



**Region Halland**

# Sustainable Image Inference in Practice



White Paper within the project Data-Driven Organizations / SAIL  
Version: 1.0 | Date: 2025-12-16 | Author: Aixia AB and Region Halland



**Aixia AB** | +46 31 762 02 40



**Region Halland** | +46 35 13 48 00

## Table of Contents

<b>1. Executive Summary</b> .....	<b>4</b>
<b>2. Background</b> .....	<b>4</b>
<b>3. Objectives and Scope</b> .....	<b>5</b>
<b>4. Methodology</b> .....	<b>5</b>
4.1. <i>Standardized Performance Tests with ResNet50</i> .....	5
4.1.1. Raw Performance .....	5
4.1.2. Cost Efficiency .....	6
4.1.3. Ease of Use .....	6
4.2. <i>The Radiology Workflow</i> .....	6
<b>5. Hardware Overview</b> .....	<b>7</b>
<b>6. Limitations</b> .....	<b>7</b>
<b>7. Results</b> .....	<b>8</b>
7.1. <i>Standardized Tests</i> .....	8
7.1.1. Portability .....	8
7.1.2. Raw Performance .....	8
7.1.3. Cost Efficiency .....	9
7.1.4. Power Capping .....	11
7.2. <i>Radiology Workflow Results</i> .....	12
7.2.1. Portability .....	12
7.2.2. Performance .....	12
<b>8. Discussion and Analysis</b> .....	<b>13</b>
<b>9. Recommendations and Best Practices</b> .....	<b>14</b>
9.1. <i>Hardware Selection Should Be Driven by Actual Operational Needs</i> .....	14
9.2. <i>Optimisation Requires Dedicated Resource</i> .....	14
9.3. <i>Architectural Choices Affect Long-Term Maintainability</i> .....	15
9.4. <i>Centralised Versus Distributed Architecture</i> .....	15
9.5. <i>Energy and Operational Costs</i> .....	15
9.6. <i>Competence and Maintenance</i> .....	16
<b>10. Conclusion</b> .....	<b>16</b>
<b>11. References</b> .....	<b>17</b>



<b>Appendix A – The project “Data-driven organisations – Best practices for operationalisation of AI in Sweden</b> .....	<b>18</b>
<i>SAIL (Sustainable AI Infrastructure Lifecycle)</i> .....	19
<b>Appendix B – Glossary</b> .....	<b>20</b>
<b>Appendix C – Cost Efficiency Calculation</b> .....	<b>23</b>



## 1. Executive Summary

Organisations aiming to run AI inference in on-premises environments face a critical question: which hardware truly delivers value at a reasonable cost?

We tested twelve hardware platforms – from Raspberry Pi to NVIDIA DGX B200 – using standardized ResNet50 benchmarks and a real-world medical imaging workflow from Region Halland. Three key insights emerged:

- Raw hardware performance means little without software optimisation. Our B200 initially delivered 298 images per second, but with NVIDIA DALI optimisation, performance increased to over 30,000 images per second. The bottleneck was in data feeding, not the GPU.
- Some real-world workflows are CPU-bound. When large parts lack GPU support, expensive GPUs become unnecessary – existing VM infrastructure was fully sufficient for our use case.
- ARM-based devices offered the best energy efficiency but required extensive customisation. Several critical components lacked ARM support entirely.
- Power capping delivers substantial cost savings on high-end GPUs. Our H100 achieved 51% better cost efficiency with power capping – reducing power from 579W to 298W while losing only 22% throughput.

We recommend starting with a thorough workflow analysis. For most image analysis needs, mid-range GPUs such as the RTX A10 or L4 are sufficient. Data-centre systems are justified only under sustained heavy loads. Allocate budget to software optimisation as well as to hardware – a well-tuned mid-range GPU frequently offers better value than an underutilised flagship model. Sustainable on-prem AI inference does not necessarily require top-tier hardware. Match infrastructure to actual needs and invest smartly in optimisation for cost-efficient AI deployment.

## 2. Background

Building a sustainable AI lifecycle within on-premises infrastructure requires a careful evaluation of different hardware platforms. The choice of hardware not only affects system performance but also has a decisive impact on power consumption, maintenance needs, and overall operational costs. Moreover, hardware selection can fundamentally influence how quickly and smoothly an organisation can establish a functional inference environment.

As part of the national initiative Data-Driven Organisations (DDO), Aixia AB has assessed hardware platforms ranging from compact edge devices to full-scale data centre systems. The goal has been to evaluate both performance and sustainability potential in real-world applications.

In collaboration with Region Halland, we carried out a case study for computer-based analysis of vascular changes in CT (Computed Tomography) scans – a task that today requires time-consuming manual review by radiologists. Region Halland is aiming to implement a workflow consisting of a mix of ML- and algorithm-based analytics tools for this purpose. Our focus has been to examine how efficiently this workflow can be implemented across different hardware platforms, and how performance and energy consumption vary between systems. It is important to emphasize that the goal was not to evaluate the medical accuracy of the ML models, but rather to assess the practical feasibility, efficiency, and sustainability aspects of running ML inference locally in a healthcare environment. Hereafter, we refer to this workflow as the *radiology workflow*.

By combining standardized benchmarks with a concrete, healthcare-oriented case study, this white paper aims to provide organisations at the beginning of their AI journey with practical guidance on selecting hardware for AI inference. While the analysis is grounded in a specific medical application,



many of the technical and operational considerations are highly relevant for other organisations in the initial stages of their AI adoption.

This white paper is primarily intended for decision-makers and technical specialists evaluating the implementation of AI inference solutions in on-premises environments. It includes several technical concepts and terms central to understanding AI hardware and performance. For readers less familiar with these, a detailed glossary is provided in Appendix E, explaining all key technical terms.

### 3. Objectives and Scope

This white paper focuses specifically on image-based inference in environments ranging from edge devices to data centres. A separate white paper within the DDO project, titled *Benchmarking Large Language Models for ICD-10 Code Generation*, addresses LLM inference and its distinct hardware requirements.

Our primary objective is to highlight how different hardware platforms affect performance, energy efficiency, and operational viability in AI inference – particularly in contexts where both resources and technical support are limited. The ambition is to provide organisations with practical guidance on how to match suitable hardware to actual needs, avoiding overinvestment in capacity that remains underutilized.

We combine two complementary testing approaches: a standardized benchmark using ResNet50 for objective cross-platform comparison, and an applied case study based on Region Halland’s radiology workflow. The case study does not aim to evaluate medical accuracy but focuses instead on how effectively the radiology workflow can be implemented across different hardware platforms, considering performance, energy consumption, and technical complexity.

### 4. Methodology

Our methodology is based on two complementary testing approaches that together provide insight into AI performance in practical applications. The standardized benchmark (4.1) establishes what hardware is capable of under controlled, near-ideal conditions - a clean dataset, a well-supported model, and no external dependencies. The radiology workflow (4.2) tests what that same hardware can deliver in an operational environment, with legacy software, proprietary components, and real-world constraints. Comparing the two reveals not just ideal hardware performance, but also how real world constraints can sometimes limit how much of that performance is accessible in practice.

#### 4.1. Standardized Performance Tests with ResNet50

ResNet50 [4] was selected as the test model since it’s a proven and widely supported architecture that delivers reliable benchmark results. The tests were conducted using the Imagenette-320 dataset [5] (JPG images scaled to 320 pixels divided in 10 classes) to simulate realistic image analysis scenarios.

##### 4.1.1. Raw Performance

For each platform, we measured the raw performance by two distinct types of tests. First, we conducted a baseline test using ONNX Runtime FP32 with batch size 1 to establish a fair reference point that all but one system could handle without customisation.

We then optimised each platform by implementing TensorRT FP16 precision, determining optimal batch sizes, configuring parallel streams, and employing NVIDIA DALI-accelerated input pipelines where available.



Both low-latency scenarios (batch size 1) and maximum throughput configurations were measured to capture a range of use cases. Our raw performance measurements focused on throughput, in this case images per second).

### 4.1.2. Cost Efficiency

In addition to raw performance, we measured cost efficiency as images processed per Swedish Krona. This metric was calculated by measuring average power consumption during inference and converting it together with the throughput metric using formulas detailed in Appendix C.

Cost efficiency testing was conducted in two phases across our hardware range. First, we tested all twelve platforms at their default power settings to establish a comprehensive baseline of energy efficiency across different hardware categories – from edge devices to data centre GPUs.

Secondly, to explore the potential for further efficiency gains, we conducted power capping tests on a representative subset of five GPUs spanning different market segments: three data centre GPUs (H100, B200, A100), one workstation GPU (RTX A10), and one gaming GPU (RTX 3060).

Power capping involves imposing constraints on the maximum power consumption threshold of the GPU. This approach aims to optimize power efficiency by exploiting the non-linear relationship between power consumption and performance – whereby increased power allocation does not necessarily yield proportional performance gains [1, 2].

For each selected GPU we systematically tested different power cap levels to identify the optimal setting that maximized energy efficiency (images per Swedish Krona) while maintaining acceptable performance levels. This iterative approach allowed us to quantify the trade-off between throughput reduction and energy savings for different hardware categories.

### 4.1.3. Ease of Use

We also evaluated portability and implementation complexity for each platform. We documented the challenges encountered when deploying across different architectures – particularly the contrasts between x86 systems and ARM-based devices – as well as dependencies on proprietary tools and formats. This analysis provides valuable insight into the practical effort required to operationalize inference on each hardware platform, an often-underestimated factor that can significantly affect total project cost and time-to-deployment.

## 4.2. The Radiology Workflow

The synthetic benchmark was complemented by a real-world radiology workflow currently under development for Region Halland. This workflow mirrors authentic healthcare processes and illustrates how different system components interact in practice.

We made some small modifications to the workflow to make it handle a de-identified DICOM (Digital Imaging and Communications in Medicine) dataset. The workflow begins with the ingestion of the DICOM images. They then undergo extensive preprocessing, followed by analysis through multiple CNN models and analytical tools – some GPU-accelerated, others bound to CPU execution. The entire pipeline, from initial image loading to final reporting, is executed within a standardized Docker container, ensuring identical test conditions across all platforms.

This approach reveals real-world system bottlenecks that do not appear in synthetic, standardised benchmarks and demonstrates how hardware choices affect the entire analytical process from start to finish. Our measurements focused on total workflow execution time, overall energy consumption (in watt-hours), and resource utilisation across system components. Together, these metrics provided a holistic picture of each platform's practical performance.



## 5. Hardware Overview

The tests covered a selected range of hardware – from compact edge devices to high-performance data centre GPUs:

GPU	VRAM (GB)	System	Approx. Price (SEK)
<b>Hailo-8L</b>	N/A	Raspberry Pi 5 – ARM Cortex-A76 processor with 8GB system memory	1000
<b>RTX 3060</b>	12	Workstation with an AMD Ryzen Threadripper PRO 3955WX processor and 64GB of system memory	4000
<b>AGX Orin</b>	N/A	NVIDIA Jetson – ARM-based development platform with 32GB of shared memory	20 000*
<b>RTX 4070</b>	8	ASUS Zephyrus G14 laptop with an Intel i9-13900H processor and 32GB of system memory	25 000*
<b>L4</b>	24	Server with an Intel Xeon Gold 6530 processor and 128GB of system memory	30 000
<b>RTX 2000 Ada</b>	8	Lenovo ThinkPad P1 Gen 6 laptop with an Intel Core Ultra 9 processor and 64GB of system memory	40 000*
<b>RTX A10</b>	24	Workstation** with an AMD Ryzen Threadripper PRO 3955WX processor and 64GB of system memory	40 000
<b>GB10</b>	120	NVIDIA DGX Spark – Compact AI workstation with ARM Cortex-X925 processor and 120GB of shared memory	55 000*
<b>RTX A6000</b>	48	Workstation with an AMD Ryzen Threadripper PRO 3955WX processor and 64GB of system memory	60 000
<b>A100</b>	40	DGX system with an AMD EPYC processor and 1TB of system memory	100 000
<b>H100</b>	80	DGX system with an Intel Xeon Platinum processor and 2TB of system memory	350 000
<b>B200</b>	192	DGX system with an Intel Xeon Platinum processor and 2TB of system memory	600 000

Table 1: List of tested GPUs | \*Price listed for whole system since GPU is integrated | \*\*This is a data centre GPU, but we tested it in a workstation.

## 6. Limitations

Several methodological constraints should be considered when interpreting our results. The choice of ResNet50 as our benchmark model means that other model types could yield different relative



rankings between platforms. Results are also influenced by natural measurement variations caused by factors such as data loading, thermal conditions, and background processes.

Our evaluation focused mainly on CUDA-compatible hardware. While alternative platforms such as AMD ROCm and Intel oneAPI are maturing rapidly, they were not included in this study.

The optimised benchmarks used FP16 precision across all platforms to ensure fair comparison. Newer datacentre GPUs such as the H100 and B200 support lower precision formats (FP8, FP4) that could deliver higher throughput, but we did not evaluate these options. Organisations with workloads that tolerate lower precision may achieve better performance than our results indicate.

We chose to optimise the standardised benchmark extensively but left the radiology workflow largely unmodified. While some parallelism is already in place, there is room for further performance improvements.

Finally, cost calculations are based on simplified pricing using Swedish electricity rates, detailed in Appendix C. A complete total cost of ownership analysis would require additional factors beyond direct energy consumption.

## 7. Results

In this section, we present the results from both the standardized ResNet50 benchmarks and the real-world radiology workflow, highlighting performance, portability, and energy efficiency.

### 7.1. Standardized Tests

We evaluated the standardized ResNet50 benchmarks across three key dimensions: portability of implementation, raw performance metrics, and cost efficiency.

#### 7.1.1. Portability

The implementation of the benchmark tests on x86 and CUDA-compatible systems was carried out without major challenges for either ONNX Runtime or TensorRT. Thanks to the extensive CUDA support across NVIDIA's entire product range – from edge devices to data centre GPUs – we were able to establish a consistent and highly portable test environment.

The ONNX format proved stable and universal – all x86 and CUDA-compatible systems tested were able to run the models directly via ONNX Runtime without any modifications. For TensorRT, we used a script that converted the model from ONNX format and optimised it for the target hardware. The conversion and execution process worked smoothly on all CUDA-based systems.

For the Raspberry Pi device equipped with a Hailo accelerator, the implementation was more complex compared to the CUDA systems. Hailo requires its proprietary HEF (Hailo Executable Format), which meant the model had to be rebuilt from scratch. The initial step required registration with Hailo to gain access to their SDK – without this tool, there is no alternative way to create HEF files.

This experience underscores a fundamental limitation: the system becomes entirely dependent on the vendor's proprietary ecosystem, which directly impacts both portability and long-term maintainability.

#### 7.1.2. Raw Performance

The baseline tests without optimisations (ONNX Runtime FP32) quickly revealed that even the most powerful systems performed far below their theoretical capacity. The B200 system achieved only 298 images per second, just slightly above a laptop with a GTX 4070, which reached 214 images per second in the same baseline test. Even after implementing an optimised TensorRT engine with FP16 precision, no significant performance improvement was observed. The bottleneck was identified in I/O



handling – the CPU was unable to preprocess and transfer data fast enough to keep the GPU fully utilized.

After extensive experimentation with different optimisation techniques, we found the solution in NVIDIA DALI [6]. DALI is a specialized library for GPU-accelerated preprocessing that shifts preprocessing tasks from the CPU to the GPU, effectively eliminating the traditional CPU–GPU data transfer bottleneck. With DALI implemented, we observed a dramatic improvement: GPU utilisation rose from 5% to over 85%, resulting in a 50× increase in throughput without any loss of accuracy. With these optimisations applied, the performance gap between GPU tiers became evident – for instance, the H100 and B200 delivered roughly 10× higher throughput than the mid-tier L4.

The Hailo platform theoretically supports batch processing, but in practice, we were unable to achieve acceptable accuracy. The best result was obtained with a batch size of 8, yielding 46% top-1 and 67% top-5 accuracy – levels insufficient for production use. It is possible that further optimisations could improve these results, but within the scope of our testing methodology, this was the highest accuracy we achieved.

An unexpected finding was that the NVIDIA RTX 3060 – a gaming GPU – delivered significantly higher raw throughput than both the A10 and A6000 in our specific test scenario. There may be optimisation methods we did not apply to the lower-performing GPUs; however, even if that is the case, it remains remarkable that the RTX 3060 achieved such strong performance – even in the baseline ONNX model test.

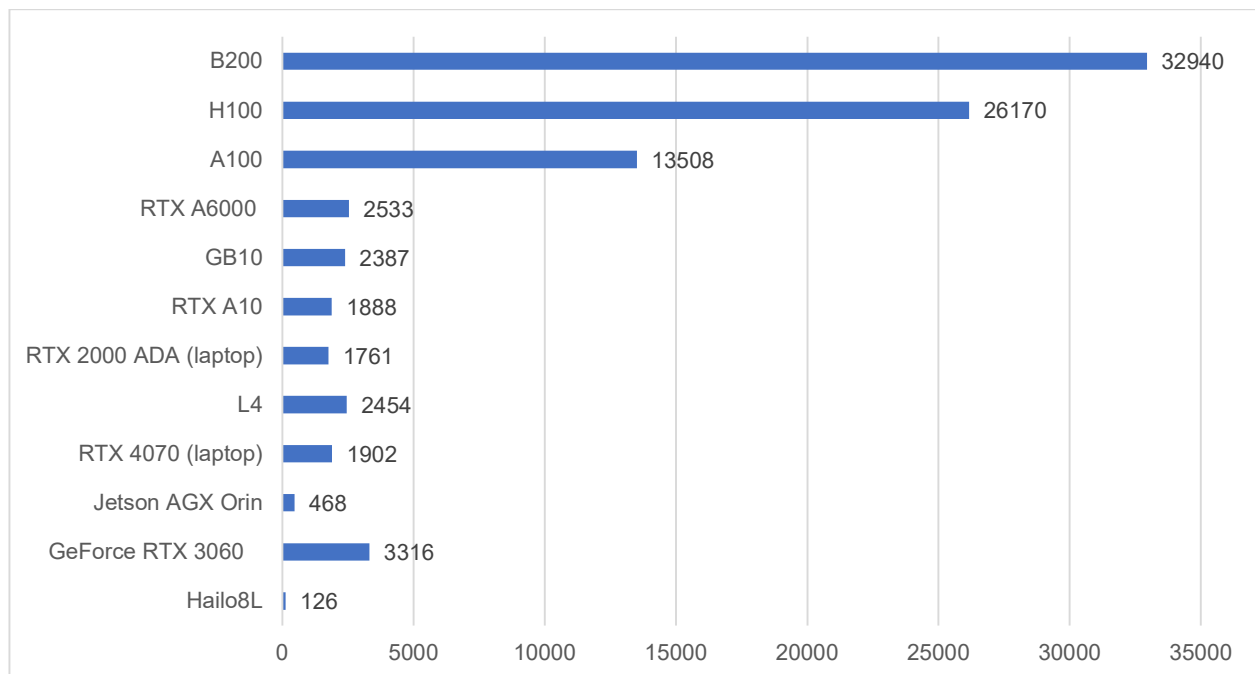


Figure 1: Images / Second

### 7.1.3. Cost Efficiency

The cost-efficiency analysis paints quite a different picture compared to the raw performance measurements. When calculating the number of processed images per Swedish Krona, the laptop systems, the edge device AGX Orin and the mini desktop system DGX Spark performed the best. The Raspberry Pi with a Hailo accelerator had the lowest power consumption, but its low throughput caused the energy efficiency to go down significantly.

## Sustainable Image Inference in Practice

The gaming GPU RTX 3060, which demonstrated surprisingly strong throughput performance, ranked much lower in cost efficiency due to its significantly higher power draw during inference compared to workstation GPUs such as the L4.

While the top-tier data centre GPUs dominated the throughput tests, they showed similar numbers as the laptops when it comes to energy efficiency – in other words, while being the most performant they were also the most power hungry.

It's important to note that the benchmark results are only showing the actual GPU power draw during inference. If we were to also factor in the acquisition costs, the idle power draw, cost of data centre cooling etc, the small devices such as the laptops and AGX Orin would be significantly more energy efficient than the top-tier data centre GPUs

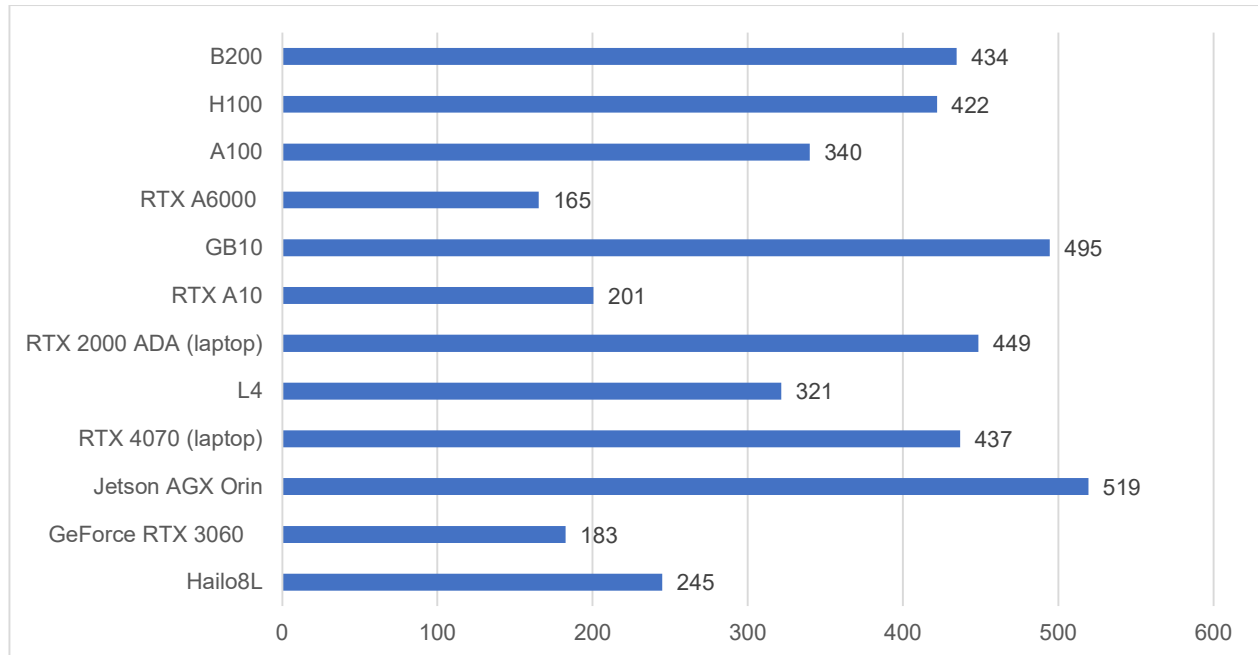


Figure 2: Million Images / Swedish Krona

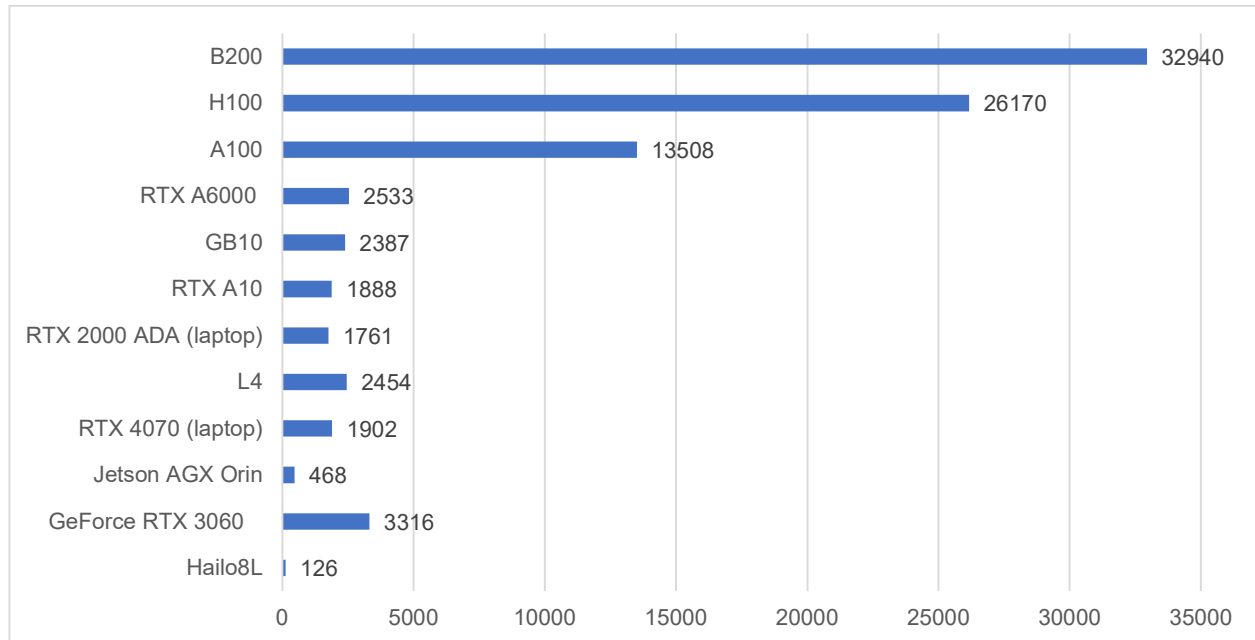


Figure 3: Effect While Idling (W)

#### 7.1.4. Power Capping

For the high-end datacentre GPUs, power capping demonstrated substantial benefits. The H100 achieved the most significant improvement, with cost efficiency increasing from 422 to 638 images per Swedish Krona – a 51% improvement – while reducing power consumption from 579W to 298W (a 48% reduction). The throughput decreased by only 22% (from 26,170 to 20,394 images / s), demonstrating the favourable trade-off between performance and energy consumption. The B200 showed similar characteristics, improving cost efficiency by 28% (434 to 556 images/SEK) with power consumption reduced from 707W to 485W (31% reduction) and a 12% throughput reduction.

The RTX A10 datacentre GPU showed a 6% decrease in cost efficiency (201 to 189 images/SEK), with power consumption reduced from 88W to 76W (14% reduction) and an 18% throughput decrease (1888 to 1537 images / s). This modest negative result suggests that the RTX A10 may already operate closer to its optimal efficiency point at default settings.

The GeForce RTX 3060 consumer GPU exhibited a dramatic 43% increase in cost efficiency (183 to 261 images/SEK) while reducing power consumption from 169W to 100W (41% reduction), with only a 16% throughput decrease. This suggests that consumer GPUs, which typically operate with higher default power limits optimized for gaming workloads, benefit substantially from power optimisation for sustained compute tasks.

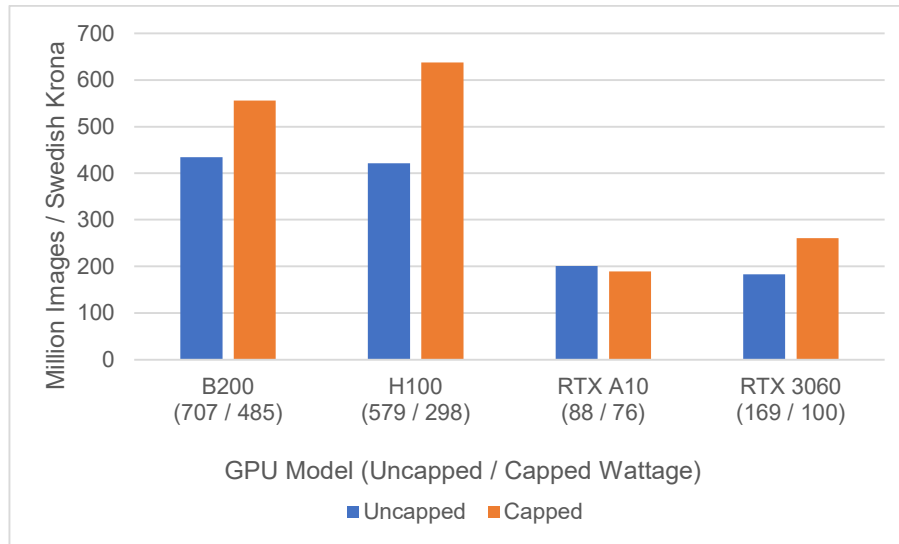


Figure 4: Million Images / Swedish Krona using default power settings VS power capped settings.

## 7.2. Radiology Workflow Results

The real-world radiology workflow provided insights into practical deployment challenges that synthetic benchmarks cannot reveal. We evaluated both the portability of the workflow across different architectures and the actual performance characteristics in a healthcare-oriented use case.

### 7.2.1. Portability

The Docker image containing the complete, pre-built radiology workflow ran seamlessly on all x86 systems. However, it could not be executed on ARM-based systems. The primary reason was that one core component of the workflow consisted of proprietary software compiled exclusively for x86, with no corresponding binaries available for ARM.

x86 and ARM are different processor architectures. Each requires software to be compiled specifically for it, producing different binary files. An x86 binary simply cannot execute on ARM hardware. For open-source software this means recompiling (porting) the software for the new architecture; for proprietary software where source code is unavailable, it means the software cannot be used at all.

Even the components that could be ported required substantial effort, as they relied on different drivers, libraries, and dependencies than their x86 counterparts. In many cases, the availability of precompiled components for ARM was limited, meaning that large portions of the workflow had to be rebuilt entirely from source.

### 7.2.2. Performance

Performance measurements yielded surprisingly consistent results – all testable systems completed the radiology workflow in 3–4 minutes. The only notable exception was the B200 system, which completed the workflow in 2.3 minutes.

Since the radiology workflow was primarily CPU-bound with only a few GPU-accelerated steps, we investigated the performance difference between running with CPU+GPU versus CPU-only. On a modern ThinkPad laptop equipped with an RTX 2000 Ada GPU, the difference was minimal – 3.2 minutes with GPU acceleration enabled versus 2.8 minutes without.

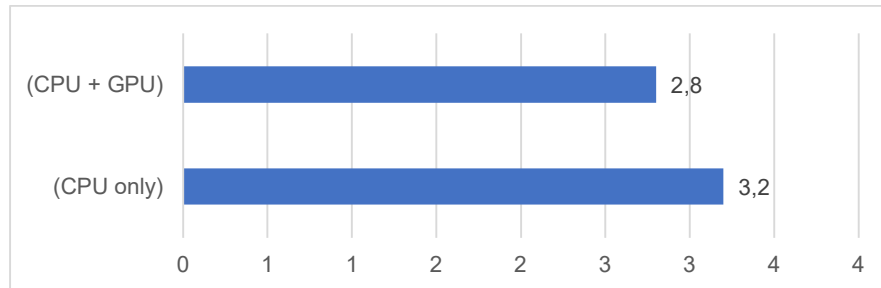


Figure 5: Time taken (minutes) to complete radiology workflow with combined CPU + GPU versus CPU only.

## 8. Discussion and Analysis

By conducting both a standardized test using the well-known ResNet50 model and a separate, domain-specific workflow, we gained valuable insights into the complex interplay between hardware, software, and operational context. The results clearly demonstrated that hardware selection is often more nuanced than it may initially appear.

The ResNet50 benchmark confirmed our expectations – small edge devices with low power consumption, both under load and at idle, offer the highest performance per invested cost. However, this assumes that the desired software stack supports the ARM architecture, which was not the case in our radiology use case. Even when ARM support is present, the implementation process becomes significantly more demanding, as libraries and other components are often less accessible than their x86 equivalents. Organisations must therefore carefully consider who will be responsible for building and optimising inference models and how comfortable those individuals are with developing custom solutions.

A key insight was that powerful and expensive hardware does not automatically translate to proportionally better performance. To unlock the theoretical capacity of the hardware, software optimisation is essential. We identified that bottlenecks often occurred in data processing and transfer, rather than in the inference process itself. Particularly for high-end systems, considerable effort was required to achieve acceptable GPU utilisation. Even when optimal performance was achieved, it is critical to question whether such capacity is truly needed. For example, in our 320-pixel image tests, the B200 system reached approximately 33,500 images per second – a throughput that very few use cases realistically require for inference workloads.

Our power capping experiments revealed another important dimension of hardware optimisation. The results demonstrated that high-end datacentre GPUs operate well beyond their optimal efficiency point at default settings. The H100's 51% improvement in cost efficiency through power capping – achieved by reducing power from 579W to 298W while losing only 22% throughput – illustrates the highly non-linear relationship between power consumption and performance [1, 2]. Similarly, the consumer RTX 3060 showed a 43% efficiency gain, suggesting that gaming GPUs with their performance-oriented default settings benefit substantially from power optimisation in sustained inference workloads. Interestingly, the RTX A10 showed minimal benefit, indicating it may already operate closer to its optimal efficiency point. These findings suggest that power capping deserves consideration as a standard optimisation technique, particularly for organisations running high-end GPUs under sustained loads where energy costs are a significant operational concern.

The radiology workflow also revealed minimal performance differences between hardware platforms. This was partly due to the lack of workflow-specific optimisation. Some degree of parallelism was already in place, with certain stages executed asynchronously, but total runtime could be reduced further. However, a major factor was that substantial portions of the radiology workflow lacked GPU

support, which explains the small observed performance variations. In medical contexts, established analysis tools are often used instead of the latest frameworks – largely because these tools have already been validated through extensive research studies [7], ensuring well-documented reliability and acceptance.

Region Halland emphasized that for CT image analysis, the daily processing volume is small, and the difference between one and five minutes of processing time has only marginal practical significance. In their context, an investment in high-end GPU hardware would therefore not be justified, even if further optimisation were possible. Since our tests showed minimal performance gains from GPU acceleration compared to pure CPU execution, we concluded that for this case a CPU-only device or virtual machine would make perfect sense, since it means less complexity.

Another choice to consider is that between centralised inference servers and distributed edge devices. This must be grounded in operational context rather than raw performance alone. Edge deployment becomes compelling when large data volumes create network bottlenecks or when time-critical applications – such as real-time surgical assistance – require resilience against network disruptions. However, these benefits must be weighed against operational complexity: centralized infrastructure offers significant advantages in maintenance, monitoring, and resource sharing, whereas distributed edge deployments require specialized expertise for installation and troubleshooting at each site, particularly for ARM-based systems [8].

Given Region Halland's modest processing volumes, non-critical timing requirements, adequate network infrastructure, and existing virtualized compute resources, a centralized approach using a standard virtual machine could be the most pragmatic choice. This minimizes both capital expenditure and operational complexity while fully meeting current needs.

## 9. Recommendations and Best Practices

Based on our findings from both the standardized benchmarks and the real-world radiology workflow, we have identified several key recommendations for organisations planning to deploy AI inference infrastructure. These guidelines aim to help avoid common pitfalls such as over-investment in underutilized hardware or underestimating the effort required for software optimisation.

### 9.1. Hardware Selection Should Be Driven by Actual Operational Needs

Map your real-world workflows before making any hardware decisions. Identify whether bottlenecks lie in GPU acceleration, CPU processing, or I/O handling by first testing on existing infrastructure or by temporarily renting a GPU instance in the cloud. As demonstrated by our radiology workflow, even simple CPU-based systems can sometimes deliver sufficient performance when the workload is not optimised for modern GPUs.

For organisations with limited inference volumes – such as Region Halland's radiology workflow – midrange GPUs like the NVIDIA L4 or edge devices such as the Jetson AGX Orin are often more than sufficient. High end GPUs are justified only under continuous high-load conditions or when millisecond-level latency is business-critical. If GPU acceleration does not provide significant benefits for a specific workflow, a CPU-based solution may be the simplest and most energy-efficient choice.

### 9.2. Optimisation Requires Dedicated Resource

Achieving theoretical performance requires substantial development effort. For systems like the B200/H100, specialized pipelines (DALI) were needed just to achieve basic GPU utilisation. Carefully evaluate whether such an investment is justified for your use case.



For most applications, ONNX Runtime or TensorRT provide sufficient performance, depending on the environment and requirements. ONNX Runtime offers portability and broad compatibility, while TensorRT delivers higher performance on NVIDIA hardware. Further optimisation should only be considered when these established tools fail to meet the necessary requirements.

### 9.3. Architectural Choices Affect Long-Term Maintainability

If your organisation relies on proprietary or legacy software, choose x86-based systems. Porting to ARM often requires significant redevelopment and, in some cases, may not even be feasible – as demonstrated in our radiology workflow.

Jetson devices offer excellent cost efficiency for well-defined, isolated inference tasks where the software stack can be fully controlled. However, note that integrating an ARM-based system always introduces some overhead compared to x86. It is therefore important to weigh whether the additional effort is justified by the potential benefits in energy savings, portability, or independence.

### 9.4. Centralised Versus Distributed Architecture

Consider whether a powerful central system or multiple smaller distributed devices best suits your organisation's needs. A central DGX or workstation GPU provides simpler administration and better resource utilisation under variable loads but introduces a single point of failure and often requires a dedicated server facility. Multiple edge devices or smaller GPUs increase redundancy and can reduce network load through local processing, but they add administrative overhead.

For organisations with geographically distributed operations or strict data privacy requirements, a distributed architecture may be necessary. In contrast, organisations with centralized IT operations and high data centre availability requirements often benefit from a powerful central solution supported by a robust backup strategy.

An important consideration is whether the organisation needs to train new models or fine-tune existing ones locally. Although this white paper focuses on inference, training imposes significantly higher hardware demands. If local training or fine-tuning is required, at least one high-memory workstation or a data centre GPU may be warranted, even if inference workloads are limited.

### 9.5. Energy and Operational Costs

Power consumption affects operational costs in multiple ways. Data centre GPUs have high idle power consumption, which can drastically affect total energy costs when utilisation is low. If investing in systems such as a DGX, it is therefore crucial to ensure that the system does not remain idle for extended periods throughout the day.

For organisations running high-end GPUs under sustained loads, power capping should be considered as a standard optimisation technique. Our testing showed that datacentre GPUs like the H100 can achieve significantly better cost efficiency through power capping, with modest throughput reductions. Consumer GPUs optimized for gaming workloads showed even more dramatic benefits, with the RTX 3060 achieving 43% efficiency gains. However, mid-range datacentre GPUs like the RTX A10 may already operate near their optimal efficiency point and show limited benefits from power capping.

The practical implementation of power capping is straightforward on NVIDIA hardware using `nvidia-smi` commands or management APIs, making it a low-effort optimisation for organisations with

appropriate monitoring infrastructure. The optimal power cap varies by GPU model and workload, requiring some experimentation to find the best balance between throughput and energy efficiency for your specific use case.

Systems exceeding 500 watts often require dedicated cooling and reinforced electrical infrastructure. These additional requirements should be included in the total cost calculation, as their cumulative impact can exceed the hardware cost over the system's lifecycle. Power capping can also reduce cooling requirements and extend hardware lifespan by lowering operating temperatures.

### 9.6. Competence and Maintenance

More advanced systems frequently require specialized expertise for operation and maintenance. Edge devices with proprietary ecosystems (such as Hailo) and high-performance data centre GPUs (such as the DGX series) demand significantly higher levels of technical skill compared to standard x86-based systems with well-established software stacks.

Organisations should assess whether they possess or can recruit the necessary expertise internally, or if vendor support agreements are required. These costs for competence and support should be included in the total cost of ownership, as they can become substantial over time – particularly for proprietary solutions where the organisation becomes dependent on a single vendor ecosystem.

## 10. Conclusion

Our study demonstrates that sustainable image inference can often be achieved without relying on the most advanced hardware. The decisive factor for practical value is not the hardware's theoretical capacity, but rather how effectively the software utilizes available resources. A well-optimised mid-range GPU can deliver better real-world performance and energy efficiency than an underutilized flagship model.

For CPU-bound workflows, such as Region Halland's CT analysis, investments in expensive GPU systems provide little to no practical benefit. In such cases, it is far more rational to focus on optimisation, automation, and efficient use of existing infrastructure.

At the same time, our tests have shown that edge devices and energy-efficient workstation GPUs offer an attractive balance between performance, cost, and energy consumption – provided that the software stack is compatible and properly configured.

The overarching message is that technology choices must be guided by operational needs, not by marketed hardware performance. By analysing workflows, understanding bottlenecks, and investing in software optimisation alongside hardware, organisations can achieve both high efficiency and long-term sustainability in their AI initiatives.

## 11. References

- [1] d'Aviau de Piolant, A., et al. (2025). "Improving energy efficiency of HPC applications using unbalanced GPU power capping." HAL Archives, INRIA.
- [2] Kasichayanula, K., et al. (2012). "Power aware computing on GPUs." Symposium on Application Accelerators in High Performance Computing (SAAHPC).
- [3] "NVIDIA AI Strategy: Analysis of Sustained Dominance in AI." (2025). Klover.ai. <https://www.klover.ai/nvidia-ai-strategy-analysis-sustained-dominance-ai/>
- [4] He, K., Zhang, X., Ren, S., & Sun, J. (2016). "Deep Residual Learning for Image Recognition." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778.
- [5] Howard, J. (2019). "Imagenette Dataset." <https://github.com/fastai/imagenette>
- [6] NVIDIA Corporation. "NVIDIA DALI Documentation." NVIDIA Developer. <https://docs.nvidia.com/deeplearning/dali/user-guide/docs/>
- [7] Chouffani El Fassi, S., Abdullah, A., Fang, Y., et al. (2024). "Not all AI health tools with regulatory authorisation are clinically validated." Nature Digital Medicine.
- [8] Intel Corporation. (2024). "How Edge Computing Is Driving Advancements in Healthcare." Intel. <https://www.intel.com/content/www/us/en/learn/edge-computing-in-healthcare.html>



## Appendix A – The project “Data-driven organisations – Best practices for operationalisation of AI in Sweden

This material has been produced as part of the Vinnova-funded project Data-driven organisations – Best practices for operationalisation of AI in Sweden (DDO), a project lasting just under two years with twenty participants from private sector, public sector, and academia. Together, they tackled issues concerning large- and small-scale operation of AI solutions and how to enable and use AI broadly across an organisation.

The work focused on three specific use cases: Local sustainable operation of AI, legal and technical prerequisites for effective infrastructure, and how to create the best conditions for keeping thousands of AI models in operation.

A compilation of all material produced within the framework of DDO is available on AI Sweden's website – <https://www.ai.se/en/project/data-driven-organizations-best-practices-ai-operationalization-sweden>.

The organisations that participated in DDO were:

- Aixia <https://aixia.se>
- Hewlett Packard Enterprise <https://www.hpe.com>
- Hopsworks <https://www.hopsworks.ai>
- IBM <https://www.ibm.com>
- Linköpings Universitet <https://liu.se>
- NetApp <https://www.netapp.com>
- Predli <https://www.predli.com>
- Proact <https://www.proact.se>
- RISE <https://www.ri.se>
- RedHat <https://www.redhat.com>
- Region Halland <https://www.regionhalland.se>
- Sahlgrenska University Hospital <https://www.sahlgrenska.se>
- Statistics Sweden (Statistiska Centralbyrån) <https://www.scb.se>
- The Swedish Tax Agency (Skatteverket) <https://www.skatteverket.se>
- Stormgrid <https://www.stormgrid.ai>
- The Swedish Transport Administration (Trafikverket) <https://www.trafikverket.se>
- Volvo Parts <https://www.volvogroup.com>
- Region Västra Götaland <https://www.vgregion.se>
- Santa Anna <https://www.santa-anna.se>
- AI Sweden <https://www.ai.se/en>

The project was funded by the participating organisations and Vinnova. AI Sweden is in part financed by the EU.



## **SAIL (Sustainable AI Infrastructure Lifecycle)**

This white paper is produced within the SAIL use case, which is part of the DDO project. SAIL focuses on building cost-efficient, environmentally sustainable, and operationally effective infrastructure that supports the entire AI lifecycle – from exploration and development to training, deployment, inference, and long-term operations.

The goal is to enable organisations to adopt and scale AI sustainably by reducing costs, minimizing environmental impact, and ensuring that AI systems can operate and evolve over time without unnecessary complexity.

Key areas of focus include:

- Designing scalable and flexible AI infrastructure that grows with organisational needs
- Exploring hardware, cloud, and modular/shared platform options
- Creating practical guidelines for long-term AI operations
- Optimizing resources and reducing technical and administrative overhead

The outcome will be a validated model and actionable recommendations that help organisations of all sizes and maturity levels build, operate, and evolve AI solutions in a sustainable, efficient, and future-proof way.

In addition to this white paper, SAIL has also produced the following white papers:

### **MLOps on-prem without Kubernetes – A Faster Path to Production**

Demonstrates how an efficient on-prem MLOps pipeline can be implemented without Kubernetes, emphasising simplicity, reproducibility, and rapid deployment to production.

### **The AI Implementation Spectrum – Strategies for Sustainable and Scalable Adoption**

Introduces a framework for understanding different maturity levels of AI implementation and how organisations can build scalable and sustainable strategies across the full AI lifecycle.

### **Benchmarking Large Language Models for ICD-10 Code Generation**

Evaluates different hardware and software configurations to identify the most efficient and sustainable setup for running large language model inference when generating ICD-10 codes from clinical notes.



## Appendix B – Glossary

Term	Description
<b>AI Inference / Inference</b>	The process of using a trained AI model to make predictions or analyse new data. Differs from training, which is the process of creating the model.
<b>ARM</b>	A processor architecture commonly used in mobile devices and edge computing. Offers high energy efficiency but may have limited software compatibility compared to x86.
<b>Batch Size</b>	The number of images or data samples processed simultaneously by the model. Larger batches can improve throughput but require more memory.
<b>Benchmark</b>	Standardized performance tests used to objectively compare different hardware platforms.
<b>Bottleneck</b>	The part of a system that limits overall performance, regardless of how powerful the other components are.
<b>CNN (Convolutional Neural Network)</b>	A type of neural network particularly effective for image analysis and pattern recognition.
<b>CPU-Bound</b>	When a workflow's performance is limited by CPU processing capacity rather than GPU capability.
<b>CT (Computed Tomography)</b>	A medical imaging technique that uses magnetic fields to create detailed images of the body's internal structures.
<b>CUDA (Compute Unified Device Architecture)</b>	NVIDIA's parallel computing platform that enables software to leverage GPU acceleration.
<b>DALI (Data Loading Library)</b>	NVIDIA's library for GPU-accelerated data preprocessing, which offloads data preparation from the CPU to the GPU to eliminate bottlenecks.
<b>Data Centre GPU</b>	High-performance GPUs designed for continuous operation in data centres, such as the A100, H100, and B200.
<b>DGX</b>	NVIDIA's line of integrated AI systems designed for data centre deployment, combining multiple GPUs with optimised networking and software in a single platform.
<b>DICOM (Digital Imaging and Communications in Medicine)</b>	A standard format for medical images that includes both image data and metadata.
<b>Docker</b>	A platform for packaging and running applications in isolated containers, ensuring consistent environments across systems.
<b>Edge Device</b>	Compact computing devices (such as Raspberry Pi or Jetson) designed to run AI models locally with low power consumption.
<b>Fine-Tuning</b>	The process of taking a pre-trained AI model and further training it on a specific dataset to adapt it for a particular task or domain.
<b>FP4 / FP8 / FP16 / FP32</b>	Floating-point precision formats indicating the number of bits used to represent numerical values. FP32 (32-bit) offers the highest precision, FP16 (16-bit) uses less memory and is faster, while FP8 (8-bit) and FP4 (4-bit) are supported by newer GPUs such as the H100 and B200 for even higher throughput. Lower precision reduces accuracy but can significantly improve performance for workloads that tolerate it.
<b>GPU (Graphics Processing Unit)</b>	A specialized processor originally designed for graphics but highly effective for parallel computations in AI.
<b>GPU Utilisation</b>	The percentage of a GPU's total capacity that is actively used during operation.
<b>Hailo</b>	A company producing AI accelerator chips designed for edge devices, requiring proprietary tooling and model formats (HEF).
<b>HBM (High Bandwidth Memory)</b>	High-speed memory used in advanced GPUs for faster data access.
<b>HEF (Hailo Executable Format)</b>	A proprietary model format required for Hailo AI accelerators.

Term	Description
<b>I/O (Input/Output)</b>	Data transfer between different system components, often a bottleneck in AI workflows.
<b>Idle Power Consumption</b>	The power usage of a system that is powered on but not actively performing tasks.
<b>ImageNet</b>	A large-scale image dataset containing millions of labelled images across thousands of categories, widely used for training and benchmarking image classification models.
<b>Imagenette</b>	A simplified subset of the ImageNet dataset containing 10 classes, used for faster testing and benchmarking.
<b>Jetson</b>	NVIDIA's platform of embedded AI computing devices designed for edge deployment, including the AGX Orin used in this study.
<b>Latency</b>	The time delay between data input and the corresponding output. Critical for real-time applications.
<b>LLM (Large Language Model)</b>	Large-scale language models, such as GPT, used for text generation and natural language processing.
<b>nvidia-smi</b>	NVIDIA's command-line utility for monitoring and managing NVIDIA GPUs, including setting power caps and querying utilisation.
<b>ONNX (Open Neural Network Exchange)</b>	An open standard format for AI models that enables portability across frameworks and hardware platforms.
<b>ONNX Runtime (ORT)</b>	Software used to efficiently execute ONNX models across different hardware platforms
<b>On-Prem / On-Premises</b>	IT infrastructure that is operated locally within an organisation rather than in cloud environments.
<b>oneAPI</b>	Intel's unified programming model for accelerated computing across CPUs, GPUs, and other processors.
<b>Optimisation</b>	The process of improving performance and efficiency by fine-tuning software and hardware configurations.
<b>Portability</b>	The ease with which software can be moved and executed on different hardware platforms without extensive modification.
<b>Power Capping</b>	A technique that limits the maximum power consumption of a GPU to improve energy efficiency. Exploits the non-linear relationship between power and performance – reducing power often causes a smaller proportional decrease in throughput.
<b>Proprietary Software</b>	Software owned by a company with restricted access and usage rights, unlike open-source software.
<b>Raspberry Pi</b>	A low-cost, compact single-board computer commonly used for edge computing and IoT applications.
<b>ResNet50</b>	A well-known and proven image classification model with 50 layers, often used as a benchmark for AI performance.
<b>ROCm</b>	AMD's open-source platform for GPU computing, comparable to NVIDIA's CUDA ecosystem.
<b>SDK (Software Development Kit)</b>	A software development package containing tools and libraries for building applications on a specific platform
<b>Stream</b>	Parallel execution paths on a GPU that enable simultaneous processing of multiple tasks.
<b>TCO (Total Cost of Ownership)</b>	The complete cost of a system over its lifecycle, including acquisition, energy, cooling, maintenance, and operational overhead.
<b>TensorRT</b>	NVIDIA's optimisation library designed to maximize inference performance on NVIDIA GPUs.

Term	Description
<b>Throughput</b>	The amount of data (e.g., images) processed per unit of time, typically measured in images per second
<b>Top 1 / Top 5 Accuracy</b>	Metrics indicating model accuracy. Top 1: the correct prediction is the first choice. Top 5: the correct answer appears within the top five choices.
<b>VRAM (Video RAM)</b>	Dedicated memory on a GPU used to store models and data during processing.
<b>VM (Virtual Machine)</b>	A software-emulated computer running on physical hardware, allowing flexible resource allocation.
<b>Workstation GPU</b>	Professional GPUs (such as the RTX A-series or L4) designed for workloads that fall between consumer-grade graphics and data centre operations.
<b>x86</b>	The most common processor architecture for desktops and servers, offering broad software compatibility.

## Appendix C – Cost Efficiency Calculation

The energy cost calculations data in this white paper are based on Swedish electricity prices for 2024. According to historical data from Elbruk.se (<https://www.elbruk.se/elpris-historik-2024>), the average electricity price across 2024 was 0.386 Swedish Kronor per kWh.

To determine the cost efficiency metric "Million images per Swedish Krona" presented in Picture 3 (page 9), we applied the following formula:

$$\text{Million images / SEK} = ((\text{images/kWh} / 0.386)) / 1,000,000$$

These calculations reflect direct energy costs only and do not include:

- Hardware acquisition costs
- Cooling and infrastructure costs
- Maintenance and operational overhead
- Network and storage costs

For a complete total cost of ownership (TCO) analysis, organisations should factor in all these elements alongside the direct energy consumption presented here.



## Project Context and Contributors

This white paper was developed as part of the Sustainable AI Infrastructure Lifecycle (SAIL) use case within the national project Data-Driven Organizations (DDO), coordinated by AI Sweden.

The purpose of the SAIL use case is to explore financially and environmentally sustainable approaches to AI infrastructure that support the entire AI lifecycle — from research and development to deployment, inference, and long-term operation.

## Project Partners

The work has been carried out in collaboration between: Aixia AB, Region Halland, and AI Sweden, with additional insights shared through the broader DDO consortium including industry, academia, and public-sector partners.

## Authors

- Olof Sandell, Aixia AB

## Use Case SAIL

- Aixia AB: Cecilia Millheim, Ellen Reinhardt, Jonas Nordin, Klas Ludvigsson, Milena Miernik, Olof Sandell, Simon Janeck
- Region Halland: Georgios Bramis, Karin Westerberg, Lina Gårdemark, Stefan Bäckström, Torbjörn Olander

Aixia AB

Hälsingegatan 10

414 63 Göteborg

[www.aixia.se](http://www.aixia.se)

Region Halland

Box 517

301 80 Halmstad

[www.regionhalland.se](http://www.regionhalland.se)

