A Re-configurable Software-Hardware CNN Framework for Automatic Detection of Respiratory Symptoms

Hasib-Al Rashid, Haoran Ren, Arnab Neelim Mazumder, Mohammad M. Sajadi, and Tinoosh Mohsenin

Abstract Detection of respiratory symptoms has long been an area of extensive research to expedite the process of machine aided diagnosis for various respiratory conditions. This chapter attempts to address the early diagnosis of respiratory conditions using low power scalable software and hardware involving end-to-end convolutional neural networks (CNNs). We propose RespiratorNet, a scalable multimodal CNN software hardware architecture that can take audio recordings, speech information, and other sensor modalities belonging to patient demographic or symptom information as input to classify different respiratory symptoms. We analyze four different publicly available datasets and use them as case studies as part of our experiment to classify respiratory symptoms. With regards to fitting the network architecture to the hardware framework, we perform windowing, low bit-width quantization, and hyperparameter optimization on the software side. As per our analysis, detection accuracy goes up by 5% when patient demographic information is included in the network architecture. The hardware prototype is designed using Verilog HDL on Xilinx Artix-7 100t FPGA with hardware scalability extending to accommodate different numbers of processing engines for parallel processing. The proposed hardware implementation has a low power consumption of only 245 mW and achieves an energy efficiency of 7.3 GOPS/W which is 4.3 better than the state-of-the-art accelerator implementations. In addition, RespiratorNet TensorFlow model is implemented

H.-A. Rashid (✉) · H. Ren · A. N. Mazumder · T. Mohsenin
University of Maryland, Baltimore County, Baltimore, MD 21250, USA
e-mail: hrashid1@umbc.edu

H. Ren
e-mail: rhaoran1@umbc.edu

A. N. Mazumder
e-mail: arnabm1@umbc.edu

T. Mohsenin
e-mail: tinoosh@umbc.edu

M. M. Sajadi
Institute of Human Virology, School of Medicine, University of Maryland, College Park, USA
e-mail: msajadi@ihv.umaryland.edu

S. Adibi et al. (eds.), *The Science behind the COVID Pandemic and Healthcare Technology Solutions*, Springer Series on Bio- and Neurosystems 15,
https://doi.org/10.1007/978-3-031-10031-4_4

on NVIDIA Jetson TX2 SoC (CPU+GPU) and compared to TX2 single-core CPU and GPU implementations to provide scalability in terms of off-the-shelf platform implementations.

Keywords Multimodal CNN · Scalable respiratory symptoms detection · Low power embedded · Audio detection · FPGA

1 Introduction

Most of the people are not that much of conscious with breathing and respiratory health and overlook the fact that their lungs are important organs that are susceptible to infections and damages. Acute respiratory infections, as well as chronic respiratory illnesses such as asthma, chronic obstructive pulmonary disease, and lung cancer, are examples of respiratory diseases. Because the symptoms of respiratory diseases are frequently quite similar, this may lead to confusion and misinterpretation. Making a prompt and correct diagnosis is critical for the treatment of the respiratory related diseases. This may have disastrous effects if the virus spreads further, especially during pandemics like the COVID-19 pandemic. The outbreak of highly contagious COVID-19 and other respiratory infections have placed tremendous strain on the healthcare system. COVID19 causes symptoms such as dry cough, fever, fatigue, dyspnea, and shortness of breath that vary in severity at various stages of the development of the disease and correspond differently with certain races, genders, and age groups. In combination with dry cough, fever was registered by over 70% of COVID-19 confirmed patients [1]. Clinical case studies indicate that the young population is less likely to experience related symptoms of COVID-19 in contrast with the elderly, which is the most affected group [2]. However, as mentioned 5earlier, these respiratory related symptoms are not unique for only present threat COVID-19. A wide range of chronic and infectious diseases include pulmonary disorders and they develop respiratory symptoms due to the essential organ that they affect, the lung, whose auditory signals detected by various diagnostic instruments are among the first to be studied by a medical expert. As a result, establishing a diagnostic differentiator is critical for determining a fast and accurate diagnosis of respiratory symptoms and taking necessary measures.

Cough is a common sign of respiratory illnesses [3]. Cough is a common lung illness sign and a normal human defensive mechanism to protect the respiratory system Korpáš and Tomori [4]. During treatment, analyzing the cough sound may provide useful information about the coughing pathophysiological processes that lead to specific cough patterns Korpáš et al. [5]. Changes in cough sound are regarded as a crucial indication of the progression of respiratory illness and the efficacy of treatment Korpáš et al. [5]. Because coughs are often seasonal, a cough classifier or detector must have a very low false alarm rate to be regarded clinically trustworthy. Furthermore, this system must be very sensitive to variations in cough noises in order to identify any unusual occurrence [6].

Our previous works show promising results on detecting various respiratory diseases from cough sounds and respiratory sounds [7–9]. This chapter introduces RespiratorNet, a scalable and multimodal deep Convolutional Neural Network (DCNN) model running on tiny processors (e.g. tiny FPGAs and processors on cell-phones and tablets) to assess patients similar to what doctors do at triage and telemedicine, using passively recorded cough audio, speech, and self-entry information (such as age, gender and fever). The proposed software and hardware framework is scalable and can potentially have a great impact by bringing proactive healthcare to users' finger tips and to estimate the necessity of whether they need to attend clinics and have themselves further examined with the use of more specialized test-kits or facilities. This chapter is extensive extension from our previous work [8]. The main contributions of this work include:

- Propose RespiratorNet, a scalable multimodal CNN software hardware framework that can take audio recordings and speech recordings from individuals along with demographic information and other entries of the subject and be configured for classifying respiratory symptoms. RespiratorNet allows the software and hardware to quickly integrate new sensors data that are customized to various types of scenarios.
- Perform input audio window size tuning, network architecture optimization and extreme bitwidth quantization, with the goal of reducing computation complexity and memory size for low power hardware implementation while meeting the accuracy requirements.
- Design a parameterized and flexible hardware in verilog HDL for different input modalities and numbers of processing engines (PE) that replicate the RespiratorNet architecture for low power deployment.
- A comprehensive FPGA hardware implementation and benchmarking of the proposed work with different three case studies, and comparisons with the state-ofthe-art FPGA implementation results.
- Implement the TensorFlow model of RespiratorNet on embedded Nvidia Jetson TX2 board and measure its implementation characteristics for various CPU and GPU configurations.

2 Related Work

Audio based medical diagnosis has recently become an active area of research with the advancement of different machine learning and deep learning algorithms. Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) Networks have shown impressive performance in image and time-series classification tasks [10–12] as well as audio recognition tasks [9, 13, 14]. Using chest-mounted sensors, Amoh and Odame [15] used both DCNNs and recurrent neural networks (RNNs) to classify cough sound. Deep learning was used to detect sleep apnea in Nakano et al. [16]. DCNNs showed promising performance in the heart sound classification in Ryu et al. [17]. Lung sounds were classified using DCNN in Aykanat et al. [18] and RNN

[19]. Although the reported accuracy is quite high, these researches were done on unpublished data set which limit the reproducibility and further improvement of the work on this domain. The 2017 International Conference on Biomedical Health Informatics (ICBHI) [20] presented a benchmark respiratory audio data set to promote research into the classification of respiratory sound systems. Since then, researchers proposed various algorithms [21–24] using different deep learning techniques to classify respiratory cycle anomalies such as the precise locations of wheezes and crackles within the cycle of each respiratory sound recording. That dataset helped the researchers to propose a number of algorithms to identify respiratory cycling irregularities such as the exact position of the wheezes and crackle within the cycle of every sound recording in the respiratory system. Acharya and Basu [25] proposed Log quantized deep CNN-RNN based model for respiratory sound classification for memory limited wearable devices. Recently, a research group from MIT already showed Covid-19 diagnosis using cough recording with high accuracy Laguarta et al. [26]. Two different datasets [27, 28] were published to classify multiple environmental sound which include cough sounds among the other classes. Recently, a group from EPFL published one of the biggest crowd sourced cough datasets [29]. These dataset help researchers to address audio classification based health monitoring systems which is in demand now-a-days due to Covid-19 pandemic.

3 RespiratorNet Framework

The high level overview of the proposed RespiratorNet framework is presented in Figure 1. RespiratorNet can take any kind of audio recording from the user and classify accordingly. RespiratorNet can also process human speech and classify whether there is any sign of shortness of breath in the speech. Moreover, to fine tune the classification accuracy, RespiratorNet can take numeric information as input related to demographic or symptoms vectors. We evaluated RespiratorNet with human cough sounds, recorded speech, and respiratory sounds integrated with demographic information which is explained in the following section. The detailed architecture of the RespiratorNet framework is presented in Fig. 2. As the input is in the form of audio recordings, it is divided into window frames to extract features, since the right windows to distinguish between static and continuous signals are crucial. Windowing involves first standardizing the independent variables and then creating sliding T windows with S growing over the results. If the channels are referred by M in the multimodal signals, then window images of shape 1T 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 Tt has previous states for each data point from $(t\ T + 1)$ … t where t is represented as the timestep. Then the window frames are forwarded into the CNN layers for necessary feature extraction and classification.

Our CNN layers are flexible in terms of number of layers. We can decide particular number of CNN layers based on the evaluation case studies. To extract the correlation between the one-dimensional audio signals, we used one-dimensional CNN layers in

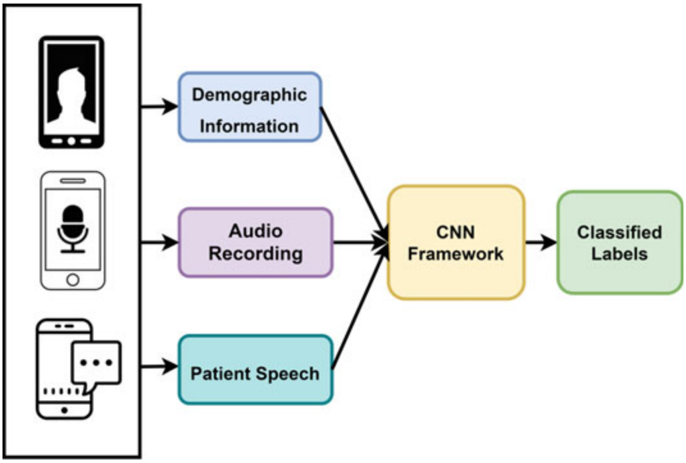

Fig. 1 The proposed multimodal *RespiratorNet* framework to classify respiratory symptoms. Some of the input information is auditory, such as the sound and frequency of coughing and speech that can detect patient's shortness of breath. Other input data can be sensed or entered manually such as demographic information. *RespiratorNet* is flexible and scalable in the sense that it allows the device to quickly integrate new sensors data that are customized to various types of scenarios, such as home appointments, hospital visits or even identification of symptoms in public settings with non-contact sensors

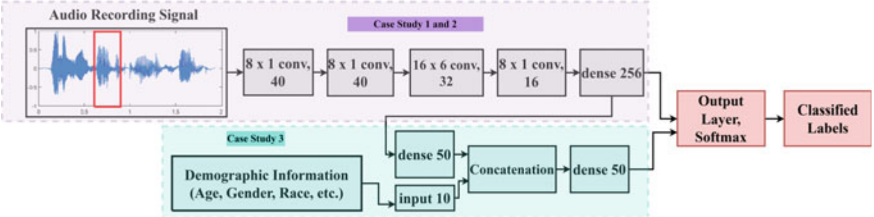

Fig. 2 The detailed architecture of the proposed flexible *RespiratorNet* in which end-to-end CNN is implemented that can be used for cough detection, dyspnea detection, and respiratory sound detection with/without the integration of other input vectors such as demographic information. The input and computation will differ according to the audio window size selected

the beginning of the model. The feature map size reduction is done by striding in the CNN layers. When we get the required small feature map size, the output is flattened and then forwarded to a number of fully connected layers to isolate sufficient window frame information with interconnections between nodes. At the end, the output is seen in the form of the probability distribution of the last fully layer with Softmax activation function.

In previous work, authors showed that if the domain specific knowledge is concatenated with the deep learning model, it improves the model accuracy. Based on this intuition, we have given flexibility to our model to process numeric information in the form of input vectors in parallel to the feature extraction. After the audio processing with the CNN layers, these input vectors containing numeric data is concatenated with the flattened output from the convolution framework of the classification model. This concatenated output is further processed through the fully connected layers to finalize the output label.

4 Experimental Results and Analysis

In this section, RespiratorNet is evaluated using three respiratory symptoms bearing case studies including *Cough detection, Dyspnea Detection and Detection of Respiratory Sound with Demographic Information* and in depth analysis and experimental results are provided.

4.1 Case Study 1: Cough Detection

We evaluated RespiratorNet for cough detection on three different datasets: ESC-50 Piczak [28], FSD Kaggle2018 [27], and Coughvid [29].

ESC-50 The ESC-50 dataset contains a total of 2,000 audio recordings of normal environmental sounds. It has 50 equally distributed classes including "coughing", so that each class has 40 audio recordings. All the audio recordings are 5 s in length, and are stored as single-channel audio waveform files at 44.1 kHz sampling rate. The dataset is originally divided into 5 folds with 400 audio recordings per fold. For each cross-validation round, we use 3 folds as train set, onefold as validate set, and onefold as test set.

Input sound duration is a key factor here to better distinguish sounds across the 50 classes. During preprocessing, we first load each audio recording with the default 44.1 kHz sampling rate and apply initial audio-wise regularization to the range of −1 to 1. Next, we crop the audio recording into windows, and discard silent windows if the window-wise maximum amplitude is less than a certain threshold. Each extract window has the same label as the audio recording which it is cropped from. At last, we apply the (−1, 1) regularization again to each window individually.

We consider a window and its label as one instance of model input. However, since the sound of an audio recording may only exist in some of the extracted windows, we evaluate the predictions at audio recording level by probability-voting Piczak [28]. Specifically, we sum up all the softmax model outputs for every window extracted from one test audio recording, and make a prediction based on the summed-up output.

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. The window stride is always 0.25 s. We used Tensor-Flow Abadi et al. [30] for implementation of the models and associated methods and Librosa [31] for audio processing.

Figure 3a shows the accuracy results for this applications with respect to window size. As evident in Fig. 3a, all the experiments show similar performances on overall accuracy metric. As for the performance on cough detection, 1 s windows show good and balanced performance of extracting distinctive feature. Thus, a window size of 1 s is chosen for our implementation scenario.

FSDKaggle2018 Similar to ESC-50, the FSDKaggle2018 dataset contains 41 sound classes and cough is one of them. There are 11,073 audio recording samples, where each of the audio recordings is an uncompressed PCM 16 bit, 44.1 kHz, mono audio file. The dataset is separated into a train set with approximately 9.5 k samples and a test set with about 1.6k samples. The audio recordings spread unequally amongst the 41 classes for both the train set and the test set, with a similar category distribution

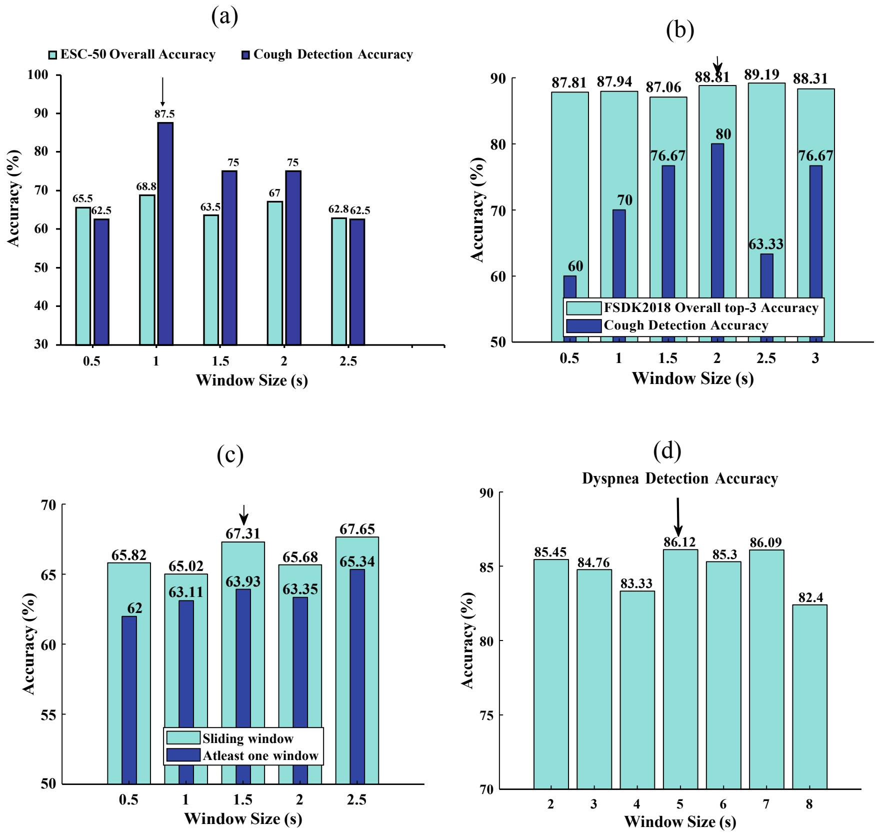

Fig. 3 Detection Accuracy with different window sizes for **a** ESC-50 cough detection, **b** FSDK2018 cough detection, **c** CoughVID cough detection and **d** Dyspnea detection. For window size experiments, padding is applied to 2D-convolutions

between them. Out of the 9.5k samples in the train set, 3.7k were listened by human participants and were annotated with ground truth label. The rest 5.8k samples have non-verified annotations with. The estimated accuracy of the non-verified annotations for each class is at least 65–70%. In contrast to the train set, the test set contains only manually-verified annotated samples.

To take fully use of both verified annotated audio recordings and non-verified annotated audio recordings, we handle them differently during training. Firstly, we train the model with verified annotated audio recordings only for initial convergence. Then, we use the entire train set to fine-tune the model. However, before each fine-tune round, we relabel the non-verified annotations. The new label is generated by mixing up the non-verified annotation and the prediction of the audio recording by the current model, with a mix-up ratio same as the ratio between the non-verified annotations quality and the test accuracy of the current model.

Same as our work on the ESC-50 dataset, we use 44.1 kHz sampling rate and same window extraction method. Meanwhile, we apply normalization and silence filtering during preprocessing and sliding window probability-voting at testing. The model hyper-parameters are also the same except training epoch number and learning rate decay. 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. Figure 3b shows the accuracy results for this applications with respect to window size. As evident in Fig. 3b, all the experiments show similar performances on overall top-3 accuracy metric. As for the performance on cough detection, 2 s windows show good and balanced performance of extracting distinctive feature. Thus, a window size of 2 s is chosen for our implementation scenario.

CoughVID CoughVID is a crowdsourced dataset for machine learning researchers aiming to find the connections between COVID-19 diagnosis and cough sound features. It provides over 20,000 cough recordings donated by participants, as well as a wide range of other subjects such as ages, genders, geographic locations, and especially, COVID-19 statuses. As a quality check, the dataset organizers include a ML based cough detection result for each audio recording as well, which is a probability of how likely the audio recording contains at least one cough sound.

As an initial step of taking fully advantages of this dataset for COVID-19 research, we evaluate our previous work on cough detection with it. In details, we use models trained on the ESC-50 dataset to predict cough existence, and compare with an assumed ground truth based on the affiliated probability. We consider two cough existence prediction schemes here. For the first one, we predict the audio recording contains cough if cough class is among the top-5 predictions of the sliding-window probability-voting results. For the second one, if at least one window gives a cough prediction among the top-5 predictions, we consider the audio recording has cough. As recommended by the dataset organizer, the assumed ground truth labels are generated by whether the affiliated cough existence probability is greater than 80% or not. Figure 3c shows the results for both schemes by different input window sizes, in accuracy of exist and non-exist binary prediction.

4.2 Case Study 2: Dyspnea Detection

We also assess the efficiency of RespiratorNet on dyspnea detection, with a dataset collected from our participants. For each participant, we record two audio recordings. One is the sound of reading an article paragraph normally, and the other one is reading the same paragraph after some strenuous exercises, so that some gasp sounds would be included. We label the two audio recordings as normal and dyspnea accordingly. The recordings are recorded by various devices and then re-sampled at a sampling rate of 44.1 kHz. Each recording has a length between 30 and 60 s. After window extraction with different configurations, we could have about 3000 windows to be divided into train, validation, and test sets, while making sure that no window in the test set is overlapped with any window in the train set.

Most of the model configurations are the same as the previous work. One difference is that we do not apply silence filtering for this case study, due to the fact that audio recordings may include gasps. The other one is that we use window-wise prediction at testing, since we are doing a binary classification on the relatively small dataset. It is obvious from Fig. 3d that the window size of 5 s and 7 s works better for the model of dyspnea detection. The number of computation would be increased by a higher window size. We therefore chose to use the 5 s window for this application.

4.3 Case Study 3: Detection of Respiratory Sound with Demographic Information

In Sects. 4.1 and 4.2, we evaluate the performance of RespiratorNet only with auditory input. In this one, we also include demographic information. The dataset [32] we use for this case study comprises 920 recordings collected from 126 participants with annotations unequally disperse among 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. The length of each recording varies from 10 to 90 sonds, often be controlled with 20 s samples.

While the majority of this dataset are COPD-diagnosed participants, by taking only audio recordings captured by Welch Allyn Meditron Master Elite Electronic Stethoscope, one of the four devices used for this dataset, we generate a random subset encompassing 63 participants. We split it into a semi-balanced train and a test set of 52 and 11 participants that include 5 types of pulmonary classes. In consequence, we eliminate Asthma, Pneumonia, and LRTI.

Each selected audio sample is cut into 5 s windows with a stride of 1 s for data augmentation. Therefore, about 1600 windows are generated from the total 2000s of the training dataset, and 368 windows are generated from the total 460 s testing data.

The selection of the 5 s window is empirically inferred from the experience varying from 1 to 10 s.

Table 1 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%)

We performed a series of experiments, from audio input only, to merging the age group information with auditory signal. Table 1 contrasts the two sets of studies, suggesting that the COPD and healthy conditions are diagnosed with higher accuracy and resulting in a total test accuracy increased by 5% when the demographic information is taken into account.

5 Hardware Architecture Design

The hardware architecture must be built with special care for accurate processing and functionality in order to incorporate RespiratorNet for the detection of cough and dyspnea along with the classification of respiratory sounds with demographic or symptoms details. This applies to basic design needs such as parallel calculation and effective memory sharing. This architecture is also modeled mainly to comply with the latency requirement with a low area and utilization overhead. In order to achieve the required performance and power efficiency requirements, the hardware architecture thus implemented here is reconfigured to any number of filters, processing engines (PEs) and layers for any model.

5.1 FPGA Design Flow and Framework

The main blocks that dominate the logic flow and memory footprint in terms of computation and resources are explained below:

The **Convolution** module performs 1D and 2D convolution depending on the software architecture requirement. The control unit defines the functionality of the convolution by using the address generator to address the convolution process dynamically, involving stride and corner case scenarios. The **Fully Connected** module represents the functionality of the fully connected layers where all the neurons are connected to each other. The block is also guided to a matrix vector multiplication with proper addressing by the control unit and an address generator. The generated data are collected in the PE array. The **PE array** uses uses a multiplier and an adder with ReLu activation feature to duplicate the MAC process. This module also spreads the data into various arrays to allow parallel processing, depending on the number of PEs initialized in the parameters. All the necessary modules have been integrated in the **Top** module. Furthermore this block also maintains a logic flow and controls the data path to PE array, Convolution and the Fully connected modules. The demographic/numerical information used in the case study 3 is provided in **Symptoms/Demographic Vector** block. The numerical information shall be stored as an one dimensional vector which after processing, is concatenated to the feature map memory. The control unit supervises this concatenation process, while the state machine controls the layer flow after the concatenation.

The finite-state machine (FSM) controls the process flow and logic for convolutional and fully connected layer operation. The address generated through the layer functionality is sent to the on-chip Block RAM (BRAM) memory instantly where each of the memory locations has a data width of 8-bits. Consequently, the input data from the feature map passes through the multiply-accumulate unit inside the PE array, and the product of the computation is saved on the output memory through ReLu activation logic. The PE logic is implemented only through a pipeline of an adder, a multiplier, and an accumulator which saves resources. The PE array ensures parallel execution of the convolution setup as evident from the Fig. 4, where 8-bit 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 continues storing these values until all values are received. As a result, the PE arrays are completely independent of each other in terms of data dependency.

5.2 Effect of Parallelism

One of the goals of this work is to introduce scalability in the hardware with regards to serial and parallel operation as per the requirement of different applications. Especially, in deep convolutional neural networks, convolutional layers dominate the computation overhead which directly affects the latency and throughput of the hardware. Hence, it is imperative to find the sweet spot for efficient parallelism existing within the convolutional layers. Among all the parallelism mechanisms studied in [33], output channel tiling provides the best throughput in FPGA fabric which performs convolution across multiple output channels for a given input channel, simultaneously. As a result, we also design the parallelism based on output channel tiling in our hardware. The outcome of the parallelism approach is illustrated in Fig. 5, in terms of the energy efficiency of our hardware accelerator under different data width precision. Our RTL (Register Transistor Level) design can achieve an energy efficiency up to 12.7 GOPS/W when implemented for 8 PEs. Similarly, the performance threshold for 2 and 4 PEs go up to 5.7 and 7.3 GOPS/W, respectively.

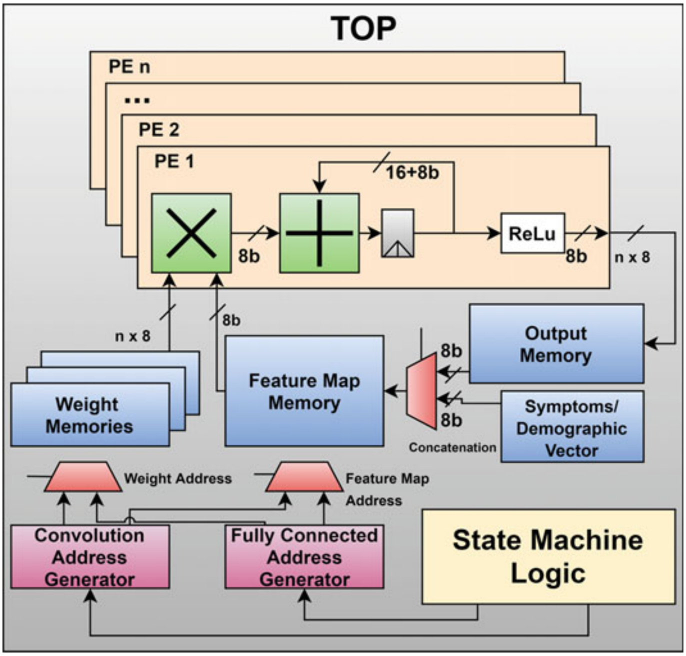

Fig. 4 RespiratorNet hardware architecture designed for the case studies that includes feature map memory and weight memory addressed by the convolution and fully connected modules to fetch data into the Processing Engine (PE) array. The PE array conducts MAC operations and temporarily saves data to the output memory. The control logic of the top defines the functionality of the convolution and fully connected modules. The symptoms vector are only used in Case Study 3 where demographic information and audio samples are supplied to the model. This data is concatenated to the feature map memory to process the finishing fully connected layers of the model. In the top module, finite-state machine manages the concatenation logic

5.3 Quantization: Fixed Point Precision Analysis

All the case studies explored in this work use the quantization level of 8 bits. Going below this level does not provide an appropriate trade-off in terms of hardware performance and model accuracy which is clearly visible from Fig. 6. In the software side, the quantization is applied on kernel weights, bias, and activations for all the convolution layers and fully-connected layers, other than the first layer and the last layer. According to Fig. 6, our model shows acceptable performance while shrinking the model size even to 1/8 of the original 32-bit model. Thus, our proposed hardware architecture has been implemented using a data width of 8-bit fixed-point precision for all four case studies. Even though the change of the data width does not amount to any variation in functional behavior, it affects the operating frequency and power consumption which in turn alters the energy consumption of the hardware. So, it is pivotal to figure out an operating frequency that is consistent with different data width precisions to properly analyze the effect of changing bits over the on-chip

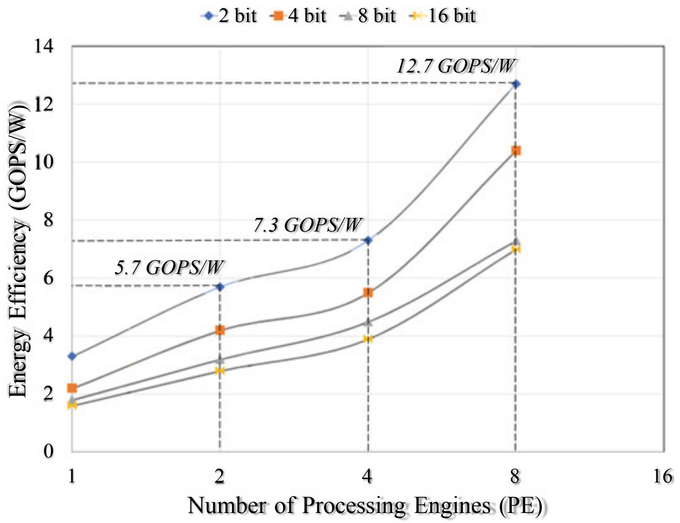

Fig. 5 A scatter plot illustrating the performance against different PE configurations and bit precisions for our proposed hardware. Depending on the input and process flow our hardware is scalable up to 12.7 GOPS/W for 8 PEs in terms of FPGA implementation at 80 MHz clock frequency

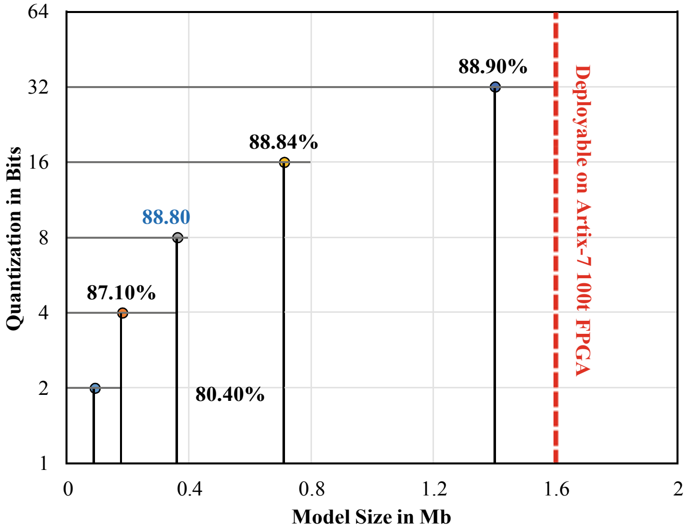

Fig. 6 The effect of quantization on the cough detection architecture is illustrated here. The red line indicates the deployment capacity (1.6 Mb) of the Artix-7 FPGA. Our 8-bit quantized model provides the best trade-off in terms of model size and detection accuracy

energy consumption. In this case, our hardware framework runs at a frequency of 80 MHz to investigate the variation in energy consumption ranging from 16-bit down to 8-bit fixed-point precision as shown in Fig. 7a. As evident in the plot, an 8-bit

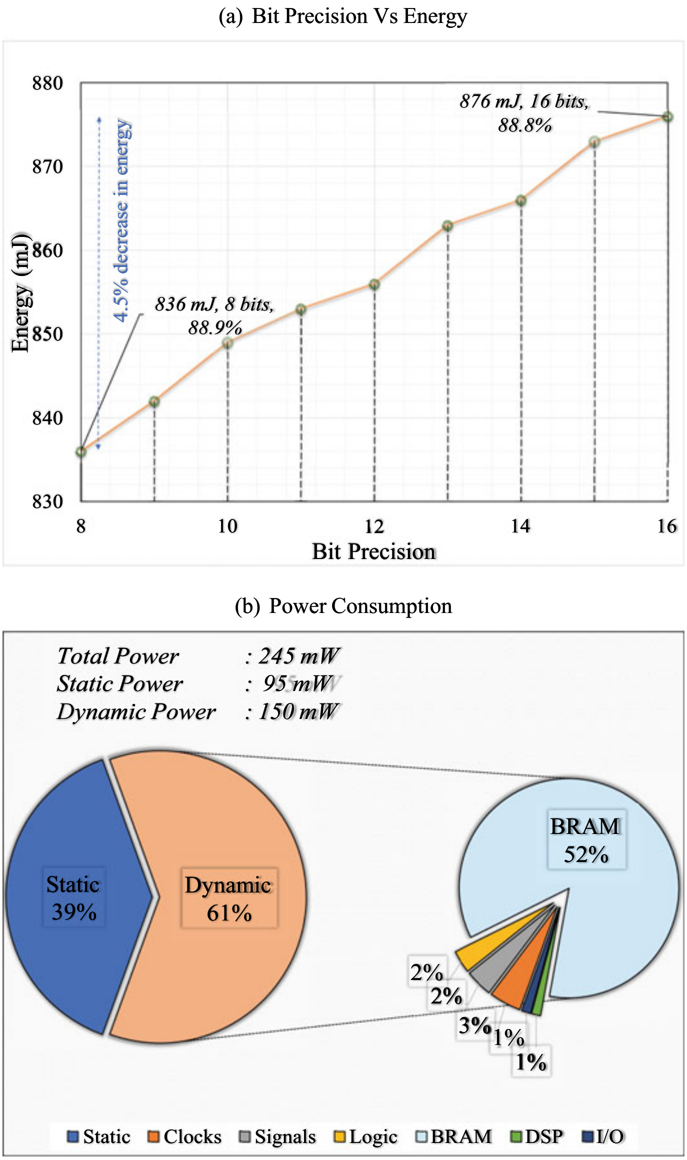

Fig. 7 **a** Illustration of the trend for FPGA energy consumption against different fixed point precision on the respiratory sounds dataset network and **b** breakdown for power consumption in the proposed hardware for a setting of 8 PEs running at 80 MHz

implementation over its 16-bit counterpart results in an energy saving of 4.5% without much deviation in the model classification accuracy for the respiratory sounds dataset network. The proposed hardware utilizes 8 PEs to implement all the different model configurations explored in this work. With a configuration setting of 8 PEs and 8-bit fixed-point precision, most of the on-chip dynamic power is dedicated to BRAMs with only a fraction of the total dynamic power being utilized in clocks, signals, logic, and other areas as highlighted in Fig. 7b. Also, per our analysis, as the number of processing engines increases, the power consumption of the BRAMs and DSPs increases to accommodate the parallel processing of the framework.

6 Hardware Implementation and Results

6.1 *FPGA Implementation*

On the Artix-7 100t FPGA (Field Programmable Gate Array), the previously mentioned software frameworks are implemented at a clock frequency of 80 MHz. The design of the RTL (Register Transistor Level) is defined in Verilog HDL and synthesized using the Xilinx Vivado 2018.2 tool for the FPGA portion. The option for the Artix-7 100t FPGA comes from the fact that the applications are targeted for embedded implementation of low power, making this component ideal for our objective, with only 135 BRAMs as onchip memory. The results tabulated in the 2 table represent the output of the hardware in this work for the various case studies. In terms of computation, the model with the highest overhead is the one that detects diseases from respiratory sound analysis. The energy consumption of 836 mJ is considerable in this case, with 6 billion operations. Depending on the calculations and size of the model, our RTL design has different results, with energy efficiency varying from 4.1 GOPS/W to 7.3 GOPS/W.

The Table 2 compares various recent hardware designs aimed at CNN acceleration. Ma et al. [34] offers a scalable hardware platform that demonstrates the versatility to deploy CNN architectures in high-level synthesis and optimization. In Huang et al. [35] implementation of a 23 layer, SqueezeNet is introduced. In addition to this on Jafari et al. [10], a low-power multimodal CNN system is accelerated using the same Artix-7 FPGA component used in our work. Our proposed system, when compared, is and 4.3 more energy efficient than the [10, 34] implementations. Although the design is marginally ahead in terms of energy efficiency in Huang et al. [35], with a consumption of less than 33, our work draws significantly low power.

Table 2 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		[5]	[10]	[31]
Application	Cough detection (FSDKaggle2018)	Dyspnea detection	Respiratory sounds with demographic info	Cough detection (CoughVID)	Time-series	Image	Image
FPGA platform	Artix-7	Artix-7	Artix-7	Artix-7	Artix-7	Virtex-7	Stratix - V
Input dimension	88,200 × 1	220,500 × 1	220,500 × 1	66,150 × 1	60 × 40 × 1	256 × 256 × 3	256 × 256 × 3
Model size (Kb)	357	198	320	359	N/A	N/A	N/A
Computations (GOP)	2.4	0.6	6	1.8	0.05	0.78	1.5
Fixed point precision	8-bit	8-bit	8-bit	8-bit	8–16 bit	8–16 bit	8–16 bit
#PE used	8	8	8	8	8	N/A	N/A
Frequency (MHz)	80	80	80	80	100	110	100
Latency (s)	2.3	0.4	3.41	2	0.015	0.004	0.012
BRAM (Used %)	81 (60%)	81 (60%)	81 (60%)	81 (60%)	35 (30%)	2715 (92%)	1552 (61%)
Total power (mW)	244	240	245	244	175	27,700	19,765
Energy (mJ)	561	96	836	488	0.35	110.8	237.2
Performance (GOPS)	1	1.5	1.8	0.9	0.3	213.7	134.4

(continued)

Table 2 (continued)

Architecture		This	Work		[5]	[10]	[31]
Efficiency (GOPS/W)	4.1	6.3	7.3	3.7	1.7	7.7	6.8

Table 3 Deploying the RespiratorNet model to commercial off-the-shelf devices including a dual-core Denver CPU, a quad-core ARM A57 CPU, and a combination of ARM CPU+Pascal GPU from the NVIDIA TX2 board

Configuration	CPU Freq (MHz)	GPU Freq (MHz)	Power (mW)	Latency (s)	Performance (GFLOP/s)	Energy (J)	Energy efficiency (GFLOP/s/W)
Denver CPU	345	–	881	10.0	0.019	8.81	0.021
	2035	–	3170	0.9	0.215	2.85	0.068
ARM A57 CPU	345	–	1168	3.7	0.052	4.32	0.045
	2035	–	4425	0.6	0.322	2.66	0.073
TX2 CPU+GPU	2035	1300.5	9106	0.1	1.935	0.91	0.210

6.2 NVIDIA Jetson TX2 Implementation

The trained TensorFlow model of RespiratorNet was implemented on embedded NVIDIA Jetson TX2 platform for evaluating the energy-latency trade-off. Trading off between the computation complexity and the classification accuracy, trained ML models can be deployed to tiny processors and edge devices (e.g. tiny FPGAs, a cell-phone, tablet). At least two hardware-level characteristics are attributed to all DCNN models: the model size and the number of operations per inference, all of which are upper-bounded by the platform resources to which they are deployed or by the inference deadline. Both the hardware resource constraints and the diagnostic latency should follow the application objectives while bringing all the components of the system together. After setting the batch-size to 1, two mobile CPUs like Denver (dual-core) and ARM-Cortex A57 (quadcore) as well as an embedded CPU+GPU implementation with different frequency settings are deployed on the trained model of RespiratorNet. The TX2 development board has been used to calculate all of the parameters as it provides precise on-board power measurement. Table 3 summarizes the implementation. From the Table 3 it can be seen that Denver CPU with a low frequency setting dissipates the least power and takes 10 s to classify one frame. However, the most energy efficient implementation, ARM CPU+GPU, dissipates approximately 10 more power compared to Denver to classify the same frame in 0.1 s. For both the cases, we provided a 5 s frame of recording to the memory.

7 Conclusion

In this chapter, to identify various respiratory symptoms, we propose RespiratorNet, a scalable multimodal CNN software hardware architecture that can take audio recordings, speech information, and other sensor modalities from patient demographic or symptom information. We evaluate and use four distinct publicly accessible databases as case studies to identify respiratory symptoms as part of our experiment. The hardware prototype for RespiratorNet is also scalable and flexible to accommodate different input modalities, data width bit precisions and parallel processing engine numbers. The proposed implementation of hardware has a low power consumption of o 245 mW and achieves an energy efficiency of 7.3 GOPS/W that is 4.3 *times* higher than the implementations of state-of-the-art accelerators. Furthermore the RespiratorNet TensorFlow model is implemented on the NVIDIA Jetson TX2 SoC (CPU+GPU) to provide scalability in terms of off-the-shelf platform implementations and is compared to TX2 single-core CPU and GPU implementations.

Acknowledgements We acknowledge the support of the University of Maryland, Baltimore, Institute for Clinical Translational Research (ICTR) and the National Center for Advancing Translational Sciences (NCATS) Clinical Translational Science Award (CTSA) grant number UL1TR003098.

References

1. Zhao, X., Zhang, B., Li, P., Ma, C., Gu, J., Hou, P., Guo, Z., Wu, H., Bai, Y.: Incidence, clinical characteristics and prognostic factor of patients with COVID-19: a systematic review and meta-analysis. MedRxiv (2020)
2. Lee, P.-I., Hu, Y.-L., Chen, P.-Y., Huang, Y.-C., Hsueh, P.-R.: Are children less susceptible to COVID-19? J. Microbiol. Immunol. Infect. (2020)
3. Cho, S.-H., Lin, H.-C., Ghoshal, A.G., Muttalif, A.R.B.A., Thanaviratananich, S., Bagga, S., Faruqi, R., Sajjan, S., Brnabic, A.J.M., Dehle, F.C., et al.: Respiratory disease in the Asia-Pacific region: cough as a key symptom. In: Allergy & Asthma Proceedings, vol. 37
4. Korpáš, J., Tomori, Z.: Cough and Other Respiratory Reflexes./Kasˇel' a Ine´ Respiracˇne´ Reflexy. Veda (1979)
5. Korpáš, J., Sadlonˇova´, J., Vrabec, M.: Analysis of the cough sound: an overview. Pulmonary Pharmacol. **9**(5–6), 261–268 (1996)
6. Amoh, J., Odame, K.: DeepCough: a deep convolutional neural net-work in a wearable cough detection system. In: 2015 IEEE Biomedical Circuits and Systems Conference (BioCAS). IEEE, pp. 1–4 (2015)
7. Ren, H., et al.: End-to-end scalable and low power multi-modal CNN for respiratory-related symptoms detection. In: 2020 IEEE 33rd International System- on-Chip Conference (SOCC) (SOCC 2020)
8. Mazumder, A.N., Ren, H., Rashid, H.-A., Hosseini, M., Chandrareddy, V., Homayoun, H., Mohsenin, T.: Automatic detection of respiratory symptoms using a low power multi-input CNN processor. IEEE Des. Test **2021**, 1–1 (2021). https://doi.org/10.1109/MDAT.2021.3079318
9. Hosseini, M., Ren, H., Rashid, H., Mazumder, A., Prakash, B., Mohsenin, T.: 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, pp. 1–5, in press
10. Jafari, A., et al.: SensorNet: a scalable and low-power deep convolutional neural network for multimodal data classification. IEEE Trans. Circ. Syst. I Reg. Papers **66**(1), 274–287 (2019). https://doi.org/10.1109/TCSI.2018.2848647
11. Rashid, H.-A., Manjunath, N.K., Paneliya, H., Hosseini, M., Mohsenin, T.: A low-power LSTM processor for multi-channel brain EEG artifact detection. In: 2020 21th International Symposium on Quality Electronic Design (ISQED). IEEE (2020)

12. Shea, C., Page, A., Mohsenin, T.: SCALENet: a scalable low power accelerator for real-time embedded deep neural networks. In: ACM Proceedings of the 28th Edition of the Great Lakes Symposium on VLSI (GLSVLSI). ACM (2018)
13. Abdel-Hamid, O., Mohamed, A.-R., Jiang, H., Deng, L., Penn, G., Yu, D.: Convolutional neural networks for speech recognition. IEEE/ACM Trans. Audio Speech Lang. Process. **22**(10), 1533–1545 (2014)
14. Piczak, K.J.: Environmental sound classification with convolutional neural networks. In: 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP). IEEE, pp. 1–6 (2015)
15. Amoh, J., Odame, K.: Deep neural networks for identifying cough sounds. IEEE Trans. Biomed. Circ. Syst. 10(5), 1003–1011 (2016)
16. Nakano, H., Furukawa, T., Tanigawa, T.: Tracheal sound analysis using a deep neural network to detect sleep apnea. J. Clin. Sleep Med. **15**(8), 1125–1133 (2019)
17. Ryu, H., Park, J., Shin, H.: Classification of heart sound recordings using convolution neural network. In: 2016 Computing in Cardiology Conference (CinC). IEEE, pp. 1153–1156 (2016)
18. Aykanat, M., Kurt, O.K.B., Saryal, S.: Classification of lung sounds using convolutional neural networks. EURASIP J. Image Video Process. **1**, 65 (2017)
19. Perna, D., Tagarelli, A.: Deep auscultation: predicting respiratory anomalies and diseases via recurrent neural networks. In: 2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS). IEEE, pp. 50–55 (2019)
20. Rocha, B.M., Filos, D., Mendes, L., Vogiatzis, I., Perantoni, E., Kaimakamis, E., Natsiavas, P., Oliveira, A., Ja´come, C., Marques, A., et al.: A respiratory sound database for the development of automated classification. In: International Conference on Biomedical and Health Informatics. Springer, pp. 33–37 (2017)
21. Liu, R., Cai, S., Zhang, K., Hu, N.: Detection of adventitious respiratory sounds based on convolutional neural network. In: 2019 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS). IEEE, pp. 298–303 (2019)
22. Perna, D.: Convolutional neural networks learning from respiratory data. In: 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, pp. 2109–2113 (2018)
23. Pham, L., McLoughlin, I., Phan, H., Tran, M., Nguyen, T., Palaniappan, R.: Robust Deep Learning Framework For Predicting Respiratory Anomalies and Diseases (2020). arXiv:2002.03894
24. Demir, F., Sengur, A., Bajaj, V.: Convolutional neural networks based efficient approach for classification of lung diseases. Health Inf. Sci. Syst. **8**(1), 4 (2020)
25. Acharya, J., Basu, A.: Deep neural network for respiratory sound classification in wearable devices enabled by patient specific model tuning. IEEE Trans. Biomed. Circ. Syst. **14**(3), 535–544 (2020)
26. Laguarta, J., Hueto, F., Subirana, B.: COVID-19 artificial intelligence diagnosis using only cough recordings. IEEE Open J. Eng. Med. Biol. (2020)
27. Fonseca, E., Plakal, M., Font, F., Ellis, D.P.W., Favory, X., Pons, J., Serra, X.: General-purpose tagging of freesound au- dio with audioset labels: Task description, dataset, and baseline (2018). arXiv:1807.09902
28. Piczak, K.J.: ESC: Dataset for environmental sound classification. In: Proceedings of the 23rd ACM international conference on Multimedia, pp. 1015–1018 (2015)
29. Orlandic, L., Teijeiro, T., Atienza, D.: The COUGHVID crowd- sourcing dataset: A corpus for the study of large-scale cough analysis algorithms (2020). arXiv:2009.11644
30. Abadi, M. et al.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems (2015). http://www.tensorflow.org/
31. McFee, B., Raffel, C., Liang, D., Ellis, D.P.W., Matt McVicar, Battenberg, E., Nieto, O.: librosa: Audio and Music Signal Analysis in Python (2015)
32. Rocha, B.M., Filos, D., Mendes, L., Serbes, G., Ulukaya, S., Kahya, Y.P., Jakovljevic, N., Turukalo, T.L., Vogiatzis, I.M., Perantoni, E., et al.: An open access database for the evaluation of respiratory sound classification algorithms. Physiol. Measur. **40**(3), 035001 (2019)

33. Zhang, C., Li, P., Sun, G., Guan, Y., Xiao, B., Cong, J.: Optimizing FPGA-based accelerator design for deep convolutional neural net- works. In: Proceedings of the 2015 ACM/SIGDA international symposium on field- programmable gate arrays, pp. 161–170 (2015)
34. Ma, Y., Suda, N., Cao, Y., Seo, J., Vrudhula, S.: Scalable and modularized RTL compilation of convolutional neural networks onto FPGA. In : 2016 26th International Conference on Field Programmable Logic and Applications (FPL). IEEE, pp. 1–8 (2016)
35. Huang, C., Ni, S., Chen, G.: A layer-based structured design of CNN on FPGA. In: 2017 IEEE 12th International Conference on ASIC (ASICON). IEEE, 1037–1040 (2017)