ISSN 0002-306X. Proc. of the RA NAS and NPUA Ser. of tech. sc. 2023. V. LXXVI, N3

UDC 621.382 **MICROELECTRONICS**

DOI: 10.53297/0002306X-2023.v76.3-343

M.T. GRIGORYAN

INVESTIGATING THE PERFORMANCE INDICES OF YOLO MODELS IMPLEMENTED ON A DPU: A COMPARATIVE ANALYSIS

A comprehensive analysis of the performance indices achieved by implementing various "You Only Look Once" dataset (YOLO) models on a dedicated Deep Learning Processing Unit (DPU). YOLO models are renowned for their real-time object detection capabilities, making them a good choice for a range of applications including autonomous vehicles, surveillance systems, and robotics. In this study, the YOLO models are deployed onto a specialized hardware accelerator, specifically a DPU, to assess their inference speed, accuracy, and power efficiency. By conducting an in-depth comparative evaluation of multiple YOLO variants, including YOLOv3, YOLOv4, YOLOv5, YOLOv6 insights into how each model interacts with the DPU architecture are revealed. The experiments involve benchmarking these models across diverse datasets and varying hardware configurations. The results not only highlight the advantages and limitations of employing DPUs for YOLO-based applications but also help in choosing the most suitable model-DPU combination based on specific performance requirements. This study contributes to the optimization of real-time object detection systems and assists practitioners in making informed decisions regarding the model and hardware selection.

Keywords: FPGA, DPU, object detectio, YOLO.

Introduction. In recent years, deep learning models have made remarkable advancements in computer vision tasks, particularly in real-time object detection. The YOLO architecture's ability to simultaneously predict object classes and bounding box coordinates in a single pass allows implementation of applications such as autonomous driving, surveillance, and interactive robotics.

As the demand for real-time processing in various domains continues to grow, the importance of optimizing the performance of YOLO models becomes paramount. Hardware acceleration has emerged to meet these demands, with DPUs offering dedicated resources for efficient neural network inference. DPUs are designed to accelerate the execution of deep learning workloads, reducing latency, and increasing throughput, making them an attractive option for deploying YOLO models in resource-constrained environments.

By conducting thorough experiments involving these models and varying DPU configurations, the work shows how the synergy between YOLO architectures

such as YOLOv3 [1], YOLOv4 [2], YOLOv5 [3], YOLOv6 [4] and DPUs influences overall system performance. As the fields of computer vision and deep learning converge, this study contributes to the ongoing efforts in optimizing real-time object detection systems, ultimately pushing the boundaries of what is achievable in terms of speed, accuracy, and efficiency.

Background. Real-time object detection requires low-latency processing to keep up with the high frame rates of video streams or live camera feeds. Achieving real-time performance is challenging due to the need for fast and accurate object localization and classification, often within tight time constraints. Efficient hardware architectures and algorithm optimizations are crucial to meet the demands of real-time applications.

Several research articles have explored these challenges and proposed solutions to enhance the object detection performance. For instance, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks" by Ren, et al [5] introduces the Faster R-CNN architecture that combines region proposal networks and CNNs for accurate and efficient object detection. "SSD: Single Shot MultiBox Detector" by Liu, et al [6] presents a single-shot detection method that achieves real-time performance by using multi-scale feature maps. "YOLO: Real-Time Object Detection" by Redmon, et al [7] introduces the YOLO framework, which achieves real-time object detection by dividing the image into a grid and predicting the bounding boxes and the class probabilities directly.

In the following sections, we will discuss the proposed design and FPGA implementation, aimed at addressing these challenges and provide an efficient hardware acceleration solution for object detection.

Approach. Dataset training and evaluation is first done on the GPU device. Then the float model is preprocessed and quantized by Xilinx Vitis AI tool. Quantization is a technique used to reduce the memory footprint and computational requirements of CNNs by presenting and performing computations with lower precision data types. In a standard CNN, weights and activations are typically presented as 32-bit floating-point numbers (FP32), which consume significant memory and require higher computational resources. Quantization aims to replace these higher precision representations with lower precision data types, such as fixed-point or integer values, which have fewer bits.

To capture activation statistics and improve the accuracy of quantized models, the Vitis AI quantizer needs to run several iterations of inference to calibrate the activations. A calibration image dataset input is therefore required. Generally, the quantizer works well with 100–1000 calibration images. This is because there is no need for back propagation, the un-labeled dataset is sufficient.

After calibration, the quantized model is transformed into a DPU deployable model which follows the data format of a DPU. This model can then be compiled by the Vitis AI compiler and deployed to the DPU.

The overall flow is illustrated in Fig.1.

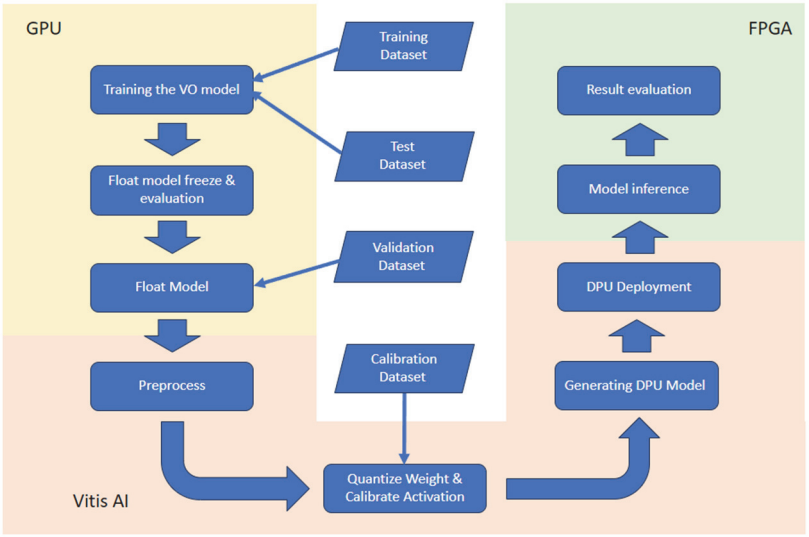

Fig.1. The DPU deployment flow

Results. This section presents the results of our experiments on testing YOLOv3, YOLOv4, YOLOv5, and YOLOv6 object detection models on the ZCU104 FPGA platform. The evaluation is conducted in terms of accuracy, frames per second (FPS), and power consumption.

The accuracy of the object detection models was assessed by using standard benchmark datasets, including COCO. The models were evaluated on a range of object classes and sizes to measure their ability to detect objects accurately. Object detection models are typically evaluated and checked for their performance using various metrics and methods. One of the common ways to check is the mAP. It is calculated using the following formula (1):

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \,, \tag{1}$$

where N is the number of the classes, AP_i the average precision of the class i.

Average precision is calculated based on formula (2):

$$AP = \sum_{k=0}^{k=n-1}[Recalls(k) - Recalls(k+1)] * Precisions(k). \tag{2}$$

Precision and recall are described in formulas (3) and (4).

The precision is calculated as the ratio between the number of positive samples correctly classified to the total number of samples classified as positive. The formula for the precision is as follows:

$$Precision = \frac{True_{positive}}{True_{positive} + True_{negative}}. \quad (3)$$

The recall is calculated as the ratio between the number of Positive samples correctly classified as positive to the total number of positive samples:

$$Precision = \frac{True_{positive}}{True_{positive} + False_{negative}}. \quad (4)$$

Table 1 shows the evaluation of several objects. The metrics used for evaluation include the mean average precision (mAP).

Table 1

The object detection mean accuracy

	yolov3	yolov4	yolov5	yolov6
Mean Accuracy	0.755967731	0.766655457	0.719760493	0.768194773
Person	0.917891229	0.896341966	0.818452187	0.878627467
Car	0.752605661	0.77842973	0.713794419	0.745153376
Bicycle	0.774284912	0.77842973	0.713794419	0.745153376
Dog	0.865213667	0.876938071	0.779942692	0.866484686
Chair	0.684339229	0.697351131	0.64940328	0.696721576

Table 1 presents the accuracy results for YOLOv3, YOLOv4, YOLOv5, and YOLOv6 on the COCO dataset. The mAP values indicate the overall detection performance of each model.

The real-time processing capability of the FPGA-based models is crucial for applications such as autonomous vehicles and surveillance systems. The FPS rate is achieved for each model during the inference on the ZCU104 FPGA. It is calculated based on the number of processed images divided by the process time in seconds. The results are shown in Table 2.

Table 2

The frame per second rate results

	yolov3	yolov4	yolov5	yolov6
FPS	28.9241	21.6409	18.972	25.668

These results highlight the models' varying computational demands and their ability to process the frames at high speeds.

Efficiency in power consumption is a key factor in embedded systems, particularly in resource-constrained environments. The power consumption of the

FPGA-based object detection models was measured using power monitoring tools. The DPU architecture was synthesized and implemented on the ZCU104 FPGA using the Xilinx Vivado tool. The synthesized and implemented designs are shown in Fig.2 and Fig.3.

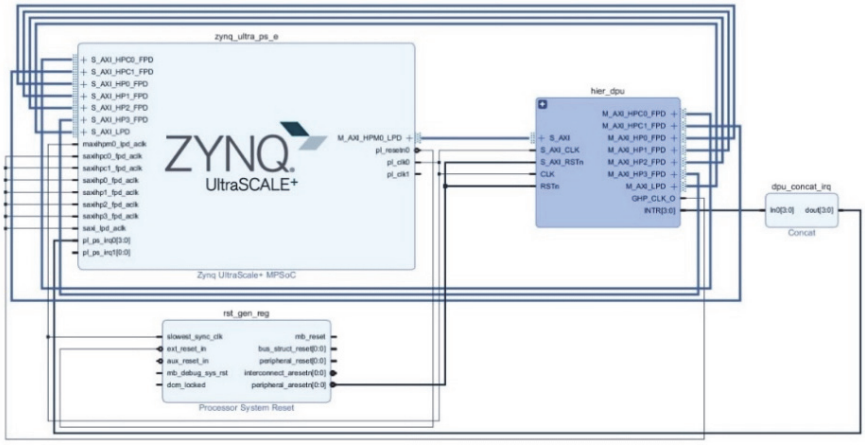

Fig. 2. The synthesized DPU design

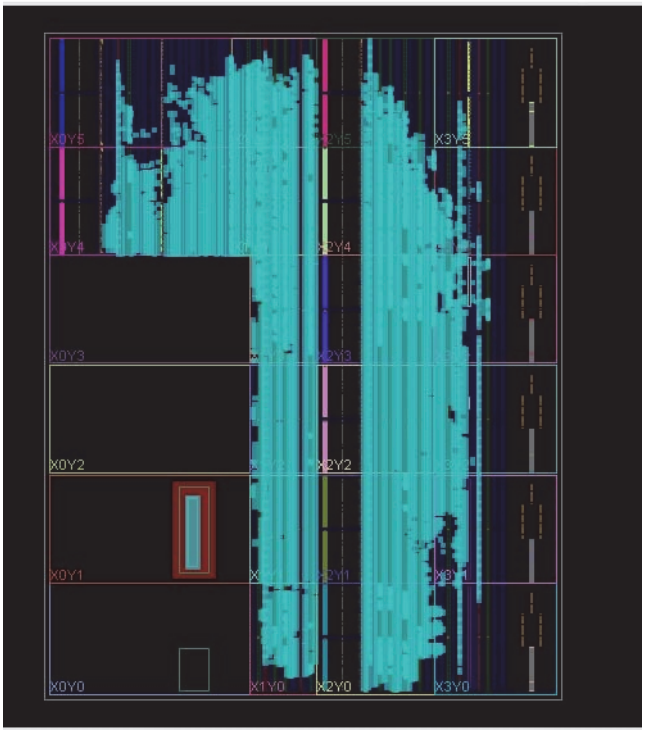

Fig. 3. Implemented DPU design

The resource utilization for the DPU architecture implementation on the ZCU104 board is as follows: LUT –28%, FF – 24%, BRAM – 83%, DSP – 34%.

Table 3 provides an overview of the power consumption for YOLOv3, YOLOv4, YOLOv5, and YOLOv6 on the ZCU104 FPGA. The power consumption was measured using the Xilinx Power Estimator (UG440) [8].

Table 3

Power consumption results (W)

	yolov3	yolov4	yolov5	yolov6
Power	13.812	15.282	14.625	13.326

Conclusion. In this study, a comprehensive analysis of the YOLO object recognition models is conducted, focusing on their potential benefits when implemented on Field-Programmable Gate Arrays (FPGAs). The performance in terms of real-time processing, latency reduction, energy efficiency, customizability, and resource utilization were evaluated for each model.

The results demonstrate several notable advantages of implementing YOLO models on FPGAs. Firstly, FPGA acceleration allowed to achieve real-time processing capabilities with a minimum of 19 fps for yolov5 and 29 fps for yolov3. The mean accuracy of the YOLO models consistently shows promising results, with yolov6 showing an average mean accuracy of approximately 0.7682.

Another benefit is low power consumption of FPGA-based YOLO implementations, with a worst case of 15.282w for yolov4 model. This factor makes them particularly attractive for resource-constrained environments and battery-powered devices, aligning well with the growing demand for energy-efficient computing.

REFERENCES

1. **Joseph Redmon, Ali Farhadi.** YOLOv3: An Incremental Improvement. – ArXiv, 2018.
2. **Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao.** YOLOv4: Optimal Speed and Accuracy of Object Detection. - ArXiv, 2020.
3. **Glenn Jocher.** Ultralytics/yolov5. - Zenodo, 2020.
4. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications/ **Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, Yiduo Li, Bo Zhang, Yufei Liang, Linyuan Zhou, Xiaoming Xu, Xiangxiang Chu, Xiaoming Wei, Xiaolin Wei.** – ArXiv, 2022.

5. **Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun.** Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. – ArXiv, 2015.
6. SSD: Single Shot MultiBox Detector /**Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg.**– ArXiv, 2015.
7. **Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi.** You Only Look Once: Unified, Real-Time Object Detection. – ArXiv, 2015.
8. https://docs.xilinx.com/r/en-US/ug440-xilinx-power-estimator

National Polytechnic University of Armenia. The material is received 20.10.2023.

Մ.Տ. ԳՐԻԳՈՐՅԱՆ

DPU-Ի ՎՐԱ ԻՐԱԿԱՆԱՑՎԱԾ YOLO ՄՈԴԵԼՆԵՐԻ ԱՇԽԱՏԱՆՔԻ ՉԱՓՈՒՄՆԵՐԻ ՈՒՍՈՒՄՆԱՍԻՐՈՒՄ. ՀԱՄԵՄԱՏԱԿԱՆ ՎԵՐԼՈՒԾՈՒԹՅՈՒՆ

Ներկայացվել է կատարողականության ցուցանիշների համապարփակ վերլուծություն, որը ձեռք է բերվել «You Only Look Once» տվյալների բազայի (YOLO) տարբեր մոդելների ներդրմամբ՝ օգտագործելով համապատասխան խոր ուսուցման մշակման սարք (DPU): YOLO մոդելները հայտնի են իրենց իրական ժամանակում օբյեկտների հայտնաբերման հնարավորություններով, ինչը նրանց դարձնում է լավ ընտրություն մի շարք ծրագրերի դեպքում, ներառյալ ինքնավար մեքենաները, հսկողության համակարգերը և ռոբոտաշինությունը: Սույն ուսումնասիրության մեջ YOLO մոդելները տեղադրվում են մասնագիտացված սարքային արագացուցչի, մասնավորապես՝ DPU-ի վրա, որը կատարվում է դրա աշխատանքի արագությունը, ճշգրտությունը և էներգիայի սպառման արդյունավետությունը գնահատելու համար: Կատարելով YOLO մոդելի բազմաթիվ տարբերակների համեմատական գնահատում՝ ցույց է տրվում, թե ինչպես է յուրաքանչյուր մոդել փոխազդում DPU ճարտարապետության հետ: Մոդելները ներառում են YOLOv3, YOLOv4, YOLOv5, YOLOv6**:** Փորձերը ներառում են այս մոդելների չափորոշիչների ուսումնասիրումը տարբեր տվյալների հավաքածուներում և տարբեր սարքային կազմաձևերում: Արդյունքները ոչ միայն ընդգծում են YOLO մոդելների հիմքով նախագծերի DPU-ների վրա կիրառման առավելություններն ու սահմանափակումները, այլ նաև նպաստում են՝ ընտրելու առավել հարմար մոդել-DPU համակցությունը, տվյալ պահանջները հաշվի առնելով: Ուսումնասիրությունը նպաստում է իրական ժամանակում օբյեկտների հայտնաբերման համակարգերի լավարկմանը և օգնում նախագծողներին մոդելի և սարքավորման ընտրության դեպքում՝ բարձրացնելով տեղեկացվածությունը որոշումներ կայացնելու հարցում:

Առանցքային բառեր. FPGA, DPU, օբյեկտների հայտնաբերում, YOLO:

М.Т. ГРИГОРЯН

ИССЛЕДОВАНИЕ ПОКАЗАТЕЛЕЙ ПРОИЗВОДИТЕЛЬНОСТИ МОДЕЛЕЙ YOLO, РЕАЛИЗОВАННЫХ НА DPU: СРАВНИТЕЛЬНЫЙ АНАЛИЗ

Представлен всесторонний анализ показателей производительности различных моделей, созданных на основе набора данных “You Only Look Once” (YOLO) и реализованных на процессоре глубокого обучения (DPU). Модели YOLO известны своими возможностями обнаружения объектов в реальном времени, что позволяет их использовать для целого ряда приложений, включая автономные транспортные средства, системы наблюдения и робототехнику. В исследовании модели YOLO развертываются на специализированном аппаратном ускорителе, в частности DPU, для оценки скорости, точности и энергоэффективности их работы. Путем проведения сравнительной оценки нескольких вариантов YOLO, включая YOLOv3, YOLOv4, YOLOv5, YOLOv6, показано, как каждая модель взаимодействует с архитектурой DPU. Эксперименты включают сравнение этих моделей с различными наборами данных и различными конфигурациями оборудования. Результаты не только подчеркивают преимущества и ограничения использования DPU для приложений на основе YOLO, но также помогают выбрать наиболее подходящую комбинацию модель-DPU на основе конкретных требований к производительности. Исследование способствует оптимизации систем обнаружения объектов в реальном времени и помогает специалистам-практикам принимать обоснованные решения относительно выбора модели и оборудования.

Ключевые слова: FPGA, DPU, обнаружение объектов, YOLO.