Dual Edge TPU를 이용한 실시간 휴대용 자동차 번호판 인식기 성능향상에 관한 연구

박시형[1], 김지섭[1], 강인경[1], 정영락[1]
[1]동서울대학교 컴퓨터 소프트웨어과 학부생
qkrtlgud0620@naver.com, wltjq6702@gmail.com, ingyeong7161@gmail.com, wjddudfkr127@gmail.com

A Study on the Performance Improvement of a Real-Time Portable Vehicle License Plate Recognition System Using Dual Edge TPU

Si-Hyeong Park[1], Ji-Seob Kim[1], In-Gyeong Kang[1], Young-Rak Jeung[1]
[1]Dept. of Computer Software, Dong-Seoul University

Abstract

This study proposes an enhanced real-time portable license plate recognition system by replacing the YOLOv8n detection model with EfficientDet_Lite0 and upgrading the AI accelerator from a USB Coral Edge TPU to a PCIe-based Dual Edge TPU. The proposed pipeline detects license plates using a custom-trained EfficientDet_Lite0 and recognizes characters using LPRNet. Experimental results show that recognition accuracy is comparable to the previous approach, while average processing speed increases from 18.53 FPS to 45.60 FPS. Additionally, the total model size is reduced from 13.6 MB to 6.6 MB, achieving approximately a 2.5-fold improvement in both speed and model size.

1. Introduction

The existing Raspberry Pi 5-based real-time portable license plate recognition system demonstrated the feasibility of real-time inference using a low-power, lightweight platform. Prior studies achieved an average processing speed of 18.53 FPS using the YOLOv8n model with a USB Coral Edge TPU, but faced limitations in recognizing plates of high-speed vehicles [1]. In this study, we replaced the YOLOv8n detection model with EfficientDet_Lite0 and upgraded the AI accelerator from a USB Coral Edge TPU to a PCIe-based Dual Edge TPU, thereby enhancing real-time performance in terms of speed, efficiency, and model size.

2. LPRNet (License Plate Recognition Network) model

LPRNet is an end-to-end and lightweight deep learning model that significantly improves inference speed by removing complex preprocessing and post-processing steps, compared to traditional OCR methods [2]. Its compact architecture makes it well-suited for on-device AI applications.

3. EfficientDet_Lite0 model

EfficientDet_Lite0 is a lightweight detection model optimized for mobile environments and based on Google's EfficientDet series. It achieves competitive detection performance while maintaining a compact model size of approximately 5.6 MB by utilizing BiFPN-based feature fusion and a compound scaling strategy [3]. In this study, EfficientDet_Lite0 was employed for the license plate detection stage. Figure 1 illustrates the overall architecture of the EfficientDet_Lite0 model.

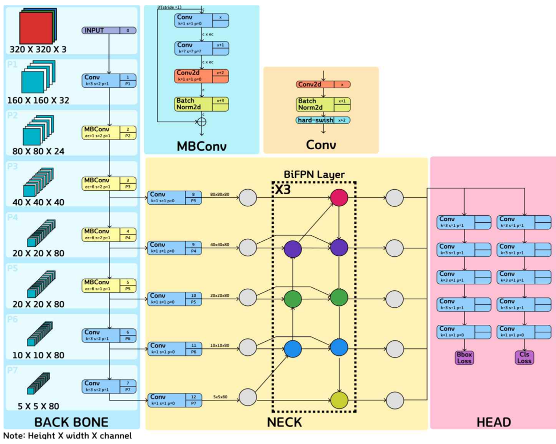


Figure 1. EfficientDet_Lite0 model architecture

4. Model Development

In this study, we utilized the "Vehicle Type/Model Year/License Plate Recognition Video Data" provided by the Ministry of Science and ICT and the National Information Society Agency (NIA). For the detection stage, the EfficientDet_Lite0 model was transfer-learned on Korean license plate data, quantized to 8-bit integers, and converted to the TensorFlow Lite (TFLite) format, resulting in improved computational efficiency over the YOLOv8n model. For character recognition, the LPRNet model pre-trained in previous work was converted from TensorFlow (.pb) to TensorFlow Lite (.tflite) format.

5. Experimental Setup and Results

Table 1 summarizes the experimental setup used in this study.

Table 1. Experimental Environment Configuration

Category	Item		Description
Raspberry Pi 5	H/W	CPU	Broadcom BCM2712
		RAM	8GB LPDDR4X-4267
	S/W		OS==Debian12(BookWorm) Python == 3.9.12, 3.11.2 opencv-python == 4.10.0.84 torch==2.0.1, torchvision==0.15.2
Dual Edge TPU	Performance		8TOPS
	S/W		edge-tpu-silva == 1.0.4
	power consumption		4W
Camera Module 3	S/W		libcamera == 0.4.0 rpicam-apps == 1.6.0
Touch Display	Size		7 inch
Power Supply	Battery		USB PD (5V/2.4A)
	PD Trigger		5V 5A mode

Table 2 summarizes the performance evaluation metrics used in the experiment.

Table 2. Performance Evaluation Metrics

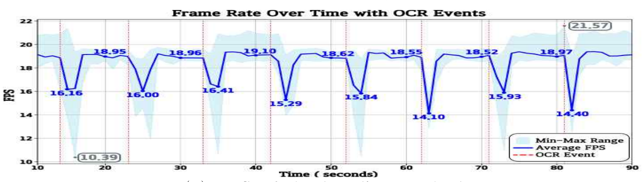
Metric	Description
mAP (Mean Average Precision)	Performance of the object detection model was evaluated using 100 test images.
CER (Character Error Rate)	Character Error Rate (CER) of LPRNet was measured using 5,667 test images as a metric for inference performance.
EMA (Exact Match Accuracy)	Exact Match Accuracy of LPRNet was measured using 5,667 test images as a metric for inference performance.
FPS (Frame Per Second)	Average FPS was measured as a metric for processing speed by running the entire pipeline 10 times for 90 seconds.

Table 3 summarizes the performance and model size of the detection and recognition networks.

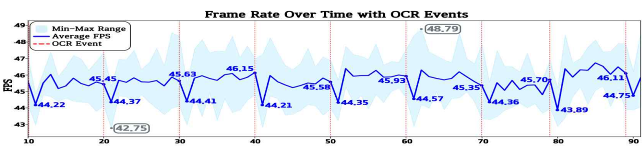
Table 3. Comparison of Performance and Model Size for Detection and Recognition Models

Module	Method	Model size	Metric	
			mAP@50-95↑	
Detection	Baseline	3.4 Mbyte	0.715	
	Proposed	5.6 Mbyte	0.876	
Recognition			CER↓	EMA↑
	Baseline	10.2 Mbyte	0.0024	0.9834
	Proposed	1 Mbyte	0.0029	0.9811

The detection module achieved a performance improvement of 0.16, while the recognition module showed comparable performance to the existing method. The total model size was reduced from 13.6 MB to 6.6 MB in the proposed method. Figure 2 compares throughput: the proposed pipeline sustained 42.75 - 48.79 FPS, averaging 45.60 FPS, while the baseline sustained 14.10 - 21.57 FPS, averaging 18.53 FPS.



(a) FPS of the baseline Method



(b) FPS of the Proposed Method

Figure 2. Comparison of FPS Between the Baseline and Proposed Methods

Figure 3 illustrates the overall system architecture of the proposed method. In a Python 3.11 environment, images captured by Camera Module 3 are stored frame-by-frame using a multi-buffering technique. The frames are then merged in a Python 3.9 environment to reconstruct the video; afterwards, the detection module locates license plates and the recognition module reads the detected plates, displaying the results on the touch display.

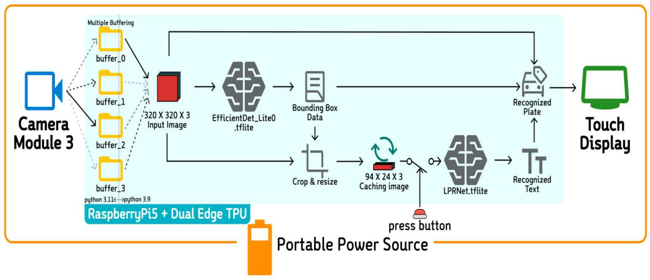


Figure 3. Overall System Architecture of the Proposed Method

Figure 4 displays the real-time operation screen of the proposed portable license plate recognition system.

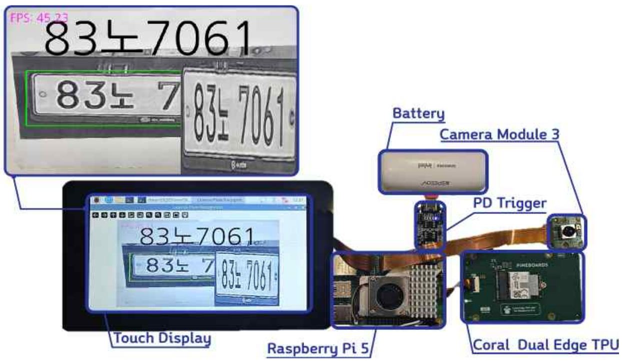


Figure 4. Operation Screen of the Proposed Real-Time Portable License Plate Recognition System

6. Conclusion

This study improved the average processing speed of the existing USB Coral Edge TPU-based system from 18.53 FPS to 45.60 FPS—approximately 2.5 times faster—by using the Dual Edge TPU and EfficientDet_Lite0 model. The total model size was also reduced from 13.6 MB to 6.6 MB, achieving lightweight optimization. These results demonstrate the system's potential for high-speed object detection and recognition in edge computing environments.

REFERENCES

[1] 김지섭, 강인경, 박시형, 이재희, "Edge Computer(Raspberry Pi5)를 이용한 실시간 휴대용 자동차 번호판 인식기 개발에 관한 연구", 한국정보처리학회 ASK 2025 학술발표대회 논문집, 2025년 5월

[2] Sergey Zherzdev, Alexey Gruzdev, "LPRNet: License Plate Recognition via Deep Neural Networks," arXiv preprint arXiv:1806.10447, 2018

[3] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," arXiv preprint arXiv:1911.09070, 2020.