Leveraging Ultra-Low-Power Wearables Using Distributed Neural Networks

Meghna Roy Chowdhury, Archisman Ghosh, Md Faizul Bari and Shreyas Sen
{mroycho, ghosh69, mbari, shreyas}@purdue.edu

2024 IEEE 20th International Conference on Body Sensor Networks (BSN) | 979-8-3315-3014-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/BSN63547.2024.10780699

***Abstract*—Traditional deep learning models incur high computational overhead (approximately 1-30W) and are unsuitable for Ultra-Low-Power (ULP) wearable systems like smart glasses. In contrast, TinyML architectures are power-efficient but less accurate. We propose distributing a neural network (NN) between a ULP wearable node and a resource-rich hub to maintain high accuracy and low power consumption for applications like human-machine vision. Output features from the node are transmitted to the hub via low-power communication, where the remaining network runs on traditional GPUs. We introduce a Figure of Merit (FoM) to determine the optimal NN distribution point and a customized multiplier unit for ULP operation. Achieving ULP of 284µW and 9mW for different Autoencoder (AE) networks, our approach is 700× lower in power consumption compared to traditional GPU implementation.**

***Index Terms*—Neural Network, wearables, tinyML, ultra-low power, GPU, Autoencoder, power-efficient MAC**

I. Introduction

A. Motivation

The advent of wearable technology, particularly in the realm of low-power-body-sensor systems, has revolutionized personal devices. For instance, Meta×Rayban's latest smart glasses capture high-definition images and videos at 30 fps from an inbuilt camera [1]. These glasses, and other similar wearable devices, are increasingly integrating body sensors to provide real-time health and activity monitoring. However, they often have a short battery life and require frequent charging. This limitation underscores the need for low-power sensing and cost-effective inference algorithms in wearable technology to extend device usability and enhance functionality while performing advanced machine learning (ML) for inferences.

Recent advancements in low-power sensing and communication technologies (e.g. Bluetooth, Wireless Body Area Network, Human Body Communication) have significantly improved the functionality and efficiency of wearable devices and body sensors [2]. Various ML architectures have been explored for vision tasks, with AEs being particularly notable for their applications in image reconstruction and denoising [3]. However, implementing these on ULP wearable nodes remains a challenge due to limited hardware constraints.

TinyML offers a promising approach by optimizing ML models specifically for wearables, by reducing the computation cost (CC) or FLOPs [4]. For wearable devices and body sensors, this optimization is crucial as it directly impacts the battery life and usability of the device. However, these models typically experience a 10-15% reduction in accuracy compared to traditional NNs due to the power and memory limitations in wearables [5].

Another approach for efficient ML is called Data Offloading, where data is transferred to cloud servers for implementation. Recent research has explored this for ML classification tasks. For example, Yang et al. demonstrated the effectiveness of sharing low-level feature maps between a camera and a gateway, Pinkham et al. optimized performance through the use of on-sensor and on-device caches [6], [7]. These studies highlight the potential of distributed computing to improve the efficiency of wearable devices and body sensors. However, this approach may increase communication costs, introduce latency, and raise privacy concerns.

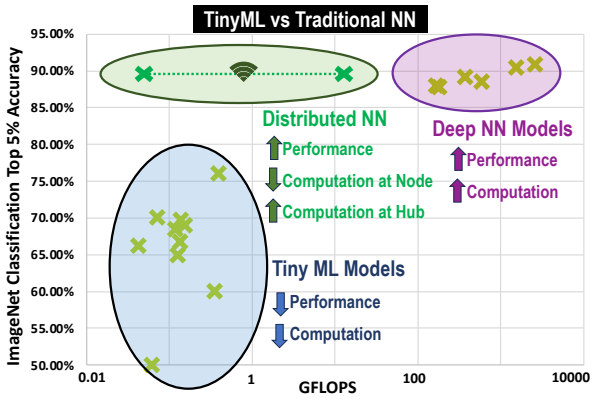

Fig. 1. Distributed NN solves tradeoff between FLOPs & Accuracy

To bridge the gap between accuracy and CC, we propose distributing traditional NNs between a ULP wearable node and a resource-rich hub, aiming to achieve high performance while maintaining $\leq$10mW at the wearable node. Fig. 1 shows that distributing NN can solve the tradeoff between TinyML and traditional NN. To the best of the authors' knowledge, this is the first work to use some data offloading concepts specifically for ULP ($< 10mW$) wearable machine vision using AE.

B. Our contribution

Existing literature on offloading primarily focuses on classification tasks, which typically involve dimension reduction. However, machine vision problems using AE exhibit fluctuating data volumes. Additionally, these studies lack power consumption data for wearable nodes and emphasize efficient processing on resource-rich hubs. Our contributions are:

- Conceptualizing and analyzing the **distribution of traditional NNs for ULP wearables**, by developing a **FoM** based on **data volume (DV)** and **computation cost (CC)** to find the optimal distribution point.
- **Formalizing conditions for distributing NNs** in terms of **Communication Energy** (E_{comm}) and **Computation Energy of the NN** ($E_{hub} + E_{node}$) specifically for **ULP wearable nodes**.
- **Designing and benchmarking a customized multiplier specially** tailored for **ULP wearable nodes, operating**

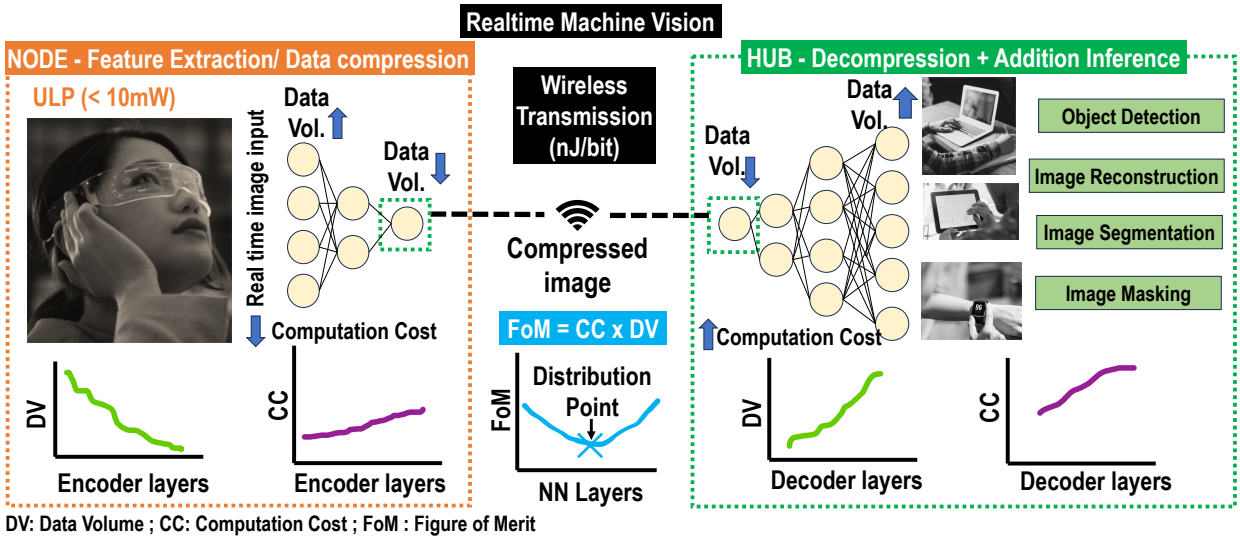

Fig. 2. Proposed concept to bridge the tradeoff between computation cost and performance by Distribution of NN to a ULP node and a resource-rich hub.

at ≤10mW while processing **data from body sensors at 30 fps**.

II. Proposed Idea

Our proposed distribution method is illustrated in Fig. 2, where the ULP wearable node is depicted as smart glasses that capture real-time images as input to an NN, transmitting compressed features maps to a resource-rich hub (smartphone, laptop, etc.) via low-power wireless communication. The NN distribution is guided by a FoM, which considers the DV and CC for each NN layer. We use image reconstruction with an AE as an illustrative example. Our methodology is directly applicable to other AE networks, and the general framework applies to other NNs.

A. Architecture of AE used for Reconstruction

Fig.3(a) illustrates the architecture of an AE for image reconstruction trained on the CelebA dataset [8]. The model receives a 128×128 RGB image as input and consists of 3 encoding, a latent space, and 3 decoding blocks. Each encoding block, comprising Conv2D, MaxPool, and Batch Normalization (BN) layers, facilitates feature extraction, dimensionality reduction, and training acceleration, respectively. The latent space, a 1D array derived from the third encoding block's output feature map, is represented by Dense blocks, symbolizing the compressed image. The decoding blocks, including ConvTranspose2D, UpSample2D, and BN layers, restore the image to its original format from the compressed 1D representation.

B. Formalization of Distribution point

In any CNN, the computationally intensive layers include the Conv2D, ConvTranspose2D, and Dense layers, as this is where the multiplication of the input feature with n filters takes place. The number of MAC operations for this depends on various hyperparameters, such as the number of filters, kernel size, strides, etc. MaxPool and Upsample layers have minimal contribution to the overall FLOPs, as they are used for dimensionality reduction and increase, respectively. Eq.1(a) and 1(b) provide the formula for the number of MAC units in the computationally heavy layers. Eq.2 provides the formula for calculating the data volume of the output feature in various layers. Based on the DV and CC, we establish a FoM for each layer using Eq.3 which guides us in determining the optimal point to partition the AE and distribute it between the node and hub. (Note: The CC refers to the cumulative CC in terms of #MAC.) As shown in Fig. 3(b), the point of distribution lies in the Dense layer, where FoM is minimum. (Note that the computation energy per layer is proportional to the number of FLOPs.) Fig.3(c) illustrates the distribution of the NN based on the FoM. The features of the NN at the distribution point are transmitted through low-power communication from the node to the hub where the rest of the NN is implemented.

Computation Cost (in terms of Number of MAC):

Convolution Layers: $\#MAC = (Kernel\_size \times \#Channel) \times (\frac{Output\_size}{Stride} \times \#Filters)$ (1a)

Dense Layers: $\#MAC = (Input\_size \times Output\_size)$ (1b)

Data Volume: $DV = Width \times Height \times \#Channels$ (2)

Figure of Merit: $FOM = DV \times CC$ (3)

C. Communication Requirements

Based on the understanding that energy efficiency for wireless communication technologies like Bluetooth or emerging Human Body Communication typically ranges from nJ/bit to 100s of pJ/bit [2], the criteria for determining *if an NN should be distributed* can be formalized as follows:

- *The NN should incorporate dimensionality reduction* because E_{comm} per bit is orders of magnitude higher than computation energy per bit. Thus, ensuring that the DV of the feature map is suitable for transmission is crucial.
- Due to the ULP constraint of the wearable node & the hub being resource-rich, *Energy of NN at the node* (E_{node}) << *Energy of NN at the hub* (E_{hub}). The contrary defies the concept of Distribution of NN.
- $E_{node} \approx E_{comm}$ for lossless data transmission from the wearable node to the hub.
- The total CC of the NN $(E_{node} + E_{hub}) >> E_{comm}$. Otherwise, the complete NN could be implemented in the node.

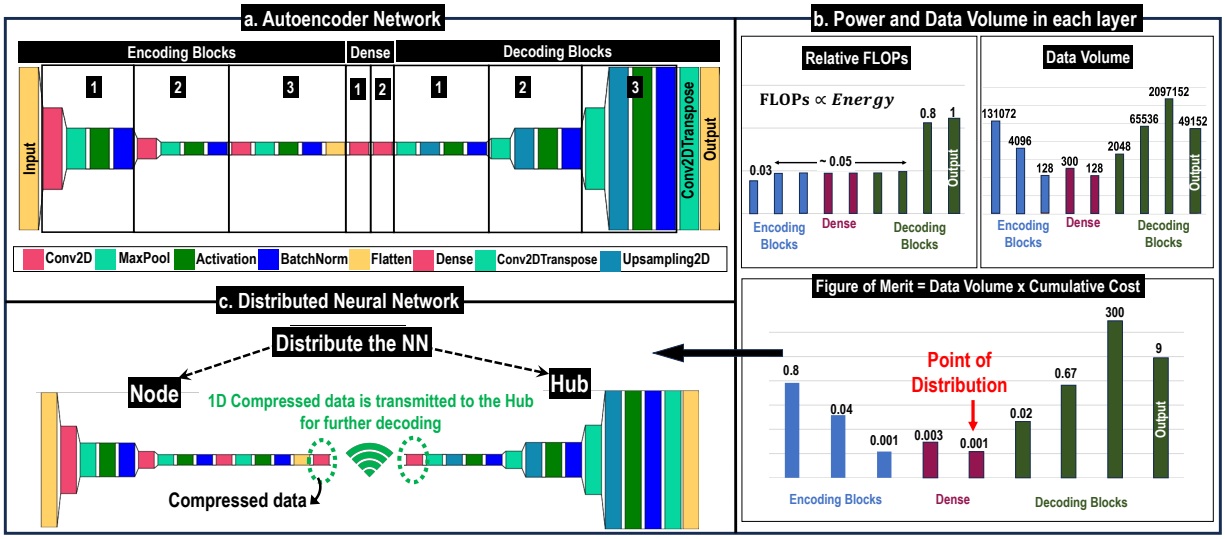

Fig. 3. (a) AE model before distribution. (b) Relative energy, data volume, and FoM of each layer. (c) Distribution of NN based on FoM.

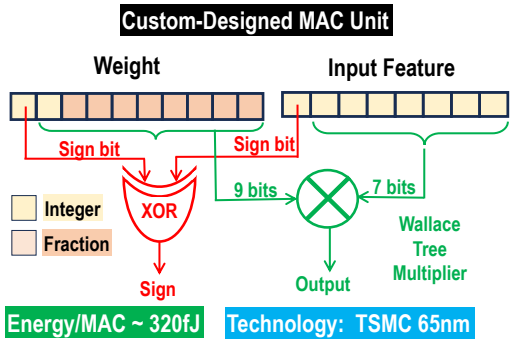

Fig. 4. Customized multiplier unit using fixed-point number for the 8x10 bit multiplication of Input Features (8bits) with Weights (10bits).

D. ULP Wearable node

To ensure compatibility with low-power communication and extend the battery life of wearable devices, the energy consumption of the NN at the node (E_{node}) must not exceed 10mW. Since the wearable node has fewer FLOPs than the hub, it needs specialized low-power hardware. We designed a customized multiplication unit for a ULP wearable node (see Fig. 4) which performs 8×10-bit floating-point multiplications of features (8 bits) with weights (10bit-decimal) using Fixed Point (FP) notation using the Wallace Tree Multiplier [9]. The output sign is determined by an XOR operation on the MSB of the feature and weight, as both MSBs represent the sign bit. The MAC unit was coded in Verilog HDL, and synthesized with Synopsys Design Compiler using 65nm TSMC CMOS technology. We use Eqs.1(a) and 1(b) to determine FLOPs per layer and Eq.(4) for power estimation.

$$\textbf{Power} = (\#MAC) \times (Energy/MAC) \times (Frames/sec) \quad (4)$$

E. Other Resource Considerations

Factors such as throughput, latency, memory & battery capacity are crucial for NN distribution. These factors are interconnected and indirectly related to CC. A larger battery increases throughput by supporting more computations, more memory allows greater data storage, and lower latency enables quicker inferences. All these enhance throughput. Essentially, when discussing CC, throughput, and DV, we are not ignoring the device's battery capacity, memory, latency, and thermal envelope, which are vital for optimizing ULP node distribution. There is a trade-off between E_{comm} and latency: reducing CC can increase latency, while reducing latency may raise CC.

III. Evaluation of Proposed Methodology

We evaluate the scalability of the proposed methodology using two different traditional AEs, and their power consumption in the wearable node with and without distribution.

A. Scalability of Proposed Distribution of NN

We first consider an example AE model for face image reconstruction using the CelebA dataset [8], which achieves an accuracy of 85%. It consists of 4 encoding (B1-B4) & decoding (B5-B8) blocks (Fig.5(a)). Based on the FOM defined in Eq.3, we find the optimum distribution point at the Dense Layer B5. Since this Dense layer has the least data volume, the feature map can be transmitted with reduced loss.

The second example (Fig.5(b)) is an image-denoising AE with 13 blocks (5 Encoding, 1 Dense, 7 Decoding) taken from [10] but without skip connections. It achieves an accuracy of 86% with the dataset [11]. The input and output dimensions are $256\times256\times3$. On calculating the FoM, the optimum distribution point is in B3. This demonstrates that the optimal point for distribution may not always be in the latent space and can be applied to other NN architectures as well.

B. Low Power Encoding under Distributed NN paradigm

Fig.4 shows our customized multiplier unit for the ULP wearable node, consuming an average processing energy of $320fJ/MAC$. However, a recent advance in in-memory/near-memory computing shows that the energy requirement for memory access and memory blocks can be limited within an 80% overhead relative to computational resources (MAC units) [12]. (Note that memory access and memory energy optimization are problem-dependent and will be explored by a monolithic implementation as part of future work.) In comparison to NVIDIA's MCM-GPU architecture (~250pJ/MAC) [13] and an ASIC implementation of a MAC unit (~1.9pJ/MAC) [14],

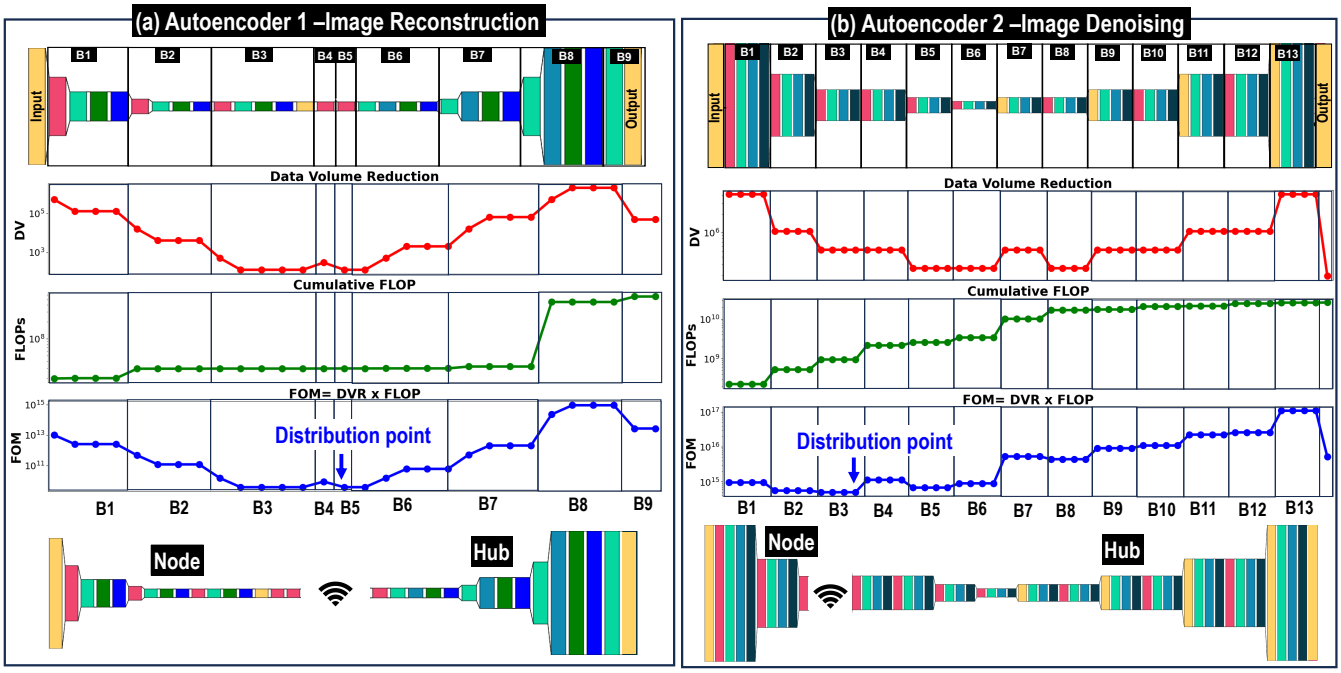

Fig. 5. (a) AE1- Image Reconstruction with Distribution point at B5 (b) AE2- Image Denoising with Distribution point at B3

our multiplier consumes ∼700× and ∼6× lower energy respectively. It should be noted that the custom ASIC for CNN in [14] utilizes generic fused FLOPs, while ours performs 8×10 bit FP multiplication. Fig.6 shows the improved power numbers achieved at the node with our customized multiplier at $30fps$. The analysis of our evaluation shows that we can achieve a ULP wearable node of ∼10mW with good inference accuracy with the Distribution of NN and our custom multiplier.

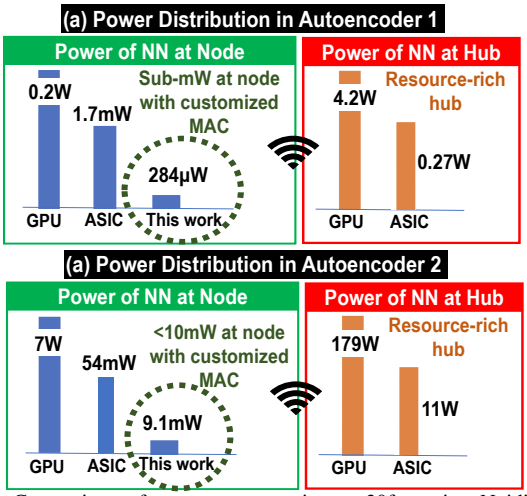

Fig. 6. Comparison of power consumption at 30fps using Nvidia MCM GPU [13], ASIC [14] and Custom Multiplier, with and without distribution of Neural Network in (a) AE1 & (b) AE2.

IV. Conclusion

Our proposed methodology of Distribution of NNs between a wearable node and a GPU-equipped hub balances the trade-off between cost and accuracy. The optimal distribution point is found using a FoM, which is based on CC and DV. We established conditions for NN distribution and designed a low-power MAC unit consuming $320fJ$, setting a new benchmark for ULP-embedded wearable sensing nodes. The scalability of our approach was confirmed by analyzing two AEs for image reconstruction and denoising. Using our custom multiplication unit, the power consumption at the wearable node is $\approx 284\mu W$ and $\approx 9.1mW$, which are 700× & 6× lower than state-of-the-art GPU and ASIC, respectively. This method applies to AE-like NNs and can be extended to other NNs. Our approach paves the way for energy-efficient ULP AI applications in machine vision for mobile health and personalized healthcare.

References

[1] https://about.fb.com/news/2023/09/new-ray-ban-meta-smart-glasses/ (Accessed on 05/30/2024).

[2] S. Maity et al., "Wearable health monitoring using capacitive voltage-mode human body communication," in *EMBC*, IEEE, 2017.

[3] S. Chen. et al., "Auto-encoders in deep learning—a review with new perspectives," *Mathematics*, 2023.

[4] W. Liu et al., "A survey of deep neural network architectures and their applications," *Neurocomputing*, 2017.

[5] https://paperswithcode.com/sota/image-classification-on-imagenet?metric=Top%205%20Accuracy&dimension=GFLOPs (Accessed on 05/13/2024).

[6] S.-W. Yang et al., "Distributed neural networks for scalable real-time analytics," 2017. US Patent App. 14/849,924.

[7] R. Pinkham et al., "Near-sensor distributed dnn processing for augmented and virtual reality," *JETCAS*, 2021.

[8] https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html. (Accessed on 05/13/2024).

[9] C. S. Wallace, "A suggestion for a fast multiplier," *IEEE Trans Comput.*, 1964.

[10] https://github.com/nsarang/ImageDenoisingAutoencdoer/tree/master. (Accessed on 05/13/2024).

[11] https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html. (Accessed on 05/13/2024).

[12] N. Verma et al., "In-memory computing: Advances and prospects," *SSCS Magazine*, 2019.

[13] A. Arunkumar et al., "Mcm-gpu: Multi-chip-module gpus for continued performance scalability," *ACM SIGARCH*, 2017.

[14] J. Park et al., "9.3 a 40nm 4.81 tflops/w 8b floating-point training processor for non-sparse neural networks using shared exponent bias and 24-way fused multiply-add tree," in *ISSCC*, IEEE, 2021.