Intel® AI Edge Systems Verified Reference Blueprint – Efficiency Optimized Edge AI on Intel® Core Ultra Processors for Computer Vision and GEN AI

Reference Architecture

Revision 1.0
October 2025

Authors
Abhijit Sinha
Yuan Kuok Nee
Timothy Miskell

Key Contributors
Jessie Ritchey
Edel Curley

Document Number: 868537

Contents

Figures

Tables

Revision History

Document Number	Revision Number	Description	Revision Date
868537	1.0	Initial release	October 2025

§

1 Introduction

Intel® AI Edge Systems are a range of optimized commercial AI systems delivered and sold through OEM/ODM in the Intel® ecosystem. They are commercial platforms, verified-configured, tuned, and benchmarked using Intel® 's reference AI software application on Intel® hardware to deliver optimal performance for Edge workloads.

Intel® AI Edge Systems offer a balance between computing and AI acceleration to deliver optimal TCO, scalability, and security. Intel® AI Edge systems enable our partners to jumpstart development through a hardened system foundation verified by Intel® and to increase the trust in their system performance. AI Edge systems enable the ability to add AI functionality through continuous integration into business applications for better business outcomes and streamlined implementation efforts.

To support the development of these AI Edge systems, Intel® is offering reference design and verified reference blueprints with AI Edge system configurations that are tuned and benchmarked for different AI Edge System types that support Edge use cases. Verified reference blueprints (VRB) include Hardware BOM, Foundational Software configuration (OS, Firmware, Drivers) tested and verified with supported Software stack (software framework, libraries, orchestration management).

This document describes a verified reference blueprint using architecture for the Intel® Core Ultra processor family.

Intel® AI Edge Systems Verified Reference Blueprint – **Efficiency Optimized Edge AI on Intel® Core Ultra processors** for Computer Vision and GEN AI is based on a single-node architecture, that provides an environment to execute multiple AI workloads that are common to be deployed at the edge, such as the Intel® Automated Self-Checkout Reference Package and "Gen AI".

All Intel® AI Edge Systems Verified Reference Blueprints feature a workload-optimized stack tuned to take full advantage of an Intel® Architecture (IA) foundation. To meet the requirements, OEM/ODM systems must meet a performance threshold that represents a premium customer experience.

There are two configurations for Intel® AI Edge Systems Verified Reference Blueprint – **Efficiency Optimized Edge AI on Intel® Core Ultra processors** for Computer Vision and GEN AI detailed in this blueprint:

- Intel® Core Ultra Processor (Series 1) – Core® Ultra 5 125H with 12 cores and with storage and integrated platform acceleration products from Intel® for maximum containerized workload density.
- Intel® Core Ultra Processor (Series 2) – Core® Ultra 7 255H, with 16 cores and with storage and add-in platform acceleration products from Intel® targeting optimized value and performance-based solutions.

Bill of Materials (BOM) requirements detail for the configurations are provided in Chapter 2 of this document.

Intel® AI Edge Systems Verified Reference Blueprint is defined in collaboration with end users to demonstrate the solution's value for AI Inference use cases. The solution leverages

hardened hardware, firmware, and software to allow customers to integrate on top of this known-good foundation.

Intel® AI Edge Systems Verified Reference Blueprint provides numerous benefits to ensure end users have excellent performance for their AI Inference applications. Some of the key benefits of the Reference Blueprint on the Intel® Core Ultra Processor Family and Intel® ARC integrated GPU and integrated NPU include:

- High core count and per-core performance
- Compact, power-efficient system-on-chip platform
- Streamlined path to cloud-native operations
- Accelerated AI inference with integrated processor capabilities
- The X^e kernel of Intel® GPU and iGPU integrates Extended Vector Engine (XVE) and Extended Matrix Engine (XMX), which accelerate AI workflow and provides powerful and real-time computing power support for AI inference at the edge.
- The NPU is an integrated Neural Processing Unit that contains Neural Compute Engines, which consist of hardware acceleration blocks for AI operations like Matrix Multiplication and Convolution, alongside Streaming Hybrid Architecture Vector Engines for general computing tasks
- Platform-level security enhancements

§

2 Design Compliance Requirements

This chapter focuses on the design requirements for Intel® AI Edge Systems Verified Reference Blueprint – **Efficiency Optimized Edge AI on Intel® Core Ultra processors** for Computer Vision, and GEN AI.

2.1 Platform Requirements

The checklists in this chapter are a guide for assessing the platform's conformance to Intel® AI Edge Systems Verified Reference Blueprint – **Efficiency Optimized Edge AI on Intel® Core Ultra processors** for Computer Vision, and GEN AI. The hardware details for the two configurations used in the blueprint are detailed below.

Table 1. Platform Configuration Core Ultra 5 Processor Series 1

Ingredient	Requirement	Required/ Recommended	Quantity
Processor	Intel® Core Ultra 5 Processor 125H 12C/16T 18M Cache (or higher)	Required	1
Memory	64G GB DDR5 5600 MT/s	Required	2
Network	Intel® Ethernet Network Adapter i226-V/LM/IT (2.5 Gbps)	Required	1
Storage (Boot/Capacity Drive)	1 TB or equivalent	Required	1
iGPU	Intel® Arc™ graphics (part of SoC)	Required	1
NPU	Intel® AI Boost (part of SoC)	Required	1
IP cameras	4K video streaming with support for at least 15 FPS and RTSP	Optional	4

Table 2. Platform Configuration Core Ultra 7 Processor Series 2

Ingredient	Requirement	Required/ Recommended	Quantity
Processor	Intel® Core Ultra 7 Processor 255H 16C/16T	Required	1
Memory	32GB DDR5 5600 MT/s	Required	2
Network	Intel® Ethernet Network Adapter i226-V/LM/IT (2.5 Gbps)	Required	1
Storage (Boot/Capacity Drive)	1 TB or equivalent	Required	1
iGPU	Intel® Arc integrated Graphics (part of SoC)	Required	1

Ingredient	Requirement	Required/ Recommended	Quantity
NPU	Intel® AI Boost (part of SoC)	Required	1
IP cameras	4K video streaming with support for at least 15 FPS and RTSP	Optional	8

2.2 BIOS Settings

To meet the performance requirements for an Intel® AI Edge Systems Verified Reference Blueprint – **Efficiency Optimized Edge AI on Intel® Core Ultra processors** for Computer Vision, and GEN AI, Intel® recommends using the BIOS settings for enabling processor P-state and C-state with Intel® Turbo Boost Technology ("turbo mode") enabled. Hyperthreading is recommended to provide higher thread density.

Refer to the following table for the set of recommended BIOS settings.

Table 3. Recommended BIOS Settings

Setting	Value
Hyper-Threading	Enabled
Turbo Mode	Enabled
VT-d	Enabled
Re-Size BAR Support	Enabled

Note: Hyperthreading is Disabled on Core Ultra Series 2 processors

BIOS settings differ from vendor to vendor. Please contact your Intel® Representative if you do not see the exact setting in your BIOS.

2.3 Solution Architecture

Figure 1 shows the architecture diagram of Intel® AI Edge Systems Verified Reference Blueprint – **Efficiency Optimized Edge AI on Intel® Core Ultra processors** for Computer Vision, and GEN AI. The Efficiency Optimized Edge AI on Intel® Core Ultra blueprint is one of a series of blueprints that have been developed on Intel® Xeon, Intel® Core, Intel® Core Ultra and Intel® ARC™ products. The blueprints align with the following categories:

- Efficiency-optimized Edge AI enhances AI performance with Intel® Core Ultra CPUs, iGPUs, and NPUs, ideal for low power while maximizing potential performance.
- Scalable performance Edge AI delivers Server Grade AI performance on Intel® Xeon and Intel® Core with built-in AI accelerators and the option to add a discrete Intel® GPU.
- Mainstream and entry Edge AI offer a balance between computing, inferencing, total cost of ownership, and low power on Intel® Core and Intel® Atom CPUs

The software stack used in this blueprint consists of two categories of AI software:

1. Vision AI

2. Gen AI

All applications are containerized using docker.

For the Vision AI use case, we are using the Intel® Automated Self-Checkout application, which measures stream density. The video data is ingested and pre-processed before each inferencing step. The inference is performed using two models: YOLOv11 and EfficientNet. The YOLOv11 model detects objects, and the EfficientNet model classifies Objects.

The Gen AI benchmark leverages the OpenVINO™ Gen AI LLM Benchmarking framework and is deployed in a containerized manner.

Figure 1. Architecture of the Intel® AI Edge Systems Verified Reference Blueprint

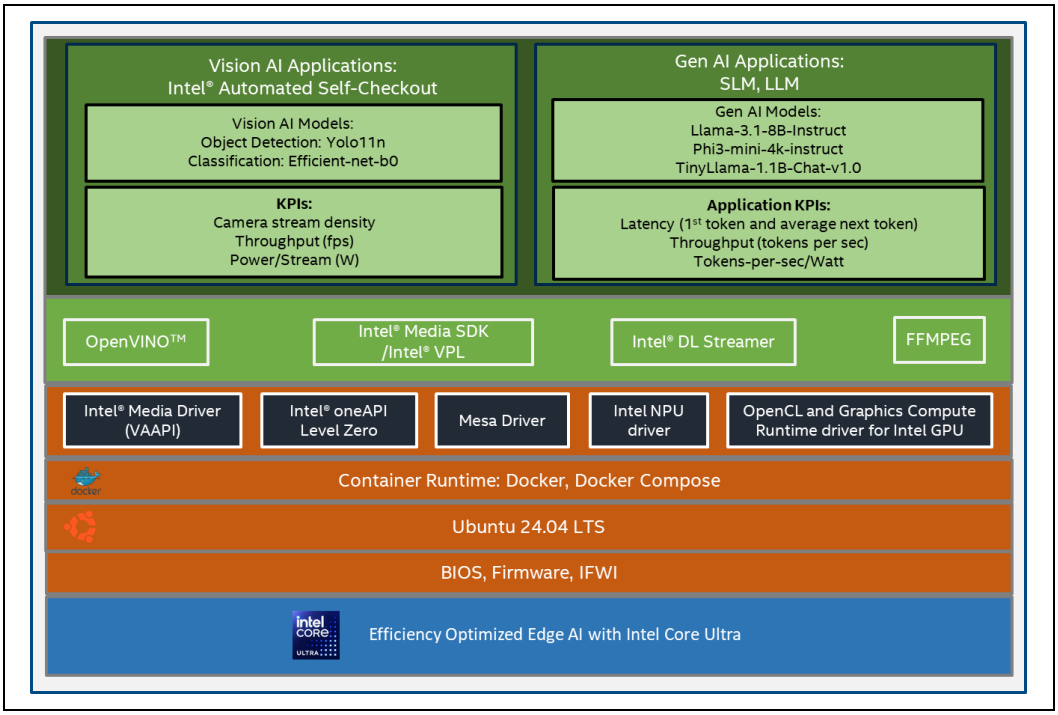


Figure 2 shows the architecture diagram for the Intel® Automated Self-Checkout application, which in this case is deployed containerized via Docker. Vision AI use case measures stream density in terms of the number of supported cameras at the target FPS, accounting for all stages within the processing pipeline. The video data is ingested and pre-processed before each inference stage. The inference is performed using two models: YOLOv11 and EfficientNet. The YOLOv11 model performs object detection while the EfficientNet model performs object classification. For additional information refer to the Appendix.

Figure 2. Test Methodology for Retail Self-checkout Pipeline

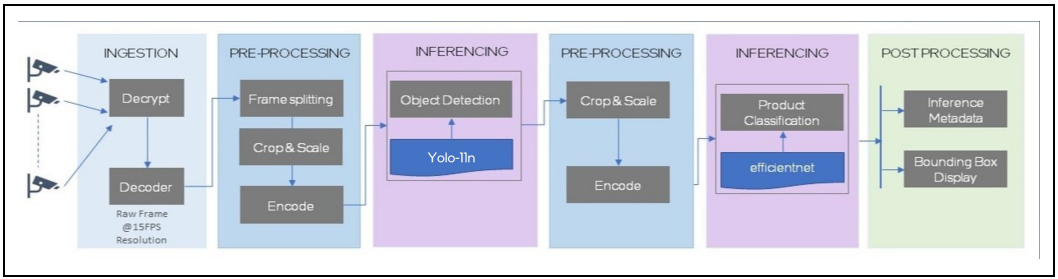


Table 4 is a guide for assessing the conformance to the software requirements for the Intel® AI Edge Systems Verified Reference Blueprint.

Table 4. SW Configuration

Ingredient	SW Version Details
OS	Ubuntu* 24.04.2 LTS
Kernel	6.14.0-24-generic
OpenVINO™ Gen AI	2025.2.0
Docker Engine	28.3.2
Docker Compose	2.36.2
OpenCL Compute Runtime Version	25.27.34303.5
Intel® oneAPI Level Zero	1.22.4
Intel® NPU Driver	1.19.0
Media Driver VAAPI	1.22.0
Intel® oneVPL	2.15
Mesa	25.0.7
OpenCV	4.6.0
DLStreamer	2025.0.1.3
FFmpeg	6.1.1
Python	3.12

2.4 Platform Technology Requirements

This section lists the requirements for Intel® 's advanced platform technologies.

The Reference Blueprint recommends that Intel® Virtualization Technology (VT) to be enabled to reap the benefits of hardware virtualization. Either Intel® Boot Guard or Intel® Trusted Execution Technology establishes firmware verification, allowing for platform static root of trust.

Table 5. Platform Technology Requirements

Platform Technologies		Enable/Disable	Required/Recommended
Intel® VT	Intel® CPU Virtual Machine Extension (VMX) Support	Enable	Required
	Intel® I/O Virtualization	Enable	Required
Intel® Boot Guard	Intel® Boot Guard	Enable	Required
Intel® TXT	Intel® Trusted Execution Technology	Enable	Recommended

2.5 Platform Security

For Intel® AI System for the Edge, it is recommended that Intel® Boot Guard Technology be enabled so that the platform firmware is verified suitable during the boot phase.

In addition to protecting against known attacks, all Intel® Accelerated Solutions recommend installing the Trusted Platform Module (TPM). The TPM module enables administrators to secure platforms for a trusted (measured) boot with known trustworthy (measured) firmware and OS. This allows local and remote verification by third parties to advertise known safe conditions for these platforms through the implementation of Intel® Trusted Execution Technology (Intel® TXT).

2.6 Side Channel Mitigation

Intel® recommends checking your system's exposure to the "Spectre" and "Meltdown" exploits. This reference implementation has been verified with Spectre and Meltdown exposure using the latest Spectre and Meltdown Mitigation Detection Tool, which confirms the effectiveness of firmware and operating system updates against known attacks.

The spectre-meltdown-checker tool is available for download at https://github.com/speed47/spectre-meltdown-checker.

§

3 Platform Tuning

3.1 Boot Parameter Setup

For the workload testing, note that it is not necessary to enable hugepage support, nor is it necessary to enable isolcpu support. To enable GPU optimizations, edit the "/etc/default/grub" file and update the "GRUB_CMDLINE_LINUX" to include the following parameters:

" i915.enable_guc=3 i915.max_vfs=7 i915.force_probe=* udmabuf.list_limit=8192 iommu=pt vt.handoff=7 "

After modifying the grub file, run "update-grub" and "reboot" to apply the changes and verify the change with "cat /proc/cmdline":

```
cat /proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-6.14.0-27-generic root=UUID=ccddff00-ea0a-
441f-86ea-c903c958f4ab ro quiet splash i915.enable_guc=3 i915.max_vfs=7
i915.force_probe=* udmabuf.list_limit=8192 iommu=pt vt.handoff=7
```

3.2 CPU Scaling Governor to Performance

To maximize performance, the CPU scaling profile can be set to performance mode through the OS.

$ echo performance | sudo tee /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor

3.3 Install Intel® ARC™ GPU Drivers

Refer to the following for instructions on installing the Intel® Client GPU/iGPU driver:

https://dgpu-docs.intel.com/driver/client/overview.html

https://github.com/intel/compute-runtime/releases/tag/25.27.34303.5

Refer to Table 4 for a list of the installed software versions.

3.4 Install the Intel® NPU Driver

Refer to the following for instructions on installing the Intel® NPU Driver:
https://github.com/intel/linux-npu-driver/releases/tag/v1.22.0

§

4 Performance Verification

This chapter aims to verify the performance metrics for the Intel® AI Edge Systems Verified Reference Blueprint to ensure that no anomalies are seen. Refer to the information in this chapter to ensure that the platform's performance baseline is as expected.

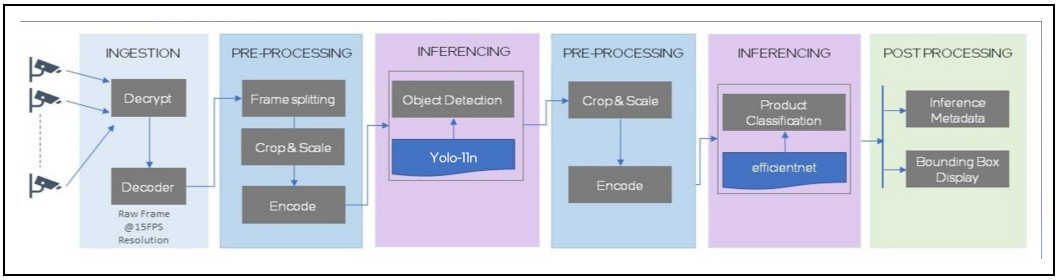
The solutions were tested on August 20, 2025, with the following hardware and software configurations listed in Table 1, Table 2, and Table 4.

4.1 Vision AI Proxy Workload

The Automated Self-Checkout Reference Implementation provides critical components to build and deploy a self-checkout use case using Intel® hardware, software, and other open-source software such as OpenVINO™. For instance, in this case, all the models in the pipeline are converted into OpenVINO™ format. In addition, this proxy workload makes use of both GStreamer for media processing and DLStreamer for inferencing, which includes detection and classification. This reference implementation provides a pre-configured automated self-checkout pipeline optimized for Intel® hardware. For more details, see Appendix.

The video stream is cropped and resized to enable the inference engine to run the associated models. The object detection and product classification features identify the SKUs during checkout. This proxy workload supports either running directly on the CPU or fully offloading to the GPU or NPU, including encoding/decoding and inferencing.

Figure 3. Vision AI Video Analytics Pipeline



The Yolo11n model is used for Object detection, and the Efficient-b0 model for Object classification, which are optimized using OpenVINO™ at INT8 precision. Individual system results may vary as power and performance are affected by use, configuration, and other factors. Details are at intel.com/performanceindex.

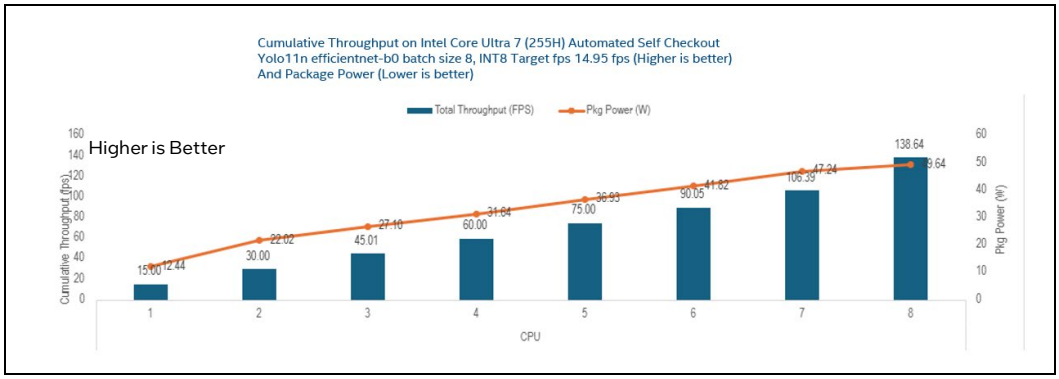
For this Verified Reference Blueprint, we have measured the average package Power dissipated in Watts. The stream density measurement is recorded as the number of camera streams supported at a target fps of 14.95.

The section below summarizes the Vision AI results, with detailed graphs across CPU, iGPU, and NPU documented.

4.1.1 Vision AI Performance on Intel® Core Ultra 7 255H

The results for the Vision AI Workload running on the Core Ultra 7 255H CPU are shown below.

Figure 4. Vision AI Performance Intel® Core Ultra (Series 2) – Core Ultra 7 255H on CPU

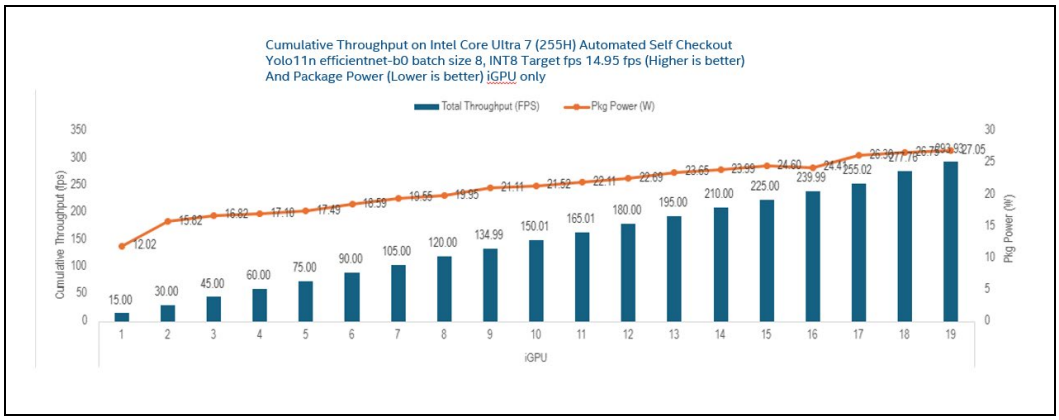


The graph above shows that the Intel® Core Ultra Series 2 – Core Ultra 7 -255H can support up to 8 IP camera streams at 14.95 FPS per stream, for an aggregate of up to 138.64 FPS (CPU only). The graph also shows the power dissipated as the stream density is increased.

The average package Power consumed by the Intel® Core 255H is 49.64W while running 8 IP camera streams with INT8 precision and Batch Size 8 at a target of 14.95 FPS on the CPU exclusively for Vision AI.

The results for the Vision AI workload running on the iGPU only are shown below.

Figure 5. Vision AI Performance on Intel® Core Ultra Processors (Series 2) Core Ultra 7 – 255H on iGPU

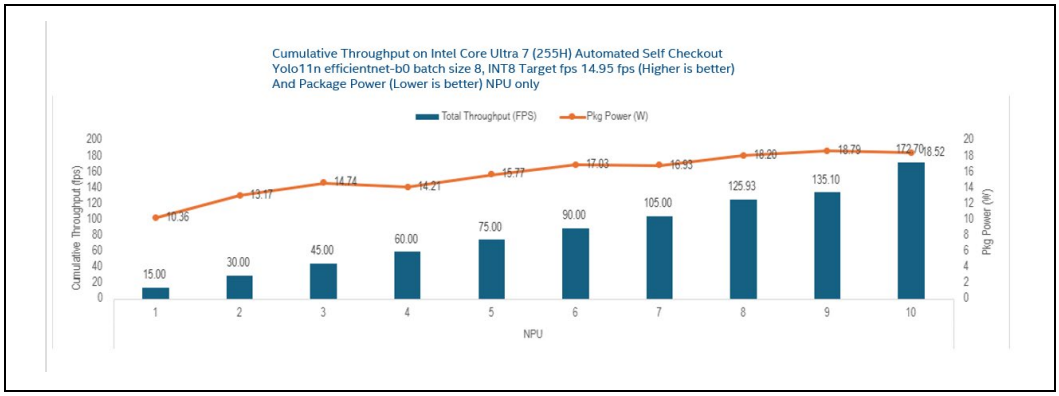


The graph above shows that Intel® Core Ultra Processors (Series 2)—Core Ultra 7 - 255H with iGPU can support up to 19 IP camera streams at 14.95 FPS per stream. The graph also shows the power dissipated as the stream density increases.

The average package Power consumed by the Intel® Core 255H is 27.05W while running 19 IP camera streams with INT8 precision and Batch Size 8 at a target of 14.95 FPS on iGPU exclusively for Vision AI.

The results for the Vision AI workload running on the NPU only are shown below.

Figure 6. Vision AI Performance Intel® Core Ultra (Series 2) – Core Ultra 7 255H on NPU



The graph above shows that the Intel® Core Ultra Processors (Series 2 – Core Ultra 7 - 255H with NPU can support up to 10 IP camera streams at 14.95 FPS per stream, for an aggregate of up to 172.7 FPS. The graph also shows the power dissipated as the stream density is increased.

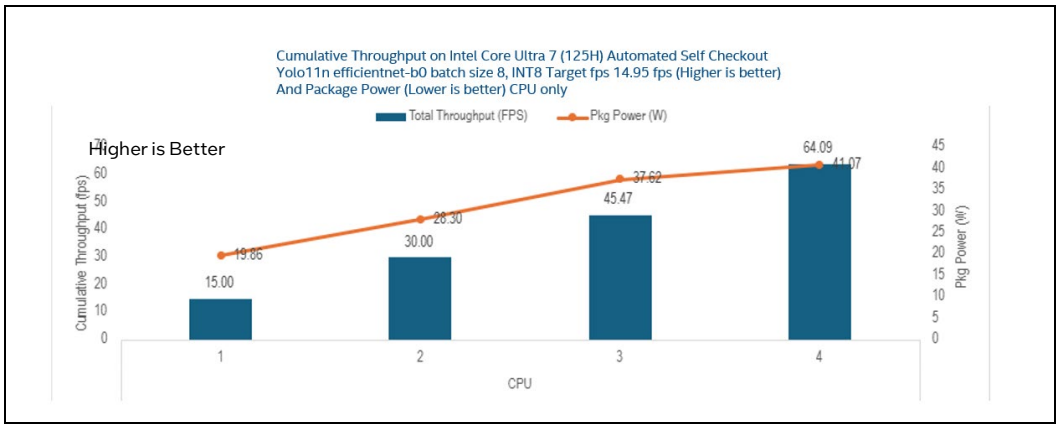
The average package Power consumed by the Intel® Core 255H is 18.52 W while running 10 IP camera streams with INT8 precision and Batch Size 1 at a target of 14.95 FPS on the CPU exclusively for Vision AI.

Note: Intel® recommends using a batch size of 1 for Vision AI NPU tests

4.1.2 Vision AI Performance on Intel® Core Ultra 5 - 125H

Below are the results for the Vision AI Workload running on the Core Ultra 5 - 125H CPU only.

Figure 7. Vision AI Performance on Intel® Core Ultra (Series 2) – Core Ultra 5 125 on CPU

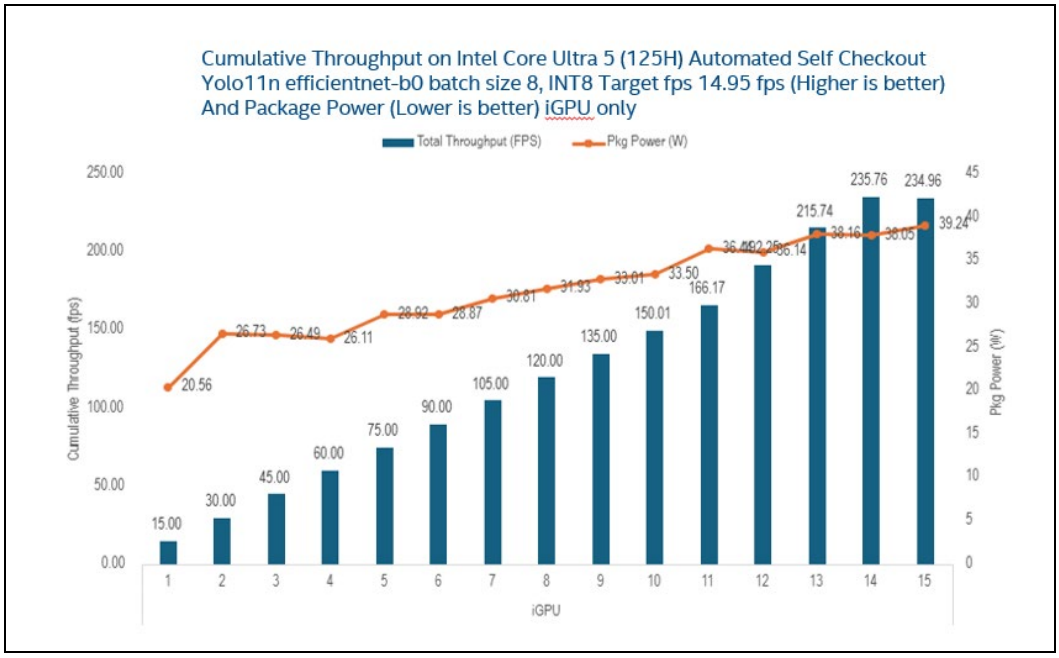


The graph above shows that the Intel® Core Ultra 5 Series 2 -125H should be able to service up to 4 IP camera streams at 14.95 FPS per stream, for an aggregate of up to 64.09 FPS (CPU only). The graph also shows the power dissipated as the stream density is increased.

The average package Power consumed by the Intel® Core 125H is 41.07W while running 4 IP camera streams with INT8 precision and Batch Size 8 at a target of 14.95 FPS on the CPU exclusively.

The results for the Vision AI workload running on the iGPU only are shown below.

Figure 8 . Vision AI Performance on Intel® Core Ultra 5 – 125H on iGPU

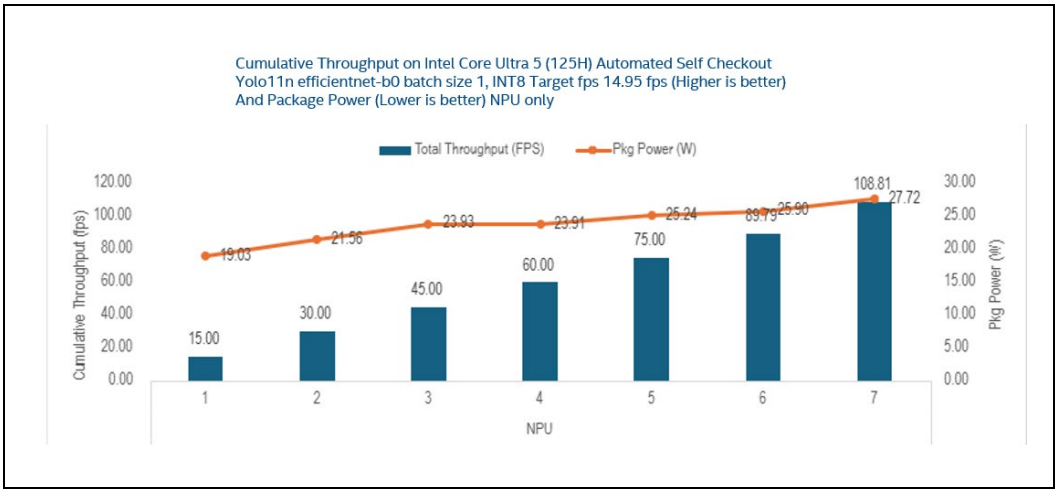


The graph above shows that the Intel® Core Ultra Processors (Series 2) – Core Ultra 5 - 125H with iGPU should be able to service up to 15 IP camera streams at 14.95 FPS per stream, for an aggregate of up to 234.96 FPS. The graph also shows the power dissipated as the stream density is increased.

The average package Power consumed by the Intel® Core 125H is 39.24W while running 15 IP camera streams with INT8 precision and Batch Size 8 at a target of 14.95 FPS on iGPU exclusively.

The results for the Vision AI workload running on NPU only are shown below.

Figure 9. Vision AI Performance Intel® Core Ultra 5 - 125H on NPU



The graph above shows that Intel® Core Ultra Processors (Series 2) – Core Ultra 5- 125H with NPU can support up to 7 IP camera streams at 14.95 FPS per stream, for an aggregate of up to 108.81 FPS. The graph also shows the power dissipated as the stream density is increased.

The average package Power consumed by the Intel® Core 125H is 27.72 W while running 7 IP camera streams with INT8 precision and Batch Size 1 at a target of 14.95 FPS on the NPU exclusively.

Note: Intel® recommends using a batch size of 1 for Vision AI NPU tests

4.2 Gen AI Proxy Workload

The large language model (LLM) proxy workload highlights the Gen AI processing capabilities of the Intel® AI Edge Systems Verified Reference Blueprint—Efficiency **Optimized Edge AI on Intel® Core Ultra processors** for Computer Vision and GEN.

Intel® AI Edge Systems Verified Reference Blueprint—**Efficiency Optimized Edge AI** on Intel® Core Ultra processors ensures that the system's results follow the expected results, as shown below, to baseline the platform's performance. The results shown include performance values for throughput (tokens/s) and time taken to generate the first token. Also, the average power measured during AI inferencing is recorded.

The GEN AI testing leveraged the models in the table below..

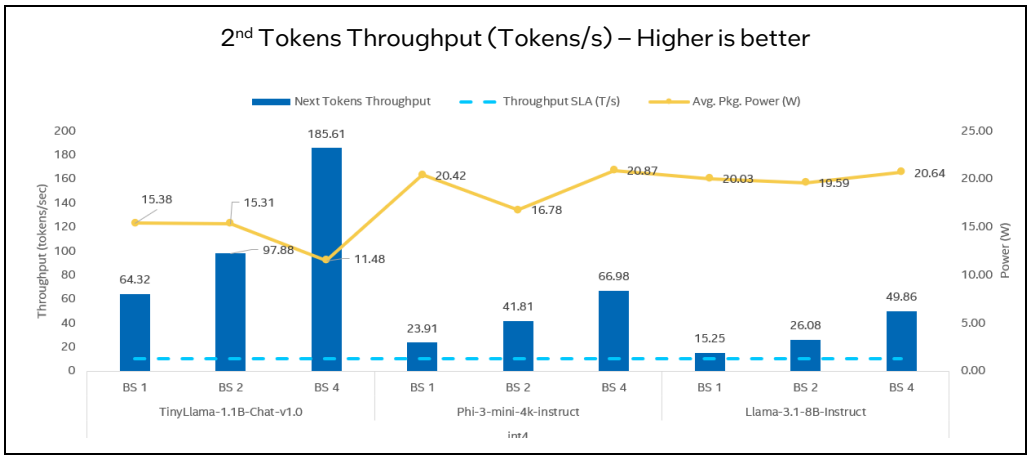
Table 6. Gen AI Models

Model Name	Parameters
Llama-3.1-8B-Instruct	8B
Phi-3-Mini-4k-instruct	3.8B
TinyLlama-1.1V-Chat-v1.0	1.1B

4.2.1 GEN AI Performance on Intel® Core Ultra 7 – 255H iGPU

The results for the GEN AI Workload (Throughput) on Intel® Core Ultra 7 – 255H iGPU are shown below.

Figure 10. GEN AI Throughput (Tokens/s) Intel® Core Ultra 7 – 255 iGPU Performance



This graph shows the 2nd Tokens (Tokens/s) throughput and the Power measured for the GEN AI models. Important data points to note in the above graph are:

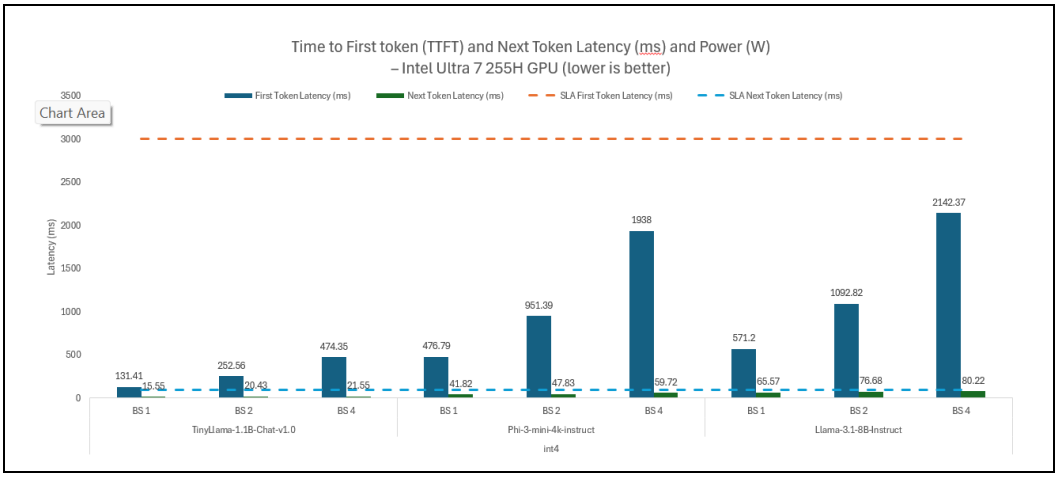
The Intel® Core Ultra 7 255H can run AI Inferencing for Llama-3.1-8B-Instruct up to 15.25 tokens/s on an iGPU with a Batch size of 1 at INT4 precision and a Power Efficiency of 0.76 Tokens/s/W.

The Intel® Core Ultra 7 255H can run AI Inferencing for Phi-3-mini-4k-Instruct up to 23.91 tokens/s on an iGPU with a Batch size of 1 at INT4 precision and a Power Efficiency of 1.17 Tokens/s/W.

The Intel® Core Ultra 7 255H can run AI Inferencing for TinyLlama 1.1B-Chat-v1.0 up to 64.32 tokens/s on an iGPU with a Batch size of 1 at INT4 precision and a Power Efficiency of 4.18 Tokens/s/W.

The results for the GEN AI TTFT running Intel® Core Ultra on iGPU are shown next.

Figure 11. GENAI Time to First Token Latency on Intel® Core Ultra 7 – 255 iGPU Performance



The graph above shows the time to first (TTFT) for the 3 GEN AI LLM models tested.

The industry's acceptable values for First Token Latency and Second Token Latency are 3s and 10ms, respectively.

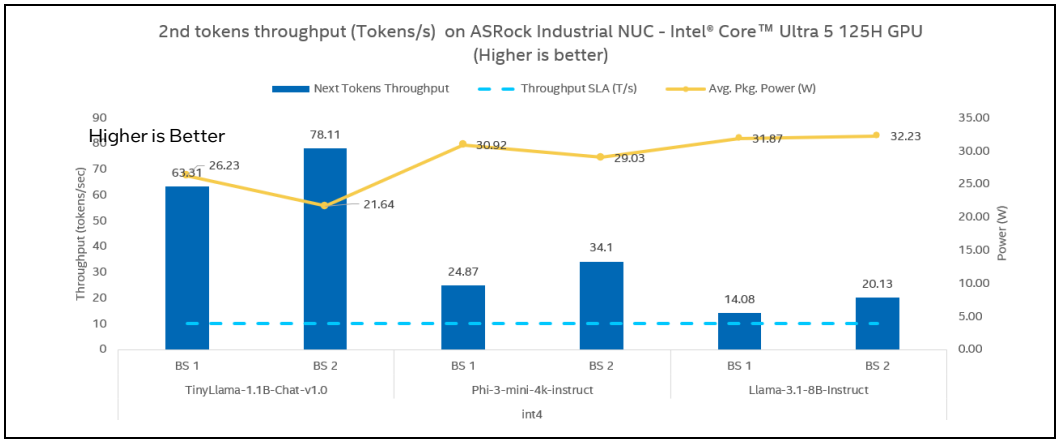
Note: Important data points to note in the above graph are:

- The TTFT (Time to First Token Latency) of the Intel® Core Ultra 7 255H for AI Inferencing on iGPU using Llama-3.1-8B-Instruct is 571.2 ms for a Batch size of 1 at INT4 precision.[1]
- The TTFT (Time to First Token Latency) of the Intel® Core Ultra 7 255H for AI Inferencing on iGPU using Phi-3-mini-4k-Instruct is 476.79 ms for a Batch size of 1 at INT4 precision
- The TTFT (Time to First Token Latency) of the Intel® Core Ultra 7 255H for AI Inferencing on iGPU using TinyLlama 1.1B-Chat-v1.0 is 131.41 ms for a Batch size of 1 at INT4 precision.

4.2.2 GEN AI Performance on Intel® Core Ultra 5 – 125H iGPU

The results for the GEN AI Workload (Throughput) on Intel® Core Ultra 5 – 125H iGPU are shown below.

Figure 12. GEN AI Throughput (Tokens/s) Intel® Core Ultra 5 125 iGPU Performance



The graph above shows the tokens/s for the 3 GEN AI models tested and the Power dissipated.

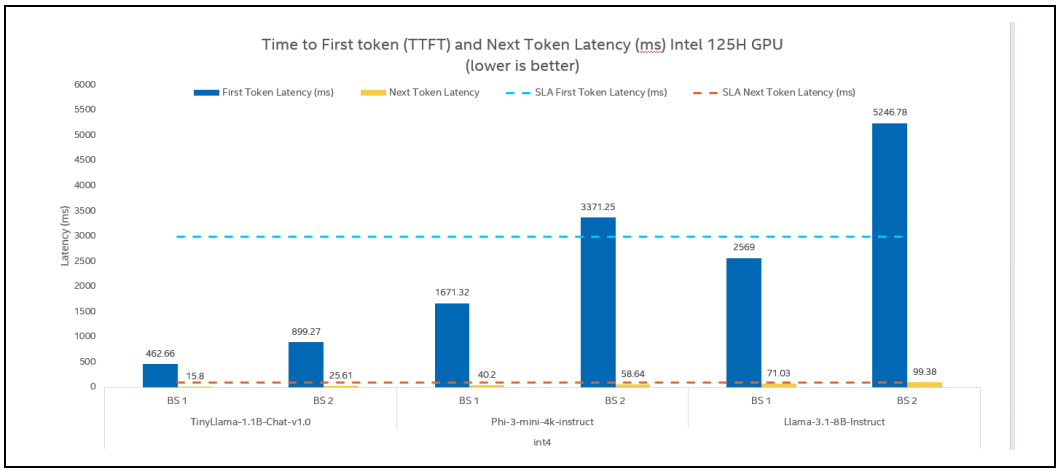
The Intel® Core Ultra 5 125H can run AI Inferencing for Llama-3.1-8B-Instruct up to 14.08 tokens/s on iGPU with a Batch size of 1 at INT4 precision with a Power Efficiency of 0.44 Tokens/s/W.

The Intel® Core Ultra 5 125H can run AI Inferencing for Phi-3-mini-4k-Instruct up to 24.87 tokens/s on iGPU with a Batch size of 1 at INT4 precision with a Power Efficiency of 0.8 Tokens/s/W.

The Intel® Core Ultra 5 125H can run AI Inferencing for TinyLlama 1.1B-Chat-v1.0 up to 63.31 tokens/s on iGPU with a Batch size of 1 at INT4 precision with a Power Efficiency of 2.41 Tokens/s/W.

The results for the GEN AI TTFT running Intel® Core Ultra 5 125H on iGPU only are shown below.

Figure 13. GEN AI Time to First Token Latency on Intel® Core Ultra 5 – 125H



The graph above shows the time to first token for the 3 GEN LLM AI models tested.

The acceptable values in the industry for First Token Latency and Second Token Latency are 3s and 10ms, respectively.

The TTFT (Time to First Token Latency) of the Intel® Core Ultra 5 125H for AI Inferencing on iGPU using Llama-3.1-8B-Instruct is 2569 ms.

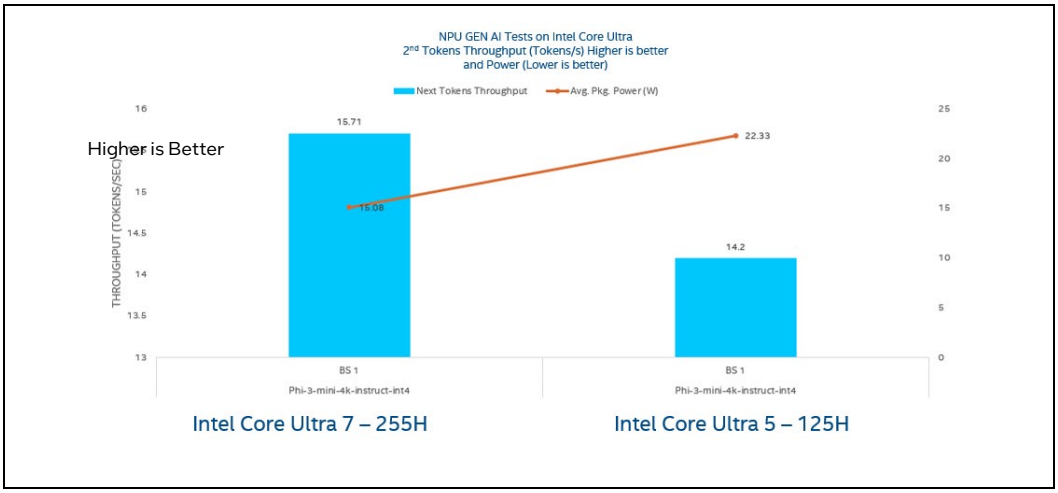
The TTFT (Time to First Token Latency) of the Intel® Core Ultra 5 125H for AI Inferencing on iGPU using Phi-3-mini-4k-Instruct is 1671.32 ms

The TTFT (Time to First Token Latency) of the Intel® Core Ultra 5 125H for AI Inferencing on iGPU using TinyLlama 1.1B-Chat-v1.0 is 462.66 ms.

4.2.3 GEN AI Performance on NPU on Intel® Core Ultra 7 – 255H and Intel® Core Ultra 5 – 125H

The results for the GEN AI models running on NPU for both configurations are shown below.

Figure 14. GEN AI Throughput (Tokens/s) on Intel® Core Ultra 7 255 and Core Ultra 5 125H NPU Performance



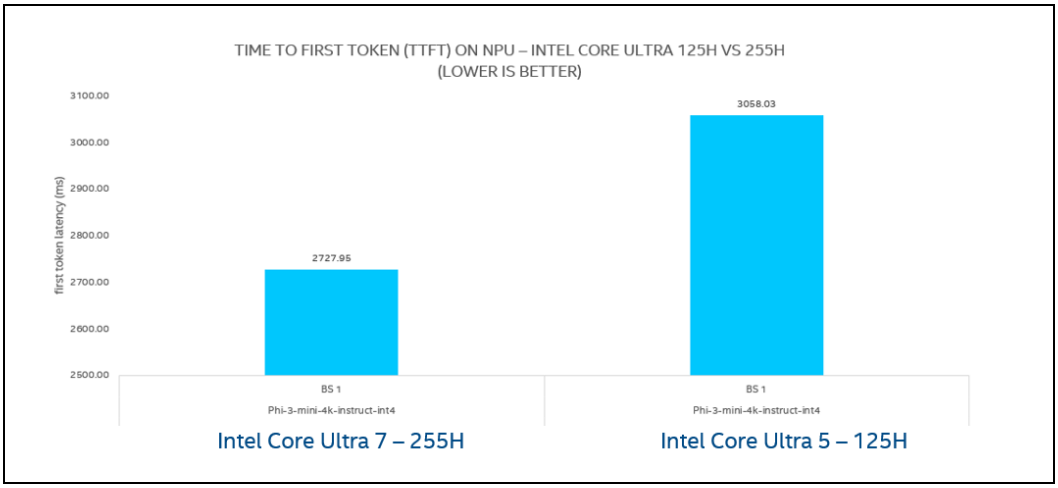
The graph above shows that the tokens/s for the Phi-3-mini-4k-instruct model and the Average Power dissipated.

The Intel® Core Ultra 7 255H can run AI Inferencing for Phi-3-mini-4k-instruct up to 15.7 tokens/s on NPU with a Batch size of 1 at INT4 precision with a Power Efficiency of 1.04 Tokens/s/W.

The Intel® Core Ultra 5 125H can run AI Inferencing for Phi-3-mini-4k-instruct up to 14.2 tokens/s on NPU with a Batch size of 1 at INT4 precision with a Power Efficiency of 0.94 Tokens/s/W.

The results for the GEN AI TTFT on Intel® Core Ultra on NPU are shown below.

Figure 15. GENAI Time to First Token on Intel® Core Ultra 7 255 and Core Ultra 5 125H NPU Performance



The graph above shows the Time to First Token Latency (TTFT) for the Phi-3-mini-4k-instruct model.

The TTFT of the Intel® Core Ultra 5 125H for AI Inferencing on NPU with Batch size 1 at INT4 precision using Phi-3-mini-4k-instruct is 3058 ms.

The TTFT of the Intel® Core Ultra 7 255H for AI Inferencing on NPU with Batch size 1 at INT4 precision using Phi-3-mini-4k-instruct is 2727.95 ms.

§

5 Summary

The Intel® AI Edge Systems Verified Reference Blueprint – **Efficiency Optimized Edge AI on Intel® Core Ultra processors** for Computer Vision and GEN AI defined on Intel® Core Ultra processors addresses the capabilities for AI inference by offering the following value proposition, detailed within the tables below. This was tested in August 2025.

Table 7. Vision AI Performance on Intel® Core Ultra Processors

SKU	#Camera Streams			Package Power (W)			Power per stream (Watt/Stream)		
AI Inference Engine	CPU	iGPU	NPU	CPU	iGPU	NPU	CPU	iGPU	NPU
Core Ultra 5 125H	4	15	7	41.07	39.24	27.72	10.27	2.62	3.96
Core Ultra 7 255H	8	19	10	49.64	27.05	18.52	6.21	1.42	1.85

A summary of results for Vision AI is shown above for a Batch size of 8 for CPU and iGPU, and a Batch Size of 1 for NPU at INT 8 precision. The Power per stream is derived from the Package Power measurement and could prove useful as a comparison point.

A summary of the results for GEN AI with a Batch size of 1 is shown in the table below. For additional information, please consult the previous section.

Table 8. GEN AI Performance on Intel® Core Ultra Processors (Batch size 1 and INT4 precision)

SKU		Tokens/s		TTFTL (ms)		Package Power(W)	
AI Engine	Model	iGPU	NPU	iGPU	NPU	iGPU	NPU
Core Ultra 5 125H	Llama-3.1-8B-Instruct	14.08	N/A	2569	N/A	31.87	N/A
	Phi-3-mini-4k-instruct	24.87	14.2	1671.32	3058.03	30.92	22.33
	TinyLlama-1.1B-Chat-v1.0	63.31	N/A	462.66	N/A	26.23	N/A
Core Ultra 7 255H	Llama-3.1-8B-Instruct	15.25	N/A	571.2	N/A	20.03	N/A
	Phi-3-mini-4k-instruct	23.91	15.71	476.79	2727.95	20.42	15.08
	TinyLlama-1.1B-Chat-v1.0	64.32	N/A	131.41	N/A	15.38	N/A

The Intel® Core Ultra 7 255H GPU features an XMX (Xe Matrix Extension) engine that delivers 4.5x better performance for Time to First Token Latency compared to the Intel® Core Ultra 5 125H for AI inferencing using Llama-3.1-8B-Instruct on the iGPU.

We also observe that the Intel® Core Ultra 7 255H iGPU is up to 1.46 times more Efficient than the Intel® Core Ultra 5 125 iGPU for AI Inference using Phi-3-mini-4k-Instruct, as it uses less power to generate similar Throughput (Tokens per second).

This blueprint, combined with architectural improvements, feature enhancements, and integrated Accelerators, provides a significant performance and scalability advantage in support of today's AI workload.

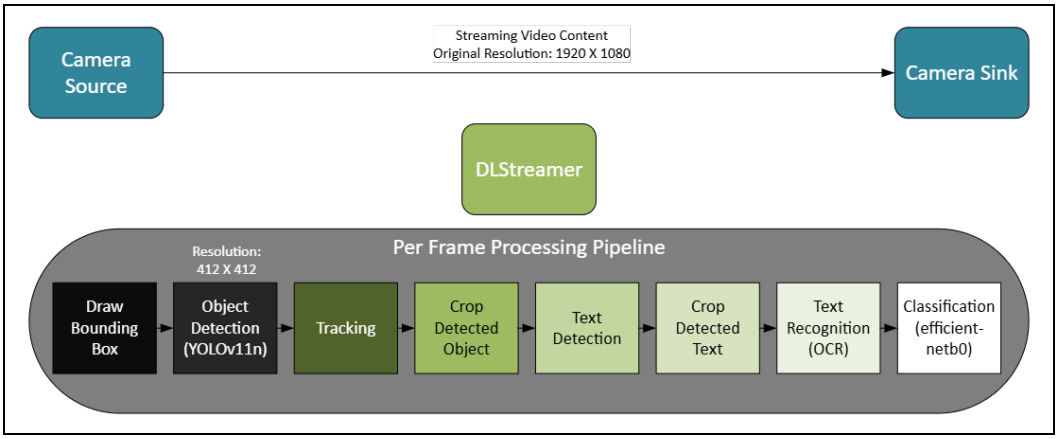
§

Appendix A Appendix

The following section provides detailed instructions for benchmarking a platform with each of the proxy workloads for Vision AI, Gen AI, along Network Security AI. The benchmarking process leverages the tools and scripts provided as part of the Intel® AI Edge Systems Verified Reference Blueprint will be available later, please reach out to your Intel® Field Representative for access.

A.1 Automated Self-Checkout Test Methodology

Figure 16. Test Methodology for the Automated Self-Checkout Proxy Workload



The Intel® Automated Self-Checkout Reference Package provides critical components required to build and deploy a self-checkout use case using Intel® hardware, software, and other open-source software. It is a part of Intel® 's Edge AI Suites – a collection of building blocks, industry-specific libraries and sample applications designed to help develop optimized AI solutions - https://github.com/open-edge-platform/edge-ai-suites.

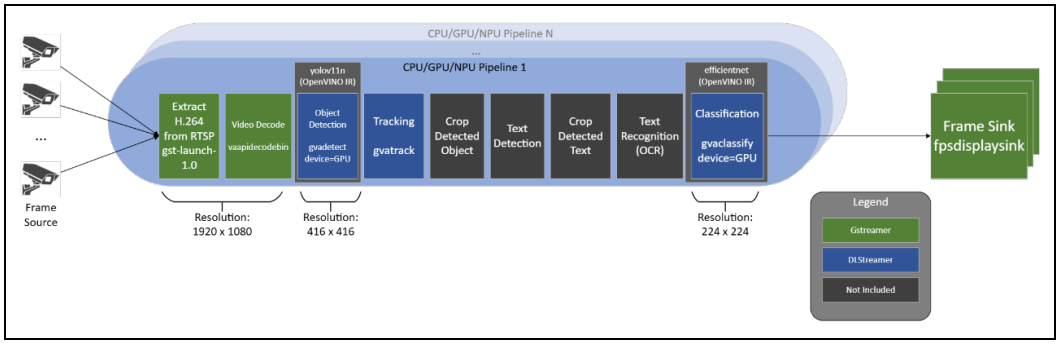
Vision workloads are large and complex and need to go through many stages. For instance, in the pipeline shown within the figure below, the video data is ingested, pre-processed before each inferencing stage, inferenced using two models – YOLOv11 and EfficientNet, and post-processed to generate metadata along with drawing the bounding boxes for each frame. The camera source plays back pre-recorded video content, which is then processed by the media analytics pipeline. The video stream NPUt is decoded within the CPU pipeline using software-based decodebin API calls, while for the GPU pipeline the decoding is offloaded using vaapidecodebin API calls. The video content is freely available from https://www.pexels.com .

The Intel® Automated Self-Checkout Reference makes use of Intel® Deep Learning Streamer (Intel® DL Streamer), which leverages the open-source media framework GStreamer to provide optimized media operations along with the Deep Learning Inference Engine from the OpenVINO™ Toolkit to provide optimized inference. DLStreamer accelerates the media analytics pipeline for the Vision AI use case and allows for offloading to the underlying Intel® ARC™ and Intel® Data Center Flex GPUs.

The media analytics pipeline for Vision AI utilizes DLStreamer to performs object classification on the Region(s) of Interest (ROI) detected by gvadetect using the gvaclassify element and Intermediate Representation (IR) formatted object classification model. The models used for detection are in OpenVINO™ Intermediate Representation format, which is optimized for Intel® CPUs and GPUs. One advantage for the OpenVINO™ IR format is that the models can be used as-is without the need for retraining to leverage Intel® CPUs and GPUs. The Vision AI pipeline also uses object tracking for reducing the frequency of object detection and classification, thereby increasing the throughput, using gvatrack. The pipeline publishes the detection and classification results within a JSON file, which is then parsed, and the final results are reported in a log file.

Note: The GStreamer multi-media framework is used to stream video content by the frame source and the frame sink endpoints. The current release does not make use of the underlying media engines, offloading to the media engines is planned for future releases of the Intel® Automated Self-Checkout Reference.

Figure 17. Detailed Test Methodology for Retail Self-Checkout Pipeline



The test measures the number of streams that the system can sustain at the target FPS. For each test iteration, the number of camera streams is monotonically increased until the currently measured FPS value falls below the target FPS value. The number of streams is then monotonically decremented until the target FPS is met.

- Upon test completion the results are captured for the average FPS, the cumulative FPS, along with the peak number of streams achieved at the target FPS.

To run the automated self-checkout test please go to:

https://github.com/intel-retail/automated-self-checkout/blob/v3.5.1/README.md

A.2 Gen AI Test Methodology using OpenVINO™ with Gen AI

The Gen AI benchmark leverages the OpenVINO™ Gen AI LLM Benchmarking framework and is deployed in a containerized manner.

```
The version below was used for data collection in this document.

https://github.com/OpenVINO™toolkit/OpenVINO™.genai/tree/2025.2.0.0/tools/llm_bench
```

The LLM models were quantified for NPU. For more information on using LLMs on NPU, see below.

https://docs.OpenVINO™.ai/2025/OpenVINO™-workflow-generative/inference-with-genai/inference-with-genai-on-npu.html#export-an-llm-model-via-hugging-face-optimum-intel

The list of models optimized for NPU are shown at this link.

https://huggingface.co/collections/OpenVINO™/llms-optimized-for-npu-686e7f0bf7bc184bd71f8ba0