Proceedings of the 2025 SICE Festival with Annual Conference
September 9-12, 2025, Chiang Mai, Thailand

Detection of OCR Misclassifications in ALPR via Rapid Prototyping

Saung Hnin Pwint Oo[1†], Yuki Yoshikawa[2] and Ikusaburo Kurimoto[3]

[1,2]Computer Engineering Department, KOSEN - King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand
[1]E-mail: saunghninpwint.oo@kmitl.ac.th and [2]E-mail: yuki.yo@kmitl.ac.th
[3]Faculty of International Economics and Management, International Pacific University of Tokyo, Chiba, Japan
[3]E-mail: ikusaburo.kurimoto@ipu-japan.ac.jp

Abstract: This paper originates from the limitations of traditional Optical Character Recognition (OCR) systems, particularly in recognizing multilingual texts (English and Thai characters) within Automatic License Plate Recognition (ALPR). A common challenge in current OCR models is character confusion, such as misreading between "0" and "O", and "8" and "B," which can significantly impact character recognition accuracy. Since OCR performance directly affects the overall effectiveness of ALPR systems, such misclassifications lead to incorrect vehicle identification. Therefore, this paper introduces a rapid prototyping approach for ALPR, focusing on multilingual license plates, especially Thai license plates that include both Latin (English) and non-Latin (Thai) characters. It aims to emphasize the identification and analysis of OCR misclassifications in how characters are recognized by OCR.

Keywords: Optical Character Recognition, Object Detection, Automatic License Plate Recognition, Rapid Prototyping

1. INTRODUCTION

Automatic License Plate Recognition (ALPR), also known as Automatic Number Plate Recognition (ANPR), is a technology of computer vision (CV) that plays an essential role in contemporary intelligent transportation systems. It is widely used for vehicle monitoring in various applications, including traffic law enforcement, toll payment systems, vehicle tracking, access control, parking management, and border surveillance. ALPR systems identify vehicles by extracting license plate information from images or videos, which detect the license plate region and recognize the characters on the plate.

ALPR comprises multiple processes, including capturing ground truth images, image preprocessing, and implementation of the ALPR pipeline. The pipeline contains three main stages: license plate detection, character segmentation, and Optical Character Recognition (OCR), followed by post-processing to refine the results. Figure 1 shows an overview of the general ALPR workflow.

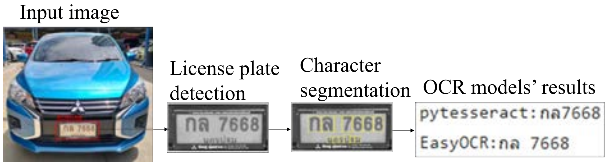

Fig. 1 Overview of the general ALPR workflow.

Among these stages, OCR[1] plays a pivotal role, as its quality significantly impacts the overall system performance. Inaccurate OCR causes vehicle misidentification so ALPR becomes challenging in real-time environments where lighting conditions, weather effects, time of day, motion blur, angle views and license plate designs vary.

Despite several researches have explored ALPR advancements ranging from traditional methods such as edge detection to modern Deep Learning (DL) based techniques, significant limitations remain in OCR in recognizing multilingual license plates. Popular OCR engines such as Tesseract and EasyOCR are primarily trained on English or Latin-based datasets. Hence, they perform well on Latin-based license plates but often struggle with non-Latin characters. Typical OCR errors involve confusion between visually similar characters and misreading of digits and letters. For example, in Thai license plates, OCR may confuse Thai characters with English characters (e.g., "ภ" with "P"), similar characters (e.g., "ก" with "ภ"), digits with characters (e.g., "0" with "O"). These errors are often caused by script similarities, low-resolution images, inconsistent fonts, and poor lighting conditions. So, robust OCR systems must accurately convert visual text on license plates into machine-readable data, maintaining the ALPR's effectiveness under dynamic environments. Thus, the enhancement of ALPR is necessary to integrate with advanced techniques such as character-level classification, misclassification analysis and post-processing techniques for error correction.

This paper proposes a rapid prototyping approach to ALPR, emphasizing the identification and analysis of OCR misclassifications in Thai license plates. The primary objective is to address challenges in license plate detection and character recognition, ultimately reducing vehicle misidentification. Therefore, an ALPR system is rapidly built, evaluated, and refined prior to full deployment. By using publicly available datasets and applying image preprocessing techniques, we enhance the dataset diversity and improve model generalization. For license plate detection, we apply a fine-tuned Faster R-CNN ResNet-50 model, achieving high precision with a 90% confidence threshold. This model reliably localizes license plate regions, which are then processed by an OCR model (EasyOCR in this study) for character recognition. Experimental results describe that the OCR model

† Saung Hnin Pwint Oo is the presenter of this paper.
[1]OCR allows machines to automatically recognize segmented characters and extract text from printed or handwritten documents, images, or video frames.

performs well on license plates where text is aligned horizontally. However, OCR accuracy decreases in cases of angular distortions, such as skewed or side-view images, leading to misclassification. Therefore, we analyze OCR misclassifications at the character level and investigate post-processing strategies including post-OCR processing, text region adjustment, automatic skew correction, and image quality enhancement. Among these strategies, this study specifically implements character reordering as a post-processing method to correct character placement errors in OCR results. This method filters out irrelevant characters using format-specific rules and error patterns, then rearranges the recognized characters to match the standard Thai license plate format. Implementation of this method leads to an improvement in recognition accuracy, increasing from 32% without character reordering to 47% with the applied method. The other strategies remain conceptually proposed and are reserved for future work to further enhance system accuracy and robustness.

Rest of this paper is structured as follows. Section 2 reviews existing approaches related to ALPR. Section 3 outlines the theoretical backgrounds applied in this study. Section 4 presents a simplified explanation of our proposed methodology. Section 5 discusses the experimental results, and Section 6 concludes the paper.

2. EXISTING APPROACHES

Thai license plate detection and recognition has been extensively researched to enhance the effectiveness of ALPR systems. Ogiuchi et al. (2014) developed a Thai ALPR system employed a Bag of Features approach utilizing Histogram of Oriented Gradients (HOG) and a linear Support Vector Machine (SVM) for plate detection, achieving a 94.6% plate detection rate and a 92.0% character recognition rate [1]. Puarungroj and Boonsirisumpun (2018) proposed a DL-based method for recognizing Thai motorcycle license plates, which consist of three lines indicating category, location, and number [2]. It used MobileNets and Inception-v3, achieving 96.94% line recognition accuracy and 91.76% character recognition accuracy. To recognize low-resolution province names, Kraisin and Kaothanthong (2018) introduced a two-step classification method in which names were first grouped by character length, followed by HOG feature extraction and classification, reaching 90% accuracy [3]. Rattanawong et al. (2021) presented a system based on Mask R-CNN and YOLO to detect multiple license plates per image, achieving an overall accuracy of 89.08% [4].

Extending the relevance of ALPR to other languages, Alwateer et al. (2024) presented a system for Saudi license plates using YOLOv8 for real-time detection and CNN for character recognition [5]. The system integrated Local Interpretable Model-agnostic Explanations to enhance transparency, achieving high performance with mAP@0.5 of 0.96 and mAP@0.95 of 0.97. To address challenges such as inconsistent environmental conditions, data imbalance, and accuracy, Akoushideh et al. (2024) employed YOLO for detection, DeepSORT for tracking, and specialized preprocessing and counting methods, resulting in 99.18% accuracy for nighttime detection [6]. Wang et al. (2022) introduced a segmentation-free framework that handles adverse weather and lighting conditions, while maintaining high accuracy and low computational demand, achieving over 99% accuracy and processes at over 70 fps [7]. Zandi and Rajabi (2022) developed a deep CNN framework for Iranian ALPR, utilizing YOLOv3 for plate detection and Faster R-CNN for character classification, achieving 99.6% mAP, 98.26% recall, 98.08% accuracy with YOLOv3, and 98.97% recall, 99.9% precision, 98.8% accuracy with Faster R-CNN [8]. Shafi et al. (2022) explored a ALPR system for non-standard license plates in Pakistan using a 53-layer CNN and YOLOv3, achieving 97.82% detection accuracy and 96% character recognition accuracy [9]. Izidio et al. (2020) presented a CNN-based ALPR system to optimize embedded devices, using Tiny YOLOv3 for plate detection and a fine-tuned CNN for character recognition, resulting in 99.37% detection and 98.43% recognition accuracy on Brazilian license plates [10]. Lin et al. (2022) introduced a real-time edge-AI-based ALPR system utilizing the AGX XAVIER embedded device, which integrates YOLOv4-Tiny for license plate detection and a modified YOLOv4 for character recognition, successfully reducing latency and bandwidth usage [11]. Tested in real-life environments in Taiwan, Bansal et al. (2024) enhanced an ALPR system achieving recognition accuracy rates of 97% during the day and 95% at night. The paper used a fine-tuned YOLO object detector, achieving 94% accuracy and 80% recall, while effectively addressing challenges such as lighting, weather, and camera angles through advanced augmentation techniques, including inverted and flipped characters [12]. An Artificial Neural Network-based OCR algorithm for ALPR was proposed by Zhai et al. (2013), achieving a 97.3% character recognition rate in real-time with an FPGA board [13].

Traditional ALPR systems mainly rely on classical CV and Machine Learning techniques, such as edge detection, template matching, and SVMs, which were effective in simpler environments but struggle with dynamic environment varying lighting and camera angles [1, 3]. Modern ALPR systems have shifted to DL-based approaches, using models such as CNN, R-CNN, and YOLO for simultaneous detection and recognition, offering improved accuracy and speed [2, 4-6, 8-13]. These systems also employ transfer learning, edge AI technology, and data augmentation techniques, making them robust and capable of handling diverse and challenging conditions.

3. THEORETICAL BACKGROUNDS

This section outlines the theoretical background of our approach, covering key concepts relevant to object detection and optical character recognition.

3.1. Object detection

Object detection is applied in many areas of CV such as facial recognition, video surveillance, and autonomous driving. It identifies the categories of objects in images and extract their exact locations using bounding boxes. For instance, YOLO (You Only Look Once)[2] is one of the most notable object detection frameworks in real-time detection with significance performance [14]. Traditional object detection methods based on hand-crafted features combined with sliding window techniques [15, 16]. Modern detection systems have evolved significantly by integrating DL models, which enhance performance and reduce computational complexity.

In 2024, Ross Girshick et al introduced Region-based Convolutional Neural Networks (R-CNN), which integrates deep CNNs for feature extraction, SVM for classification, and regression for localizing bounding box regions [17]. Its architecture can be seen in Figure 1 of the original paper [17]. It is computationally slow because it individually processes each region proposal. To overcome this limitations, they developed Fast R-CNN processing the entire image once [18]. It uses a Region of Interest (RoI) Pooling layer to extract features from proposed regions for classification and bounding box refinement. Furthermore, Faster R-CNN [19] was introduced by Shaoqing Ren et al, proposing the Region Proposal Network (RPN) which is a fully CNN that shares features with the detection network for defining regions and objects in the regions. Faster R-CNN consists of two stages: first, the RPN proposes object regions, and then, the Fast R-CNN classifies these regions and fine-tunes the bounding boxes. This integration helps training become faster and more efficient making it suitable for real-time applications on modern GPUs. To handle different object sizes and shapes, Faster R-CNN uses predefined anchor boxes of multiple scales and aspect ratios. The RPN learns to regress these anchors to match the actual objects, increasing the model's flexibility and accuracy. Due to its robust architecture and high performance, Faster R-CNN has become the foundation of many practical object detection systems. An illustration of the original Faster R-CNN architecture is presented in Figure 2 of the paper [19] which is adopted as the following figure.

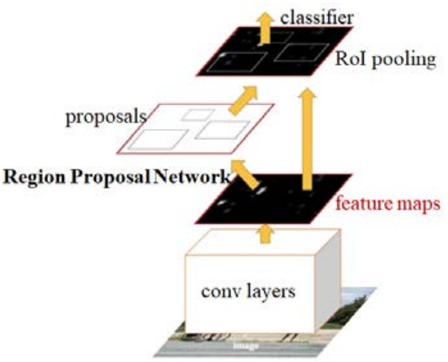

Fig. 2 The Faster R-CNN architecture adopted from [19].

3.2. Optical Character Recognition (OCR)

OCR extracts texts from documents, PDFs, and images, converting them into machine-readable formats for digital processing. Modern OCR systems apply pattern recognition, feature extraction, and DL to handle variations in fonts, handwriting, and text orientation [20]. The process starts with image capture, followed by preprocessing steps like noise removal, binarization, and segmentation to detect characters or words. Then, it classifies and recognizes characters based on visual traits like shape, stroke, and structural patterns.

OCR is widely used for digitizing printed contents, supporting real-time recognition, and enabling multilingual text extraction and translation. In real-time scenarios, such as ALPR, OCR processes live video from webcam to identify characters. OCR's accuracy depends on the quality of the input image and the effectiveness of preprocessing techniques. To enhance accuracy, advanced OCR systems also integrate language models to incorporate grammatical and contextual information. However, OCR still faces limitations when processing images with poor resolution, noisy backgrounds, complex formatting, or unconventional fonts [21].

4. METHODOLOGY

This section presents the methodology of the proposed multilingual ALPR system, which is developed using a rapid prototyping approach. This system involves data collection, image preprocessing, license plate detection and character recognition using OCR, analysis of misclassified characters, post-processing to refine OCR outputs. Each component is designed and improved based on rapid prototyping principles, enabling a fast development cycle for testing, evaluation, and enhancement to optimize ALPR performance. The following figure shows an overview of the proposed approach.

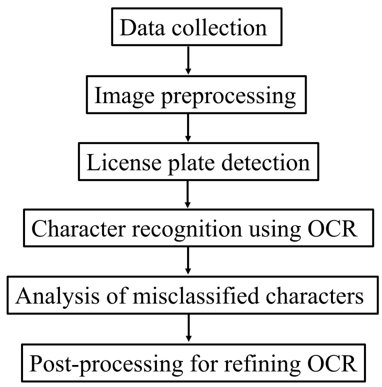

Fig. 3 Structure of the proposed approach.

4.1. Data collection and image pre-processing

To develop and evaluate our system, we collect a dataset consisting of 227 car images with Thai license plates. They are obtained from publicly available open-source platforms and web searches such as Google[3], Kaggle[4], and Roboflow[5]. To annotate the license plate locations in images, we use the LabelImg[6] tool to manually draw bounding boxes around each license plate. The

[2] https://docs.ultralytics.com/

[3] https://www.google.com/

[4] https://www.kaggle.com/

[5] https://roboflow.com/

[6] https://pypi.org/project/labelImg

tool generates corresponding XML files containing the bounding box coordinates and labels, forming the ground truth dataset required for training our detection model. Dataset contains some images having multiple vehicles in same image so the generated XML includes multiple bounding boxes for that image. Figure 4 illustrates the annotation process using the LabelImg tool, where a bounding box is drawn to mark the license plate location.

Fig. 4 License plate annotation using LabelImg with bounding box saved in XML format.

After collecting data, we apply a series of image pre-processing techniques to enhance the quality of images, such as re-sizing, random horizontal flipping, color jittering with saturation adjustment and normalization of pixel values. They augment and help our detection model generalize under dynamic situations.

4.2. License plate detection

We develop a License Plate Detection (LPD) model to predict bounding boxes around license plates, using the custom dataset described in the previous section. The model is implemented with the PyTorch-based Detecto package[7], which is built on a pre-trained Faster R-CNN ResNet-50 with Feature Pyramid Network (FPN) architecture. Leveraging transfer learning, we fine-tune the pre-trained model on our annotated dataset, training it for 25 epochs with a learning rate of 0.001 and a step size of 5 for learning rate scheduling. After training, we set a confidence score threshold of 90% during prediction to enhance detection precision, as evaluated by Intersection over Union (IoU). The model retains bounding box predictions with IoU is greather than or equal to 0.9, effectively filtering out uncertain detections. Figure 5 illustrates IoU, where the ground truth is in green, the prediction in red, and the blue mask shows an IoU of 0.9959.

Fig. 5 Illustration of Intersection over Union (IoU).

4.3. Character recognition using OCR

License plate images detected by the LPD system, as described in the previous section, are subsequently processed by an OCR model. In this study, we utilize EasyOCR for recognizing Thai characters and numbers on the license plates. The accuracy of character recognition depends on both the input image quality and the effectiveness of the pre-processing pipeline. Therefore, prior to OCR application, image quality is enhanced using standard pre-processing techniques, including re-sizing, gray-scale conversion, noise reduction and morphological operations such as dilation and erosion. These techniques improve the clarity and structure of the characters to facilitate accurate recognition. Figure 6 presents examples of character recognition results using EasyOCR. However, misclassifications in character recognition still occur, particularly under challenging visual conditions. These errors negatively impact the overall performance of the system. In the next section, we conduct an analysis of these misclassifications by the OCR model.

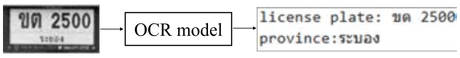

Fig. 6 License plate recognition by OCR.

4.4. Analysis of misclassifications and post-processing

We analyze the misclassifications made by the OCR model (EasyOCR) in recognizing characters on Thai license plates based on three main categories: visual distortion, character similarity, and plate orientation. Misclassifications often occur when visually similar Thai characters are degraded by blur and low resolution. In particular, side-view images pose significant challenges to the OCR model's ability to accurately interpret characters. These challenges include skewed or non-horizontal text alignment, perspective distortion, and uneven character scaling, all of which disrupt the model's ability to correctly segment and classify individual characters. Additionally, poor lighting conditions further reduce the clarity of character boundaries. Another error arises when license plate borders are mistakenly recognized as characters due to overlapping features or angular distortions that confuse the OCR model during inference. Figure 7 presents samples of Thai license plates undermining OCR accuracy and a side-view image with distorted text orientation leads to character misclassification is shown in Figure 8. Table 1 provides a summary of the misclassifications observed in EasyOCR's recognition performance on our custom dataset of Thai license plates.

Fig. 7 Samples of Thai license plates with blur, side-view distortion, and stroke-level confusion.

Fig. 8 Distorted text orientation of sample license plates.

Since OCR accuracy is degraded by skewed text orientation, motion blur and low resolution, we describe post-processing strategies to refine OCR in character recognition on Thai license plates, as followings.

- **Post-OCR Processing:** This process includes character filtering, format-based rules, and error pattern analysis to refine the results and correct common misrecognitions.

[7] https://detecto.readthedocs.io/en/latest/

Table 1 Misclassifications on Thai license plates: R_{ec} = recognized character, F_{req} = frequency, V_c = visual confusion, S_s = stroke similarity, S_v = side-view, O_r = orientation error, B_i = border interference.

Actual	Error		Reason				
	R_{ec}	F_{req}	V_c	S_s	S_v	O_r	B_i
ภ	]]	1	✓				
ภ	ก	1	✓	✓			
ส	สิ	1	✓	✓			
ก	า	1	✓				
ว	า	1	✓	✓			
ฉ	า	1	✓				
บ	ใ	1	✓				
บ	ข	1	✓	✓	✓		
ม	บ	1	✓	✓	✓		
ห	ท	1	✓	✓		✓	
ณ	0!	1	✓				
ช	ข	1	✓	✓			
ฆ	๗ ฝ ญ @	1 1 1 2	✓			✓	
ฝ	ม น	1 1	✓			✓	
ฌ	เม	1	✓	✓		✓	
Border	|	1					✓

- **Text Region Adjustment:** Techniques such as cropping, padding, and border removal are applied to eliminate unnecessary elements like plate frames.
- **Automatic Skew Correction:** The skew angle of the text is estimated, and the text region is rotated to achieve horizontal alignment for accurate OCR interpretation.
- **Image Quality Enhancement:** To address motion blur and low resolution, techniques such as noise reduction, contrast adjustment, and deblurring are used to improve the clarity of character strokes in the image.

5. EXPERIMENTAL RESULTS

We developed the LPD model using transfer learning based on the pre-trained Faster R-CNN with a ResNet-50 wih FPN architecture. We trained our model using two inputs: (1)images containing Thai license plates, and (2)ground-truth bounding boxes corresponding to each license plate in the image. Our custom dataset of 227 annotated car images was collected, featuring a variety of Thai license plates under different conditions. This dataset was divided into 80% for training (182 images) and 20% for testing (45 images). The model was trained for 25 epochs becoming the total loss steadily decreased and converged to approximately 0.24, as shown in Figure 9.

During inference, the model predicts bounding boxes for detected license plates. Since these predictions are not always a perfect match with the ground-truth annotations, we assess detection quality using the IoU metric. To minimize false positives where a license plate is incorrectly detected or mislocalized, we set a strict IoU threshold of 0.9. Thus, only predictions with an IoU above this threshold are accepted as correct detections. Figure 10 shows example outputs of license plate detection using the LPD model. After setting the threshold, the model successfully localizes two license plates insides corresponding red bounding boxes within a single image.

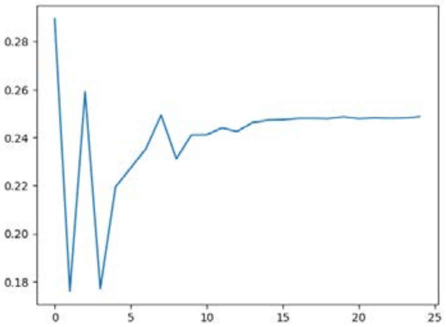

Fig. 9 Training loss curve over 25 epochs.

Fig. 10 Detected license plates in a car image.

After localizing the license plates using the LPD model, we cropped the detected regions and applied the cropped images to EasyOCR. In this experiment, we primarily focus on recognizing the license plate number, including the alphabetical prefix and numeric sequence, while excluding the province names from evaluation. The OCR model correctly recognized 15 out of 47 license plates. The skewed orientation of many license plates significantly impairs the OCR model's ability to accurately recognize characters. It results in an overall accuracy of 32%, which is considerably low and highlights the need for post-processing refinement. Therefore, we specifically investigate post-OCR processing technique that involves reordering the recognized characters to conform to the standard Thai license plate format, consisting of an alphabetical prefix followed by a numeric sequence and, optionally, the province name. Hence, the number of correctly recognized license plates increases to 22 out of 47, getting the accuracy to 47%. To assess its effectiveness, we compare it with PyTesseract, which achieved 17% accuracy (8 out of 47) on the same test set. The dataset includes blurred, angled, low-resolution plates with unclear strokes, posing challenges for general OCR engines. These results emphasize the need for our proposed post-processing strategies to address OCR misclassifications. Thus, the remaining strategies, such as text region adjustment, automatic skew correction, and image quality enhancement, will be explored in future work.

6. CONCLUSION

The effectiveness of an ALPR system depends on both accurate license plate detection and reliable OCR-based character recognition. Using a fine-tuned Faster R-CNN ResNet-50 model, our object detection system demonstrated high precision in extracting the license plate regions with IoU $\geq$ 0.9. However, OCR performance using EasyOCR on Thai license plates is initially limited, achieving only 32% accuracy due to issues such as skewed text orientation. To improve recognition, we apply post-OCR processing to reorder characters into the standard Thai license plate format. This refinement increased accuracy to 47%, demonstrating the importance of post-processing in OCR-based systems. In future work, we will evaluate the enhanced post-processing techniques, including text region adjustment, automatic skew correction, and image quality enhancement. These methods aim to address OCR failures resulting from plate misalignment, poor lighting, or motion blur. Furthermore, we will perform a comprehensive mAP evaluation across multiple IoU thresholds, comparing our approach with alternative OCR engines. We will also expand our dataset to include a wider range of Thai license plates captured under varied conditions. Thus, further improvements in recognition accuracy will be examined through these enhanced techniques and expanded datasets.

ACKNOWLEDGMENT

This work was financially supported by King Mongkut's Institute of Technology Ladkrabang Research Fund: KREF046820.

REFERENCES

[1] Y. Ogiuchi, M. Higashikubo, S. Panwai, and E. Luenagvilai, "Automatic license plate detection and recognition in thailand," *SEI technical review*, vol. 78, 2014.

[2] W. Puarungroj and N. Boonsirisumpun, "Thai license plate recognition based on deep learning," *Procedia Computer Science*, vol. 135, 2018.

[3] S. Kraisin and N. Kaothanthong, "Accuracy improvement of a province name recognition on thai license plate," in *2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*. IEEE, 2018.

[4] S. Rattanawong, G.-S. Hsu, and S.-L. Chung, "Thailand license plate detection and recognition," in *2021 25th international computer science and engineering conference (ICSEC)*. IEEE, 2021.

[5] M. Alwateer, K. O. Aljuhani, A. Shaqrah, R. ElAgamy, G. Elmarhomy, and E.-S. Atlam, "Xai-salpad: explainable deep learning techniques for saudi arabia license plate automatic detection," *Alexandria Engineering Journal*, vol. 109, 2024.

[6] A. Akoushideh, S. S. Sadat, and A. Shahbahrami, "Detecting, identifying, and counting vehicles based on deep learning algorithms in video surveillance systems," *Journal of Engineering and Applied Research*, vol. 1, no. 2, 2024.

[7] S.-R. Wang, H.-Y. Shih, Z.-Y. Shen, and W.-K. Tai, "End-to-end high accuracy license plate recognition based on depthwise separable convolution networks," *arXiv preprint arXiv:2202.10277*, 2022.

[8] M. Shahidi Zandi and R. Rajabi, "Deep learning based framework for iranian license plate detection and recognition," *Multimedia Tools and Applications*, vol. 81, no. 11, 2022.

[9] I. Shafi, I. Hussain, J. Ahmad, P. W. Kim, G. S. Choi, I. Ashraf, and S. Din, "License plate identification and recognition in a non-standard environment using neural pattern matching," *Complex & Intelligent Systems*, vol. 8, no. 5, 2022.

[10] D. M. Izidio, A. P. Ferreira, H. R. Medeiros, and E. N. d. S. Barros, "An embedded automatic license plate recognition system using deep learning," *Design Automation for Embedded Systems*, vol. 24, no. 1, 2020.

[11] C.-J. Lin, C.-C. Chuang, and H.-Y. Lin, "Edge-ai-based real-time automated license plate recognition system," *Applied Sciences*, vol. 12, no. 3, 2022.

[12] S. Bansal, A. Jain, M. Sharma, G. Kumar, S. Ojha, and H. Walia, "Enhancing alpr: a two stage yolo model with data augmentation for improved accuracy and robustness," *Multimedia Tools and Applications*, 2024.

[13] X. Zhai, F. Bensaali, and R. Sotudeh, "Real-time optical character recognition on field programmable gate array for automatic number plate recognition system," *IET Circuits, Devices & Systems*, vol. 7, no. 6, 2013.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, 2004.

[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. IEEE, 2005.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014.

[18] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[20] S. Wang, G.-J. Qi, and H. Li, "A comprehensive overview of optical character recognition technology," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 6, 2021.

[21] R. Smith, "An overview of the tesseract ocr engine," *Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR)*, 2007.