Neural Network Acceleration for Pervasive Extreme Edge Intelligence

Honors Poster Presentation
May 7, 2025

Varun Jain | Rohit Sahu, Dr. Henry Duwe

1. Pervasive Intelligence and its Importance

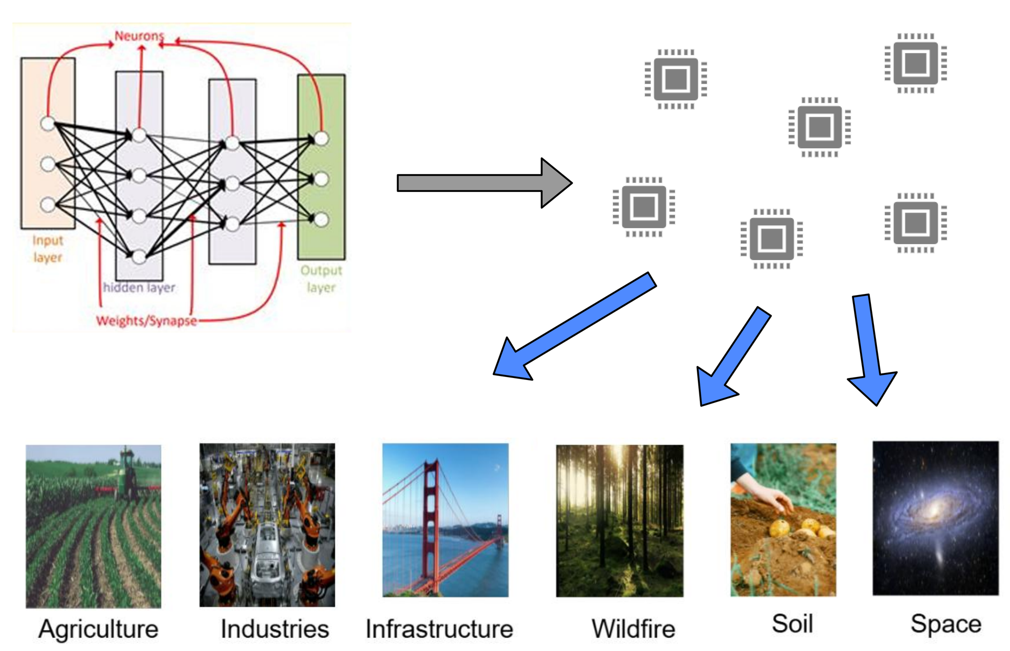


Fig 1. Pervasive Intelligence and Importance

2. Why do we need Local Intelligence?

- Devices often function **without internet connectivity**
- Often deployed in **remote** or **hard-to-access locations**
- All data processing must be done **locally on the device**
- Demands **efficient, low-power on-device AI capabilities**

3. Why choose Neural Networks

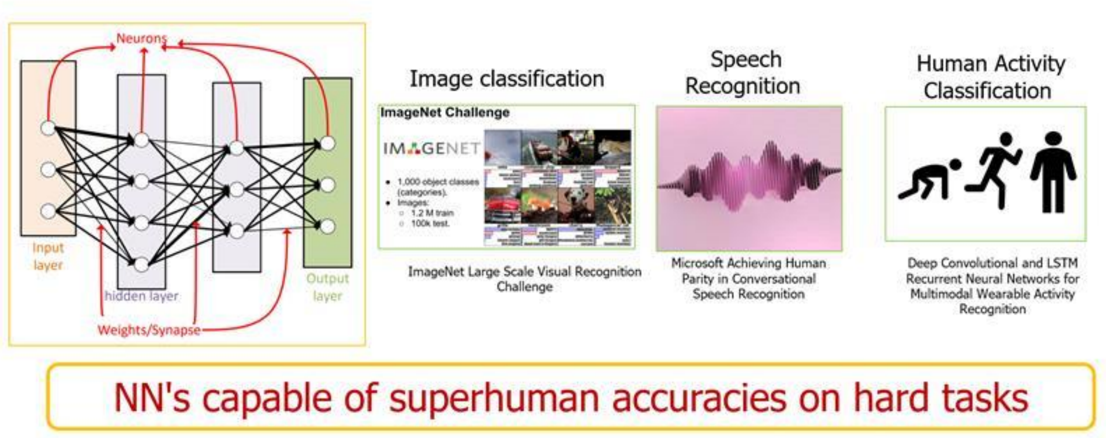
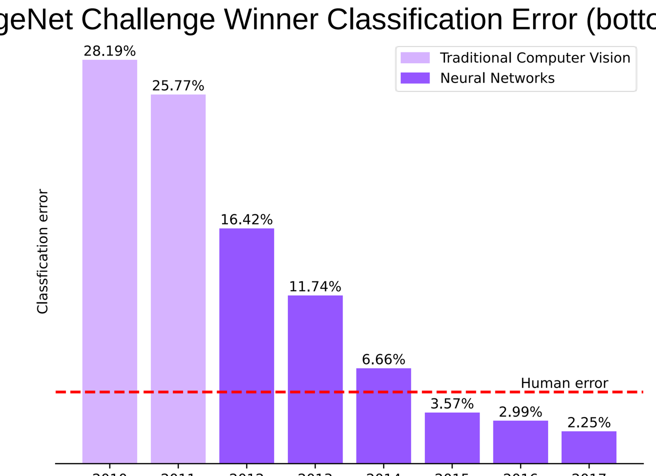


Fig 2. Neural Networks and Capabilities (top), ImageNet Challenge Winner Classification Error (bottom)



- NNs offer **high performance** with **lower memory usage**
- Ideal for **resource-constrained devices** in pervasive intelligence systems

4. Characteristics of Nodes

- Nodes often operate **in isolation** without continuous power
- Deployed in **large numbers** and must have **low cost per node**.
- Depend on **ultra-low-power microprocessors**
- Designed under **strict energy constraints**
- Feature **limited storage** and **computational resources**
- *Table 1* compares common microprocessors:
  - **Non-Volatile Memory (NVM)**
  - **SRAM**
  - **Operating Frequency**

Processors	NVM (KB)	SRAM (KB)	Freq (MHZ)
ATSAML11E16A Rev. B	64	16	32
EFM32HG322F64 Rev. B	64	8	25
STM32L412 Rev. A	128	40	80
MSP432P401R Rev. C	256	16	48
MSP430FR5969	64	2	16
MSP430FR5994	256	8	16

Table 1. Common Microprocessors for Edge Computing

5. Project Objective

- Develop and deploy a **high-capability Neural Network** under **resource constraints**
- Target platform: **MSP430FR5994 (MSP430) microcontroller**
- Dataset: **CIFAR-10 image classification**
- Optimize the **accuracy–energy tradeoff**
  - **Energy efficiency directly impacts latency**

6. Related Works

- **Current Approaches**: TinyML systems often rely on compression, quantization, and optimized runtimes to enable inference on memory- and energy-constrained devices.
- **NAS-Based Model Design**: We use Neural Architecture Search (NAS) to generate models that fit MSP430 constraints by minimizing size while preserving accuracy.
- **Pooling for Efficiency**: NAS-selected models utilize global average pooling in later layers to reduce parameter count and overfitting risk.

7. Methodology

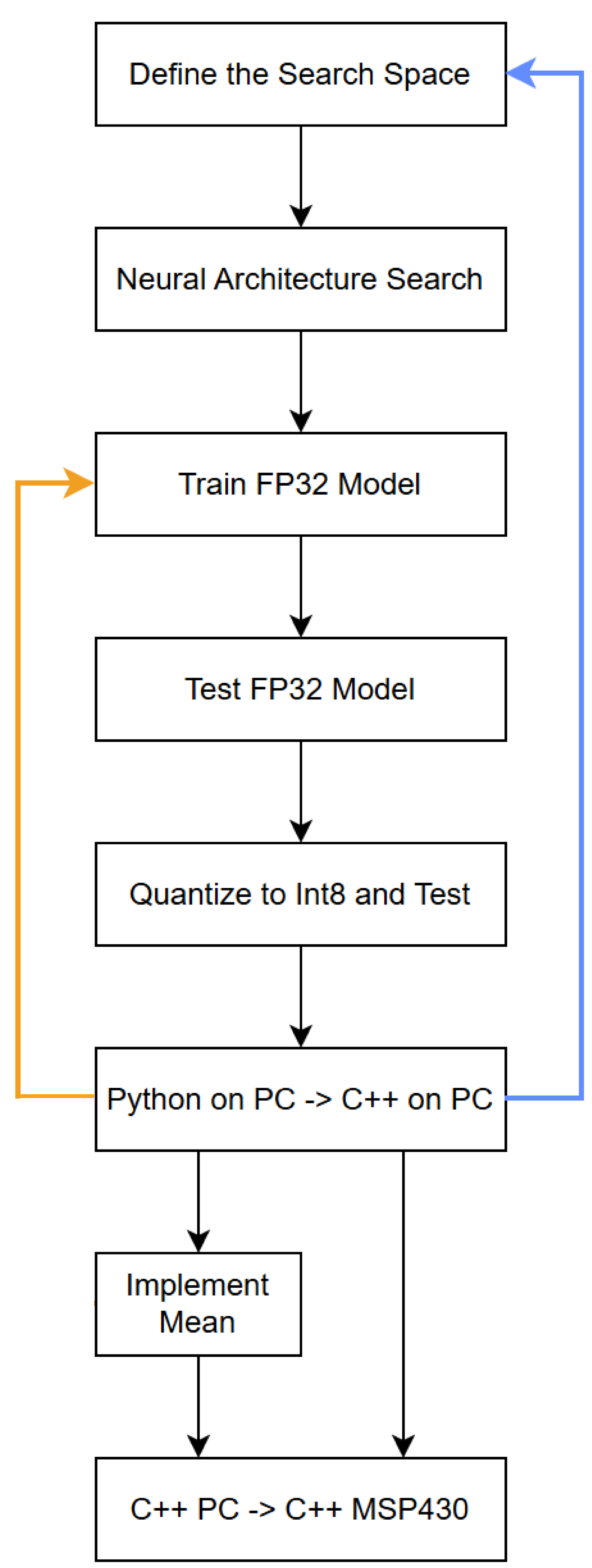


Fig 3. Methodology Workflow

- Neural Network **operator options are predefined**
- **Neural Architecture Search (NAS)** generates a candidate model
- Model is:
  - Trained using **FP32**
  - Tested
  - **Quantized to Int8** for MSP430 deployment
- Tested in a **C++ simulation environment** (MSP430-like)
- Since C++ lacked a **Mean operator**:
  - *Search space modified* to create an **Alternate model** (Blue Arrow)
  - *Original model also modified* to remove Mean (creating a **Non-Mean model**) (Amber Arrow)
- **Mean operator implemented** separately for original (**Mean model**)
- All three models:
  - Trained → Quantized → Tested
  - Deployed and tested on **MSP430 C++ environment**
- **Result Labels**:
  - *Mean model* – original NN with Mean
  - *Non-Mean model* – modified NN without Mean
  - *Alternate model* – generated from adjusted NAS

8. Implementing the Mean Operator

- **TFLite Micro** lacked support for the **Per-Channel Mean** operator
- Required **custom implementation** due to framework limitations
- Arithmetic in **quantized (Int8)** vs **real (FP32)** domains differs
- *Image on the right* shows comparison of Mean operations

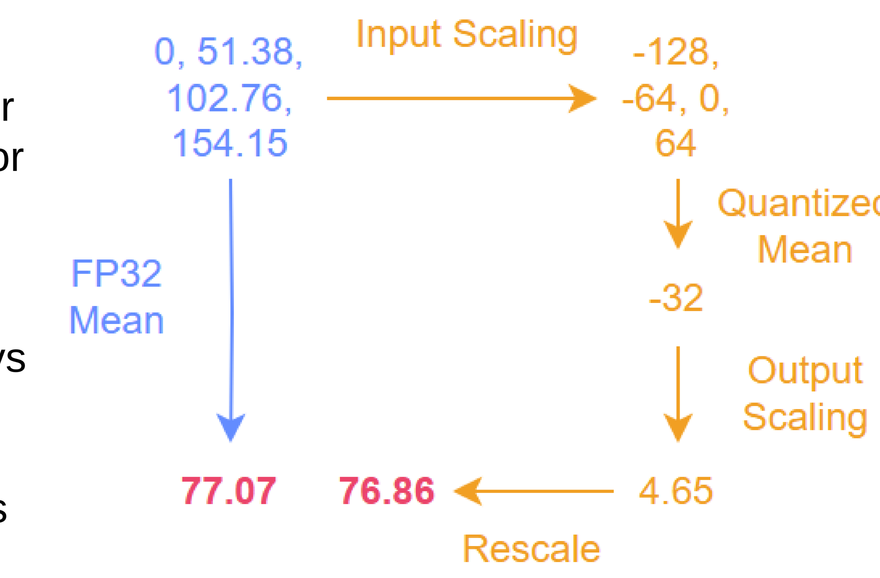


Fig 4. FP32 Mean vs. Int8 Mean

9. Results

9.1 Accuracy and Memory Requirements

Alternate and **Mean models** fit **within our constraints**

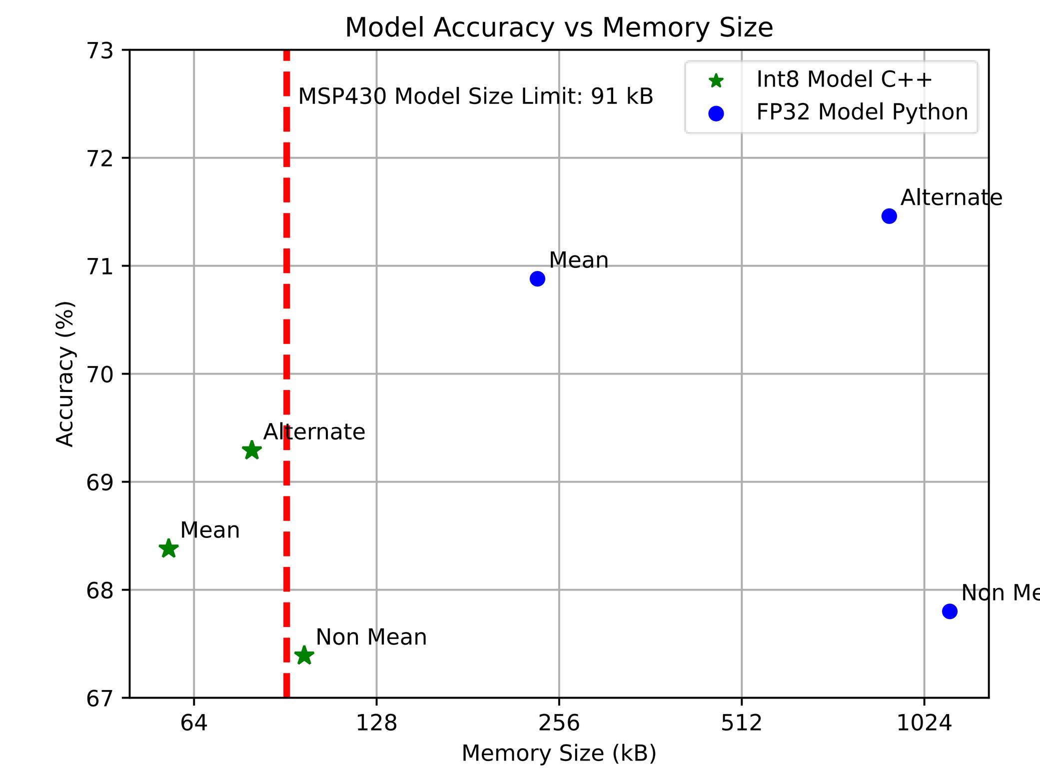


Fig 5. CIFAR-10 Model Accuracy vs Memory Size

9.2 MSP430 Energy and Latency

- Models evaluated in MSP430-compatible C++ environment
- **Int8 Mean** and **Int8 Alternate** models were **fully deployed and measured**
- Other models (marked with *) have **estimated energy and latency values**

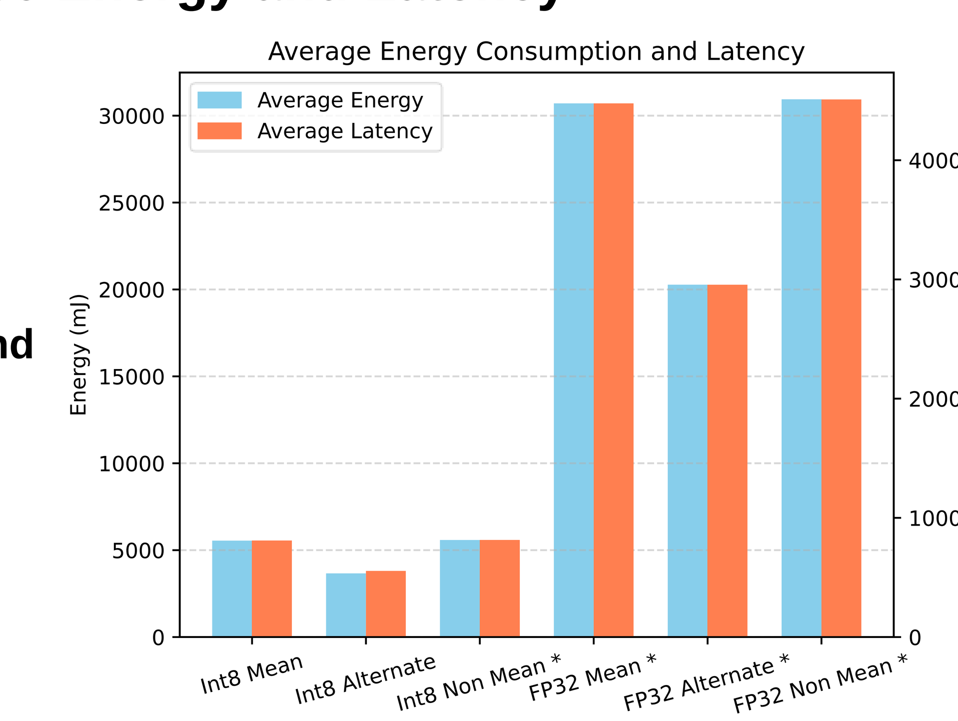


Fig 6. Energy and Latency on MSP430

10. Conclusion

- Deployed **quantized neural networks** with a custom **Mean operator** on the **MSP430 microcontroller**
- The **Int8 Alternate model** offers the best balance of **accuracy**, **energy efficiency**, and **memory usage**

References

[1] R. Sahu, V. Deep, and H. Duwe. HANNA: Harvesting-Aware Neural Network Architecture Search for Batteryless Intermittent Devices. 1-10. 10.1109/IPCCC59868.2024.10850328.
[2] ImageNet, "Large Scale Visual Recognition Challenge (ILSVRC),". Available: https://www.image-net.org/challenges/LSVRC/
[3] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Technical Report, Univ. of Toronto, 2009.
[4] C. Banbury et al., "Benchmarking TinyML Systems: Challenges and Direction," *arXiv:2003.04821*, 2020.
[5] V. Narayanan, R. Sahu, J. Sun, and H. Duwe, "BOBBER: A Prototyping Platform for Batteryless Intermittent Accelerators," in *Proceedings of the 2023 ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, 2023.
[6] R. Sahu, R. Toepfer, M. D. Sinclair, and H. Duwe, "DENNI: Distributed Neural Network Inference on Severely Resource Constrained Edge Devices," *2021 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, 2021