Automatic Detection of Respiratory Symptoms Using a Low Power Multi-Input CNN Processor

Arnab Neelim Mazumder*, Haoran Ren*, Hasib-Al Rashid*, Morteza Hosseini*, Vandana Chandrareddy*, Houman Homayoun†, and Tinoosh Mohsenin*

* Department of Computer Science & Electrical Engineering, University of Maryland, Baltimore County
†University of California, Davis

***Abstract*—This paper proposes a scalable hardware involving end-to-end multi-input Deep Convolutional Neural Networks (DCNN) for early diagnosis of respiratory symptoms and disease detection. Our analysis provides evidence that detection accuracy improves up to 5% when additional information relating to the demography or physiology of the subject is appended to the deep learning framework. The proposed hardware takes windowed raw audio signals as input and processes them in tandem with the demographic information using 8 processing engines. The hardware is implemented on Artix-7 FPGA and consumes 245 mW with an energy efficiency of 4.9 GOPS/W, which is 1.5× higher than the existing works on FPGA and GPU platforms.**

***Index Terms*—Audio processing, respiratory symptoms detection, CNN, FPGA, low power hardware implementation.**

I. Introduction

Diseases originating from pulmonary infections have been one of the leading causes of death over the years. This has forced the research community to come up with innovative techniques to tackle and prevent the growth of these infectious diseases. Coughing and Dyspnea are the common symptoms that are reported among patients, and they are usually the first symptoms of most respiratory illnesses. Many of these pulmonary diseases such as Chronic Obstructive Pulmonary Disease (COPD), Asthma, and Pneumonia are contagious in nature as they infect the lungs which in turn leads to pulmonary disorders. However, the very nature of these illnesses makes it fairly easy for medical personnel to detect respiratory symptoms by just hearing the respiratory sound of a patient. Since the process of detecting respiratory symptoms is completely dependent on respiratory noises, there is a significant avenue to use low-end devices to automate the detection process thereby minimizing the direct contact between health care providers and patients.

Our aim in this work is to replicate what doctors do at triage when they first encounter a patient and allow patients to be automatically informed about their respiratory symptoms in the form of early-stage diagnosis. To this end, we are proposing to implement an end-to-end deep convolutional neural network (DCNN) on general low power processors to imitate early-stage symptom evaluation from passively recorded audio and self-inserted information about the conditions. Authors in [1] proposed a related work in this direction. However, they just mentioned a high-level overview of their software architecture and no detailed results has been published for comparison. The end-to-end systems conduct the extraction of features in combination with classification and have experienced popularity, particularly in the classification of images. These systems will extract a new discriminating feature that humans are unable to design by automatically optimizing the design of the feature extractor as connection weights of neurons. Similarly, if the sounds from the audio recording could be learned directly from the raw waveform, we would be able to derive a new feature that reflects knowledge other than the log-mel feature, and this new feature could help boost classification efficiency. Most of the audio processing works propose a method in which a CNN is applied to a log-mel feature-map or MFCC (Mel Frequency Cepstrum Coefficients) feature-map. However, our research is highly motivated by the research works presented in [2]. To reduce the extra power required for the audio preprocessing by log-mel or MFCC feature extractor hardware in low-power embedded devices, we are proposing a novel end-to-end CNN replacing the conventional log-mel-CNN or the MFCC-CNN for embedded hardware design. In terms of using low-end devices, both FPGA (Field-Programmable Gate Array) and edge devices can be suitable options. But for this specific application, the general processors in use must be able to complete the task quickly and in an energy-efficient manner. Thus, we decide to implement our networks on FPGA and demonstrate that FPGA implementation is more energy-efficient compared to CPU (Central Processing Unit)+GPU (Graphics Processing Unit) platform deployment in terms of power consumption to support our claim. To the best of our knowledge, there is no existing work for low-power hardware-based solutions for detecting respiratory symptoms. The main contributions are stated as follows

- Proposing a framework that is *scalable* and can take audio recordings from individuals along with additional information in the form of demographic details, related textual information or external numeric data of the subject and be configurable for respiratory symptoms.
- Performing input audio window size tuning, with the goal of reducing computational complexity for low power hardware implementation while meeting the accuracy requirements.
- Perform extreme bitwidth quantization of the model to improve power consumption and memory requirements.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MDAT.2021.3079318, IEEE Design and Test

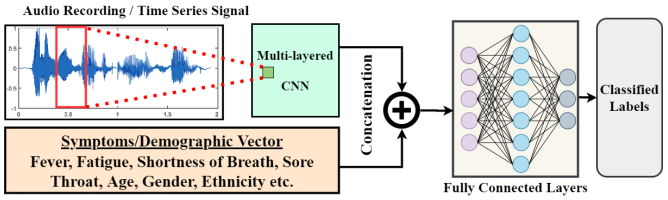

Fig. 1: The proposed architecture takes 2 types of input data including Audio, and Symptoms/Demographic Vector. The framework has a DCNN module that extracts respiratory sound features from the audio data to diagnose respiratory diseases. Then if demographic information is concatenated with the audio data, the accuracy of the classification can be boosted.

- Implementing a parameterized and scalable hardware for different numbers of processing engines (PE) that replicates the end-to-end DCNN architecture for energy-efficient deployment and comprehensive benchmarking of three different case studies on both FPGA and CPU+GPU platforms.

II. Overall Framework

The overview of the architecture used in this work is shown in Fig. 1 where the multimodality corresponds to the model being able to take multiple inputs of data. The windowing procedure involves normalizing the data first, followed by the generation of sliding windows of length T with an increment step of S through the data. For unimodal audio recordings, the windowed frames will be of shape $1 \times T$ with label selected as the label of the majority class of the samples inside the frame. As a result, a windowed frame at T_t has previous states for each data point from $(t-T+1)$ ... t where t is represented as the timestep.

The convolution layers can represent one-dimensional convolution to facilitate the exact correlation between the one-dimensional windowed audio signals. While the convolution operation helps to capture relevant feature information, striding in convolution allows the reduction of the feature map size. Once the feature map is considerably small containing necessary features, the output is flattened for further processing. Next, fully connected layers isolate pertinent information from the flattened output with inter-connections between nodes. Finally, the output is represented in the form of the probability distribution of the last fully connected layer with Softmax activation.

In addition to this, the proposed framework can also process numeric information in the shape of input vectors. Our analysis suggest that the inclusion of numeric information in terms of demography, physiology, or any other related data in parallel with the processed input signals, increases the accuracy of the framework. This additional data is concatenated with the flattened output from the convolution framework of the classification model. The novelty in our work stems from the fact that we use an end-to-end DCNN model to classify the audio signals which matches the performance of the models that use traditional digital signal processing techniques like log-mel and MFCC. The feasibility of our network to decompose the raw audio signals enhances the flexibility for embedded deployment.

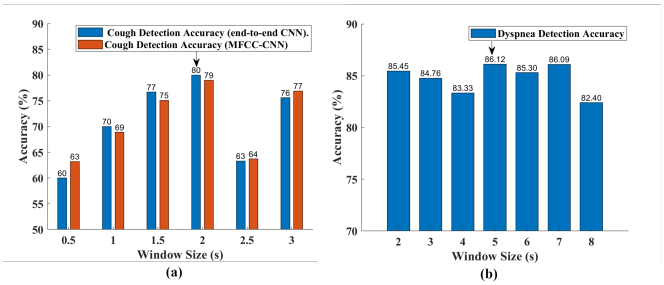

Fig. 2: Detection Accuracy with different window sizes for (a) cough detection with end-to-end CNN and MFCC-CNN, and (b) dyspnea detection architecture. Performance for the MFCC-CNN model is similar to that of the end-to-end CNN model which makes our model comparable to popular MFCC approachs for audio pre-processing.

III. Experimental Setup, Evaluation and Model Optimizations

In this section, we describe the experimental setups for three of our case studies, our evaluation procedure, and the optimization methods for energy-efficient hardware implementation.

A. *Case Study 1: Cough and Dyspnea Detection*

We use the FSDKaggle2018 dataset [3] to test the efficiency of the proposed system on cough detection. There are 41 sound classes in the FSDKaggle2018 dataset, where cough is one of the classes. The total number of audio recordings in this dataset is 11,073 arranged in uncompressed PCM 16 bit, 44.1 kHz, mono audio file. The train set contains 9.5k samples spread unequally amongst 41 classes where 3.7k samples have manually-verified ground truth annotations, and 5.8k have non-verified annotations. The non-verified annotations have a quality estimate of at least 65-70% in each category. Separately, the test set consists of 1.6k samples with manually-verified annotations and with a distribution of categories identical to that of the train set.

We compare our work with EnvNet [2], another end-to-end CNN network built for ESC-50 [4], which proposed a model with 48.6 million parameters at 185.30 MB model size. We can achieve the accuracy comparable to them whereas our architecture for cough detection is 193× smaller in size.

We first train the model with verified annotated audio recordings and then fine-tune the model with the entire train set, while relabelling audio recordings with non-verified annotations on the fly. The new label is a mix-up of the non-verified annotation and the prediction of the input audio recording from the current model where the ratio between the non-verified annotations quality and the accuracy of the current model is the same. The raw data is loaded with the default 44.1 kHz sampling rate and regularized to the range of -1 to 1. During window extraction the label of each window is kept the same as the audio recording it is extracted from. Meanwhile, if the maximum amplitude inside a window is less than a certain threshold, we consider it as silent and discard it. During testing, we evaluate audio recording level predictions by probability-voting [5] meaning for all the windows extracted from one test audio recording, we sum up the softmax outputs and predict a label for the audio recording based on the summed-up output. We consider the overall top-3 accuracy and recall score of the cough class as our metric to assess the proposed architecture on cough detection.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MDAT.2021.3079318, IEEE Design and Test

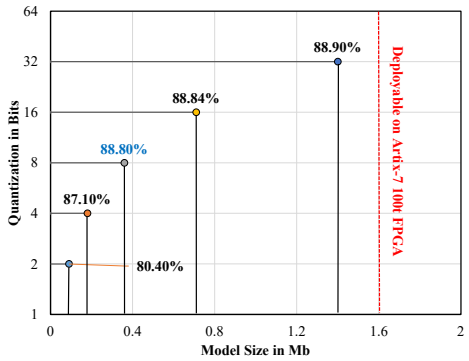

Fig. 3: The effect of quantization on the cough detection architecture is illustrated here. The blue line indicates the deployment capacity (1.6 Mb) of the Artix-7 FPGA. Our 8-bit quantized model provides the best trade-off in terms of model size and detection accuracy.

To assess the efficiency of the proposed method on dyspnea detection, we collected our own dataset. For each participant, we collected a regular 30 to 60 seconds recording of reading an article paragraph normally, and in a short-of-breath manner to capture in between gasps. The recordings come from various devices, hence it is re-sampled at a sampling rate of 44.1 kHz. After window extraction, the final dataset comprises of 3000 windows. We, therefore, guarantee that no window in the test set is overlapped with any window in the train set while dividing into train, validation, and test subsets. Since we are operating with a very small number of samples, while testing, we use window level prediction.

Our models are trained using stochastic gradient descent (SGD) with a momentum of 0.6 for 100 epochs under the categorical cross-entropy criterion using an initial learning rate of 0.001 that decays by 0.1 every 33 epochs. For silent window removal, the amplitude threshold is 0.2. We used TensorFlow [6] for implementation of the models and associated methods and Librosa [7] for audio processing.

Table I shows the model architecture for this case study with respect to window size T seconds. We have also experimented with MFCC-CNN to have a better comparison with our end-to-end CNN approach. Fig. 2 shows the accuracy results for both applications with respect to window sizes. For the cough detection application, we selected the maximum window size as 3s because cough is a single sound event. It is normally unrealistic that this sound event will last longer than 3s. However, to detect dyspnea, we selected a maximum of 8s window as we had to process a speech recording which was a continuous sound event. From Fig. 2 it is obvious that we choose a 2s window for cough detection to balance performance and parameter overhead. Also, it is apparent that the window size of 5s and 7s work best for the dyspnea detection model. However, considering the architecture defined in Table I, a higher window size will result in a higher number of computations. Moreover, the accuracy was following a decreasing trend with the growing window size for the dyspnea feature over time, and hence we decided to use a 5s window as our input for dyspnea detection application.

Moreover, we examine the feasibility of our model to be quantized to low bit-width values to further reduce the model size. We apply quantization on kernel weights, bias, and activations for all the convolution layers and fully-connected layers with the help of Qkeras library in [8], except the first and the last layer. For the first and last layer, we manually quantize the data, kernel weights, and the activation itself. To accommodate this, we maintain the range of both the data and the weights but change the values themselves to have the lowest margin of 0. Then, to map these values to required bits, we multiply them with a constant so that the highest value corresponds to the maximum range for that specific bit representation. Thus, by maintaining the range of the data and scaling them we represent all values to any bit format. With regards to the quantized ReLu activation, the function still outputs the input directly if the value is positive and produces a zero otherwise, the only change is that the output now maps to quantized bit values instead of the general 32-bit representation. According to Figure 3, our model shows acceptable performance while shrinking the model size even to 1/8 of the original 32-bit model, and for this work, we chose an 8-bit representation as it provides an appropriate trade-off for accuracy and model size.

TABLE I: Model Layer Parameters for the Two Case Studies with Respect to Window Size T seconds

Layer	Cough Detection	Dyspnea Detection
Input Layer	$44100 * T$	$44100 * T$
Conv_1D	40 x [8x1]	40 x [8x1]
Conv_1D	40 x [8x40]	40 x [8x1]
Stride	[441x1]	[441x1]
Conv_2D	32 x [16x6]	32 x [32x8]
Stride	[5,5]	[10,5]
Conv_2D	16 x [4x1]	16 x [8x4]
Stride	$[2 * T,1]$	$[T,1]$
Dense	[256]	[128]
Output	[41]	[2]
Model Size (KB)	357	198

B. Case Study 2: Detection of Respiratory Sound with Demographic Information

We used a public respiratory sound database [9] for the auditory dataset, which comprises of 920 recordings collected from 126 participants annotated with 8 forms of respiratory conditions including Upper Respiratory Tract Infection (URTI), Asthma, COPD, Lower Respiratory Tract Infection (LRTI), Bronchiectasis, Pneumonia, and Bronchiolitis. The length of each recording varied from 10 to 90 seconds, often controlled with 20-second samples. Each captured audio sample is cut into frames with a duration of 5s and a phase of 1s for data augmentation of the respiratory sound database, which implies that every two consecutive frames overlap with a duration of 4s, and for every 20s, captured sample results in 16 5s frames. Therefore, 1600 frames are generated from the total 2000 seconds of the training dataset, and 368 frames of 5s samples are generated from the total 460s testing data. The selection of the 5 frames is empirically inferred from the experience of frames varying from 1s to 10s.

For the classification of respiratory sound frames with size 1×220,500, we used a DCNN model. We used two one-dimensional convolution layers to extract specific audio characteristics. Fig. 4 shows the architecture for our proposed model where for efficient classification of the diseases, the subsequent layers involve two-dimensional convolutional layers with striding. Finally, the fully connected layers feed the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MDAT.2021.3079318, IEEE Design and Test

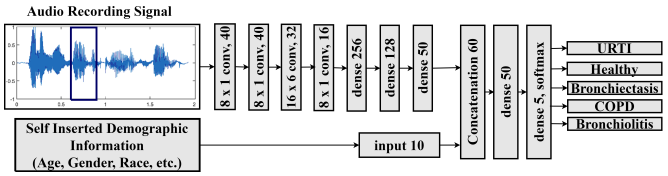

Fig. 4: The proposed framework to classify respiratory problems is based on DCNN components that process data from a user under test. Auditory information in the form of the unimodal signal captured from a medical electronic device such as a microphone or stethoscope is fed as one of the inputs. The other input is demographic information, such as age, gender, and ethnicity which can either be manually inserted or estimated by extracting relevant features in relation to the input type using a computer vision algorithm. For case study-2, the demographic information is readily available in conjunction with patient audio, and hence, we simply concatenate this additional information into the final fully-connected layers.

TABLE II: Respiratory sound classification accuracy and model complexity with and without taking the demographic information into account.

DCNN characteristics	Sensitivity (%)					Accuracy (%)
	URTI	Healthy	COPD	Bronchiec.	Bronchiol.	Test
Without Demographic Info.	21	66	96	88	4	78
With Demographic Info.	16	72	100	88	15	**83 (+5%)**

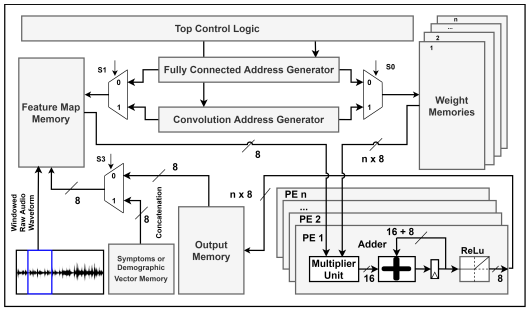

Fig. 5: Hardware framework designed for the case studies that include feature map memory and weight memory addressed by the convolution and fully connected modules to fetch data into the PE array. The feature map memory takes windowed raw audio signals as input while the weight memory cycles layer specific weights to perform MAC operations inside the PE array. The resultant data is stored into the output memory temporarily. Top control logic determines the functionality of the convolution and fully connected modules. The symptoms vector is only used for case study 2 where demographic information is fed to the model along with audio samples. This information is concatenated to the feature map memory to process the final fully connected layers of the model. The concatenation logic is controlled by the finite-state machine in the top module.

necessary feature information to the extended model to produce a generalized output that classifies 5 types of pulmonary conditions. We merged the extracted features from the audio dataset with related demographic information coming from the age groups. Table II contrasts the two sets of studies, suggesting that the COPD and stable conditions are diagnosed with higher accuracy and resulting in a total test accuracy increase by 5% when the demographic information is taken into account [10].

The study in [11] is the closest to ours in terms of building a DCNN on the same respiratory sound database. The key distinction between the two works is that our model integrates additional details alongside the audio data and proposes a framework for fixing and enhancing diagnosis accuracy by incorporating as much existing information inside the dataset as possible. The other distinction is that our chosen dataset is semi-balanced between 5 groups of respiratory sounds captured from a single diagnostic instrument that has been indistinguishably used by 61 subjects diagnosed with 7 of the 8 classes in the sample, while the dataset collection in [11] is excessively dominated with COPD recordings.

IV. Hardware Accelerator Design

To implement the proposed models of cough and dyspnea detection along with the model for classifying respiratory sounds with demographic information, the hardware accelerator must be designed with specific consideration for precise processing and functionality. This refers to fundamental design requirements such as parallel computation and efficient memory sharing. The hardware framework thus introduced here is reconfigurable to any number of filters, processing engines (PE), and layers to fit any model in order to achieve desired performance and energy efficiency specifications. The main blocks that dominate the logic flow and memory footprint in terms of computation and resources are explained below:

- **Convolution**: The convolution module performs 1D and 2D convolution depending on the software architecture requirement. It is dictated by the control unit to undertake the convolution functionality with the address generator taking care of the dynamic addressing of the convolution mechanism involving striding and corner case scenarios.
- **Fully Connected**: This module represents the functionality of the fully connected layers where all the neurons are connected to each other. This block is also directed by the control unit and an address generator to perform matrix-vector multiplication with proper addressing. The data thus generated is fetched into the PE array.
- **PE array**: This array replicates the MAC operation using a multiplier and an adder with ReLU activation function. Along with this, based on the number of PEs initialized in the parameters, this module distributes the data into different arrays to facilitate parallel processing.
- **Top**: The top module instantiates all required modules in the design. Besides, this block also maintains the logic flow and controls the data path to PE array, Convolution, and the Fully connected modules.
- **Symptoms/Demographic Vector**: This block contains the demographic/numeric information used in the respiratory sound classification model. The numeric information is stored in the form of a one-dimensional vector which is concatenated with the feature map memory after the initial model has completed processing. This concatenation operation is supervised by the control unit while the state machine manages the flow of the layers after the concatenation.

The finite-state machine (FSM) regulates the control logic and address generation for the convolution and fully connected layers. Moreover, the address generated for the layers involved is instantly sent to the on-chip Block RAM (BRAM) memory to fetch the data for computation. Each memory location can be stacked with data for a minimum width of 8 bits. The data stored in the BRAMs are forwarded to the PE array to perform multiply-accumulate (MAC) operations in a parallel execution setup. Consequently, the input data is multiplied using the 8-bit multiplier. To optimize memory footprint this hardware

TABLE III: Implementation results on FPGA, and CPU+GPU for the proposed case studies. The results for our work are obtained for 8-bit fixed point precision at a clock frequency of 80 MHz on the FPGA.

Architecture	This work (FPGA)			This work (CPU+GPU)			[12]
Application	Case Study 1		Case Study 2	Case Study 2			Image classification
	Cough	Dyspnea					
Platform	Artix-7	Artix-7	Artix-7	Nvidia TX2 GPU+CPU	Arm A57 CPU	Denver CPU	Artix-7
Model Size (Kb)	357	198	320	320	320	320	N/A
Computations (GOP)	2.4	0.6	6	6	6	6	0.00115
Fixed Point Precision	8-bit	8-bit	8-bit	8-bit	8-bit	8-bit	8-16 bit
#PE used	8	8	8	N/A	N/A	N/A	8
Frequency (MHz)	80	80	80	2035(CPU),1300(GPU)	2035	2035	100
Latency (s)	2.3	0.7	5	0.2	0.6	0.9	0.002
Total Power (mW)	244	240	245	9106	4425	3170	175
Energy (mJ)	561	96	836	1821	2655	2853	0.35
Performance (GOPS)	1	0.9	1.2	30	10	6.7	0.6
Efficiency (GOPS/W)	4.1	3.8	**4.9**	3.3	2.3	2.1	3.4

does not include any buffer or filter caches. The adder in the PE array performs the addition of the output data from the multiplier and stores it in the accumulator. This accumulated data is processed with ReLU activation logic in all the PEs and is saved in the output memory. To save resources, the logic of the PE is only implemented through a pipeline of an adder, a multiplier, and an accumulator.

This hardware is designed with output channel tiling using the rooftop model to ensure efficient parallelism of resources. As evident from Fig. 5, 8-bit packed values are read from the feature map memory but n×8 values are processed from weight memory for parallel operation where n is equal to the numbers of PEs in the array. The output from each PE is stored and concatenated until all packed values are received. This makes sure that there is no data dependency among any of the PEs as they operate separately on specific filters.

V. Hardware and GPU Implementation and Results

The software frameworks discussed previously are implemented on the Artix-7 100t FPGA at a clock frequency of 80 MHz. The RTL (Register Transistor Level) design is described in Verilog HDL and synthesized for the FPGA part using the Xilinx Vivado 2018.2 tool. The choice for the Artix-7 100t FPGA comes from the fact that the applications are directed for low power embedded implementation, making this part suitable for our goal, with 135 BRAMs as on-chip memory. The results tabulated in Table III represents the performance of the hardware for the different case studies in this work. For all configurations, the hardware is deployed for 8 PEs, thus giving a peak performance of 1.28 GOPS (2*PEs*Frequency/1000) for all MAC operations involved. The memory consumption is limited to only 60% of the total available BRAM space. Even though executed on the same platform, the case studies have different filters, number of layers, and computations. The model with the biggest overhead in terms of computation is the one that detects diseases by analyzing respiratory sounds. With 6 billion operations, the energy consumption of 836 mJ is considerable in this case. Our RTL design has varying performance depending on the computations and size of the model with energy efficiency ranging from 3.8 GOPS/W to 4.9 GOPS/W.

To establish a point of reference we also deployed our trained network (Case Study 2) on two mobile CPUs including Denver(dual-core) and Arm-Cortex A57(quad-core) for the same frequency setting along with a separate implementation on Nvidia TX2 GPU+CPU. All of these experiments were performed on the Nvidia TX2 development board that ensures precise power and latency measurements. According to the implementation, the Denver CPU has the least power consumption but takes 0.9s to classify one frame whereas the most energy-efficient implementation (CPU+GPU) has the most power dissipation but takes one-fourth the time required to classify a single frame. However, when compared with FPGA implementation, Case Study 2 deployed on Artix-7 FPGA is almost 1.5× more efficient than the best performing GPU deployment, thus proving the point that for real-time application, FPGA deployment is a better option. Also, our implementation is compared to the network deployed on [12] where a low power multimodal CNN framework is accelerated on Artix-7 FPGA and the numbers show that our framework is 1.5× more energy efficient.

VI. Conclusion

An automated detection architecture of respiratory symptoms can play a pivotal role in the early diagnosis of related respiratory problems. Hence, we proposed a scalable energy-efficient hardware framework that implements the end-to-end CNN architecture for cough, dyspnea, and respiratory disease detection with an accuracy of 80%, 87.3%, and 83% respectively. With audio samples as input data, we used precise windowing to correlate relevant features while also considering the feasibility of fitting the model on low-power embedded devices. To that extent, we introduce quantization of weights and data to 8-bit level for ensuring memory economy. Finally, our FPGA hardware mimics the software DCNN architectures while demonstrating better energy-efficiency compared to GPU+CPU implementations, thus providing an avenue to judge the possibility of using these models on low-power embedded platforms.

References

[1] A. N. Belkacem, S. Ouhbi, A. Lakas, E. Benkhelifa, and C. Chen, "End-to-end ai-based point-of-care diagnosis system for classifying respiratory illnesses and early detection of covid-19," *arXiv preprint arXiv:2006.15469*, 2020.

[2] Y. Tokozume and T. Harada, "Learning environmental sounds with end-to-end convolutional neural network," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 2721–2725.

[3] E. Fonseca, M. Plakal, F. Font, D. P. Ellis, X. Favory, J. Pons, and X. Serra, "General-purpose tagging of freesound audio with audioset labels: Task description, dataset, and baseline," *arXiv preprint arXiv:1807.09902*, 2018.

[4] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*. ACM Press, pp. 1015–1018. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2733373.2806390

[5] ——, "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2015, pp. 1–6.

[6] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[7] B. McFee, C. Raffel, D. Liang, D. P. W. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," 2015.

[8] C. N. C. J. au2, A. Kuusela, S. Li, H. Zhuang, T. Aarrestad, V. Loncar, J. Ngadiuba, M. Pierini, A. A. Pol, and S. Summers, "Automatic deep heterogeneous quantization of deep neural networks for ultra low-area, low-latency inference on the edge at particle colliders," 2020.

[9] B. M. Rocha, D. Filos, L. Mendes, G. Serbes, S. Ulukaya, Y. P. Kahya, N. Jakovljevic, T. L. Turukalo, I. M. Vogiatzis, E. Perantoni *et al.*, "An open access database for the evaluation of respiratory sound classification algorithms," *Physiological measurement*, vol. 40, no. 3, p. 035001, 2019.

[10] M.Hosseini, H.Ren, H.Rashid, A.Mazumder, B.Prakash, and T.Mohsenin, "Neural networks for pulmonary disease diagnosis using auditory and demographic information," in *epiDAMIK 2020: 3rd epiDAMIK ACM SIGKDD International Workshop on Epidemiology meets Data Mining and Knowledge Discovery*. ACM, 2020, pp. 1–5, in press.

[11] Z. Tariq, S. K. Shah, and Y. Lee, "Lung disease classification using deep convolutional neural network," in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2019, pp. 732–735.

[12] A. Jafari *et al.*, "Sensornet: A scalable and low-power deep convolutional neural network for multimodal data classification," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 1, pp. 274–287, Jan 2019.

Arnab Neelim Mazumder is pursuing his Ph.D. degree in the Computer Science and Electrical Engineering Department at University of Maryland Baltimore County, MD, USA. He received his B.S degree from CUET, Bangladesh. His research interests focus on deep neural networks, FPGA and ASIC designs, and low-power embedded systems.

Haoran Ren is a Master's student majoring in Computer Science at University of Maryland Baltimore County (UMBC). He focuses on machine learning related research in conjunction with computer vision and IoT.

Hasib-Al Rashid is a Ph.D. student in the Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, USA. He received the B.Sc. degree in Electrical and Electronic Engineering from CUET, Bangladesh. His research interests include machine learning and deep learning applications and hardware accelerators for them. He is also an IEEE student member since 2014.

Morteza Hosseini received his MSc degree in Electrical Engineering from Sharif University of Technology, Tehran, Iran, in 2012. He is currently working towards his PhD degree in Computer Engineering at University of Maryland, Baltimore County, MD, USA. His current research interests include optimizing deep neural networks for energy-efficient deployment to hardware.

Vandana Chandrareddy graduated from Vishweshwaraya Technological University, India with a Bachelor's degree in Electronics and Communication Engineering and currently. Currently, she is a Computer Engineering Master's student at University of Maryland, Baltimore County. Her focus is on building Hardware accelerators for machine learning models and implementing the same on FPGA and ASICs.

Houman Homayoun is currently an Associate Professor in the Department of Electrical and Computer Engineering at University of California, Davis. He conducts research in hardware security and trust, data-intensive computing and heterogeneous computing, where he has published more than 150 technical papers in the prestigious conferences and journals on the subject and directed over $9M in research funding from NSF, DARPA, AFRL, NIST and various industrial sponsors.

Tinoosh Mohsenin is currently an Associate Professor with the Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, where she is also the Director of the Energy Efficient High Performance Computing Lab. She has authored or co-authored over 130 peer-reviewed journal and conference publications. Her research focus is on designing energy efficient embedded processors for machine learning and signal processing, knowledge extraction techniques for autonomous systems, wearable smart health monitoring, and embedded big data computing.

Contact Information Direct questions and comments about this article to Arnab Neelim Mazumder, University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD, 21250; e-mail: arnabm1@umbc.edu