SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Tuberculosis Classification Using Deep Learning and FPGA Inferencing

Fazrul Faiz Zakaria[1,2,4,*], Asral Bahari Jambek[1,4], Norfadila Mahrom[3,4], Rafikha Aliana A Raof[3,4], Mohd Nazri Mohd Warip[2,4], Phak Len Al Eh Kan[2,4], Muslim Mustapa[5]

[1] Micro System Technology, Centre of Excellence (CoE), Universiti Malaysia Perlis (UniMAP), Perlis, Malaysia
[2] Advanced Computing, Centre of Excellence, Universiti Malaysia Perlis (UniMAP), Perlis, Malaysia
[3] Sports Engineering Research Centre, Centre of Excellence (SERC), Universiti Malaysia Perlis, Perlis, Malaysia
[4] Faculty of Electronic Engineering Technology, Universiti Malaysia Perlis (UniMAP), Perlis, Malaysia
[5] Programmable Solutions Group, Intel Malaysia

ARTICLE INFO

Article history:
Received 6 October 2022
Received in revised form 27 December 2022
Accepted 18 January 2023
Available online 9 February 2023

Keywords:
Tuberculosis classification; Deep-learning; FPGA inferencing

ABSTRACT

Among the top 10 leading causes of mortality, tuberculosis (TB) is a chronic lung illness caused by a bacterial infection. Due to its efficiency and performance, using deep learning technology with FPGA as an accelerator has become a standard application in this work. However, considering the vast amount of data collected for medical diagnosis, the average inference speed is inadequate. In this scenario, the FPGA speeds the deep learning inference process enabling the real-time deployment of TB classification with low latency. This paper summarizes the findings of model deployment across various computing devices in inferencing deep learning technology with FPGA. The study includes model performance evaluation, throughput, and latency comparison with different batch sizes to the extent of expected delay for real-world deployment. The result concludes that FPGA is the most suitable to act as a deep learning inference accelerator with a high throughput-to-latency ratio and fast parallel inference. The FPGA inferencing demonstrated an increment of 21.8% in throughput while maintaining a 31% lower latency than GPU inferencing and 6x more energy efficiency. The proposed inferencing also delivered over 90% accuracy and selectivity to detect and localize the TB.

1. Introduction

Tuberculosis (TB) is a significant public health concern in some parts, particularly in underdeveloped nations. While most people have tuberculosis in their lungs, others may have an infection in other bodily organs [1]. As a result, diagnosing tuberculosis is significantly more complex than other infectious illnesses, necessitating many tests [2, 3]. Significant intra- and inter-observer variability in chest X-ray (CXR) readings, on the other hand, might result in over-or under-diagnosis of TB [4, 5]. Although the CXR is an effective tool for tuberculosis screening, a suspected individual requires clinical, biochemical, and genetic studies before diagnosing and administering treatments. CXR is a primary tool for triaging and screening for tuberculosis as part of the World Health

* *Corresponding author.*
E-mail address: ffaiz@unimap.edu.my

https://doi.org/10.37934/araset.29.3.105114

Organization's (WHO) systematic screening strategy to ensure early and accurate diagnosis for all people with tuberculosis due to its relatively high sensitivity, depending on how the CXR is interpreted [6].

Deep learning has made it easier for convolutional neural networks (CNNs) to outperform other recognition algorithms regarding image-based classification and recognition problems. CNN is the best choice for complex medical problem solving because it can automatically find valuable features from the data itself. In the past, CAD systems with deep-learning algorithms have been very good at detecting medical diseases. They have generated a wide range of high-quality diagnostic solutions while highlighting suspicious features [7].

FPGAs have gained favor as hardware accelerators for improving the computation efficiency of CNN models due to recent advancements in FPGA technology. Current FPGAs have been reported to have performance equivalent to GPUs, with 9.2 TFLOPS for the Intel Stratix 10 FPGA and up to 40 TFLOPS for the Intel Agilex FPGA [8]. Additionally, the efficiency of data transfer between the FPGA and external memory is frequently the most significant shortcoming of FPGA-based accelerators. It has been improved by integrating High Bandwidth Memory (HBM2) into the FPGA die in the same package, such as the Intel Stratix 10 MX FPGA and Xilinx Virtex UltraScale+ with HBM2. As a result, academia and business have shown considerable interest in FPGA-based CNN accelerators. PipeCNN [9], hls4ml [10] from Fermilab, and Intel's OpenVINO toolkit [11] are successful examples. The deep neural network has demonstrated its efficacy in medical fields such as tuberculosis [12, 13]. Using an FPGA in conjunction with a deep neural network accelerates the complicated computations occurring within the neurons. Apart from its quick calculation speed, the FPGA's adequate computing power and low latency induction significantly benefit from low latency and high throughput [14].

This work aims to demonstrate the performances of accelerated deep learning using FPGA inference for tuberculosis classification across many devices using Intel DevCloud for the Edge Computing, more precisely, an FPGA. The inference platforms include the Intel® Xeon® Gold 6258R and the Intel® UHD Graphics 620. Each device was chosen because of its outstanding performance in its specific architecture. The accuracy of model inference across many devices will be evaluated using the same batch of photos and their associated accuracy, specificity, sensitivity, and inference time per image. This value indicates which device processes data quicker in synchronous mode. The second section deploys the model asynchronously over several machines to compute their processing power in a single second and the resulting delay. Model reading and loading times are supplied to help visualize the expected delay while distributing a model across devices.

2. Background Study

2.1 Deep Learning in Medical Diagnosis

One of the most challenging challenges in medical image processing is providing vital information about the organs' shapes and sizes. There is no deep learning algorithm capable of doing this task flawlessly. CNN is a subset of deep learning algorithms most frequently used to analyze visual data. A CNN's structure consists of a single input layer, a single output layer, and numerous hidden layers, as shown in Figure 1. Convolutional, pooling, and wholly linked layers are examples of hidden layers. Convolutional layers collect critical information from pictures and convert it to a feature map; pooling layers lower the input dimension; fully connected levels connect every neuron in one layer to every neuron in another, producing classifications.

A convolutional layer's fundamental component is the convolution kernels, a collection of matrices representing specific target patterns within the input picture. The example in Figure 2 uses a black and white image with an X in the center as the input image. While the two slashes and one

cross are the specific patterns we are looking for, these features are decomposed into convolution kernels. After multiplying the original image by the convolution kernels and pooling, we may obtain three outputs corresponding to various regions of the original image. Thus, CNN extracts essential aspects from the input image and judgments based on these retrieved features. CNN has been applied to feature classification in various fields and consistently outperforms people and other algorithms in classification accuracy. When CNN is used for medical diagnosis, medical pictures serve as the network's input, and the network generates models that enable the identification of specific diseases. Thus, physicians may use CNN to double-check their diagnosis results.

Additionally, deep learning is computationally costly. Since neural networks must typically deal with big datasets with sophisticated layer designs, training a model from the start might take many weeks. In contrast, standard algorithms usually require only a few minutes, hours, or days. Additionally, this degree of data processing is very hardware-dependent since it necessitates parallel processing capability. CNN provides one answer in the form of pooling layers, which lower the dimensionality of the parameters and significantly reduce the amount of data. Despite the difficulties described above, deep learning outperforms other learning algorithms if the amount of data collected reaches a particular threshold. Algorithms are trained to make choices by examining data sets and commenting on matching events. This feature enables the machine to do far more complicated tasks and reduces repetitive labor.

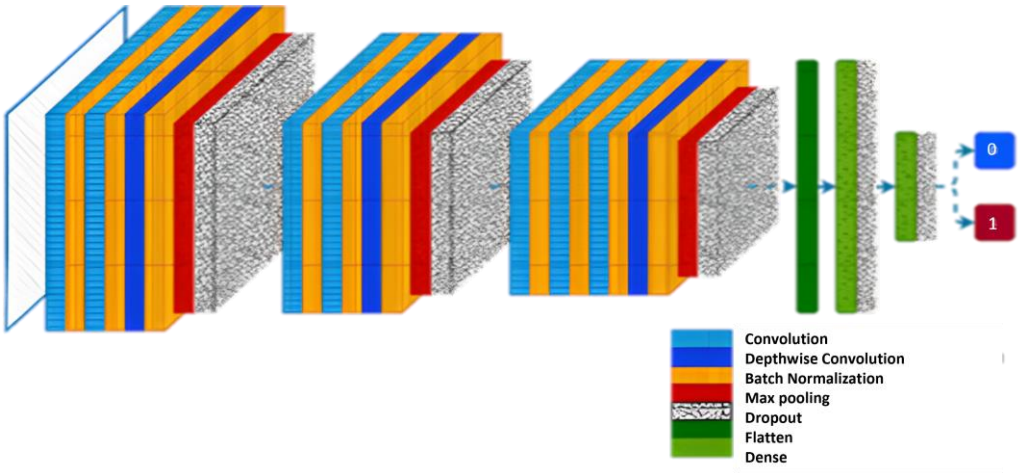


Fig. 1. CNN Architecture

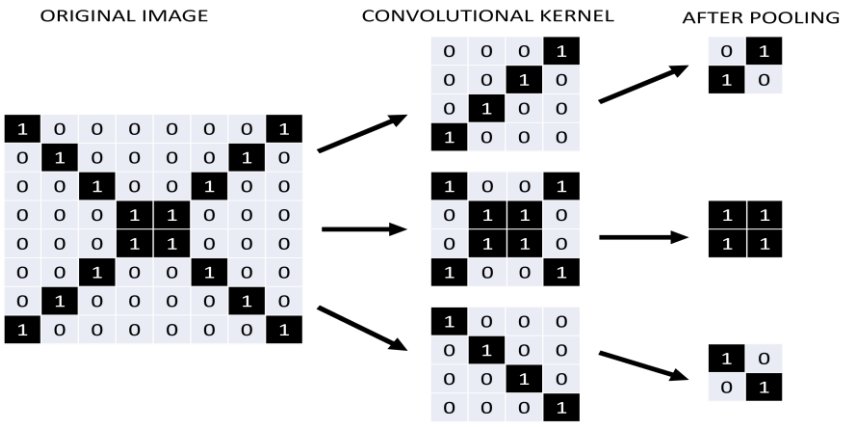


Fig. 2. Example of convolution kernels

2.2 Deep Learning in Medical Diagnosis

The OpenCL heterogeneous platform based on FPGA employs a heterogeneous mode in which the CPU acts as the host, and the FPGA acts as the device. The FPGA acts as a parallel acceleration device, significantly increasing the processing capability of the CPU. The network creation process consists of four steps: reading the network structure file, reading the weight file, defining the parameters of each network layer, assigning storage space, and reading the data configuration file. When the FPGA-side kernel program is ready to process this data, it will read them in batches or all at once from global memory. After the FPGA processes this data, it writes them back to the off-chip global memory, either at once or in batches. Finally, the host-side application will read the processed data from the off-chip global memory to the host-side storage. The CPU should execute specific processes with low parallelism. A balanced weight pruning strategy for hardware efficiency design is required to eliminate local imbalances and maximize resource consumption [15]. Then, using the same calculation method, propagate forward to obtain the output of neurons in each layer, and lastly, get the production of neurons in all output layers. The output feature map is calculated using the maximum or average value of the input feature map data in the local perception window. As a result, if the FPGA is assigned an excessive number of tasks, the proportion of work performed by the host and FPGA devices in the acceleration system must be modified.

3. Experimental Setup

This section discusses the dataset, model, model conversion, inference, and hardware computing devices as the platform for FPGA-based inferencing utilized in this work.

3.1 Dataset and Model

3.1.1 National library of medicine dataset

The TB classification challenge was solved using a publicly accessible dataset. The National Library of Medicine (NLM) in the United States [16] has made two lung X-ray datasets public: the Montgomery County (MC) and Shenzhen, China (CHN) datasets. Both databases contain 138 and 660 posterior-anterior (PA) chest X-ray images. The photos in the MC database had a resolution of 4,020 x 4.892 or 4,892 x 4.020 pixels, but the photographs in the CHN database had a resolution of varied but about 3000 x 3000 pixels. Out of 138 chest X-ray pictures in the MC database, 58 were collected from various tuberculosis patients, and 80 were taken from healthy participants. Three hundred thirty-six (336) photos were collected from multiple tuberculosis patients in the CHN database, whereas 324 images were taken from everyday people. As a result, this NLM database contains 406 healthy and 394 tuberculosis-infected X-ray pictures.

3.1.2 ResNet-50 binary

ResNet-50 is a deep learning model for image classification that allows applications to characterize a picture with a maximum error rate of 3.57% [17]. The model contains input, output, and hidden layers that define its underlying method via a network of linked neurons. Information is transmitted from one layer to the next, as shown in Figure 3. FPGAs are perfectly positioned to incorporate new research in deep learning architectures and models into hardware without requiring a new silicon spin. Intel has significantly refined the ResNet-50 model for low-bit (FP11) precision inferencing, allowing vision applications to operate more efficiently. Additionally, this new model's hardware-level optimization and numerous software-level improvements allow smooth application

integration while greatly enhancing performance. The ResNet-50 model consists of 150,528 input neurons, 1,000 output neurons, and 50 layers for 3,8 billion operations. With recent enhancements to the OpenVINO SDK, the Intel PAC with Arria 10 FPGA can run the ResNet-50 model at higher performance than previously reported.

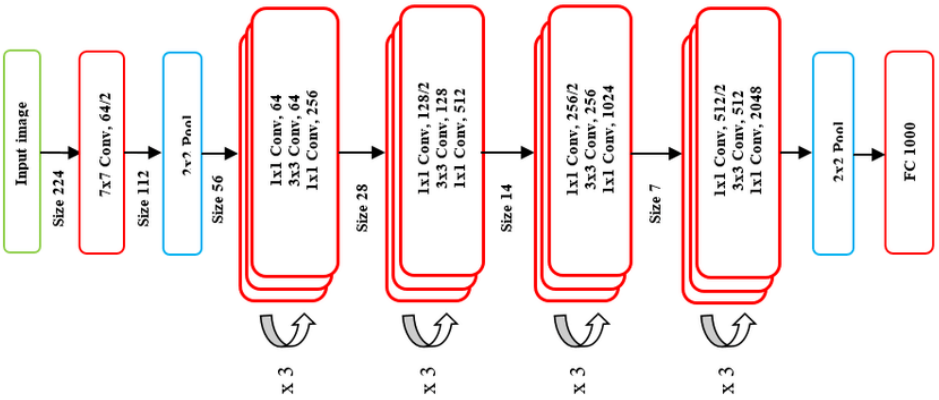


Fig. 3. ResNet-50 Architecture

3.2 Model Conversion and Inference

As shown in Figure 4, the Model Optimizer and Inference Engine are the two components of Open Visual Inference and Neural Network Optimization (OpenVINO). The OpenVINO toolbox distributes the workload across Intel devices to optimize performance. The Model Optimizer is a command-line, cross-platform utility that helps transition the training and deployment environment on a target inference engine. A network model trained with a supported framework serves as the Model Optimizer's input. It executes static model analysis and optimizes the input of deep learning models for optimal execution on endpoint target devices, which can be a CPU, GPU, FPGA, or a mix thereof (HETERO). The result of the Model Optimizer is an Intermediate Representation (IR) that may be utilized as input by the specified target Inference Engine. The Inference Engine is a C++ library comprising a set of C++ classes that infer data (images) to provide a result. The C++ library offers an API for reading IR, setting input and output formats, and executing models on devices. In our research, the objective of the Deployment and Inference phase is to deploy the trained model on an FPGA to speed up the classification process.

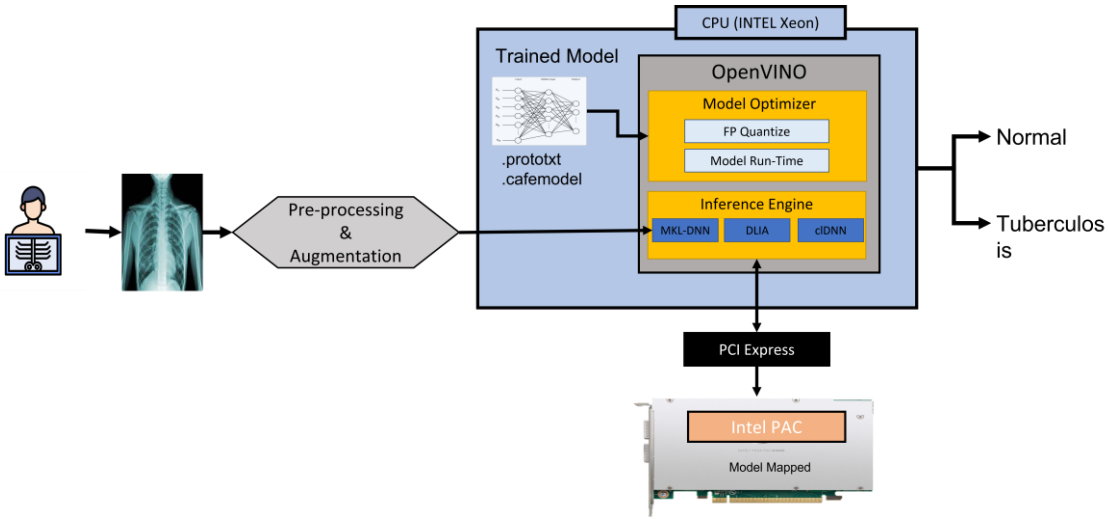


Fig. 4. Deep learning with FPGA inferencing

3.3 Computing Devices for Inference

Table 1 below illustrates the computing device's name and abbreviation used in this section. Their respective device type is shown together to understand the device type better. The alias of each computing device will be brought into the discussion in the next section.

Table 1
Computing device and abbreviation.

Computing Device	*Alias*	*Type*
Intel® Xeon® Gold 6230R	XEON	CPU
Intel® UHD Graphics 620	GPU	GPU
Intel® Arria® 10 GX 1150 FPGA with Mustang-F100-A10	FPGA	FPGA

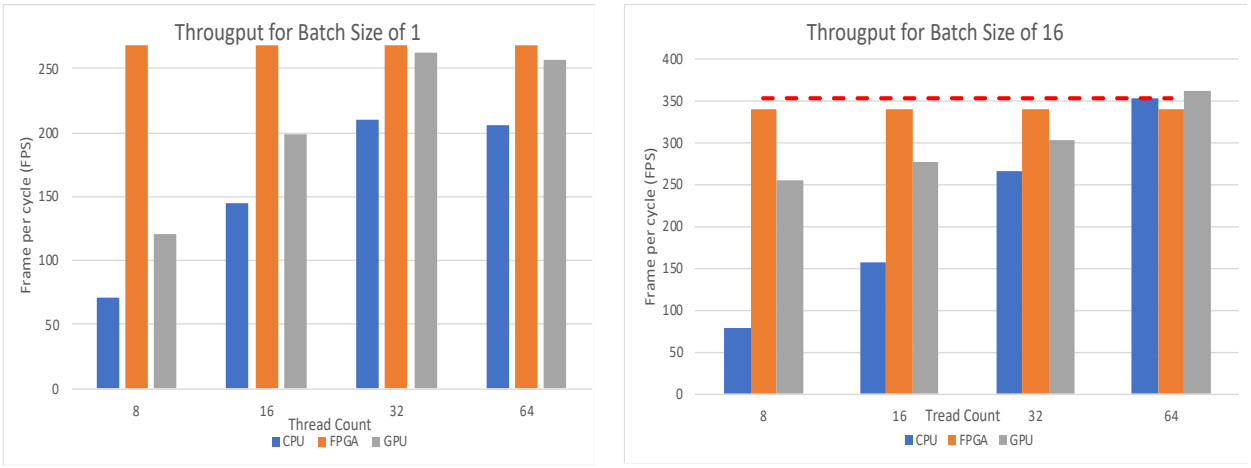
4. Result, Comparison, and Analysis

We configured the Intel PAC on a Dell Precision 7920 Tower Workstation with an Intel Xeon Gold 6230R processor operating at 2.10 GHz and CentOS Linux (release 7.6). We compare the performance of FPGAs to that of CPUs, highlighting both devices' throughput, latency, energy efficiency, and model performance.

4.1 Throughput and Latency

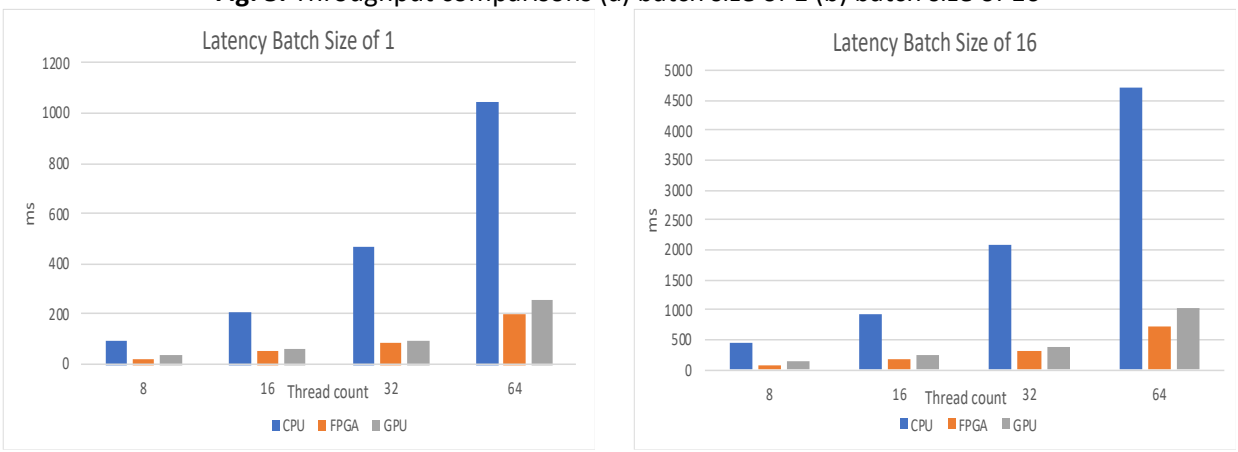
Throughput is the rate at which a set of images are processed per second. This unit is denoted as frames per second (FPS). One or more images can be processed in batches, often known as batch size. Figure 5 depicts the throughput of the CPU, FPGA, and batch sizes of 1 and 16 for various CPU configurations, i.e., thread count. As stated, the FPGA performance is consistent regardless of the number of threads for a particular batch size, suggesting that the FPGA delivers predictable performance. The FPGA also shows 21.8% more throughput than GPU at 64 thread count for a single batch. In contrast, as illustrated by the horizontal dashed lines, Both CPU and GPU surpass the FPGA at 9.7% and 3.8%, respectively, in FPS only when the batch size is 16 and the thread count is 64. In conclusion, increasing the number of CPU threads from 32 to 64 increases throughput by just 24.3%.

The inference is another name for latency, which refers to the amount of time needed to process a request. A request in this context can have a batch size of 1 or 16, as shown in Figure 6. Compared to the CPU, the FPGA produces a lesser latency while processing with higher FPS. On batch inference, FPGA achieves a 0.6:1 throughput-to-latency trade-off compared to GPU, 0.3:1.



(a) (b)

Fig. 5. Throughput comparisons (a) batch size of 1 (b) batch size of 16

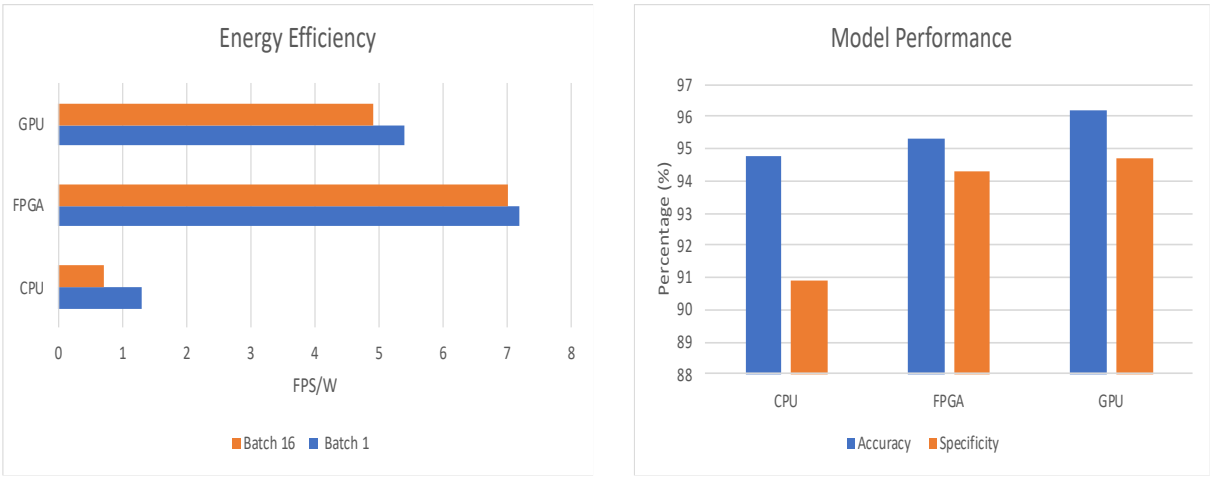


(a) (b)

Fig. 6. Latency comparison (a) batch size of 1 (b) batch of 16

4.2 Efficiency

Performance efficiency is the throughput achieved, normalized by the power consumed. The price of this rather sublinear performance increase is reduced performance efficiency: by a factor of 0.4, as in Figure 7(a). The measured system power of the CPU execution reached a maximum of 416 W. With an estimated board power of 50W on the PAC, the FPGA achieves a much higher performance efficiency - a factor of 6x - compared to the CPU, with only a 20% increase in the server' power budget.



(a) (b)

Fig. 7. (a) Energy Efficiency (b) Accuracy and specificity performance

4.3 Performance

Figure 7(b) shows the model performance evaluation on different computing devices. All the devices' performance evaluated is above 90% accuracy, indicating that the deployed model can classify the given tuberculosis images with above 90% of accurateness, with the GPU having a 1.2% and higher precision over the FPGA and CPU, respectively. The specificity shows how well the model can localize tuberculosis from the x-ray images. As shown in Figure 7(b), the FPGA inference can deliver nearly the same specificity performance of GPU with only 0.4% differences. Nevertheless, the performance differences are insignificant. The change in the model performance is a phenomenon obtained through the changes in the execution plugin when running inference on a different platform. The performance is expected to change when tested with other images.

5. Discussion

Based on the experiment's observation, FPGA inferencing offers several advantages, including high performance, low latency, and power efficiency. The hardware can be optimized specifically for deep learning computations, providing faster and more efficient inference than traditional CPUs or GPUs. Additionally, FPGAs can be customized to meet specific requirements, such as low latency or high throughput. However, FPGA inferencing has disadvantages, such as complexity, cost, and flexibility. The development of FPGA-based inference solutions can be more complex and costly, and updating the solution to accommodate new deep learning models or changes in requirements can take more work. Additionally, the transfer of large amounts of data to and from an FPGA can be slower, impacting overall performance.

6. Conclusions

This work describes the Intel Programmable Accelerator Card (PAC) with Arria 10 FPGA for deep learning inference. Incorporating the Intel PAC on an x86-based Dell Precision 7920 Tower Workstation enhances ResNet-50 performance compared to the previously available ResNet-50 model. Specifically, we studied the implemented model's throughput, latency, and efficiency. While the quad-socket CPU arrangement produced 330 FPS at 0.79 FPS/Watt (a 60 percent efficiency drop compared to the dual-socket configuration), the FPGAs obtained 1,251 FPS at 6 FPS/Watt with a 20

percent power increase per PAC to the server's power budget. Ongoing hardware and software system stack enhancements are anticipated to increase these performance metrics. Overall, using FPGAs for deep learning inferencing can provide significant benefits. Still, it is essential to carefully consider the trade-offs and choose the right approach for a given use case.

Acknowledgment
The author gratefully acknowledges financial support from the Malaysian Ministry of Higher Education (MOHE) under the Fundamental Research Grant Scheme (FRGS) (Grant number: FRGS/1/2021/TK0/UNIMAP/02/6).

References

[1] Bhalla, Ashu Seith, Ankur Goyal, Randeep Guleria, and Arun Kumar Gupta. "Chest tuberculosis: Radiological review and imaging recommendations." *Indian Journal of Radiology and Imaging* 25, no. 03 (2015): 213-225. https://doi.org/10.4103/0971-3026.161431

[2] van't Hoog, Anna H., Helen K. Meme, Kayla F. Laserson, Janet A. Agaya, Benson G. Muchiri, Willie A. Githui, Lazarus O. Odeny, Barbara J. Marston, and Martien W. Borgdorff. "Screening strategies for tuberculosis prevalence surveys: the value of chest radiography and symptoms." *PloS one* 7, no. 7 (2012): e38691. https://doi.org/10.1371/journal.pone.0038691

[3] Kebede, Wakjira, Gemeda Abebe, Esayas Kebede Gudina, Elias Kedir, Thuy Ngan Tran, and Annelies Van Rie. "The role of chest radiography in the diagnosis of bacteriologically confirmed pulmonary tuberculosis in hospitalised Xpert MTB/RIF-negative patients." *ERJ Open Research* 7, no. 1 (2021). https://doi.org/10.1183/23120541.00708-2020

[4] Van Cleeff, M. R. A., L. E. Kivihya-Ndugga, H. Meme, J. A. Odhiambo, and P. R. Klatser. "The role and performance of chest X-ray for the diagnosis of tuberculosis: a cost-effectiveness analysis in Nairobi, Kenya." *BMC infectious diseases* 5, no. 1 (2005): 1-9. https://doi.org/10.1186/1471-2334-5-111

[5] Graham, S., K. Das Gupta, J. Hidvegi R, R. Hanson, J. Kosiuk, K. Al Zahrani, and D. Menzies. "Chest radiograph abnormalities associated with tuberculosis: reproducibility and yield of active cases." *The International Journal of Tuberculosis and Lung Disease* 6, no. 2 (2002): 137-142.

[6] Nguyen, Lan Huu, Andrew J. Codlin, Luan Nguyen Quang Vo, Thang Dao, Duc Tran, Rachel J. Forse, Thanh Nguyen Vu et al. "An evaluation of programmatic community-based chest X-ray screening for tuberculosis in Ho Chi Minh City, Vietnam." *Tropical medicine and infectious disease* 5, no. 4 (2020): 185. https://doi.org/10.3390/tropicalmed5040185

[7] Yamashita, Rikiya, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. "Convolutional neural networks: an overview and application in radiology." *Insights into imaging* 9 (2018): 611-629. https://doi.org/10.1007/s13244-018-0639-9

[8] Nurvitadhi, Eriko, Ganesh Venkatesh, Jaewoong Sim, Debbie Marr, Randy Huang, Jason Ong Gee Hock, Yeong Tat Liew et al. "Can FPGAs beat GPUs in accelerating next-generation deep neural networks?." In *Proceedings of the 2017 ACM/SIGDA international symposium on field-programmable gate arrays*, pp. 5-14. 2017. https://doi.org/10.1145/3020078.3021740

[9] Wang, Dong, Jianjing An, and Ke Xu. "PipeCNN: An OpenCL-based FPGA accelerator for large-scale convolution neuron networks." *arXiv preprint arXiv:1611.02450* (2016). https://doi.org/10.1109/FPT.2017.8280160

[10] Duarte, Javier, Song Han, Philip Harris, Sergo Jindariani, Edward Kreinar, Benjamin Kreis, Jennifer Ngadiuba et al. "Fast inference of deep neural networks in FPGAs for particle physics." *Journal of Instrumentation* 13, no. 07 (2018): P07027. https://doi.org/10.1088/1748-0221/13/07/P07027

[11] Zunin, V. V. "Intel OpenVINO Toolkit for Computer Vision: Object Detection and Semantic Segmentation." In *2021 International Russian Automation Conference (RusAutoCon)*, pp. 847-851. IEEE, 2021. https://doi.org/10.1109/RusAutoCon52004.2021.9537452

[12] Patel, Mahisha, Amitabh Das, Vineet Kumar Pant, and M. Jayasurya. "Detection of Tuberculosis in Radiographs using Deep Learning-based Ensemble Methods." In *2021 Smart Technologies, Communication and Robotics (STCR)*, pp. 1-7. IEEE, 2021. https://doi.org/10.1109/STCR51658.2021.9588936

[13] Gao, Xiaohong, Richard Comley, and Maleika Heenaye-Mamode Khan. "An enhanced deep learning architecture for classification of tuberculosis types from CT lung images." In *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 2486-2490. IEEE, 2020. https://doi.org/10.1109/ICIP40778.2020.9190815

[14] Zhang, Sida. "Application of FPGA in Deep Learning." In *2020 International Conference on Advance in Ambient Computing and Intelligence (ICAACI)*, pp. 52-55. IEEE, 2020. https://doi.org/10.1109/ICAACI50733.2020.00015

[15] Gankidi, Pranay Reddy, and Jekan Thangavelautham. "FPGA architecture for deep learning and its application to planetary robotics." In *2017 IEEE Aerospace Conference*, pp. 1-9. IEEE, 2017. https://doi.org/10.1109/AERO.2017.7943929

[16] Jaeger, Stefan, Sema Candemir, Sameer Antani, Yì-Xiáng J. Wáng, Pu-Xuan Lu, and George Thoma. "Two public chest X-ray datasets for computer-aided screening of pulmonary diseases." *Quantitative imaging in medicine and surgery* 4, no. 6 (2014): 475. https://doi.org/10.3978/j.issn.2223-4292.2014.11.20

[17] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016. https://doi.org/10.1109/CVPR.2016.90