

Research on Hardware Acceleration Optimisation Strategies for Deep Learning in Computer Vision

Yifan Gao *

School of Electrical Engineering, Shandong University, Jinan, China

* Corresponding author: 201800190090@mail.sdu.edu.cn

Abstract. As deep neural network (DNN) models get larger and more complicated, the importance of hardware acceleration becomes more and more apparent. This paper discusses various hardware acceleration strategies for deep learning, especially in the area of computer vision. It explores the use of GPUs, FPGAs, and ASICs, detailing their respective strengths and weaknesses in accelerating DNNs. This paper argues that the future of DNN hardware acceleration lies in hybrid approaches that combine the advantages of different architectures. Software advances such as improved compilers and synthesis tools will also play a critical role in making these techniques more accessible. By utilizing the appropriate hardware technology for a given task and continuing to innovate in both hardware and software, computer vision will make significant advances in performance, efficiency, and scalability. This hybrid approach is key to the future of DNN hardware acceleration, offering a path to overcome the limitations of any single type of hardware.

Keywords: Deep neural network, Computer vision, Hardware accelerator.

1. Introduction

DNNs, the basis for numerous contemporary AI applications, have seen a golden phase of development since their breakthroughs in speech recognition and image recognition tasks. In many domains, DNNs have demonstrated computational excellence over humans. As shown in Fig. 1, since AlexNet was assessed in a major competition for visual recognition on the ImageNet1k dataset in 2012, the overall aim of the literature has been to develop improved accuracy and efficiency in networks while emphasizing the scale and complexity of the models [1]. The significant increase in the computational complexity of the models has led to an increasing arithmetic demand, as well as more demanding requirements on the power consumption and efficiency of the chips.

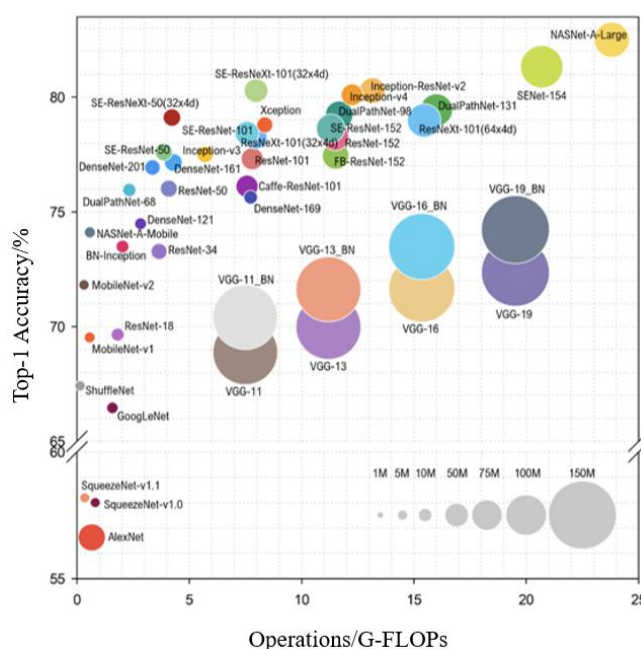


Figure 1. Ball graph showing the relationship between computational complexity and Top-1 ImageNet-1k accuracy. Each ball's size reflects the intricacy of the model [1]

Hardware acceleration strategies have become effective solutions for accelerating large-scale deep neural networks. For the large number of computational resources demanded, traditional CPU-based systems are often unable to meet it efficiently, so some specialized hardware accelerators have been developed and utilized in a large number of ways, such as Graphics Processor Units (GPUs), Neural Processing Units (NPU), amongst others. These accelerators are designed to deal with the parallelizability of neural network computation, thus greatly improving the computational efficiency.

A significant development that transformed deep learning is hardware acceleration, and hardware acceleration has supported a variety of computational models such as image recognition and speech recognition. Models like Convolutional Neural Networks (CNNs) and advanced architectures such as TRResNet have gained significant performance gains through hardware acceleration [2]. The advantages of hardware-accelerated deep learning models in computer vision are manifold. From self-driving cars that require real-time image processing to medical imaging systems that require high-precision diagnostics, enhancements to hardware-accelerated models help accomplish more complex and accurate tasks. This, in turn, drives innovation and enables researchers and engineers to develop more advanced algorithms and applications.

This paper provides an overview of noteworthy hardware components that have been created or modified to speed DNN-based computer vision algorithms, including FPGAs, GPUs, and other cutting-edge mobile hardware platforms. The performance of this hardware in practical applications, as well as its architectural features and optimization techniques, will be thoroughly covered. DNN-based methods for tasks like image segmentation, object recognition, and image classification have been extensively reviewed in the literature [3]. To gain a comprehensive understanding of the current state and to foster further developments in computer vision, researchers must carefully review the body of work on hardware accelerators.

2. Hardware accelerators

2.1. Graphics processing units (GPUs)

The original purpose of GPU development was to speed up graphics processing. For jobs like integrated transform, lighting, and rendering, a GPU is made expressly for such purposes. In addition to being a potent graphics engine, a modern GPU is also a highly parallelized compute engine that can perform computations in enormous parallel with tremendous memory bandwidth and throughput. GPGPU, or general-purpose graphics processing unit, is a technology that extends the capabilities of GPUs beyond traditional graphics rendering applications. It is particularly useful for speeding up data-intensive tasks like machine learning and scientific computing.

GPGPUs are composed of thousands of in-order cores that operate at lower frequencies and rely on smaller-size caches, while multicore CPUs are often out-of-order, Mult instructional processors. High-performance GPU-accelerated programs using parallel programming can be created using a variety of development platforms, like Compute Unified Device Architecture (CUDA). Common applications include GPU-accelerated embedded devices, data centers, high-performance servers, etc.

Numerous hardware suppliers have developed GPUs, with Nvidia, Intel and AMD/ATI emerging as dominant players in the market. These GPUs cater to a wide array of specialized applications. Recent years have seen the rapid expansion of neural network technology, which has greatly benefited GPU computing advancement and produced many noteworthy improvements.

For instance, CNNs have multiple ways to take advantage of algorithmic parallelism [4]: (i) Parallelize matrix convolution operations using kernels; (ii) Independently carry out different subsampling and pooling procedures; (iii) To simultaneously activate every neuron in the fully connected layer, employ binary tree multipliers. Because of their strong floating-point computing capabilities and effective parallel processing architecture, GPGPUs are seen to be the ideal choice for speeding up deep learning. Being able to handle copious volumes of data concurrently enhances the training and reasoning speed of deep neural networks, particularly when handling intricate models and large-scale datasets.

Additionally, new software libraries and hardware accelerators keep coming out as deep learning applications get more complicated and diversified. TensorRT from NVIDIA, for instance, offers additional reasoning optimization, and Tensor Processing Unit (TPU) of Google is a hardware accelerator designed specifically for deep learning applications, which has been developed into the fourth generation TPU v4 [5]. These developments have aided in the rapid advancement of the field of deep learning and further solidified GPGPU's essential role in it.

It's probable that researchers have overlooked actual GPU usage as a measure of network quality in recent advances in neural network design paradigms [2]. GPU training speed and maximum batch size are frequently disregarded, even though inference speed is sometimes assessed. However, speed and largest batch size are all important considerations in real-world applications. Low GPU utilisation can lead to wasted resources, increasing training time and cost. Therefore, increasing GPU utilisation not only speeds up the model training process, but also reduces energy consumption and operational costs.

A model called TResNet improves on these aspects by carefully designing some of these factors [2]. TResNet employs a number of architecture-specific optimisations and engineering techniques to maximise GPU utilisation and throughput. It is shown that on the ImageNet dataset, TResNet provides better GPU throughput than existing models on the top-1 accuracy curve. As shown in Table 1, the performance metrics of the various ResNet50 and TResNet versions (M, L, and XL) are compared in the table. ResNet50 is faster in training than TResNet, but TResNet is more accurate overall. Specifically, TResNet-M, TResNet-L, and TResNet-XL had greater Top-1 accuracies than ResNet50 (79.0%), at 80.8%, 81.5%, and 82.0%, respectively.

Furthermore, on three widely-used downstream single-label categorization databases, it attains a new maximum accuracy. This demonstrates that the TResNet model not only increases accuracy and efficiency by making optimal use of hardware resources, but also demonstrates efficient solutions in real-world applications, further illustrating the importance of fine-grained architectural design and targeted optimisation for models to handle complex tasks.

Table 1. Compared to ResNet50, TResNet models precision and GPU throughput on ImageNet [2]

Models	Top Training Speed (img/sec)	Top Training Speed (img/sec)	Ma Train Batch Size	Top-1 Accuracy.(%)
TResNet-M	730.00	2930.00	512.00	80.80
TResNet-L	345.00	1390.00	316.00	81.50
TResNet-XL	250.00	1060.00	240.00	82.00
ResNet50	805.00	2830.00	288.00	79.00

2.2. Field-programmable gate arrays (FPGAs)

GPUs are popular for DNN hardware acceleration and provide very fast throughput. However, they are not as suitable for energy-constrained applications like Internet of Things devices because to their enormous power consumption. DNN acceleration is therefore shifting to different approaches based on FPGAs that consume less energy. FPGAs provide flexible parallel processing capabilities, adaptable data types, and particular to the application hardware designs, making it easy to adapt to the latest DNN models, which are characterized by higher sparsity and compact network architectures.

FPGAs have been more and more popular in deep learning applications in recent years as their benefits in high-performance, low-power computing have become more apparent. FPGAs' programmability enables them to be tuned for certain DNN models, maximizing hardware utilization and minimizing power consumption. Because of their adaptability, FPGAs are especially well suited for embedded systems and Internet of Things (IoT) devices, where responsiveness and energy efficiency are crucial.

Furthermore, FPGAs provide dynamic reconfiguration, which enables runtime reprogramming of the hardware to meet varying workload and optimization requirements. This provides FPGAs with a notable edge when managing various DNN jobs.

An FPGA is composed of numerous types of programmable logic blocks connected by a hierarchy of adjustable interconnects. Block RAMs, LUTs, DSPs and other key elements like flip-flops are common components of modern FPGAs. Fig. 2 depicts an ordinary FPGA architecture for DNN implementation. [6].

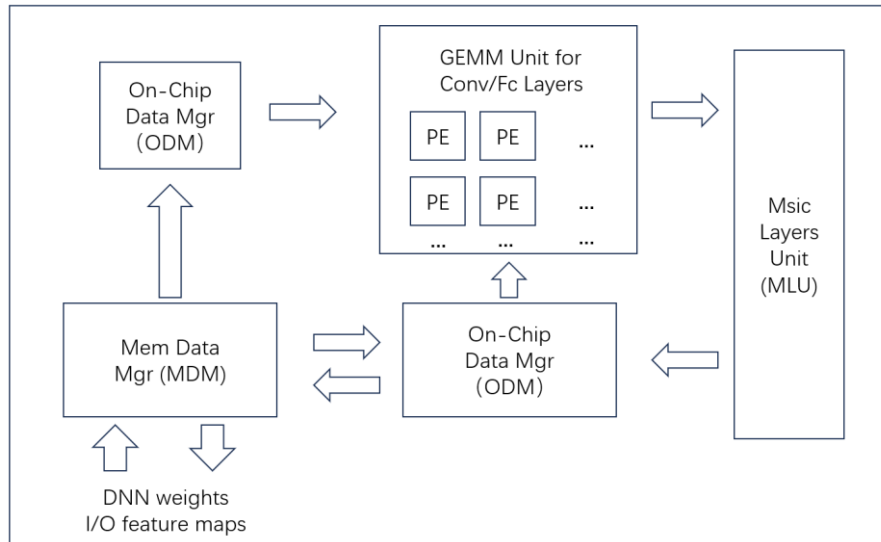


Figure 2. A concrete instance of a FGPA architecture used to achieve DNNs [6]

A DNN model is run through the FPGA architecture in several stages, as seen in Fig. 2. The weights and input and output functions feature maps of the DNN are provided by the memory data manager (MDM) and are first transferred to the on-chip data manager (ODM); then the generalized matrix multiplication (GEMM) unit receives the data and computes the convolutional and fully-connected layers, with multiple processing units (PEs) performing the computational tasks in parallel; the results are again passed through the ODM to the Miscellaneous Layer Unit (MLU) for the processing of other types of layers, and finally the results are output.

In real-world applications, DNN models are typically trained or optimized on high-performance computing platforms (such as GPUs), while DNN inference is mostly carried out on input data using FPGA accelerators based on pre-trained DNN models. GPUs may greatly accelerate the DNN training process because of their high bandwidth and powerful parallel processing capabilities, particularly when working with large-scale datasets and sophisticated models. FPGAs are especially well-suited for edge computing and real-time applications because they have low power consumption, great energy efficiency, and minimal inference latency when the DNN model is deployed to an FPGA for inference after training is finished.

The use of High-Level Synthesis (HLS) tools, which can automatically compile DNN algorithms described in a programming language into a Hardware Description Language (HDL) and generate low-level hardware configurations for use on FPGAs, has been shown to simplify the process of porting DNN models from GPUs to FPGAs. This method significantly lowers design complexity, enabling more developers to profit from FPGAs.

Strategies for accelerating DNNs with FPGAs include the use of highly parallel architectures to process multiple neurons and layers simultaneously, as well as the optimisation of specific computations such as convolution operations and matrix multiplication through customized hardware designs to improve computational efficiency and reduce latency. The flexibility of FPGAs allows for specialized optimization of different models to achieve optimal performance. A hardware accelerator proposed in the literature effectively reduces the complexity of the design by merging and reusing certain types of data flow and control logic [7]. In addition, it reuses general-purpose MAC (multiply-add operations) units when performing convolutional operations, thus minimizing resource consumption. This approach not only improves the utilization of hardware resources, but also reduces development cost and time.

In computer vision, hardware/software co-design approaches are widely used for DNN acceleration. For example, to accelerate the popular lightweight object detection neural network Tiny YOLOv3 [8], the researchers offloaded the most complex convolutional operations to be executed on a parallel pipeline hardware accelerator by analyzing the complexity of each operation. In addition, they effectively reduced the running cost by fusing the batch normalization layer into the convolutional layer, a process that only requires a one-time reparameterization when offline. This approach not only improves inference speed but also reduces energy consumption, demonstrating the great potential of hardware/software co-design for efficient DNN inference.

2.3. Application-specific hardware accelerators

To address a particular processing demand, a system for computation is usually heterogeneous, with a variety of processors included in it, such as CPUs that have been combined with other processors. Application-specific coprocessors are processors that assist CPUs. There are numerous specialized hardware units that are made expressly for neural network and/or other AI applications, in addition to well-known coprocessors like GPUs and FPGAs. These hardware units can be either standalone devices or coprocessors.

Google created the Tensor Processing Units (TPUs) [9,10], an ASIC with specificity made specifically for the TensorFlow framework and neural networks. Large amounts of low-precision (e.g., 8-bit) data are its target. Numerous Google apps, like AlphaGo and the search engine, are already powered by it [10]. The goal of Nervana neural network processor (NNP) created by Intel [11] is to maximize the efficiency of its fundamental hardware components while also offering the necessary flexibility for deep learning primitives.

Large-scale neural network models may be processed quickly and effectively by Google's recently released fourth-generation TPU (TPU v4) [5], which is intended for deep learning applications. The TPU v4 is especially well-suited to managing challenging computer vision tasks like image identification and target detection owing to its exceptionally high throughput and low latency.

In addition, graph core's IPU (Intelligent Processing Units) also excel in computer vision tasks. The architecture of IPU allows for massively parallel computation on-chip, enabling efficient execution of CNNs and other models. the IPU can dramatically increase inference speed and power efficiency through high-bandwidth communication between processing elements.

In addition to increasing efficiency and performance, the use of these ASIC hardware accelerators in computer vision neural networks stimulates the creation of new models and algorithms that enable the execution of increasingly difficult and precise vision tasks. These developments imply that ASICs will remain essential to the development of deep learning and artificial intelligence.

3. Summary

In summary, this paper provides a comprehensive analysis of three key types of hardware acceleration architectures for deep learning in computer vision: GPUs, FPGAs, and many other hardware accelerators tailored for certain uses. Each architecture offers unique advantages and tradeoffs for different aspects of the deep learning task.

Originally designed for rendering graphics, GPUs have evolved into powerful parallel processors ideal for deep learning due to their high memory bandwidth and efficient parallel computing capabilities. They excel in tasks that require extensive matrix operations and large-scale data processing, making them the preferred choice for training complex models and performing inference on massive datasets. However, GPUs have high power consumption and may not be the most efficient choice for real-time or edge computing applications.

FPGAs provide a middle ground with their reconfigurability and energy efficiency. They are well-suited for deploying pre-trained models in real-time applications due to their low power consumption and customizability for specific tasks. they can achieve high-performance inference by exploiting parallelism at the hardware level, but typically lag behind GPUs in terms of training speed. The

complexity of programming and optimizing FPGAs remains a major barrier to their widespread adoption.

Specialized integrated circuits (including dedicated processors such as TPUs and Intel Nervana NNPs) can provide high efficiency and performance for specific deep learning tasks due to their tailored design. However, they lack flexibility and adaptability because they are designed for specific applications and cannot be reprogrammed, making them less versatile than GPUs and FPGAs.

The future of DNN hardware acceleration is likely to involve hybrid approaches that combine the advantages of different architectures. For example, integrating a GPU for training with an FPGA or ASIC for inference could provide a balanced approach that maximises performance and efficiency. Advances in software, such as improved compilers and advanced synthesis tools, will also play a critical role in streamlining the development process and making these technologies more accessible. Overall, each hardware acceleration architecture has its own unique strengths and weaknesses. By utilizing technologies that are appropriate for a given task, and by continuing to innovate in both hardware and software, the area of computer vision can make significant advances in performance, efficiency and scalability.

References

- [1] Bianco, S.; Cadene, R.; Celona, L.; Napoletano, T. Benchmark Analysis of Representative Deep Neural Network Architectures. *IEEE Access*. 2018, 6, 64270 – 67277.
- [2] Ridnik T, Lawen H, Noy A, et al. Tresnet: High performance gpu-dedicated architecture [C]//proceedings of the IEEE/CVF winter conference on applications of computer vision. 2021: 1400 - 1409.
- [3] Feng X, Jiang Y, Yang X, et al. Computer vision algorithms and hardware implementations: A survey[J]. *Integration*, 2019, 69: 309 - 320.
- [4] Magnus Halvorsen, Hardware Acceleration of Convolutional Neural Networks, MS thesis, Norwegian University of Science Technology, 2015.
- [5] Jouppi N, Kurian G, Li S, et al. Tpu v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings[C]//Proceedings of the 50th Annual International Symposium on Computer Architecture. 2023: 1 - 14.
- [6] Eriko Nurvitadhi, et al., Can FPGAs beat GPUs in accelerating next-generation deep neural networks? in: *International Symposium on Field-Programmable Gate Arrays*, 2017.
- [7] Wu X, Ma Y, Wang M, et al. A flexible and efficient FPGA accelerator for various large-scale and lightweight CNNs [J]. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2021, 69 (3): 1185 - 1198.
- [8] Redmon J, Farhadi A. Yolov3: An incremental improvement [J]. *arXiv preprint arXiv: 1804.02767*, 2018.
- [9] Joe Osborne, Google's Tensor Processing Unit Explained: This Is what the Future Computing Looks like, *TechRadar*, 2016.
- [10] Norm Jouppi, Google Supercharges Machine Learning Tasks with TPU Custom Chip, *Google Cloud*, 2017.
- [11] Naveen Rao, Intel Nervana Neural Network Processor (NNP) Redefine AI Silicon, *Intel Website*, 2017.