



VEHICLE NUMBER PLATE RECOGNITION

Ms.Deshmukh V.H.¹, Manjit Ghule²

¹ Faculty Guide & Supervisor | ² Dattakala Group of Institutions, Faculty of Engineering, Pune, Maharashtra, India, India

Corresponding author: manjitghule26@gmail.com

ARTICLE ID IJREK-2026-00122	ARTICLE TYPE Research Article	RESEARCH AREA Machine Learning & Deep Learning	PUBLICATION Volume 1, Issue Issue 1 (2026)
RECEIVED 19 October 2026	REVISED 21 October 2026	ACCEPTED 19 October 2026	PUBLISHED 24 October 2026

Abstract

Scholarly manuscript entitled 'vehicle number plate recognition'.

Keywords: Research; Technology

1.1 PROBLEM STATEMENT

The problem is to locate a vehicle registration plate and recognise its characters accurately under varying lighting, distance, orientation, and background conditions while minimising incorrect readings. 1.2 Objectives The objectives are to acquire permitted vehicle images; detect the plate region; enhance the crop; recognise alphanumeric characters; validate the output; store authorised records with timestamps; and evaluate detection and recognition performance. 1.3 Scope The prototype is intended for academic demonstration using permitted images or a controlled test environment. It may later be extended to parking or entry-management workflows. It should not make enforcement or other consequential decisions automatically.

2. LITERATURE REVIEW

Traditional number-plate recognition systems use grayscale conversion, edge detection, morphology, and connected components to locate likely plate regions. These methods can be lightweight but may be sensitive to shadows and complex backgrounds. Modern approaches use object-detection networks for plate localisation and OCR or learned text-recognition models for transcription. Public datasets such as CCPD and UFPR-ALPR can support experiments, but differences in plate formats, cameras, and image conditions may limit generalisation.

2.1 RESEARCH GAP

High performance on curated datasets does not guarantee reliability in real settings. Evaluation should report both plate localisation and exact text recognition, consider low-light and blurred images, and define how low-confidence predictions are reviewed. Images and recognised plate numbers also require appropriate access and retention controls.

3. PROPOSED SYSTEM

The system accepts an image or video frame, detects the number plate, crops the plate region, applies resizing and enhancement, and passes the image to OCR. The predicted text is normalised and checked for confidence. Results are displayed to the operator and may be stored securely with timestamps and confidence information.

Figure 1. Proposed processing flow (conceptual): Image/Video Input → Plate Detection → Plate Cropping and Enhancement → OCR Character Recognition → Format/Confidence Check → Display or Secure Record Storage Vehicle Number Plate Recognition System | Page 2

3.1 MAIN MODULES

Module

Purpose Image acquisition

Read permitted still images or video frames from a camera or dataset. Plate detection

Locate the likely registration plate using image processing or an object-detection model. Preprocessing

Crop, resize, reduce noise, improve contrast, and correct perspective where feasible. OCR engine

Convert plate characters into text and return confidence information where supported. Validation

Check output format and flag low-confidence or ambiguous readings for review. Storage and interface

Display results and store permitted records with timestamps and restricted access.

TABLE I

Module	Purpose
Imageacquisition	Readpermittedstillimagesorvideoframesfromacameraordataset .
Platedetection	Locate the likely registration plate using image processing or an object-detection model .
Preprocessing	Crop resize reducenoise improvecontrast andcorrectperspectivewherefeasible , , , , .
OCREngine	Convertplatecharactersintotextandreturnconfidenceinformationwheresupported .
Validation	Check output format and flag low-confidence or ambiguous readings for review .
Storageandinterface	Displayresultsandstorepermittedrecordswithtim estampsandrestrictedaccess .

3.2 SYSTEM ARCHITECTURE

A prototype may use Python and OpenCV for image processing, an object detector for plate localisation, and Tesseract OCR or a trained recognition model for character extraction. Flask or FastAPI can provide an interface, while SQLite can store prototype records. Testing should begin with stored images before camera deployment. 4. Methodology

4.1 DATASET AND DATA GOVERNANCE

The prototype may use a documented public dataset such as CCPD or UFPR-ALPR after reviewing its terms. Local data should be collected only with permission. Training, validation, and test sets should be separated to prevent near-duplicate frames from inflating results. This paper does not claim that experiments have already been completed.

4.2 IMAGE PREPROCESSING

The image is checked for resolution and readability. The plate crop is resized while preserving its aspect ratio where possible. Grayscale conversion, denoising, thresholding, contrast adjustment, and perspective correction may improve OCR, but each step should be tested because excessive filtering can distort characters.

4.3 PLATE DETECTION AND CHARACTER RECOGNITION

Plate localisation can use contour-based techniques as a baseline or a trained object detector for complex scenes. OCR estimates the characters. Format validation may support recognition but should not silently replace uncertain characters. Confidence information and the original crop should be retained where permitted for human review.

4.4 CONCEPTUAL ALGORITHM

1. Load an authorised image or frame. 2. Detect and crop the plate. 3. Preprocess the crop. 4. Run OCR. 5. Check confidence and format. 6. Flag uncertain readings. 7. Display or securely store the result. 8. Evaluate errors and update the model only after validation.

5. IMPLEMENTATION PLAN

Layer

Suggested technology		
Responsibility Image processing		
Python, OpenCV		
Image loading, cropping, enhancement, perspective correction. Plate localisation		
OpenCV baseline or object detector		
Find plate bounding boxes. Text recognition		
Tesseract OCR or trained OCR model		
Extract alphanumeric plate text. Application/API		
Flask or FastAPI		
Accept images and return recognition results. Storage		
SQLite		
Store authorised records and timestamps. Interface		
HTML/CSS or simple Python UI		
Display plate crop, recognised text, and confidence.		

TABLE II

Layer	Suggestedtechnology	Responsibility
Imageprocessing	Python OpenCV ,	Imageloading cropping enhancement perspectivecorrection , , , .
Platelocalisation	OpenCVbaselineorobjectdetector	Findplateboundingboxes .
Textrecognition	TesseractOCRortrainedOCRmodel	Extractalphanumericplatetext .
Application/API	FlaskorFastAPI	Acceptimagesandreturnrecognitionresults .
Storage	SQLite	Storeauthorisedrecordsandtimestamps .
Interface	HTML/CSSorsimplePythonUI	Displayplatecrop recognisedtext andconfidence , , .

5.1 SECURITY AND PRIVACY REQUIREMENTS

Restrict access to images and recognised registration numbers, protect API endpoints, validate uploaded files, and keep credentials outside source code. Retention should be limited to the stated purpose. Low-confidence readings should be checked by a person and should not alone trigger consequential denial or enforcement.

6. EVALUATION PLAN AND DISCUSSION

Evaluate the system on held-out images with varied lighting, distance, orientation, and image quality. Measure plate detection and text recognition separately. Exact-match recognition is important because one incorrect character may identify a different vehicle. Report the dataset, test conditions, model configuration, common errors, and processing time. No numerical performance results are reported because no controlled experiment has been performed.

7. ADVANTAGES AND LIMITATIONS

The system can reduce manual data entry, process images consistently, support searchable records, and integrate with authorised parking workflows. Limitations include blur, glare, low resolution, dirt, damaged plates, occlusion, steep viewing angles, and confusion between similar characters. Real-world performance requires representative testing and human review. Vehicle Number Plate Recognition System | Page 3

8. ETHICAL AND RESPONSIBLE USE

Vehicle images and registration numbers can be sensitive in context. Collection should have an authorised purpose and follow applicable requirements. Use access controls, data minimisation, and defined retention periods. The system should not be used for unauthorised tracking or consequential decisions based solely on uncertain OCR output.

9. FUTURE SCOPE

Future work may include real-time video processing, multi-frame voting, lightweight edge models, support for multiple plate layouts, confidence calibration, and integration with an authorised parking management system. Further experiments can compare traditional image processing with deep-learning detectors under consistent conditions. 10. Conclusion This paper proposed a Vehicle Number Plate Recognition System combining image acquisition, plate localisation, preprocessing, OCR, confidence checking, and secure result storage. It can support parking and access-management applications while reducing manual effort. Reliability depends on image quality, representative evaluation, and careful handling of uncertain predictions. The system is suitable as an academic prototype, but performance must be established through controlled testing before real-world deployment.

REFERENCES

[1] Du, S., Ibrahim, M., Shehata, M., and Badri, W. (2013). Automatic License Plate Recognition (ALPR): A State-of-the-Art Review. IEEE Transactions on Circuits and Systems for Video Technology, 23(2), 311–325. <https://doi.org/10.1109/TCSVT.2012.2203741>

[2] Laroca, R., et al. (2018). A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector. 2018 International Joint Conference on Neural Networks (IJCNN).

[3] Xu, Z., Yang, W., Meng, A., Lu, N., Huang, H., Ying, C., and Huang, L. (2018). Towards End-to-End License Plate Detection and Recognition: A Large Dataset and Baseline. European Conference on Computer Vision (ECCV).

[4] OpenCV. OpenCV Documentation. <https://docs.opencv.org/>

[5] Tesseract OCR. Tesseract User Manual. <https://tesseract-ocr.github.io/>

Publication archival note: Official scholarly contribution published under CC BY 4.0 International License. Permanent record on <https://www.ijrek.org> | Article ID: IJREK-2026-00122