YOLO real-time object detection on EV3-Robot using FPGA hardware Accelerator

Dinda Pramanta
Kyushu Institute of Information Sciences, 6-3-1, Saifu, Dazaifu, 818-0117, Japan
E-mail: pramanta@kiis.ac.jp

Ninnart Fuengfusin, Arie Rachmad Syulistyo, Hakaru Tamukoh
Kyushu Institute of Technology, 2-4 Hibikino, Wakamatsu-ku, Kitakyushu, 808-0196, Japan
E-mail: ninnart@brain.kyutech.ac.jp, syulistyo.arie-rachmad788@mail.kyutech.jp, tamukoh@brain.kyutech.ac.jp

Abstract

The growing demand for robots necessitates faster and more precise processing. However, running large Artificial Intelligence (AI) models from cloud data centers to mobile robots via inference models uses considerable computation resources, which leads to power limitations, particularly for mobile robots. The use of reconfigurable semiconductor devices at the hardware level is a promising solution to this problem. We introduce the educational kit EV3-Robot with a co-design methodology utilizing Field-programmable Gate Arrays (FPGA) Kria KV260 as a hardware accelerator specifically for object detection. We apply the You Only Look Once (YOLO) model for object detection, which provides real-time results for practical applications. Additionally, we analyze the processing times of the local PC and EV3-Robot.

Keywords: EV3-Robot, FPGA, Object detection, YOLO.

1. Introduction

Japan has an ageing population and a decreasing number of young people [1]. To address this issue, mobile robots have gained attention [2]. For example, robots are expected to perform daily indoor tasks, such as picking up and delivering objects. To accomplish such tasks, robots must be capable of recognizing and detecting specific objects [3].

In the field of Artificial Intelligence (AI), Neural Networks (NN) have been extensively researched [4], [5], [6]. Neural networks (NN) are equipped with state-of-the-art perceptual abilities, adaptive learning, and sophisticated human interaction. However, processing this information requires fast and accurate computation, especially in mobile robots where real-time interaction is almost unavoidable. This highlights the increasing need for robots equipped with reliable NNs. In contrast, You Only Look Once (YOLO) has gained attention for its fast and high accuracy [7].

Reconfigurable semiconductor devices at the hardware level offer a promising solution to the challenge of running large YOLO models from cloud to mobile robots via inference models, which require significant computation resources and result in low power consumption. We present the educational kit EV3-Robot, which utilizes a co-design methodology that employs local-PC and Field-programmable Gate Arrays (FPGA) Kria KV260 as a hardware accelerator for the YOLO model.

In our previous study on hardware-based communication [8], [9], the communication processes played a pivotal role. The first step was to design the client-server communication model, where the EV3-Robot served as the server and the local PC as the client. Next, we set the feedback for motor movements during specific tasks, in this case, using Ev3dev Python Socket Connection via Bluetooth to be able detect humans as objects. After that, we execute the deep-learning processor unit (DPU) of the YOLO model on the KV260 board. Finally, we evaluate the processing time.

2. Methodology

2.1. *You Only Look Once version 3 (YOLOv3) as detection system*

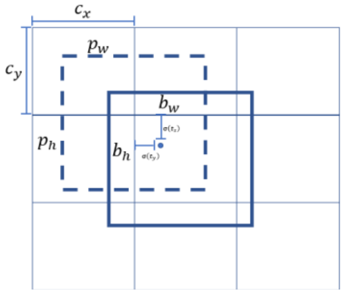


Fig.1 Bounding box [10]

Fig. 1 illustrates how YOLOv3 predicts the anchor box by using the bounding box and size clustering. The feature map cell is represented by c_x , c_y, and the pre-selected bounding box size is p_w , p_h . The predicted coordinates are b_x, b_y, b_w, b_h which can be calculated through Eq. (1), (2), (3), and (4) where the $\sigma(t_x)$ and $\sigma(t_y)$ are pixel values.

$$b_x = \sigma(t_x) + c_x \quad (1)$$
$$b_y = \sigma(t_y) + c_y \quad (2)$$
$$b_w = p_w e^{t_w} \quad (3)$$
$$b_h = p_h e^{t_h} \quad (4)$$

2.2. *System architecture*

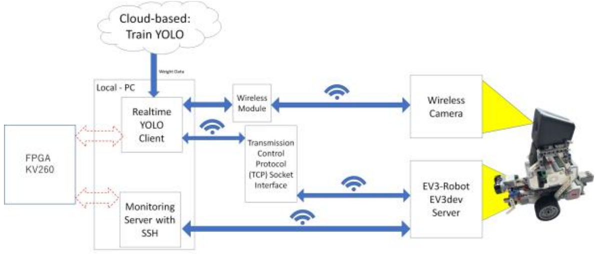
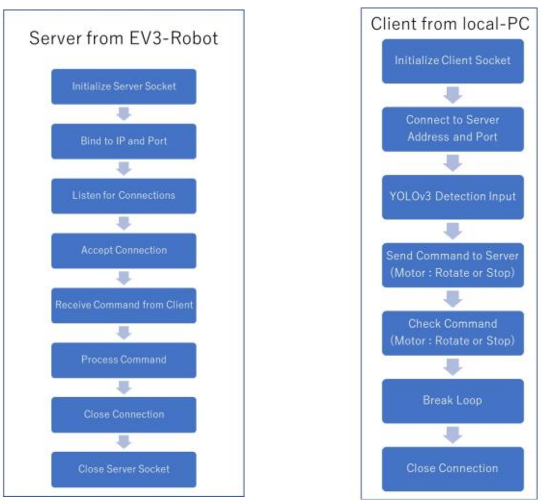


Fig. 2. YOLO real-time object detection on EV3-Robot using FPGA hardware accelerator.

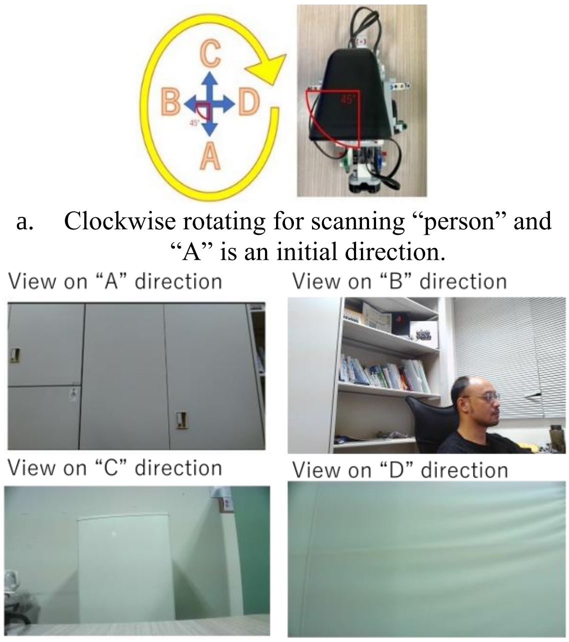
Fig. 2 displays a real-time implementation of YOLO for object detection using an EV3-Robot. The EV3-Robot is equipped with various sensors, although we exclude the input sensor functions in this system. The operating system (OS) used is ev3dev, which is Linux-based and stored on a Secure Digital (SD) card. The connection to the computer is established through Bluetooth and operates at 2.4GHz. The robot is powered by an ARM-9 processor and includes a built-in mini-LCD display. It has four input ports for sensor functions and four output ports for actuators or motors. The EV3-robots motor is connected using the output ports only for feedback.

2.3. *Ev3dev Python Socket Connections for* feedback



a. Server diagram flow b. Client diagram flow

Fig. 3. Client-server communication systems.



a. Clockwise rotating for scanning "person" and "A" is an initial direction.

b. Camera views on each direction

Fig. 4. EV3-Robot motor rotation

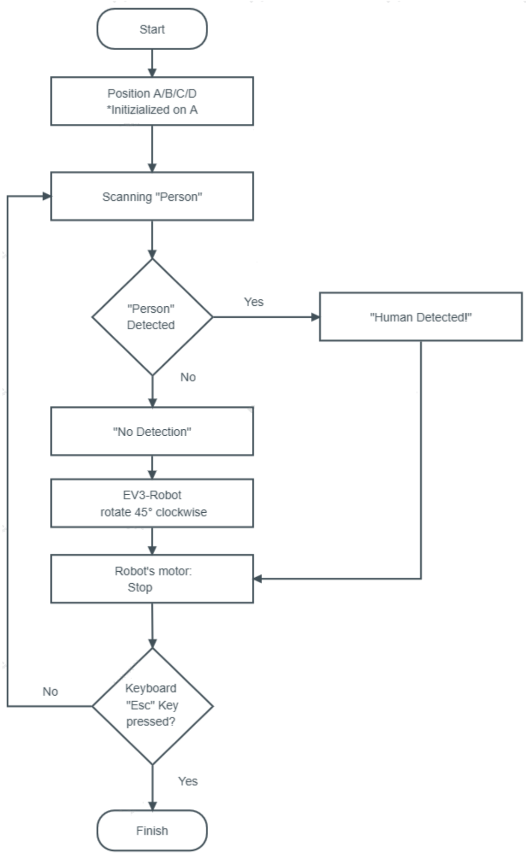


Fig. 5. Feedback sequences.

Fig. 3 illustrates a remote-control flow that utilizes python socket connections [11] to set motor movements as feedback. The communication involves the use of Internet Protocol (IP) addresses to establish connections. The local PC acts as a client and sends commands to an EV3-Robot, which serves as the server, enabling control over the robot.

Fig. 4 depicts a feedback design for detection that utilizes the motor functions of the EV3-Robot. The surrounding environment is scanned by setting up four directional points, denoted as “A”, “B”, “C”, and “D”. The direction is rotated 45° clockwise at regular intervals, in this case, every 4 seconds.

Fig. 5 displays the sequence of the motor rotation process. The robot is equipped with a standby mode, referred to as 'scanning for person', which enables it to scan while executing real-time object detection processes until the desired class, 'person', is detected. Finally, we have implemented an exit process triggered by the 'Esc' command from keyboard inputs to conclude the sequence.

2.4. *Getting the weight data from cloud to local-PC*

The pre-trained COCO weight dataset [12] was utilized. As we were testing real-time communication during detection, we only focused on the specified class from COCO, which is “person” (class_id = 0).

This was added to the client part and a real-time processing loop was performed for 'person' object detection. The wireless camera is mounted to perform object detection using the YOLOv3 model. It focuses on specific classes and determines whether a class_id = 0 (person) is detected. If so, it sends a message via the socket connection.

2.5. *Deep-learning Processor Unit (DPU) in KV260*

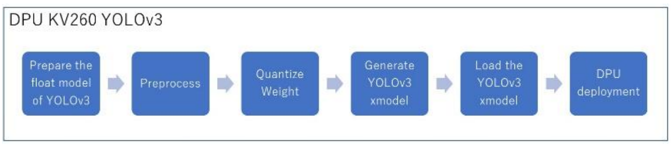


Fig. 6. DPU flow.

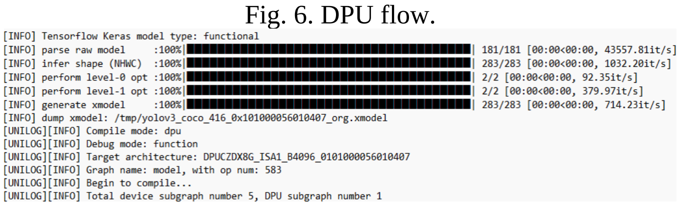


Fig. 7. YOLOv3 xmodel.

Fig. 6 displays the application of DPU and its integration into the KRIA KV260 board FPGA. Initially, we prepare the YOLOv3 model [13]. Next, we proceed to the pre-processing stage to install and verify the dependencies or libraries. Subsequently, we quantize the model from 32-bit floating-point weights to 8-bit.

Fig. 7 shows that we successfully generated the specified model for the KV260 architecture model, referred to as the “xmodel”. Finally, we load “xmodel” into DPU deployment for real-time hardware acceleration.

3. Results and Discussion

3.1. *Real-time remote-control communication of client-server systems*

Table 1. The latency results of the first four command keys.

Keyboard Key	Message	Latency (ms)
Up	“going UP! received”	522
Right	“going RIGHT! received”	652
Down	“going DOWN! received”	511
Left	“going LEFT! received”	661

Latency is a priority in client-server communications, especially for YOLO applications that require real-time interaction and responsiveness from the EV3-Robot. Achieving low-latency communication is crucial for current system, so we define latency as the round-trip time (RTT) divided by two in milliseconds (ms) as shown in (5).

$$Latency\ (ms) = \frac{RTT}{2} \tag{5}$$

The outcomes of the initial four command keys are displayed in Table1. Each key is associated with a confirmation message, allowing us to verify when the instruction was issued. Our findings indicate that the latency remains below $1{,}000\ ms$.

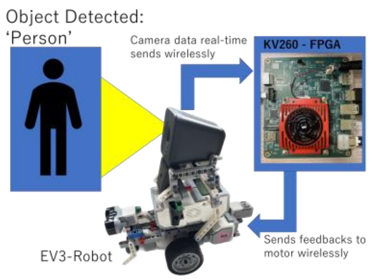
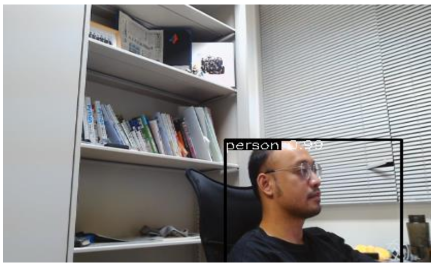


Fig.8. FPGA hardware accelerator task

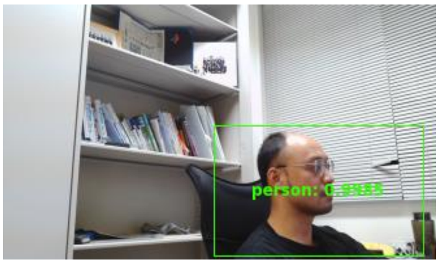
Table 2. Result comparison during N-step = 1.

Local-PC	FPS	Inference-time (*ms*)	Hardware Accelerator	"person" Object Detection
13th Gen Intel(R) Core (TM) i7-13700F 2.10 GHz.	2.63	270.3	None	Success
	3.78	264.4	FPGA KV260	Success

3.2. *Real-time visualizations*



a. Local-PC without hardware accelerator.



b. Local-PC with hardware accelerator FPGA.

Fig. 9. EV3-Robot real-time visualizations

Fig. 8 shows a task that FPGA hardware accelerator task. Fig. 9(a) and ~~Fig. 9(b)~~ are displaying the real-time visualization from the EV3-Robot's perspective. The class detection for 'person' was achieved in the "B" direction.

3.3. *FPGA hardware accelerator*

Table 2 compares the local-PC and EV3-Robot's performance during the detection task. The use of hardware accelerators resulted in an increase in the number of frames per second (FPS) and a reduction in inference time.

4. Conclusion

Real-time object detection on the EV3-Robot using FPGA hardware acceleration has been successfully implemented. Equipped with the FPGA Kria KV260 accelerator and the YOLO model, the EV3-Robot demonstrates a practical and effective solution for real-time object detection. This paves the way for enhanced performance and broader applications in the field of mobile robotics.

Acknowledgements

This research was supported by JSPS KAKENHI Grand Numbers 23H03468.

References

1. Coulmas, F., 2007. Population decline and ageing in Japan-the social consequences (Vol. 16). Routledge.
2. Yoshimoto, Y. and Tamukoh, H., 2021. FPGA implementation of a binarized dual stream convolutional neural network for service robots. Journal of Robotics and Mechatronics, 33(2), pp.386-399.
3. Hori, S., Ishida, Y., Kiyama, Y., Tanaka, Y., Kuroda, Y., Hisano, M., Imamura, Y., Himaki, T., Yoshimoto, Y., Aratani, Y. and Hashimoto, K., 2017. Hibikino-musashi@ home 2017 team description paper. arXiv preprint arXiv:1711.05457.
4. Hinton, G., 2018. Deep learning—a technology with the potential to transform health care. Jama, 320(11), pp.1101-1102.
5. LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. nature, 521(7553), pp.436-444.
6. Paugam-Moisy, H. and Bohte, S.M., 2012. Computing with spiking neuron networks. Handbook of natural computing, 1, pp.1-47.
7. Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).
8. Pramanta, D., Morie, T. and Tamukoh, H., 2017. Synchronization of Pulse-Coupled Phase Oscillators over Multi-FPGA Communication Links. J. Robotics Netw. Artif. Life, 4(1), pp.91-96.
9. Pramanta, D. and Tamukoh, H., 2019, December. High-speed synchronization of pulse-coupled phase oscillators on multi-FPGA. In International Conference on Neural Information Processing (pp. 318-329). Cham: Springer International Publishing.
10. Redmon, J. and Farhadi, A., 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.
11. R. Hempel and D. Lechner, "ev3dev," 2023. Available online: https://www.ev3dev.org/, accessed on 30 January 2023. [Online]. Available: https://www.ev3dev.org/
12. Lin, T.Y., et al, 2014. Microsoft coco: Common objects in context. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13 (pp. 740-755). Springer International Publishing.
13. Darknet model, accessed on 30 January 2023. [Online] Available: https://pjreddie.com/darknet/yolo

Authors Introduction

Dr. Dinda Pramanta

He earned his Bachelor's degree from Telkom University in 2013 and completed both Master's and Doctoral degrees at the Kyushu Institute of Technology in 2016 and 2020. After a post-doctoral fellowship with NEDO, Japan, from 2020 to 2021, he is currently an Assistant Professor at Kyushu Institute of Information Sciences since 2021. His research interests include neural networks, hardware, and AI for educational applications.

Dr. Ninnart Fuengfusin

He earned his B.Eng. degree from King Mongkut's University of Technology Thonburi, Thailand, in 2016, and his M.Eng. and D.Eng. degrees from Kyushu Institute of Technology, Japan, in 2018 and 2021, respectively. Currently a postdoctoral researcher at the Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology, his research focuses on deep learning, efficient neural network design, and digital hardware design.

Mr. Arie Rachmad Syulistyo

A member of IEEE, he earned his bachelor's degree in computer science from the University of Brawijaya, Indonesia, in 2011, and his master's degree from the University of Indonesia in 2015. Currently pursuing a Ph.D. in life science and system engineering at Kyushu Institute of Technology, Japan, his research interests encompass computer vision and machine learning.

Dr. Hakaru Tamukoh

A member of IEEE, he obtained his B.Eng. degree from Miyazaki University in 2001 and completed his M.Eng. and Ph.D. degrees at the Kyushu Institute of Technology in 2003 and 2006, respectively. After serving as a Postdoctoral Research Fellow, he became an Assistant Professor at the Tokyo University of Agriculture and Technology from 2007 to 2013. Currently a Professor at the Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology, his research interests encompass digital hardware design, soft computing, and home-service robots.