Grinn AstraSOM-1680

CPU vs NPU Performance

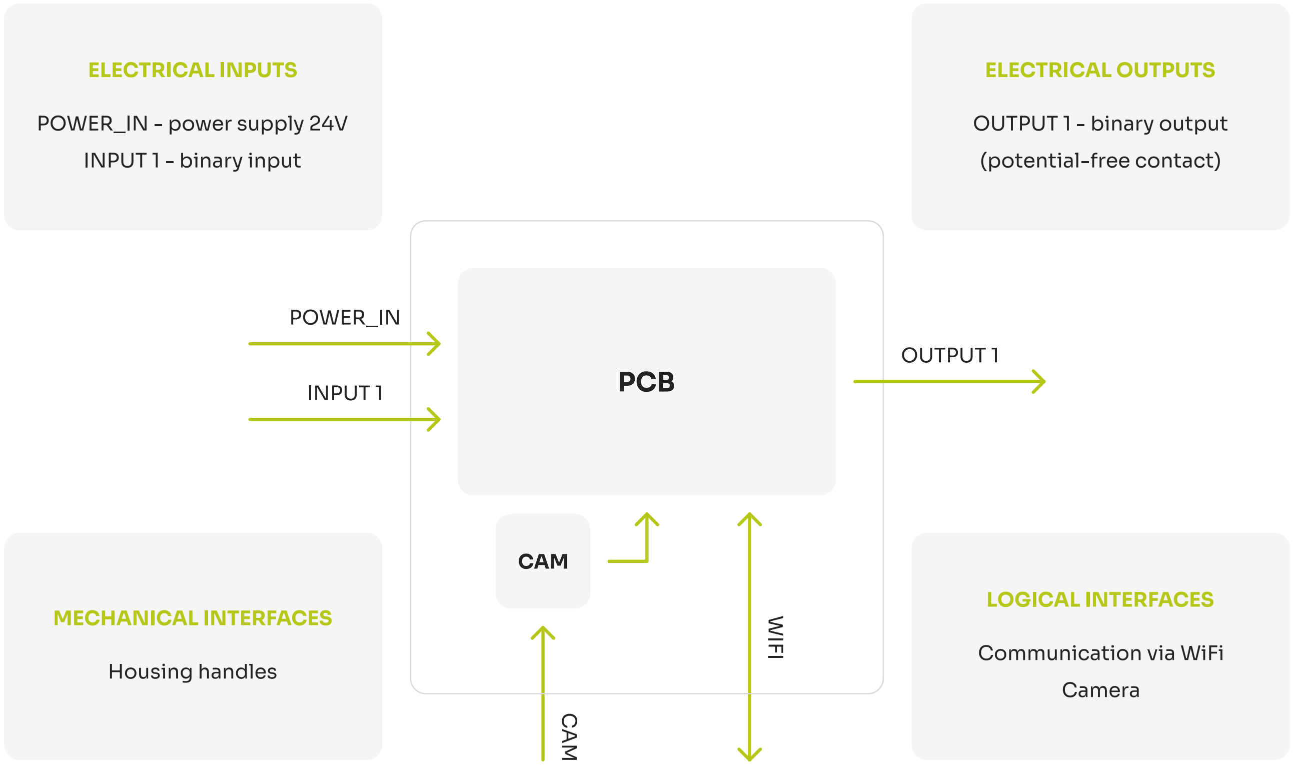
www.grinn-global.com

1 | Introduction

> This document presents a comparative analysis of inference performance on the Grinn AstraSOM-1680 System-on-Module. The evaluation was conducted using the benchmark model tool, commonly employed for assessing inference times of machine learning models.

The primary focus was to assess whether the CPU or NPU is better suited for running the inference of the model responsible for Automatic Number Plate Recognition (ANPR), a critical component of Grinn's vehicle access system. Grinn's new project aims to revolutionize vehicle access systems for high-traffic areas such as office complexes. Traditional access control systems, such as card-based gate openers, become inefficient in environments where hundreds of vehicles enter and exit daily. Grinn's solution leverages Edge AI technology to streamline the process, ensuring a faster and more user-friendly experience. By automating license plate recognition, the system minimizes the need for manual intervention, allowing for seamless vehicle entry and exit, particularly valuable during peak hours.

The implementation of this system significantly improves both security and efficiency. By eliminating the need for physical access cards, the likelihood of unauthorized access due to stolen or lost credentials is mitigated.



ELECTRICAL INPUTS

POWER_IN - power supply 24V
INPUT 1 - binary input

ELECTRICAL OUTPUTS

OUTPUT 1 - binary output (potential-free contact)

MECHANICAL INTERFACES

Housing handles

LOGICAL INTERFACES

Communication via WiFi
Camera

2 | Testing Methodology

The evaluation process involved recording execution times for both CPU and NPU processing. The model used for benchmarking was **yolov5s-int8.tflite**, executed within the **TensorFlow Lite (TFLite Runtime)** framework. Hardware acceleration was leveraged through the XNNPACK delegate for CPU execution, while theTIM-VX delegate was employed to offload computations to the NPU. To explain the details of the delegates used, here is a breakdown of their functionalities and how they impact performance on the Grinn AstraSOM-1680.

The XNNPACK delegate is designed to optimize AI inference on CPU-based systems. It enhances processing speed by utilizing SIMD (Single Instruction, Multiple Data) techniques, which allow the CPU to handle multiple calculations simultaneously. While this approach provides some acceleration over standard CPU execution, it does not take full advantage of **Synaptics SL1680's** built-in NPU (Neural Processing Unit). As a result, performance improvements are limited by the CPU's processing power, making XNNPACK a viable option only for lightweight AI models or cases where no dedicated AI hardware is available.

In contrast, the TIM-VX delegate is specifically designed to offload AI workloads to the NPU, significantly improving inference efficiency. By leveraging Astra 1680's dedicated AI acceleration hardware, TIM-VX allows for multiple times faster processing compared to CPU-based execution. Additionally, offloading AI tasks to the NPU reduces CPU load, enabling it to handle other critical functions such as data management and security protocols. This not only enhances system performance but also optimizes power efficiency, as NPUs are specifically engineered to handle AI computations with minimal energy consumption.

When comparing the two, XNNPACK is suitable for scenarios where AI models are lightweight or where NPU support is unavailable, while TIM-VX is the preferred choice for maximizing Synaptics SL1680's hardware potential, ensuring real-time AI inference and optimal resource allocation. If high performance, low latency, and efficiency are priorities, TIM-VX is clearly the better option for AI workloads on this platform.

The quantized model was generated using a representative dataset specifically created for ANPR. This dataset includes real-world license plate images collected from various environments, ensuring that the model can handle diverse lighting conditions, different plate designs, and varying angles. The quantization process improves inference efficiency while maintaining high accuracy, making it well-suited for deployment on the Grinn AstraSOM-1680.

The core hardware utilized for this evaluation is the Grinn AstraSOM-1680, a compact yet powerful systemon-module based on the Synaptics SL1680. This device is equipped with two cameras, enabling simultaneous recognition of vehicles entering and exiting the facility. Unlike traditional setups that require extensive server infrastructure, this Edge AI-powered solution processes all computations locally, reducing latency and eliminating the need for costly data centers.

Let’s see, based on the benchmark, which approach proves to be better and most optimal for real-time ANPR within the constraints of Grinn’s Edge AI deployment.

3 | Benchmark Results

This section presents the benchmarking results for AI model execution on the Grinn AstraSOM-1680, comparing CPU-based inference using the XNNPACK delegate and NPU-accelerated inference using the TIM-VX delegate. The tests were performed using the TensorFlow Lite Benchmark Tool, and the execution times were measured to evaluate inference performance, startup behavior, and initialization overhead.

3.1 | CPU Execution (XNNPACK delegate)

Command Used:

```
benchmark_model --graph = yolov5s - int8 . tflite
```

Output:

```
INFO : Inference timings in us: Init : 39018 , First inference : 84116 , Warmup ( avg):
82888.1 , Inference ( avg): 82692.3
```

Analysis of CPU performance:

- Initialization Time (39.02 ms) The initialization time includes model loading, memory allocation, and optimization steps before the first inference. The CPU initialization is relatively efficient, completing within 39 ms.

- First Inference Time (84.12 ms) This represents the time taken for the first run of the model after initialization. It is only slightly higher than the average inference time (82.69 ms), suggesting that the impact of initial computations, such as caching or memory optimization, is minimal. The CPU achieves stable inference times almost immediately, meaning that additional optimizations in subsequent iterations have little effect on performance.

- Average Inference Time (82.69 ms per frame) The inference speed stabilizes at 82.69 ms per image during sustained execution. This translates to 12 FPS (frames per second), which might not be optimal for real-time ANPR applications.

The CPU with XNNPACK delivers solid performance; however, in real-time ANPR applications, its capabilities may be limited as inference times exceed 80 ms per frame. This could impact the maximum achievable frame rate and introduce slight delays in high-traffic environments.

3.2 | NPU Execution (TIM-VX delegate)

Command Used:

```
benchmark_model --graph = yolov5s - int8 . tflite -- external_delegate_path =/ usr/lib/
libvx_delegate .so
```

Output:

```
INFO : Inference timings in us: Init : 15294 , First inference : 2393143 , Warmup ( avg):
2.39314 e+06 , Inference (avg ): 11126.7
```

Analysis of NPU performance:

- Initialization Time (15.29 ms). The NPU initializes significantly faster than the CPU, requiring only 15.29 ms. This faster initialization reduces startup latency, making it ideal for low-power, always-on AI systems.
- First Inference Time (2.39 s). The first inference is significantly slower at 2.39 seconds. This delay is expected due to initial model loading, graph compilation, and internal memory allocation by the NPU. However, this overhead does not affect subsequent inferences.
- Average Inference Time (11.13 ms per frame). Once initialized, the NPU delivers an average inference speed of just 11.13 ms per image. This translates to an effective frame rate of 90 FPS, which is more than sufficient for real-time ANPR processing.

4 | CPU vs. NPU Performance Comparison

Metric	CPU (XNNPACK)	NPU (TIM-VX)	Improvement Factor
Initialization Time	39.02 ms	15.29 ms	-
First Inference Time	84.12 ms	2393.14 ms	-
Average Inference Time	82.69 ms	11.13 ms	8 x faster
Frame Rate	12 FPS	90 FPS	7,5 x higher

5 | NPU vs. CPU Power consumption

5.1 | Testing Methodology

The following methodology was applied:

- **Baseline Power Measurement:** Power consumption in an idle state was recorded.
- **Inference Power Measurement:** A fully quantized model was used to assess the NPU's power consumption. The evaluation was conducted using the following models:
  - **YOLOv8n**
  - **YOLOv5s**
  - **MobileNetV2 1.0 224**

Model	NPU (TIM-VX)
YOLOv8n	2.00W
YOLOv5s	2.00W
MobileNetV2 1.0 224	2.39W

- **Execution Details:** The model was executed using TensorFlow Lite (LiteRT) runtime.
- **Platform-Specific Delegates:** Platform-specific delegate library was utilized to offload computations to the NPU.

The power consumption measurements confirm that the Grinn AstraSOM-1680 effectively executes fully quantized models entirely on the NPU, with no fallback to CPU processings. The observed inference power consumption remained within a low range, demonstrating the energy-efficient nature of the NPU:

- YOLOv8n and YOLOv5s operated at approximately 2.0W
- MobileNetV2 1.0 224 operated at approximately 2.39W

The idle power consumption of 1.5W indicates that the platform maintains a low baseline power requirement, making it suitable for continuous operation in power-constrained environments. The results confirm that the Grinn AstraSOM-1680 delivers fully NPU-accelerated inference with low power consumption (around 2.0–2.4W), making it a viable solution for energy-efficient edge AI deployments requiring real-time processing. For comparison, running the same inference on the CPU (without NPU) results in a significantly higher power consumption of 5.1W.

6 | Key Findings and Recommendations

- NPU provides a massive speedup. The TIM-VX delegate reduces inference time by nearly 8× compared to XNNPACK on CPU, making it the optimal choice for real-time ANPR execution. A frame rate of 90 FPS ensures smooth, high-speed processing of license plates, even in high-traffic environments.

- CPU is not ideal for real-time inference. With an average inference time of 82.69 ms per frame, the CPU can only sustain 12 FPS, which is not sufficient for seamless ANPR performance. High inference times introduce latency, making it challenging to process multiple vehicles simultaneously.

- NPU startup time is longer, but execution is much faster. The first inference on NPU takes significantly longer (2.39 s) due to model compilation and hardware optimization. However, once initialized, inference time stabilizes at 11 ms, delivering a huge boost in efficiency.

- Power efficiency and resource allocation. Using the NPU reduces CPU load, allowing the system to allocate computational resources to other critical tasks such as data storage, security protocols, and cloud integration. Additionally, NPU-based inference is generally more power-efficient, making it ideal for Edge AI deployments.

7 | Conclusion

Based on the benchmark results, the NPU (TIM-VX delegate) significantly outperforms the CPU (XNNPACK delegate) in terms of inference speed and efficiency. While CPU-based inference can process only 12 FPS, the NPU can achieve 90 FPS, making it far more suitable for real-time ANPR applications.

This advantage, however, comes with the trade-off of longer initialization time compared to CPU-based processing. While the NPU significantly accelerates inference once fully operational, the initial setup involves additional steps such as graph compilation and resource allocation. Despite this, the overall performance gain in real-time ANPR applications outweighs these initial delays, making the NPU the optimal choice for handling high-traffic environments efficiently.

Strzegomska 140A
54-429 Wrocław
Poland

+48 71 716 40 99
office@grinn-global.com
support@grinn-global.com