iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IX **Month of publication:** September 2025

DOI: https://doi.org/10.22214/ijraset.2025.74265

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com

Robust ANPR System Using YOLOv8 and TrOCR with Aspect-Ratio-Based Splitting and Post-Processing

Deon Menezes[1], Sajal Patel[2], Abdul Shaikh[3], Nishaad Parikh[4]
[1]*CEO/Research Engineer, Virelity Pvt. Ltd*
[2, 3, 4]*AI/ML Intern, Virelity Pvt. Ltd.*

Abstract: ***In this paper we present a modular and efficient Automatic Number Plate Recognition (ANPR) system which combines YOLOv8 - based plate detection and transformer-based OCR via TrOCR. A novel aspect-ratio-based splitting strategy helps it recognize vertically stacked number plates. The system has webcam capture and logs cleaned time-stamped output as part of a clear process to follow at some later point in time. Our approach has significant improvements to accuracy and robustness.***

I. INTRODUCTION

ANPR systems are pushed into more fields, such as traffic monitoring, tolling, parking, and security. Traditional OCR systems exhibit a fundamental disadvantage when used in outof-control environments because of changes in plate layout, lighting, and stacking. This paper establishes a deep learningbased pipeline involving YOLOv8 and TrOCR, alongside original alignment and processing logic to enable high confidence.

II. METHODOLOGY

A. System Architecture

The pipeline consists of the following steps:

1) Capturing input (image or webcam)
2) Detecting the number plate using YOLOv8
3) Working out aspect ratio, separate and follow alignment if needed
4) Running OCR with TrOCR model
5) Cleaning output (converting to upper case + alphanumeric filter)
6) Save output with timestamp
7) Optionally displaying the processed plate image

B. YOLOv8 for Plate Detection

Using a custom-trained YOLOv8 model, we localize number plates based on the input image. The YOLOv8 model outputs bounding boxes, which are used to quantify the crop • of the plate region for further processes.

Fig. 1. Camera sees numberplate

C. Aspect-Ratio-Based Splitting

Plates with height greater than width (aspect ratio less than 2.3) are likely vertically stacked. These are split horizontally into two parts and then concatenated side by side: aligned plate = concat(resize(top),resize(bottom))

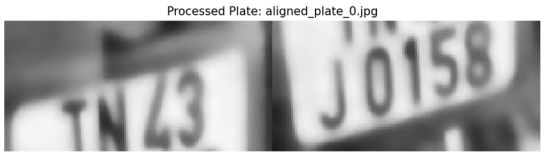


Fig. 2. Camera splits numberplate

D. OCR via TrOCR

We used Microsoft's TrOCR-base-printed model, which employs a ViT encoder and GPT2-style decoder to process sequences. The input plate image first goes through a processor that normalizes the input for the model.

E. Post-processing

To standardize output:

Text is converted to uppercase using Python's upper(). Non-alphanumeric characters are removed using re.sub("[^{A}– *Z*0 − 9]","",*text*). This turns noisy predictions like "tn43-j@0158" into clean, usable formats like TN43J0158.

F. Logging and Debugging

Every OCR result is saved to a text file with a timestamp for Eg.:

Timestamp: 2025-07-07 15:44:21

Detected Plate: TN43J0158

Images such as cropped and aligned plates are saved locally for manual inspection and debugging.

III. VISUAL WORKFLOW

Figure 3 illustrates the end-to-end ANPR pipeline, starting from image capture and ending with the cleaned OCR result. The workflow includes key components like YOLOv8 for detection, aspect-ratio-based plate alignment, TrOCR inference, and post-processing.

IV. CHALLENGES FACED

While building the ANPR system, several practical and technical challenges were encountered:

A. Two-Row Number Plates

Some vehicle registration plates in India present the registration number in two lines or rows. This presentation causes problems for OCR models that have been trained on sequences that are just a single line. So we introduced an aspect-ratio based splitting heuristic: - plates that have a height greater than width ratio (usually ¡ 2.3) are split horizontally into top and bottom halves, made to the same size then stitched together to appear side-by-side as a single line of text. However, accurately vertically aligning and resizing made split text is still error-prone from angled or skewed views.

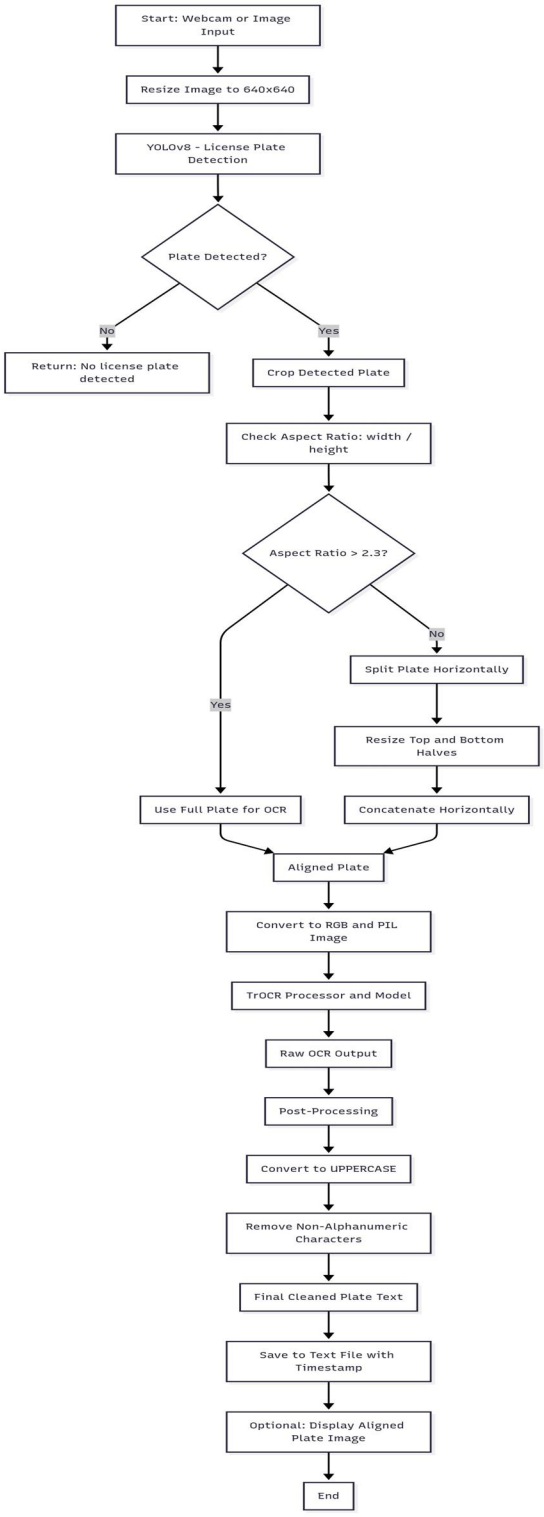
B. Nutbolts and Impurities Misclassified as Characters

Artifacts such as screws, bolt heads, rust spots, or sticker damage near characters are often captured as valid text when OCR processing occurs. These noisy features cause the model to hallucinate or add unwanted characters. This was somewhat mitigated with the use of morphological cleaning in preprocessing and character-level filtering in postprocessing but even with this added step, the model will misdetect in low resolution.

C. Misidentification of Characters and Words

Due to variations in fonts, blurriness, illumination, and image compression, certain characters (e.g., 'O' vs. '0', 'B' vs. '8', 'I' vs. '1') are frequently confused. This misidentification leads to invalid or incorrect plate numbers. We attempted to reduce such confusion through case normalization, Fig. 3. End-to-end ANPR Pipeline: from input capture to cleaned OCR output alphanumeric-only filtering, and evaluation of edit distance to expected plate formats. However, improving OCR accuracy under these conditions still requires fine-tuning or domainspecific language modeling.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com



V. RESULTS

A. Dataset and Setup

The YOLOv8 model was performed fine-tuning on a dataset generated from Roboflow that included various Indian license plates. The TrOCR model was only used in inference mode and not fine-tuned. Testing was done on bike and car plates in the real work setting.

B. Performance

TABLE I
SYSTEM PERFORMANCE

Component	Metric
YOLOv8 Detection	mAP@0.5 = 94.6%
TrOCR OCR	CER = 2.3%
ANPR Pipeline	Word Accuracy = 91.8%

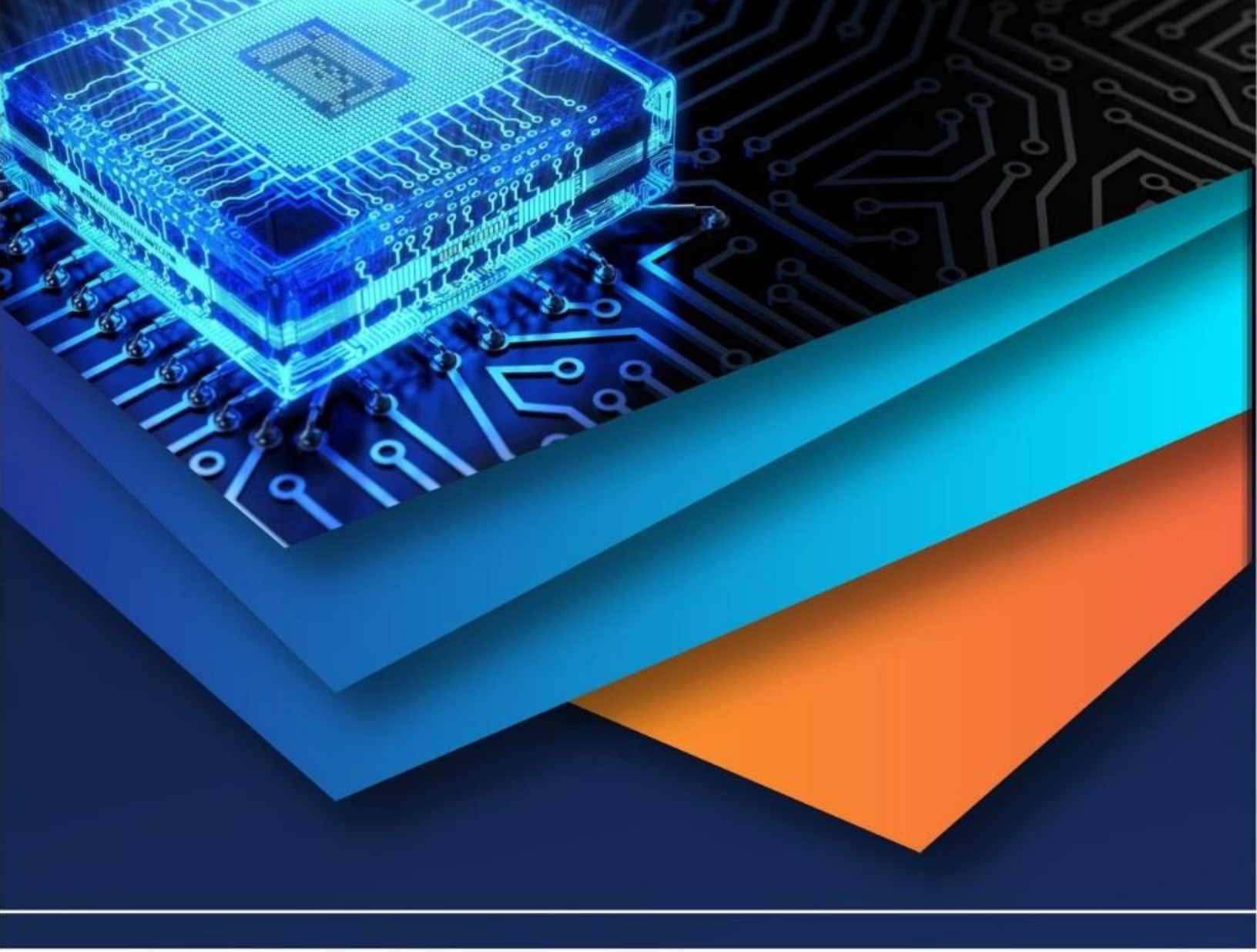
Post-processing (case normalization + filtering) significantly improved the clarity of predictions.

VI. CONCLUSION

We present a comprehensive and practicable ANPR system based upon YOLOv8 and TrOCR with added reasoning process to handle stacked plates correctly, and post-processing to generate clean outputs. The full system can accept a webcam feed in real time and has been proven to perform across a spectrum of lighting conditions and vehicle style plates. Future work will include real time multiple vehicle tracking and support for multi-language plates.

REFERENCES

[1] Jocher, G., et al., "YOLOv8: Ultralytics 2023," https://github.com/ ultralytics/ultralytics, 2023.
[2] Li, M., et al., "TrOCR: Transformer-based Optical Character Recognition," arXiv preprint arXiv:2109.10282, 2021.
[3] Roboflow: Label and Train Custom Object Detection Models, https:// roboflow.com.
[4] Baek, J., Kim, G., Lee, S., et al., "What is Wrong with Scene Text Recognition Model Comparisons?" ICCV, 2019.
[5] Silva, S., Jung, C. R., "License Plate Detection and Recognition in Unconstrained Scenarios: A Survey," IEEE Trans. on Intelligent Transportation Systems, vol. 18, no. 2, pp. 377–391, Feb. 2017.
[6] Redmon, J., Farhadi, A., "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018.
[7] Zhang, H., et al., "Real-Time Automatic License Plate Recognition Based on Efficient Deep Learning Models," Sensors, vol. 22, no. 3, 2022.
[8] Smith, R., "An Overview of the Tesseract OCR Engine," Proc. ICDAR, 2007. Discusses post-processing strategies for noisy OCR outputs.
[9] OpenALPR Benchmark Datasets, https://github.com/openalpr/ benchmarks, accessed July 2025.
[10] Liao, M., Shi, B., Bai, X., Wang, X., and Liu, W., "TextBoxes: A Fast Text Detector with a Single Deep Neural Network," AAAI, 2017.
[11] Wang, T., Wu, D. J., Coates, A., and Ng, A. Y., "End-to-End Text Recognition with Convolutional Neural Networks," ICPR, 2012.
[12] Huang, C. S., and Lee, H. J., "A Real-Time and Robust License Plate Localization Algorithm Based on Multiscale Morphological Processing," IEEE ICIP, 2004.
[13] Zherzdev, S., and Gruzdev, A., "LprNet: License Plate Recognition via Deep Neural Networks," Journal of Open Source Software, vol. 3, no. 30, 2018.
[14] Liu, Y., Chen, H., Shen, C., He, T., Jin, L., and Wang, L., "ABCNet: Real-time Scene Text Spotting with Adaptive Bezier-Curve Network," CVPR, 2020.
[15] Nayef, N., et al., "ICDAR2019 Robust Reading Challenge on ArbitraryShaped Text—RRC-ArT," Proc. ICDAR, 2019.
[16] Montazzolli, S., and Jung, C. R., "Real-Time Brazilian License Plate Detection and Recognition Using Deep Convolutional Neural Networks," Journal of Electronic Imaging, vol. 26, no. 5, 2017.
[17] Neuhold, G., Ollmann, T., Rota Bulo, S., and Kontschieder, P., "The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes," ICCV, 2017.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY