International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Efficient License Plate Detection and Recognition with YOLOv7 and OCR

Vincent Fanditama Wijaya*[1], Benfano Soewito[2]

Submitted: 25/01/2024 **Revised:** 03/03/2024 **Accepted:** 11/03/2024

Abstract*:* This journal paper conducts a thorough investigation of YOLOv7's role in License Plate Recognition (LPR) for vehicles, with a specific focus on precision, recall, F1 score, mean Average Precision (mAP), and character recognition. Yet, the adaptability of Easy-OCR to various camera angles and its resource-efficient performance on compact devices present it as a valuable alternative. The future of character recognition in LPR systems should prioritize overcoming environmental challenges, license plate obstructions, and regional design variations. Solutions involve advanced pre-processing techniques and specialized algorithms tailored to address these issues. Simultaneously, refining the integration of Raspberry Pi, YOLOv7, and Easy OCR can lead to a scalable and efficient smart gate parking system. YOLOv7 excels in license plate detection, consistently achieving high precision (0.769), recall (0.8571), and F1 scores surpassing 0.66. Its steadily rising mAP (0.653) further underscores its ability to accurately localize license plates. In character detection, YOLOv7 outperforms Easy-OCR, boasting an average similarity rate of 94% compared to Easy-OCR's 80%. This innovative solution holds immense promise for enhancing parking efficiency, access control, and generating valuable urban data insights, ultimately addressing global parking challenges and improving the quality of urban living.

Keywords: *YOLOv7, License Plate Recognition, Character Recognition, Performance Metrics, Optical Character Recognition*

1. Introduction

License Plate Recognition (LPR), also known as Automatic License Plate Recognition (ALPR), is a technology that has witnessed remarkable progress in recent years. Its increasing relevance stems from the crucial role it plays in enhancing safety and security, optimizing traffic management, and improving overall urban infrastructure. LPR systems utilize a combination of image processing, pattern recognition, and machine learning techniques to identify and interpret license plates on vehicles with a high degree of accuracy and efficiency. Conventional parking systems in buildings are one of the most common problems faced in modern cities today. Parking is a basic requirement for private vehicles, but the increasing number of vehicles has made it increasingly difficult to find adequate parking spaces. Conventional parking systems used in public buildings, such as malls, offices, or shopping centers, generally only rely on conventional payment arrangements through ticket and digital money card methods.

YOLOv7, an acronym for "You Only Look Once," represents a breakthrough in the realm of real-time object detection models driven by deep learning algorithms. This family of models has exhibited exceptional performance across a spectrum of computer vision tasks, owing to its unique architectural design. YOLOv7's architecture is engineered for the efficient and precise detection of objects within images, positioning it as a highly promising solution for the task of license plate localization [1]. The efficacy of YOLOv7 in license plate detection is underpinned by its ability to identify the presence and location of license plates rapidly and accurately within diverse and often challenging visual scenes. This robust object detection capability is particularly vital, where license plate recognition plays a pivotal role in vehicle identification and monitoring.

Easy-OCR is a lightweight and user-friendly tool capable of seamless integration with diverse systems, ranging from mobile devices and cameras to Internet of Things (IoT) devices [2]. Employing Easy-OCR for License Plate Recognition offers a range of advantages, including cost-effectiveness, flexibility, and scalability [3]. Easy-OCR assumes three crucial roles in the license plate character recognition process. Firstly, it specializes in character extraction, distinguishing and isolating individual characters from the license plate image. This involves the intricate task of detecting and segmenting characters, effectively separating them from the image's background and any extraneous elements. Secondly, Easy-OCR engages in character recognition once the characters have been extracted. Employing advanced machine learning techniques, it identifies and interprets these characters, recognizing specific letters, numbers, or symbols displayed on the license plate. Finally, Easy-OCR's capabilities extend to providing the recognized characters as text output, typically in a machine-readable format. This output proves invaluable for diverse

1 Computer Science Department, BINUS Graduate Program – Master of Computer Science, Bina Nusantara University, Jakarta 11380, Indonesia
2 Computer Science Department, BINUS Graduate Program – Master of Computer Science, Bina Nusantara University, Jakarta 11380, Indonesia,
ORCID ID : 0000-0002-3284-5433
** Corresponding Author Email: vincent.wijaya007@binus.ac.id*

applications, such as vehicle identification, access control, or the generation of transaction records.

Previous research has explored similar ideas, like using OCR technology and mobile apps for parking, but often required installing many cameras [4], which can be wasteful. However, recent advances achieved an impressive 93% accuracy in detecting Arabic license plates [5]. Another study demonstrated that Mobilenet v2 SSD can detect license plates with 96% accuracy, outperforming OCR.

In this study, we propose using YOLOv7 for license plate detection and comparing it with Easy-OCR for character recognition. The choice of camera angle for license plate recognition significantly impacts the recognition process's effectiveness, with implications for overall results. In license plate recognition, system performance relies on accurate detection and character recognition. We explore using YOLOv7 for license plate detection and compare Easy-OCR with it for character recognition. The YOLOv7 algorithm is chosen for its superior object detection performance in terms of accuracy and speed [6]. The camera angle choice plays a crucial role in optimizing the license plate detection and character recognition phases, ultimately influencing the License Plate Recognition system's success.

The contribution of this research paper can be summarized in several key points:

1. The paper introduces YOLOv7, a cutting-edge model, for license plate detection, enhancing accuracy and speed.
2. It compares Easy-OCR and YOLOv7, offering insights into their accuracy, efficiency, and reliability for character recognition on license plates.
3. The paper extends its contribution by implementing a Smart Gate Parking Prototype using Raspberry Pi as a Smart Gate Parking Prototype.

This journal paper focuses on advancing license plate recognition systems by introducing the use of the YOLOv7 model for license plate detection, renowned for its exceptional accuracy, F1-score, mAP, and precision. Additionally, it conducts a comprehensive performance comparison between Easy-OCR and YOLOv7 for character recognition, evaluating factors such as accuracy, similarity on resource-constrained devices like Raspberry Pi, reliability, and overall effectiveness. The study develops a specialized model tailored for detecting Indonesian license plates on vehicles, harnessing the power of YOLOv7. Upon successful detection, the system precisely crops the license plate image and employs character recognition techniques for character deciphering. The choice of camera angle emerges as a pivotal determinant in optimizing both license plate detection and character recognition phases, significantly influencing the overall success of License Plate Recognition systems. Moreover, the paper adds to its contribution by implementing a Smart Gate Parking Prototype using Raspberry Pi, demonstrating the practical application of the research in real-world scenarios and enhancing automated vehicle identification and surveillance systems.

2. METHOD

2.1. Research Methodology Step

Figure 1 illustrates the research process, which comprises four key phases: 1) Identifying license plate regions on vehicles using YOLOv7, 2) Detecting characters on license plates with YOLOv7, 3) Recognizing characters on license plates using Optical Character Recognition (OCR), and 4) Developing an IoT prototype. For the initial step, we acquire annotated data on Indonesian license plates from the Roboflow dataset, followed by image preprocessing in Roboflow (e.g., Auto-Orient and 640x640 resolution). The dataset is then divided into training (70%), validation (20%), and testing (10%) subsets for modeling using YOLOv7 and PyTorch. After modeling, we rigorously evaluate precision and accuracy metrics to gauge model performance and select the most promising weight parameters for IoT device implementation.

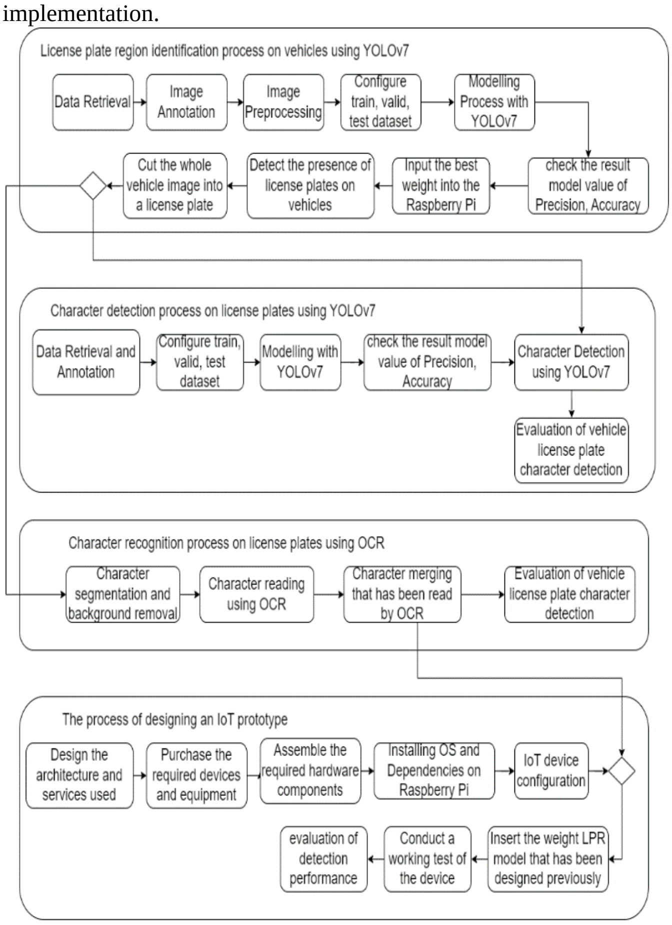


Fig. 1. Stages of Research Activities

In the second phase of our study, we used YOLOv7 for character recognition on license plates. This phase begins with the acquisition of a carefully annotated license plate dataset, divided into training, validation, and testing sets. We focus on modeling each character within the YOLOv7 framework. After modeling, we evaluate critical performance metrics like Precision and Accuracy to assess the model's effectiveness. We also analyze the similarity scores assigned to detected characters on license plates for comparison with OCR-based detection methods.

In our third study phase, we focus on character recognition for Indonesian license plates using OCR. We start with extensive image preprocessing by breaking down the license plate image into individual character components. We perform character segmentation and background removal to improve character texture and readability. The OCR process sequentially recognizes each character on the license plate until all are identified. We evaluate OCR accuracy by comparing it to YOLO-based character detection results, using similarity scores for this comparison.

In the final phase of our study, which focuses on developing IoT prototype devices, we follow a structured process. This begins with designing the device architecture and identifying required services. We then acquire the necessary equipment and carefully assemble and interconnect the hardware. After ensuring the installation of the essential operating system and dependencies on the Raspberry Pi, we configure the IoT device, integrating License Plate Recognition (LPR) software. This leads to a thorough testing and performance evaluation phase to assess device functionality and efficiency.

2.2. Data Collection and Image Preprocessing

In this research, the dataset is obtained through the universe dataset on Roboflow. The dataset consists of 2312 images of black, white, red, and yellow vehicle license plates (cars and motorcycles). The angle and distance of taking pictures of vehicles with license plates were taken randomly. The photos were also taken with varying weather and lighting. The dataset in the form of images collected has certainly been annotated based on 1 class, namely license plates. Image formats are .jpg and .png. An example of an illustration of an image on a dataset that has been annotated based on the location of the license plate attached to the vehicle is shown in Figure 2.

In computer vision tasks, image preprocessing is an important step. It involves converting raw visual data into a machine learning algorithm-friendly format. Auto-orienting, resizing, and augmentation are some of the most used image preprocessing methods. Using the auto-orienting technology, a picture is automatically rotated to the desired orientation. When dealing with images that are not properly aligned, this method is helpful. The EXIF metadata that is present in the image file can be used to implement auto-orienting [7]. Another crucial method used in image preprocessing is resizing. It contains resizing a picture while maintaining its aspect ratio. The size of images in a dataset is frequently standardized by resizing. You may, for instance, change all of the images' dimensions to 640x640 [7]. In the process of augmentation, existing images are subjected to random changes to provide fresh training examples. Rotation, which rotates an image by a certain angle, is a typical sort of augmentation. For instance, [8], [9] you may rotate a picture between -20 and 20 degrees.

Fig. 2. Illustration of an image in the annotated datasets

2.3. Proposed Model

This paper presents a comprehensive research model for the recognition of license plates and their integration into IoT (Internet of Things) devices. Our model consists of four distinct phases, each meticulously designed to contribute to the overarching goal of efficient and accurate license plate recognition for Indonesian vehicles.

2.3.1. Phase 1: License Plate Region Identification and YOLOv7 Modeling

YOLOv4 [10] is the latest official version of YOLO before YOLOv7. It incorporates various techniques to improve the speed and accuracy of YOLOv3 [11], such as Self-adversarial-Training (SAT), Weighted-Residual-Connections (WRC), Cross-Stage-Partial-Connections (CSP), Cross mini-Batch Normalization (CmBN), and Mish activation. YOLOv4 also uses a modified version of the Darknet backbone network, called CSPDarknet53, and a new neck structure, called SPP (Spatial Pyramid Pooling). YOLOv5 [12] is an unofficial variant of YOLO that was released by Ultralytics in 2020. It is not a continuation of YOLOv4, but rather a reimplementation of YOLO using PyTorch, a popular deep-learning framework. YOLOv5 also introduces some changes to the network architecture, such as using EfficientNet as the backbone, In this paper, we propose YOLOv7, a new state-of-the-art real-time object detection model that surpasses all the previous versions and variants of YOLO in both speed and accuracy. YOLOv7 achieves 56.8% AP on MS COCO at a speed of 30 FPS on a GPU, outperforming all the existing real-time object detectors [13]. YOLOv7 segmentation in the mask branch is a SingleStage approach, which makes it efficient in this area [14]. Before being sent into the backbone network, the input image is first downsized to 640 640 pixels [15]. Another CNN model for object detection is Faster R-CNN [16], which consists of two modules: a region proposal network (RPN) that generates candidate bounding boxes, and a classifier network that predicts the class and refinement of each box. Faster R-CNN can use different backbone networks, such as VGG, ResNet, or MobileNet, to extract features from the input image [17]. Faster R-CNN achieves high accuracy on various object detection benchmarks, but it is relatively slow compared to other methods.

To find the license plate to be recognized, direct object detection is used. The goal is to speed up the detection process by skipping the whole car detection step, which inevitably extends the execution time. The YOLO (You Only Look at Once) v7 architecture is used in the detection process, which is supported by the Tensorflow object detection API model. The model is created from a dataset that is trained using transfer learning and then used to recognize license plates.

2.3.2. Phase 2: Character Recognition with YOLOv7

Our second phase focuses on character recognition within license plates, utilizing the YOLOv7 framework as Figure 3 shows the YOLOv7 Architecture. We delve into character-level modeling, and the YOLOv7 model is evaluated rigorously to gauge its performance, with Precision and Accuracy metrics at the forefront. Additionally, we conduct a detailed analysis of similarity scores for characters, facilitating comparisons with OCR-based methods.

A major area of study in computer vision is the use of neural networks for character recognition. Character recognition using neural networks is efficient. A technique for picture character recognition using convolutional neural networks was developed by [18]. [19] employed finite state approaches and neural networks for optical character recognition and post-correction. [20] uses deep learning to recognize handwritten characters. Convolutional neural networks were utilized by [21] for character recognition from images.

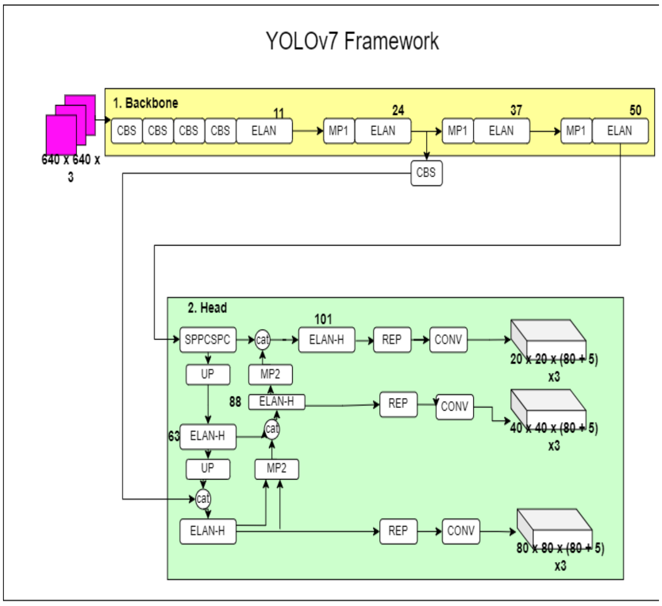


Fig. 3. YOLOv7 Architecture

2.3.3. Phase 3: Character Recognition via Optical Character Recognition (OCR)

A technique called optical character recognition (OCR) makes it possible to convert different kinds of texts or photos into editable, searchable, and analyzable data. Numerous studies have examined how well various OCR programs perform on various document kinds and languages. Hegghammer [22] performed a benchmarking experiment, for instance, contrasting the precision of Tesseract, Amazon Textract, and Google Document AI on photos of English and Arabic text with various kinds of fake noise. Additionally, he discovered that English had far greater accuracy than Arabic. A thorough review of the 176 handwritten OCR research articles published between 2000 and 2019 was offered in a different study [23]. A study by [19] proposed a post-processing method for improving the OCR quality of historical Finnish newspapers and journals.

Easy OCR is a Python library that allows users to perform optical character recognition (OCR) on a variety of documents or pictures with minimal configuration and coding. Several studies have compared or used Easy OCR for different OCR tasks and applications. For example, [24] evaluated the performance of Easy OCR on Arabic historical manuscripts and found that it achieved higher accuracy than Tesseract, and Kraken [25] used Easy OCR to extract text from license plates and vehicle identification numbers in images captured by traffic cameras.

After the vehicle license plate detection process, there is a process of identifying each character on the license plate using the Easy-OCR framework. Easy-OCR has the advantage of being easy to use and configure in various programming languages, friendly to run on devices that have minimum resources such as Raspberry Pi, and quite proven accuracy on Arabic license plate data which reached 93% [26]. Figure 4 explains the process flow for character identification using Easy-OCR. We also conducted a detailed analysis of similarity scores for characters, facilitating comparisons with YOLOv7 character recognition methods.

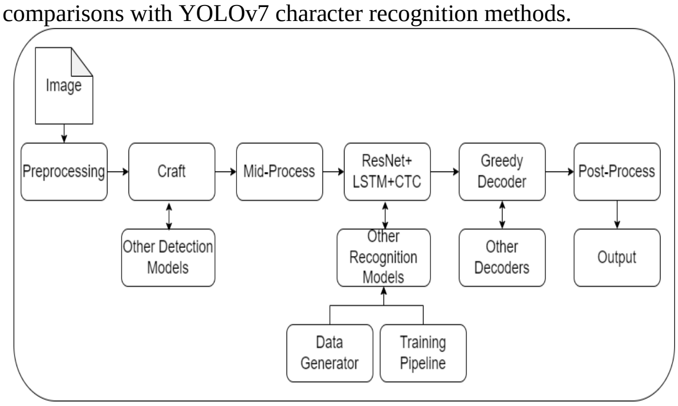


Fig. 4. Stages of Identification Character using Easy-OCR

2.3.4. Phase 4: Development of IoT Prototype Devices

The final phase of our model focuses on the integration of license plate recognition capabilities into IoT prototype devices. This phase encompasses the design of device architecture, the acquisition of necessary hardware, and meticulous assembly. We ensure the installation of essential operating systems and dependencies on the Raspberry Pi platform. Once the IoT device is configured, we seamlessly integrate the License Plate Recognition (LPR) software, allowing real-time recognition tasks. Subsequently, a comprehensive testing and performance evaluation phase assesses the functionality and efficiency of the IoT prototype device in practical scenarios.

Traditional LPR methods usually consist of three steps: license plate location, character segmentation, and character recognition. License plate location methods can be based on color features, edge features, or a combination of both [27]. However, LPR is a difficult undertaking that needs managing a variety of aspects such as illumination, weather, motion blur, and occlusion. Several studies have proposed different methods and techniques to improve the performance of LPR in complex environments. For example, [28] proposed an end-to-end LPR approach that uses a resampling-based cascaded framework and two convolutional neural networks (CNNs) for license plate detection and recognition. They also introduced vertex estimation and weight-sharing techniques to enhance the accuracy and speed of LPR [29] and presented a performance enhancement method for multiple LPRs in challenging conditions. They also employed a look-up table classifier with adaptive boosting and modified census transform for character recognition [30] and developed a unique abstractive method utilizing a seq2seq neural model to produce a related work section for a scientific paper.

Through the implementation of this research model, we aim to address the complexities and challenges associated with license plate recognition, character detection, and recognition on Indonesian license plates. Additionally, the successful integration of our model into IoT devices promises novel applications and advancements in computerized methods for reading license plates.

When the vehicle is in front of the parking gate, the camera will detect the presence of the vehicle license plate, If the presence of the vehicle license plate is detected, the LPR system will continue to identify the characters on the license plate using OCR. Any detection failure in LPR will experience repetition for detection. Furthermore, the Raspberry Pi will check the license plate data in the online database whether to get authorization to enter / exit. If the license plate is already in the database, the gate will open, otherwise, if the license plate is not in the database, an error message will appear on the display message. A flowchart illustration of the flow of the Automatic Gate can be seen in Figure 5 below.

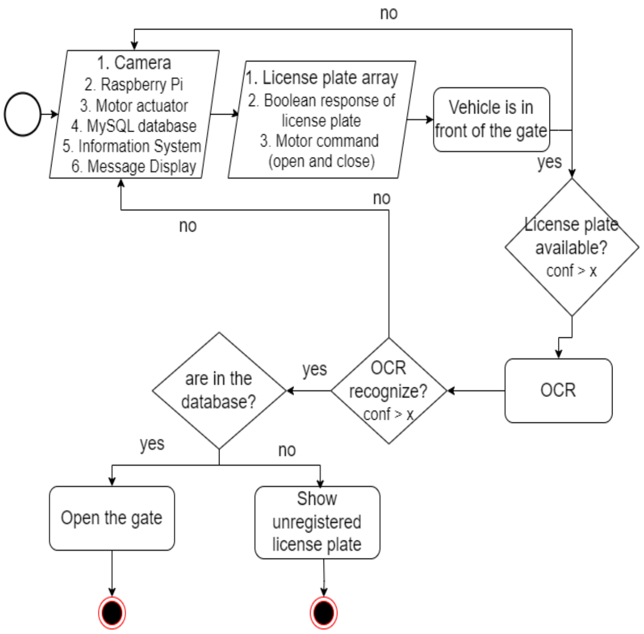


Fig. 5. Automatic Gate Flowchart

2.4. Evaluation Method

There are several evaluation methods to assess the performance of the License Plate Recognition algorithm, including recall (1), precision (2), mean Average Precision (3), F1 score (4), and character recognition similarity (5).

$$Recall = \frac{TP}{TP + FN} \quad (1)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2)$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \quad (3)$$

$$\text{F1 Score} = 2 \times \frac{\text{recall } \times \text{precison}}{\text{recall } + \text{precison}} \quad (4)$$

$$\text{Similarity} = 100 \times \frac{\text{Total Identified Characters}}{\text{Total Factual Characters}} \quad (5)$$

3. Results and Discussion

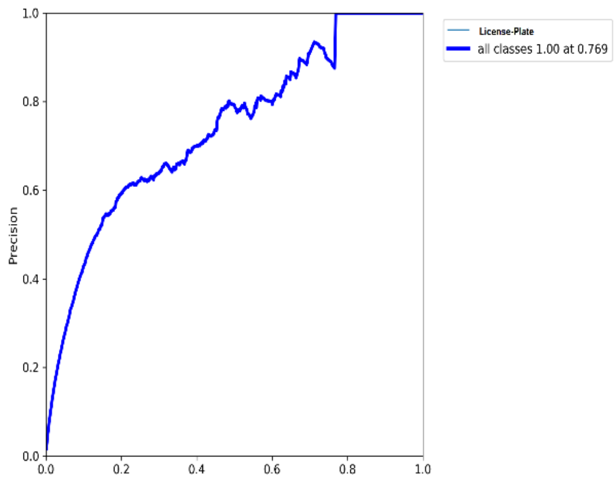
In this research study, we experimented to evaluate the performance of YOLOv7 in the context of License Plate Recognition (LPR) for vehicles. Our experiment was designed to assess the model's precision, recall, mean Average Precision (mAP), F1 score, and their evolution over training epochs and batch sizes.

3.1. YOLOv7 Training Results on License Plate Detection

We trained the YOLOv7 model on a dataset containing annotated images of vehicles with affixed license plates. The dataset encompassed a diverse range of scenarios, including varying lighting conditions, vehicle types, and shooting angle of the camera. We employed a grid search approach to determine the optimal hyperparameters for the YOLOv7 architecture, including the number of epochs and batch size.

Our experimentation revealed that after training for a total of 100 epochs with a batch size of 16, YOLOv7 achieved outstanding results in license plate detection for vehicles. The precision, recall, and F1 score consistently improved as the training progressed. The precision value is 0.769 as Figure 6, indicating a high level of accuracy in identifying license plates. The recall value as Figure 7, is 0.8571, demonstrating the model's ability to effectively detect license plates in a variety of scenarios. The F1 scores as Figure 8 consistently exceeded 0.66, affirming the robustness and effectiveness of YOLOv7 in LPR. Furthermore, our experiments demonstrated a steady increase in mAP as Figure 9 with training epochs, reaching a good value of 0.653 by the end of the training process. This result underscores the model's proficiency in localizing license plates with high precision while minimizing false positives.

Fig. 6. The YOLOv7 license plate detection precision graph



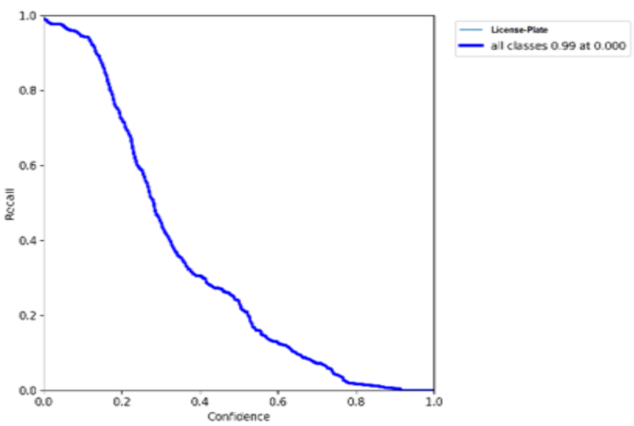


Fig. 7. The YOLOv7 license plate detection recall graph

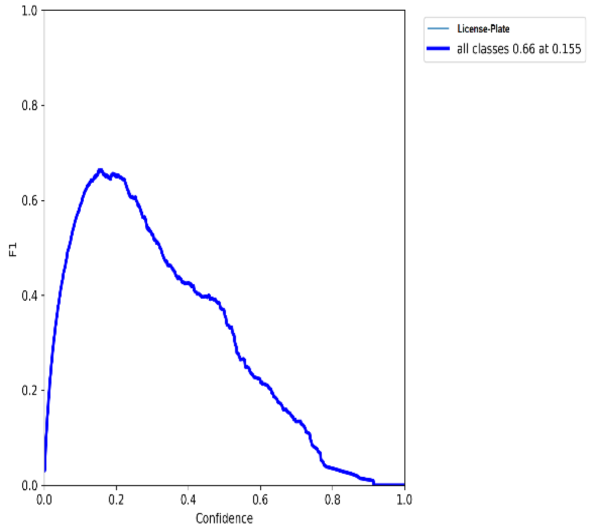


Fig. 8. The YOLOv7 license plate detection F1-Score graph

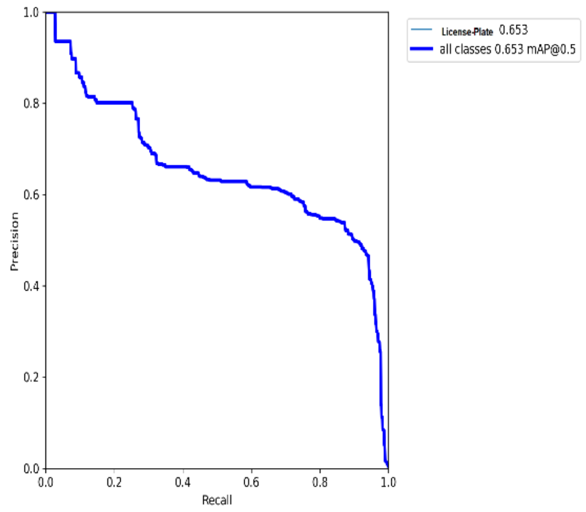


Fig. 9. The YOLOv7 license plate detection mAP@0.5 graph

3.2. Comparison Results of Character Detection using YOLOv7 and Easy OCR

The process of assessing the accuracy of recognizing characters on license plates involves a method where we compare the character, we expected should be on the plate with the characters the system predicts. In this prediction, we compare the Easy-OCR versus YOLOv7 for character detection. In this evaluation, we employ images featuring different kinds of vehicles and conduct five trial runs for each. Then we calculate the similarity percentage for each license plate separately and subsequently determine the overall average for all the samples tested. You can find the outcomes of these tests in Table 1.

The results of character detection similarity percentage between YOLOv7 and Easy-OCR reveal an interesting contrast. In a series of five trials, YOLOv7 consistently outperformed Easy-OCR, with an average similarity rate of 94% compared to Easy-OCR's 80%. After conducting five trials, it became evident that we limited the trials to this number due to the negligible differences in similarity observed beyond this point. This suggests that YOLOv7 is more accurate in recognizing characters in various contexts. However, it's important to note that Easy-OCR has unique advantages that make it a valuable choice, particularly for applications on small devices like the Raspberry Pi. Easy-OCR's ability to run efficiently on resource-constrained hardware makes it a practical option for scenarios where YOLOv7 might be impractical due to its computational demands. This is particularly useful for embedded systems and edge devices. Additionally, when recognizing license plates, the camera angle is quite important. While YOLOv7 may excel in terms of character recognition accuracy, Easy-OCR's adaptability to different camera angles could be a significant advantage in real-world LPR applications. Easy-OCR's versatility in handling non-ideal angles and less-than-optimal image conditions could make it a preferred choice in situations where capturing license plate data under varying camera angles is a common occurrence.

Table 1 . Character Detection Similarity Percentage Results between YOLOv7 and Easy-OCR

Trial	**YOLOv7**	**Easy-OCR**
1st trial	93.89%	80.21%
2nd trial	94.01%	79.31%
3rd trial	94.01%	78.50%
4th trial	94.01%	80.55%
5th trial	94.01%	80.02%
Average Similarity	**94%**	**80%**

Character recognition in license plate recognition (LPR) systems can sometimes face challenges beyond the choice of model and software. For instance, environmental factors like poor lighting or adverse weather conditions can significantly impact recognition accuracy. Moreover, physical obstructions such as dirt, mud, or even a simple layer of dust on the license plate can hinder character detection, leading to false positives or negatives in the OCR process. In some cases, foreign objects like bolts or stickers might adhere to the license plate's surface, further complicating character recognition. These obstructions can cause character shapes to deform or become partially obscured, making it difficult for both YOLOv7 and Easy-OCR to accurately identify the characters. As a result, there's a need for pre-processing techniques to clean up the image or for specialized algorithms designed to handle such situations. Additionally, variations in license plate design and font styles across different regions and countries pose another challenge for character recognition systems. While YOLOv7 and Easy-OCR may perform well on standard license plates, they might struggle with non-standard designs or exotic fonts, which are more common in some regions.

3.3. Implementation of Smart Parking Prototype using Raspberry Pi

The culmination of our License Plate Recognition (LPR) project, developed as a pioneering prototype for smart gate parking, underscores the transformative potential of combining the Raspberry Pi, YOLOv7, and Easy OCR technologies. In an era where urbanization is on the rise and parking congestion poses significant challenges, our innovative solution offers a ray of hope for improving the efficiency and security of parking facilities. At its core, the Raspberry Pi serves as the brains behind this operation, offering a cost-effective and reliable computing platform. By harnessing YOLOv7's state-of-the-art object detection capabilities, our system can swiftly and accurately identify vehicles and, more importantly, their license plates. The real magic happens when Easy OCR steps in, extracting license plate information with remarkable precision.

This smart gate parking prototype holds immense promise for revolutionizing the way we manage parking spaces as shown in Figure 10. With real-time license plate recognition, it streamlines access control, reducing wait times, and enhancing security by only allowing authorized vehicle entry. Moreover, the system can record and log license plate data, offering valuable insights into parking facility usage patterns and enabling more efficient space allocation.

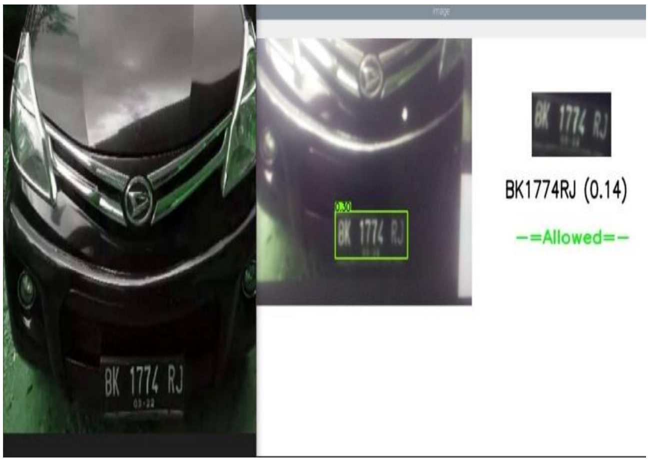


Fig. 10. Prototype for Smart Gate Parking using LPR System

Beyond the practical benefits, this project underscores the transformative potential of collaborative innovation. By leveraging open-source technologies and the collective knowledge of the developer community, we have crafted a solution that has far-reaching implications for smart urban infrastructure. As we move forward, we envision this prototype evolving into a scalable, real-world solution that can help alleviate parking challenges in cities worldwide, enhancing the overall quality of urban living. The future of smart gate parking has never looked brighter.

4. Conclusion

In conclusion, our evaluation of YOLOv7's performance in license plate detection has yielded positive results, demonstrating high precision (0.769), recall (0.8571), and F1 scores consistently exceeding 0.66. The model also exhibits a steady increase in mean Average Precision (mAP) at 0.653, affirming its proficiency in accurately localizing license plates within images. Furthermore, YOLOv7 outperforms Easy-OCR in character detection, achieving an average Similarity Score of 94% compared to Easy-OCR's 80%. Nonetheless, it's important to acknowledge Easy-OCR's unique strengths, including its adaptability to various camera angles and resource-efficient performance on small devices like Raspberry Pi, making it a viable alternative. Our experimentation and implementation on IoT devices, specifically Raspberry Pi, for Object Detection and Optical Character Recognition (OCR) have shown promising results. This algorithm encompasses critical stages such as image preprocessing, YOLOv7-based object detection, OCR within the Region of Interest (ROI), and decision-making based on OCR outcomes, with the option to visualize the processed results. The primary output of this IoT algorithm is the 'action_status' variable, a crucial determinant for license plate recognition. Depending on the OCR results, it classifies detected text as 'Allowed' or 'Prohibited,' facilitating decisions regarding access permissions. Placing the camera on the car's front bumper ensures efficient capture of license plates.

Future work in character recognition for License Plate Recognition (LPR) systems should prioritize tackling environmental challenges, like adverse weather and poor lighting, by developing robust pre-processing techniques and specialized algorithms. Additionally, efforts should be directed at improving character recognition even when license plates are obstructed by dirt or stickers. The integration of these technologies has the potential to enhance parking efficiency in urban areas by reducing congestion, improving access control, and streamlining the parking experience. Upgrading LPR system hardware, such as using higher-quality cameras and more powerful processors, can enhance detection accuracy and speed. Moreover, exploring offloading data to cloud servers offers scalability and improved data handling capabilities for LPR systems, making it a viable consideration for future developments.

Acknowledgment

This publication is fully supported by Bina Nusantara University.

Author contributions

Vincent Fanditama Wijaya: Writing-Original draft preparation, Visualization, Investigation, Writing-Reviewing and Editing.

Benfano Soewito: Conceptualization, Methodology, Field study, Writing-Original draft preparation.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] H. Moussaoui, N. EL AKKAD and M. BENSLIMANE, "Arabic and Latin License Plate Detection and Recognition based on YOLOv7 and Image Processing Methods," 2023.

[2] N. B. P. Awalgaonkar and R. Chaugule, "Automatic License Plate Recognition System using SSD," *International Symposium of Asian Control Association on Intelligent Robotics and Industrial Automation,* pp. 394-399, 2021.

[3] Q. M. Aljelawy and T. M. Salman, "Detecting License Plate Number Using OCR Technique and Raspberry Pi 4 With Camera," in *2nd International Conference on Computing and Machine Intelligence,* Turkey, 2022.

[4] R. Lookmuang, K. Nambut and S. Usanavasin, "Smart Parking using IoT Technology," in *5th International Conference on Business and Industrial Research (ICBIR),*

2018.

[5] A. O. Agbeyangi, O. A. Alashiri and A. E. Otunuga, "Automatic Identification of Vehicle Plate Number using Raspberry Pi," in *International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS)*, 2020.

[6] L. Cao, X. Zheng and L. Fang, "The Semantic Segmentation of Standing Tree Images Based on the Yolo V7 Deep Learning Algorithm," *Electronics,* vol. 12, no. 4, p. 929, 2023.

[7] L. Zhou, C. Zhang, F. Liu, Z. Qiu and Y. He, "Application of Deep Learning in Food: a review.," *Comprehensive Reviews in Food Science and Food Safety,* vol. 18, no. 6, pp. 1793-1811, 2019.

[8] P. Chlap, H. Min, N. Vandenberg, J. Dowling, L. Holloway and A. Haworth, "A Review of Medical Image Data Augmentation Techniques for Deep Learning Applications," *Journal of Medical Imaging and Radiation Oncology,* vol. 65, no. 5, pp. 545-563, 2021.

[9] C. Shorten and T. M. Khoshgoftaar, "A Survey on Image Data Augmentation for Deep Learning," *Journal of Big Data,* vol. 6, no. 1, pp. 1-48, 2019.

[10] A. Bochkovskiy, C. Y. Wang and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv,* vol. 10934, no. 2004, pp. 4-5, 2020.

[11] S. Singha and B. Aydin, "Automated Drone Detection using YOLOv4," *Drones,* vol. V, no. 3, p. 95, 2021.

[12] G. Jocher, L. Changyu, A. Hogan, L. Yu, P. Rai and T. Sullivan, " Ultralytics/yolov5: Initial Release," *Zenodo,* 2020.

[13] C. Y. Wang, A. Bochkovskiy and H. Y. M. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-art for Real-time Object Detectors," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[14] K. Jiang, T. Xie, R. Yan, X. Wen, D. Li, H. Jiang and J. Wang, "An Attention Mechanism-Improved YOLOv7 Object Detection Algorithm for Hemp Duck Count Estimation," *Agriculture,* vol. 12, no. 10, p. 1659, 2022.

[15] W. Zhu, Q. Wang, L. Luo, Y. Zhang, Q. Lu, W. C. Yeh and J. Liang, "CPAM: Cross Patch Attention Module for Complex Texture Tile Block Defect Detection," *Applied Sciences,* vol. 12, no. 23, p. 11959, 2022.

[16] X. Sun, P. Wu and S. C. Hoi, "Face Detection using Deep Learning: An Improved Faster RCNN Approach," *Neurocomputing,* no. 299, pp. 42-50, 2018.

[17] L. Yang, J. Zhong, Y. Zhang, S. Bai, G. Li, Y. Yang and J. Zhang, "An Improving Faster-RCNN with Multi-Attention Resnet for Small Target Detection in Intelligent Autonomous Transport with 6G," *IEEE Transactions on Intelligent Transportation Systems,* 2022.

[18] A. Narayan and R. Muthalagu, "Image Character Recognition using Convolutional Neural Networks," in *Seventh International Conference on Bio Signals, Images, and Instrumentation*, Philadelphia, 2021.

[19] S. Drobac and K. Lindén, "Optical Character Recognition with Neural Networks and Post-Correction with Finite State Methods," *International Journal on Document Analysis and Recognition (IJDAR),* vol. 23, no. 4, pp. 279-295, 2020.

[20] B. Rajyagor and R. Rakhlia, "Handwritten Character Recognition using Deep Learning," *Int. J. Recent Technol. Eng,* vol. 8, no. 6, pp. 2277-3878, 2020.

[21] M. Al-Arini, F. Al-Hafiz and S. Amash, "Character Recognition from Images using a Convolutional Neural Network," *Sixth International Congress on Information and Communication Technology,* vol. 4, pp. 403-413, 2022.

[22] T. Hegghammer, "OCR with Tesseract, Amazon Textract, and Google Document AI: a Benchmarking Experiment," *Journal of Computational Social Science,* vol. V, no. 1, pp. 861-882, 2022.

[23] J. Memon, M. Sami, R. A. Khan and M. Uddin, "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)," *IEEE Access,* vol. 8, pp. 142642-142668, 2020.

[24] K. Baierer, R. Dong and C. Neudecker, "Okralact-A Multi-Engine Open Source OCR Training System," *5th International Workshop on Historical Document Imaging and Processing,* pp. 25-30, 2019.

[25] N. Saleem, H. Muazzam, H. M. Tahir and U. Farooq, "Automatic License Plate Recognition using Extracted Features," *4th International Symposium on Computational and Business Intelligence,* pp. 221-225, 2016.

[26] M. M. Abdellatif, N. H. Elshabasy, A. E. Elashmawy and M. AbdelRaheem, "A Low Cost IoT-based Arabic License Plate Recognition Model for Smart Parking Systems," *Ain Shams Engineering Journal,* vol. 6, no. 102178, p. 14, 2023.

[27] C. Yang and L. Zhou, "Design and Implementation of License Plate Recognition System Based on Android," in *11th International Conference on Computer Engineering and Networks*, Singapore, 2022.

[28] Y. Wang, Z. P. Bian, Y. Zhou and L. P. Chau, "Rethinking and Designing a High-performing Automatic License Plate Recognition Approach," *IEEE Transactions on Intelligent Transportation Systems,* vol. 23, no. 7, pp. 8868-8880, 2021.

[29] K. Khan, A. Imran, H. Z. U. Rehman, A. Fazil, M. Zakwan and Z. Mahmood, "Performance Enhancement Method for Multiple License Plate Recognition in Challenging Environments," *EURASIP Journal on Image and Video Processing,* vol. 1, pp. 1-23, 2021.

[30] H. Yu, X. Wang, Y. Shao, F. Qin, B. Chen and S. Gong, "Research on License Plate Location and Recognition in Complex Environment," *Journal of Real-Time Image Processing,* vol. 19, no. 4, pp. 823-837, 2022.