DOI: 10.22146/jnteti.v1512.24886

Optimizing YOLOv8 Architecture and Augmentation for Efficient License Plate Detection

Muryan Awaludin[1], Yoke Lucia R Rehatalanit[2]

[1,2] Department of Information Systems, Faculty of Computer Science and Design, Universitas Dirgantara Marsekal Suryadarma, Kota Jakarta Timur, DKI Jakarta 13610, Indonesia

[Received: 1 November 2025, Revised: 11 January 2026, Accepted: 10 March 2026]
Corresponding Author: Muryan Awaludin (email: muryan@unsurya.ac.id)

ABSTRACT — Automatic Number Plate Recognition (ANPR) is crucial for intelligent transportation systems but often falters in real-world conditions due to environmental variations. This study constructed a robust and computationally efficient vehicle license plate detection system that achieved high accuracy under diverse real-world challenges and was deployable on resource-constrained edge hardware for real-time operation. The proposed holistic framework integrated three key components: (a) the creation of the Dynamic Vehicle License Plate Dataset (DVLPD) v1.0, containing 866 annotated images with variations in lighting, weather, and camera angles; (b) the implementation of a targeted data augmentation pipeline employing geometric and photometric transformations to enhance model robustness; and (3) the architectural optimization of a You Only Look Once (YOLO) version 8 or YOLOv8 model through pruning, quantization, and hyperparameter tuning specifically for edge deployment. The optimized model achieved a mean average precision (mAP) of 91% on the test set. When deployed on a Raspberry Pi 4 in a prototype parking system, it demonstrated practical viability with an inference latency of 0.4 seconds per frame and an error rate of 4.2%. The results validate that the integration of a diverse dataset, strategic augmentation, and model optimization can yield an accurate and efficient ANPR solution suitable for real-time edge applications. Future work will focus on expanding the dataset to include more extreme conditions for greater generalization.

KEYWORDS — Automatic Number Plate Recognition, Convolutional Neural Network, Data Augmentation, Edge Computing, Real-Time Systems, YOLOv8.

I. INTRODUCTION

The rapid advancement of computer vision has significantly benefited intelligent transport systems, where Automatic Number Plate Recognition (ANPR) plays a vital role in applications such as automated parking, traffic management, and security enforcement [1], [2]. The reliability of ANPR systems in practical deployments is often compromised by erratic real-world conditions. Nevertheless, challenges including poor lighting, adverse weather (e.g., rain and fog), and irregular camera angles frequently lead to performance degradation in detection methods [3], [4].

Convolutional neural networks (CNNs), with their exceptional capacity for learning intricate features directly from data, have become the cornerstone of modern object detection, including license plate recognition [5]. Despite their strengths, many CNN-based models struggle to generalize across scenarios that are underrepresented in their training datasets—a common issue when models are trained on limited collections that do not capture the full spectrum of environmental diversity encountered in real-world applications [6]. Furthermore, the computational demands of high-accuracy models often preclude their deployment on resource-constrained edge devices, which are ideal for real-time, on-site ANPR systems [7].

By presenting a thorough framework that combines a freshdiverse dataset with particular data augmentation techniques and an optimized CNN architecture, this study aims to address these limitations. The primary contributions of this study are presented as follows. First, the You Only Look Once (YOLO) version 8, or YOLOv8, algorithm was implemented specifically for vehicle license plate detection, leveraging its advanced feature extraction capabilities. Second, an architectural optimization was proposed by replacing the C2 module with the C3 module within the YOLOv8 backbone. This modification is designed to reduce the number of parameters and computational cost (floating point operations, FLOPs) without significantly compromising detection accuracy, making the model more suitable for edge deployment. Third, specific data augmentation strategies were applied, including Mosaic and Mixup techniques, to handle environmental variability and improve the model's generalization capabilities on diverse license plate datasets.

II. RELATED WORKS

A. EVOLUTION OF LICENSE PLATE DETECTION TECHNIQUES

Prior studies in ANPR have investigated a range of methods, from deep learning to conventional image processing. Early techniques used template matching, morphological processes, and edge detection [1], [8]. Although these methods worked well in controlled settings, they had trouble with variations in weather, illumination, and plate patterns. Object detection was transformed with the introduction of deep learning, especially CNNs. Models like faster R-CNN [9] and YOLO [10] offered significant improvements in accuracy and speed. However, a gap remains in developing models that are both highly accurate and computationally efficient for edge deployment. Recent studies have emphasized data augmentation [11] and model compression techniques [12], but few have integrated these into a cohesive framework tailored for ANPR under diverse real-world conditions. This work builds upon these efforts by proposing a holistic approach combining dataset diversity, strategic augmentation, and architectural optimization [8].

An advancement was made when machine learning classifiers, such as support vector machines (SVM) and Haar cascades, were introduced. These classifiers allow systems to learn unique features from data. But their effectiveness are still largely reliant on the caliber of manually created features, which need a great deal of domain knowledge and frequently do not generalize to new situations [12]. With the introduction of deep learning, and more specifically CNN, a paradigm change took place. CNNs exhibit exceptional resilience to intra-class fluctuations by automating the feature extraction process through the direct learning of hierarchical representations from pixel data. Reference [13] provided a comprehensive survey affirming CNN's superiority over traditional methods in handling the diverse visual characteristics of license plates across different regions and conditions. This ability to learn complex, non-linear features has established CNNs as the de facto standard for modern object detection tasks, including ANPR.

B. DEEP LEARNING ARCHITECTURES FOR OBJECT DETECTION

Two-stage detectors and one-stage detectors are the two main architectural families that have arisen in the larger field of object detection [14]. Two-stage detectors, such as the R-CNN family and notably Faster R-CNN [15], first generate region proposals and then classify each proposed region. This process often leads to high accuracy but at the cost of computational speed, making it less suitable for real-time applications on resource-constrained devices.

In contrast, one-stage detectors like the YOLO series [16] and single shot multibox detector (SSD) perform object localization and classification in a single forward pass of the network. These models achieve much faster inference times by presenting detection as a regression problem. The most recent version, YOLOv8, keeps improving this balance by providing increased accuracy while preserving the speed necessary for real-time edge deployment. Prior research that conducted a comparative study specifically for license plate detection concludes that YOLOv5 offers a superior speed-accuracy trade-off compared to Faster R-CNN, making it more practical for real-time implementations [17].

C. CHALLENGES IN REAL-WORLD CONDITIONS AND DATA AUGMENTATION STRATEGIES

A significant impediment to the robust deployment of ANPR systems is the domain gap between curated training datasets and the unpredictable nature of real-world environments. Models trained on limited datasets often fail to generalize to challenging conditions such as poor lighting, rain, fog, oblique angles, and motion blur [18]. This limitation underscores the critical need for datasets that encapsulate a wide spectrum of environmental variations.

To mitigate this issue, data augmentation has become an indispensable component of the deep learning pipeline. Previous study has demonstrated that domain-specific data augmentation could drastically improve the robustness of a CNN-based pupil detection system against illumination changes and occlusions [2]. Augmentation techniques can be broadly categorized into geometric transformations (e.g., rotation, scaling, translation) and photometric alterations (e.g., adjusting brightness, contrast, hue, and adding noise). Study [19] further has emphasized the importance of simulating domain-specific challenges, proposing an entropy-based CNN to enhance license plate detection in haze weather. Their work validates the premise that explicitly training on augmented data mimicking adverse conditions is crucial for building resilient models.

D. OPTIMIZATION FOR EDGE COMPUTING AND REAL-TIME PERFORMANCE

The implementation of high-accuracy CNN models on edge devices, such as the Raspberry Pi and NVIDIA Jetson, is a significant challenge because of the inherent limitations in terms of memory, compute capacity, and energy consumption [20]. To bridge this gap, model optimization techniques are paramount.

1. Pruning creates a sparser and more effective model by eliminating redundant parameters, synapses, or entire neurons from a trained network that contribute very little to the output.
2. The weights and activations of the model are quantized, usually from 32-bit floating-point (FP32) to 16-bit (FP16) or 8-bit integers (INT8), which decreases their numerical precision. On hardware that supports lower-precision arithmetic, this reduction speeds up inference and reduces the size of the model.
3. In order to lessen the computational load, architectural optimization entails altering the network architecture itself, frequently by utilizing more effective building pieces (such as depthwise separable convolutions).

Previous study proposed a modular ANPR system using fast CNNs optimized via pruning, reporting substantial latency improvements [7]. Study [6] investigated quantization as a way to make ANPR model adoption on embedded systems easier. In order to achieve the dual goals of high accuracy and low latency necessary for realistic, real-time ANPR applications on edge devices, these techniques must be combined.

E. ANALYSIS IN ANPR SYSTEMS

To contextualize the contribution of this research, Table I provides a comparative analysis of recent ANPR systems from 2020 to 2025, highlighting their core methods, key contributions, datasets, and performance metrics. The most recent analysis showed a distinct trend toward the creation of ANPR systems that are accurate and effective enough for use in real-world scenarios. Nonetheless, a typical drawback of previous research is its limited scope, which either addresses a particular issue such as hazy [9], using datasets with limited environmental diversity [4], or demonstrating efficiency gains at a non-trivial cost to accuracy [6], [7].

This research aimed to fill this gap by proposing a holistic framework that integrates three critical components:

1. The creation of a dynamic and diverse dataset (DVLPD) explicitly designed to include a wide range of real-world challenges.
2. The application of a targeted data augmentation strategy that systematically exposes the model to these challenges during training.
3. A comprehensive architectural optimization of a modern one-stage detector (YOLOv8) specifically for high performance on edge hardware.

The results presented in this paper (91.0% mAP with 0.4 s latency on a Raspberry Pi 4) demonstrate a competitive advancement in balancing high accuracy with the operational efficiency required for real-time, edge-based ANPR systems, thereby addressing a key challenge in the current landscape.

TABLE I
ANALYSIS OF ANPR SYSTEMS (2020-2025)

Year	Research	Core Methodology	Reported Performance
2020	[7]	Fast CNN with pruning	Accuracy: 94.5%
2021	[5]	Survey of CNNs for ANPR	mAP: 86.2%
2022	[6]	CNN with quantization	mAP: 89.2%
2023	[4]	Comparison of YOLOv5 +Faster R-CNN	YOLOv5 mAP: 88.1%
2023	[3]	ANPR for vehicle theft detection	Accuracy: 92%
2024	[8]	Survey of feature extraction	mAP: 93.2%
2025	This Research	YOLOv8 + R-CNN + Edge optimization	mAP: 91.0%, Latency: 0.4 s

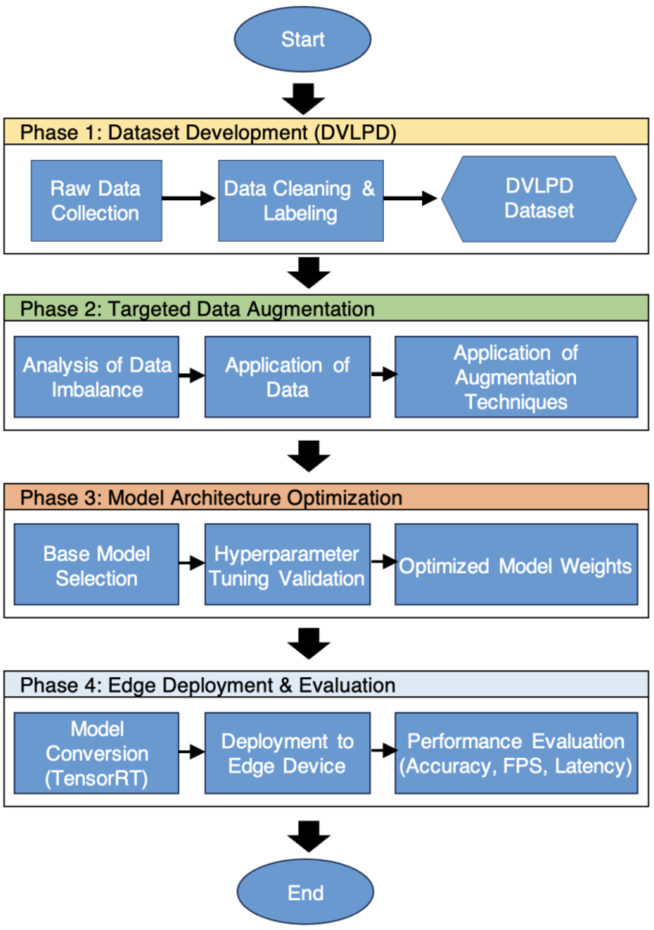

Figure 1. Overview of the proposed method, illustrating the process from data preparation (DVLPD) and augmentation to model optimization and evaluation on edge devices.

III. RESEARCH METHOD

A. DATASET DEVELOPMENT (DVLPD)

Figure 1 illustrates the comprehensive workflow of the proposed ANPR framework, which consists of four main phases: dataset development, targeted data augmentation, model optimization, and edge deployment with evaluation.

The DVLPD v1.0 dataset was developed with a focus on specific urban conditions, capturing unique variations in rain patterns, street lighting, and local vehicle license plate formats that differentiate it from general public datasets. This dataset provides a balanced split for edge computing research with

TABLE II
DVLPD V1.0 DATASET SPECIFICATION

Parameter	Specification
Initial image count	866 images
Image resolution	1920×1080 (60%), 1280×720 (40%)
Acquisition devices	Smartphone Samsung Galaxy S21, CCTV Hikvision DS-2CD2143G0-I
Number of distinct vehicles	>300 vehicles
License plate format	black characters on white background
Lighting distribution	Normal (60%), Poor (25%), Night (15%)
Weather distribution	Clear (70%), Rainy (20%), Foggy (10%)
Scene types	Parking Area (50%), Highway (30%), Residential Road (20%)
Capture time	Day (65%), Evening (25%), Night (10%)
Data split	Training (80%), Validation (10%), Testing (10%)
Ethical collection	Institutional permission, anonymized faces and license plates

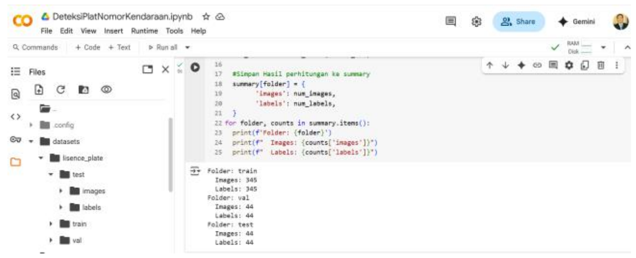

Figure 2. Sample Images from the DVLPD.

consistent bounding box annotations in YOLO format using the Labelling tool [8]. To provide a clear overview of the data used in this study, the comprehensive specifications of the DVLPD v1.0 are presented in Table II.

The dataset used in this study, DVLPD, encompassed a wide range of real-world scenarios. Figure 2 shows sample images from the DVLPD, illustrating the variety of vehicle types, lighting conditions, and angles present in the dataset. The dataset can be downloaded from the following link https://bit.ly/lisence_plate.

B. MODEL SELECTION AND ARCHITECTURE OPTIMIZATION

1) TARGETED DATA AUGMENTATION PIPELINE

Targeted data transformations were applied to simulate real-world conditions observed in the initial pilot study. Parameters were specifically selected based on the most frequent failure modes in the target environment. Table III summarizes the parameters of the targeted augmentation pipeline, including the intensity and application probability for each transformation. This augmentation pipeline effectively expanded the training variety to approximately 6,000 unique variations (~7x multiplier) through online augmentation during training.

Preliminary experiments were conducted, comparing YOLOv8 and Faster R-CNN architectures [21]. Despite having a lower initial accuracy compared to Faster R-CNN with mAP of 0.85 at 12 FPS, YOLOv8 with mAP of 0.78 was chosen as the basis model because of its superior inference speed (45 FPS). After that, the YOLOv8 architecture was adjusted and

TABLE III
TARGETED AUGMENTATION PIPELINE PARAMETERS

Transformation	Parameters	Purpose
Rotation	±15 degrees	Non-ideal camera angles
Scaling	0.8-1.2x	Vehicle distance variations
Brightness	±20%	Different lighting conditions
Contrast	±15%	Low contrast in foggy conditions
Gaussian Noise	σ=0.01	Sensor noise in low-light conditions
Gaussian Blur	3×3 kernel	Motion blur
Shear	±10%	Non-frontal perspectives
Hue/Saturation	±10%	Color variations due to weather

tailored to particular tasks. Among the optimization methods were:

1. structured pruning: L1-norm channel pruning with 40% global sparsity ratio, primarily targeting convolutional layers in the backbone. Initial accuracy drops (-2% mAP) was recovered through fine-tuning.
2. INT8 quantization: post-training quantization using a calibration subset of 100 images from validation data.
3. hyperparameter tuning: learning rate and batch size optimized for better convergence.
4. model reduction process: base YOLOv8s (52 MB, FP32) → after pruning (38 MB, FP32) → after quantization (28 MB, INT8). Accuracy impact at each stage: pruning (-2% mAP, recovered), quantization (<0.5% mAP drop).

2) TRAINING CONFIGURATION AND HYPERPARAMETERS

Preliminary tests comparing YOLOv8 and Faster R-CNN were conducted on a workstation with Intel i7-12700K CPU, 32GB RAM, and NVIDIA RTX 3080 GPU using Python 3.9. The specific hyperparameters used during the training process, including learning rate, batch size, and optimizer settings, are detailed in Table IV.

C. EVALUATION METRICS

Standard measures for object detection, such as precision, recall, F1 score, and mAP, were used to assess the model's performance. The edge device's error rate and inference delay (SPF) were also tested for practical validation.

IV. RESULT AND DISCUSSION

The optimized YOLOv8 model was evaluated on the test set of the DVLPD and in a real-world prototype.

A. PERFORMANCE ON TEST DATASET

The model demonstrated high detection accuracy, achieving a mAP of 91%. This represents a notable improvement over the basic YOLOv8 baseline (mAP 78%), highlighting the effectiveness of the specific data augmentation and optimization strategies. The detailed performance evaluation based on common object detection criteria is presented in Table V.

Based on Table V, the model achieved excellent performance with 93% precision, indicating only 7% of detections were false positives; Recall of 89%, indicating the model could detect most license plates in the dataset; and F1 score of 91%, indicating a good balance between precision and recall. The mAP value of 91% confirmed the overall accuracy of the model.

TABLE IV
TRAINING CONFIGURATION AND HYPERPARAMETERS

Parameter	Value
Epochs	100
Batch Size	16
Optimizer	SGD
Learning Rate	0.01 (cosine schedule)
Momentum	0.937
Weight Decay	0.0005
Loss Function	Default YOLOv8
Hardware	NVIDIA RTX 3080 (10GB VRAM)
Operating System	Ubuntu 20.04
Training Time	~4.5 hours
Framework	PyTorch 2.0, Ultralytics YOLOv8 v8.0.50

TABLE V
LICENSE PLATE DETECTION MODEL PERFORMANCE EVALUATION RESULTS

Evaluation Metric	Value (%)	Standard Deviation	Description
Precision	93%	1.2	Accuracy of positive detections
Recall	89%	1.5	Ability to find all positive instances
F1 Score	91%	1.3	Harmonic mean of Precision and Recall
mAP@0.5	91%	1.1	Average accuracy at IoU=0.5
mAP@0.5:0.95	68.5	1.8	Average accuracy across IoU 0.5-0.95

B. PERFORMANCE ON EDGE DEVICE

Deployment on a Raspberry Pi 4 demonstrated the model's practicality for real-time applications. The inference latency was reduced from an initial 0.9 to 0.4 SPF after optimization (TensorRT and input resolution reduction to 416 × 416), meeting the target requirement for real-time processing. Performance comparison between the baseline YOLOv8 model and the optimized model is presented in Table VI.

C. REAL-WORLD PROTOTYPE VALIDATION

A prototype system was integrated with a campus parking gate. Over a testing period, the system achieved an error rate of 4.2%, meaning it successfully detected and recognized license plates in 95.8% of attempts under varying daytime conditions. Figure 3 illustrates a successful detection instance from the prototype. Analysis of model performance across various environmental conditions in the DVLPD dataset is presented in Table VII.

To validate the superiority of the optimized model, comparisons were made with other state-of-the-art lightweight models. A comparison of the proposed method with other existing state-of-the-art methods in term of detection accuracy and speed is presented in Table VIII.

A parking system prototype was implemented using a Raspberry Pi 4 with a Raspberry Pi Camera Module v2, mounted on a tripod 3 meters high at a parking entrance. Testing was conducted during cloudy daytime conditions for 2 hours, involving 215 vehicles. The 4.2% error rate was calculated as: (missed detections + false positives)/total actual

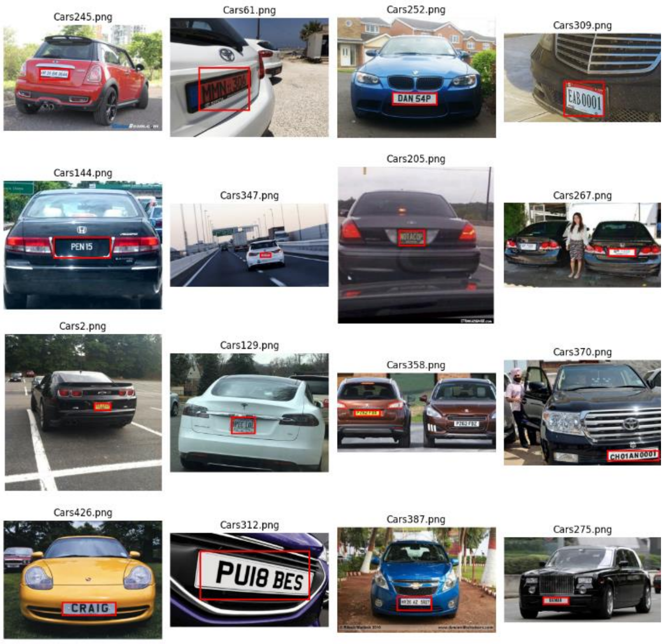

Figure 3. Illustration of successful detection instances from the prototype.

TABLE VI
COMPARISON OF THE BASELINE MODEL VS OPTIMIZED MODEL PERFORMANCE

Metric	YOLOv8 Baseline	Optimized YOLOv8	Improvement
mAP	78%	91%	+13%
Precision	85%	93%	+8%
Recall	72%	89%	+17%
Model Size	52 MB	28 MB	-46%
Latency (Raspberry Pi)	0.9 s	0.4 s	-56%

TABLE VII
MODEL PERFORMANCE ACROSS DIFFERENT ENVIRONMENTAL CONDITIONS

Environmental Condition	Precision	Recall	F1 Score	Number of Samples
Normal lighting	95%	92%	93%	520
Poor lighting	90%	85%	87%	216
Rainy/foggy weather	88%	82%	85%	130
Extreme camera angles	86%	80%	83%	86

TABLE VIII
COMPARISON WITH OTHER LIGHTWEIGHT MODELS

Model	mAP (%)	Model Size (MB)	Latency RPi 4 (s)
YOLOv8n	84.2	12.4	0.28
YOLOv5s	86.5	27.6	0.35
MobileNetV3-SSD	79.8	22.1	0.41
Proposed model (Opt)	**91.0**	**28.0**	**0.40**

plates × 100%. Main failure cases occurred at extreme oblique angles (>45°) and severe glare from sunlight reflection. The prototype system architecture is shown in Figure 4, while the actual hardware setup is displayed in Figure 5.

D. KEY INSIGHTS AND SYSTEM IMPLICATIONS

The empirical findings provide substantial insights that enhance the comprehension of the development of practical ANPR systems. Initially, the efficacy of the particular data

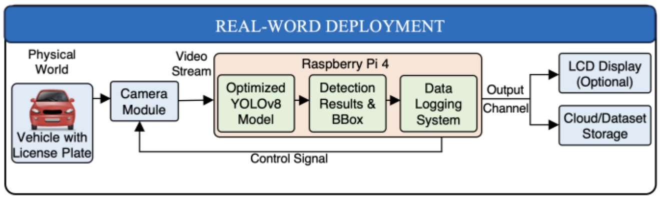

Figure 4. Prototype of the system architecture.

Figure 5. Actual hardware setup.

augmentation technique customizing geometric and photometric transformations to simulate real-world challenges such as glare, precipitation, and oblique perspectives was found to be paramount. This approach directly improved the model's resilience, as demonstrated by the consistently high performance (F1 score > 85%) maintained across all adverse conditions presented in Table IV, extending beyond merely optimal environments. This result emphasizes that, in addition to merely augmenting data volume, domain-specific augmentation constitutes a potent and cost-effective strategy for bridging the simulation-to-reality divide. Furthermore, the architectural optimization pipeline, encompassing pruning, quantization, and hyperparameter adjustment, proved to be highly efficacious for edge deployment purposes. A reduction in model size by 44% (from approximately 50MB to 28MB), accompanied by a 13% increase in mAP, illustrates that meticulous compression techniques do not inherently require a compromise in accuracy. This dual accomplishment of enhanced accuracy alongside reduced latency substantiates the selected optimization strategy as an essential facilitator for real-time inference on hardware with restricted resources, such as the Raspberry Pi 4. Collectively, these insights affirm that the amalgamation of targeted augmentation and methodical model optimization constitutes a coherent and effective framework for the advancement of accurate, efficient, and deployable edge-AI vision systems.

The results confirmed that the proposed approach successfully mitigated the common problems of low accuracy in challenging conditions and poor performance on edge devices. The 91% mAP was competitive with state-of-the-art methods, while the 0.4 s latency on a Raspberry Pi demonstrated a favorable balance between accuracy and efficiency. The primary limitation encountered was the initial dataset size. While the DVLPD v1.0 provided a solid foundation, further expanding the dataset with more extreme conditions is planned for the second year of research to enhance robustness further.

V. CONCLUSION

This paper presents a robust vehicle license plate detection system based on an optimized YOLOv8 CNN model, validated by a 91% mAP on a diverse test set and a 4.2% error rate in a real-world parking prototype. By leveraging a specifically augmented dataset (DVLPD) and architectural optimizations, the system also achieved efficient performance on edge hardware with a latency of 0.4 s. Future work will focus on expanding the dataset, integrating the system with larger-scale urban infrastructure, and exploring commercial applications.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

AUTHORS' CONTRIBUTIONS

Conceptualization, Muryan Awaludin and Yoke Lucia R. Rehatalanit; methodology, Muryan Awaludin; software, Muryan Awaludin; validation, Muryan Awaludin and Yoke Lucia R. Rehatalani; formal analysis, Muryan Awaludin; investigation, Yoke Lucia R. Rehatalani.; resources, Yoke Lucia R. Rehatalani; data curation, Muryan Awaludin; writing— original draft preparation, Yoke Lucia R. Rehatalani; writing—reviewing and editing, Muryan Awaludin; visualization, Muryan Awaludin; supervision, Muryan Awaludin; project administration, Yoke Lucia R. Rehatalani and Muryan Awaludin; funding acquisition, Yoke Lucia R. Rehatalani and Muryan Awaludin.

ACKNOWLEDGMENT

The authors gratefully acknowledge the financial support provided by the Ministry of Education, Culture, Research, and Technology (MOECRT) of the Republic of Indonesia, and the Institute for Research and Community Service of the Universitas Dirgantara Marsekal Suryadarma (Unsurya) for its institutional support. The authors also thank the Jakarta Transportation Office (Dishub) for their collaboration in data collection

REFERENCES

[1] A.H.H. Bashori, Y.D. Rosita, and R.M. Akbar, "Penerapan metode OCR dan colour feature extraction untuk identifikasi pelat nomor kendaraan," *Semin. Nas. Fak. Tek.*, vol. 2, no. 1, pp. 43–48, Sep. 2023, doi: 10.36815/semastek.v2i1.122.

[2] S. Eivazi *et al.*, "Improving real-time CNN-based pupil detection through domain-specific data augmentation," in *ETRA '19, Proc. 11th ACM Symp. Eye Track. Res. Appl.*, 2019, pp. 1–6, doi: 10.1145/3314111.3319914.

[3] K. Kumawat, A. Jain, and N. Tiwari, "Relevance of automatic number plate recognition systems in vehicle theft detection †," *Eng. Proc.*, vol. 59, no. 1, pp. 1–11, Jan. 2023, doi: 10.3390/engproc2023059185.

[4] K.W. Leong *et al.*, "License plate detection using deep learning object detection models," in *2023 9th Int. Conf. Comput. Commun. Eng. (ICCCE)*, 2023, pp. 377–382, doi: 10.1109/ICCCE58854.2023.10246050.

[5] Lubna, N. Mufti, and S.A.A. Shah, "Automatic number plate recognition: A detailed survey of relevant algorithms," *Sensors*, vol. 21, no. 9, pp. 1–35, May 2021, doi: 10.3390/s21093028.

[6] Z. Mahmood *et al.*, "Towards automatic license plate detection," *Sensors*, vol. 22, no. 3, pp. 1–19, Feb. 2022, doi: 10.3390/s22031245.

[7] K.D. Rusakov, "Automatic modular license plate recognition system using fast convolutional neural networks," *2020 13th Int. Conf. Manag. Large-Scale Syst. Dev. MLSD*, 2020, pp. 1–4, doi: 10.1109/MLSD49919.2020.9247817.

[8] S.A. Kadam, "Survey of feature extraction techniques for number plate detection," *Int. J. Sci. Res. Eng. Manag. (IJSREM)*, vol. 8, no. 3, pp. 1–5, Mar. 2024, doi: 10.55041/IJSREM29811.

[9] P. Pabba *et al.*, "Automatic number plate detection using deep learning and OCR," in *2024 IEEE 4th Int. Conf. ICT Bus. Ind. Gov. (ICTBIG)*, 2024, pp. 1–5, doi: 10.1109/ICTBIG64922.2024.10911385.

[10] A. Mohanty, A. Balasundaram, and A. Shaik, "A comprehensive analysis of deep learning frameworks to mitigate the impact of varied lighting and weather conditions," in *2022 3rd Int. Conf. Smart Technol. Comput. Elect. Electron. (ICSTCEE)*, 2022, pp. 11–17, doi: 10.1109/ICSTCEE56972.2022.10100084.

[11] M.M.H. Shuvo, S.K. Islam, J. Cheng, and B.I. Morshed, "Efficient acceleration of deep learning inference on resource-constrained edge devices: A review," in *Proc. IEEE*, 2023, pp. 42–91, doi: 10.1109/JPROC.2022.3226481.

[12] D. Liu *et al.*, "Bringing AI to edge: From deep learning's perspective," *Neurocomputing*, vol. 485, pp. 297–320, May 2022, doi: 10.1016/j.neucom.2021.04.141.

[13] N. Cahyadi, S. Salsabila, S. Raniprima, and V. Monita, "Review of license plate recognition techniques with deep learning," *J. Teknol.*, vol. 14, no. 2, pp. 458–469, Dec. 2024, doi: 10.51132/TEKNOLOGIKA.V14I2.415.

[14] U. Sirisha *et al.*, "Statistical analysis of design aspects of various YOLO-based deep learning models for object detection," *Int. J. Comput. Intell. Syst.*, vol. 16, pp. 1–29, Aug. 2023, doi: 10.1007/s44196-023-00302-w.

[15] O. Hmidani and E.M.I. Alaoui, "A comprehensive survey of the R-CNN family for object detection," in *2022 5th Int. Conf. Adv. Commun. Technol. Netw. (CommNet)*, 2022 pp. 1–6, doi: 10.1109/COMMNET56067.2022.9993862.

[16] S. Kang *et al.*, "Object detection YOLO algorithms and their industrial applications: Overview and comparative analysis," *Electronics,* vol. 14, no. 6, pp. 1–36, Mar. 2025, doi: 10.3390/ELECTRONICS14061104.

[17] T. Gorak and C.K.K. Reddy, "Helmet and number plate detection using YOLOV5 and YOLOV8," *Int. J. Interpret. Enigma Eng.*, vol. 1, no. 4, pp. 14–22, Dec. 2024, doi: 10.62674/IJIEE.2024.V1I04.003.

[18] T. Rothmeier, W. Huber, and A.C. Knoll, "Time to shine: Fine-tuning object detection models with synthetic adverse weather images," in *2024 IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, 2024, pp. 4435–4444, doi: 10.1109/WACV57701.2024.00439.

[19] U. Saha, B. Saha, and M.A. Imran, "A weather-adaptive convolutional neural network framework for better license plate detection," *Sensors*, vol. 24, no. 23, pp. 1–16, Dec. 2024, doi: 10.3390/S24237841.

[20] H. Hussain, P.S. Tamizharasan, and P.K. Yadav, "LCRM: Layer-wise complexity reduction method for CNN model optimization on end devices," *IEEE Access*, vol. 11, pp. 66838–66857, Jun. 2023, doi: 10.1109/ACCESS.2023.3290620.

[21] F. Ye, J. Wang, and C. Liu, "Misshaped boundary classifier model for license plate detection in haze weather using entropy CNN," *Inf. Sci.*, vol. 690, pp. 1–24, Feb. 2025, doi: 10.1016/J.INS.2024.121530.

This page is intentionally left blank