Dual Channel Fusion of Passive - Active Acoustic Cues for Intent Detection using Multi Head Attention Residual CNN

Harish Balasubramanian*, Unnathi Annapurna ShashiKumar*, Anushri Suresh†, Luis Edmundo Breña‡, Andreas Andreou§

*Department of Biomedical Engineering, Johns Hopkins University, Baltimore, United States
†Department of Computer Science, Johns Hopkins University, Baltimore, United States
‡Department of Engineering Management, Johns Hopkins University, Baltimore, United States
§Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, United States

***Abstract*—With the objective of improving the reliability of gesture detection, this research proposes a novel multimodal gesture recognition system that incorporates passive ultrasonic and active auditory signals. The method efficiently reduces noise and concentrates on important gesture information in the 40 kHz range by processing ultrasonic and audio spectrograms within the frequency range of 25 kHz to 50 kHz utilizing multi-head attention residual CNN. This method, which is optimized for real-time performance on edge devices like Arduino, provides a low-power solution with applications in industrial automation, hands-free control, and assistive technologies—areas where traditional approaches usually underperform in complex environments.**

***Index Terms*—Acoustic signals, CNN, 25kHz, low power, edge devices, gesture detection**

I. Introduction

Natural user interfaces are increasingly combining voice and gesture inputs to provide intuitive, hands-free control of devices. Acoustic sensing has enabled voice-command recognition through passive listening, as seen in smart speakers and voice assistants that achieve high accuracy using deep neural networks [1]. Beyond speech, passive audio can also capture incidental sounds from user actions (e.g., finger snaps or clothing friction) as implicit gestures. In parallel, active acoustic sensing techniques use sound as a probing signal—for instance, emitting an inaudible ultrasonic tone and analyzing its echoes or Doppler shifts to detect movement. Prior work has shown that standard speakers and microphones can act as a sonar to sense hand motions in mid-air, demonstrating contactless gesture recognition without cameras [2][3].

Recent advances in ultrasonic sensing have made high frequency signals a powerful modality for gesture detection [4]. Ultrasound (above 20 kHz) is inaudible to humans, allowing devices to transmit ultrasonic pulses and listen for reflections from hand or arm movements without causing disturbance. This method can reliably track gestures like waves, swipes, or taps by measuring echo time-of-flight or Doppler shifts [2][3][4]. Ultrasonic gesture sensing has been implemented in smartphones and even wearable accessories, turning ordinary devices into short-range motion sensors [3][5]. Such wearable sensing devices illustrate the trend of embedding sensors into everyday accessories—from smartwatches and rings to smart glasses—to continuously capture both voice and motion data in sync with user actions.

To interpret these multimodal signals, modern AI algorithms for multimodal learning are applied. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have a long track record in processing audio waveforms and time-series sensor data [6][7]. Building on these, transformer architectures with self-attention mechanisms achieve state-of-the-art performance in sequence modeling [8] and have been extended to multimodal fusion tasks [9]. In a multimodal command system, a deep learning model can be designed with dual input streams—one for audio and one for ultrasonic data—using attention to align and fuse information from spoken and motion cues. This allows the model to learn a unified representation of a command that encompasses how it sounds and how it appears as a gesture, and to rely more on the stronger modality under noisy conditions. Key applications would include the following:

Smart Home IoT Control: Users can control lights, appliances, and media by either speaking a command or performing a hand motion—for example, saying “ON” or making an “ON” gesture to power up a device [10].

Accessibility Assistive Technology: Multimodal input provides alternate ways for people with speech/ mobility impairments, voice/gesture as fallback options [11].

Robustness in Challenging Conditions: In noisy environments or when silence is required, the system can fall back on whichever modality is available, ensuring reliable command detection. This complementary design improves robustness, as one input mode can compensate when the other is hindered.

In summary, combining acoustic and ultrasonic modalities leverages the complementary strengths of voice and gesture inputs. During training, our system records paired voice commands and gestures—e.g., speaking “ON” while performing the “ON” motion—to teach the AI model their association. At runtime, it can recognize the command if either the audio cue or the gesture is detected, providing a robust multimodal fall-back mechanism. This fusion of wearable ultrasonic sensing with voice recognition is an innovative step towards reliable, context-flexible smart interfaces for everyday applications.

II. Methodology

Our system is a multimodal command recognition framework that interprets user actions through both voice and gesture inputs. It combines audio sensing and ultrasonic sensing to capture synchronized representations of the same command. The goal is to reliably predict the user's intended command from either modality, or both, using a unified deep learning model. We designed and implemented each component of this pipeline, from custom hardware to the neural architecture, as described below:

A. Data Collection Hardware

We developed a custom two-device hardware setup for capturing data. One device is a standard microphone for audio (voice commands), and the other is an ultrasonic sensor dedicated to gesture recognition. The ultrasonic device emits and listens to high-frequency sound to track hand or body movements. Both devices are synchronized in time, so that when a user performs a spoken command and a gesture together, the system records them concurrently. This synchronized capture hardware ensures a time-aligned multimodal dataset for training. The data is sent to a matlab script to sample the data for every 2 seconds. This creates an ample amount of data for the DL pipeline. We collected data for 5 gestures, up, down, backward, forward, power. Figure 3 - 7 depicts the signal acquired plotted on matlab.

B. DL Pipeline

The complete inference stack, summarised in Fig. 3, begins with two synchronised two-second waveforms: a 25 kHz microphone trace and a 100 kHz ultrasound echo envelope. These windows are long enough to capture at least one full gesture cycle yet short enough to satisfy the latency budget of a real-time system.

For each branch, we first derive a compact time–frequency representation. The microphone signal is segmented with a 1,024-sample Hann window that advances every 256 samples (10.24 ms), producing a complex short-time Fourier transform (STFT). Magnitude-squared spectra are then projected onto a 64-bin Mel filter-bank and log-compressed, yielding a 192×64 log-Mel spectrogram.

The ultrasound path uses the same STFT parameters but retains the linear-frequency scale and discards bins outside the 25–50 kHz micro-Doppler band, preserving fine harmonic structure. This results in a 391×513 log-power spectrogram. Both tensors are clipped to the $[-80, 20]$ dB range and streamed directly into the network without further feature engineering.

Feature extraction is performed by two weight-tied, one-dimensional convolutional pipelines operating in parallel. The microphone branch begins with a 16-filter, 3-tap Conv1D followed by three residual blocks; each block contains a Conv-ReLU-Conv-LayerNorm sequence with a skip connection. The ultrasound branch follows an analogous design but employs eight-filter convolutions and two such residual blocks. Stride-two convolutions are used where indicated to halve the temporal resolution, reducing memory while retaining spectral resolution. The resulting feature maps are concatenated along the channel dimension to form a joint latent representation.

Cross-modal interaction is learned by a two-head self-attention module. Independent linear projections generate 8-dimensional queries, keys, and values for each head; the resulting attention weights allow any time step in either modality to draw information from every other step. The head outputs are concatenated, projected back to the original channel width, and combined with the residual input; a subsequent layer normalization stabilizes the training.

A lightweight classification head converts the fused sequence to class scores. Global average pooling eliminates the temporal dimension, a 16-unit ReLU dense layer provides non-linear mixing, and a 0.1 dropout layer mitigates over-fitting. Final class probabilities are produced by a six-way soft-max.

The network is trained end-to-end in TensorFlow 2.15 with the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate 10^{-3}). Cross-entropy loss is minimized for 120 epochs with early stopping on the validation F1-score. A five-fold, subject-exclusive split ensures speaker- and actor-independent evaluation.

C. Quantisation and Edge Device Deployment

Figure 2 explains the end-to-end deployment pipeline. Quantisation-aware training reduces all kernels—including the attention projections—to 8-bit integers; the resulting TFLite model occupies only 94 kB. Although this INT-8 flat-buffer fits comfortably inside the Nano 33 BLE Sense Rev2's 1 MB flash and, with activation tiling, keeps peak SRAM below 170 kB, live inference could not be demonstrated. The TFLite-Micro interpreter stalled during operator resolution because two kernels used in our network—Layer Normalisation and the composite math inside Multi-Head Attention (batched BatchMatMul + Softmax)—are still missing from the CMSIS-NN / TFLM kernel set shipped with Arduino-CLI 2.3.0. Compilation succeeded, but runtime reported "unresolved custom op". Experiments with placeholder resolvers or replacing LayerNorm by per-channel bias layers still tripped on the ensuing BatchMatMul constraint; refactoring attention into depth-wise separable convolutions met memory limits but produced an unacceptable accuracy drop. In summary, the model's memory footprint is MCU-ready, but real-time deployment awaits upstream TFLite-Micro support—or hand-written kernels—for LayerNorm and the small-matrix matmul/soft-max pair that underpin the attention block.

D. Experimental Setup

Two different subjects (one male, one female; ages 24) performed a vocabulary of five arm–hand gestures—*power* (fist squeeze), *forward*, *backward*, *up*, and *down*—and spoke the word simultaneously. Each gesture was sustained for roughly two seconds while the participants stood 10 cm from the microphone and ultrasound module. Every subject contributed exactly 22 labelled recordings per class, giving 220 samples in total (11 × 2 s waveforms per person per class). Audio was digitized at 25 kHz and the active-sonar channel at 100 kHz, producing synchronized two-second clips of 50,000 and 200,000 samples, respectively.

The corpus was stratified with an 80/20 split: for each gesture we used 40 examples (20 per subject) for training and validation, and the remaining 4 examples (two per subject) for final testing, resulting in balanced class counts across the five gestures.

E. Audio Real-time Setup

Our second setup consists of a real-time configuration using an STM32L476 board and the X-NUCLEO-CCA02M2 expansion board, [1] which features two microphones. A beamforming algorithm was implemented to create a spatially focused audio stream and improve the sound signal quality.

For this setup, an ultrasound signal was simulated, and the model was run on an Intel computer that served as a bridge between the board and the model. Although this model did not operate on the edge, it provided a valuable experimental setup, as the running model was the quantized version.

We will briefly explain the beamforming setup and its execution on the board.

In our firmware, each of the two on-board MEMS microphones from the CCA02M2 expansion board outputs a 1-bit Pulse Density Modulation (PDM) stream. The STM32's Digital Filter for Sigma-Delta Modulators peripheral decimates this stream into 16 kHz, 16-bit Pulse Code Modulation samples. With a PDM clock frequency of $f_{\mathrm{PDM}} \approx 3.072$ MHz the DFSDM filter implements the following:

$$x_m[n] = \sum_{k=0}^{R-1} h[k]\, \mathrm{PDM}_m[nR + k],$$

$$R = \frac{f_{\mathrm{PDM}}}{f_s} \approx \frac{3072}{16} = 192$$

for microphones $m = 1, 2$.

These two PCM streams,$\{\, x_1[n],\, x_2[n]\}$ feed into a lightweight two-tap fractional-delay beamformer. Internally, we maintain a state for each microphone, defined as:

$$s_m[n] = \alpha(d)\, s_m[n-1] + \beta(d)\, x_m[n],$$

where the coefficients (α, β) are pre-computed based on the physical spacing d between the microphones. Finally, a simple difference is used to implement a cardioid response:

$$y[n] = s_1[n] - s_2[n].$$

Since $\alpha < 1$, this filter approximates the time delay

$$\tau = \frac{d}{c}$$

for an incoming plane wave, reinforcing sounds from 0° while canceling those from 180°.

By delivering a cleaner, spatially focused audio stream

$$z[n]$$

to downstream models, we enhance the signal-to-noise ratio and reduce reverberation, resulting in improved accuracy and robustness.

III. Results and Discussion

A. Model Validation Results

We evaluated the model using the validation portion of the dataset, which was collected through a synchronized multimodal setup involving both ultrasound and sound sensors. This dataset reflects real-time voice and gesture inputs intended for embedded applications. The inference model was deployed in a quantized format, with a total size of 92.3 KB, making it suitable for microcontrollers.

As shown in Figure 8, the model was able to accurately discriminate between the defined classes, achieving perfect classification (100%) in most action categories such as power, backward, up, down, and noise. However, a slight degradation was noted in the forward class, which was misclassified as noise in 25% of the validation samples. This indicates that the system occasionally confuses soft or subtle utterances of the forward command with ambient background sounds, a common challenge in embedded speech-action recognition systems.

The t-SNE visualization in Figure 9 supports these findings. There is visible separation among the clusters representing each class, but the forward and backward classes show moderate intra-class cohesion. This means that while the embeddings for samples in these classes are relatively well-grouped, they are not as tightly clustered as those of other classes. Additionally, some inter-class overlap is observed, particularly with the forward cluster, which aligns with the confusion matrix results.

We attribute these misclassifications partly to sensor noise during data capture and limitations in preprocessing, especially considering the small training dataset used for this evaluation. Nevertheless, these results demonstrate that the model can achieve high discriminative performance even under tight memory constraints and data scarcity.

B. Real-time setup results

Figure 10 illustrates the evolution of the five output probabilities per sample when we provide the model with a constant dummy ultrasound tensor while varying the real-time audio input. We cycle through ten samples for each command: Up, Down, Forward, Backward, and Power. Although we keep the ultrasound input constant—thereby theoretically eliminating any contribution from it—we still observe clear, synchronized increases and decreases in the class probabilities driven solely by the audio signal.

Overall, these results confirm that even when the ultrasound stream is constant, the audio input alone is sufficient to elicit

[1] The algorithm implementation could be found here: github.com/luisedmundo354/beam_forming

strong, temporally aligned responses in the correct output node. Occasional small dips and spikes within each ten-sample window indicate that the frozen ultrasound coefficients introduce a slight bias; however, the dominant effect remains that of the live audio signal, allowing the model to effectively discriminate among all five commands in real time.

C. Limitations

We recognize several limitations: the current reliance on a prototype ultrasonic sensor and the need for a larger, more diverse dataset to fully characterize performance. Future work could focus on integrating a commercial ultrasound module and expanding our dataset to cover more speakers and acoustic environments. We believe these advances will bring us closer to a reliable, low-power multimodal voice interface for real-world applications.

D. Conclusion

In this work, we presented a novel multimodal neural-network classifier designed for a closed set of commands and demonstrated its effectiveness both in validation and on-device testing. By fusing audio and ultrasound inputs, our architecture achieves enhanced robustness to noise and variability, and our tiny-model variants successfully run in real time on edge hardware. We also introduced a custom STM32L476 port—now open-sourced alongside our model on GitHub—that paves the way for practical deployment in embedded systems.

References

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

APPENDIX

IMPLEMENTATION

The complete codebase for our system is available at the following GitHub repository:

Link to our code:

https://github.com/anushrisuresh/multimodal_gesture_rec

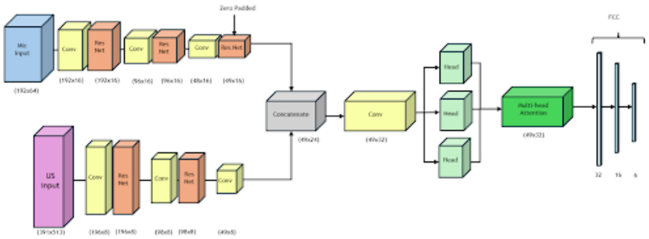

Fig. 1. DL architecture with two sensor branches, the 2-head attention module, and final classification head

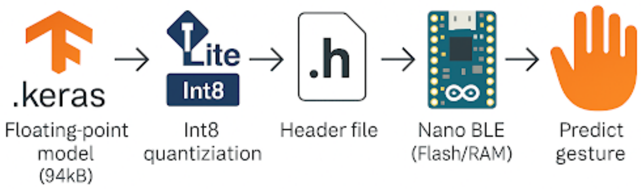

Fig. 2. End-to-end deployment pipeline

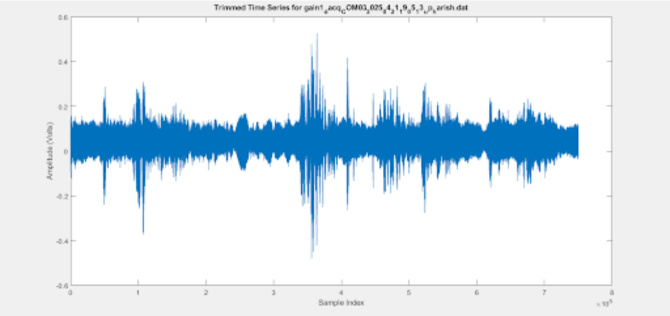

Fig. 3. Gesture Up

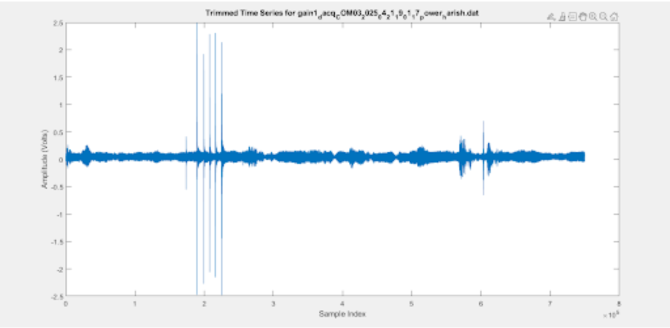

Fig. 4. Gesture Power

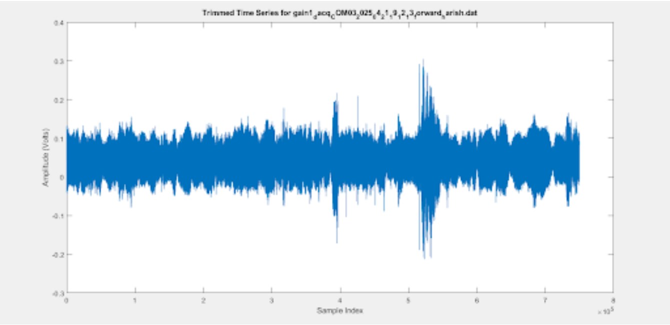

Fig. 5. Gesture Forward

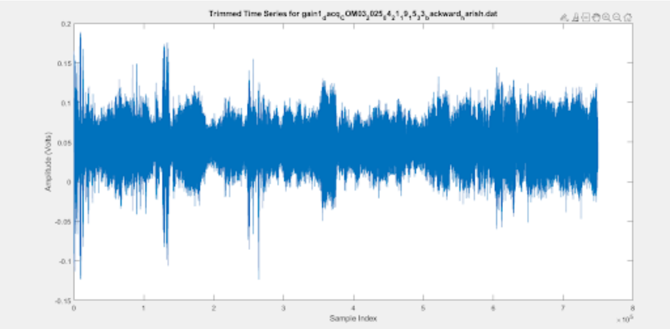

Fig. 6. Gesture Backward

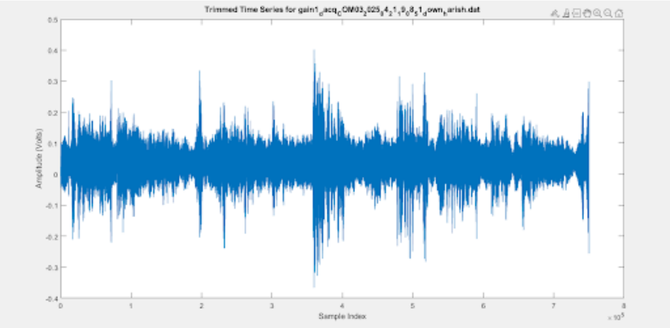

Fig. 7. Gesture Down

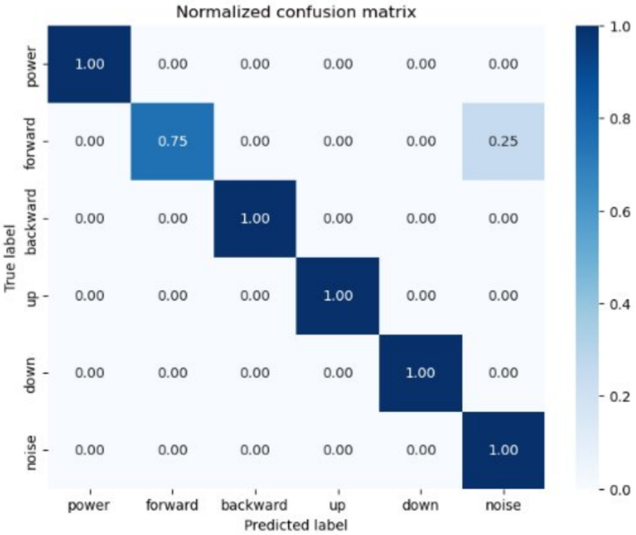

Fig. 8. Confusion Matrix

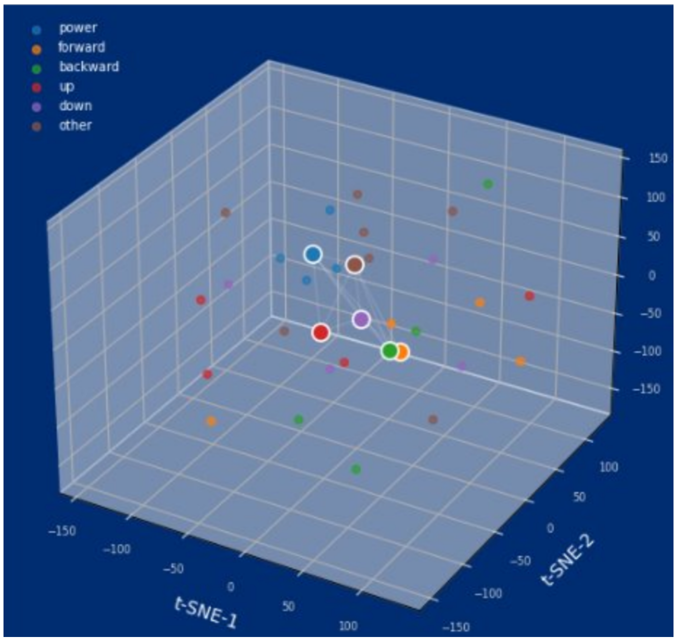

Fig. 9. 3D t-SNE Visualization of Results

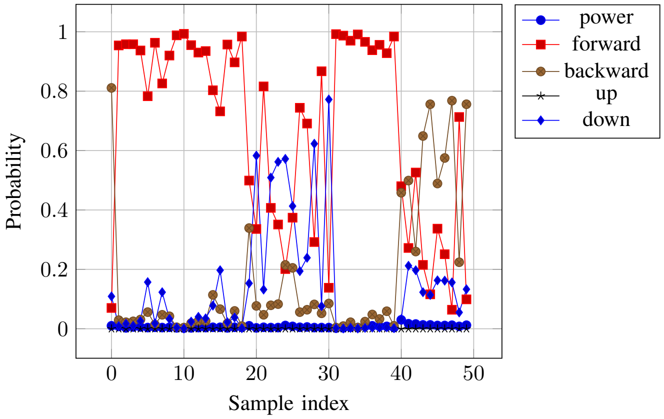

Fig. 10. Results of Audio Real-time setup from 10 samples for each class