A Practical Guide to Edge AI Power Efficiency

White Paper

HAILO

Empowering Intelligence

A Practical Guide to Edge AI Power Efficiency

Growing Demand for AI at the Edge

In the last decade we witness a growing need to process the massive amounts of data generated by devices and sensors[1]. This has given rise to new technologies such as deep learning. The use of machine vision and audio and speech processing (i.e. NLP, natural language processing) is booming across industries: adopted by governments and municipalities for smart city and defense applications; employed by businesses in robotics and autonomy to optimize and control manufacturing, logistics and retail operations for better business outcomes; used by consumers for personal security and utility in the home and on the go.

The more human tasks we transfer to computers and machines, the more abundant and pervasive the data and the wider the demand for neural network processing becomes. This surging demand for complex computing is no longer satisfied by centralized processing in datacenters. For many applications, running small AI tasks in datacenters is inefficient from a power and cost perspective due to the excessive use of compute, storage and bandwidth resources, as well as high latency and insufficient data privacy. Concerns such as these are propelling the fast evolution of deep learning outside of the datacenter – AI inference at the edge.

Edge AI first employed existing general-purpose processor architectures – CPUs and GPUs. However, as computing demands expand and multiply, their shortcomings – chiefly inefficiency in neural processing – become more and more

apparent. Under space, power supply, heat dissipation, cost and/or deployment limitations, they provide insufficient performance, thereby limiting possible edge applications[2].

Domain-specific AI processing has emerged to address this gap and is increasingly expanding what is possible at the edge[2]. The game-changer AI processors offer significantly higher cost and power efficiency. However, processors and architectures vary in the relative performance and efficiency gains they offer over customers' existing computing solutions. Thus, it is important to use task-specific, actual measurement to evaluate and compare them.

How Powerful Does Edge AI Need to Be?

To get a general sense of how much compute is needed at the edge, we first need to approximate a typical neural workload. The initial accelerated growth in the size of neural networks (number of parameters and operations, i.e. computing power and memory required) has slowed and stabilized. It appears that there is a range where, on average, the best tradeoff between neural networks' accuracy and size can be achieved. After a certain point, increasing network size offers diminishing returns on the accuracy front. For image classification of the ImageNet dataset, which is probably the most thoroughly studied task in modern computer vision, this range is around 20 million parameters and 4 GMACs, or 8 GOPs per frame. Despite gradual improvement in accuracy achieved by advances in the field, the tradeoff point stays the same[3,4].

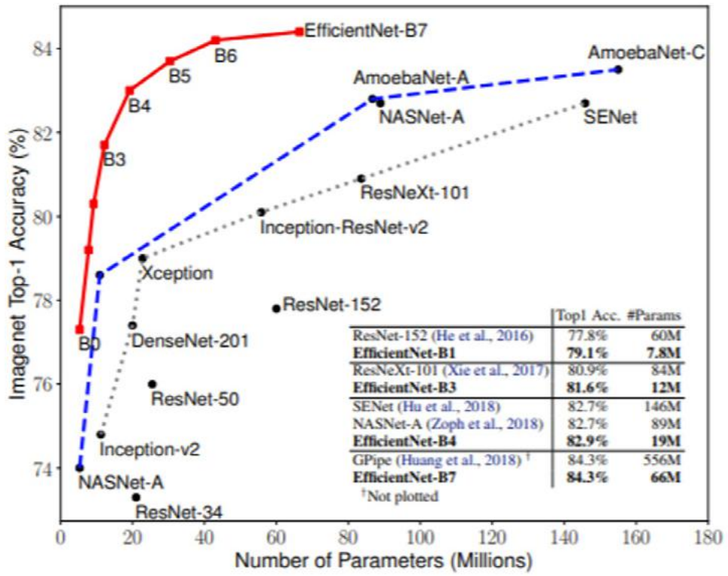

Figure 1: Model Size vs. ImageNet Accuracy[3]

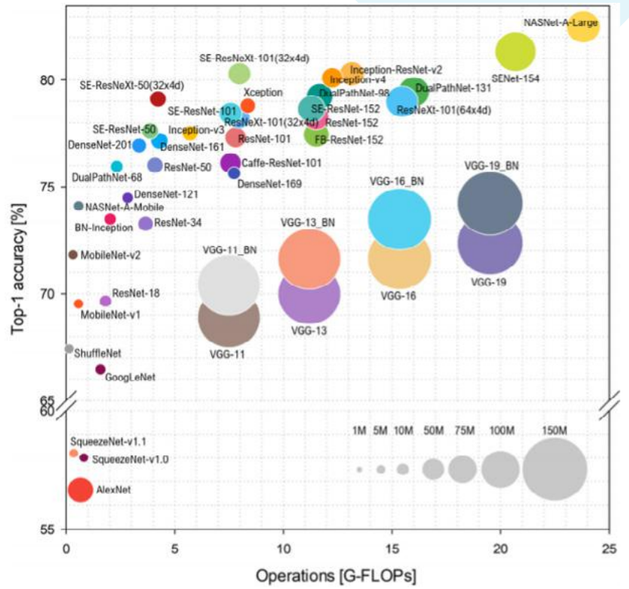

Figure 2: Model Accuracy vs. Number of Operations, ImageNet Dataset[4]

In this "golden" region of best accuracy-size and accuracy-compute tradeoffs, networks such as RegNetX-4.0GF, ResNext-50, Inception v3 and ResNet-50 can be found. This region represents the typical "engineering" choice for selecting the model size. It is therefore no surprise that ResNet-50, specifically, is very often used for AI performance benchmarks and as backbone in the development of new neural models. The computational cost of CNN backbones is (roughly) linearly dependent on the input resolution, and therefore the suitable metric to describe a backbone would be compute-per-pixel. Considering the data in the graph in ***Figure 1*** was obtained for input images of 224x224 resolution, we derive that the typical backbone for image processing tasks is roughly 150 thousand operations per pixel.

Next, we should consider the input and output of our typical neural model. In the case of neural processing, parameters like number of video inputs, input resolution, as well as desirable latency and/or frame rate are major considerations in determining throughput and power consumption. Thus, net processing **capacity (C)** in terms of operations per second (and when divided by 10^{12} results in terra operations per second or **TOPS**) is a given **workload** (i.e. neural network model or application, in our case a ResNet-50-like model) at a given **resolution** (Width x Height), and given **frame rate (FPS):**

C ~ FPS x W x H

To determine FPS and resolution, it is useful to consider some real-world use cases. To start with a relatively simple one, **instance segmentation for automated defect detection** is becoming common in many types of manufacturing. The task requires a high-resolution, high-frame rate camera on the fast-paced manufacturing or assembly line. Such cameras are usually 2-5 MP (mega-pixels) resolution and the speed of operation requires real-time processing input images – we can assume 30 FPS (frames per second). Roughly speaking, this one neural network and one high-resolution camera use case **requires 9-22.5 TOPS** of AI processing, depending on the camera's resolution[i].

Furthermore, there are many applications that require running several neural models on several inputs simultaneously. For instance, consider a camera poll with **3 cameras covering a busy city intersection.** Each camera covers 120° and is medium resolution (let us assume 640 x 480, i.e. 0.3MP). For best ROI, our intersection camera system will need to

i. On average, ResNet-50 requires roughly 150 thousand operations per pixel.
For 2MP camera: 30 FPS x each frame is 2MP = 60 MP/sec x 150k operations per pixel (ResNet-50 or similar) = 9 TOPS.
For 5MP camera: 30 x 5MP frame x 150k = 22.5 TOPS

perform several tasks, i.e. run several neural networks. For example, it will recognize the different intersection areas – road, sidewalk, crosswalk, building etc., detect vehicles and pedestrians and perform pose estimation to detect suspicious activity and emergency situations. These are 3 different neural models that need to run on all 3 video streams at the same time. For simplicity's sake, let us assume all of these models are roughly the size of the typical ResNet-50 and that we require real-time processing to keep up with the pace of traffic, which includes speeding cars that might run a red light. This camera system needs approximately **12 TOPS of AI processing performance**[ii]**.** This figure should be adjusted according to camera resolution – choosing to use 1.3MP or 2MP cameras in order to detect smaller objects or lower deployment density by covering longer distances will increase it significantly.

These are, of course, only rough approximations based on common, simplified requirements. These examples serve to illustrate the general understanding that **real-world deep learning-based applications require many TOPS of AI processing performance.** In reality, every application and environment pose different requirements and challenges. Many will require much more than what is described here: higher resolution, more complex neural model/s, more sensors etc.

ii. 3 cameras x 300k pixels per frame x 30 FPS x 3 NNs x 150k operations per pixel each (ResNet-50 or similar) = 12 TOPS

Why Power Efficiency is Important at the Edge

Unlike the virtually unlimited resources of the datacenter, edge deployments create unique constraints for computing hardware. Principal among these are device **power considerations**: power source, thermal limit, and energy limit.

First, the device's **power source is often limited**. The power supply of commercial devices is highly standardized (the power source has a given rating) and deviation from the standard is not a viable option for manufacturers. Single cameras common in the market can be limited to as little as 3W. Here are several examples of power supplies in real-world devices, which are common targets for integrating AI capabilities:

- Typical IP-security cameras used in homes and offices are designed for 5V or 12V input and consume under 4W (for example, see this reference design[iii]).
- A domestic security CCTV camera consumes on average about 2.5W-3W[iv, 5].
- The more advanced and powerful camera option – a PoE (Power over Ethernet[v]) camera, is usually limited to just under 13W[v].

iii. See for example this reference design from TI: www.ti.com/lit/ug/tidu188/tidu188.pdf?ts=1607328084344&ref_url=https%253A%252F%252Fwww.google.com%252F

iv. According to a a 2019 survey, a domestic CCTV system that includes a DVR (digital video recorder) and 4 cameras consumes, on average, up to 24W. This would mean, roughly speaking, 2.5-3W per camera and 10-15W for the DVR

v. IEEE 802.3: standard IEEE 802.3af and the newer IEEE 802.3at standard

Even a high-power camera or an entire system cannot include a high-performance CPU core that consumes over 40-60W or a GPU-based AI processor that consumes up to 20-30W. The common IP camera cannot even fit a 5W-consuming processor.

Devices also have a **thermal limit** – an operating temperature limit that prescribes how much heat its components can emit. Exceeding it will lead to system failure (shutdown or meltdown), so the device has a TDP (thermal design power), which is the maximal heat the system can dissipate. Common provisions include heatsinks (passive cooling) and fans (active cooling). Regardless of the cooling solution, the ability to dissipate heat is limited, and therefore the more efficient the processor is, the more it can do without reaching the TDP.

Finally, the edge often has an **energy limit**, as some edge devices are battery operated, which limits their available power supply per battery charging cycle. With a finite amount of power available in each cycle, how much power the device consumes translates into the device's life span. The more power it uses, the shorter the duration between recharges, and the more the device's utility is hindered.

Let us consider a standard laptop, typically with a 50Wh power supply. In active use the laptop will require a recharge in less than 2 hours. For a compute-intense task like video streaming and editing, for example, it might be even sooner. However, if we were to offload the task's neural processing from the CPU and GPU (general-purpose compute elements, which were not meant for deep learning operations), we could achieve significant power savings and prolong the charging cycle.

Figure 3 breaks down power consumption for such an active use scenario, showing just how much power is dedicated to compute (CPU and Graphics (GPU) – 60%.

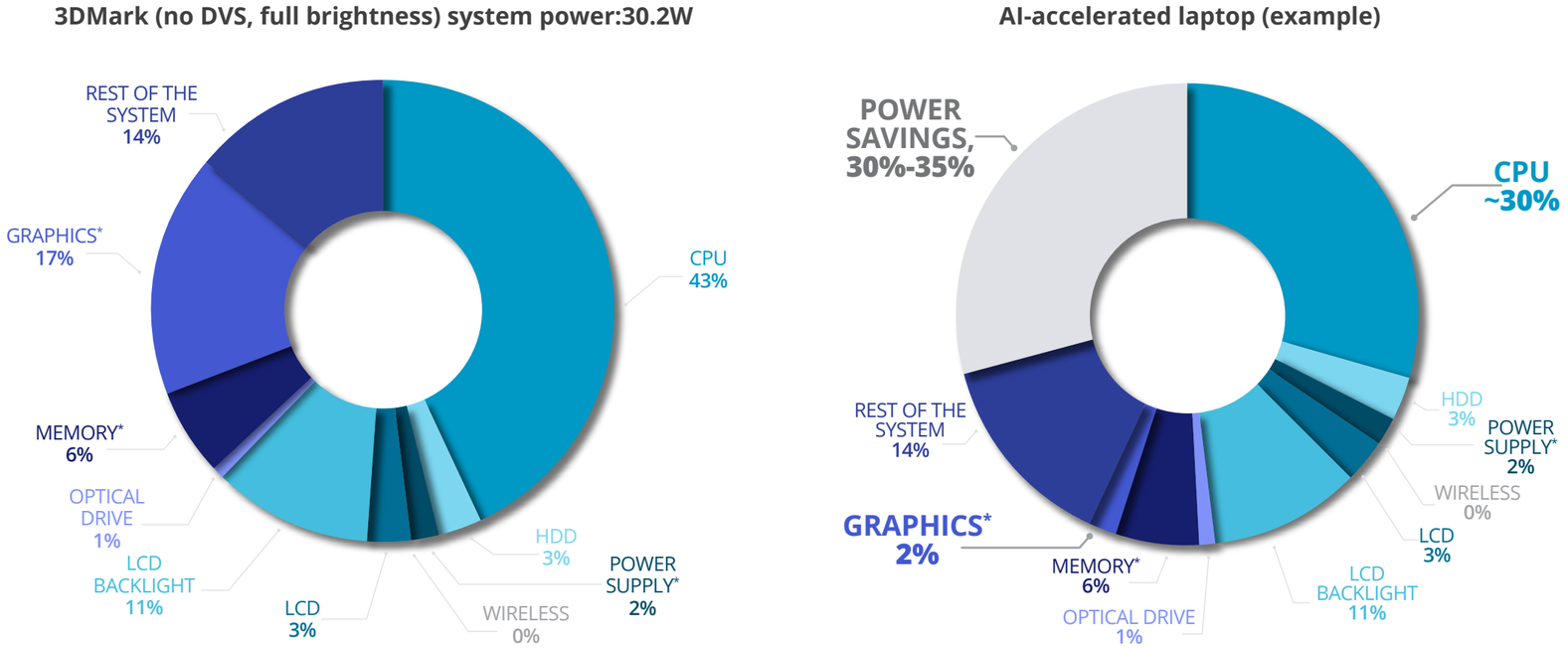

Figure 3: Laptop Power Consumption Breakdown in Active Use – Non-Accelerated[7] vs. Accelerated

We know that a leading accelerator can provide at least x10 the power efficiency of a GPU[vi] and that a GPU is generally at least x6 more efficient than a CPU for neural tasks. Assuming that almost all the graphics processing and about 1/3 of the CPU activity are used for a neural workload, we can reduce the GPU power consumption from ~17% to ~2% (of overall consumption) and cut CPU consumption by almost the whole 1/3 (reduce by ~13% to ~30% of overall power consumption).

Thus, the compute power consumption is cut by about 1/2 – from 60% down to 30%-35%, and overall power consumed for the laptop decreases by close to a 1/3. This extends the recharge cycle for the 50Wh battery by 50%, adding as much as a whole hour of work (total 3 hours vs. original 2 hours).

In traditional processing terms, these power source, energy and thermal limitations translate to compromising function (i.e. which applications and system capabilities are possible) and utility (i.e. how convenient or limiting the use of the end device is). Many edge environments demand a better than traditional use of power and heat dissipation resources. Simply put, the edge processor must generate more performance for every unit of power consumed and heat generated.

vi. Based on Hailo-8 vs. Nvidia GPU-based products benchmarks: https://hailo.ai/product-hailo/hailo-8/#perfomance

Power Efficiency – Some Definitions

Energy efficiency indicates the amount of data that can be processed or the number of executions of a task that can be completed for a given unit of energy. For a given hardware implementation, consuming certain power (P) while executing the given workload, system efficiency is:

eff = C / P

From a thermal management perspective, the required thermal resistance is:

R = (Tj - Tc) / P

Or, in terms of efficiency:

R = (Tj - Tc) x eff / C

When:

P is the measured power

Tj – junction temperature, a property of the device defining the maximum allowed internal temperature

Tc – ambient or case temperature, a property of the product defining the maximum desirable temperature allowed for the case

Lower R is better, and these formulas demonstrate that the higher the power (P), the more challenging the thermal design is. As the efficiency (eff) goes down it limits the allowed capacity (which translates into input resolution, FPS, complexity of neural model etc.) for a given thermal resistance, which means that better efficiency is required to fit more capacity.

How Power Efficiency is Calculated

Power efficiency is measured as unit of performance per unit of power consumed. The most common is TOPS/W, but throughput/W, inferences per second (IPS)/W and FPS/W are also used very wildly. Manufacturers of AI processors often publish their chips' maximum (theoretical) performance and power consumption (or TDP). Armed with these readily available figures, many observers are quick to divide one by the other to find the processors' power efficiency (also referred to as rated power efficiency). This is wrong and often creates a misleading view of the processor's true capabilities.

The TOPS figure is usually a theoretical top limit, which implies 100% utilization of the chip's architecture. It is nearly impossible to reach, as it neglects the utilization aspect that is a result of the resource allocation throughout the execution. There will always be some degree of inefficiency, which is especially true for general purpose processors, the architectures of which have inherent bottlenecks running neural networks as they were designed for a very different workload. Different neural tasks have varying workloads that will require varying processing throughput. Most neural networks do not require the entirety of the chip's processing power to run well. Actual throughput for a given task will vary between architectures, software toolchains and even performance goals[vii]. These, and many other variations, make throughput (FPS and resulting TOPS) a point measurement that is meaningful only when it is attached to a defined task performed in known conditions.

Maximum power consumption, on the other hand, can realistically be reached and is meaningful as a top limit. However, power consumption also varies with task and measurement conditions and thus, in this context, it should also be used as a task-/goal-specific measurement. The combined measurement of throughput relative to power then, is only meaningful when attributed to a specific task performed on a specific setup. In other words, **there is no "general" power efficiency attributable to a processor.**

vii. For instance, some architectures allow a tradeoff of throughput, power consumption and latency, so the developer can choose to lower throughput to achieve lower latency on a limited power budget or to maximize FPS and + minimize latency by elevating power consumption

That said, a truly power efficient processor should be consistent in its efficiency, i.e. maintain the same level of power efficiency as workloads vary and scale up and down.

Moreover, throughput, not TOPS, is the more precise, real-world measurement. Throughput is the amount of data that can be processed in a given time period, i.e number of inferences in deep learning at the edge terms or FPS in vision processing terms. **Inferences or FPS per Watt for a given neural task or application is not only a more precise way of evaluating and comparing hardware[viii], but also a better understood real-world metric.**

AI processors currently on the market offer a range of capabilities, especially in terms of power efficiency. Though generally more efficient than general purpose processors, their relative advantage varies significantly, as Figure 4 demonstrates.

It is also worth considering the efficiency of different types of AI processors. For the most part, domain-specific processors should have better efficiency than heterogenous-compute SoCs. The former are designed for neural processing with architectures that are optimized for high efficiency. The latter usually have an efficient neural core whose neural efficiency is "weighed down" by the far less efficient CPU and/or GPU.

viii. "The number of operations per inference [e.g. the ResNet-50 model requires 6.98 giga operations per second. This number is multiplied by FPS/ inferences per second to generate a performance figure in TOPS] depends on the DNN model; however, the operations per joules may be a function of the ability of the hardware to exploit sparsity to avoid performing ineffectual MAC operations... the number of MAC operations and weights in the DNN model are not sufficient for evaluating energy efficiency... because the number of MAC operations and weights do not reflect where the data are accessed and how much the data are reused, both of which have a significant impact on the operations per joule. Therefore, the number of MAC operations and weights are not necessarily a good proxy for energy consumption."[8] (Sze, V. et al., 2020)

For heterogenous architecture processors, it is also often unclear how the performance and power have been measured for the published benchmarks, specifically whether the measured power and throughput refer to the dedicated AI core, the entire neural process (AI core plus some of the neural workload shouldered by the GPU and CPU cores) or the entire chip. To achieve an apples-to-apples comparison of AI capabilities, measuring power and throughput for the chip just will not do. It is necessary to measure for the entire neural workload, be it on a single core, or spread across several heterogenous ones.

REMARKS:

- *All performance results are for ResNet-50 at 224x224 resolution , batch=1.*
- *All performance results are INT8, except for Nvidia Jetson Nano (FP16, it doesn't support INT8) and Intel Myriad X (only FP16 benchmarks are available).*
- *The publicly-available benchmarks vary in methods and conditions (model versions, optimizations, hardware and setup, software toolchain capabilities, measurement tools and methods etc.), probably significantly. Thus, the above is not an apples-to-apples comparison.*

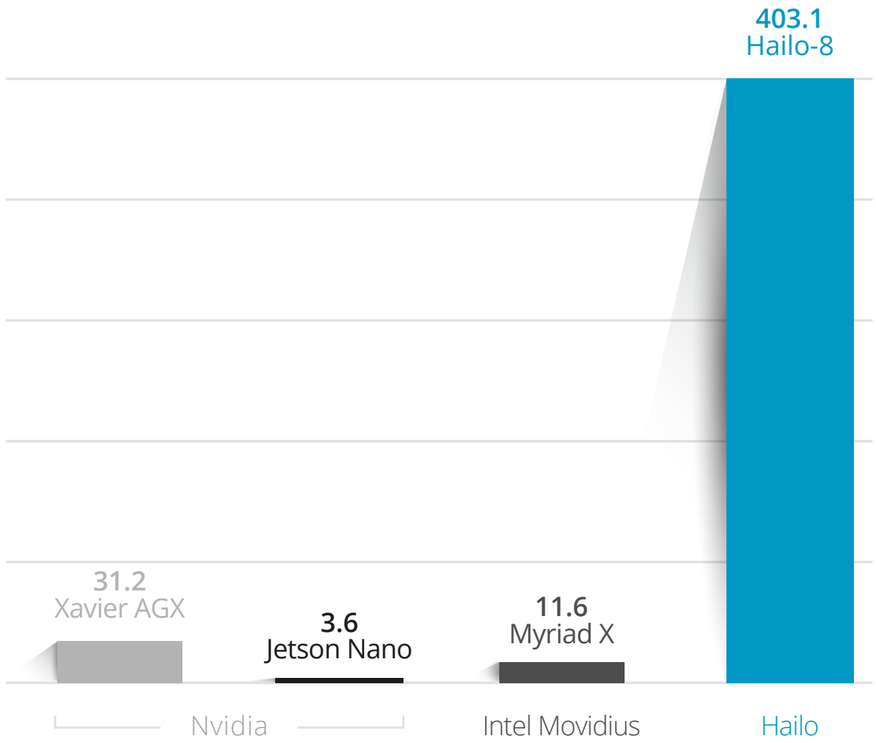

Figure 4: Comparison of Efficiency (FPS/W) in ResNet-50 Benchmarks[9]

Also important to take into account are “overhead” and limitations posed by additional, non-computational elements. In many architectures, we should account for on- or off-chip memory required for the neural processing as a major factor when it comes to throughput and power consumption. Other components that could impact throughput (or measurement thereof) are bandwidth and I/O limitations. They could be capping higher neural core throughput and measuring at the core will not reflect actual performance accurately.

In conclusion, **the best way to compare AI processor performance is to run and measure throughput and power consumption for the minimal system that is able to process the neural network, end to end,** including all its relevant components (mainly memory). It is important to evaluate the processors using **well-defined and preferably commonly used neural models,** eliminating any divergence in major configurable processing parameters (input resolution, precision, batch size, hardware set up, etc.) or accounting for it. While published benchmarks achieve the former, the latter is often neglected.

Conclusion

As the demand for smart applications at the edge grows, so does the need for efficient neural computation. Though there are multiple constraints at the edge and across verticals, those of power and heat dissipation are at the heart of the very feasibility of advanced deep learning at the edge. The edge device's limited resources call for maximum possible power efficiency in neural processing, which can be achieved using dedicated processing architectures. AI processors are increasingly made available on the market, but they vary widely in type, architecture and the power efficiency they can bring to various tasks.

To make an informed decision when choosing the most power-efficient chip, it is important to broadly consider architectural characteristics (e.g. heterogenous SoC vs. purpose-built AI accelerator) and performance limitations (e.g. maximum TOPS and power consumption). However, it is crucial to measure and compare neural processing power efficiency based on actual measured throughput and power on a specific task, as general processor power efficiency is not a reliable, meaningful measure of the chip's capabilities.

1. Coughlin, T (2017, September 27) Making Storage And Memory For Unstructured Data. Forbes. https://www.forbes.com/sites/tomcoughlin/2017/09/27/making-storage-and-memory-for-unstructured-data/?sh=3c9baafe3bb9
   2020 Cisco Networking Trend Report - https://www.cisco.com/c/dam/m/en_us/solutions/enterprise-networks/networking-report/files/GLBL-ENG_NB-06_0_NA_RPT_PDF_MOFU-no-NetworkingTrendsReport-NB_rpten018612_5.pdf
2. Jouppi N. P., Young C., Patil N., Patterson D. (2018) A Domain-Specific Architecture for Deep Neural Networks, Communications of the ACM, Vol. 61 No. 9, Pages 50-59, https://cacm.acm.org/magazines/2018/9/230571-a-domain-specific-architecture-for-deep-neural-networks/fulltext
3. Tan, M. and Quoc V. Le. (2019) EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. ArXiv abs/1905.11946. https://arxiv.org/pdf/1905.11946.pdf
4. Bianco S., Cadène R., Celona L. and Napoletano P. (2018) Benchmark Analysis of Representative Deep Neural Network Architectures. IEEE Access. 6. 64270-64277. 10.1109/ACCESS.2018.2877890. https://arxiv.org/pdf/1810.00736.pdf
5. Andrei, H., Ion, V., Diaconu E., Enescu, A. and Udroiu, I. (2019) Energy Consumption Analysis of Security Systems for a Residential Consumer. 1-4. 10.1109/ATEE.2019.8725002
   Also see example DVR spec: https://www.security.honeywell.com/uk/-/media/SecurityUK/Resources/ProductDocuments/HSFV-HQADVR-01-UK0617-DS-R-pdf.pdf
6. PoE (Power over Ethernet), Video Security Guide, 18/05/15. https://videosecurityguide.wordpress.com/2015/05/18/poe-power-over-ethernet/
7. Mahesri, A. and Vardhan V. (2004) Power consumption breakdown on a modern laptop. In Proceedings of the 4th international conference on Power-Aware Computer Systems (PACS'04). Springer-Verlag, Berlin, Heidelberg, 165–180. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.87.5604&rep=rep1&type=pdf
8. Sze, V., Chen Y. H., Yang, T. J. and Emer, J. S. (2020) How to Evaluate Deep Neural Network Processors: TOPS/W (Alone) Considered Harmful. IEEE Solid-State Circuits Magazine, vol. 12, no. 3, pp. 28-41. https://www.rle.mit.edu/eems/wp-content/uploads/2020/09/2020_sscs_dnn.pdf

9. Figures for the ResNet-50 benchmark efficiency comparison are based on the following data:

Vendor	AI Processor	Resolution	Precision	Batch	FPS	Power (W)	Inferences / W
Nvidia	**Xavier AGX**	224x224	INT8	1	358	11.5	31.2
Nvidia	**Jetson Nano**	224x224	FP16	1	36	10	3.6
Hailo	**Hailo-8**	224x224	INT8	1	1223	3.03	403.1
Intel Movidius	**Myriad X**	224x224	FP16	1	29	2.5	11.6

Sources for Nvidia published benchmarks and instructions:
Jetson Xavier AGX: https://developer.nvidia.com/embedded/jetson-agx-xavier-dl-inference-benchmarks, retrieved 10/03/2021
Jetson Nano: https://developer.nvidia.com/embedded/jetson-nano-dl-inference-benchmarks ; https://forums.developer.nvidia.com/t/deep-learning-inference-benchmarking-instructions/73291, retrieved 10/03/2021
For Intel Myriad X, figures are for resnet-50-TF run on NCS 2, retrieved 10/03/2021 from here: https://docs.openvinotoolkit.org/latest/openvino_docs_performance_benchmarks.html#resnet_50_tf For Hailo, figures are for ResNet_v1_50, platform 3.6.0 release

HAILO

Empowering Intelligence

www.hailo.ai