Low-Latency AI Execution Systems for Resource-Constrained Devices

Arihant Rangoli, YSP Student Researcher, *St. John's Preparatory School*

Mohammad Abdi, Ph. D. Candidate, Electrical and Computer Engineering, *Northeastern University*

Shariar Rifat, Ph. D. Candidate, Electrical and Computer Engineering, *Northeastern University*

Francesca Menghello, Principal Research Scientist, Electrical and Computer Engineering, *Northeastern University*

Francesco Restuccia, Associate Professor, Electrical and Computer Engineering, *Northeastern University*

Northeastern University
Michael B. Silevitch and Claire J. Duggan Center for STEM Education

Northeastern University
College of Engineering

Abstract

Artificial intelligence is becoming increasingly important in **cyber-physical systems** such as **autonomous robots**, **virtual reality**, drones, and connected vehicles, where real-time processing is essential. However, many **resource-constrained** devices lack the computational power to execute modern deep learning models with low latency. This research investigates **semantic edge computing**, in which an **AI model** is partitioned across heterogeneous devices to improve performance while maintaining accuracy. The project implemented distributed inference using the **ResNet-50** neural network on a **Raspberry Pi**, **Jetson Orin**, and **GPU edge workstation**. Performance differences between **CPU** and **GPU** execution were evaluated through latency measurements, and the model was prepared for deployment in **Unity** using the **ONNX** format for **Meta Quest** virtual reality headsets. Additionally, the architecture of other **ResNet-based models** was investigated, along with techniques for reducing inference latency without sacrificing model accuracy. This work advances the development of resilient edge AI systems for **low-latency**, real-time **computer vision** applications.

Experimental Methods and Materials

Figure 4. System Hardware Overview

Raspberry PI

- ARM CPU
- Executes first part of the model
- Low Power Consumption

NVIDIA JETSON ORIN

- CPU + GPU
- Higher compute capability
- Executes beginning layers

GPU Edge Workstation

- High-performance GPU
- Executes remaining layers
- Returns final prediction

Meta Quest 2

- Runs Unity application
- Displays results
- Interactable World

- Implemented distributed **ResNet-50 inference** in **Python** using **PyTorch**.
- Partitioned the model across a Raspberry Pi, Jetson Orin, and GPU workstation using **semantic edge computing**.
- Executed initial layers on edge devices and offloaded remaining layers to the **GPU workstation**.
- Evaluated inference on the Raspberry Pi (**CPU**) and Jetson Orin (**CPU/GPU**).
- Used **SSH** for **remote deployment** and debugging.
- Exported the model to **ONNX** and deployed it in Unity.
- Integrated the pipeline with **Meta Quest 2** for future VR deployment.
- Measured **inference latency** and **execution time** across all hardware configurations.

Conclusion

- Demonstrated successful distributed ResNet-50 inference using semantic edge computing across **resource-constrained devices** and a GPU workstation.
- Achieved low-latency **AI inference** while maintaining model performance.
- Established a foundation for **real-time computer vision** on Raspberry Pi, Jetson Orin, and Meta Quest platforms.

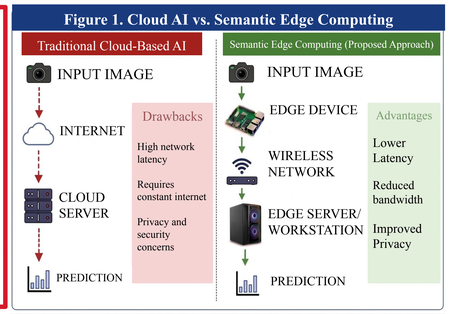
Future Steps

- Integrate Meta Quest **passthrough cameras** for live image capture.
- Perform **real-time object recognition** using the distributed ResNet-50 pipeline.
- Enable immersive **AI-powered VR applications.**

Background

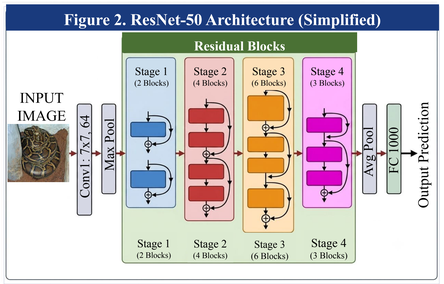
- Traditional **cloud AI** sends raw images to remote servers, **increasing** latency and bandwidth usage.
- **Semantic edge computing** executes the initial ResNet-50 layers on the **edge device**.
- Only compact **feature representations** are transmitted to the edge server.
- The edge server completes **inference** and returns the **prediction**.
- This approach **reduces latency** and **bandwidth** while maintaining inference accuracy.

Figure 1. Cloud AI vs. Semantic Edge Computing



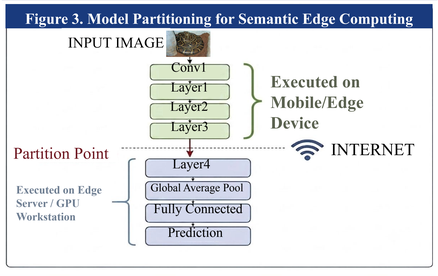
- ResNet-50 is a 50-layer **convolutional neural network** with residual connections that improve training and accuracy.
- Its **computational complexity** makes full execution difficult on resource-constrained devices.
- **Distributed inference** addresses this limitation by partitioning the network across **multiple devices**.

Figure 2. ResNet-50 Architecture (Simplified)



- ResNet-50 is partitioned between an edge device and a **GPU edge server**.
- **Early layers** execute on the Raspberry Pi or Jetson Orin to extract **feature representations**.
- Intermediate features are **transmitted wirelessly** for completion of the remaining layers.
- Distributed execution reduces the **computational load** while preserving model performance.

Figure 3. Model Partitioning for Semantic Edge Computing



Results

- All three configurations correctly classified the input image as **rock python**.
- **PC-only** execution served as the baseline with **33.56 ms** inference latency.
- **PC + Raspberry Pi** reduced latency to **21.93 ms** (**35%** improvement over the baseline).
- **PC + Jetson Orin** achieved the lowest latency at **13.91 ms** (**59%** faster than the baseline).
- GPU acceleration on the **Jetson Orin** provided the greatest performance improvement.
- The **Jetson** configuration produced a substantially higher prediction confidence (**95.82%**) than the PC-only and **Raspberry Pi** configurations (**51.83%**), despite using the same input image.

TOP 5 Predictions

rock python	**— 45.27%**
boa constrictor	**— 2.78%**
horned viper	**— 1.75%**
sidewinder	**— 1.41%**
night snake	**— 1.06%**

Stem (on-device): 40.0 ms
Network: 139.5 ms
Server tail: 36.4 ms
Total Latency: 215.8 ms

Configuration	Latency	Top Pred.	Confidence
Plain PC (baseline)	33.56 ms	Rock python	51.83%
PC + Raspberry Pi Split	21.93 ms	Rock python	51.83%
PC + Jetson Split	**13.91 ms**	**Rock python**	**95.82%**

- Developed a custom **Unity VR** application using **ResNet-50** exported in **ONNX** format.
- **Interactable world** with multiple different **image categories**.
- Implemented a **client-server architecture**, with the **Meta Quest** executing the **model stem** locally and offloading the remaining layers to a remote server.
- Achieved a total **inference latency** of **222.8 ms**, consisting of:
  - **61.7 ms** local headset inference
  - **125.6 ms** network transmission
  - **35.6 ms** server-side inference
- Latency exceeded **local configurations** (**13.91–33.56 ms**) due to **network communication** overhead.
- The distributed architecture enables **real-time VR applications** by offloading computationally intensive processing from the headset.
- Maintained **accurate** classification (**rock python**, **45.27%** confidence) despite the **distributed execution**.

References

[1] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in International Conference on Learning Representations (ICLR), 2021.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[3] A. Kirillov et al., "Segment Anything," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2023, pp. 4015–4026, doi: 10.1109/ICCV51070.2023.00371.

[4] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in Advances in Neural Information Processing Systems, vol. 32, 2019.

[5] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, Oct. 2016, doi: 10.1109/JIOT.2016.2579198.

Acknowledgements

Mobile Embedded NeTworked Intelligent Systems (MENTIS) Laboratory

Mohammad Abdi, Ph. D. Candidate, Electrical and Computer Engineering, Northeastern University

Shariar Rifat, Ph. D. Candidate, Electrical and Computer Engineering, Northeastern University

Francesca Menghello, Principal Research Scientist, Electrical and Computer Engineering, Northeastern University

Francesco Restuccia, Associate Professor, Electrical and Computer Engineering, Northeastern University

Center for STEM Education

Claire Duggan, Executive Director

Jennifer Love, Associate Director

Ahmed Othman, **Frances Corcell**, **Kenneth Rene Santizo Carbajal**, YSP Coordinators

Nicolas Fuchs, Program Manager

Mary Howley. Administrative Officer

Young Scholars Program at Northeastern University
Claire Duggan, Program Director

This work was supported by the National Science Foundation