HAC-POCD: Hardware-Aware Compressed Activity Monitoring and Fall Detector Edge POC Devices

Hasib-Al Rashid
University of Maryland, Baltimore County
hrashid1@umbc.edu

Tinoosh Mohsenin
Johns Hopkins University
tinoosh@jhu.edu

2023 IEEE Biomedical Circuits and Systems Conference (BioCAS) | 979-8-3503-0026-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/BioCAS58349.2023.10389023

***Abstract*—Edge Point of Care (POC) devices are crucial for human activity recognition (HAR) and fall detection because they enable real-time analysis and fast intervention, which can greatly improve outcomes in situations of patient and elderly monitoring. The emergence of Artificial Intelligence (AI) has sparked renewed enthusiasm for integrating AI algorithms into low-power embedded systems, broadening the potential applications of the POC devices. This paper introduces HAC-POCD, a system for multimodal human activity recognition and fall detection that processes different modalities of complementary images, designs deep neural network (DNN) models, and employs model compression techniques including knowledge distillation and low bit-width quantization with memory-aware considerations to fit models within lower memory hierarchy levels, reducing latency and enhancing energy efficiency on resource-constrained edge devices. With compact inference model of 58 KB, we achieved 95.6% accuracy on HAR case-study. Our compact inference model, deployed on resource constrained hardware, GAPuino and Raspberry Pi 4, demonstrated low latencies within milliseconds and very high energy efficiency.**

***Index Terms*—Multimodal Neural Networks, Model Compressions, Human Activity Recognition, Fall Detection, Edge POC Devices**

I. Introduction

The surge in the aging population and a renewed emphasis on fitness have catalyzed progress in Human Activity Recognition (HAR) and fall detection technologies [1], [2]. These innovations are pivotal in elder care, healthcare, sports coaching, rehabilitation, and beyond. Notably, rapid fall detection, especially for the elderly, can greatly mitigate the risk of serious injuries [3]. Deep neural networks, crucial in diverse sectors like robotics [4], [5], healthcare [6], [7], and automation [8], are pushing boundaries of what machines can perceive and achieve. Traditional HAR and fall detection integrate wearable and ambient sensors with machine learning to discern activities and identify falls. However, while advancements are noteworthy, real-world applications face hurdles, particularly in computational overhead and energy use. The promise of multimodal deep neural networks (M-DNN), which can assimilate diverse data streams, often remains untapped in edge devices [9], [10]. Potentials of multimodal deep neural networks (M-DNN), processing multiple modalities of complementary data are thus ignored for edge device deployment. Because implementing M-DNN models in resource-limited edge machine learning applications is challenging due to the growing number of model parameters and computations. Recently, ARIS [11], CoughNet-V2 [12], TinyM^{2}Net [13], OMAD [14], RhythmEdge [15] and authors in [16]–[18] implemented various optimized unimodal and multimodal neural networks on different edge devices and tiny devices.

In this paper, we tackle the challenge of implementing M-DNN models for HAR and Fall detection on various resource-constrained point-of-care (POC) hardware. To achieve energy-efficient M-DNN models on edge processing hardware, we leverage the advantages of state-of-the-art compression techniques including knowledge distillation and low bit-width quantization. We propose a hardware aware, to be specific memory aware knowledge distillation and quantization technique that reduces the model size significantly while maintaining the model accuracy based on application needs. We assess HAC-POCD with UP-Fall detection dataset to establish HAR as an edge machine learning application. HAC-POCD is subsequently implemented onto GAPuino and Raspberry Pi 4B to evaluate real-time performance on diverse resource-constrained hardware. The primary contributions of this paper are as follows:

- We propose HAC-POCD, an end-to-end system for multimodal human activity and fall detection neural networks implementable on resource constrained hardware. HAC-POCD introduces multimodal data (two different directional images) to be adapted in tinyML models to improve the application specific accuracies while maintaining required performance metrics for edge POC hardware deployments.
- We compress the M-DNN model with hardware aware knowledge distillation and uniform 8-bit quantization to reduce memory consumption and computational complexity for edge POC hardware implementation.
- We implemented our inference model on two resource-contrained hardware, GAPuino and Raspberry Pi 4B boards and explored their energy efficiency with our multimodal models.

II. Proposed HAC-POCD System

A. Multimodal DNN Model Architecture Design

The method of building Multimodal Deep Neural Network (M-DNN) models by adding different data modalities from the physical world and pre-processing them to suit the neural network formulation is shown in Figure 1 (a). The goal of

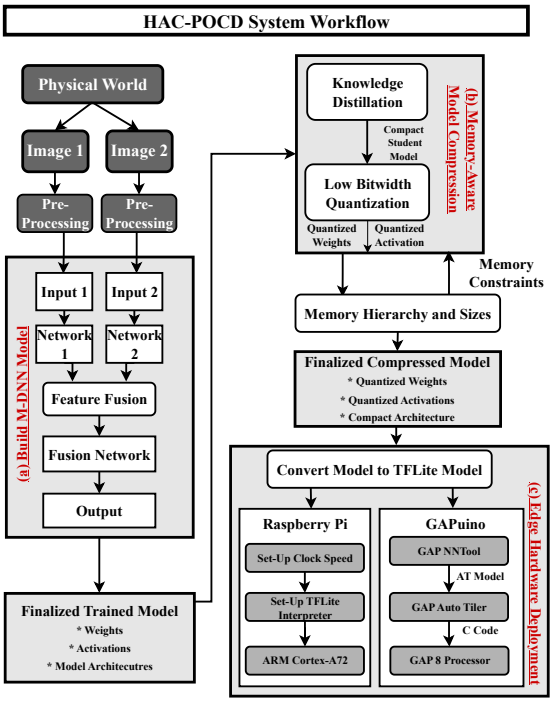

Fig. 1. The flow diagram of proposed HAC-POCD system. We consider pre-processed multimodal inputs for our proposed HAC-POCD. Proposed HAC-POCD is the sequential combination of the steps shown in the diagram.

using multimodal data is to use complimentary information from each modality for a particular learning activity, which will ultimately result in a more robust representation and better outcomes than if only one modality were used. In order to do classification tasks, we developed a multimodal learning problem that makes use of various data modalities. Due to its proven superiority to other fusion techniques, we used the *Intermediate fusion* methodology in this work to fuse several modalities [19]. We created unimodal models for each modality when creating the multimodal model. We emperically decided the hyperparameters (filter size, number of filters) for each unimodal network. To combine each modality, we selected the unimodal networks with the highest accuracy. Features are collected from each unimodal network, fused, and sent to the fusion network. The classification output is ultimately represented by the Softmax activation function as a probability distribution of the last fully connected layer.

B. Memory-Aware Model Compression

Memory-aware model compression techniques reduce deep learning models' memory footprint without sacrificing performance. Our proposed technique applies knowledge distillation and quantization to M-DNNs, where a smaller student model emulates a larger teacher model, minimizing memory needs. The aim is to fit the model onto tiny processors' on-chip memories (L1 and L2 memories), considering several factors for memory-aware compression.

- **Memory Hierarchy and Sizes:** Memory hierarchy of the target hardware platform, such as on-chip SRAM, off-chip DRAM, or Flash memory, is taken into account by memory-aware model compression in HAC-POCD to make sure that the compressed model can be successfully stored and executed within the available lower level of memory hierarchies for faster and more effective deployment.
- **Knowledge Distillation:** This approach involves training a smaller student model to mimic the behavior of a larger, more accurate teacher model. The student model should have a smaller size than the teacher model, with fewer layers, parameters, or connections, to reduce memory requirements and enable deployment on memory-constrained devices. Different distillation techniques can be utilized to transfer knowledge from the teacher to the student model, such as soft targets, attention transfer, or feature map matching. We selected soft targets to transfer the teacher knowledge into student. This is achieved by using the soft targets generated by the larger model as training labels for the smaller model. The soft targets are obtained by applying a temperature scaling factor, T, to the output probabilities of the larger model, which smooths out the peaks and makes the distribution more spread out. More formally, let us denote the output probabilities of the larger model as p_i and the soft targets as q_i. The soft targets are defined as follows:

$$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \tag{1}$$

where z_i is the logit (unnormalized log-probability) output of the larger model for class i. The temperature scaling factor, T, controls the "softness" of the targets, with higher values of T resulting in softer targets. The smaller model is then trained to minimize the Kullback-Leibler (KL) divergence between its output probabilities, p'_i, and the soft targets, q_i:

$$\mathcal{L} = \sum_i q_i \log \frac{q_i}{p'_i} \tag{2}$$

where p'_i is the output probability of the smaller model for class i. The KL divergence measures the difference between two probability distributions, and the loss function encourages the smaller model to learn a similar distribution to that of the larger model.
- **Compact Network Architecture:** Models with reduced memory footprints without sacrificing accuracy can be created with the use of efficient and compact network designs. To further compress the student model without significantly sacrificing accuracy, we implemented depth-wise separable convolution layers in place of standard convolution layers.
- **Hardware-Aware Quantization:** Model memory footprint can be reduced by lowering the bit-width of parameters and activations. Hardware-agnostic quantization methods like uniform or mixed-precision can help, but may disregard hardware capabilities and limitations, potentially yielding suboptimal performance. Contrastingly,

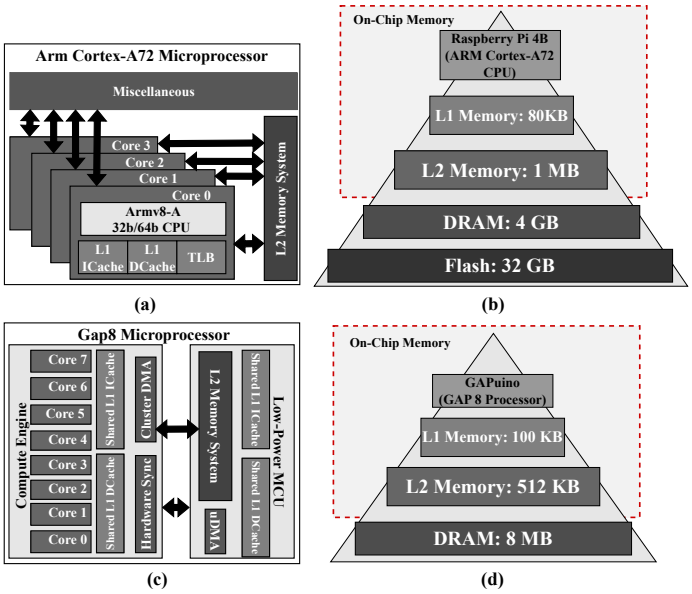

Fig. 2. (a) Hardware Architecture for GAP8 microprocessor. (b) Memory Hierarchy of GAP8. GAP 8 microprocessor has L1 Memory of 100 KB (80 KB shared in compute engine + 20 KB for low power MCU.), l2 memory of 512 KB and 8MB of DRAM (c) Hardware Architecture for Arm Cortex-A72 microprocessor used in Raspberry Pi 4B. (d) Memory Hierarchy of ARM Cortex-A72 CPU, which has L1 Memory of 80 KB (48 KB Instruction Cache + 32 KB Data Cache), L2 memory of 1 MB, DRAM of 4 GB and external flash was 32 GB

hardware-aware quantization optimizes bit-width for parameters and activations without sacrificing accuracy, considering target hardware. We employed this approach, standardizing our model to uniform 8-bits, compatible with our target hardware's int8 support. Using Tensorflow Lite's post-training, we produced a model performing solely integer arithmetic.

C. Deployment on Resource-Constrained Hardware

We used the GAPuino board, with a GAP8 [20] microprocessor, as our main edge deployment hardware. Its nona-core 32-bit RISC-V cores, neural processor, and high-performance make it suitable for long-lasting, battery-powered edge computing and IoT applications. GAP8 incorporates autonomous peripherals, an ultra-low-power micro-controller, and a compute engine. The micro-controller has a standard unit and fabric controller. It features L1 cache for the MCU core, eight additional cores sharing similar data and instruction caches in the compute engine. The entire chip shares a 512KB L2 cache. Operating at different voltage and frequency domains optimizes power use. GAP8's architecture and memory hierarchy are illustrated in figure 2(a) and (b). The GAPFlow toolchain, which consists of NNTOOL and AutoTiler, is used in this work, as shown in Figure 1(c). The DNN architecture is modified by NNTOOL, providing AutoTiler compatibility and changing weights for GAP8. AutoTiler generates C code that is GAP8 compatible while algorithmically optimizing the memory layout. Despite automation, human modifications of the maximum stack sizes or the heap capacity are occasionally required for particular DNNs. AutoTiler allocates all of the L1 and L2 memory by default, which may result in heap overflows, data corruption, and stack problems. By allocating heap memory prior to DNN initialization, the Real-Time Operating System (RTOS) of the GAP8 further exacerbates the issue by decreasing the amount of space that is accessible.

To compare the performance of the HAC-POCD system on an edge device, the Raspberry Pi 4B, which has an Arm Cortex-A72 microprocessor, is used as a secondary edge platform in this paper. Memory hierarchy and hardware architecture are shown in figure 2 (c) and (d).

III. Experimental Results and Analysis

Our study uses the UP-Fall Detection dataset [21], which comprises multimodal data from 17 healthy individuals aged 18-24 performing various activities, including five types of falls. The data was captured using wearable and ambient sensors, and vision devices, with all datasets publicly available. The subjects were recorded by two cameras, one lateral and one frontal, both placed 1.82m above the floor. Subjects performed falls from right to left, with cameras maintaining equal distance from them throughout the experiments. In this work, we use only the information from two cameras, which run at 18 fps, from the dataset, taking advantage of the multiple camera distributions. The activities performed are: Falling forward using hand, Falling forward using knees, falling backwards, falling sideward, falling sitting in empty chair, walking, standing, sitting, picking up an object, Jumping, Laying. Images are resized to 64x64 to reduce hardware memory demands, and the data is split into 70% for training, 10% for validation, and 20% for testing. HAC-POCD utilizes parallel CNN layers to process two image modalities, employing MobileNet-V2 afor both of them to extract and fuse features for multi-class classification. The teacher model, illustrated in figure 3, adopts pre-trained ImageNet weights and is trained for 200 epochs using categorical cross-entropy loss and the Adam optimizer. The teacher model achieves 97% accuracy in activity recognition.

Figure 4(a) presents HAC-POCD evaluation results. Single modality data yields lower classification accuracies compared to the uncompressed multimodal model. Incorporating multimodal image data improves accuracy to 97%, while the 8-bit quantized multimodal model achieves 96% accuracy. We then experimented to design the student model from scratch incorporating memory aware knowledge distillation. Our target was to compress the student model down to some KB so that we could fit the model on the L1 and L2 caches of the hardware we used for deployment. To this end we experimented with different filter sizes, number of neurons in dense layers and also replacing the 2nd and 3rd CNN layers to their depthwise separable counterparts so that we could achieve ultimate compression for the student model. Student Model 5 achieves around 95.6% of accuracy with only 58 KB of model size. The experimental results are shown in the figure 4(d). We selected this as our final student model for inference which achieves 89.65$\times$ reduction in model size

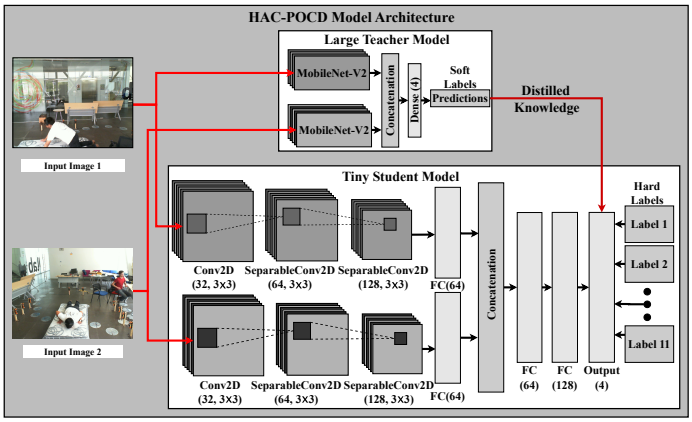

Fig. 3. The model architecture of the proposed *HAC-POCD*. Here, Conv2D = 2 dimensional CNN, SeparableConv2D = 2 dimensional depthwise-separable CNN and FC = Fully Connected Layer.

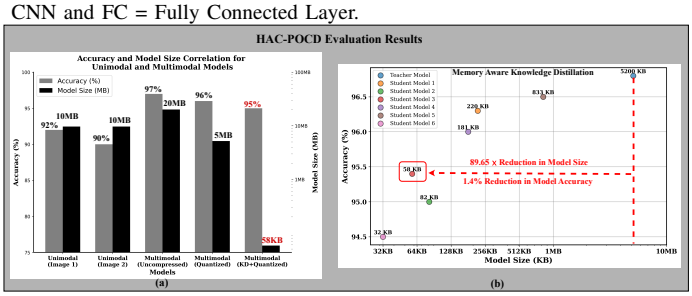

Fig. 4. (a) HAC-POCD classification results for HAC-POCD in terms of both unimodal and multimodal settings. The multimodal setting improved 7% accuracy compared to unimodal (image 2) classification setting. Model compression techniques reduce 1.4% accuracy of the multimodal setting. (d) Experiments for memory-aware knowledge distillation.

from its teacher model at the cost of 1.4% reduction in model accuracy.

IV. Device Implementation Results and Analysis

To evaluate the HAC-POCD approach, we deployed the trained models on the GAP8 processor. Table I reports resource utilization of HAC-POCD implemented on GAP8 processor. It uses 47.5 KB of L1 memory and 178 KB of L2 memory which is 44% of the available L2 memory. This inference model as well does not require off-chip DRAM to store its weights and activations which ensures the minimum latency.

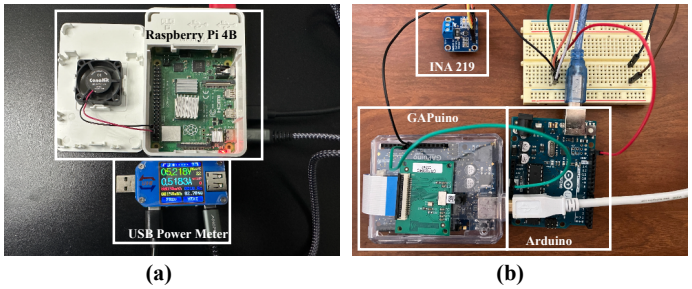

Fig. 5. (a) Raspberry Pi 4B power measurement setup. USB power measurement device was used for raspberry Pi 4B. (b) GAPuino board power measurement setup. INA219 and Arduino measure the GAP8 power consumption.

TABLE I
Resource utilization data of HAC-POCD implemented on GAP8 Processor

Resources	L1 Memory	L2 Memory	DRAM
Available for Use (KB)	52.7	400	8000
Inference Model Utilization (KB)	47.5 (90%)	178 (44%)	0

TABLE II
Implementation Results of the proposed HAC-POCD and Comparisons with State-of-the-art Works.

Architectures	This Work		[13]		[22]	
Application	Activity Recognition		COVID-19 Detection	Object Classification	Vehicle Clasifications	COVID-19 Detection
Modality Used	Image+Image		Audio+Audio	Image+Audio	Image+Audio	Audio+Audio +Audio
Operations (GOP)	2.38		-	-	0.42	0.01
Edge Devices	Raspberry Pi 4B	GAPuino	Raspberry Pi 4B		Raspberry Pi 4B	
Frequency (MHz)	1500	175	1500		1500	
Latency (ms)	9.95	49.20	1200	798	1240	980
Power (mW)	787	307.6	1700	959	1567	994
Energy (mJ)	7.83	15.13	2040	765.28	1800	974.12
Performance (GOP/s)	239.20	48.37	-	-	0.33	0.01
Energy-Efficiency (GOP/s/W)	**303.93**	**157.26**	-	-	0.22	0.01

Figure 5 (a) shows Raspberry Pi 4B power measurement setup where USB power measurement device was used and the figure 5 (b) displays the power measurement setup used in this work for GAPuino board, using INA 219 sensor and Arduino board.

Table II reports latency and power consumption of the HAC-POCD implemented on GAPuino and Raspberry Pi 4B Boards. Our Raspberry Pi implementation has energy efficiency of 303.93 GOP/s/W and our GAPuino implementation has energy efficiency 157.26 GOP/s/W. We have also compared our both the implementations with state-of-the-art multimodal models deployed on resource-constrained hardware devices. As our work targets hardware-aware model compression, both of our case-studies outperforms previous implementations with hardware-agnostic compressed models.

V. Conclusion

This paper presents HAC-POCD, a system that takes different complimentary image data, designs DNN models, employs model compression techniques including knowledge distillation and low bit-width quantization incorporating hardware-aware design considerations to compress the models to fit within lower levels of the memory hierarchy, resulting in reduced latency and enhanced energy efficiency while deployed on resource-constrained tiny devices. To assess the effectiveness of HAC-POCD, we evaluated with UP-Fall detection dataset. Our results showed that, despite the utilization of tiny inference model, we were able to attain an accuracy of 95.6% accuracy for the human activity recognition task using an inference model size of only 58 KB. Our compressed model was deployed on two edge hardware platforms, namely Raspberry Pi and GAPuino development boards, attaining latencies within the range of a few milliseconds and power consumption in the milliwatt range.

VI. Acknowledgement

This research was supported by the National Science Foundation CAREER Award under Grant No. 1652703.

REFERENCES

[1] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1192–1209, 2012.

[2] N. Noury, A. Fleury, P. Rumeau, A. K. Bourke, G. Laighin, V. Rialle, and J.-E. Lundy, "Fall detection-principles and methods," in *2007 29th annual international conference of the IEEE engineering in medicine and biology society*. IEEE, 2007, pp. 1663–1666.

[3] M. E. Tinetti and C. S. Williams, "Falls, injuries due to falls, and the risk of admission to a nursing home," *New England journal of medicine*, vol. 337, no. 18, pp. 1279–1284, 1997.

[4] M. Navardi and T. Mohsenin, "Mlae2: Metareasoning for latency-aware energy-efficient autonomous nano-drones," *UMBC Student Collection*, 2023.

[5] M. Navardi, E. Humes, and T. Mohsenin, "E2edgeai: Energy-efficient edge computing for deployment of vision-based dnns on autonomous tiny drones," in *2022 IEEE/ACM 7th Symposium on Edge Computing (SEC)*. IEEE, 2022, pp. 504–509.

[6] F. B. Emdad, S. M. Ho, B. Ravuri, and S. Hussain, "Towards a unified utilitarian ethics framework for healthcare artificial intelligence," 2023.

[7] L. Ayinde, M. P. Wibowo, B. Ravuri, and F. B. Emdad, "Chatgpt as an important tool in organizational management: A review of the literature," *Business Information Review*, p. 02663821231187991, 2023.

[8] M. S. Anwar, E. Dey, M. K. Devnath, I. Ghosh, N. Khan, J. Freeman, T. Gregory, N. Suri, K. Jayaraja, S. R. Ramamurthy *et al.*, "Heteroedge: Addressing asymmetry in heterogeneous collaborative autonomous systems," *arXiv preprint arXiv:2305.03252*, 2023.

[9] F. B. Emdad, S. Tian, E. Nandy, K. Hanna, and Z. He, "Towards interpretable multimodal predictive models for early mortality prediction of hemorrhagic stroke patients," *AMIA Summits on Translational Science Proceedings*, vol. 2023, p. 128, 2023.

[10] H.-A. Rashid *et al.*, "A deep metric for multimodal registration," in *2022 IEEE Healthcare Innovation Point-Of-Care Technologies Conference (HI-POCT)*. IEEE, 2022.

[11] P. R. Ovi *et al.*, "Aris: A real time edge computed accident risk inference system," in *2021 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2021, pp. 47–54.

[12] H.-A. Rashid, M. M. Sajadi, and T. Mohsenin, "Coughnet-v2: A scalable multimodal dnn framework for point-of-care edge devices to detect symptomatic covid-19 cough," in *2022 IEEE Healthcare Innovations and Point of Care Technologies (HI-POCT)*. IEEE, 2022, pp. 37–40.

[13] H.-A. Rashid, P. R. Ovi, A. Busart, Carl Gangopadhyay, and T. Mohsenin, "Tinym2net: A flexible system algorithm co-designed multimodal learning framework for tiny devices," *ArXiv*, 2022.

[14] E. Dey and N. Roy, "Omad: On-device mental anomaly detection for substance and non-substance users," in *2020 IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE)*, 2020, pp. 466–471.

[15] Z. Hasan, E. Dey, S. R. Ramamurthy, N. Roy, and A. Misra, "Rhythmedge: Enabling contactless heart rate estimation on the edge," in *2022 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2022, pp. 92–99.

[16] P. R. Ovi, E. Dey, N. Roy, A. Gangopadhyay, and R. F. Erbacher, "Towards developing a data security aware federated training framework in multi-modal contested environments," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications IV*, vol. 12113. SPIE, 2022, pp. 189–198.

[17] M. Navardi, P. Dixit, T. Manjunath, N. R. Waytowich, T. Mohsenin, and T. Oates, "Toward real-world implementation of deep reinforcement learning for vision-based autonomous drone navigation with mission," *UMBC Student Collection*, 2022.

[18] M. Navardi, A. Shiri, E. Humes, N. R. Waytowich, and T. Mohsenin, "An optimization framework for efficient vision-based autonomous drone navigation," in *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, 2022, pp. 304–307.

[19] S. R. Stahlschmidt, B. Ulfenborg, and J. Synnergren, "Multimodal deep learning for biomedical data fusion: a review," *Briefings in Bioinformatics*, vol. 23, no. 2, p. bbab569, 2022.

[20] E. Flamand, D. Rossi, F. Conti, I. Loi, A. Pullini, F. Rotenberg, and L. Benini, "Gap-8: A risc-v soc for ai at the edge of the iot," in *2018 IEEE 29th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, 2018, pp. 1–4.

[21] L. Martínez-Villaseñor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núñez-Martínez, and C. Peñafort-Asturiano, "Up-fall detection dataset: A multimodal approach," *Sensors*, vol. 19, no. 9, p. 1988, 2019.

[22] H.-A. Rashid *et al.*, "Tinym^{2}net-v2: A compact low power software hardware architecture for Multimodal deep neural networks," *ACM Transactions on Embedded Computing Systems*, 2023.