Synergizing LLMs, GPUs, and CPUs for Efficient Workload Consolidation

White Paper

Intel Authors:

Rohit D'Souza, AI Product Manager, Intel
Oluwaseyi Ogebule, GPU Product Manager, Intel
Chee Bin Chong, Software Enabling and Optimization Engineer, Intel

DFI Authors:

Jarry Chang, General Manager, Product Center, DFI
Waterball Liu, Solution Product Manager, Product Center, DFI

Contributors:

Linda Chen, Sales Account Manager, Intel
James Lo, Sales Account Engineer, Intel

November 2024

Document Number: 837196-1.0

Performance varies by use, configuration and other factors. Learn more on the Performance Index site.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. No product or component can be absolutely secure.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or visiting the Intel Resource and Documentation Center.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Check with your system manufacturer or retailer or learn more at intel.com.

Intel is committed to respecting human rights and avoiding complicity in human rights abuses. See Intel's Global Human Rights Principles. Intel's products and software are intended only to be used in applications that do not cause or contribute to a violation of an internationally recognized human right.

Contents

Figures

Tables

Revision History

Date	Revision	Description
November 2024	1.0	Initial release.

1.0 Abstract

This white paper presents an innovative solution in the form of an Electric Vehicle (EV) charging station that integrates three AI pipelines: face detection, speech recognition, and chatbot interaction. The solution provides a seamless and efficient user experience by leveraging the computational capabilities of the 14th Gen Intel® Core™ Processor, Intel® UHD 770 integrated GPU, and Intel® Arc™ A380 Graphics discrete GPU.

The face detection pipeline enables quick unlocking of the interactive screen. The speech recognition pipeline facilitates intuitive communication by precisely transcribing the spoken language. The chatbot pipeline delivers context-aware responses, enabling engaging interactions. Together, these pipelines create a comprehensive AI system capable of transforming user experience across various applications in video analytics, natural language processing, and generative AI.

This paper provides the high-level architecture for the solution and performance of each of the pipelines. In doing so, it demonstrates the potential of Intel CPUs and GPUs to revolutionize AI-driven user interaction.

2.0 Introduction

Intel Corporation and DFI Inc., a global leading provider of high-performance computing technology across multiple embedded industries for more than 40 years, have a longstanding partnership. The first strategic collaboration was on the Intel 486 series motherboard back in 1992. Recently, DFI partnered with Intel to create Electric Vehicle (EV) chargers using the 14th Gen Intel® Core Processor paired with Intel® Arc™ GPU.

2.1 The Solution – EV Charging Station with Chatbot

DFI's EV charging station platform represents a significant advancement in integrating AI into everyday infrastructure. As shown in Figure 1, the EV charger platform is equipped with an AI chatbot (named Synthia), which interacts with users through touchscreens and voice commands. The system includes two front screens, a back screen, and audio-visual input capabilities, enabling sophisticated user engagement. Key technologies include OpenAI's Whisper for speech recognition, facial detection algorithms, and a conversational AI chatbot powered by an LLM (Mistral Instruct-7B). These functionalities are supported by PyTorch models and optimized using Intel's OpenVINO toolkit to run efficiently on Intel CPU and GPU. This setup leverages the power-efficient features of the Intel Arc GPU for the Edge and utilizes the Intel Core Processor along with its integrated GPU, Intel UHD Graphics 770. These three components handle the three individual workloads to enhance the overall system responsiveness and user experience.

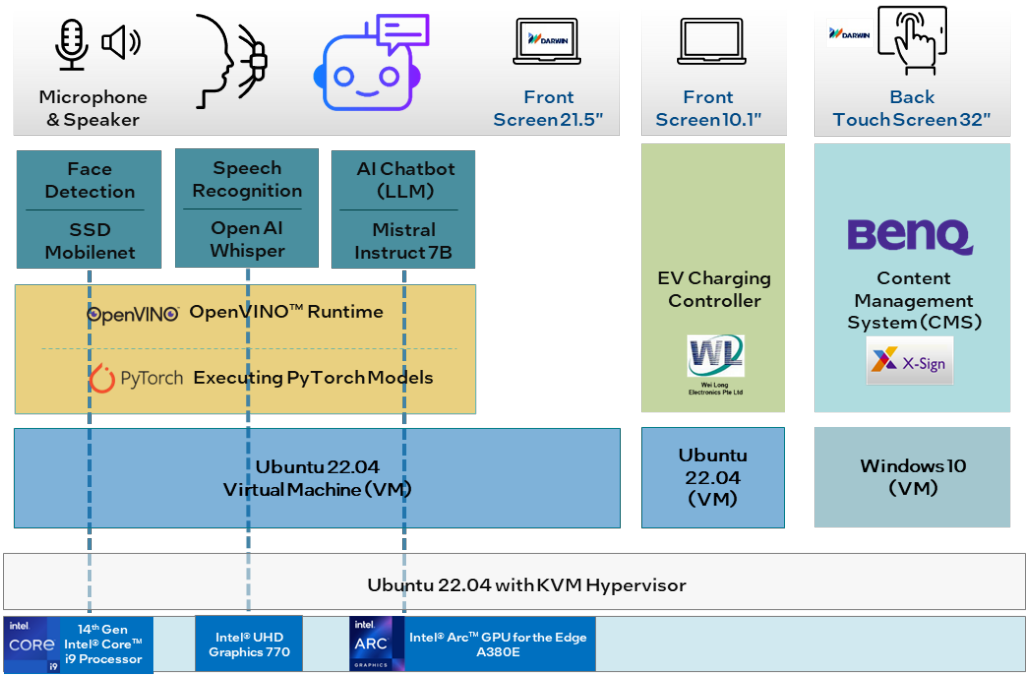

Figure 1: Electric vehicle charging station running three AI pipelines

2.2 Key Benefits of an Intel Solution

2.2.1 Workload Consolidation

Workload consolidation offers several benefits when implemented on Intel Core processors.

- Efficiency and cost savings
  - Consolidating multiple workloads onto a single system reduces capital expenditures as well as the physical footprint.
  - Lower operational costs for power and cooling due to fewer systems.
  - Reduced management and ease of implementation.
- Enhanced Security
  - Consolidation minimizes system-to-system communication, thus improving security.
  - Potential attack vectors are reduced due to fewer interfaces.

Workload consolidation on Intel Core Processors streamlines operations, reduces costs, and improves security.

2.2.2 Scalability

Intel Arc GPUs are designed to be compatible with a wide range of CPU hosts. Specifically, the Arc series pairs with several generations of Intel Core Processors, maximizing the value of existing investments. This combination allows for significant system scalability by improving the efficiency of CPU-only setups (freeing up CPU resources by offloading workloads to GPU). Alternatively, it could enhance an existing solution by managing new and more demanding workloads through the addition of the GPU.

3.0 Hardware AI Accelerators

3.1 Intel® Core™ Processors (14th Gen)

Intel® Core™ Processors (14th Gen) (codenamed Raptor Lake-S Refresh) are built for IoT Edge applications. The following features list key AI capabilities[1]:

- Supports up to 24 cores and 32 threads of hybrid architecture, providing a balance between performance and power efficiency
  - Up to 8 Performance cores (P-cores) and 16 Efficient cores (E-Cores)
  - Only P-cores are assigned to the virtual machines.
  - Intel® Thread Director organizes tasks by performance or efficiency needs
- AI Acceleration with Intel® Deep Learning Boost (Intel® DL Boost)
  - Accelerates AI inference tasks directly on the processor
- Intel® UHD Graphics 770 driven by X^{e}® architecture provides parallelization for AI workloads
  - Supports up to 32 Execution Units (EUs)
- PCIe 5.0 and PCIe 4.0
- Single root I/O virtualization (SR-IOV)

3.1.1 Intel® Deep Learning Boost (Intel® DL Boost)

The Intel® DL Boost technology offers acceleration for AI workloads. It includes Vector Neural Network Instructions (VNNI) delivering a significant performance improvement by combining three instructions into one. Thus, maximizing compute resources, utilizing cache better, and reducing bandwidth bottlenecks. VNNI is widely used in AI tasks such as image recognition, object detection, and natural language processing.

In the EV Charger solution, DFI Inc. utilizes this key AI capability to execute the face detection pipeline on CPU cores.

3.1.2 Intel® UHD Graphics 770

The Intel® UHD Graphics 770 is an integrated GPU. It is based on the Intel X^{e}-core architecture with vector and matrix engines[2]. In the Intel X^{e}-core architecture an

[1] For additional information visit NEX Intel® Core™ processors (14th Gen) (Raptor Lake-S Refresh) Gold Deck

[2] Heterogeneous AI Powerhouse: Unveiling the Hardware and Software Foundation of Intel® Core™ Ultra Processors for the Edge

execution unit (EU) is a fundamental component responsible for executing instructions. With support for up to 32 EUs, the 14th Gen Intel® Core Processors provide a significant amount of parallelization for AI inference workloads.

In this EV Charger solution, DFI Inc. takes advantage of this parallel processing capability by executing the entire automated speech recognition AI pipeline on the Intel® UHD Graphics 770.

3.1.3 Intel Graphics Single Root I/O Virtualization (SR-IOV)

SR-IOV is a key feature that DFI Inc. utilized to bring out the AI capabilities of the Electric Vehicle charger solution.

SR-IOV defines a standard method for sharing a physical device function by partitioning the device into multiple virtual functions (VF). Each virtual function is assigned to a virtual machine, achieving near-native performance for the virtual machine(VM)[3]. In other words, it allows resource sharing directly in hardware rather than in software. The key benefits of Intel Graphics SR-IOV are:

- Efficiently use GPU resources in a virtual system by sharing a physical GPU with VMs
- Improved media AI analytics, video transcode, and virtual desktop infrastructure (VDI) workloads performance on VMs
- Support for multiple guest operating systems
- Support for independent displays

The integrated GPU with SR-IOV capabilities can be configured to appear in the PCI address space as multiple VFs. By creating multiple VFs for the same physical GPU, one can assign individual VFs to different VMs[4]. In other words, VFs such as AI inferencing can be assigned to different VMs to improve the overall AI performance of the pipeline.

3.2 Intel® Arc™ GPU for the Edge

The Intel® Arc™ GPUs for the Edge are based on the highly scalable Intel Xᵉ-core architecture and designed to enable innovation for AI, visual computing, and media processing. Intel Arc GPUs pair seamlessly with select Intel® Core™ CPU Processors for a complete solution. Built on Intel's advanced Xᵉ Graphics architecture, the Intel Arc GPU delivers scalable performance across various computing environments, from integrated graphics to high-performance discrete graphics. Intel Xᵉ HPG

[3] Intel® Graphics SR-IOV Enablement Toolkit
[4] Intel iGPU (Integrated Graphics) SR-IOV - The Catalyst for IoT Virtualization in Factory Automation

architecture provides purpose-built acceleration for key edge usages and workloads, including the Intel® Xᵉ Matrix Extensions (Intel® XMX) AI Engine to speed up inference and the Xᵉ Media Engine for faster transcode and other media-processing tasks. Intel Arc GPUs target the edge specifically with five-year long-life availability and support, diverse edge-focused form factors, and support for edge-constrained usage conditions.

The Intel Arc A380E is ideal for Edge AI applications like the smart EV charger station. Operating at a TDP of just 75W, the A380E is an ideal candidate for systems where power efficiency is critical. With 6GB of GDDR6 memory, A380E is well-equipped to handle complex computational tasks. This memory, combined with OpenVINO toolkit optimizations, allow the A380E to efficiently run the Mistral Instruct-7B PyTorch-based model quantized to INT4 precision on the EV charger system.

4.0 End-to-End Pipelines

The solution from DFI utilizes three unique AI pipelines for – (1) Face Detection, (2) Speech Recognition, and (3) LLM-based chatbot. Each pipeline is executed on a different hardware accelerator.

4.1 Face Detection Pipeline

Face detection employs a video analytics pipeline requiring video decode, pre/post-processing, display, and AI inference in the form of detection using the SSD-MobileNet-v2 AI model as shown in Figure 2. In this implementation, the entire face detection pipeline is executed on the 14th Gen Intel® Core Processor.

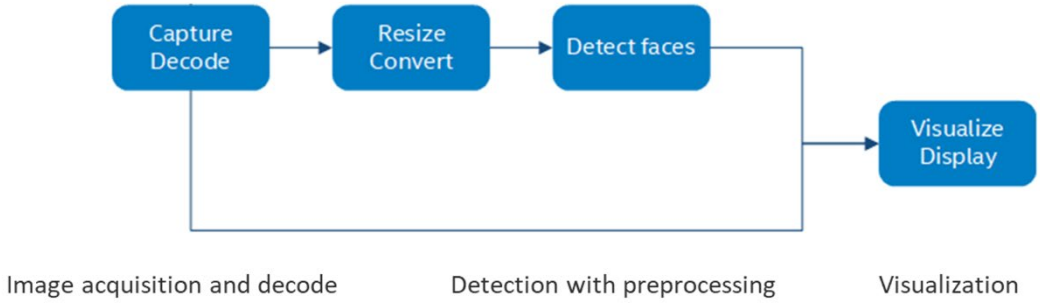

Figure 2: Face detection pipeline running on 13th Gen Intel Core Processor

4.2 Face Detection Pipeline Performance

Inference Device	AI Model	Resolution	Precision	Throughput (FPS)	Latency (ms)	Number of streams
CPU	SSD-MobileNet-v2	1080p@30 FPS	FP32	283.56	3.49	1

Table 1: Performance of face detection pipeline[5]

[5] Python Version: Python 3.10.12; OpenVINO Version: 2024.1.0

4.3 Automated Speech Recognition (ASR) Pipeline

Automated Speech Recognition is also known as speech-to-text. It is the process by which human speech is converted to text. ASR enables machines to mimic human ability to comprehend and respond to verbal communication.

In this implementation, the ASR pipeline uses the OpenAI Whisper model. It is an encoder-decoder based transformer model and excels in speech recognition and translation functions. The entire pipeline is executed on the built-in GPU of the 14th Gen Intel® Core processor.

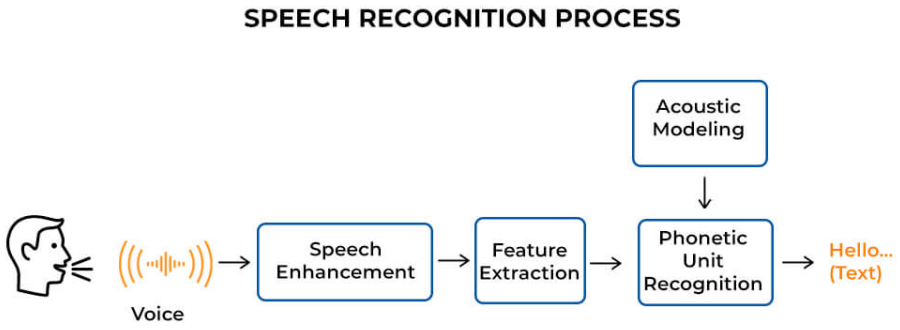

Figure 3: Automated Speech Recognition (ASR) pipeline with OpenAI Whisper model

4.4 Automated Speech Recognition Pipeline Performance

Inference Device	AI Model[6]	Precision	Latency	Number of streams
Integrated GPU	OpenAI Whisper	FP16	795 ms	1

Table 2: Performance of the automated speech recognition pipeline[7]

In this measurement, latency is the execution time of the entire transcription and based on an entire sentence.

[6] Source for Whisper model: https://github.com/openai/whisper/blob/main/README.md
[7] Python Version: Python 3.10.12; OpenVINO Version: 2024.1.0

4.5 Chatbot Pipeline

The chatbot pipeline, as shown in this implementation, uses the Mistral-7B Large Language Model (LLM). The 7.3 billion parameter model is quantized to INT4 precision and then loaded on the 6 GB memory provided by the Intel® Arc™ A380 discrete GPU[8]. Mistral-7B-Instruct-v0.1 provides impressive performance while supporting longer sequences efficiently in comparison to other models with higher parameters. The maximum number of tokens that the Mistral 7B Instruct v0.1 model can process is 8,192 tokens. This limit applies to the total length of input tokens and generated tokens combined. No limit default input token, the limit combined for input and output.

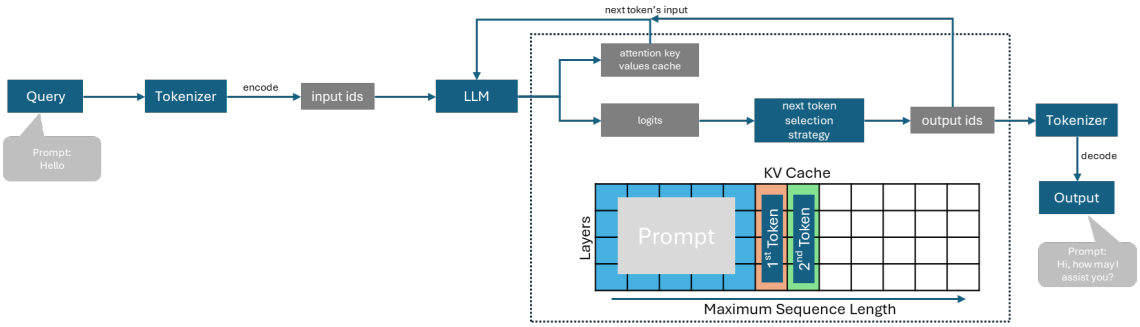

Figure 4: Chatbot pipeline utilizing the Mistral-7B LLM running on the Intel Arc A380 discrete GPU

4.6 Chatbot Pipeline Performance

Inference Device	AI Model	Precision	Throughput (tokens per second)	Number of streams
Discrete GPU	Mistral 7B-Instruct-v0.1	INT4	11.36	1

Table 3: Performance of the chatbot pipeline utilizing a large language model[9]

[8] Performance measured on Intel® Arc™ A380 as a proxy for Intel® Arc™ A380E. Results may vary.

[9] Python3: Python 3.10.12; IPEX LLM Version: 2.1.0b20240811; Discrete GPU: A380 6GB; Discrete GPU Driver: 24.13.29138.7; GPU Driver: intel-level-zero-gpu_1.3.29138.7

5.0 Conclusion

In this white paper, we explored the powerful synergy of Intel's cutting-edge technologies – the 14th Gen Intel® Core Processor with integrated GPU, the Intel® Arc Discrete GPU, and the OpenVINO Software Toolkit. The Electric Vehicle Charger AI solution leverages the unique capabilities of each component to deliver exceptional performance, scalability, and efficiency.

The 14th Gen Intel® Core Processors enable workload consolidation by providing multiple hardware accelerators built into a single package. For added AI scalability, Intel Core processors can be paired with discrete Intel® Arc GPUs. In edge AI deployments where cost, power, and security are key considerations, workload consolidation offers significant benefits in terms of reduced capital expenses, lower operational costs for power and cooling, and enhanced security due to fewer interfaces.

Our AI solution represents a convergence of innovation enabling developers and ISVs to unlock the full potential of AI workloads. By combining Intel's hardware and software technologies, we empower enterprises to build intelligent applications that drive real-world impact.

6.0 References

[1] NEX Intel® Core™ processors (14th Gen) (Raptor Lake-S Refresh) Gold Deck

[2] Heterogeneous AI Powerhouse: Unveiling the Hardware and Software Foundation of Intel® Core™ Ultra Processors for the Edge

[3] Intel® Graphics SR-IOV Enablement Toolkit

[4] Intel iGPU (Integrated Graphics) SR-IOV - The Catalyst for IoT Virtualization in Factory Automation

7.0 Appendix A: Face detection pipeline execution command and output log

```
(chatbot_env) user@ubuntu-desktop:~/Chatbot/models/intel/face-detection-0200/FP32$ benchmark_app -m face-detection-0200.xml -hint none -nireq 1 -nstreams 1
[Step 1/11] Parsing and validating input arguments
[ INFO ] Parsing input parameters
[Step 2/11] Loading OpenVINO Runtime
[ INFO ] OpenVINO:
[ INFO ] Build ..... 2024.1.0-15008-f4afc983258-releases/2024/1
[ INFO ]
[ INFO ] Device info:
[ INFO ] CPU
[ INFO ] Build ..... 2024.1.0-15008-f4afc983258-releases/2024/1
[ INFO ]
[ INFO ]
[Step 3/11] Setting device configuration
[Step 4/11] Reading model files
[ INFO ] Loading model files
[ INFO ] Read model took 7.95 ms
[ INFO ] Original model I/O parameters:
[ INFO ] Model inputs:
[ INFO ]     image (node: image) : f32 / [N,C,H,W] / [1,3,256,256]
[ INFO ] Model outputs:
[ INFO ]     detection_out (node: detection_out) : f32 / [...] / [1,1,200,7]
[Step 5/11] Resizing model to match image sizes and given batch
[ INFO ] Model batch size: 1
[Step 6/11] Configuring input of the model
[ INFO ] Model inputs:
[ INFO ]     image (node: image) : u8 / [N,C,H,W] / [1,3,256,256]
[ INFO ] Model outputs:
[ INFO ]     detection_out (node: detection_out) : f32 / [...] / [1,1,200,7]
[Step 7/11] Loading the model to the device
[ INFO ] Compile model took 70.91 ms
[Step 8/11] Querying optimal runtime parameters
[ INFO ] Model:
[ INFO ]   NETWORK_NAME: torch-jit-export
[ INFO ]   OPTIMAL_NUMBER_OF_INFER_REQUESTS: 1
```

```
[ INFO ]   NUM_STREAMS: 1
[ INFO ]   AFFINITY: Affinity.CORE
[ INFO ]   INFERENCE_NUM_THREADS: 12
[ INFO ]   PERF_COUNT: NO
[ INFO ]   INFERENCE_PRECISION_HINT: <Type: 'float32'>
[ INFO ]   PERFORMANCE_HINT: LATENCY
[ INFO ]   EXECUTION_MODE_HINT: ExecutionMode.PERFORMANCE
[ INFO ]   PERFORMANCE_HINT_NUM_REQUESTS: 1
[ INFO ]   ENABLE_CPU_PINNING: True
[ INFO ]   SCHEDULING_CORE_TYPE: SchedulingCoreType.ANY_CORE
[ INFO ]   MODEL_DISTRIBUTION_POLICY: set()
[ INFO ]   ENABLE_HYPER_THREADING: False
[ INFO ]   EXECUTION_DEVICES: ['CPU']
[ INFO ]   CPU_DENORMALS_OPTIMIZATION: False
[ INFO ]   LOG_LEVEL: Level.NO
[ INFO ]   CPU_SPARSE_WEIGHTS_DECOMPRESSION_RATE: 1.0
[ INFO ]   DYNAMIC_QUANTIZATION_GROUP_SIZE: 0
[ INFO ]   KV_CACHE_PRECISION: <Type: 'float16'>
[Step 9/11] Creating infer requests and preparing input tensors
[ WARNING ] No input files were given for input 'image'!. This input will be filled with
random values!
[ INFO ] Fill input 'image' with random values
[Step 10/11] Measuring performance (Start inference asynchronously, 1 inference
requests using 1 streams for CPU, limits: 60000 ms duration)
[ INFO ] Benchmarking in inference only mode (inputs filling are not included in
measurement loop).
[ INFO ] First inference took 7.31 ms
[Step 11/11] Dumping statistics report
[ INFO ] Execution Devices:['CPU']
[ INFO ] Count:          17015 iterations
[ INFO ] Duration:       60004.46 ms
[ INFO ] Latency:
[ INFO ]    Median:      3.79 ms
[ INFO ]    Average:     3.49 ms
[ INFO ]    Min:         2.08 ms
[ INFO ]    Max:         9.42 ms
[ INFO ] Throughput:   283.56 FPS
```

8.0 Appendix B: Code to load and convert Whisper model to OpenVINO IR format

Input for whisper model for the result in table.

Audio files:

wget
https://storage.openvinotoolkit.org/models_contrib/speech/2021.2/librispeech_s5/how_are_you_doing_today.wav -P audio

transcription: how are you doing today?

Whisper model link
https://github.com/openai/whisper/blob/main/README.md

Load and Convert to OpenVINO IR format:

```
import whisper
model_id= “base”
model = whisper.load_model(model_id,”cpu”)
model.eval()

whisper_encoder_xml =
Path(f"./models/whisper/whisper_{model_id}_encoder.xml")
whisper_decoder_xml = Path(f"./models/whisper/whisper_{model_id}_decoder
.xml")

mel = torch.zeros((1, 80, 3000))
audio_features = model.encoder(mel)

if not whisper_encoder_xml.exists():
   whisper_encoder_xml.parent.mkdir(exist_ok=True)
   encoder_model = ov.convert_model(model.encoder, example_input=mel)
   ov.save_model(encoder_model, whisper_encoder_xml)

if not whisper_decoder_xml.exists():
   whisper_decoder_xml.parent.mkdir(exist_ok=True)
   for idx, block in enumerate(model.decoder.blocks):
      block.forward = partial(block_forward, block)
```

```
block.attn.forward = partial(attention_forward, block.attn)
if block.cross_attn:
   block.cross_attn.forward = partial(attention_forward, block.
```

9.0 Appendix C: Code to download and quantize Mistral-7B-Instruct-v0.1 to INT4 precision

```
from ipex_llm.transformers import AutoModelForCausalLM
model_name = 'mistralai/Mistral-7B-Instruct-v0.1'
model = AutoModelForCausalLM.from_pretrained(
    model_name,
    load_in_4bit=True,
    optimize_model=True,
    trust_remote_code=True,
    low_cpu_mem_usage=True,
    revision="9ab9e76e2b09f9f29ea2d56aa5bd139e4445c59e",
    use_cache=True)
model.save_low_bit("./models/mistral-model")

model = model.half().to("xpu")
```