Automatic Detection of Respiratory Symptoms Using a Low Power Multimodal CNN Processor

Arnab Neelim Mazumder, Haoran Ren, Hasib-Al Rashid, Morteza Hosseini, Vandana Chandrareddy,

Houman Homayoun, and Tinoosh Mohsenin

***Abstract*—The ongoing impact concerning the COVID-19 pulmonary infection has once again highlighted the importance of machine aided diagnosis. This paper proposes a scalable hardware involving end-to-end multimodal Deep Convolutional Neural Networks (DCNN) for respiratory symptom and disease detection. Our analysis provides evidence that detection accuracy improves up to 5% when additional information relating to demography or physiology of the subject is appended to the deep learning framework. The proposed hardware is implemented on Artix-7 FPGA and consumes 245 mW with energy efficiency of 7.3 GOPS/W, which is 4.3× higher than the existing work.**

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

I. Introduction

The onset of highly contagious COVID-19 and other lower respiratory infectious diseases have generated immense strain over the health system. During this pandemic, an automated early-stage and machine assistance is critical for initial diagnosis of the disease and for evaluating its severity.

Pulmonary disorders involve a wide variety of chronic and infectious diseases and they acquire respiratory symptoms because of the important organ, lung, that they affect, whose auditory signals captured by different medical instruments are among the first to be scrutinized by a medical expert. COVID-19, for instance, develops symptoms such as dry cough, fever, weakness, dyspnea, and shortness of breath that vary in severity at various stages of disease development, and differently correlate with certain race, gender, and age groups. Over 70% of COVID-19 confirmed patients reported fever in combination with dry cough [1]. In comparison to the elderly, which is the most vulnerable category [2], clinical case reports show that the young population is less likely to experience relevant symptoms of COVID-19.

Our aim in this paper is to allow deep learning algorithms running on general low power processors to evaluate patients similar to what clinicians do in primary care and telemedicine, to use passively recorded audio and/or video and self-declared information, to bring proactive healthcare to the hands of consumers. The main contributions of this work are as follows:

- Proposing a scalable multimodal framework that can take audio recording from individuals along with demographic information of the subject and be configured for respiratory symptoms.
- Performing input audio window size tuning, with the goal of reducing computation 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.
- Implementing a parameterized and scalable hardware for different numbers of processing engines (PE) that replicate the end-to-end CNN architecture for low-power deployment.
- A comprehensive implementation and benchmarking of the proposed work with four different case studies, and comparisons with the state-of-the-art FPGA implementation results.

II. Overall Framework

The overview of the architecture used in this work is shown in Fig. 1. In this case, the input can be audio recordings, multimodal time-series signals, or numeric information related to demographic or symptoms vector. If the input is in the form of recordings or signals, they are segmented into window frames to extract features as it is pivotal to create proper windows to determine between static and continuous signals. The windowing procedure involves normalizing the independent variables first, followed by the generation of sliding windows of length T with an increment step of S through the data. If the channels are referred by M in the multimodal signals, then window images of shape $1 \times T \times M$ are created with a label assigned to each window image as the label of the current time step. As a result, a window image at location T_t has previous states for each data point from $(t - T + 1)$ ... t where t is represented as the timestep.

The window frames go into the convolution layer for relevant feature extraction. Based on the application, there can be several pairs of these layers. Moreover, the convolution layers can represent one-dimensional convolution to facilitate the exact correlation between the one-dimensional 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 from the audio or time-series samples, the output is flattened to initiate the ordering of the fully connected layers. These fully connected layers isolate pertinent information about the window frames with interconnections 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 form of input vectors in parallel to the feature extraction. Our analysis suggests that the inclusion of numeric information in terms of demography, physiology, or

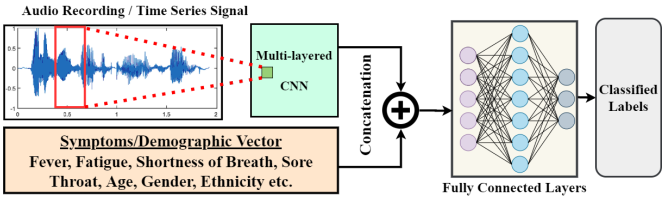
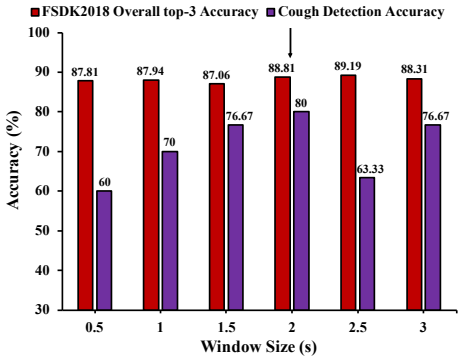
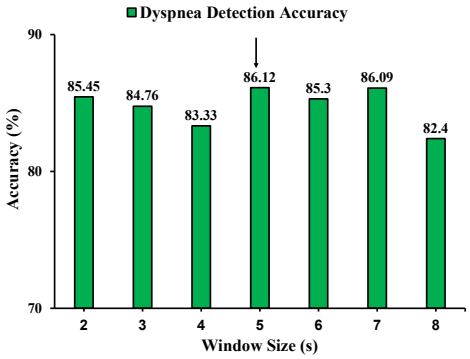


Fig. 1: The proposed architecture that takes 2 types of input data including Audio, and Symptoms/Demographic Vector. The architecture has a deep CNN module that extracts respiratory sound features from the audio data and then detects respiratory diseases. Then if demographic information is concatenated with the audio data, it can be used to classify the pre-defined respiratory disease.



(a)



(b)

Fig. 2: Detection Accuracy with different window sizes for (a) cough detection, and (b) dyspnea detection architecture. For window size experiments only, padding is applied to 2D-convolutions.

any other related data in parallel with the processed input signals increases the accuracy of the framework. This input vector containing numeric data is concatenated with the flattened output from the convolution framework of the classification model. Then, this concatenated output is processed through the fully connected layers to finalize the output label.

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 implementing into low power hardware.

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. Each recording is an uncompressed PCM 16 bit, 44.1 kHz, mono audio file. The train set contains 9.5k samples spread unequally amongst 41 classes, with a total length approximately 18 hours. Out of the 9.5k samples, 3.7k 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. 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. The overall length of the test set is approximately 2 hours.

During training, we first train the model with verified annotated audio recordings. Then, we 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. The mix-up ratio is the same as the ratio between the non-verified annotations quality and the accuracy of the current model.

We load the raw data with the default 44.1 kHz sampling rate for each audio recording and regularize it to the range of -1 to 1. Next, we apply window extraction and label each window by the same label 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 silence and discard it. Finally, we regularize each window individually to the range of -1 to 1 again before feeding it into the model.

Although the model takes window level inputs, during testing, we evaluate audio recording level predictions by probability-voting [4]. In other words, 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 metrics to assess the proposed architecture on cough detection.

To assess the efficiency of the proposed method on dyspnea detection, we collected our own dataset. For each participant, we collected a regular recording of reading an article paragraph normally, and a short-of-breath recordings while reading the same paragraph again after some strenuous exercises, with gasps captured. The recordings are recorded by various devices and then re-sampled at a sampling rate of 44.1 kHz. The lengths of the recordings vary from 30 to 60 seconds. After extracting windows without silence filtering, with different window size and window stride configurations, the final dataset comprises about 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.

Models are trained using stochastic gradient descent (SGD) with a momentum of 0.6 for 100 epochs under the categorical cross-entropy criterion. The learning rate is initially defined as 0.01, and then it is decreased according to the convergence performance. For silent window removal, the amplitude threshold is 0.2. We used TensorFlow [5] for implementation of the models and associated methods and Librosa [6] for audio processing.

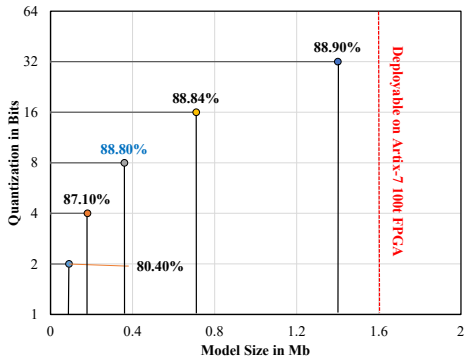


Fig. 3: The effect of quantization on the cough detection architecture is illustrated here. The red line indicates the deployment capacity (1.6 Mb) of the Artix-7 FPGA. Our 8-bit quantized model provides the best trade-off in terms of model size and detection accuracy.

Table I shows the model architecture for this case study with respect to window size T seconds. Figure 2 shows the accuracy results for both applications with respect to window size. As evident in Figure 2 (a), all the experiments show similar performances on overall top-3 accuracy metric. As for the performance on cough detection, 2s windows show good and balanced performance of extracting distinctive feature. Thus, a window size of 2s is chosen for our implementation scenario. From Fig. 2 (b) it is apparent that the window size of 5s and 7s work best for the dyspnea detection model, consider the fact of even with same model size as defined in Table I, a higher window size will increase the number of computations. We hence decide to use 5s window as our input for this application. We optimized our model here from the initial proposed model in [7].

Moreover, we examine the feasibility of our model to be quantized to low bit-width values, to further reduce the mode size. We apply quantization on kernel weights, bias, and activations for all the convolution layers and fully-connected layers, except the first and the last layer. According to Figure 3, our model shows acceptable performance while shrinking the model size even to 1/8 of the original 32-bit model. Also, derived from the trade-off between model size and performance, we choose the 8-bit quantized model while evaluating our implementation design.

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

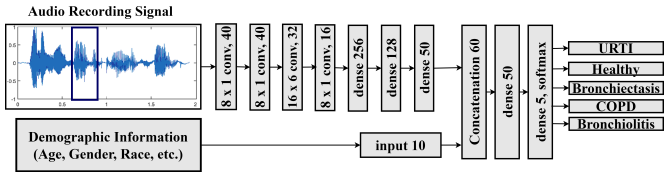


Fig. 4: The proposed framework to classify respiratory problem has two DCNN components that process data from a user under test. Auditive information, such as the audio sound captured from a medical electronic device such as a microphone or stethoscope, is part of the information, and demographic information, such as age, gender , and ethnicity, is part of that information, which can either be calculated or manually inserted using a computer vision algorithm.

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

We used a public respiratory sound database [8] for the auditory dataset, which comprises 920 recordings collected from 126 participants annotated with 8 forms of respiratory conditions including Upper Respiratory Tract Infection (URTI), Asthma, Chronic Obstructive Pulmonary Disease (COPD), Lower Respiratory Tract Infection (LRTI), Bronchiectasis, Pneumonia, and Bronchiolitis. Four types of medical instruments, including the AKG C417L Microphone, 3M Littmann Classic II SE Stethoscope, 3M Littmann 3200 Electronic Stethoscope and Welch Allyn Meditron Master Elite Electronic Stethoscope, were used to capture the recordings. 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 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. The input from the audio recordings for the DCNN model is a one-dimensional vector where the size depends on the window chosen for the system. We used two one-dimensional convolution layers to extract specific characteristics with a follow-up of non-overlapping max-pooling to downsample the feature map in order to better use the one-dimensional data. Figure 4 shows the architecture for our proposed model. For efficient classification of the diseases, the subsequent layers involve two-dimensional convolutional layers with striding. Finally, the fully connected layers feed the necessary feature information to the extended model to produce a generalized output that, as in our extracted dataset, classifies 5 types of pulmonary conditions.

We first performed a series of experiments to achieve the highest accuracy. Then, we merged the audio dataset with 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

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%)**

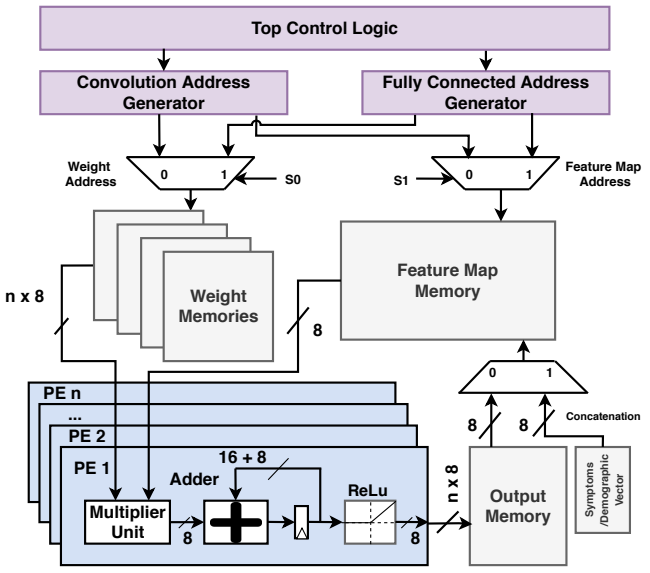


Fig. 5: Hardware framework designed for the case studies that include feature map memory and weight memory addressed by the convolution and fully connected modules to fetch data into the PE array. PE array performs MAC operations and stores the data into the output memory temporarily. Top control logic 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.

increase by 5% when the demographic information is taken into account [9].

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. This accelerator is also modeled to primarily meet the latency requirement of the intended application with a low area and utilization overhead. 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 does not include any buffer or filter caches. The adder in the PE array performs 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 the 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 Implementation and Results

The software frameworks discussed previously are implemented on the Artix-7 100t FPGA (Field Programmable Gate Array) at a clock frequency of 80 MHz. The RTL (Register Transistor Level) design is described in Verilog HDL and

TABLE III: Implementation results and comparisons of the proposed case studies with recent CNN hardware designs. The results for our work are obtained for 8-bit fixed point precision at a clock frequency of 80 MHz.

Architecture	This work			[10]	[11]	[12]
Application	Cough Detection	Dyspnea Detection	Respiratory Disease Detection With Demographic Information	Time-Series Classification	Image Classification	Image Classification
FPGA Platform	Artix-7	Artix-7	Artix-7	Artix-7	Virtex-7	Stratix - V
Input Dimension	88200 x 1	220500 x 1	220500 x 1	60 x 40 x 1	256 x 256 x 3	256 x 256 x 3
Model Size (Kb)	357	198	320	N/A	N/A	N/A
Computations (GOP)	2.4	0.6	6	0.05	0.78	1.5
Fixed Point Precision	8-bit	8-bit	8-bit	8-16 bit	8-16 bit	8-16 bit
#PE used	8	8	8	8	N/A	N/A
Frequency (MHz)	80	80	80	100	110	100
Latency (s)	2.3	0.4	3.4	0.015	0.004	0.012
BRAM (Used %)	81 (60%)	81 (60%)	81 (60%)	35 (30%)	2,715 (92%)	1,552 (61%)
Total Power (mW)	244	240	245	175	27,700	19,765
Energy (mJ)	561	96	836	0.35	111	237
Performance (GOPS)	1	1.5	1.8	0.3	213	134
Efficiency (GOPS/W)	4.1	6.3	**7.3**	1.7	7.7	6.8

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 only 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. As a result, 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 from 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 4.1 GOPS/W to 7.3 GOPS/W.

To establish a point of reference for our design in terms of processing performance, several recent hardware designs targeted for CNN acceleration are compared in Table III. [12] presents a scalable hardware framework that demonstrates the flexibility of high-level synthesis and optimization for deploying CNN architectures. In [11] implementation of a 23 layer, SqueezeNet is introduced. Along with this, a low power multimodal CNN framework is accelerated on [10] using the same Artix-7 FPGA part used in our work. The primary point of comparison between these works and our work is based on the energy efficiency of these designs and the reason that these frameworks target 8-bit fixed point precision. Since the energy efficiency metric takes into account the operational overhead with energy consumption for executing the computations involved, it best reflects the quality of an RTL design. When compared, our proposed framework is 1.1× and 4.3× more energy-efficient than the implementations of [12] and [10]. Even though the design in [11] is slightly ahead in terms of energy efficiency, our work draws significantly low power with a consumption of less than 33×.

VI. Conclusion

While the global research community is still figuring out how to tackle the novel COVID-19 pulmonary infection, an automated detection of respiratory symptoms can play a pivotal role in early diagnosis of related respiratory problems. We propose a scalable low power hardware framework that implements the end to end CNN architecture for cough, dyspnea, and respiratory disease detection with accuracy of 88.9%, 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. Furthermore, our experiments also suggest that detection accuracy improves by 5% when demographic information is fed to the deep learning architecture along with the extracted features from the audio samples. Finally, our FPGA hardware implements the software DCNN architectures and provides an avenue to judge the possibility of using these models on low power embedded platforms.

References

[1] X. Zhao, B. Zhang, P. Li, C. Ma, J. Gu, P. Hou, Z. Guo, H. Wu, and Y. Bai, "Incidence, clinical characteristics and prognostic factor of patients with covid-19: a systematic review and meta-analysis," *MedRxiv*, 2020.

[2] P.-I. Lee, Y.-L. Hu, P.-Y. Chen, Y.-C. Huang, and P.-R. Hsueh, "Are children less susceptible to covid-19?" *Journal of Microbiology, Immunology, and Infection*, 2020.

[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, "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.

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

[6] 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.

[7] A. N. Mazumder, H.-A. Rashid, and T. Mohsenin, "An Energy-Efficient low power LSTM processor for human activity monitoring," in *2020 IEEE 33rd International System-on-Chip Conference (SOCC) (SOCC 2020)*, Sep. 2020, in press.

[8] 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.

[9] 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.

[10] 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.

[11] C. Huang, S. Ni, and G. Chen, "A layer-based structured design of cnn on fpga," in *2017 IEEE 12th International Conference on ASIC (ASICON)*. IEEE, 2017, pp. 1037–1040.

[12] Y. Ma, N. Suda, Y. Cao, J.-s. Seo, and S. Vrudhula, "Scalable and modularized rtl compilation of convolutional neural networks onto fpga," in *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2016, pp. 1–8.