj Reddy Engineering College for Women, India

Abstract
EnhanceLPR is an innovative web application designed to streamline and automate the complex process of vehicle license plate recognition from static images, addressing the common challenges of poor image quality and manual data entry. Users can effortlessly upload images of vehicles, after which the system initiates an intelligent, multi-stage AI pipeline. This pipeline, meticulously orchestrated by Genkit and powered by Google's versatile Gemini models, performs a comprehensive analysis: it begins by accurately detecting and isolating license plate regions, even from angled perspectives. Crucially, it then assesses the clarity of the cropped plate and, if necessary, employs an AI-driven enhancement process to improve readability, a vital step for handling blurry or low-resolution captures. Subsequently, advanced Optical Character Recognition (OCR) is performed to extract alphanumeric characters from the (potentially enhanced) plate. The final stage involves validating this extracted text against standard Indian license plate formatting rules, providing immediate feedback on compliance. The application is built with a modern tech stack featuring Next.js and React for a dynamic and responsive user experience, further refined with ShadCN UI components and Tailwind CSS for a polished and intuitive interface. EnhanceLPR delivers clear visual feedback throughout each stage of the analysis, offering a robust and practical tool for efficient license plate data extraction and initial verification, beneficial for scenarios requiring quick processing of vehicle identification from images.

Introduction

EnhanceLPR (Enhanced License Plate Recognition) is a smart web-based application that helps users automatically detect, enhance, and extract text from vehicle license plates in images. The app is designed for Indian license plates and uses advanced AI models to perform tasks like plate detection, blur checking, enhancement, text reading (OCR), and validation. Users simply upload an image, and the system shows all the steps and final result in an easy-to-understand interface.

Existing System

Currently, license plate recognition (LPR) systems rely on traditional image enhancement techniques like bilinear/bicubic interpolation or basic CNN-based super-resolution to improve low-quality plate images. These methods generically upscale images by interpolating pixels but fail to reconstruct critical details (e.g., distorted characters, motion blur), leading to poor OCR accuracy—especially in real-world scenarios like surveillance footage or low-light conditions.

Proposed System

1. AI-Powered License Plate Detection

- Uses Google Gemini Vision for accurate plate localization
- Supports all vehicle types (cars, bikes, trucks)
- Handles tilted/partial plates via adaptive bounding boxes
- Works in diverse lighting/weather conditions

1. Intelligent Image Enhancement

- Auto-blur detection
- Conditional enhancement
- Non-destructive sharpening that preserves text integrity
- 4x resolution boost for unreadable plates

2. India-Specific OCR & Validation

- **Pre-trained on Indian plates** (all state codes, HSRP formats)
- Validates against:
- Standard format: [State][XX][XX][XXXX] (e.g. MH 04 AB 1234)
- **Rejection feedback** (e.g. "Invalid: TN plates require 2-digit district code")

3. User-Centric Interface

- Drag-and-drop image upload (JPG/PNG)
- Side-by-side visual comparison:
- Original image
- Detected plate crop
- Enhanced version (if applicable)

3 Design

Architecture

Project architecture represents number of components we are using as a part of our project and the flow of request processing i.e. what components in processing the request and in which order. An architecture description is a formal description and representation of a system organized in a way that supports reasoning about the structure of the system. Architecture is of two types. They are

(1) Software Architecture
(2) Technical Architecture

Software Architecture

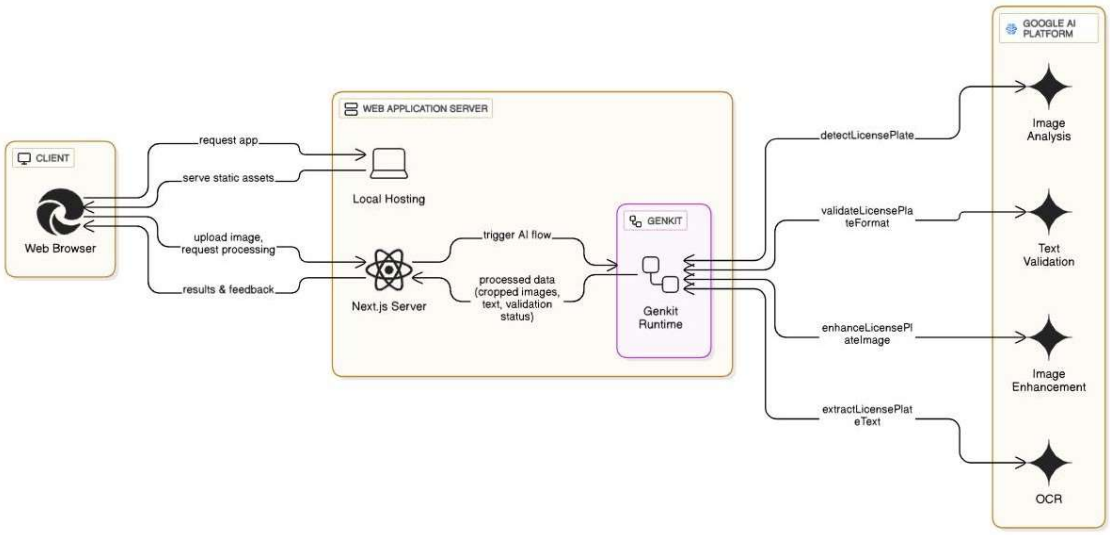


Fig 3.1.1 Software architecture

Technical Architecture:

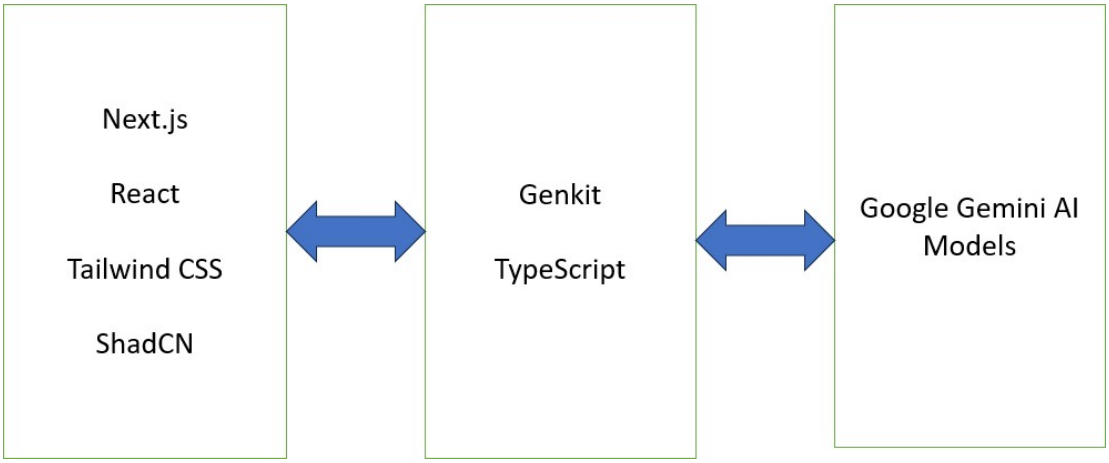


Fig 3.1.2 Technical architecture

Implementation

Technologies

The EnhanceLPR (License Plate Recognition) system leverages advanced multimodal AI, web technologies, and an iterative development approach to detect, enhance, extract text from, and validate vehicle license plates from user-uploaded images. Users submit an image through a Next.js and React-based frontend, styled with Tailwind CSS and ShadCN UI components. The backend processing is orchestrated by Genkit, which manages a series of flows calling Google Gemini AI models for sophisticated tasks including license plate detection and cropping, conditional image enhancement, optical character recognition (OCR), and format validation against Indian standards. Developed using TypeScript for robust code and following an Agile methodology for flexibility and rapid feedback, EnhanceLPR delivers processed plate images, extracted alphanumeric characters, and a clear validation status, aiding users in tasks requiring accurate license plate information.

Testing

Overview of Testing

Software testing is essential to ensure the EnhanceLPR system works reliably and meets requirements. As EnhanceLPR uses advanced AI to process visual data, thorough testing helps catch errors, improve accuracy, and avoid false results—critical for end-users like police or administrators.

Test Cases

EnhanceLPR – Test Case Table:

Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
Verify successful upload of a valid image (JPEG)	Valid vehicle image, .jpeg format	Image displays in "Original Image" section. "Analyze Plate" button becomes enabled. No errors.	Image loaded, button enabled	Pass
Verify successful upload of a valid image (PNG)	Valid vehicle image, .png format	Image displays in "Original Image" section. "Analyze Plate" button becomes enabled. No errors.	Image loaded, button enabled	Pass
Verify upload of an invalid file type (e.g., TXT)	A .txt file	Error message "Invalid file type. Please upload an image." is displayed. "Original Image" section remains empty or clears. "Analyze Plate" button remains disabled.	Error message shown, no image	Pass
Verify upload with no file selected	User clicks upload but cancels file selection	No image displayed. "Analyze Plate" button remains disabled.	No action taken, button stays disabled	Pass

Verify uploading a new image clears previous results	1. Upload Image A, process. 2. Upload Image B.	After Image B is uploaded, all result sections are cleared. "Original Image" section shows Image B.	Previous results cleared	Pass

Plate Detection & Cropping:

Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
Process a blurry but detectable license plate	Blurry plate crop	'Enhanced Plate' section shows a visibly clearer image	Enhanced image sharper than input	Pass
Process a clear license plate (no enhancement needed)	Clear, sharp plate image	Enhancement skipped, same image shown	Enhancement skipped	Pass
Enhancement fallback if detection fails to crop	Image with no detected plate	No enhancement performed, fallback image shown	Enhancement skipped on fallback	Pass

Image Enhancement:

Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
Detect plate in a clear, front-facing vehicle image	Clear image, license plate directly facing camera	Cropped plate shown in 'Detected Plate' section	Cropped plate displayed	Pass
Detect plate in an image with plate at a slight angle	Clear image, slightly angled plate	Cropped image still shown, slightly rotated	Cropped plate shown, acceptable angle	Pass
Image with no visible license plate	Car without a plate or scenery	Message: 'No plate detected', fallback image used	Fallback handled correctly	Pass
Image where plate is very small or heavily obscured	Distant/obscured plate	Similar fallback as above	Fallback handled correctly	Pass

Text Extraction(OCR):

Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
Extract text from a clear, enhanced plate image	Enhanced image with clean alphanumeric plate	Extracted text matches plate characters	Accurate extraction	Pass
Extract text from a plate with unusual font/spacing	Stylized but readable plate	Best interpretation of characters shown	Slight variation but acceptable	Pass
Attempt OCR on an image with no visible text	Pure noise/unreadable image	'No text extracted' message shown	Text field empty or error shown	Pass

OCR on plate with mixed case or special characters	Plate with lowercase or special symbols	Characters extracted accurately (if trained)	As expected or fallback to uppercase	Pass

Format Validation:

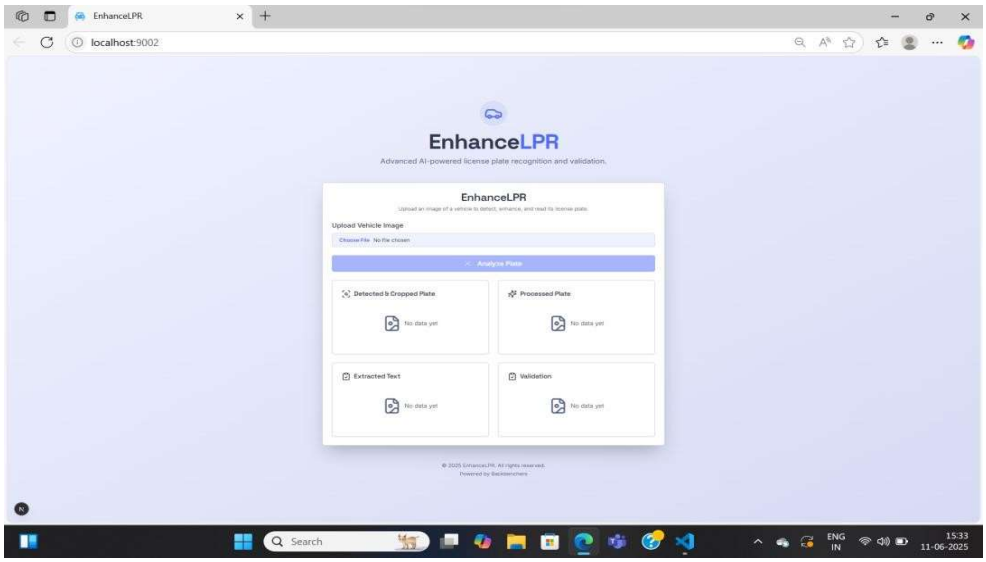
Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
Validate a correctly formatted Indian license plate	Text: 'MH01AB1234'	Shows 'Valid Format'	Format marked valid	Pass
Validate a correctly formatted plate with spaces	Text: 'KA 05 N 123'	Shows 'Valid Format'	Format marked valid	Pass
Validate an incorrectly formatted plate (extra characters)	Text: 'DL1ABC12345'	'Invalid Format' with explanation	Format marked invalid	Pass
Validate an incorrectly formatted plate (symbols)	Text: 'AP??AY0001'	'Invalid Format' with reason (invalid characters)	Format marked invalid	Pass
Validate with empty extracted text	Text: ''	Message: 'No text extracted to validate.'	Format marked invalid	Pass
Validate with gibberish text	Text: 'XYZ123ABC'	'Invalid Format' with reason	Format marked invalid	Pass

Overall Workflow & Error Handling:

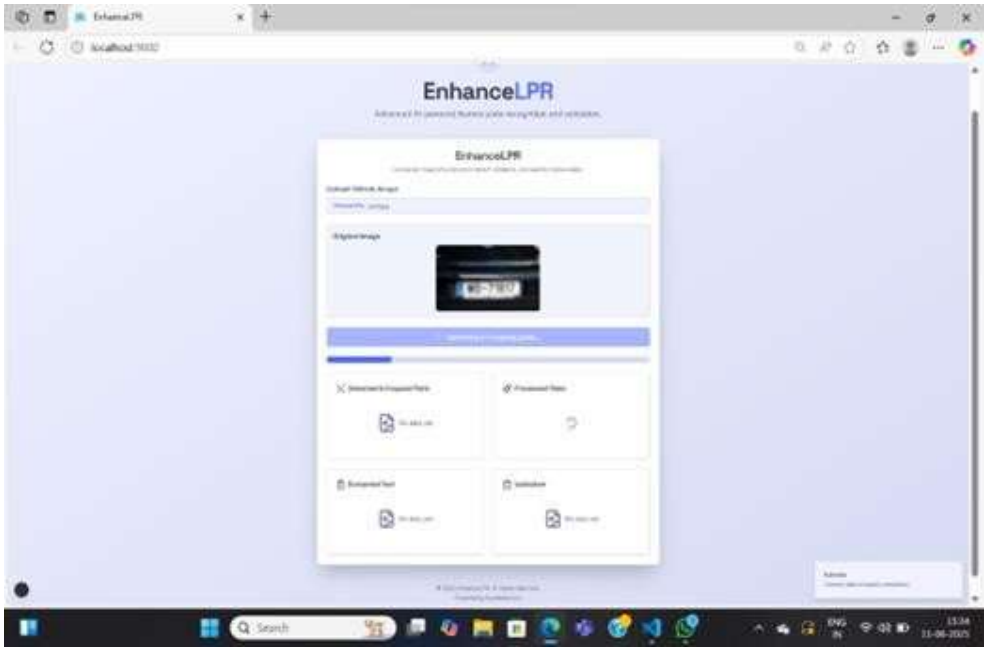
Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
Full successful pipeline run (clear image)	Clear image, standard plate	All sections processed, 'Analysis Complete!' message	Complete, no errors	Pass
Full pipeline run (blurry image)	Blurry but usable image	Enhancement applied, full result shown, 'Analysis Complete!'	All steps completed correctly	Pass
Pipeline failure at detection step	Simulate detection API failure	Error message shown, processing stops	Error handled, button re-enabled	Pass
Pipeline failure at enhancement step	Simulate enhancement API failure	Enhancement skipped or error shown, OCR may proceed	Error shown, process continued or aborted	Pass
'Analyze Plate' button disabled during processing	Click while processing	Button disabled until processing ends	Button disabled correctly	Pass

Results

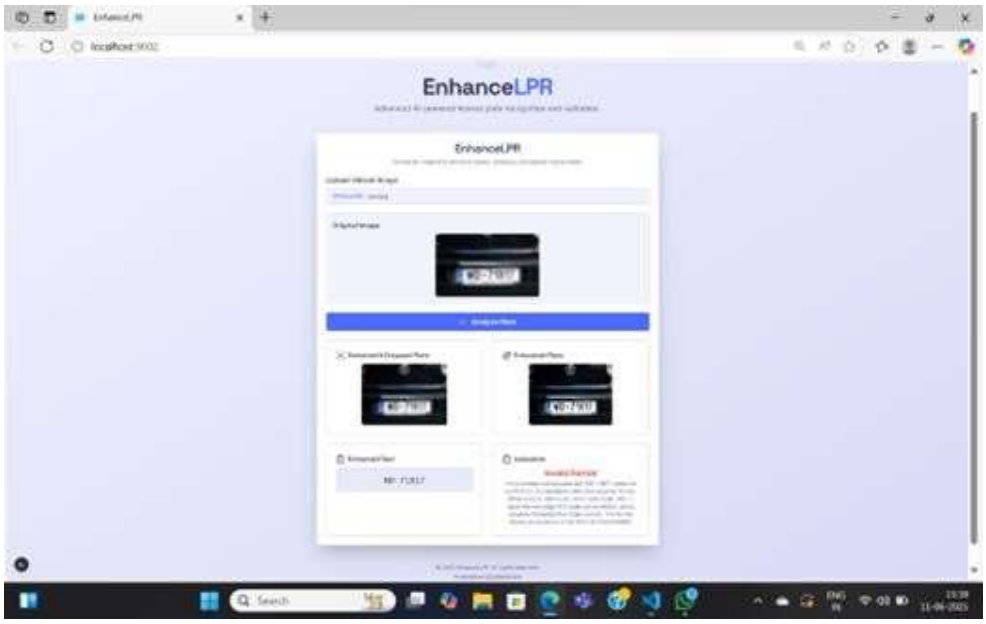
Screenshots



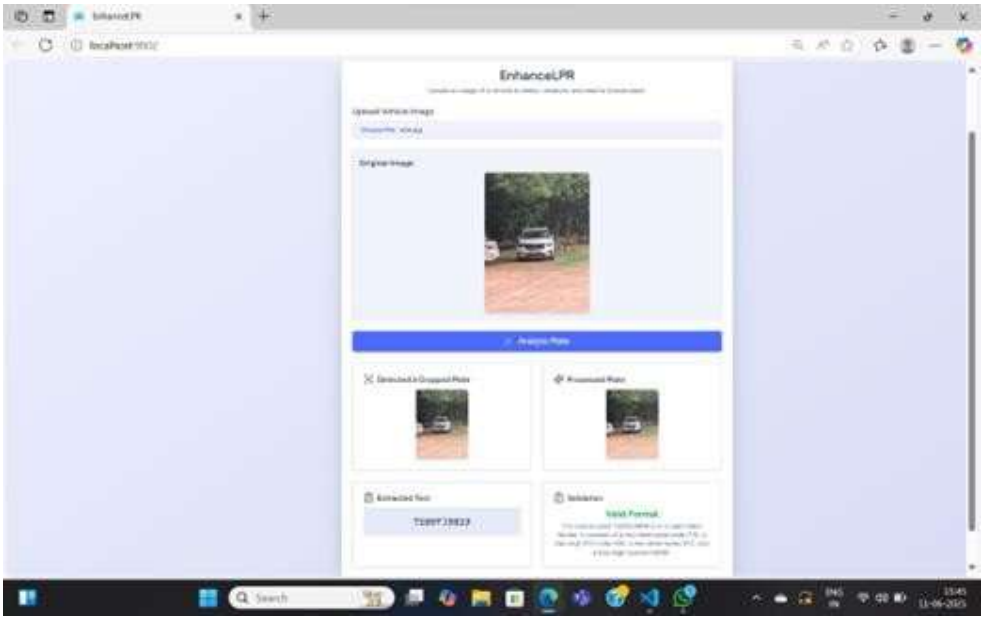
Screenshot 6.1 EnhanceLPR Home Page (Upload Interface with No File Selected)



Screenshot 6.2 EnhanceLPR Processing Image (Detecting & Cropping License Plate)



Screenshot 6.3 EnhanceLPR Output with Invalid License Plate Format



Screenshot 6.4 EnhanceLPR Output with Valid Indian License Plate Format

Conclusion & Future Scope

Conclusion

The EnhanceLPR project successfully demonstrates the design and implementation of an AI- powered system for the comprehensive analysis of vehicle license plates. By integrating a modern web frontend built with Next.js, React, and ShadCN UI, with a sophisticated backend AI pipeline orchestrated by Genkit and powered by Google Gemini models, the system effectively automates the process of license plate detection, conditional image enhancement, optical character

recognition (OCR), and format validation against Indian standards.
The project successfully achieved its core objectives:

- Providing a user-friendly interface for image upload and results display.
- Implementing a robust, multi-stage AI pipeline capable of handling various image qualities.
- Delivering clear and actionable outputs, including the processed plate images, extracted text, and validation status.
- Leveraging advanced multimodal AI capabilities for tasks that traditionally required separate specialized models.

EnhanceLPR showcases the potential of modern AI tools and frameworks to build practical and intelligent applications. The use of TypeScript ensured code quality and maintainability, while the Agile-inspired iterative development (implied by the conversational AI assistance in Firebase Studio) allowed for flexible refinement. The system provides a solid foundation for further development and offers a valuable tool for scenarios requiring accurate and efficient license plate information retrieval.

Future Scope

While EnhanceLPR provides a strong proof-of-concept and a functional application, there are numerous avenues for future development and enhancement:
Model Fine-Tuning and Specialization:

- **Region-Specific Models:** Fine-tune the detection, OCR, and validation models on datasets specific to particular Indian states or regions to improve accuracy for local plate variations and conditions.
- **Challenging Conditions Training:** Collect and train/fine-tune models on images taken in adverse conditions (e.g., night time, rain, extreme angles, very blurry plates) to enhance robustness.

Expanded Functionality:

- **Video Processing:** Extend the system to process video feeds from CCTVs or dashcams for real-time license plate recognition.
- **Vehicle Characteristics:** Augment the AI to also identify vehicle make, model, and color, linking this information to the detected plate.
- **Batch Processing:** Allow users to upload and process multiple images simultaneously.
- **Historical Data & Analytics:** Implement a database to log all processed plates, detection confidence, OCR results, and user feedback. This would enable analytics, trend identification, and performance monitoring.

Integration Capabilities:

- **Database Integration:** Connect with external databases (e.g., RTO databases, stolen vehicle databases) for real-time verification or flagging of suspicious plates.
- **API Development:** Expose the EnhanceLPR functionalities via an API for integration into other existing law enforcement or traffic management systems.

User Experience and Interface Enhancements:

- **Advanced Image Editing Tools:** Allow users to manually adjust cropping or apply basic filters if automated results are not satisfactory.
- **Confidence Scores Display:** Show confidence scores for detection and OCR to give users a better understanding of the AI's certainty.
- **User Accounts and Management:** Implement user authentication and roles for tracking usage and securing data.
- **Mobile Application:** Develop native mobile applications (iOS/Android) for on-field use by personnel.

References

[1] **Zherzdev, S., & Gruzdev, A.** (2018). LPRNet: License plate recognition via deep neural networks. *arXiv*. https://doi.org/10.48550/arXiv.1806.10447

[2] **K. V. Kadambari and V. V. Nimmalapudi (2020).**Deep Learning Based Traffic Surveillance System for Missing and Suspicious Car Detection arXiv: *2007.08783* — https://arxiv.org/abs/2007.08783

[3] **S. Sirisha Nadiminti, Pranav Kant Gaur & Abhilash Bhardwaj (2022)** Exploration of an End-to-End Automatic Number-Plate Recognition Neural Network for Indian Datasets.arXiv: *2207.06657* — https://arxiv.org/abs/2207.06657

[4] Y. Wang, J. Zhang & X. Zhao (2022).End-to-End High Accuracy License Plate Recognition Based on Depthwise Separable Convolution Networks.arXiv: *2202.10277* — https://arxiv.org/abs/2202.10277

[5] **R. Laroca, E. V. Cardoso & D. R. Lucio et al. (2022)**.On the Cross-dataset Generalization in License Plate Recognition.arXiv: *2201.00267* — https://arxiv.org/abs/2201.00267

[6] **D. Schirrmacher, T. Mühling & W. Samek**

(2023).Benchmarking Probabilistic Deep Learning Methods for License Plate Recognition.arXiv: *2302.01427* — https://arxiv.org/abs/2302.01427