www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 17(4), 2026, 57-73. ISSN: 2007-1558.
https://doi.org/10.61467/2007.1558.2026.v17i4.1294

Fine-Tuning TrOCR for Automatic Number Plate Recognition: A YOLOv11-Based Comparison with Tesseract and EasyOCR

*Manuel Antonio García-Herrera, Betania Hernández-Ocaña, José Hernández-Torruco, Oscar Chávez-Bosquez**

Universidad Juárez Autónoma de Tabasco, División Académica de Ciencias y Tecnologías de la Información, Cunduacán, Tabasco, México
✉Corresponding author: oscar.chavez@ujat.mx

Abstract. Automatic Number Plate Recognition (ANPR) systems typically involve two stages: localizing the license plate in an image and recognizing its characters via Optical Character Recognition (OCR). While deep learning has significantly improved detection accuracy—with models like YOLOv11 offering robust, real-time performance—the OCR stage remains an open issue, especially under challenging conditions such as motion blur, low illumination, or partial occlusion. This paper evaluates and compares three OCR engines—Tesseract, EasyOCR, and TrOCR—within an ANPR pipeline using YOLOv11 for detection in a real-world university campus environment. A key contribution is the fine-tuning of TrOCR on a custom dataset of 651 license plate images, designed to capture domain-specific visual patterns and noise. Performance is quantified using detection metrics (precision, recall) and recognition accuracy (Character Error Rate, CER). YOLOv11 demonstrated high detection performance (precision: 0.938, recall: 0.995). In the baseline OCR comparison, TrOCR outperformed the other engines (CER: 0.461) versus EasyOCR (0.626) and Tesseract (1.027). Following fine-tuning, TrOCR's CER improved by approximately 44% to 0.258, confirming that retraining a robust model can enhance character recognition accuracy in difficult, real-world scenarios.
Keywords: Computer vision, Deep Learning, Transformer architecture.

Article information

Received: Jan 16, 2026
Accepted: May 19, 2026

1 Introduction

Automatic Number Plate Recognition (ANPR) is an innovative technology applicable across sectors such as public safety, intelligent transportation systems, and smart urban infrastructure. As demand for efficient vehicle identification grows in areas such as toll collection, access control, and traffic monitoring, the development of reliable ANPR systems has been the subject of extensive research (Hamood Bakhtan et al., 2016). Generally, the ANPR workflow consists of 2 primary phases: localizing and detecting the license plate within an image, and recognizing the alphanumeric characters from the extracted plate region using Optical Character Recognition (OCR) techniques.

Recently, advances in Deep Learning have substantially improved the accuracy of both detection and recognition tasks in ANPR pipelines. Convolutional Neural Networks (CNNs) and object detection frameworks like the YOLO (You Only Look Once) family are now standard for achieving real-time, high-precision plate detection (Laroca et al., 2021). Specifically, this study uses YOLOv11, a state-of-the-art object detection model, chosen for its speed and dependability in identifying license plate regions across a variety of environmental conditions (Jocher & Qiu, 2024).

Following detection, OCR engines extract characters from the plates. While numerous OCR models have been proposed and successfully deployed, their performance may vary based on input quality, lighting, viewing angles, and the presence of noise or occlusion. Among the widely used OCR engines, Tesseract remains a traditional open-source option, while EasyOCR introduces neural network-based enhancements. More recently, transformer-based models like TrOCR have been introduced, utilizing sequence-to-sequence learning to boost robustness in text recognition tasks (Li et al., 2023). However, the performance of pre-trained OCR models is often constrained when applied to data such as license plates, where visual patterns and noise characteristics differ significantly from those found in generic text datasets.

This study evaluates and compares Tesseract, EasyOCR, and TrOCR when integrated into an ANPR system using YOLOv11 for detection in a realistic scenario on one of our University campuses. Additionally, this paper aims to incorporate and evaluate a fine-tuned TrOCR model within the proposed ANPR pipeline, trained on a custom dataset created specifically for this purpose. Evaluation metrics such as precision, recall, and character error rate (CER) are used to quantify OCR detection (Tavares, 2024). Specifically, the contributions of this paper include:

- A real-world ANPR test case was designed at a university campus entrance under uncontrolled environmental conditions.
- Comparative analysis between the baseline versions of Tesseract, EasyOCR, and TrOCR.
- A domain-specific fine-tuning strategy for TrOCR was implemented and evaluated.
- A statistical exploration (mean, median, percentiles, and failure categorization) was incorporated to assess performance beyond global-average metrics.
- An empirical analysis of recognition errors under true challenging conditions (motion blur, low illumination, occlusion, character confusion) was provided.

2 Related works

Numerous studies have explored Automatic Number Plate Recognition (ANPR) systems, predominantly using deep learning methodologies that benefit from modular detection and recognition pipelines (Zherzdev & Gruzdev, 2018). For detection, YOLO-based models are widely favored for their real-time inference and localization accuracy, with improvements in backbone architectures and training strategies enhancing performance across versions (Wang et al., 2022). In the field of OCR, approaches have evolved from classic engines like Tesseract (Anwar et al., 2022) to CNN-based solutions like EasyOCR (Dhyani & Kumar, 2023), and recently to transformer-based models such as TrOCR, which leverage attention mechanisms to achieve state-of-the-art results on various benchmarks.

Current research focuses on optimizing ANPR for resource-constrained environments while maintaining high accuracy. For instance, the Light-Edge architecture was validated on low-power hardware and shown to be robust against oblique views, motion blur, and glare (Sonnara et al., 2025). Similarly, optimized YOLOv8 training strategies have enabled reliable performance under low-light and occlusion conditions on edge devices (Satya et al., 2025), while tailored augmentation schemes have enabled models like YOLOv8s to withstand severe distortions such as rain and glare (Al-Hasan et al., 2024). Additional work has addressed challenges such as script variation in mixed lighting (Sarhan et al., 2024), cloud deployment costs across YOLO versions, and perspective distortion correction via segmentation to achieve high character accuracy on edge devices (Nguyen, 2025). Furthermore, studies have highlighted the importance of evaluating robustness against adversarial attacks alongside raw accuracy (Vizcarra et al., 2024).

In the field of OCR, classical engines such as Tesseract were used to extract characters from license plates. EasyOCR introduced CNN-based enhancements that improved recognition robustness under moderate distortions. Transformer-based models such as TrOCR have emerged, using pre-trained language models and attention mechanisms to achieve state-of-the-art results across various text recognition benchmarks, including scene, printed, and handwritten text. These architectures proved their utility in a variety of inputs, making them suitable candidates for complex ANPR scenarios.

Recent research indicates that transformer-based OCR models, such as the TrOCR architecture, which combines a vision encoder with a transformer-based text decoder, benefit significantly from domain-specific fine-tuning for structured text recognition tasks (Lauar & Laurent, 2024). Originally pre-trained on large-scale synthetic and real-world datasets, TrOCR captures task-specific visual and linguistic patterns through adaptation, thereby improving accuracy across diverse text domains. Additionally, studies show that retraining or fine-tuning OCR models on datasets that reflect real-world distortions, such as blur, low resolution, and illumination variability, substantially reduces recognition errors. These findings support the implementation of targeted fine-tuning strategies in ANPR pipelines, where license plates present constrained vocabularies and distinct visual characteristics compared to general scene text (Baek et al., 2019).

Similarly, integrated detection–recognition architectures have been introduced to overcome performance degradation in complex environments. In the YOLOv5-PDLPR framework, YOLOv5 is employed for license plate detection, while a dedicated recognition module is designed for character extraction. This module incorporates a Multi-Head Attention mechanism for precise character identification, a global feature extractor to enhance the completeness of the representation, and a parallel decoder to improve inference efficiency. Experimental results have shown that this combined approach achieves accurate, real-time recognition with strong robustness under challenging conditions (Tao et al., 2024).

More recently, modular ANPR systems have been proposed to enhance flexibility and efficiency in real-world applications. In one such approach, YOLOv8-based plate detection is integrated with transformer-based OCR using TrOCR to form a unified recognition pipeline. An aspect-ratio-based splitting strategy is incorporated to enable the accurate recognition of vertically stacked license plates. Additionally, webcam-based image acquisition and time-stamped logging of processed results are implemented to ensure traceability. Experimental findings indicate that this configuration provides notable improvements in accuracy and robustness (Menezes et al., 2025).

3 Custom dataset

We collected a set of videos over two days within the same week, deliberately introducing diverse environmental conditions to ensure representativeness. Recording sessions took place under both cloudy and sunny skies, spanning intervals between 07:00 and 15:00. Special focus was placed on peak traffic hours—specifically 08:00, 10:00, and 12:00—to capture natural variations in traffic density and lighting. All footage was shot from a fixed vantage point on the left side of the main entrance to the Universidad Juárez Autónoma de Tabasco's Chontalpa Campus in Tabasco, México. This site, pre-selected by campus administration, was chosen based on the sun's trajectory throughout the day to account for shifting shadow and glare patterns. An image of the test environment is shown in Figure 1.

Fig. 1. Single entrance to the Chontalpa Campus.

3.1 Vehicles dataset (video)

A compilation of 25 MP4 videos in which 198 vehicles were recorded passing through the camera's field of view. The dataset prioritizes capturing license plates across a wide range of visual scenarios, including lighting effects such as glare and shadow, oblique angles caused by vehicle movement and positioning, and partial occlusions from surrounding objects or other vehicles. Incorporating these variations allows the dataset to mirror real-world traffic conditions, serving as a rigorous testing platform to assess the ANPR system's effectiveness and adaptability.

All video data was standardized using a preprocessing pipeline. Original videos were cropped to a 1:1 aspect ratio and resized to 848×848 pixels to optimize memory usage, reduce computational load, and improve the performance of detection and tracking algorithms. Figure 2 illustrates this preprocessing workflow.

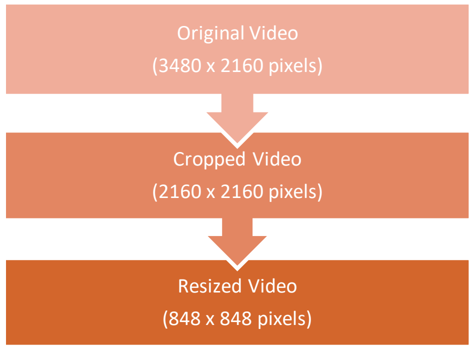


Fig. 2. Video preprocessing steps.

3.2 Plates dataset (image)

A total of 651 license plate images were used for fine-tuning. We include 2 to 4 instances of each plate in the vehicles dataset (198 license plates), capturing plates at different distances, angles, and illumination conditions, to expand the base dataset with plates under different conditions. The existing literature suggests that this number is sufficient to fine-tune a model using transfer learning (Rather et al., 2024; Shu, 2025). The resulting dataset was split into training (90%) and test (10%) subsets to enable effective evaluation during training. All images were resized and converted to RGB, and text labels were standardized (uppercase and stripped). Figure 3 shows the steps performed.

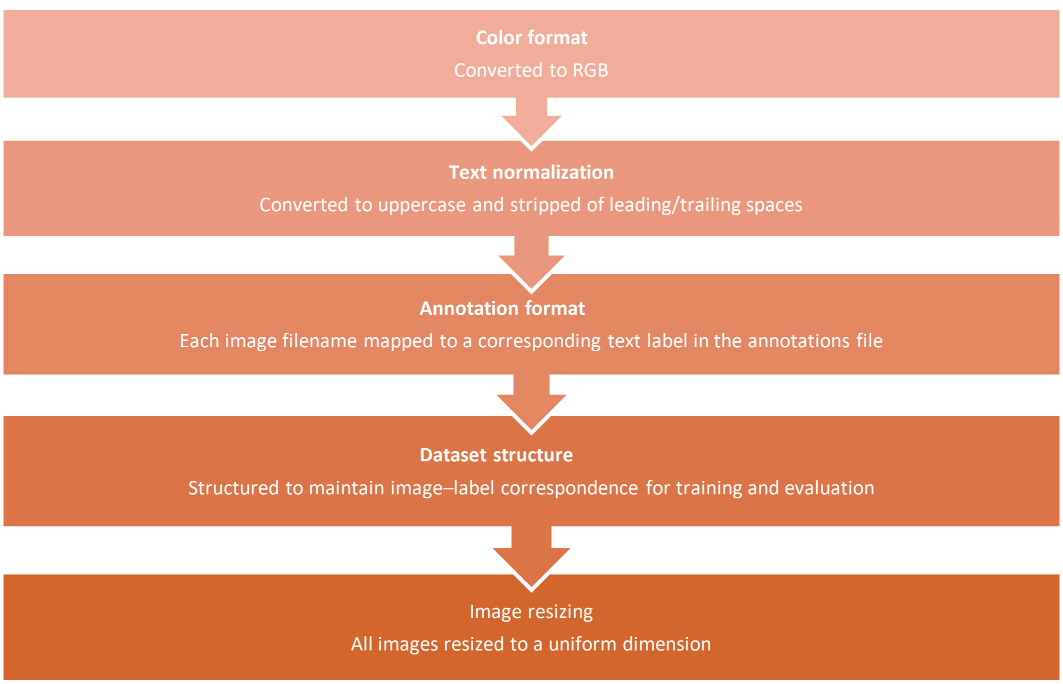


Fig. 3. Image processing steps.

The dataset was structured such that each image filename corresponded to a textual label in the annotations file, as exemplified in Table 1.

Table 1. Nomenclature filename and annotations

Image	Annotation
001.jpg	WLD-715-B
002.jpg	DNF-410-E
003.jpg	WSY-425-A
004.jpg	WWD-310-B
…	…
651.jpg	WWZ-957-B

Figure 4 shows examples of license plates used in the re-training phase. As we can notice, some plates are captured at different angles and with different illuminations.

Fig. 4. Example of standardized license plates.

4 Baseline OCR Evaluation

The experimental setup was designed to comprehensively evaluate the performance of the OCR models. The evaluation process was divided into two main stages: license plate detection and Optical Character Recognition (OCR).

First, license plates were detected in each video frame using the YOLO object detection algorithm. YOLOv11 was selected for its state-of-the-art performance in real-time object detection, particularly for small and complex objects such as vehicle license plates. The detection stage evaluation was conducted using two standard metrics: precision and recall. These metrics evaluate the model's performance by identifying true license plates while minimizing false detections. Once the license plates were located, the three OCR engines (Tesseract, EasyOCR, and TrOCR) were applied independently to the cropped license plate regions. Tables 2, 3, and 4 show the main parameters and values of the 3 OCR engines. This enabled a comparative analysis of each OCR method’s effectiveness at character recognition in the collected dataset.

Table 2. Tesseract configuration

Parameter	Value	Description
OEM	OEM DEFAULT	Auto engine selection
PSM	PSM AUTO (3)	Auto text layout
Language Data	*.traineddata	Basic dictionary rules
Preprocessing	Enabled	Binarize + deskew

Table 3. EasyOCR configuration

Parameter	Value	Description
lang_list	[’en’]	List of languages
gpu	True	Use GPU if available, else CPU
recog_network	standard	OCR model (standard, simplified)
detector	True	Enable text detection
recognizer	True	Enable text recognition

Table 4. TrOCR configuration

Parameter	Value	Description
trocr_processor	Required	Image preprocessor
trocr_model	Required	Text recognition model

device	'cuda'	Processing device
batch_decode	True	Skip special tokens

The complete workflow for this stage is summarized in Figure 5.

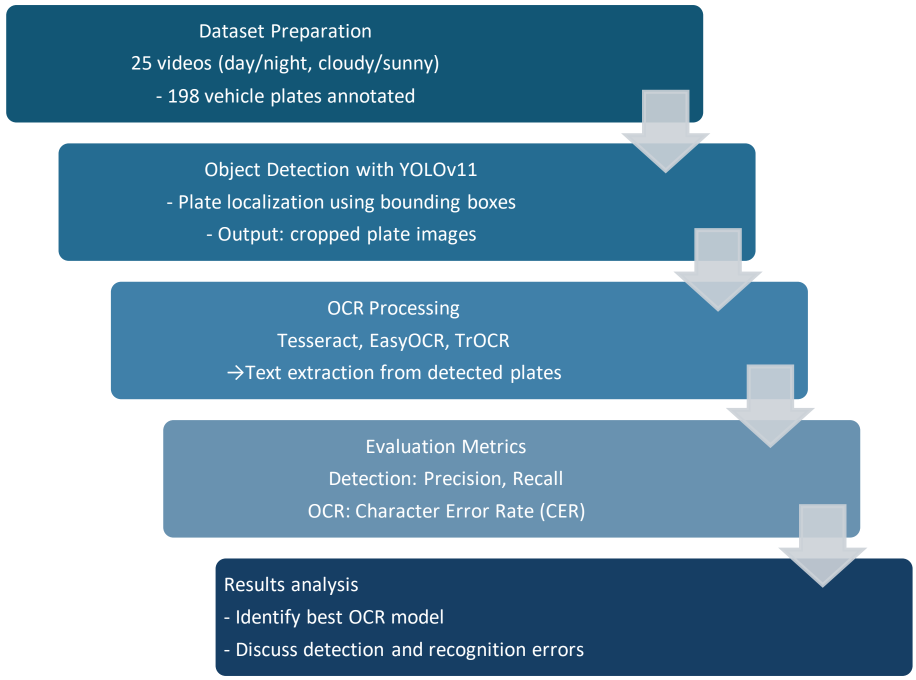


Fig. 5. Comparison analysis workflow.

4.1 Baseline ANPR Pipeline

The ANPR pipeline shown in Figure 6 was selected to process video input and to evaluate OCR performance. The methodology includes the following stages:

- **Frame extraction:** Individual frames were isolated from continuous video sequences to simulate real-time traffic surveillance footage. This approach enabled controlled variations in image quality, motion blur, and lighting conditions.
- **License plate detection:** YOLOv11 was employed to detect and localize license plates within each frame. The model generated bounding boxes to identify plates, which were then used to crop the Regions of Interest (RoIs).
- **RoI extraction:** The detected license plate regions were cropped from the original frames and prepared for OCR processing. Preparation involved basic pre-processing steps, such as resizing and normalization, to guarantee input consistency across the various OCR engines.
- **OCR:** Tesseract, EasyOCR, and TrOCR were used to record character-level and full-plate outputs for subsequent analysis.

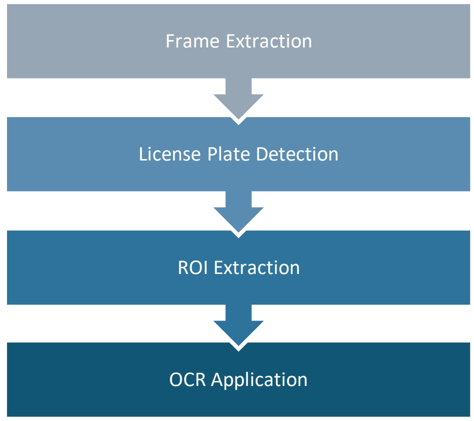


Fig. 6. ANPR pipeline.

All experiments were executed with GPU acceleration via Google Colab, a cloud-based platform that enables users to run Python code in a browser environment with access to hardware accelerators such as GPUs (Google, 2025). This methodological framework ensured that each TrOCR version was evaluated using consistent inputs and detection outputs, facilitating a fair and objective comparison of their recognition capabilities.

4.2 Plate detection results

To measure the performance of the YOLOv11 object detection model, a confusion matrix is presented in Table 5. This matrix illustrates the results for the license plate detection task.

Table 5. Confusion matrix for plate detection

		Predicted	
		Positive (1)	Negative (0)
Actual	Positive (1)	**197** True Positive (TP)	**1** False Negative (FN)
	Negative (0)	**13** False Positive (FP)	–

The YOLOv11 model's performance in detecting license plates in video frames was evaluated using standard metrics. From Table 5, the results show that the model detected 197 of the 198 license plates, missing only 1. Additionally, the model detected 13 fake license plates, including car brands or bumper stickers on some vehicles. Figure 7 shows a false-positive example (left) and the solely false-negative case.

Fig. 7. False positive (right) and false negative (left) cases.

Precision quantifies the proportion of correct positive detections relative to all positive detections made. In this case, it is defined as the ratio of correctly detected license plates (true positives) to the total number of license plates detected (true positives plus false positives). Precision is calculated as follows (Sporici et al., 2020):

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

Recall measures the detector's ability to identify all actual license plates in the dataset. It is computed as the ratio of correctly detected license plates to the total number of license plates in the dataset (i.e., the sum of true positives and false negatives). This metric measures the model's ability to identify all relevant instances and is expressed as (Spruck et al., 2021):

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

Based on these values, detection performance metrics are presented in Table 6.

Table 6. Performance metrics for plate detection

Metric	Value
Precision	0.9380
Recall	0.9950

The precision obtained indicates that 93.8% of the detected license plates were genuine, with a relatively low rate of untruthful plates. This near-perfect recall demonstrates that almost all license plates in the test set were successfully detected, with only 1 plate missed. These results indicate that the YOLOv11 model achieved high performance in the license plate detection task, particularly under challenging conditions in the videos, such as low light, motion blur, and distorted text.

4.3 Baseline ANPR comparison

The second stage of the pipeline involved applying OCR engines to the previously extracted RoI from detected license plates. The performance was measured using the character error rate (CER), which evaluates the accuracy of recognized text by counting character-level errors relative to the total number of characters in the ground truth. CER takes into account insertions, deletions, and substitutions of characters (Drobac & Lindén, 2020):

$$\text{CER} = \frac{S + D + I}{N} \tag{3}$$

where:

- S is the number of substitutions,
- D is the number of deletions,
- I is the number of insertions,
- N is the total number of characters in the reference text (ground truth).

A lower CER indicates better recognition performance, with 0 as perfect text recognition (Thennal et al., 2024). As an illustrative example, Table 7 presents the results for three license plate recognition experiments.

Table 7. Example OCR outputs for three experiments

Video / Image	Real plate	Tesseract		EasyOCR		TrOCR	
		Output	**CER**	**Output**	**CER**	**Output**	**CER**
21.mp4 / 377.jpg	NS-7803-A	...	1.000	NS.7803.	**0.333**	VS.7803.A	**0.333**
15.mp4 / 141.jpg	YGZ-112-A	...	1.000	NGZ.742.4	0.444	GTYGZ.112.4	0.556
16.mp4 / 182.jpg	WTT-269-A	...	1.000	NTT.269.	0.667	MIT.269.1	0.556

The actual plate number was compared with the outputs of each OCR engine to evaluate their accuracy in isolating and recognizing the correct alphanumeric sequence shown in Figure 8.

Fig. 8. Example OCR outputs for three experiments.

The CER values obtained for each OCR engine are presented in Table 8.

Table 8. OCR performance: Character Error Rate (CER)

OCR Engine	CER
Tesseract	1.0271
EasyOCR	0.6260
TrOCR	**0.4605**

Among the three evaluated engines, TrOCR achieved the lowest CER (0.4605), indicating the highest recognition accuracy. EasyOCR followed with a CER of 0.6260, while Tesseract yielded the highest error rate (1.0271). These results suggest that the transformer-based architecture of TrOCR offers enhanced robustness when processing complex or degraded license plate images, particularly those affected by motion blur, low contrast, or partial occlusion. Substitutions, insertions, and deletion errors were occasionally encountered, particularly in cases where characters were partially obscured or distorted. These error types contributed significantly to the overall CER, especially in the Tesseract and EasyOCR outputs.

Despite these challenges, TrOCR demonstrated greater capacity to recognize correct sequences under adverse conditions, thanks to its attention mechanism and context-aware decoding. This finding highlights the potential of transformer-based OCR models for deployment in real-world ANPR systems, where imaging conditions are often unpredictable.

In addition, execution times varied across OCR engines despite GPU acceleration in Google Colab. Specifically, EasyOCR required approximately 3 to 5 minutes per video, TrOCR between 4 and 6 minutes, and Tesseract demonstrated the fastest performance, ranging from 1 to 2 minutes. These runtime differences depend on the model's complexity and the ANPR's processing efficiency.
In summary, while YOLOv11 achieved highly accurate license plate detection, the choice of OCR engine significantly impacted recognition performance. Transformer-based models, such as TrOCR, exhibited measurable improvements over traditional approaches, making them more suitable for robust, scalable ANPR solutions.

5 TrOCR re-training

TrOCR, a modern transformer-based OCR model that incorporates pre-trained language models for better contextual understanding (Ströbel et al., 2022), is used in its base configuration for fine-tuning in real-world ANPR conditions.

5.1 Fine-tuning strategy

The base architecture of `microsoft/trocr-base-printed` was retrained. The encoder-decoder structure was preserved, and only the parameters relevant to sequence generation were fine-tuned. The pre-trained model is made available under a free license at https://huggingface.co/microsoft/trocr-base-printed. Pixel values of each image in the dataset were extracted using the TrOCR processor, and textual annotations were tokenized and padded/truncated to a fixed length for sequence modeling using the TrOCR tokenizer.

Fine-tuning was performed for 5 epochs with a batch size of 4, a learning rate of 5e-5, weight decay of 0.01, and 50 warmup steps. Gradient updates were computed using the AdamW optimizer, and mixed-precision (fp16) training was applied when GPU resources were available. Table 9 shows the full re-training parameters.

Table 9. Re-training TrOCR parameters

Parameter	Value / Description
Number of epochs	5
Batch size	4 (per device)
Learning rate	5×10^{-5}
Weight decay	0.01
Warmup steps	50
Optimizer	AdamW
Mixed-precision training	Enabled (fp16) when GPU resources are available
Logging steps	50
Save steps	200
Maximum number of checkpoints	2

Retraining was performed on a GPU-enabled system provided by Google Colab, using PyTorch and Hugging Face Transformers, ensuring efficient computation and reproducible results. The complete fine-tuning and evaluation workflow, including dataset preparation, preprocessing, training, evaluation, and results analysis, is depicted in Figure 9.

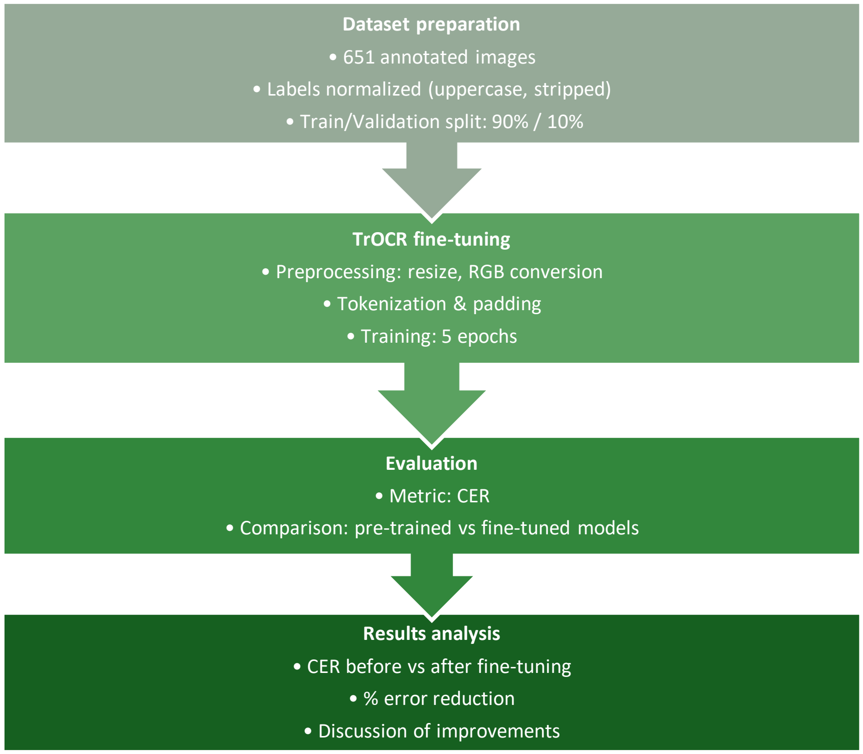


Fig. 9. Workflow for Fine-tuning TrOCR evaluation.

5.2 Baseline TrOCR evaluation

Table 10 presents representative results for three license plate recognition experiments using TrOCR with default parameters. The engine's performance was evaluated using the CER, a metric that quantifies the proportion of incorrect characters relative to the total number of characters in the ground truth. This metric accounts for character substitutions, deletions, and insertions, providing a detailed method of recognition accuracy. A CER value of 0 indicates perfect character recognition, while 1 or greater indicates no recognized character at all.

Table 10. Example OCR outputs for three experiments

Video / Image	Plate number	TrOCR baseline output	CER
22.mp4 064.jpg	WVC-961-A	 MAS.961.1	**0.667**
16.mp4 146.jpg	WTT-269-A	 MIT.269.1	**0.556**
15.mp4 127.jpg	YGZ-112-A	 GTYGZ.112.4	**0.556**

The CER value obtained for TrOCR with default parameters on the complete dataset was 0.4605, and all videos were evaluated in about 6 minutes. A consistent pattern of optical character recognition (OCR) errors was identified, primarily characterized by the substitution of visually similar symbols. Frequently occurring misclassifications included the letter "O" being read as the number "0" and the letter "B" being interpreted as the number "8". These errors generally resulted from suboptimal lighting or awkward camera angles. Beyond substitutions, insertion and deletion errors were also observed, specifically when characters were partly hidden or distorted. This motivated us to fine-tune TrOCR to achieve a lower CER.

Even with these difficulties, TrOCR demonstrated acceptable performance in identifying number plates under some conditions, an advantage attributed to its attention mechanism and context-aware decoding. These findings underscore the promise of transformer-based OCR models for practical deployment in real-world ANPR systems, which often operate under unpredictable imaging conditions.

5.3 Re-trained TrOCR results

To ensure a consistent comparison, the same OCR evaluation procedure described in the previous section was applied to the fine-tuned TrOCR model. Specifically, the second stage of the ANPR pipeline was applied to the previously extracted regions of interest (RoIs) from detected license plates. This approach allowed measuring the improvements achieved through model fine-tuning against the pre-trained model.

Table 11 presents the results of a 3 license plate recognition experiments performed with both versions of the TrOCR model. We can see a significant improvement in text recognition by examining the CER score of the fine-tuned TrOCR model.

Table 11. Example fine-tuned TrOCR outputs for a single experiment

Video / Image	Plate number	Pre-trained TrOCR		Fine-tuned OCR	
		Output	CER	Output	CER
16.mp4 / 171.jpg	G47-AJH	COMPANY	0.857	G47.AJH	**0.143**
11.mp4 / 058.jpg	WVC-305-A	GVC.305.1	0.444	WVC.305.A	0.222
04.mp4 / 437.jpg	MVR-634-A	TOTAL..GSTRA	1.222	WYR.634.A	0.333

Figure 10 shows a comparison between relevant differences in plate number recognition between the base TrOCR model and the fine-tuned one. In these examples, we can see the near-accurate plate recognition of the retrained model, which mainly reads a dash character "-" as a period ".".

Pre-trained TrOCR **Fine-tuned TrOCR**

Plate number: **G47-AJH**

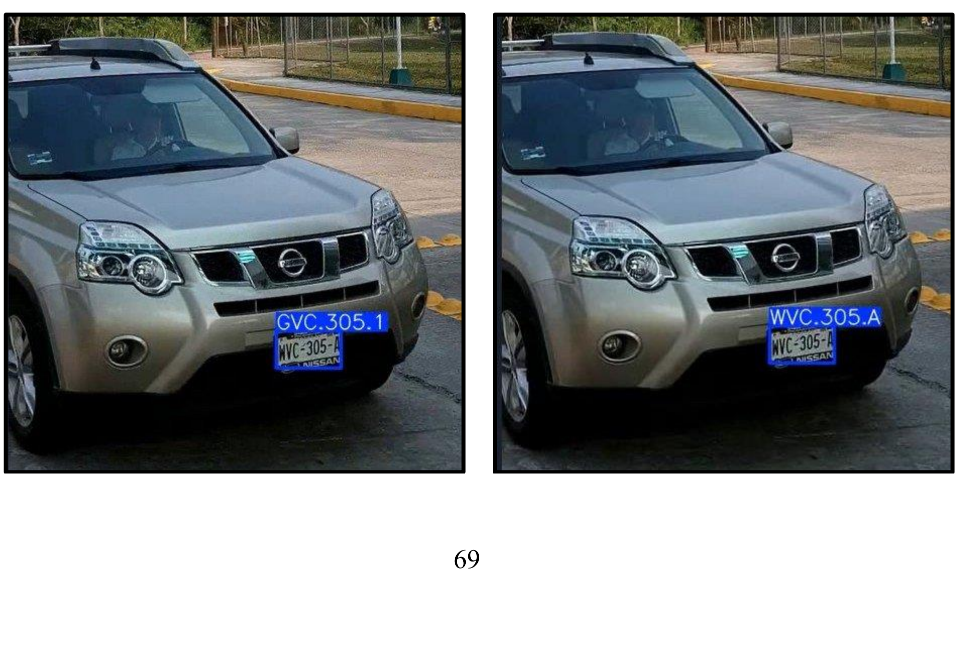


Plate number: **WVC-305-A**

Plate number: **MVR-634-A**

Fig. 10. Example of pre-trained TrOCR vs. fine-tuned TrOCR outputs.

The overall performance of the fine-tuned TrOCR model was evaluated using CER, following the same procedure as in the previous section for consistency. The baseline pre-trained TrOCR achieved a global CER of 0.4605, whereas the fine-tuned model reduced the CER to 0.2576. This improvement demonstrates a substantial increase in recognition accuracy due to specific adaptation. The CER values of both models are presented in Table 12.

Table 12. CER before and after fine-tuning

OCR model	CER
Default TrOCR	0.4605
Fine-tuned TrOCR	**0.2576**

The reduction in CER in the fine-tuned model was approximately 44%, highlighting the effectiveness of the fine-tuning procedure in improving character recognition under challenging conditions, including low contrast, motion blur, and partial occlusions. The fine-tuned TrOCR model not only reduced character-level errors but also exhibited greater robustness in recognizing visually ambiguous symbols. Runtime performance remained comparable to the pre-trained model. Figure 11 illustrates the gradual decrease in training loss during fine-tuning.

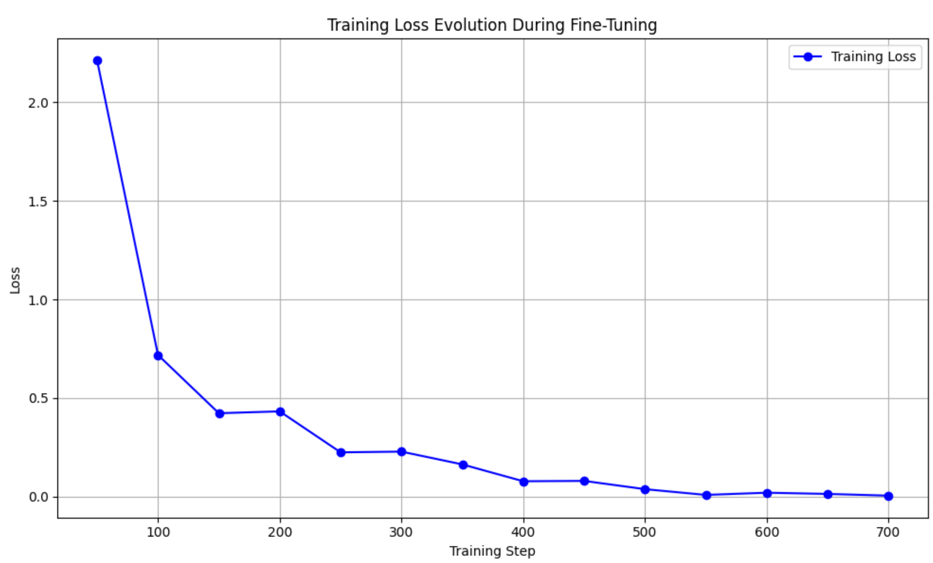


Fig. 11. Training loss curve.

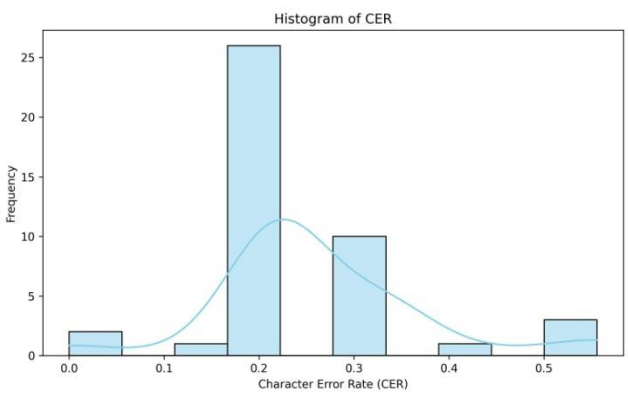
Despite the improvement after TrOCR retraining, low illumination, car movement, partial occlusions, and interchanged characters were the most common reasons for TrOCR misrecognition. Table 13 shows typical examples of OCR crashes under diverse conditions.

Table 13. Representative OCR failure examples

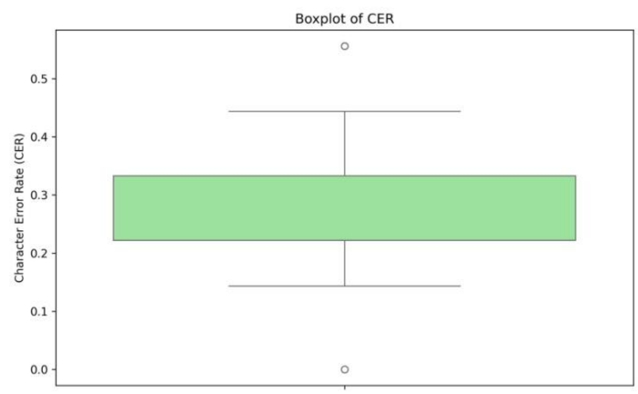
Poor condition	Video / Image	Plate number	TrOCR output	CER
Low illumination	04.mp4 / 023.jpg	MVR-634-A	WYR.634.A	0.333
Motion blur	03.mp4 / 011.jpg	A-050-VMC	A.080.C	0.556
Partial occlusion	19.mp4 / 135.jpg	UVT-6160-A	VT.6160.A	0.200
Confusion C / 0	13.mp4 / 078.jpg	DLU-574-C	DLU.574.0	0.333

Noticeably, in Experiment 19, the plate UVT-6160-A was partially misrecognized as VT.6160.A. The first character "U" was omitted, and the hyphen "-" was interpreted as a period ".". In Experiment 13, a typical OCR character confusion, "0" instead of "C", illustrates the challenges posed by visually similar symbols in license plate recognition. Additionally, common errors in single characters include interchanging "W" with "M", "O" with "0", and "8" with "B". These mistakes occur even under good conditions, such as high illumination and no occlusions.

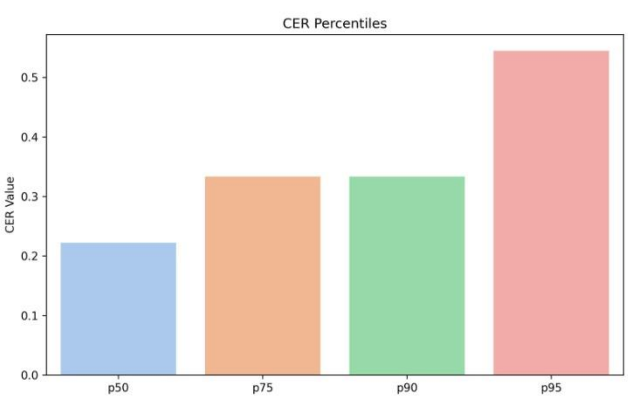
Useful results of the OCR performance analysis are illustrated in Figure 12, where the distribution of CER, boxplot representation, percentile values, and full-plate exact match rate are presented for the evaluated dataset. A median CER of 0.222 was observed for the fine-tuned TrOCR (Figure 12a). Percentile analysis revealed CER values of 0.222 (p50), 0.333 (p75), 0.333 (p90), and 0.556 (p95), indicating that most license plates were recognized with relatively low error rates, while a small subset of hard cases concentrated higher recognition errors (Figure 12b and 12c). Furthermore, a full-plate exact match rate of 4.65% was obtained (Figure 12d).



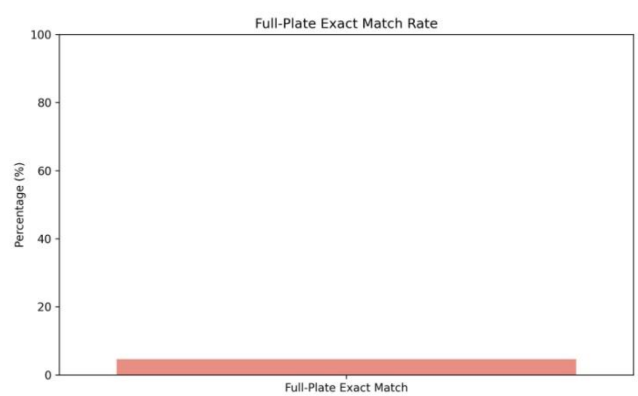
a) Histogram



b) Boxplot



c) Percentiles (p50, p75, p90, p95)



d) Full-Plate Exact Match Rate (%)

Fig. 12. Useful statistics of the TrOCR performance.

6 Conclusions

In this study, an end-to-end ANPR pipeline was developed using YOLOv11 + TrOCR for plate detection and evaluated using the standard metrics precision, recall, and character error rate (CER). YOLOv11 achieved high detection reliability (recall: 0.9950, precision: 0.9380), confirming its effectiveness under varied conditions. However, TrOCR with a default configuration reached moderate performance (CER: 0.4605). That is why we propose a fine-tuning strategy that creates a custom license plate dataset.

The benefits of fine-tuning were evident: the TrOCR model retrained on 651 license plate images produced more accurate character sequences than the pre-trained baseline, particularly in challenging conditions such as low lighting, motion blur, partial occlusions, and skewed viewing angles. The retrained TrOCR model reached a CER of 0.2576, representing a 44% improvement. In practice, the fine-tuned model enables more reliable license plate recognition in real-world ANPR applications in challenging conditions. Integrating fine-tuning demonstrates that transformer-based OCR can be effectively adapted for specific operational needs.

As future work, we propose expanding the dataset with additional plates and employing a group-based splitting strategy, ensuring that all instances of a given plate or vehicle are confined exclusively to either the training or the testing set. Also, incrementing the number of different plates would strengthen model validity. In model comparison, evaluating models on public datasets provides an objective way to assess and validate their performance. Finally, a tighter coupling between YOLOv11 and TrOCR would improve the model's speed and accuracy.

Acknowledgment

The authors would like to thank the SECIHTI (Ministry of Science in México) for supporting the Doctoral program in Computer Science at the Universidad Juárez Autónoma de Tabasco.

Use of AI tools

The authors declare that they used Grammarly Pro to edit and proofread the article in English. Only the tool's suggestions that were approved by all authors were included in the document.

References

Al-Hasan, T. M., Bonnefille, V., & Bensaali, F. (2024). Enhanced YOLOv8-based system for automatic number plate recognition. *Technologies, 12*(9), Article 164. https://doi.org/10.3390/technologies12090164

Anwar, N., Khan, T., & Mollah, A. F. (2022). Text detection from scene and born images: How good is Tesseract? In A. K. S. Pundir, N. Yadav, H. Sharma, & S. Das (Eds.), *Recent trends in communication and intelligent systems* (pp. 115–122). Springer. https://doi.org/10.1007/978-981-19-1324-2_13

Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., Oh, S. J., & Lee, H. (2019). What is wrong with scene text recognition model comparisons? Dataset and model analysis. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 4714–4722). IEEE. https://doi.org/10.1109/ICCV.2019.00481

Bakhtan, M. A. H., Abdullah, M., & Rahman, A. A. (2016). A review on license plate recognition system algorithms. In *2016 International Conference on Information and Communication Technology Management (ICICTM)* (pp. 84–89). IEEE. https://doi.org/10.1109/ICICTM.2016.7890782

Dhyani, S., & Kumar, V. (2023). Real-time license plate detection and recognition system using YOLOv7x and EasyOCR. In *2023 Global Conference on Information Technologies and Communications (GCITC)* (pp. 1–5). IEEE. https://doi.org/10.1109/GCITC60406.2023.10425814

Drobac, S., & Lindén, K. (2020). Optical character recognition with neural networks and post-correction with finite state methods. *International Journal on Document Analysis and Recognition, 23*, 279–295. https://doi.org/10.1007/s10032-020-00359-9

Google Cloud. (n.d.). *Colab Enterprise documentation*. Retrieved 31 July 2026, from https://docs.cloud.google.com/colab/docs

Jocher, G., & Qiu, J. (2024). *Ultralytics YOLO11* (Version 11.0.0) [Computer software]. GitHub. https://github.com/ultralytics/ultralytics

K, T. D., James, J., Gopinath, D. P., & K, M. A. (2024). *Advocating character error rate for multilingual ASR evaluation* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2410.07400

Laroca, R., Zanlorensi, L. A., Gonçalves, G. R., Todt, E., Schwartz, W. R., & Menotti, D. (2021). An efficient and layout-independent automatic license plate recognition system based on the YOLO detector. *IET Intelligent Transport Systems, 15*(4), 483–503. https://doi.org/10.1049/itr2.12030

Lauar, F., & Laurent, V. (2024). *Spanish TrOCR: Leveraging transfer learning for language adaptation* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2407.06950

Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., & Wei, F. (2023). TrOCR: Transformer-based optical character recognition with pre-trained models. *Proceedings of the AAAI Conference on Artificial Intelligence, 37*(11), 13094–13102. https://doi.org/10.1609/aaai.v37i11.26538

Menezes, D., Patel, S., Shaikh, A., & Parikh, N. (2025). Robust ANPR system using YOLOv8 and TrOCR with aspect-ratio-based splitting and post-processing. *International Journal for Research in Applied Science & Engineering Technology, 13*(9), 1911–1914. https://doi.org/10.22214/ijraset.2025.74265

Nguyen, D. D., Vo, T. S., Le, M. H., & Nguyen, M. S. (2025). An integration of segmentation technique on edge devices for license plate recognition. *Vietnam Journal of Science, Technology and Engineering, 67*(1), 3–13. https://doi.org/10.31276/VJSTE.2023.0099

Rather, I. H., Kumar, S., & Gandomi, A. H. (2024). Breaking the data barrier: A review of deep learning techniques for democratizing AI with small datasets. *Artificial Intelligence Review, 57*, Article 226. https://doi.org/10.1007/s10462-024-10859-3

Sarhan, A., Abdel-Rahem, R., Darwish, B., Abou-Attia, A., Sneed, A., Hatem, S., Badran, A., & Ramadan, M. (2024). Egyptian car plate recognition based on YOLOv8, Easy-OCR, and CNN. *Journal of Electrical Systems and Information Technology, 11*, Article 32. https://doi.org/10.1186/s43067-024-00156-y

Satya, B., Manongga, D., Hendry, & Aminuddin, A. (2025). Optimized YOLOv8 for automatic license plate recognition on resource-constrained devices. *Engineering, Technology & Applied Science Research, 15*(2), 21976–21981. https://doi.org/10.48084/etasr.9983

Shu, M. (2025). Utilizing transfer learning for deep learning based image classification. *Journal of Intelligence Technology and Innovation, 3*(1), 58–73. https://itip-submit.com/index.php/JITI/article/view/108

Sonnara, F., Chihaoui, H., & Filali, F. (2025). Efficient real-time license plate recognition using deep learning on edge devices. *Journal of Real-Time Image Processing, 22*, Article 159. https://doi.org/10.1007/s11554-025-01738-3

Sporici, D., Cușnir, E., & Boiangiu, C.-A. (2020). Improving the accuracy of Tesseract 4.0 OCR engine using convolution-based preprocessing. *Symmetry, 12*(5), Article 715. https://doi.org/10.3390/sym12050715

Spruck, A., Hawesch, M., Maier, A., Rieß, C., Seiler, J., & Kaup, A. (2021). 3D rendering framework for data augmentation in optical character recognition (Invited paper). In *2021 International Symposium on Signals, Circuits and Systems (ISSCS)* (pp. 1–4). IEEE. https://doi.org/10.1109/ISSCS52333.2021.9497438

Ströbel, P. B., Clematide, S., Volk, M., & Hodel, T. (2022). *Transformer-based HTR for historical documents* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2203.11008

Tao, L., Hong, S., Lin, Y., Chen, Y., He, P., & Tie, Z. (2024). A real-time license plate detection and recognition model in unconstrained scenarios. *Sensors, 24*(9), Article 2791. https://doi.org/10.3390/s24092791

Tavares, R. A. (2024). *Comparison of image preprocessing techniques for vehicle license plate recognition using OCR: Performance and accuracy evaluation* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2410.13622

Vizcarra, C., Alhamed, S., Algosaibi, A., Alnaeem, M., Aldalbahi, A., Aljaafari, N., Sawalmeh, A., Nazzal, M., Khreishah, A., Alhumam, A., & Anan, M. (2024). Deep learning adversarial attacks and defenses on license plate recognition system. *Cluster Computing, 27*, 11627–11644. https://doi.org/10.1007/s10586-024-04513-4

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2022). *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2207.026967:

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