CHRIST
(DEEMED TO BE UNIVERSITY)
DELHI-NCR, INDIA

Computer Vision
MCA574

ESE – PROJECT REPORT SUBMISSION
LICENCE_PLATE_RECOGNITION

BY

AGAMYA TANWAR (24225034)
ABIN SAJI (24225043)
RUSHALI SRIVASTAVA (24225044)

SUBMITTED TO

Dr. Preety Ma'am

SCHOOL OF SCIENCES

2025-2026

TABLE OF CONTENTS

LICENCE PLATE RECOGNITION

1.Abstract

ALPR is a text-recognition challenge that is aimed at locating a tiny and clear rectangle within a large and crowded image (1). The system relies on simple tricks with images (filtering, edge detection), geometrical steps (examining shapes) and complete OCR to interpret the plate. This renders every section comprehensible and easy to learn and perfect (2,3).

2.Introduction

2.1 Background and Motivation

The vehicle plates are governed by the national regulations that determine letters, numbers, spacing, font and size. These regulations provide powerful cues that may be utilized by ALPR, including anticipated shapes and character arrangement (1). Plate reading is part of smart transport (where a camera is used to capture an image on the road) and a database containing information on ownership, permits, tolls or violations to automatically make decisions (2). ALPR is an interesting choice to learn computer vision since it encompasses the majority of steps and may be performed with the help of modest hardware (3).

2.2 Problem Statement

We are given formally a picture in color on a grid, is image I. The assignment is to identify a string S consisting of capital letters and figures that is on the plate in the scene (4). This requires two distinct tasks: detection The first task involves finding a tight box around the plate, and recognition The second task involves reading the text in the tight box (5). Either step can be the cause of mistakes. Missing or incorrect plate locations get recognition to fail and blurry or noisy images produce OCR errors. The pipeline should therefore be constructed in such a way that there are no failures (2).

2.3 Scope of the Project

The scholars usually rank the ALPR challenge based on the light, motion, blocking, and background (6). The cases we restrict ourselves to in this project are easy-to-medium, i.e. still pictures, normal daylight, single car. In such a manner traditional methods are efficient and we can observe the effect of each step without additional twists and complications that the high-end cameras experience (3). It is good to define the scope clearly since it is similar to what the method can in actual practice and the way we test it (7).

2.4 Relevant to Road Safety and Smart Cities (SDG Mapping).

System perspective of ALPR is part of bigger cyber-physical systems in which sensors, computers, and actions interact with each other in real-time such as automatically opening gates when an authorized plate is detected (8). Assisting in the application of regulations like speed limits, red lights and parking regulations etc, ALPR contributes to safer roads and more efficient infrastructure. These are major concepts in intelligent-city research and sustainable city planning (9).

3.Literature Review

3.1 LPR systems that use the Traditional Image Processing.

Classic LPR pipelines presuppose the presence of numerous sharp edges on the plate, and rectangle form. They rely on edge detectors (Sobel, Canny) to accentuate the boundaries, and finally seal holes and eliminate noise (1). Potential plates are located using related parts or outlines and filtered using simple tests like aspect ratio, size, angle and the number of sides; these are a basic kind of a statistical classifier that uses plate shape (2). The 1-D signals can be made by viewing 2-D images by looking at vertical line tallies high points are where there is a stroke and low points are where there is a space between characters to aid in decision making (9).

3.2 Machinery Learning and Deep Learning Methods.

More recent systems adopt deep convolutional nets which learn features directly based on data, which eliminates the need to manually create filters and rules (7). detector networks Detector networks are either guessing the box coordinates or marking pixels on a feature map. Recognition nets consider plate as list of characters and apply sequence models, which pay attention to each letter (8). These nets are translation safe, noisy stable, and can be applied across fonts but require larger models, a lot of training data, and fine-tuning of hyper-parameters (7).

3.3 Comparison to the Existing Systems and Research Gaps.

The tradeoff between being understandable and performing well is presented between classical and deep methods. Rule based pipelines allow you to control every step but find it difficult to handle hard scenes, deep models are better but are black boxes (2). Researchers are developing hybrids which deep detectors are used to locate solid plates, but simple OCR checks or rule checks are used to employ known plate templates, combining the strengths of both (8). The traditional method remains handy in the academic world since students can adjust every bit and observe direct outcomes on performance (3).

4.Tools and Requirements System.

Category	Item	Specification / Details
Hardware	Processor	Dual-core CPU or better, 2.0 GHz or higher
Hardware	Memory (RAM)	Minimum 4 GB (8 GB recommended)
Hardware	Storage	≥ 500 MB free space for Python, libraries and images
Hardware	Display / GPU	Standard display; dedicated GPU not required
Software	Operating System	Windows 10 / 11 (64-bit)
Software	Python Version	Python 3.x (e.g., 3.9 or above)
Software	OCR Engine	Tesseract-OCR installed and added to system PATH
Library	OpenCV	opencv-python for image processing
Library	NumPy	Array and matrix operations
Library	Matplotlib	Visualization of intermediate results
Library	pytesseract	Python wrapper for Tesseract OCR
Library	imutils	Utility functions for contour and image handling

4.1 Hardware Requirements

Conventions, edge detection and thresholding are examples of operations running in parallel to the number of pixels thus they can operate in real time on typical resolution images even on a CPU (6). Several intermediate images (original, gray, filtered, edge map, mask, cropped plate) are mostly stored in the memory. When the megapixel images are concerned, this can easily fit into the RAM of a conventional PC (4).

4.2 Software Requirements

Python is a high-level language that is friendly and is used in quick prototyping, whereas OpenCV and NumPy (written in C/C++) offer good speed when dealing with images (5). Tesseract is an open source HP-initiated complete OCR engine with layout, language, intelligent classification thus providing a ready-to-use component of reading plates without the need to construct your own classifiers (18).

4.3 Libraries and Frameworks (OpenCV, NumPy, Matplotlib, pytesseract, imutils) used.

The architecture of OpenCV brings together a wide range of traditional algorithms on a single API so that filters, edge detectors, morphological operators and geometric transformations can be stacked together into pipelines with ease (3). NumPy is used to represent images as multidimensional arrays, allowing operations that are operated upon in a fully-vectors mode to take advantage of lower-level optimizations such as SIMD instructions and BLAS libraries; this is especially handy when you want to apply a specific convolution kernel or mask (13). Introspection Matplotlib is preferred to debug logical errors: introspection helps us understand how mask and thresholds are computed wrongly, and lets us relate theoretical understanding of each step to the actual impact it has on the data (4).

5.Methodology / System Design

5.1 Overall System Architecture

The pipeline may be regarded as a chain of transformations. Applied to the input image, with each transformation providing less uncertainty of the plate location, or a refined representation with which subsequent tasks may be performed (2). Signal-to-noise ratio is enhanced by preprocessing, the spatial support is reduced to a region of interest, foreground and background characters are distinguished by segmentation and projecting visual patterns into the discrete space of strings is facilitated by OCR, making the visual data high-dimensionality data progressively reduced to structure representations (3).

5.2 Image Acquisition and Pre-processing.

The justification of the use of grayscale conversion is that, the shape and intensity transition is where most of the information required in plate detection and recognition, and the color would be redundant in this particular task (5). Sharpening filters are a discrete approximation to second-order derivatives that add on top of an image to enhance high-frequency data, which makes edges sharper, but also adds noise; they can be used together with CLAHE, which balances histograms within local tiles, to reduce the effect (3).

5.3 Edge Detection and Contour-based Plate Localization.

The bilateral filter is a weighted average of difference in intensity and spatial distance, and it is used to keep edges intact, but smooth in the area- mathematically it is a non-linear diffusion process, and does not cross image boundaries (11). Canny edge detection uses the calculation of gradient combined with non-maximum suppression and hysteresis thresholding to generate thin connected edge images which are not much sensitive to noise as compared to a simple gradient magnitude thresholding which enhances stability of the following edge contour extraction (12). The Ramer-Douglas-Peucker approximation algorithm is used to cut down the size of the data set of points by approximating polylines within a given tolerance, and the system can be used to sort the shape (e.g. quadrilaterals) by the number of its vertices rather than by dense sampling of the boundary (2).

5.4 Plate Region Extraction And Binarization

The conceptual equivalent to pointwise multiplication of the image by a binary support function, which is one in the region of interest and zero outside the plate and the background context, is called masking (6). The thresholding used by Otsu assumes a bimodal histogram and selects the threshold that maximises between-class variance an unsupervised criterion that tends to work well when the foreground and background intensities can be separated reasonably well, which is the case on dark characters on a light plate (7).

5.5 OCR using Tesseract

The LSTM engine models used by Tesseract represent character sequences instead of individual symbols, allowing the model to capture contextual dependencies, like frequent letter-digit sequences in license plates, which allows it to remain robust to small damage to individual characters (15). The page segmentation mode (PSM) controls the way in which Tesseract breaks down the image into blocks, lines and words; the mode which is specific to one line of text minimises the ambiguity of segmentation and usually produces higher accuracy on plate-like inputs compared to general multi-line modes (16).

5.6 Post-Processing of Text Recognized.

The domain-specific post-processing may be considered as running a language model with a limited alphabet where only some patterns, e.g., two letters, two digits, two letters, three digits, etc. are allowed, and the invalid outputs can be either corrected or rejected (19). These constraints can be encoded by simple regular expressions and to favor a candidate that matches the desired pattern heuristic scoring can be used, thereby correcting the occasional errors of OCR, like confusing "0" with "O" or "1" with "I" (20).

6.Implementation Details

6.1 Workflow and Directory Structure of the Project.

An organization of the project into configuration, processing functionalities and visualization make it easy to maintain and extend the project in the future, e.g. replacing the detection module without modifying the rest of the code (4). Intermediate results can be logged and cropped plate images saved to assist in experimental evaluation, where the comparison can be made among different parameter settings using the same test set in order to discover empirically the robust configurations (2).

6.2 Code explanation Step by step.

Since software-engineering-wise, each line of the script fulfills a particular transformation, reading and validating input, preprocessing, edge computation, contour extraction, ROI cropping, thresholding and OCR (5). Not only do these steps become more readable with descriptive names of variables and modular functions, which reflect the conceptual phases outlined in the methodology, but the connection between theory and practice is strengthened (3).

6.3 Parameter Options (Canny parameters, filters, psm/oem parameters).

The canny thresholds define the strength of gradients to be considered strong or weak; a threshold that is too low may enhance noise and false edges whereas one that is too large may split up the plate contours instead of making them connected, which is why empirical tuning tries to balance connected but not overdense edge maps (12). With bilateral filtering, spatial and range standard deviations are used to regulate the level of smoothing and edge preservation: taking a larger spatial sigma spreads smoothing on broader ranges of pixels, and taking a smaller range sigma requires that only similarintensity pixels affect each other so that edges remain sharp (11). Parameters of OCR like PSM and OEM adjust the segmentation strategy of Tesseract and the engine type; the LSTM engine with the single line mode represents a theoretically informed option with short, horizontally aligned texts such as license plates (16).

6.4 Management of Failure Cases and Fallback Strategy.

Strong systems should be able to deal with the failure modes like contours lost or ROIs moved to wrong places; a fallback that cuts out a reasonable area based on prior understanding of where the plate should be is a robustness addition that can be done at a very low computational price (10). Probabilistically, such a fallback to introduce a prior on plate location exchanges some spatial accuracy with increased recall of a candidate patch, where the OCR stage detects little candidate patch at all (6).

7.Results and Discussion

7.1 Data Description

Together with the difference in distance, orientation, and illumination to a certain extent, a significant evaluation will need pictures (4). By indicating ground-truth plate numbers of every test image, it is possible to compare objectively the results of OCR with the expected outcomes, permitting computing such measures as character accuracy and the full-string accuracy instead of using solely the human eye analysis (7).

7.2 Qualitative Results (Sample Outputs)

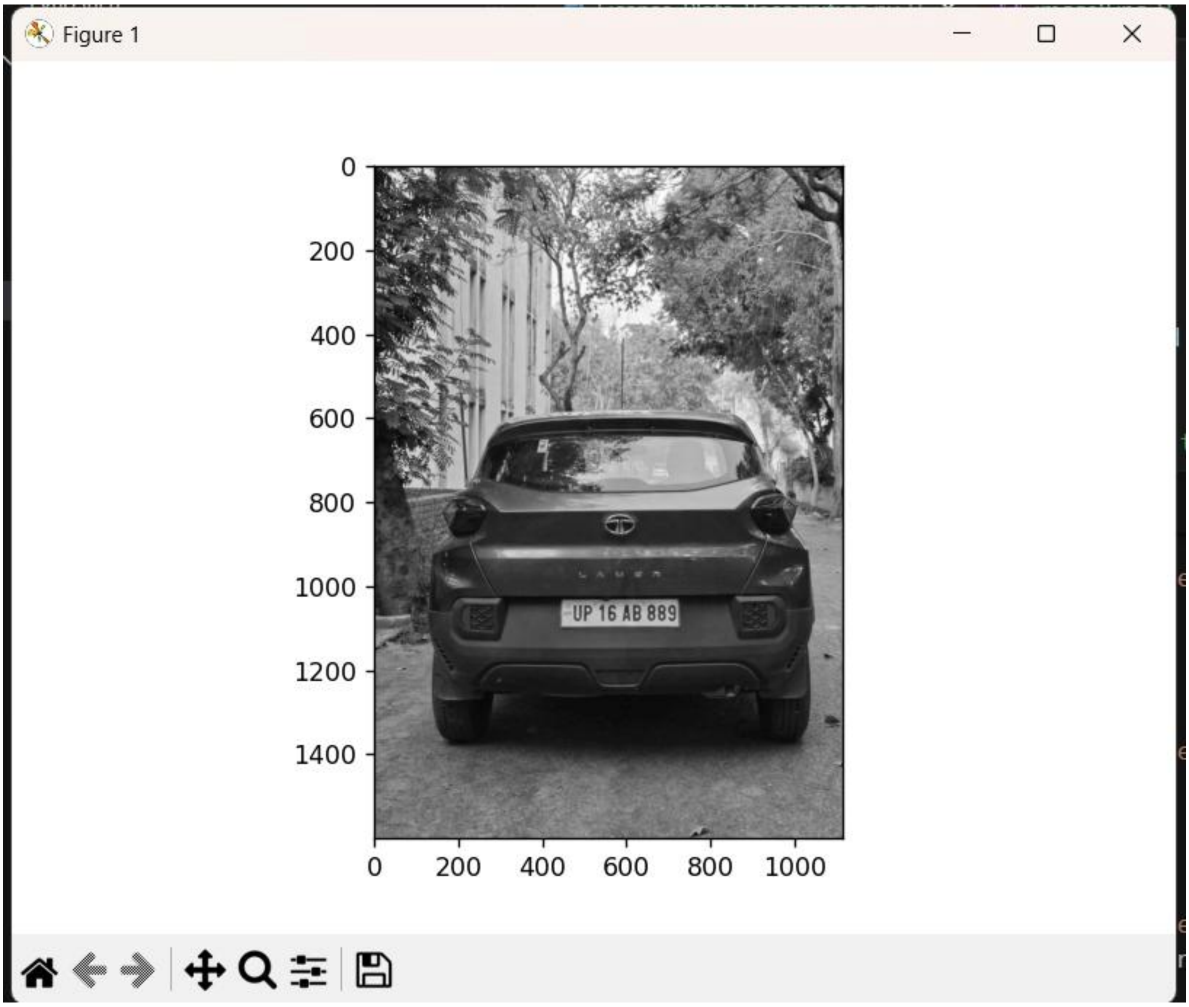
The visualization of intermediate results shows that if the cases are successful, the plate will have strong well-localized edges, a high-contrast binarization crop and demarcated character blobs of the plate as a result of the thresholding (3). Conversely, failure cases have been found to have either fragmented or noisy contours that result to the wrong masks, or low-contrast crops where characters merge with the background, which in turn is also associated with the elevated OCR error rates (6).

Picture 1: - Original car image with axes

INTERPRETATION: -

- In the entire scene, we can see the back view of the car with the number plate in the lower part of the central area, which proves that the plate is quite large and understandable in the original picture.
- The fact that there are distinct edges on the vehicle and plate indicates that Canny and contour detection will be provided with good structural information.
- Foreground objects such as trees and structures add more edges yet the powerful rectangular form and contrast of the plate still make it a special area.
- The coordinate axes on the borders are used to determine approximate plate location which is in form of pixel position and are useful in defining fallback crops.
- Lighting is natural and even, and thus, no extreme glare and deep shadows can appear on the plate, which is expected to aid dependable thresholding and OCR.

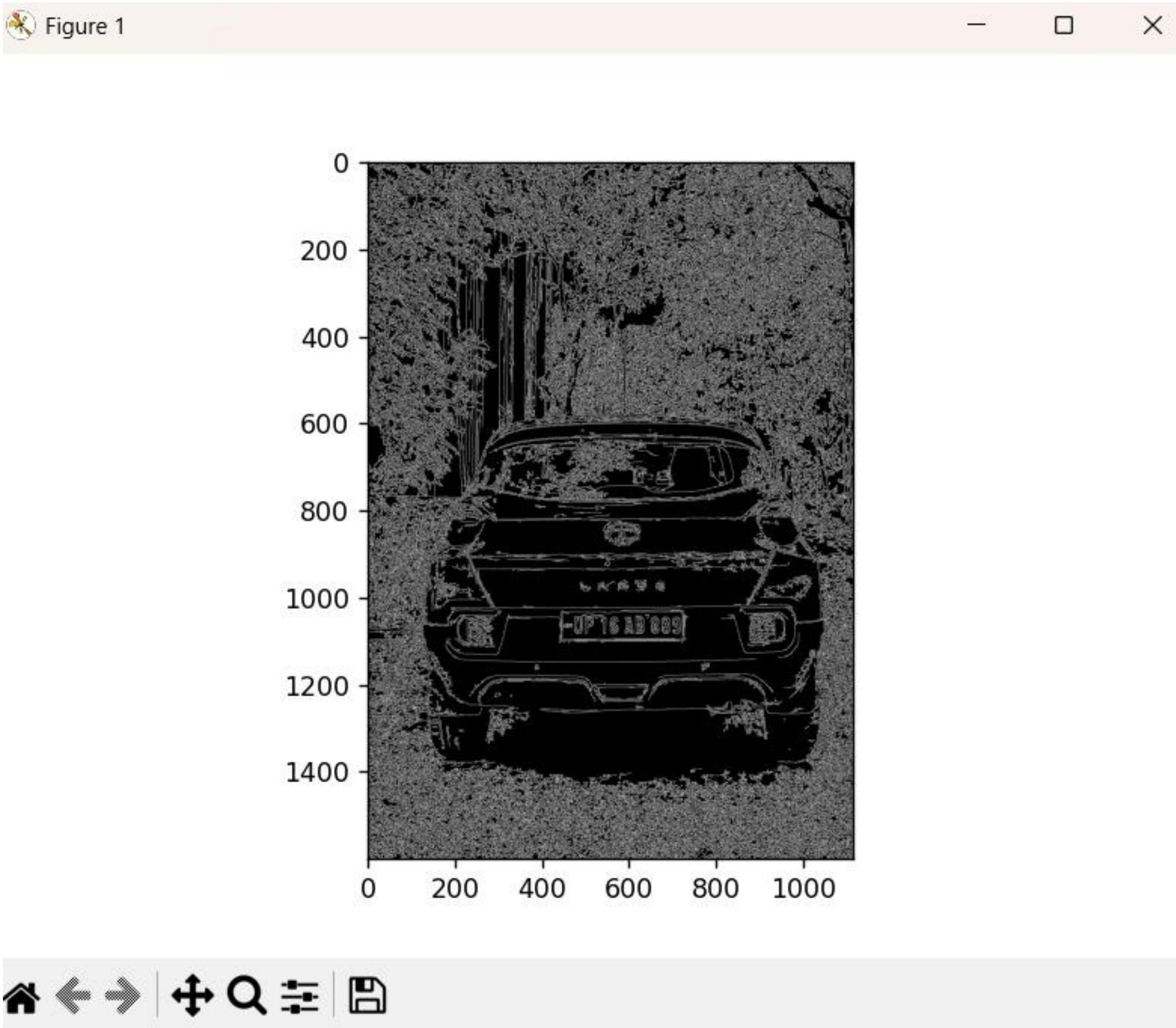
Picture 2: - Full grayscale car image



INTERPRETATION: -

- Grayscaling has been proven to be effective in removing colour information but preserving structural features such as edges, shadows and reflections required in later processing.
- The plate area is still clear in grayscale and this proves that the system does not rely on the plate color used.
- The global contrast is moderate, and this is why sharpening and CLAHE are used to enhance the local disparities between characters and background.
- The shadows under the car, in the background are darker but not domineering in the image thus they should not critically interfere with edge detection on the plate.
- This picture is where all the downstream processes begin so its level of clarity confirms that the initial read and color-conversion processes are proceeding as expected.

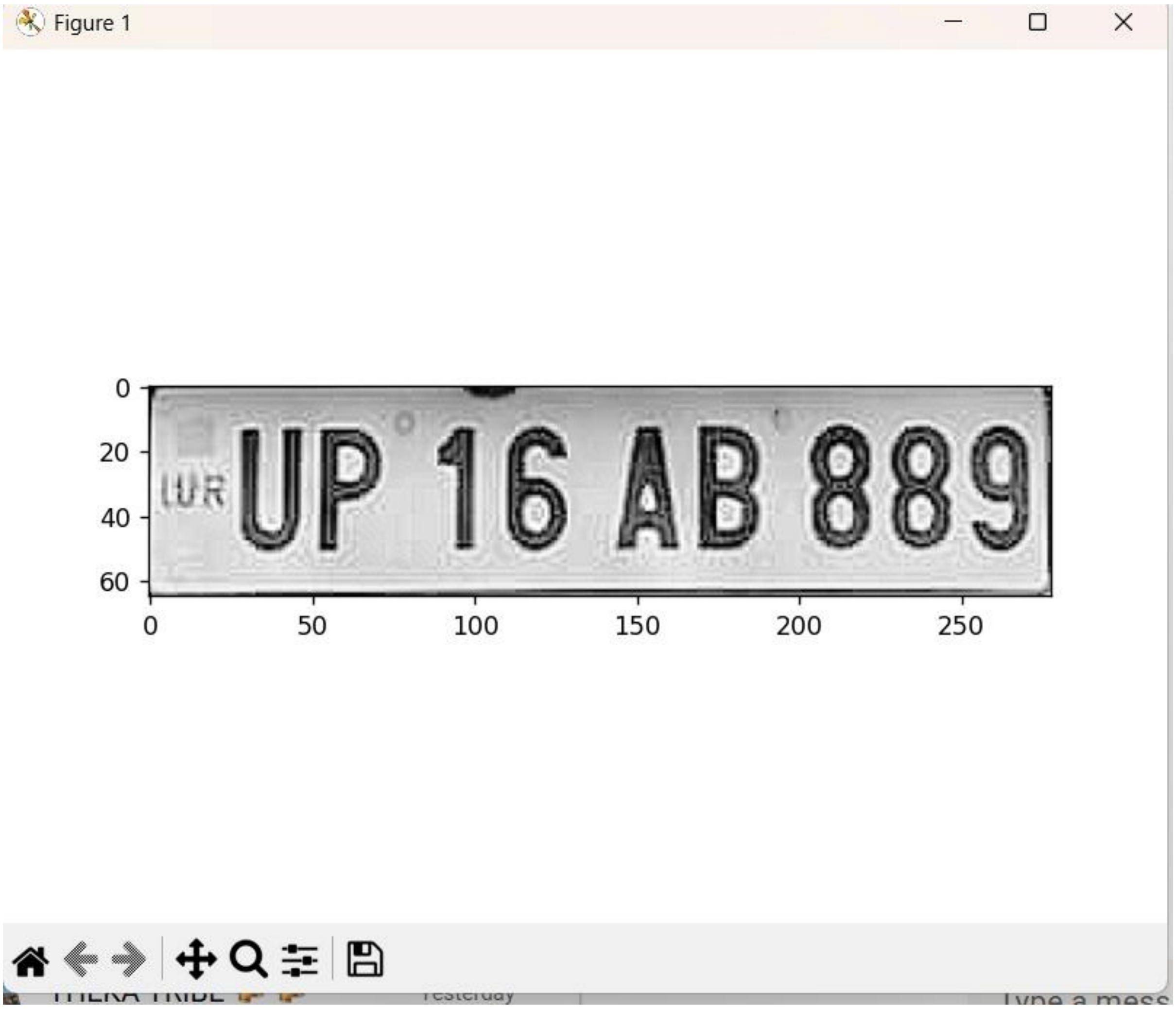
Picture 3: - Complete image edge map (Canny output)



INTERPRETATION: -

- The edges indicate that thematic intensity variations around the vehicle, plate border, window edges and lines of the trees are strong enough revealing that the selected Canny thresholds are not very strict.
- The plate region is a dense, rectangular collection of edges, and this is why the contour-based plate localization may be effective.
- There is also noise in the road and foliage, but that is normally not formed in regular shapes and can be filtered out using geometric filters (area, aspect ratio, vertex count).
- The fine texture of the background proves the importance of smoothing a bilateral filter before edge detection so as not to have too much clutter.
- This output is one of the most important diagnostic tools: in case the edges around the plate were weak or broken, it would instantly be able to explain failures in contour-detection.

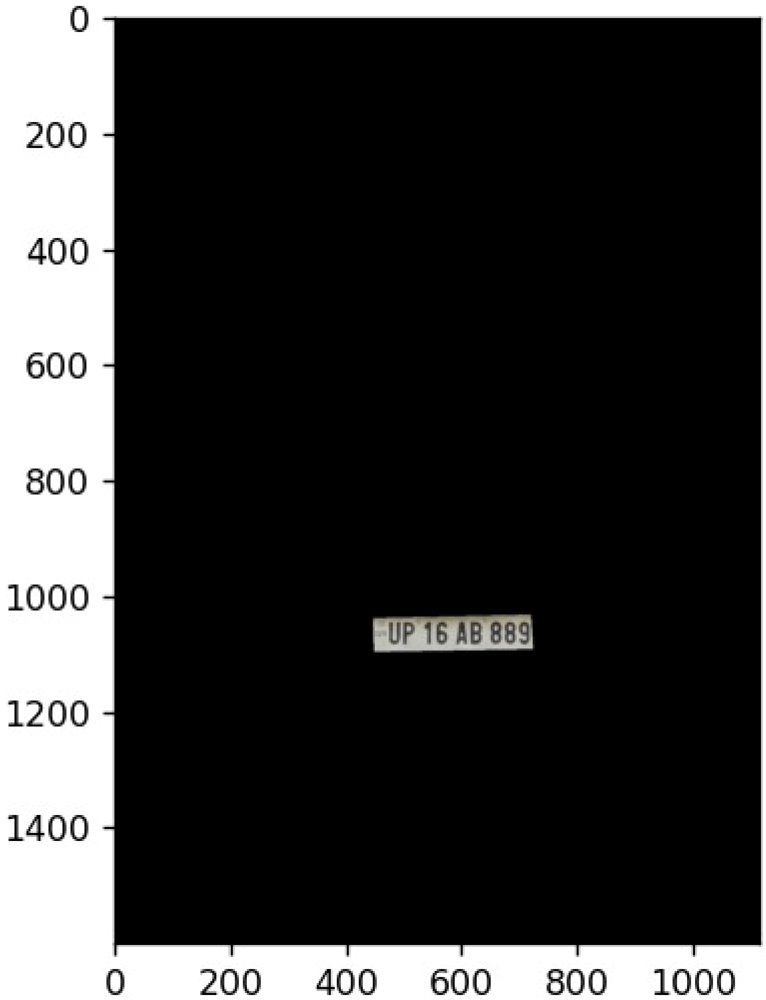
Picture 4: - Sharpened grayscale plate (close-up).



INTERPRETATION: -

- There is only the plate in the crop with very little background hence proving that the coordinate extraction of the mask and the cropping logic are not inaccurate.
- The separation of characters is done by dark text on a light background which is best suited to Otsu thresholding and median filtering.
- The horizontal orientation of the text and its aspect ratio demonstrate the fact that the contour was not skewed too much, and therefore, no perspective correction is needed in this dataset.
- There are small noise levels and slight marks, which means that more binarization and denoising should be applied prior to OCR.
- It will be obvious to the human eye that the plate can be read and this is equally true to the fact that Tesseract returns the correct text in your console output.

Picture 5: - Black image with plate area only visible.



INTERPRETATION: -

- This is achieved by the masking logic to ensure that the detected or fallback plate area is isolated and the remainder of the image is suppressed to black.
- The plate is more or less at the right location as far as the space position of the original image is concerned, meaning that the choice of the contours or the fallback box is positioned correctly.
- The geometry of the visible region (rectangular and proportions) appears to match a license plate which justifies the geometric assumptions of contour filtering.
- The non-plate areas that are totally dark indicate that bitwise masking is appropriately limiting further processing to the area of interest.
- This visualization can be also helpful to debug: in case an incorrect area was chosen, the bright spot would be placed on some other part of the picture.

Picture 6: - Output: "Text detected: UP 16 AB 889"

INTERPRETATION: -

- The printed string shows that the stage of OCR has been able to read both alphabetic and numeric characters in the right sequence.
- The distance between the output is about half the Indian standard license plate layout indicating that the selected PSM and OEM parameters may be suitable to single line plates.
- The lack of additional symbols or distorted characters shows that both thresholding and denoising created a clean binary image.
- It can also be directly utilized as a key in databases or can be utilized in application logging like parking or access control systems.
- The comparison of new output with this correct baseline can be used to assess any future changes in preprocessing or Tesseract configuration.

7.3 Accuracy and Observations

Although the data is limited in size, it is possible to have a conceptual understanding of detection errors (plate not correctly localized) and recognition errors (plate localized but misread), and study which of the two errors is dominant; this can assist in deciding whether to put effort into improving the preprocessing or into adjusting the OCR settings (2). It can be observed that the recognition accuracy could be found extremely plate-size sensitive in pixels so that in the future systems, the minimum resolution requirement in capture plates must be maintained, such as by having at least a minimum number of pixels per character height (9).

7.4 Shortcomings of the Existing Strategy.

The primary theoretical drawback of the present strategy is that it uses global hand-tuned thresholds and heuristics, which do not respond to changing environmental factors and camera properties (7). It also uses a rigid plate geometry and does not consider the 3-D effect like perspective distortion, which can greatly change the aspect ratio and spacing between the characters when the vehicle is taken at very steep angles (8).

8.Uses and Future Prospects.

8.1 Application in the real-life scenarios (Parking, Traffic Monitoring, Automation).

The ALPR in parking management systems can automatically read and write vehicle plate numbers in entry points and exit points, calculate the duration of parking and initiate payment processes without the need of physical tickets and RFID tags (5). In traffic surveillance, plate recognition with a combination of several road-side cameras will allow identification of a vehicle in different places, which can be utilized in estimating travel-time, congestion and the identification of stolen or blacklisted vehicles once compared to the central databases (7). ALPR may also be used in gated communities, campuses or industrial facilities, where it acts as an authentication system to open barriers to authorized plate numbers and tracks history of visits so that security can be audited (8).

8.2 Potential Improvements (Better Detection Models, Deep Learning, Tracking)

A significant area of enhancement is substituting contour-based detection with current object-detection networks like YOLOv8 that are capable of localizing plates more effectively along with any change in scale, occlusions, and cluttered backgrounds (9). The second improvement is to optimize Tesseract on license-plate fonts or replace it with a deep learning text recognizer that has been trained specifically on cropped plates, which has been demonstrated to work better than generic OCR engines in complicated settings (8). Lastly, it is possible to make the system robust to still image data by extending it to include video with multi-frame tracking and temporal smoothing since two or more observations of a plate can be merged to fix frame-wise recognition errors (7).

9.Conclusion

The installed ALPR system illustrates that an appropriately designed pipeline based on grayscale conversion, contrast enhancement, bilateral filtering, Canny edges, contour-based localization and Tesseract OCR can accurately identify license plates under controlled settings with a relatively simple hardware (2). Although it is still not as strong as the deep learning-based solutions implemented in the large-scale intelligent transportation systems, the project manages to demonstrate the entire process of the analysis of the raw image to text output and offers a strong foundation that can be gradually enhanced with the more sophisticated detection and recognition modules (7).

10.References

1. S. Du, M. Ibrahim, M. Shehata and W. Badawy, “Automatic License Plate Recognition (ALPR): A State-of-the-Art Review,” IEEE Trans. Circuits and Systems for Video Technology, 2013.sdnbvc
2. A. Rose and R. Hemalatha, “Contour Based Number Plate Recognition System,” National Conference on Recent Trends in IT, 2016.sdnbvc
3. A. Rose, “OpenCV: Automatic License/Number Plate Recognition (ANPR) with Python,” PyImageSearch Tutorial, 2020.pyimagesearch
4. “Automatic Vehicle Number Plate Recognition System using EasyOCR,” GeeksforGeeks, 2024.geeksforgeeks
5. “AI License Plate Recognition in Automotive Solutions,” Labellerr Blog, 2025.labellerr
6. “Automatic Vehicle Number Plate Recognition Approach Using Color and Contour Features,” International Journal of Innovative Science and Technology, 2023.journal.50sea
7. “Automatic vehicle number plate detection and recognition systems,” in Intelligent Data Security Solutions, Elsevier, 2023.sciencedirect
8. “License plate recognition system for complex scenarios based on deep learning,” Scientific Reports, Nature, 2025.nature
9. “Optimized YOLOv8 for Automatic License Plate Recognition on Embedded Platforms,” Engineering, Technology & Applied Science Research, 2025.etasr
10. “Enhanced Vehicle Detection and License Plate Recognition for Intelligent Transportation,” IEEE Xplore, 2024.ieeexplore.ieee
11. “Bilateral filter,” Wikipedia article, accessed 2025.wikipedia
12. J. G. Verwer et al., “Suitability of bilateral filtering for edge-preserving noise reduction in medical images,” Physics in Medicine & Biology, 2011.pmc.ncbi.nlm.nih
13. S. Paris, P. Kornprobst and J. Tumblin, “Bilateral Filtering: Theory and

Applications," CSE IIT Delhi Technical Report, 2008.cse.iitd
14. "Optimization of bilateral filter parameters for edge preserving impulse noise removal," IEEE Xplore, 2015.ieeexplore.ieee
15. "Tesseract Page Segmentation Modes (PSMs) Explained," PyImageSearch Blog, 2021.pyimagesearch
16. "Improve Accuracy by Tuning PSM Values of Tesseract – Part 2," CloudThat Blog, 2024.cloudthat
17. "Pytesseract | Page Segmentation Modes (PSMs)," Kaggle Notebook, 2023. kaggle
18. "Python OCR Tutorial: Tesseract, Pytesseract, and OpenCV," Nanonets Blog, 2025.nanonets
19. "Tesseract is very poor on single line images," GitHub issue thread, tesseract-ocr/tesseract, accessed 2025.github
20. "Pytesseract OCR multiple config options," StackOverflow Q&A, 2020.stackoverflow