Accelerating Multimodal LLMs on Intel Xeon 6 Processors using OpenVINO

Planning / Implementation

Large language models (LLMs) have rapidly advanced from processing text alone to understanding and generating content across multiple modalities, including images, video, and audio. These multimodal LLMs (MM-LLMs) enable a wide range of enterprise applications from document intelligence that can parse images of receipts and contracts, to customer support systems that can reason over both text and product images, to manufacturing workflows that combine defect detection with natural language reporting.

While the capabilities of MM-LLMs are transformative, their deployment poses practical challenges. Multimodal architectures typically combine a vision encoder with a large-scale transformer, resulting in substantial compute and memory requirements. Traditionally, such models have been run on GPUs, but GPU availability, cost, and energy consumption often limit adoption.

The latest Intel Xeon 6 processors, paired with Lenovo ThinkSystem servers, deliver new architectural enhancements such as Intel Advanced Matrix Extensions (AMX), high memory bandwidth, and increased core density, making them well-suited for AI inference workloads that previously demanded accelerators. When combined with OpenVINO, Intel's open-source toolkit for deep learning optimization, Xeon 6 processors provide a cost-effective and efficient path to deploy MM-LLMs.

In this paper, we demonstrate how to accelerate a state-of-the-art multimodal LLM, Qwen2.5-VL-7B-Instruct on a Lenovo ThinkSystem SR650 V4 server powered by Intel Xeon 6 processors. We compare a standard Hugging Face implementation against an OpenVINO-optimized workflow, highlighting performance gains in inference speed and efficiency. The results show that enterprises can deploy powerful multimodal AI solutions without relying on costly or scarce GPU resources.

ThinkSystem SR650 V4

The Lenovo ThinkSystem SR650 V4, powered by the latest Intel Xeon 6 processors, provides a flexible and scalable foundation for next-generation AI workloads. Designed for performance, efficiency, and reliability, the SR650 V4 is ideal for organizations looking to deploy demanding multimodal applications across data center and enterprise environments.

Figure 1. Lenovo ThinkSystem SR650 V4

At the heart of the SR650 V4, the Intel Xeon 6 delivers a leap in AI acceleration, thanks to features such as Advanced Matrix Extensions (AMX) and AVX-512, which unlock significant gains in transformer-based models. With expanded memory bandwidth, increased core density, and built-in support for AI instructions, Xeon 6 is engineered to meet the growing demands of multimodal workloads that integrate both vision and language reasoning.

When combined, Lenovo's proven server engineering and Intel's AI-optimized silicon create a platform that delivers:

- **Scalable performance** for large-scale LLM and vision workloads without the need for GPUs.
- **Lower total cost of ownership (TCO)** by leveraging existing CPU-based infrastructure.
- **Enterprise-grade reliability** for production deployments in mission-critical environments.
- **Optimized AI performance** with OpenVINO, ensuring seamless integration between hardware and software.

Together, the Lenovo SR650 V4 and Intel Xeon 6 offer enterprises the ability to deploy multimodal AI at scale, with performance, efficiency, and flexibility that align with real-world business needs.

OpenVINO Acceleration

In the following steps, we demonstrate how to run the same Qwen2.5-VL-7B-Instruct model optimized using OpenVINO.

1. Create a python virtual environment and install the needed packages.

   It is highly recommended to use a uv python virtual environment to ensure compatibility between packages, but a standard python virtual environment will work as well. Run the following on the command line.

   ```
   uv venv ov_env
   source ov_env/bin/activate
   uv pip install "torch>=2.1" "torchvision" "qwen-vl-utils" "Pillow" "gra
   dio>=4.36" --extra-index-url https://download.pytorch.org/whl/cpu
   uv pip install -q -U "openvino>=2025.0.0" "openvino-tokenizers>=2025.0.
   0" "nncf>=2.15.0"
   uv pip install -q "git+https://github.com/huggingface/optimum-intel.git
   " --extra-index-url https://download.pytorch.org/whl/cpu
   uv pip install -q "transformers>=4.49"
   ```

2. Download Multimodal LLM.

   Here we may download the Qwen2.5-VL-7B-Instruct model, convert it to OpenVINO's optimized Intermediate Representation (IR) and quantize it into an int4 representation to quicken inference speeds and reduce model footprint. Readers are encouraged to experiment with other weight formats such as bf16, and int8.

   ```
   optimum-cli export openvino --model Qwen/Qwen2.5-VL-7B-Instruct Qwen2.5
   -VL-7B-Instruct-OV-INT4 --weight-format int4
   ```

3. Import packages.

```
from optimum.intel.openvino import OVModelForVisualCausalLM
from PIL import Image
from transformers import AutoProcessor, AutoTokenizer
from qwen_vl_utils import process_vision_info
from transformers import TextStreamer
```

4. Download and define sample image.

   While any image will work, we suggest downloading and saving the image found at this url to a local directory:
   https://github.com/openvinotoolkit/openvino_notebooks/assets/29454499/d5fbbd1a-d484-415c-88cb-9986625b7b11

5. Define model and processor.

   The model should match the model we exported earlier; the purpose of the processor is to encode data from both the text and image modalities into the model.

```
model = OVModelForVisualCausalLM.from_pretrained("Qwen2.5-VL-7B-Instruct-OV-INT4")
processor = AutoProcessor.from_pretrained("Qwen2.5-VL-7B-Instruct-OV-INT4")
```

6. Define prompt and encode.

```
messages = [
    {
        "role": "user",
        "content": [
            {
                "type": "image",
                "image": f"file://{example_image_path}",
            },
            {"type": "text", "text": question},
        ],
    }
]
text = processor.apply_chat_template(messages, tokenize=False, add_generation_prompt=True)
image_inputs, video_inputs = process_vision_info(messages)
inputs = processor(
    text=[text],
    images=image_inputs,
    videos=video_inputs,
    padding=True,
    return_tensors="pt",
)
```

7. Generate output.

   Finally a call to model.generate is all that is needed to create an output.

```
generated_ids = model.generate(**inputs, max_new_tokens=128)
```

Results

We conducted a simple speed comparison of Qwen-2.5-VL-7B running on both Huggingface and OpenVINO using various weight compressions. We give each the same image and prompt “Please describe the image.” while limiting the models to 128 generation tokens. We test five variations of Qwen-2.5-VL-7B: fp32 with huggingface and fp32, fp16, int8, and int4 with OpenVINO.

Our results are shown in the following figure. Shown is the average of 30 runs with a single input image, the prompt "Please describe the image.", and a maximum of 128 output tokens.

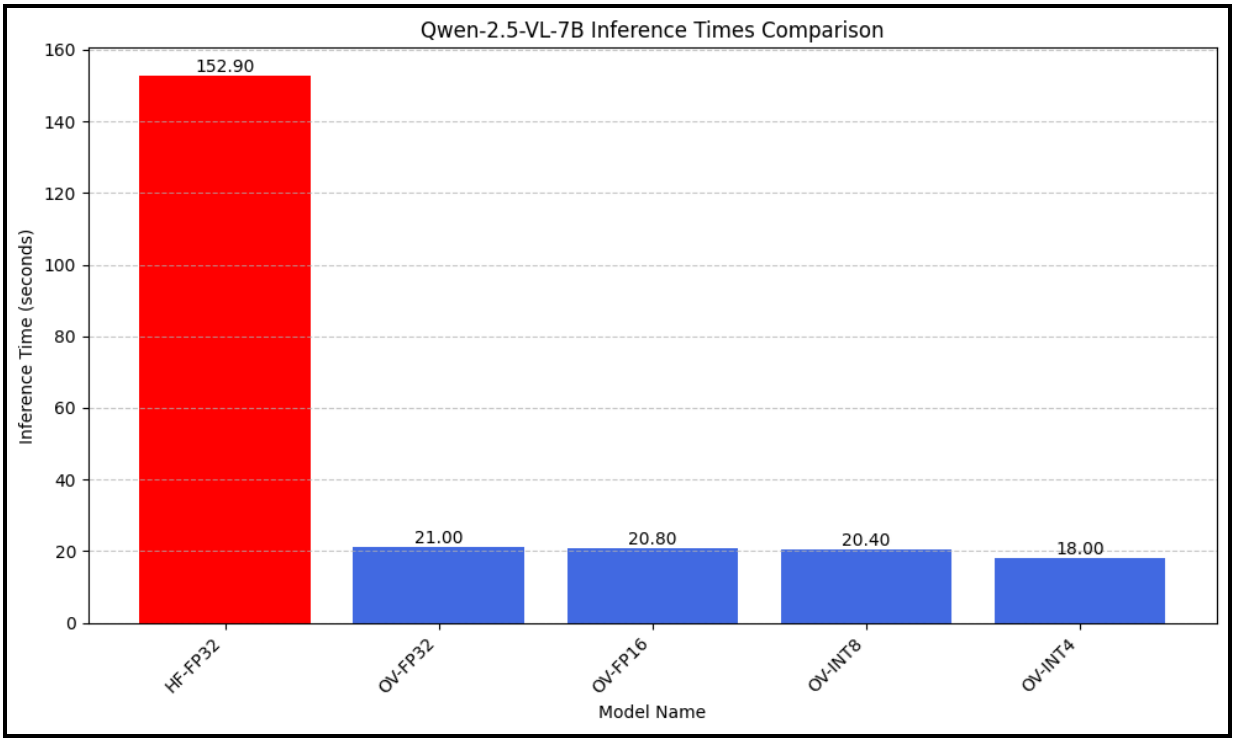

Figure 2. Comparison of Huggingface model versus OpenVINO at various quantizations

Simply converting the model to OpenVINO while keeping the same fp32 precision results in a **7.28x** speed-up. Further weight compressing the model from fp32 to int4 results in a **1.17x** relative speed-up. Combined this results in a staggering **8.49x speed-up over the standard Huggingface implementation.**

Conclusion

This paper has demonstrated a powerful and cost-effective alternative for deploying multimodal large language models (MM-LLMs) without relying on traditional, expensive GPU-centric infrastructure. By leveraging the AI-optimized capabilities of Intel Xeon 6 processors and the deep-learning optimizations of the OpenVINO toolkit, we achieved a 8.49x performance speed-up for the Qwen2.5-VL-7B-Instruct model compared to its standard Hugging Face implementation.

Our results validate the synergy between Intel's purpose-built hardware and its simple yet powerful software ecosystem. The new Intel Advanced Matrix Extensions (AMX) on Xeon 6 processors, when fully utilized by OpenVINO, unlocks a new level of AI performance on standard CPUs. This is particularly significant for enterprises that have already invested in CPU-based infrastructure for their data centers. The combination of Lenovo ThinkSystem servers and Intel Xeon 6 processors presents a compelling path to harness the transformative potential of MM-LLMs for applications like document intelligence, quality control, and customer support, while significantly reducing total cost of ownership (TCO) and energy consumption.

In an era where enterprises are seeking to democratize AI and deploy it at scale, the path forward is not always through specialized, costly accelerators. This work proves that powerful, efficient, and scalable multimodal AI inference is not only possible but practical on a CPU-based platform. The integration of OpenVINO and Intel's next-generation silicon offers a robust and economical solution, empowering organizations to deploy advanced AI models with unparalleled efficiency.

Hardware Details

The following table lists the key components of the server we used in our performance tests.

Table 1. Hardware Details

Component	Description
Server	Lenovo ThinkSystem SR650 V4
Processor	Intel Xeon 6740P 48C 270W 2.1GHz
Installed Memory	16x Samsung 64GB TruDDR5 6400MHz (2Rx4) 10x4 16Gbit RDIMM
Disk	4x ThinkSystem 2.5" U.2 PM9D3a 1.92TB Read Intensive NVMe PCIe 5.0 x4 HS SSD
OS	Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-60-generic x86_64)
OpenVINO	2025.2.0

Author

Eric Page is an AI Engineer at Lenovo. He has 6 years of practical experience developing Machine Learning solutions for various applications ranging from weather-forecasting to pose-estimation. He enjoys solving practical problems using data and AI/ML.

Notices

Lenovo may not offer the products, services, or features discussed in this document in all countries. Consult your local Lenovo representative for information on the products and services currently available in your area. Any reference to a Lenovo product, program, or service is not intended to state or imply that only that Lenovo product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Lenovo intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any other product, program, or service. Lenovo may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Lenovo (United States), Inc.
8001 Development Drive
Morrisville, NC 27560
U.S.A.
Attention: Lenovo Director of Licensing

LENOVO PROVIDES THIS PUBLICATION ”AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Lenovo may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Lenovo product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Lenovo or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary. Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Lenovo Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Lenovo product, and use of those Web sites is at your own risk. Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

© Copyright Lenovo 2025. All rights reserved.

This document, LP2305, was created or updated on September 26, 2025.

Send us your comments in one of the following ways:

- Use the online Contact us review form found at:
  https://lenovopress.lenovo.com/LP2305
- Send your comments in an e-mail to:
  comments@lenovopress.com

This document is available online at https://lenovopress.lenovo.com/LP2305.

Trademarks

Lenovo and the Lenovo logo are trademarks or registered trademarks of Lenovo in the United States, other countries, or both. A current list of Lenovo trademarks is available on the Web at https://www.lenovo.com/us/en/legal/copytrade/.

The following terms are trademarks of Lenovo in the United States, other countries, or both:
Lenovo®
ThinkSystem®

The following terms are trademarks of other companies:

Intel®, OpenVINO®, and Xeon® are trademarks of Intel Corporation or its subsidiaries.

Linux® is the trademark of Linus Torvalds in the U.S. and other countries.

Other company, product, or service names may be trademarks or service marks of others.