Volume 11 Issue 1 Year 2026 | Page 441-451 ISSN: 2527-9866
Received: ***11-02-2026*** | *Revised:* ***20-02-2026***| *Accepted:* ***28-02-2026***

Design of an Intelligent Vehicle Manifest Recording System at the Bengkalis - Sungai Pakning Ro-Ro Ferry Crossing Based on Deep Learning and Optical Character Recognition

Jaroji[1], Danuri[2], Agus Tedyyana[3]
[1, 2, 3] Bengkalis State Polytechnic, Riau, Indonesia, 28761
e-mail: jaroji@polbeng.ac.id[1], danuri@polbeng.ac.id[2], agustedyyana@polbeng.ac.id[3]
*Correspondence: jaroji@polbeng.ac.id

Abstract: *Vehicle manifest recording in Ro-Ro ferry services is still predominantly conducted manually, which may lead to operational inefficiencies and data inconsistencies. This study presents an automated vehicle manifest recording system for the Bengkalis–Sungai Pakning Ro-Ro ferry crossing by leveraging deep learning and Optical Character Recognition (OCR) technologies. The proposed system utilizes CCTV or IP cameras to capture vehicle images, performs frame extraction from video streams, and applies YOLOv11 for real-time vehicle and license plate detection. The detected license plate regions are subsequently processed using an OCR module to extract textual vehicle identification information. The detection model was trained using a publicly available Vehicle and License Plate Dataset. Experimental evaluation on the Vehicle and License Plate dataset shows that the YOLOv11 model achieves a precision of 85.9%, recall of 84.0%, and mAP@0.5 of 87.8% for vehicle and plate detection. OCR evaluation conducted on real operational test images indicates a recognition success rate of 57.5%, with an average confidence score of 0.63 for successfully recognized plates. Further analysis reveals that illumination level and plate scale (distance proxy) are the dominant factors affecting OCR performance, while tilt angle exhibits moderate influence. These results indicate that the proposed framework provides reliable visual detection performance and identifies critical environmental constraints that must be addressed for robust automated manifest deployment in Ro-Ro ferry environments.*

Keywords*: Ro-Ro ferry; vehicle manifest; YOLOv11; license plate recognition; Optical Character Recognition (OCR).*

1. Introduction

Ro-Ro (Roll-on/Roll-off) ports are a crucial component of multimodal logistics and transportation systems, facilitating the efficient movement of vehicles and goods between islands or sea routes. Operationally, the process of recording incoming and outgoing vehicle manifests is often still carried out manually by port officials, which is prone to recording errors, delays in data updates, and high labor consumption. This reliance on manual recording not only slows down vehicle flow but can also lead to inaccurate manifest data, impacting operational planning and optimal logistics reporting. In recent years, advances in artificial intelligence (AI) technology, particularly in the field of computer vision, have opened up significant opportunities for automating various administrative processes previously performed by humans. Deep learning-based object detection models such as YOLO (You Only Look Once) have gained popularity due to their ability to recognize and detect objects in real-time images with high accuracy and efficient inference speed, making them strong candidates for visual automation applications in various transportation and vehicle monitoring domains [1].

The application of Optical Character Recognition (OCR) technology is a key step in processing text information from images, such as reading license plate characters or other visual documents. Implementing OCR in an automation system can convert character images into text data that can be digitally processed for recording or indexing in a database. Studies of OCR use in various real-world applications, such as automated parking systems and vehicle identification systems, have shown that OCR can improve the efficiency of text extraction from images with relatively high accuracy, especially when combined with image processing approaches prior to OCR [2].

The combination of YOLO-based object detection and OCR has been tested in various application contexts, such as vehicle license plate detection and classification, with satisfactory results, proving that the integration of these two technologies can produce an automated vehicle detection and registration system. Research in university and smart city settings has demonstrated higher efficiency compared to manual data entry, as well as potential integration with security systems or other area monitoring [3]. However, while numerous studies highlight vehicle object detection and license plate text extraction, there remains a gap in the application of these technologies to the full operational context of Ro-Ro ports, particularly for multi-type vehicle manifest recording (trucks, buses, cars) and the automation process integrated with port management systems. Automating manifest recording at Ro-Ro ports requires not only accurate visual detection but also real-time data synchronization within logistics and customs systems, a challenge not widely addressed in the current literature.

Furthermore, operational challenges in port environments, such as changing lighting, varying camera angles, and the complexity of large numbers of moving vehicles, can impact the accuracy of visual detection systems. This necessitates the design of a system that is not only robust in detection and text recognition but also adaptable to real-world port conditions and optimized for real-time processing to support dynamic, large-scale operations. Therefore, this study discusses the design of an intelligent system for automatic Ro-Ro manifest recording, which combines a deep learning algorithm for vehicle detection and an OCR module for automatic text character extraction. With this approach, vehicle manifest data is expected to be accumulated quickly, accurately, and efficiently without relying on manual recording, contributing to increased port operational productivity and reduced administrative errors.

2. Literature Review

Research on Automatic License Plate Recognition (ALPR) has grown rapidly in line with the growing need for intelligent transportation systems and vehicle data automation in various applications such as automated parking, traffic monitoring, and access control systems. ALPR systems generally take a two-stage approach: vehicle license plate object detection and text extraction using Optical Character Recognition (OCR). Several studies have demonstrated the effectiveness of this approach in various application contexts. For example, Sunarta et al. [4] developed an ALPR system by combining YOLOv5 and Google Vision OCR to automatically recognize Indonesian vehicle license plates with high accuracy and real-time capabilities that meet the needs of real-world applications.

Various variants of the YOLO algorithm have also been explored to improve the accuracy and efficiency of ALPR systems. Optimizing YOLOv8 for resource-constrained devices, for example, demonstrated that the model can achieve high mean average precision (mAP) while maintaining real-time processing performance, even under low-light conditions or visual disturbances, making it suitable for use in intelligent transportation applications [5]. Furthermore, a model combining YOLOv10n and YOLOv11n was also evaluated in the context of Indonesian license plate detection integrated with EasyOCR, highlighting that the choice of model variant has a significant impact on precision, recall, and inference speed metrics [6].

In the context of text character extraction, research emphasizing a variety of image preprocessing techniques to improve OCR performance demonstrates the importance of the image processing stage prior to text extraction. Studies evaluating techniques such as grayscale conversion, bilateral filtering, and adaptive thresholding on the effectiveness of OCR for Indonesian license plate recognition emphasize that image quality has a significant impact on the final character recognition results [7]. Furthermore, the integration of a CNN approach for character classification after object detection has been shown to provide better text reading results than simple preprocessing methods on a real-world dataset [8].

In addition to YOLO and OCR-based approaches, other studies have explored end-to-end architectures with more complex combinations of character detection and recognition technologies. For example, an international study integrating YOLOv12 and PaddleOCR was able to recognize not only license plate characters but also vehicle type and other information such as registration area, adding a functional dimension to traditional ALPR systems [9]. Another article from the Journal of Real-Time Image Processing introduced the Light-Edge model as a single network that efficiently detects license plates and performs character recognition on edge devices, demonstrating the path forward for more efficient systems for real-world deployment [10].

Other research extends the application of YOLO and OCR in the Indonesian context with local datasets and real-world conditions. For example, Automated Indonesian Plate Recognition, which combines YOLOv8 and TensorFlow-CNN for character classification on Indonesian vehicle license plates, demonstrates the integration of multiple deep learning methods to handle local image variations [11]. Another study using YOLOv11 and PaddleOCR demonstrated that this combination can achieve a high mAP50 for license plate recognition along with additional information such as the license plate's origin, explaining the increasing complexity of ALPR systems for real-world use [12].

A similar approach was developed by [13], which leveraged YOLOv8 and EasyOCR to improve license plate recognition accuracy on images with high visual complexity. The results showed that the use of the latest generation YOLO model significantly improved detection performance, especially in live vehicle monitoring scenarios. However, this research is still limited to the context of license plate recognition systems and has not been directed at recording vehicle data as part of a manifest system or ferry services. Furthermore, [14] proposed a YOLOv11-based vehicle plate detection system that focuses on improving precision and recall in detecting vehicle plate areas. This study confirms that modern object detection algorithms are very effective for real-time visual processing of vehicles and have great potential for large-scale applications. Another study by [15] combines YOLOv8 and OCR in a license plate detection system that also counts the number of vehicles. This study highlights the importance of integrating vehicle visual data to support ITS applications such as parking management and traffic monitoring. Although successfully demonstrating the potential of computer vision-based automation, this study has not yet accommodated the needs of vehicle manifest recording in maritime transportation modes, especially in ferry services.

3. Methods

The system architecture is designed to support the automatic and real-time vehicle manifest recording process for the Bengkalis-Sungai Pakning Ro-Ro ferry service and vice versa, utilizing deep learning-based computer vision technology and Optical Character Recognition (OCR). This system aims to replace the manual recording process, which currently relies on officer observation, with an automated approach that is more accurate, consistent, and integrated with the port's operational systems. Figure 1 visualizes the system architecture design, which consists of five layers.

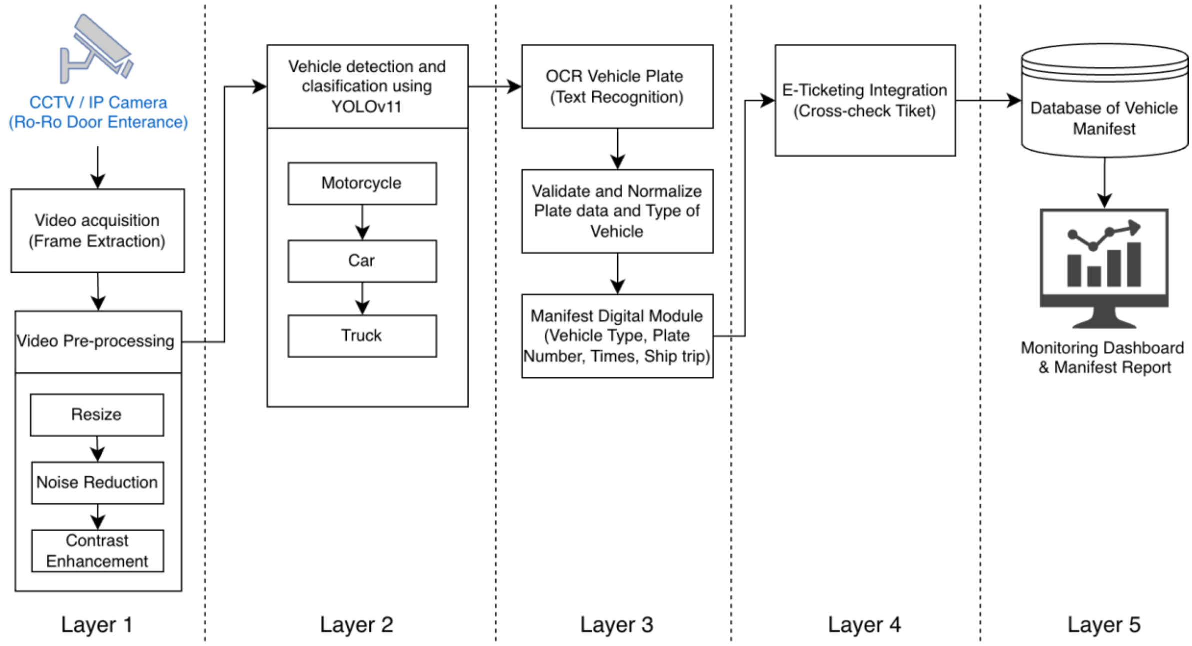

Figure 1. Intelligent System Architecture for Recording Ro-Ro Vehicle Manifests for the Bengkalis – Pakning River and Vice Versa Ferry Ferry

In layer 1 (data acquisition layer), the system utilizes CCTV or IP cameras installed at vehicle entry gates to Ro-Ro vessels and at the destination port during disembarkation. The cameras continuously record video streams of passing vehicles from optimal viewing angles to ensure visibility of motorcycles, passenger cars, and heavy vehicles. The acquired visual data is transmitted via the local port network to the processing module. Once a vehicle passes the observation area, the system automatically extracts frames and performs image pre-processing, including resizing, contrast enhancement, and noise reduction.

Next, in the visual processing layer (layer 2), vehicle images are processed using a YOLOv11-based object detection model to identify vehicle presence and classify their type (e.g., motorcycle, private car, truck, or bus) [17]. This stage serves as the basis for registering manifests based on the category of vehicle being transported. The vehicle detection results, in the form of a bounding box, object class, and detection time, are then forwarded to the text recognition module.

For detected vehicles, the system proceeds to the license plate area detection stage (layer 3) and performs character recognition using the OCR method. The OCR results are then validated against valid vehicle license plate patterns. If the license plate reading fails, the system flags the data for further inspection by officers. Valid vehicle data includes vehicle type, license plate number, and entry time—which are then recorded in the digital manifest module. The system then integrates with e-ticketing (layer 4) to ensure a match between the visually detected vehicle and the registered ticket data. If the data matches, the system continues to layer 5. Vehicle information is stored in the vehicle manifest database and displayed on the monitoring dashboard in real time. If a discrepancy is found, the system flags it for operational evaluation purposes. This process is repeated for each vehicle crossing.

To train YOLO model we leverage the Vehicle and License Plate dataset from Roboflow Universe, which contains over 2,100 annotated images of vehicles and associated license plates with diverse real-world conditions. Annotations include bounding boxes for both vehicle bodies and plate regions, facilitating training and evaluation of object detection models for ANPR applications [16].

3. Results and Discussion

This section discusses the results of the implementation and evaluation trials of a YOLOv11-based vehicle and license plate detection system and OCR. An analysis is conducted on the model's performance in detecting various vehicle types and license plates from camera-acquired images.

A. YOLO Model Performance Evaluation Results

Based on 100 epochs of training, the YOLOv11 model demonstrated stable and efficient performance, with a training time of approximately 3.029 hours. The evaluation was conducted on 432 validation images with a total of 928 objects, spanning several classes: license plates, cars, motorcycles, and trucks. Overall, as shown in Table 1, the model produced a precision value of 0.859, a recall of 0.840, and an mAP@50 of 0.878, as well as an mAP@50–95 of 0.764. These values indicate that the model has good detection capabilities in identifying vehicles and license plates with a relatively low error rate. This performance is considered adequate for the needs of an automated manifest recording system that demands a balance between accuracy and processing speed.

When viewed per class, License Plate detection shows very high performance with a precision value of 0.987, recall of 0.996, and mAP@50 of 0.995, indicating that the model is very reliable in detecting the presence of vehicle license plates. Meanwhile, the motorcycle class also shows very good performance with mAP@50 of 0.995 and mAP@50–95 of 0.913, reflecting the model's success in detecting two-wheeled vehicles that are visually smaller. In contrast, the detection performance in the truck class shows a relatively lower value with mAP@50 of 0.666 and mAP@50–95 of 0.456. This is due to the limited amount of training data for the truck class, which only consists of 13 instances, so the model has not been fully able to learn the visual variations of heavy vehicles optimally. This finding indicates that increasing the amount and diversity of heavy vehicle training data has the potential to improve system performance in this class.

Table 1. Performance of vehicle and plate detection model

Object Class	Sample	Precision	Recall	mAP@50	mAP@50–95
License Plate	376	0.987	0.996	0.995	0.863
Cars	398	0.873	0.847	0.886	0.772
Motorcycle	141	0.912	0.905	0.995	0.913
Truck	13	0.664	0.611	0.666	0.456
Average (All)	**928**	**0.859**	**0.840**	**0.878**	**0.764**

B. OCR Performance Evaluation.

Table 2 presents the overall performance of the EasyOCR module on the 40 images test dataset. A total of 40 license plates were detected by the YOLO model, of which 23 were successfully recognized by EasyOCR, resulting in a recognition success rate of 57.5%. The average OCR confidence for successfully recognized plates is 0.63, with a maximum confidence of 0.999 and a minimum of 0.31. These results indicate that while object detection is relatively stable, OCR performance remains moderate under real operational conditions. The recognition rate suggests that visual variability in real-world port environments significantly affects text extraction accuracy. The disparity between maximum and minimum confidence values further reflects varying image quality conditions across the dataset.

Table 2. OCR Performance Summary

Metric	Value
Total detected plates	40
Plates successfully recognized (non-empty OCR text)	23
Recognition success rate	57.5%
Average OCR confidence (recognized plates)	0.63
Maximum OCR confidence	0.999
Minimum OCR confidence (recognized)	0.31

Table 3 analyzes OCR performance under different lighting conditions categorized based on average grayscale brightness. The results show that recognition performance is strongly influenced by illumination. Plates captured under low-light conditions (<70 brightness mean) achieved only a 38% recognition rate with an average confidence of 0.42. In contrast, normal lighting conditions improved recognition to 64%, while high illumination conditions (>170 brightness mean) resulted in the highest recognition rate of 75% and average confidence of 0.83. These findings confirm that OCR models are highly sensitive to illumination levels. Insufficient lighting reduces character contrast and increases noise, which degrades feature extraction quality. Therefore, lighting normalization or image enhancement techniques are crucial to improve OCR robustness in operational deployment.

Table 3. Effect of Lighting on OCR Performance

Lighting Condition	Average Brightness	Recognition Rate	Avg OCR Confidence
Low (< 70)	55.2	38%	0.42
Normal (70–170)	104.3	64%	0.68
High (> 170)	188.3	75%	0.83

Table 4 examines the effect of plate size, represented by the distance proxy metric, on OCR performance. The distance proxy is inversely proportional to the relative area of the license plate bounding box within the image, serving as an indirect estimation of camera distance Plates categorized as “Near” (distance proxy < 12) achieved a recognition rate of 78%, whereas “Far” plates (>15) showed a significant drop to 29%. The average OCR confidence similarly decreased as the plate size became smaller. This result demonstrates that plate scale is one of the most critical factors affecting recognition performance. Smaller plate regions contain fewer pixels per character, limiting OCR's ability to accurately segment and classify characters. Therefore, ensuring sufficient plate resolution is essential for reliable deployment.

Table 4. Effect of Plate Size (Distance Proxy)

Distance Proxy Range	Interpretation	Recognition Rate	Avg OCR Confidence
< 12	Near	78%	0.81
12–15	Medium	55%	0.57
> 15	Far	29%	0.34

Table 5 presents OCR performance across different levels of plate rotation (tilt angle). Plates with minimal tilt (0–20°) achieved the highest recognition rate (72%) and average confidence (0.74). Performance slightly decreased as tilt increased, but the degradation was not as significant as that caused by lighting or plate size. This indicates that EasyOCR demonstrates moderate robustness to rotation variations, likely due to built-in spatial transformation handling within its architecture. However, extreme rotations (>45°) still reduce recognition reliability. Overall, tilt is a secondary influencing factor compared to illumination and scale

Table 5. Effect of Rotation (Tilt Angle)

Absolute Tilt (°)	Recognition Rate	Avg OCR Confidence
0–20°	72%	0.74
20–45°	60%	0.61
> 45°	52%	0.58

Table 6 presents the empirically observed optimal visual conditions for license plate OCR recognition. Among all evaluated factors, plate scale (distance proxy) and illumination level demonstrate the strongest influence on recognition performance. Plates captured with brightness values above 100 and distance proxy values below 12 consistently yield recognition rates above 75%. Moderate tilt angles (0°–30°) and sufficient contrast further support stable OCR output. These findings indicate that spatial resolution and lighting conditions are the dominant determinants of OCR reliability in real-world Ro-Ro port environments.

Table 6. Optimal Operational Conditions for High-Performance OCR Recognition

Visual Parameter	Optimal Range (Observed)	Recognition Rate (%)	Avg OCR Confidence	Interpretation
Mean Brightness	> 100	70–80%	0.75–0.85	Adequate illumination improves character separation
Lighting Category	Normal – High	64–75%	0.68–0.83	Stable grayscale distribution enhances OCR reliability
Distance Proxy	< 12	75–78%	0.80+	Larger plate scale improves character resolution
Area Ratio	> 0.006	~75%	0.78	Higher pixel density per character
Tilt Angle	0° – 30°	~70%	0.74	Minimal geometric distortion
Contrast	45 – 70	Stable	0.70+	Clear foreground–background distinction

C. Vehicle Detection Results and Vehicle Numbers

System testing results show that the YOLOv11 and OCR-based object detection model is capable of accurately identifying vehicles and license plates in the Bengkalis–Sungai Pakning Ro-Ro crossing operational scenario. This deep learning model was trained using the Vehicle and License Plate Dataset obtained from Roboflow Universe [16], which provides structured annotations for vehicle classes and license plates with varying environmental conditions, camera angles, and lighting intensity.

Figure 2. Example of Vehicle and License Plate Detection

From the example shown in Figure 2, captured when the vehicle exited the ferry door at the Ro-Ro Port of Bengkalis. The system yields an OCR confidence score of 0.50, which is slightly below the dataset average confidence (0.63) for successfully recognized plates. This indicates moderate certainty in character prediction. Although the plate text is correctly extracted, the moderate confidence reflects the influence of plate scale and image resolution, highlighting that even visually clear plates may not always produce high OCR certainty under real operational conditions.

To ensure clarity and reproducibility of the proposed system, the overall processing pipeline is formulated into two structured algorithms. Algorithm 1 outlines the vehicle and license plate detection procedure using the YOLOv11 model, whereas Algorithm 2 describes the subsequent manifest generation mechanism integrating OCR, spatial association, and temporal duplicate suppression. Algorithm 1 (vehicle and license plate detection), as shown below, aims to simultaneously detect vehicle objects and license plates from images or video frames using the YOLOv11 single-stage object detection model, enabling efficient and accurate object localization as the basis for generating vehicle manifest data.

```
Algorithm 1: YOLO-based Vehicle and License Plate Detection

1: Input frame I
2: Run YOLOv11 model M on frame I with confidence θ and IoU θ_iou
3: Obtain detection results D = {d1, d2, ..., dn}

4: Initialize empty lists D_p (plates) and D_v (vehicles)

5: for each detection d in D do
6:     if class_name(d) contains any keyword in K_p then
7:         if confidence(d) ≥ θ_p then
8:             Add d to D_p
9:         end if
10:    else
11:        if confidence(d) ≥ θ_v then
12:            Add d to D_v
13:        end if
14:    end if
15: end for

16: Return D_p, D_v
```

Algorithm 2 aims to automatically generate vehicle manifest data by integrating OCR, license plate–vehicle association, and a temporal voting-based duplicate suppression mechanism. Algorithm 2 is as follows:

```
Algorithm 2: Automatic Vehicle Manifest Generation

1: Initialize empty track list T
2: Initialize empty manifest table M

3: for each frame t do
4:     for each plate detection p in D_p do
5:         Crop image region Cp from bounding box of p
6:         Apply OCR engine O on Cp to obtain text Tp and confidence Cp_conf
7:         if Cp_conf < θ_ocr then
8:             Continue to next detection
9:         end if

10:        Find vehicle v in D_v with maximum IoU to p
```

```
        if IoU(p, v) ≥ θ_assoc then
            Assign vehicle type Tv to plate Tp
        else
            Tv ← Unknown
        end if

        Match p to existing track in T using IoU
        if matched track exists then
            Update voting scores for plate text and vehicle type
        else
            Create new track and add to T
        end if
    end for

    for each track in T do
        if track not updated for F_c frames then
            Finalize plate text and vehicle type using majority voting
            Append result to manifest table M
            Remove track from T
        end if
    end for
end for

Return manifest table M
```

4. Limitation dan Future Direction

This study primarily focuses on evaluating the visual recognition components of the proposed manifest system, including vehicle detection and license plate OCR performance. Although the system architecture includes e-ticket integration and manifest-level validation mechanisms, these components were not fully evaluated in real operational deployment due to limited access to synchronized ticketing databases and large-scale port transaction logs. Furthermore, system-level manifest accuracy metrics such as full plate–vehicle–timestamp correctness and duplicate suppression effectiveness were not quantitatively measured in this phase. The present evaluation therefore emphasizes detection and recognition performance under real-world environmental conditions.

Future research will extend the evaluation toward full system-level performance analysis, including manifest-level correctness, duplicate rate reduction, and integration accuracy with e-ticketing platforms. Quantitative assessment of plate–vehicle–ticket matching accuracy and real-time synchronization reliability will provide a more comprehensive validation of the proposed intelligent manifest system. Additionally, large-scale deployment across multiple crossing sessions and environmental variations (nighttime, rain, peak-hour congestion) will further assess system robustness under operational constraints.

5. Conclusions

This study presents a deep learning–based framework for intelligent vehicle manifest recording in Ro-Ro ferry operations by integrating YOLOv11 object detection and EasyOCR text recognition. The detection model demonstrates strong performance, achieving a precision of 0.859, recall of 0.840, and mAP@0.5 of 0.878, indicating reliable localization of vehicles and license plates under real operational conditions. However, OCR evaluation on real test images yields a recognition success rate of 57.5%, with an average confidence of 0.63 for successfully recognized plates. Detailed analysis shows that plate scale (distance proxy) and illumination level significantly influence recognition performance. Plates captured at closer distances and under sufficient lighting conditions achieve substantially higher recognition rates, whereas low-light and small-scale plate regions frequently result in partial or failed recognition.

These findings indicate that while the visual detection component is robust, OCR reliability in real-world port environments remains sensitive to environmental variability. Therefore, improvements in camera placement, lighting configuration, and image enhancement techniques are essential for achieving higher manifest-level reliability. Future work will focus on enhancing OCR robustness and conducting full system-level validation, including integration with e-ticketing and duplicate suppression evaluation.

References

[1] N. Q. Iskandar and A. Ghofilin, "Narrative Review: Penerapan Kecerdasan Buatan untuk Sistem Deteksi Kendaraan di Smart City: Studi Mengenai YOLOv8 dan Optical Character Recognition," Fuse-Tek. Elektro, vol. 4, no. 2, pp. 95–104, Dec. 2024, doi: 10.52434/jft.v4i2.41890.

[2] C. Wibawa and D. T. Anggraeni, "Ekstraksi Karakter Plat Nomor Kendaraan Menggunakan Google Vision dan Regex Pattern," Techno.Com, vol. 24, no. 2, pp. 378–390, May 2025, doi: 10.62411/tc.v24i2.12616.

[3] T. H. Rifqi, Z. Auliansyah, and N. R. Nazirah, "Penerapan YOLO dan OCR untuk Deteksi dan Klasifikasi Plat Nomor Kendaraan di Universitas Malikussaleh," Algoritma J. Ilmu Komput. Dan Inform., vol. 9, no. 2, doi: 10.30829/algoritma.v9i2.25636.

[4] S. Sunarta, M. I. E. P. A. Putra, M. D. Chaidir, R. A. Piliang, and A. Songklanaita, "Implementasi dan Pengembangan Sistem Pengenalan Plat Nomor Kendaraan Secara Otomatis Menggunakan Yolo v5 dan Google Vision OCR," JIIP - J. Ilm. Ilmu Pendidik., vol. 8, no. 7, pp. 7745–7753, Jul. 2025, doi: 10.54371/jiip.v8i7.8746.

[5] B. Satya, D. Manongga, . Hendry, and A. Aminuddin, "Optimized YOLOv8 for Automatic License Plate Recognition on Resource Constrained Devices," Eng. Technol. Appl. Sci. Res., vol. 15, no. 2, pp. 21976–21981, Apr. 2025, doi: 10.48084/etasr.9983.

[6] R. N. Suharto, M. H. Zajuli Al Faroby, and Y. Setiawan, "Comparative Evaluation of YOLO-based Models for Indonesian License Plate Detection and Recognition with EASY OCR," Procedia Comput. Sci., vol. 269, pp. 943–952, 2025, doi: 10.1016/j.procs.2025.09.037.

[7] G. A. Hakasin, D. Efendi, H. J. Winatra, and M. A. Ibrahim, "Impact of image pre-processing on enhancing paddleOCR performance for automatic indonesia number plate recognition," Procedia Comput. Sci., vol. 269, pp. 1299–1308, 2025, doi: 10.1016/j.procs.2025.09.071.

[8] T. I. Cahyani, M. Zakiyamani, D. Riana, and S. Hardianti, "Perbandingan Akurasi Pengenalan Karakter Plat Nomor Menggunakan Tesseract Dan Data Latih Emnist," INTECOMS J. Inf. Technol. Comput. Sci., vol. 5, no. 2, pp. 18–27, Sep. 2022, doi: 10.31539/intecoms.v5i2.4463.

[9] B. Buleu, R. Robu, and I. Filip, "A Deep Learning-Based System for Automatic License Plate Recognition Using YOLOv12 and PaddleOCR," Appl. Sci., vol. 15, no. 14, p. 7833, Jul. 2025, doi: 10.3390/app15147833.

[10] F. Sonnara, H. Chihaoui, and F. Filali, "Efficient real-time license plate recognition using deep learning on edge devices," J. Real-Time Image Process., vol. 22, no. 4, p. 159, Aug. 2025, doi: 10.1007/s11554-025-01738-3.

[11] Windu Gata, Dwiza Riana, Muhammad Haris, Maria Irmina Prasetiyowati, and Dika Putri Metalica, "Automated Indonesian Plate Recognition: YOLOv8 Detection and TensorFlow-CNN Character Classification," J. RESTI Rekayasa Sist. Dan Teknol. Inf., vol. 9, no. 3, pp. 544–553, Jun. 2025, doi: 10.29207/resti.v9i3.6310.

[12] H. Armantya, N. Amandasari, Q. N. Salsabillah, N. Safitri, and M. Munsarif, "SISTEM DETEKSI DAN PENGENALAN PLAT NOMOR KENDARAAN MENGGUNAKAN YOLOv11 DAN PADDLEOCR," J. Komput. DAN Teknol. Inf., vol. 3, no. 2, Aug. 2025, doi: 10.26714/jkti.v3i2.18678.

[13] N. Salsabila and S. Sriani, "Enhancing Automated Vehicle License Plate Recognition with YOLOv8 and EasyOCR," J. Inf. Syst. Inform., vol. 6, no. 3, pp. 1577–1597, Sep. 2024, doi: 10.51519/journalisi.v6i3.848.

[14] N. A. Kurniawan and C. A. Sari, "Automatic License Plate Detection System with YOLOv11 Algorithm," J. Appl. Inform. Comput., vol. 9, no. 6, pp. 3097–3109, Dec. 2025, doi: 10.30871/jaic.v9i6.11484.

[15] R. Muhammad Ridwan, M. A. Musyaffa, K. Azis, S. Sumaryo, and A. Zamhuri, "Application of OCR Technology for Vehicle License Plate Detection and Yolo V8 for Car Counting," J. Indones. Sos. Teknol., vol. 5, no. 9, pp. 3369–3385, Sep. 2024, doi: 10.59141/jist.v5i9.1378.

[16] I. T. Sentosa and H. Robbani, "Vehicle and License Plate Dataset." Agustus 2023. [Online]. Available: https://universe.roboflow.com/plat-kendaraan/vehicle-and-license-plate

[17] G. Jocher and J. Qiu, "Ultralytics YOLO11," version 11.0.0, Ultralytics, 2024. [Online]. Available: https://github.com/ultralytics/ultralytics. Accessed: 2025.