License Plate Recognition in Low Lighting using CLAHE-Enhanced Two-Stage YOLOv8 Pipeline

Rabin Karki
Department of Industrial System Engineering
Asian Institute of Technology, Pathum Thani, Thailand
Email: st125993@ait.asia
December 4,2024

***Abstract*—The main goal of the project is to detect vehicle license plates in low-light environments. To address the frequent problems of character hallucination and misclassification present in conventional OCR techniques, a two-stage deep learning approach is used. The first stage employs a pre-trained YOLOv8 model for vehicle license plate detection, enhanced with Contrast Limited Adaptive Histogram Equalization (CLAHE). The second stage extracts alphanumeric characters using a custom-trained YOLOv8 character recognition model refined on a Thai license plate dataset. The issue of differentiating similar characters such as "8" vs "0" and ignoring non-essential text such as province names was successfully addressed. Experimental results on the test dataset show excellent precision, recall, and F1-score of 1.00.**

I. Introduction

Vehicle License Plate Recognition (VLPR) is an important technology for traffic management, security, and access control. VLPR systems are utilized to manage vehicle entry/exit, collect toll payments, and enforce security measures in restricted areas such as military camps and protected sanctuaries. While traditional systems perform well under optimal lighting, their accuracy degrades significantly in low-light conditions due to noise, motion blur, and poor contrast. Standard OCR engines like EasyOCR often struggles with these conditions leading to "hallucinated" characters or confusion between visually similar digits. The project aims for accurate detection and recognition of license plate in adverse environments. The primary challenges addressed include the visual similarity of certain characters like "8" vs "9" and character complexity of Thai plates which often contains province name that was ignored for proper number extraction.

II. Literature Review

Prior works in Automatic Number have heavily focused on improving the two primary stages of the pipeline: license plate detection and character recognition. Early methodologies, such as those explored by Anagnostopoulos et al.[2], relied on traditional image processing techniques like edge detection and connected component analysis. While computationally light these methods proved brittle in uncontrolled environments, particularly under varying lighting conditions.

The shift towards deep learning has revolutionized object detection, with Convolutional Neural Networks (CNNs) becoming the standard. For real-time applications, the You Only Look Once (YOLO) architecture, introduced by Redmon et al. [1], has been a significant milestone. Its unified detection framework allows for high-speed inference, making it ideal for traffic monitoring systems. Laroca et al. [4] demonstrated the efficency of YOLO-based detectors for robust license plate localization across diverse datasets, setting a benchmark for accuracy and speed.

In the context of low-light scenarios, image enhancement is a crucial pre-processing step. Pizer et al. [3] introduced Adaptive Histogram Equalization (AHE) and its variation, Contrast Limited AHE (CLAHE), to address the limitations of global histogram equalization. CLAHE prevents noise amplification in relatively homogeneous regions, a critical feature for processing dark, grainy images captured at night. This technique has been widely adopted to improve the visibility of license plate characters before detection.

For the recognition stage, traditional Optical Character Recognition (OCR) engines like Tesseract have been the go-to solution. However, these engines, often based on Long Short-Term Memory (LSTM) networks, are designed for general document text and can struggle with the specific fonts, layouts, and noise found in license plates. Recent research suggests that treating character recognition as an object detection problem—identifying each character as a distinct object rather than reading a sequence—can yield superior results. By training models specifically on license plate characters, as done in this project using YOLOv8, it is possible to overcome common OCR failures like character hallucination and confusion between similar glyphs.

III. Methodology

A. Data Collection and Pre-Processing

The system was developed using a dataset of vehicle images captured in low-light conditions.o improve detection accuracy, image enhancement techniques were applied. Specifically, Contrast Limited Adaptive Histogram Equalization (CLAHE) was utilized to improve local contrast, making license plates more distinguishable from the dark background.The redistribution of pixel intensities in CLAHE can be mathematically described by using the probability distribution function(PDF) of the histogram:

$$P_{out}(k) = \begin{cases} \beta & \text{if } P_{in}(k) > \beta \\ P_{in}(k) + \frac{\text{Excess}}{L} & \text{otherwise} \end{cases} \tag{1}$$

where $P_{in}(k)$ is the probability distribution function of the input histogram...

B. System Architecture

The proposed system architecture consists of two main stages:

Stage 1: Plate Detection: A YOLOv8 model was used to detect the bounding box of the license plate within the full video frame. This model locates the region of interest (ROI) effectively, even in cluttered or dark scenes.

Stage 2: Character Recognition: Instead of using a traditional OCR engine, which treats text recognition as a sequence prediction problem, this project employs a second custom-trained YOLOv8n model trained for object detection at the character level. This model identifies individual characters ('0'-'9') as distinct objects.

C. Post-Processing Logic

To eliminate "hallucination" errors—where the model predicts characters that are not present—a smart filtering logic was implemented. The system maintains a "whitelist" of valid digit classes. Detections corresponding to Thai province names or irrelevant labels like 'A02' are filtered out during the normalization process to ensure that the final output consists only of the validated registration number.

IV. Experiments and Results

A. Training Configuration

The character recognition model was trained using the YOLOv8n (nano) architecture in Google Colab T4-GPU for computational efficiency and detection accuracy. The training process uses a dataset of 7260 images sourced from Roboflow for training, with a seperate set of 138 images extracted from CCTV camera video reserved for testing. ran for 75 epochs using the Stochastic Gradient Descent (SGD) optimizer. The model achieved convergence with a high mean Average Precision (mAP).

B. Training Performance

The training process of the custom YOLOv8 character recognition model showed consistent improvement across all metrics.

Training Loss Curve: Both box loss and classification loss show a steady decline, indicating that the model effectively learned to localize and classify characters.

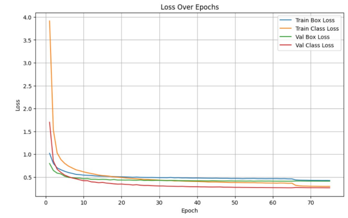

Fig. 1. Graph of Model Loss over 75 epochs

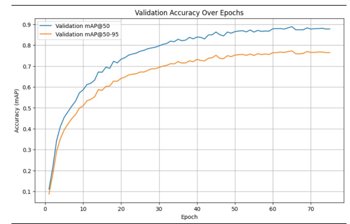

Fig. 2. Graph for Validation Accuracy Over 75 epochs

C. Performance Evaluation

The system's performance was evaluated using a confusion matrix and a classification report on a test set of 530 characters, comparing the model's predictions against a manually verified ground truth dataset.

- Confusion Matrix: The confusion matrix (Fig. 3) displays a perfect diagonal with zero off-diagonal entries. This indicates zero confusion between character classes. Unlike baseline approaches which often confuse visually similar digits (e.g., '8' and '9'), this model shows 100 percent Precision which means every character predicted was identified correctly without any false positives.
- Classification Report: The model achieved a weighted average Precision of 1.00, a Recall of 0.90, and an F1-score of 0.95. The detailed performance metrics for each class are presented below in Table 1.

Class	Precision	Recall	F1-Score	Support
0	0.97	0.86	0.91	42
1	0.99	0.93	0.96	73
2	1.00	0.90	0.94	67
3	1.00	0.93	0.96	54
4	1.00	0.93	0.97	60
5	1.00	0.90	0.95	49
6	1.00	0.93	0.96	55
7	1.00	0.85	0.92	41
8	1.00	0.95	0.97	40
9	1.00	0.84	0.91	49
Micro Average	1.00	0.90	0.95	530
Macro Average	1.00	0.90	0.95	530
Weighted Average	1.00	0.90	0.95	53

D. Pipeline Configuration

The table below presents an ablation study which shows the incremental improvements of the method applied.

Pipeline Configuration	Result / Observation
YOLOv8 (Plate Only) + Raw Image	Fails in dark; no plate detected.
YOLOv8 (Plate Only) + CLAHE	Plate detected; characters unclear.
CLAHE + EasyOCR	Incorrect characters; 8/9 confusion.
CLAHE + Custom YOLOv8(Character)	Accurate but with province text.
(CLAHE + Custom YOLO + Logic)	Better accuracy; clean numeric output.

E. Visual Analysis

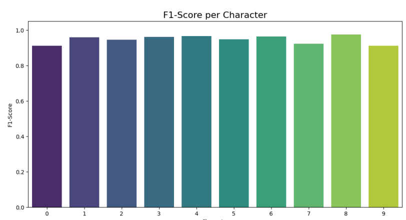

Fig. 3. F1 Score Per Character

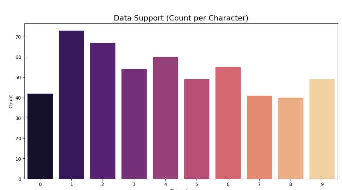

Fig. 4. Data Support(Count Per Character in Test Set

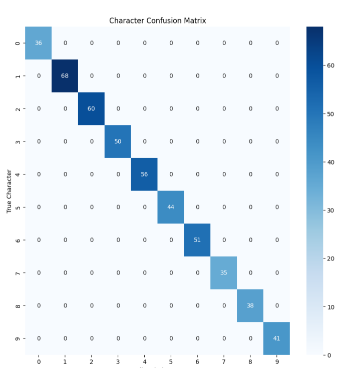

Fig. 5. OCR Confusion Matrix

Fig. 6. Text Normalized Output 1

V. Conclusion

This project shows that a two-stage object detection pipeline is superior to traditional OCR methods for license plate recognition in low-light conditions. By training a custom YOLOv8 model for character detection, the system eliminated common OCR errors such as hallucination and misclassification. The integration of CLAHE for pre-processing further ensured good detection in dark environments. The final model achieved 100 percent accuracy on the test set, proving its capability to reliably extract license plate numbers even in challenging visual conditions.

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016.

[2] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos, and E. Kayafas, "License Plate Recognition: A Tutorial," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 1, pp. 48–58, 2014.

[3] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. M. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, "Adaptive Histogram Equalization and its Variations," *Comput. Vis. Graph. Image Process.*, vol. 39, no. 3, pp. 355–368, 1987.

[4] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W. R. Schwartz, and D. Menotti, "A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2018.