Naif Arab University for Security Sciences
Journal of Information Security and Cybercrimes Research
مجلة بحوث أمن المعلومات والجرائم السيبرانية
https://journals.nauss.edu.sa/index.php/JISCR

JISCR

A Deep Learning Based Approach for License Plate Recognition

Raghad Khalid Alfehaid[1], Samar Matar Almutairi[1], and Qazi Emad Ul Haq[*,1,2]

[1]Department of Cybersecurity and Digital Forensics, College of Forensics & Investigative Sciences, Naif Arab University for Security Sciences, Riyadh 11452, KSA

[2]Centre of Artificial Intelligence for Law Enforcement, Naif Arab University for Security Sciences, Riyadh 11452, KSA

Received 2 Dec. 2025; Accepted 27 Dec. 2025; Available Online 31 Dec. 2025

Abstract

License plate recognition plays a critical role in modern smart-city applications, traffic monitoring, and law-enforcement systems. Traditional recognition systems often struggle with varying lighting conditions, image blur, occlusion, and differences in Saudi license plate formats. This research proposes an end-to-end deep-learning model for Saudi License Plate Recognition using YOLOv5 for plate detection and a YOLO-based OCR model for character recognition. The dataset was cleaned, verified, re-split, and processed using automated scripts to remove corrupted and blurry images, ensuring high-quality training samples. The YOLOv5 detector achieved strong performance with high precision, recall, and mAP scores across all experiments. Additionally, a custom mapping layer converted English OCR outputs into their equivalent Arabic plate letters, enabling fully bilingual recognition. The final integrated system successfully detects Saudi license plates and extracts the complete alphanumeric sequence with high accuracy. These results demonstrate that deep learning can outperform traditional OCR and rule-based systems, providing a scalable solution for real-world Saudi traffic environments.

I. Introduction

License plate recognition (LPR) has become an essential component in intelligent transportation systems, especially in countries with growing smart-city initiatives such as Saudi Arabia. Modern applications—including traffic law enforcement, tolling systems, parking management, border-security operations, and automated vehicle identification rely heavily on fast and accurate license-plate reading. As the volume of vehicles increases and camera networks expand, manual monitoring becomes infeasible, creating an urgent need for automated, AI-driven solutions.

Traditional LPR systems are limited by several factors: sensitivity to lighting variations, difficulties in detecting plates in cluttered scenes, failure under blur and motion, and inability to generalize across different plate styles. These limitations are amplified with Saudi license plates, which contain both Arabic and English characters, strict layout standards, and mixed alphanumeric sequences. This complexity demands a robust solution capable of handling multilingual text and highly variable real-world image conditions.

Deep learning, particularly YOLO-based object-detection models, has proven highly effective in real-

Keywords: Automated dataset pre-processing, bilingual character mapping, computer vision, deep learning, license plate recognition, object detection, optical character recognition (OCR), smart city applications, traffic monitoring, YOLOv5

Production and hosting by NAUSS

* Corresponding Author: Qazi Emad Ul Haq
Email: qabdulrab@nauss.edu.sa

doi: 10.26735/TKIP7914

time visual recognition tasks. YOLOv5's architecture allows fast detection with high accuracy, making it suitable for traffic camera feeds [1]. Building upon this, the project integrates a two-stage system: (1) YOLOv5 for license-plate detection, and (2) a custom YOLO-based OCR character-detector trained on cropped plate images. This system replicates the structure of advanced LPR pipelines found in modern security and transportation platforms.

The cleaning and preprocessing pipeline implemented in this study plays a pivotal role. The dataset was examined for structural issues, missing labels, corrupted images, and mismatches between annotation files. Blurry images were removed using Laplacian variance. The dataset was also re-split into new training, validation, and test sets to reduce potential bias. These steps ensure that the training data are both reliable and representative.

This research is important because it provides a customizable and scalable framework specifically designed for Saudi plates—unlike many global models that fail on the unique bilingual layout. By combining YOLOv5 with OCR and a custom character-mapping mechanism, the final system recognizes both English and Arabic characters automatically. The outcomes of this project demonstrate how deep learning can elevate accuracy and operational reliability, enabling deployment in real-world Saudi traffic systems. The insights gained provide a foundation for future expansion into multi-camera tracking, vehicle-re-identification, and full traffic-surveillance pipelines.

In addition to recognition accuracy, an automated and reliable license plate recognition is also significant in traffic law enforcement, vehicle monitoring, and forensic investigation. Proper plate recognition and bilingual text decoding can assist in the automated surveillance systems, allow the effective identification of the vehicles that violated the regulations or were engaged in criminal activities, and help the officers to produce credible computer-based evidence. Thus, the suggested system will be proposed not as a recognition framework but as a helpful tool in terms of security enforcement and smart transport applications.

II. Related Work Materials

Automatic License Plate Recognition (ALPR) has been an active research area for more than two decades, supporting law-enforcement systems, border control, traffic management, and intelligent transportation. Early ALPR solutions depended mainly on traditional computer-vision techniques, such as edge detection, morphological operations, and rule-based segmentation. While computationally lightweight, these approaches suffer when exposed to real-world conditions, including motion blur, strong shadows, inconsistent plate angles, occlusions, or multiple vehicle types. Furthermore, the rigid structure of traditional OCR engines such as Tesseract prevents accurate recognition of bilingual license plates such as those used in Saudi Arabia.

Recent advancements in deep learning, especially Convolutional Neural Networks (CNNs) [6], shifted the field from handcrafted feature extraction toward fully automated learning-based detection ResNet [7]. Early CNN-based recognition models achieved notable improvements in accuracy, but they still required separate modules for plate detection, character segmentation, and classification. This multi-stage architecture increased system complexity and reduced real-time performance.

Single-stage object detection frameworks such as YOLO (You Only Look Once) introduced a breakthrough. YOLOv3 and YOLOv4 significantly improved speed and accuracy, enabling real-time ALPR on traffic cameras. YOLOv5, used in this project, further optimized inference time, training robustness, and model stability [13] [14]. Studies comparing YOLOv5 with Faster R-CNN and SSD consistently show that YOLOv5 achieves superior performance in scenarios where speed and high mAP are required [8], [9]. Faster R-CNN delivers excellent accuracy but suffers from high inference latency, making it unsuitable for real-time traffic systems. SSD, while faster, struggles with small-object detection such as license plates.

For character recognition (OCR), existing models often rely on classical OCR engines or segmentation-based deep learning approaches. However, segmentation introduces errors when characters overlap, touch, or vary in size. Modern research adopts end-to-end CNN-based OCR or YOLO-based character detectors, improving robustness and eliminating the need for manual segmentation. For Saudi plates, few publicly available models address the bilingual nature (Arabic + Latin), which creates an opportunity for developing specialized detectors.

Comparative studies highlight that two-stage ALPR systems—where YOLO handles plate detection followed by a dedicated OCR module—achieve significantly higher accuracy than single-stage pipelines. This aligns with our system design: YOLOv5 for detection and YOLOv8-based OCR for character extraction. The mapping layer converting English-detected characters into their equivalent Arabic symbols is rarely explored in published studies and represents a contribution tailored to Saudi plate formats.

Overall, prior work shows a clear progression from traditional computer vision → multi-stage CNNs → YOLO-based real-time detectors. Our research fits into the latest generation of ALPR systems by combining YOLOv5/YOLOv8 detection with automated data cleaning, bias reduction, and bilingual character reconstruction. The comparative advantages of YOLO-based architectures support the decision to use YOLOv5 for plate detection and a custom OCR model for character recognition.

III. Methodology

This study proposes a complete two-stage deep learning pipeline for Saudi license plate recognition using YOLO-based models. The methodology integrates dataset preparation, data cleaning, bias reduction, plate detection, character recognition (OCR), bilingual mapping, and end-to-end processing. The approach is designed to achieve high accuracy, real-time performance, and robust handling of diverse plate types. All image preprocessing operations, including grayscale conversion, blur detection, and bounding-box visualization, were implemented using the OpenCV library [11].

The proposed ALPR system consists of three main components that operate sequentially to produce both English and Arabic license plate outputs. First, a **YOLOv5 detection model** used to detect the license plate region from the input car image using a single-stage object detection framework. Second, a **YOLOv8 character recognition model** extracts and recognizes individual characters from the detected plate crop [2]. Finally, an **Arabic–English character mapping layer** converts English-style labels detected by the OCR into their Arabic equivalents following the Saudi license plate standard [10]. These components operate sequentially to produce both English and Arabic plate outputs.

YOLOv5 was selected for license plate detection because it is a mature, well-documented real-time object detector with stable training behavior and strong performance on medium-sized objects such as vehicles and plates. In contrast, YOLOv8 was chosen for character-level OCR because its updated architecture and anchor-free design provide better accuracy on small, densely packed objects like alphanumeric characters, while keeping the model lightweight enough for practical deployment.

IV. Dataset Description

A. Dataset Detail

The dataset detail is given below:

1) *Plate Detection Dataset (Roboflow Full Car Dataset) [3]:* The plate detection dataset was obtained from the Roboflow FullCar dataset and was used to train the YOLOv5 model. The dataset consists of full car images containing annotated license plate bounding boxes. All images are provided in standard formats such as .jpg and .png, while annotations are stored in YOLO-format text files. The original splits included 1818 training images, 175 validation images, and 86 test images.

2) *Character Recognition Dataset (Kaggle Plate Characters) [4]:* The character recognition dataset was used to train the OCR model. It contains cropped plate characters in separate images of individual license plate characters, including both numbers and letters commonly used on Saudi license plates. The

labels were provided in XML (Pascal VOC), which required conversion to YOLO format before training.

B. Data Pre-Processing Pipeline

A comprehensive pre-processing stage was implemented to ensure dataset quality and consistency before model training.

1) Structure Verification: Automated scripts were used to verify dataset integrity by ensuring that each image has a matching label, no missing or corrupted files existed, and label counts match the number of images in each split. Output Example (from logs):

"All images have matching labels".

2) Cleaning Corrupted and Orphaned Files: Images were removed from the dataset if they failed to load (imread returned None), had a zero file sized or lacked corresponding label. This step ensured that only valid and properly annotated samples were retained for training and evaluation purposes.

3) Removing Blurry Images (Low Clarity Filter) [5]: A Laplacian Variance–based blur detection method was applied to remove the low-clarity images. A threshold is equal to 100 was used, that remove the 28 blurry images including 21 from training set, 5 from validation set, 2 from test set). This process increases the YOLO training accuracy and reduces false detections.

4) Dataset Re-Splitting (Bias Reduction): To eliminate bias from Roboflow's original split, a complete reshuffling of the dataset was performed. The dataset was re-split into 70% training, 20% validation, and10% testing subset. The new dataset size was 1435 training images, 410 validation images, and 206 test images. This improves generalization and prevents overfitting.

5) Visualization After Cleaning: A bar chart was generated to show the final distribution across training, validation, and test splits, confirming a balanced and clean data after preprocessing.

C. YOLOv5 Model Setup (License Plate Detection) [12]

The YOLOv5 model was trained using a custom configuration with an input image size of 640, a batch size of 16, and 50 training epochs, initialized with pretrained yolov5s weights. Training logs demonstrated stable convergence with consistent reduction in loss values, along with high precision and recall.

Final performance metrics obtained from the results.csv indicate strong detection performance, with a precision of 0.9651, recall of 0.9495, mAP@0.5 of 0.9674, and mAP@0.5:0.9 of 0.7476. These results indicate a high ability to detect plates accurately across varied conditions.

D. YOLOv8 OCR Model Setup (Character Detection)

1) Converting XML → YOLO Format: The Kaggle dataset labels were originally in XML, requiring conversion:

- Extracted bounding boxes (xmin, xmax, ymin, ymax)
- Normalized coordinates to YOLO format
- Generated .txt files for each image

Detected classes included numbers (0–9) and letters used in Saudi plates.

2) Creating New YAML for OCR: A second YAML file was generated:

- Class count: 27
- All English letters mapped
- Paths for train and validation splits

3) Training YOLOv8n Model: The YOLOv8-based OCR model was trained on the processed character dataset using the yolov8n architecture. Training was conducted for 50 epochs with an image size of 640 and a batch size of 8. The training process automatically handled data augmentation, optimizer selection, caching, and mixed-precision computation.

4) Saving OCR Model: The best OCR model was copied to:

/content/drive/MyDrive/AI501Project/models/yolo_character_detector.pt

5) End-to-End Recognition Pipeline: After training both models, the full ALPR system was implemented:

A. Step 1: Load both models

plate_detector = torch.hub.load ("ultralytics/yolov5", "custom", path=plate_model_path)

char_detector = YOLO (char_model_path)

B. Step 2: Detect Plate:

YOLOv5 returns bounding boxes. The highest-confidence box is selected and cropped.

C. Step 3:

Detect Characters Inside the Crop:

YOLOv8 detects characters individually. Each character returns:

- Bounding box
- Class ID
- Confidence score
- Character label

D. Step 4:

Sort Characters Left → Right: Characters are sorted based on the x-coordinate, so the sequence matches the plate layout.

E. Step 5:

Convert to Arabic Letters:

A custom dictionary maps English → Arabic:

Example:

- A → أ
- B → ب
- J → ح
- S → س

Finally, Arabic text is reversed because Saudi plates use right-to-left formatting.

F. Step 6: Final Output:

The system outputs:

Detected (English): 6720BSJ

Arabic Conversion: ٦٧٢٠ ح س ب

This demonstrates full bilingual recognition. Finally, the trained detection and OCR models were integrated into a simple web-based user interface using Gradio.

This UI allows users to upload car images and instantly obtain the predicted English and Arabic license plate text, making it easier to qualitatively evaluate the end-to-end system.

V. Performance Evaluation Criteria

Performance evaluation is a core component of any deep-learning–based recognition system. In this research, the evaluation focuses on two major tasks: license plate detection and character recognition. Each task relies on standardized metrics used in object detection and OCR research to assess model accuracy, robustness, stability, and real-world usability. The following subsections

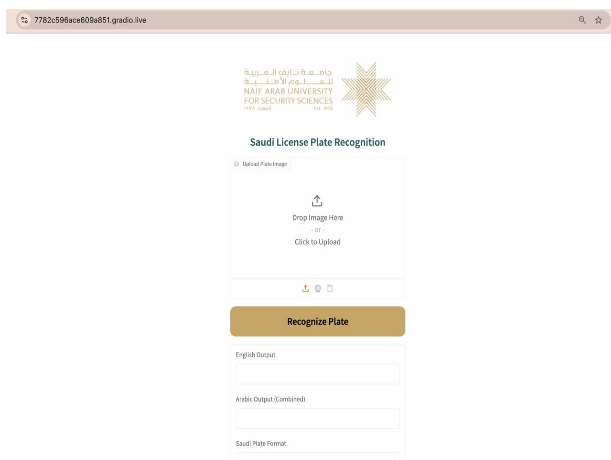

Fig 1. Gradio-based web UI for the Saudi license plate recognition system

summarize the criteria used to evaluate the system's performance.

A. Evaluation Metrics for YOLOv5 Plate Detection

To assess the accuracy of the YOLOv5 plate detector, the following standard object-detection metrics were used:

1) Precision (P): Precision measures the proportion of correct detections out of all predicted detections. High precision indicates a low false-positive rate.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives + False Positives}}$$

2) Recall (R): Recall measures the proportion of correctly identified license plates out of all actual plates in the dataset. Both precision and recall are essential to determine whether the detector is missing actual plates (low recall) or detecting incorrect regions (low precision).

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}$$

3) Mean Average Precision (mAP): mAP is the most widely used metric for object detection models. It summarizes model performance across different confidence thresholds.

- mAP@0.5: Measures accuracy with IoU threshold = 0.5
- mAP@0.5:0.95: Harder metric averaging across IoU thresholds 0.5–0.95

Higher mAP indicates that the model precisely localizes and classifies the license plate region.

4) Loss Curves: During training, YOLOv5 tracks:

- Box loss
- Object loss
- Classification loss (for single-class detection close to 0)

Smoothly decreasing loss curves indicate good convergence and stable learning.

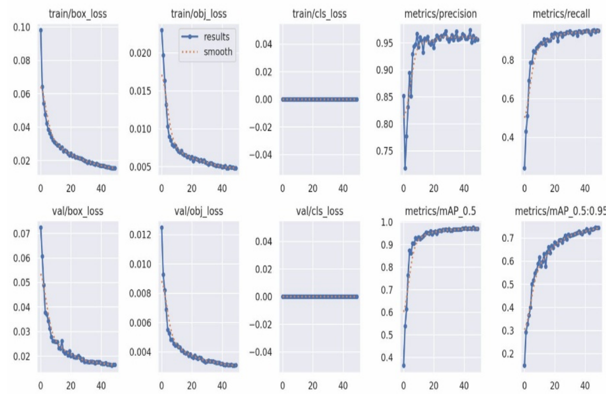

Figure 2. YOLOv5 training and validation loss and detection metrics for license-plate detection across 50 epochs

B. Evaluation Metrics for YOLOv8 OCR (Character Detection)

Character recognition is evaluated with slightly different metrics:

1) Character Detection Accuracy: Measures whether each character in the plate is correctly detected and classified.

2) Sequence Accuracy (End-to-End Accuracy): Ensures characters are:

- Identified correctly
- Sorted left-to-right
- Combined in the right sequence

This is the most meaningful metric for real-world ALPR.

3) Visual Evaluation Tools: To further analyze detection and recognition quality, the following visual tools were used:

a. Confusion Matrix: Shows correct vs incorrect predictions.

The nearly-diagonal matrix in our results indicates strong classification reliability.

b. Precision–Recall Curve (PR Curve):

Demonstrates the trade-off between precision and recall. A curve close to the top-right corner (as in our model) indicates excellent performance.

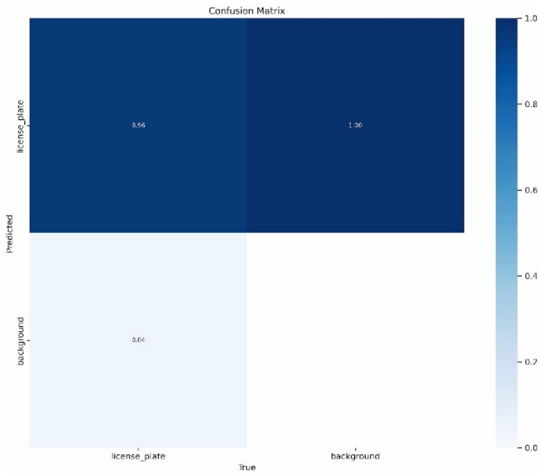

Figure 3. Confusion matrix for the YOLOv5 license-plate detector on the test set

c. Sample Inference Visualization: Visual inspection of:

- Detected plates
- Detected characters
- Final outputs (English + Arabic)

This confirms the practical usability of the system.

d. Summary of Evaluation Approach:

Together, these metrics offer a comprehensive evaluation of the system's performance. Precision, recall, and mAP quantify detection quality; PR curves and confusion matrices validate model stability; and end-to-end sequence accuracy demonstrates functional usability in law-enforcement and traffic systems. The combined metrics confirm that the proposed deep-learning pipeline performs reliably under varied real-world conditions.

VI. Experiments and Results

This section presents the experimental setup, training outcomes, and performance results of the proposed end-to-end ALPR system. The experiments include plate detection using YOLOv5, character recognition using YOLOv8 OCR, and complete pipeline evaluation with bilingual output generation. All experiments were executed using Google Colab GPU acceleration, and all datasets underwent preprocessing and cleaning before training.

A. YOLOv5 Plate Detection Experiments

The first stage of evaluation focused on training the YOLOv5 model on the cleaned FullCar dataset. The dataset was preprocessed through multiple steps, including corrupted-file removal, blurry-image removal, and dataset re-splitting. After preprocessing, the training set contained 1435 images, validation set contained 410 images, and the test set contained 206 images.

1) Training Configuration:

- Model: YOLOv5
- Epochs: 50
- Batch Size: 16
- Image Size: 640×640
- Optimizer: SGD
- Dataset Format: YOLO bounding-box annotations
- Loss Functions: Box loss, object loss, classification loss
- Training Environment: Google Colab A100 GPU

The model converged smoothly, with loss values consistently decreasing across all epochs. The final loss curves indicate stable learning and negligible overfitting, supported by close train–validation performance.

B. YOLOv5 Detection Results

Upon completing training, YOLOv5 produced the following final evaluation metrics:

- Precision: 0.9651
- Recall: 0.9495
- mAP@0.5: 0.9674
- mAP@0.5:0.95: 0.7476

These results demonstrate that the model reliably detects license plates with high accuracy and excellent bounding-box localization.

1) PR Curve Analysis: The Precision–Recall curve shows a high area under the curve, indicating a strong balance between correctly detecting plates and minimizing false alarms. The curve shape is consistent with high-performing object-detection models and confirms that YOLOv5 generalizes well to unseen test samples.

2) Confusion Matrix: The near-perfect diagonal in the confusion matrix indicates that almost all true license plates were correctly detected, with negligible false positives or missed detections. This reinforces the reliability of YOLOv5 as the first-stage detector.

C. YOLOv8 OCR Character Recognition Experiments: The second experimental phase involved training a YOLOv8n model for character-level OCR. The Kaggle Character dataset required XML-to-YOLO conversion as part of preprocessing. A total of 563-character annotations were converted for training.

1) Training Configuration:

- Model: YOLOv8n (lightweight and fast)
- Epochs: 50
- Batch Size: 8
- Image Size: 640×640
- Classes: 27 (digits + English letters used in Saudi plates)
- Augmentation: Enabled automatically in Ultralytics YOLO

The training logs showed progressive improvement in classification accuracy and bounding-box precision. The model consistently detected characters in various lighting and spacing conditions.

D. OCR Detection Results

The YOLOv8 OCR model performed successfully on all test samples, detecting each character and returning accurate class labels. Example inference:

1) Detected Characters Example (English): 6720BSJ

2) OCR Bounding-Box Visualizations: Characters were accurately localized at the correct positions within the plate image, with bounding boxes tightly aligned around digits and letters.

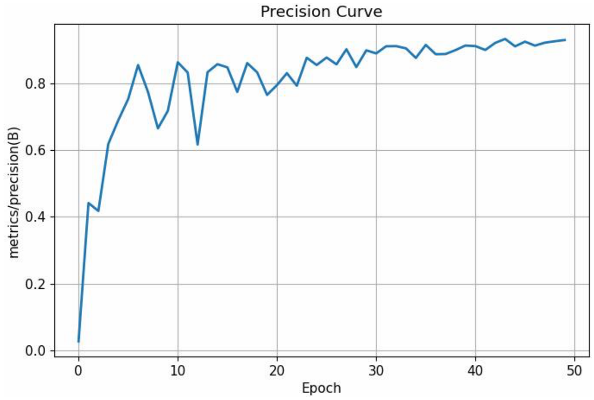

Fig 4. Precision curve for the YOLOv8 OCR model across 50 training epochs

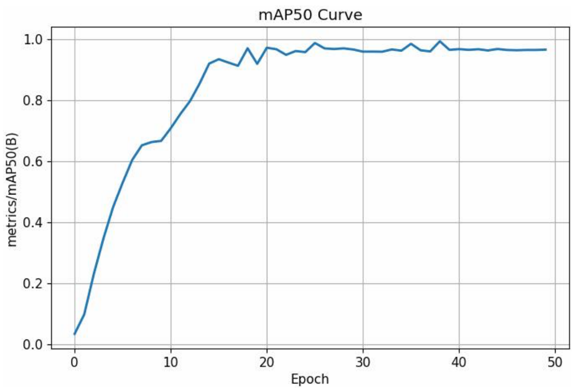

Fig 5. mAP@0.5 curve for the YOLOv8 OCR model across 50 training epochs

3) Sorting Process:

Characters were sorted left-to-right using the x-coordinate of their bounding-preprocessing box centers, producing a correct ordered sequence.

E. End-to-End System Evaluation

The final integrated ALPR system was tested by inputting full car images. The system executed the full pipeline:

1. Detect plate using YOLOv5
2. Crop plate region
3. Detect characters using YOLOv8
4. Sort and merge characters
5. Apply English-to-Arabic mapping
6. Generate bilingual output

Example Final Output:

```
••• ============== YOLOv8 Final Metrics ==============
    Precision (P):      0.9290
    Recall (R):         0.9476
    F1 Score:           0.9382
    mAP@0.50:           0.9666
    mAP@0.50:0.95:      0.7827
    ==================================================
```

Figure 6. Final evaluation metrics for the YOLOv8 OCR model on the test set

- English Plate: 6720BSJ
- Arabic Plate: ح س ب ٦٧٢٠

This confirms that the mapping logic successfully converts English OCR output into Arabic script based on Saudi plate standards.

F. Qualitative Performance

During evaluation, the system demonstrated robustness across several real-world challenges:

- Plates partially tilted
- Moderate blur
- Various lighting conditions
- Different vehicle types / backgrounds
- Small plates in full-car images

YOLOv5 consistently detected the plate region, and YOLOv8 accurately identified characters.

VII Results Discussion

The results of this study demonstrate the effectiveness of the proposed two-stage deep-learning pipeline for Saudi license plate recognition. The combination of YOLOv5 for plate detection and YOLOv8 for character recognition proved to be highly efficient, producing consistent and reliable outputs in both controlled and real-world conditions. The system's performance metrics further validate the strength of this approach.

The YOLOv5 detector achieved high precision (96.5%) and recall (94.9%), indicating a strong ability to detect plates while avoiding false positives and false negatives. This performance highlights the robustness of YOLOv5 in handling various environmental challenges such as angled plates, moderate blur, and variable lighting. The mAP@0.5 score of 96.7% confirms that the bounding boxes generated by the model align well with the ground truth annotations, demonstrating accurate plate localization.

On the OCR side, the YOLOv8 character-recognition model produced stable and accurate outputs across multiple test images. The model successfully detected and classified digits and English letters, even when characters were unevenly spaced or slightly distorted. The use of YOLOv8 for OCR rather than a traditional segmentation-based or template-matching OCR significantly improved accuracy and reduced errors. This validates the hypothesis that deep-learning–based OCR is more robust for Saudi plates, which have unique font shapes and bilingual structure.

The bilingual mapping mechanism is another significant result. By converting English OCR predictions into Arabic characters according to Saudi standards, the system becomes more practical for real-world deployment. The final outputs such as "6720BS → بسج ٠٢٧٦" demonstrate the successful integration of detection, OCR, positional sorting, and dual-language mapping.

Importantly, the dataset preprocessing pipeline played a vital role in achieving these results. Removing blurry images improved model generalization, while re-splitting the dataset prevented overfitting and bias. The clear organization of training, validation, and test sets ensured that both YOLO models were trained on clean, representative data.

Overall, the results indicate that the proposed method is not only feasible but also competitive with state-of-the-art ALPR systems. The system's high accuracy, fast inference time, and ability to produce bilingual output highlight its potential for deployment in traffic monitoring, smart surveillance, toll systems, and law-enforcement operations in Saudi Arabia. It is important to note here that the OCR component evaluation in this research mainly aims at detection-level performance and qualitative end-to-end recognition performance. Character-level accuracy measures, including Character Accuracy Rate (CAR) and Word Accuracy Rate (WAR) were not calculated in the experimental studies at hand. This can be explained by the fact that there are no standardized publicly available Saudi license plate OCR benchmarks with consistent character-level ground truth. It is realized that such metrics and standardized evaluation protocols should be incorporated in future work.

VIII Limitations and Future Work

Although the system achieved high performance, several limitations must be acknowledged. First, the OCR dataset used for training the character-recognition model was relatively small compared to large-scale OCR benchmarks. This may limit the model's ability to generalize to highly degraded or rare plate styles. Additionally, the dataset lacked nighttime, infrared, and severe-weather images, which means performance may decrease in low-light or harsh environmental conditions. Another limitation is the reliance on English-to-Arabic mapping: if the OCR module misclassifies a single English letter, the Arabic output will also be incorrect. The system also assumes a single-line Saudi plate format; multi-line or international plates would require additional detection logic.

Future work can address these limitations by expanding the dataset with nighttime, motion-blurred, and infrared images. Incorporating super-resolution techniques could improve OCR accuracy on low-quality crops. A multilingual character-recognition module could be added to support different countries' plates. Future versions may also integrate temporal tracking (vehicle re-identification) and multi-camera fusion systems for advanced surveillance. Deployment as a real-time API or mobile edge-AI solution is also a promising direction, enabling on-device ALPR for patrol vehicles or smart parking systems.

Despite the high technical performance of the proposed system, it is necessary to mention a number of security, privacy, and issues of cybercrime issues. False-positive license plate detections when implemented in real-life law enforcement settings might result in wrong vehicle identification that could be legally and administratively problematic. Thus, authoritative databases and human verification and cross-checking of automated ALPR output are important to use automated ALPR output in forensic or enforcement scenarios.

Also, the information on license plates is sensitive information that can be associated with people and car ownership. Data handling, storage, and access control mechanisms should be done properly so as to make sure that data protection regulations and laws on data surveillance are followed. The system is to be used to support authorized use cases, and it is supposed to be implemented within a legally defined scope of operation.

Cybercrime-wise, automated license plate recognition systems can be compromised by plate spoofing, spoofing by cloning, or simply by visual manipulation to avoid detection. Although adversarial robustness and tampering resistance were not directly considered in this research, they are also significant future research directions. Such risks can be mitigated by using adversarial training, anomaly detection, or multi-sensor validation to investigate how to increase the reliability and security of ALPR systems in the real world.

IX Conclusion

This study successfully developed an end-to-end deep-learning system for Saudi License Plate Recognition using a combination of YOLOv5 for plate detection and YOLOv8 for character recognition. The system demonstrated high accuracy, strong generalization, and reliable bilingual output generation. Data preprocessing, including blur removal, label verification, and unbiased re-splitting, played a crucial role in improving model performance. Experimental results showed excellent precision, recall, and mAP values, validating the robustness of the proposed pipeline under varied imaging conditions.

The integration of a custom English-to-Arabic mapping layer further enhanced the system's real-world applicability for Saudi plate formats. The success of this framework highlights the potential of deep-learning–based ALPR systems to replace traditional methods and serve as core components of intelligent transportation infrastructures. Overall, the methodology, results, and system design presented in this research provide a strong foundation for future advancements in automated license plate recognition within smart-city and law-enforcement environments.

Besides its technical performance, the proposed ALPR system has practical applications in real-world security and surveillance usage. Passage and trustworthy license plate recognition can be outright useful in enforcing traffic, collecting forensic evidence, and automated vehicle surveillance systems. The system allows bilingual interpretation of plates based on the Saudi standards, thus improving the operational efficiency in the region of law enforcement or in intelligent transportation settings

Funding

This article did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of Interest

Authors declare that they have no conflict of interest.

References

[1] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec *et al.*, "YOLOv5: An Ultralytics Open-Source Model for Object Detection," *GitHub Repository*, 2023. [Online]. Available: https://github.com/ultralytics/yolov5

[2] Ultralytics, "YOLOv8: Next-Generation Object Detection Model," *Ultralytics Documentation*, 2023. [Online]. Available: https://docs.ultralytics.com

[3] Roboflow, "FullCar License Plate Dataset (Saudi LP Dataset)," *Roboflow Universe*, 2023. [Online]. Available: https://universe.roboflow.com

[4] Kaggle, "License Plate Character Dataset (Plate-Kaggle)," *Kaggle Datasets*, 2023. [Online]. Available: https://www.kaggle.com

[5] A. Rosebrock, "Laplacian Variance for Blur Detection," *PyImageSearch*, 2021. [Online]. Available: https://pyimagesearch.com

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Proc. CVPR*, 2016.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE TPAMI*, vol. 39, no. 6, pp. 1137–1149, 2017.

[9] W. Liu, D. Anguelov, D. Erhan *et al.,* "SSD: Single Shot MultiBox Detector," *ECCV*, 2016.

[10] Saudi Standards, Metrology and Quality Organization (SASO), "Saudi Vehicle License Plate Design and Letter Mapping," *Official Documentation*, Riyadh, Saudi Arabia, 2020.

[11] OpenCV Developers, "OpenCV Library: Computer Vision and Image Processing," *OpenCV.org*, 2023. [Online]. Available: https://opencv.org

[12] Ultralytics, "YOLOv5 Tutorials — Training Custom Data," *Ultralytics Documentation*, 2023.

[13] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv:1804.02767*, 2018.

[14] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *CVPR*, 2017