Design and Implementation of a MobileNet-Based YOLO Object Detection Model for Resource-Constrained Devices

1st Seungtae Hong
On-Device System Software Research Section
Electronics and Telecommunications Research Institute
and
University of Science and Technology
Daejeon, Korea
sthong@etri.re.kr

2nd Gunju Park
On-Device System Software Research Section
Electronics and Telecommunications Research Institute
Daejeon, Korea
parkgj@etri.re.kr

3rd Jeong-Si Kim
On-Device System Software Research Section
Electronics and Telecommunications Research Institute
Daejeon, Korea
sikim00@etri.re.kr

***Abstract*— In this paper, we propose a lightweight object detection model optimized for resource-constrained environments by replacing the backbone of YOLO v8 with MobileNet v2. While conventional YOLO models achieve high detection accuracy, they require high-resolution input images and substantial computational resources, making them unsuitable for real-time deployment on embedded or mobile devices. To address this, the proposed model adopts MobileNet v2 as a lightweight backbone and introduces a connector module to integrate it with YOLO v8's neck and head. We also apply channel-reduction techniques to further minimize the model's complexity. The model is trained and evaluated using the COCO dataset, with a single-class setup focusing on the person class and an input resolution of 352×352 pixels. Experimental results show that the proposed model reduces the number of parameters by approximately 17% compared to the original YOLO v8, while achieving slightly improved detection accuracy. These results demonstrate that the proposed architecture achieves a favorable trade-off between computational efficiency and detection performance, making it suitable for real-time object detection on devices with limited resources.**

Keywords—Resource-Constrained Devices, Deep Learning, YOLO, Object Detection

I. Introduction

With the rapid advancement of object detection technologies, deep learning-based detection models centered around YOLO (You Only Look Once) have been actively adopted in both academic research and industry applications [1]. YOLO has been widely used in diverse domains such as autonomous driving, security systems, video surveillance, and smart homes due to its outstanding real-time detection performance. However, YOLO models generally require high-resolution input images (e.g., 640×640 pixels), making it difficult to achieve real-time processing on devices with severely limited memory and computational resources. As a result, there is a growing need for lightweight models that can deliver high performance while maintaining real-time capabilities in embedded and mobile environments with resource constraints.

In certain applications, however, high-resolution input may not be essential. For example, when detecting relatively large objects in simple backgrounds, high detection accuracy can still be achieved with low-resolution input. Conventional YOLO models are designed for high-resolution images and are not optimized for low-resolution environments, potentially leading to inefficient resource utilization.

To address this issue, this paper proposes a lightweight object detection model that adopts MobileNet v2 [2] as the backbone network of YOLO. The key idea is to replace YOLO's existing backbone with the lightweight MobileNet v2 while maintaining the original head structure of YOLO to preserve both compatibility and detection performance. MobileNet v2 is a lightweight architecture well-suited for low-resolution images, and in this study, the input image resolution is reduced to 352×352 pixels. Additionally, the pre-trained weights from MobileNet v2 (trained on ImageNet) are used for initialization to improve training efficiency and model performance.

II. Related Works

YOLO, first introduced in 2016, has continuously evolved as one of the most prominent object detection frameworks due to its fast inference speed and stable accuracy. The initial YOLO v1 [3] model gained significant attention for its ability to detect objects in real-time using a single CNN that processes the entire image at once. Since then, numerous improved versions have been proposed to enhance detection accuracy and robustness.

Recently, more advanced versions such as YOLO v8 [4], YOLO v11 [4], and YOLO v12 [5] have been introduced, achieving strong performance on large-scale benchmarks. In particular, YOLO v12 demonstrated significant improvements in accuracy through its complex architecture and high computational load. However, the increased demand for computational resources makes it less suitable for lightweight or real-time environments. Similarly, YOLO v11 combined multi-scale processing and attention modules to improve detection, but suffered from slower inference speed and increased memory consumption.

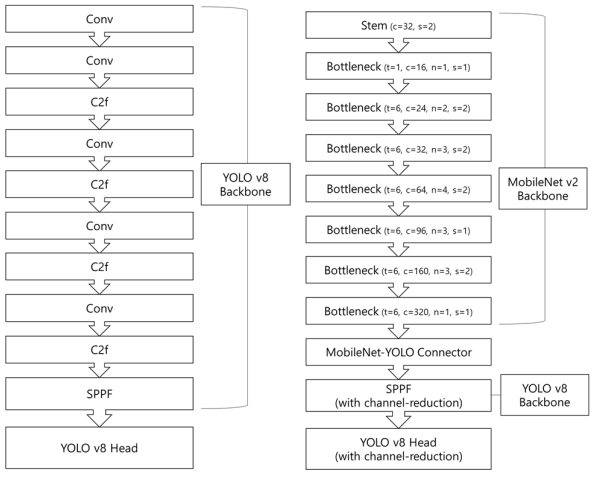
In this context, YOLO v8 remains a practical and balanced model in terms of accuracy and inference speed. YOLO v8 features a well-organized backbone, neck, and head, supports lightweight customization, and is highly compatible with various deep learning frameworks. Moreover, its availability in multiple scales (nano, small, medium, large, xlarge) makes it adaptable to a wide range of application scenarios.

III. MobileNet-Based YOLO Object Detection Model

While conventional YOLO-based models demonstrate excellent performance in terms of accuracy and speed, they are primarily designed for high-resolution images (640×640 pixels or higher). This design poses a challenge for real-time deployment on small devices with extremely limited memory and compute capabilities. Devices such as IoT systems,

drones, and wearable gadgets face significant difficulty in utilizing such high-resolution-based models efficiently.

The proposed lightweight YOLO model replaces the backbone of YOLO v8 with MobileNet v2, significantly reducing system resource requirements. MobileNet v2 utilizes depthwise separable convolution to minimize the model size and computational load while preserving performance. Its ability to extract rich features from low-resolution images makes it a strong candidate for the target input resolution of 352×352 pixels used in this study.



(a) Original YOLO v8 model (b) Proposed YOLO v8 model

Fig. 1. Comparison of YOLO v8 architectures

Figure 1 compares the original YOLOv8 architecture (a) with the proposed MobileNet v2-based YOLOv8 architecture (b). The entire backbone of YOLO v8, up to the final C2f block, is replaced with MobileNet v2. To align the output channels of MobileNet v2's final bottleneck block with the SPPF module in YOLO's neck, a MobileNet-YOLO Connector is introduced. This connector, built using convolutional layers, enables smooth integration between MobileNet v2 and YOLOv8's neck and head. Additionally, given the use of low-resolution inputs (352×352 pixels), channel-reduction techniques are applied to the SPPF and head sections of YOLO v8 to further reduce computational and memory overhead. Through this architecture, the model leverages MobileNet v2's lightweight characteristics while preserving YOLO v8's real-time detection capabilities.

IV. Experiments

To quantitatively evaluate the proposed model's efficiency and detection accuracy, experiments were conducted using the COCO dataset [6]. Instead of using the entire multi-class dataset, only the person class was extracted, reformulating the task as a single-class object detection problem. This allowed for a clearer analysis of the structural impact on detection performance.

For fairness, both the original YOLO v8 and the proposed MobileNet v2-based YOLO v8 were trained under identical settings: input resolution of 352×352 pixels, 50 training epochs, and the same data splitting method. Notably, the proposed model initialized its backbone using pre-trained weights from ImageNet, enabling faster convergence and improved performance.

Table 1 presents a comparison of parameter and gradient counts between the original and proposed models. By replacing YOLOv8's backbone with MobileNet v2, the total number of parameters is reduced by approximately 17.3%.

TABLE I. Model Structure Comparison

Model	Parameters	Gradients
Original YOLO v8	3,011,043	3,011,027
Proposed MobileNet-YOLO v8	2,489,523	2,489,507

Table 2 shows the detection performance under the same training conditions. The evaluation metrics include mAP50 and mAP50-95. Despite having fewer parameters, the proposed model achieved slightly higher accuracy on both metrics. These results confirm that MobileNet v2's lightweight architecture is highly effective in extracting features from low-resolution inputs, and that using pre-trained weights enhances learning efficiency and accuracy.

TABLE II. Detection Performance Comparison (Single Class: Person)

Model	mAP50	mAP50-95
Original YOLO v8	0.642	0.409
Proposed MobileNet-YOLO v8	0.652	0.419

V. Conclusion

In this paper, we proposed a lightweight YOLO model tailored for real-time object detection on resource-constrained devices by replacing the backbone of YOLO v8 with MobileNet v2. The proposed model retains YOLO v8's neck and head structures while significantly reducing complexity through the introduction of MobileNet v2, a MobileNet-YOLO connector, and channel-reduction techniques.

Experiments were conducted using the COCO dataset with the person class, under a reduced input resolution of 352×352 pixels. Results showed that the proposed model reduced parameter count by approximately 17%, while slightly improving detection accuracy (mAP50 and mAP50-95) compared to the original YOLO v8. This demonstrates that the proposed architecture achieves a favorable balance between resource efficiency and detection performance.

This study addresses the inefficiency of applying high-resolution-centric object detection models to real-world applications, particularly on embedded and mobile devices. It experimentally presents an architecture suitable for low-resolution, real-time object detection tasks.

Future work will explore generalization across diverse object classes and complex backgrounds. Additional metrics such as inference speed (FPS), computational cost (FLOPs), and energy efficiency will also be considered to further develop a practical lightweight object detection system..

Acknowledgment

This work was supported by Institute of Information & co mmunications Technology Planning & Evaluation(IITP) gran t funded by the Korea government (MSIT)(No.RS-2024-003

39187,Core Technology Development of On-device Robot Intelligence SW Platform)

REFERENCES

[1] V. Mohankumar, and A. Sasithradevi, "A benchmark dataset and ensemble YOLO method for enhanced underwater fish detection," ETRI Journal, 2025.

[2] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," Proceedings of the IEEE conference on computer vision and pattern recognition, 2018.

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

[4] Ultratics, Ultratics YOLO https://github.com/ultralytics/ultralytics

[5] Y. Tian, Y. Qixiang, and D. Doermann, "Yolov12: Attention-centric real-time object detectors," arXiv preprint arXiv:2502.12524, 2025.

[6] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," European conference on computer vision. Cham: Springer International Publishing, 2014.