Efficient Hardware Architecture of Convolutional Neural Network for ECG Classification in Wearable Healthcare Device

Jiahao Lu, Dongsheng Liu, *Senior Member, IEEE*, Zilong Liu, Xuan Cheng, Lai Wei, Cong Zhang, Xuecheng Zou, and Bo Liu

***Abstract*— Nowadays, with the increasing shortage of traditional medical resources, the existing portable monitoring healthcare device is no longer satisfactory. Thus, wearable healthcare device with diagnostic capability is becoming much more desirable. However, the design of wearable healthcare device faces the challenge of limited hardware resource and high diagnostic accuracy. In this paper, an efficient hardware architecture is proposed to implement a 1-D CNN with global average pooling (GAP) specially for embedded electrocardiogram (ECG) classification. The GAP is implemented by substituting division into shifting operation without extra computing resource consumption and it can largely reduce the parameters of the network. The fully pipelined processing unit (PU) array is designed to increase computing efficiency. A sign bit based dynamic activation strategy is developed for removing redundant multiplications and resource consumption of ReLU. The proposed efficient hardware architecture is implemented on Xilinx Zynq ZC706 board and achieves an average performance of 25.7 GOP/s under 200-MHz with resource consumption of 1538 LUT, which makes resource efficiency improved by more than 3× compared with non-optimized case. The averaged classification accuracy of five ECG beats classes is 99.10%. In brief, the proposed efficient hardware design is prospective for wearable healthcare device especially in ECG classification area.**

***Index Terms*— Wearable healthcare, convolution neural network (CNN), ECG classification, resource efficiency, global average pooling.**

I. Introduction

In RECENT years, wearable device is playing a vital role in individual's healthcare. Most of the existing wearable healthcare devices are designed to accurately sensing, amplifying and transmitting the weak physiological signals aiming at monitoring human health state [1]–[3]. The disease diagnosis process is usually placed on cloud platform [4]. Therefore, it still takes a lot of time and effort for the majority of patients to obtain a proper medical diagnosis from current medical system. To address this challenge, the development of wearable healthcare device which has the capability of high precision medical diagnosis is urgently needed.

With the rapid development of artificial intelligence (AI) it has been widely used in various medical disease diagnose [5]–[7]. In the field of electrocardiogram (ECG) beats classification, machine learning techniques such as convolution neural network (CNN) and long short-term memory (LSTM) have been proved to exhibit a high adaptivity and precision. Reference [24] presented a combination of CNN and LSTM for diagnosis of ECG beats. The highest accuracy that they achieved is 98.42%. Reference [25] proposed a multiresolution wavelet transform based algorithm for feature extraction of ECG signal and averaged accuracies of 96.67% and 98.39% are realized by neural network (NN) and support vector machines (SVM) classifier respectively for ECG classification. References [26] and [27] both transformed the time domain ECG signal into a 2-D image and utilized 2-D CNN model for classification and they both achieved an accuracy of over 99%.

ECG classification is usually implemented by the combination of pre-processing, feature extraction and classification algorithms. LSTM is good at processing time domain signals like ECG but it can only realize simple feature extraction in ECG classification. It still needs complicated pre-processing algorithm such as Wavelet [34] or classification algorithm such as SVM [33] to achieve a high accuracy on ECG classification, which is difficult for hardware implementation to reduce resource consumption. Unlike LSTM, the CNN can accomplish feature extraction and classification at the same time. And there is no need to design a complicated pre-processing algorithm for CNN due to its excellent self-learning capability and adaptability. So, it is more efficient for hardware to implement CNN based ECG classification model.

However, most of studies [8]–[10] about ECG beats classification are implemented based on CPU or GPU platform which is infeasible for wearable healthcare device due to the limited hardware resources. The existing FPGA based CNN accelerators are mostly aiming for 2-D image processing [11]–[15] which are suitable for classic 2-D CNN models such as

Manuscript received January 10, 2021; revised February 28, 2021 and March 24, 2021; accepted April 7, 2021. Date of publication April 23, 2021; date of current version June 4, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61874163, in part by the National Science and Technology Major Project under Grant 2017ZX01032-101, in part by the DongGuan Innovative Research Team Program under Grant 2018607226004, and in part by the Key Research and Development Project of Hubei Province under Grant YFYB2020000413. This article was recommended by Associate Editor S. Yin. *(Corresponding author: Dongsheng Liu.)*

Jiahao Lu, Dongsheng Liu, Zilong Liu, Xuan Cheng, Lai Wei, Cong Zhang, and Xuecheng Zou are with the School of Optical and Electronic Information, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: dsliu@mail.hust.edu.cn).

Bo Liu is with Hikstor Technology Company Ltd., Hangzhou 310000, China.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSI.2021.3072622.

Digital Object Identifier 10.1109/TCSI.2021.3072622

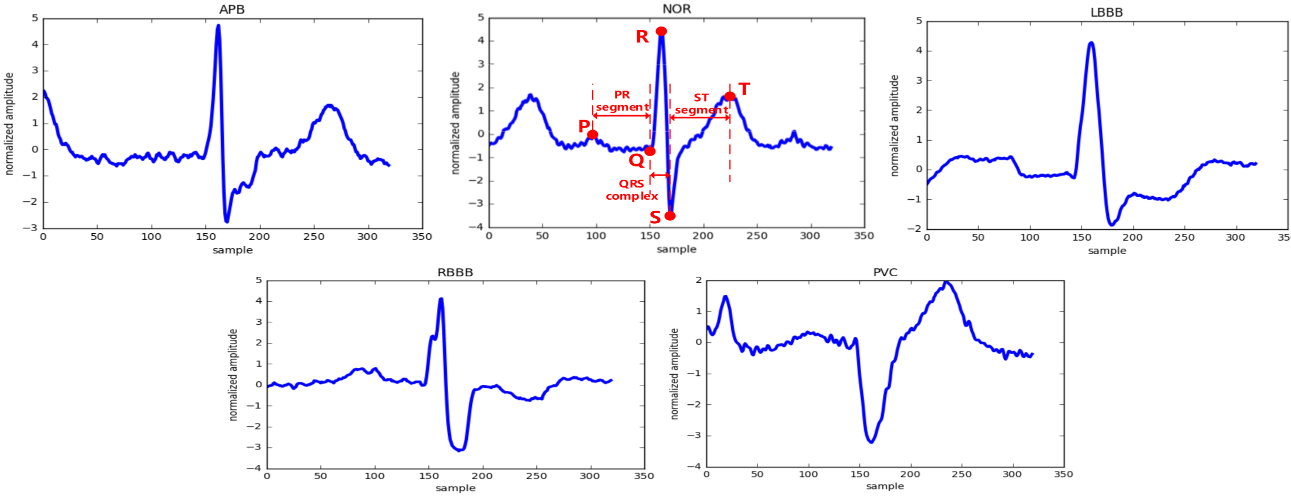


Fig. 1. Waveforms of the five ECG beats classes of APB, NOR, LBBB, RBBB and PVC.

VGG16, AlexNet and LeNet-5. Reference [29] proposed a fusion architecture based on heterogeneous algorithms for realization of VGG16 and AlexNet to evaluated the overall performance. Reference [31] presented a fast finite impulse response algorithm (FFA) based 3-parallel Fast Convolution Unit (FCU) for convolution operations to reduce the multiplications. Their design exhibited an excellent resource efficiency in the implementation of VGG16. Reference [32] designed a fully pipelined CNN accelerator based on FPGA. They applied this accelerator to LeNet-5, AlexNet and VGG16 and achieved improvements in performance and energy efficiency.

Kernels in 2-D CNN model need to slide along height and width direction of the feature maps while sliding operation in the width direction does not need to be performed in 1-D CNN. This characteristic makes the data reuse strategy design in hardware architecture more complicated compared to 1-D CNN, which will lead to a decrease in efficiency for time series applications. So, the existing 2-D CNN accelerators are not efficient for the 1-D time domain signals of ECG. Moreover, most of the previous works focused on the performance [16]–[20] and flexibility [28], [30] of their hardware architecture rather than resource efficiency. And for embedded ECG beats classification, using less resource consumption to achieve higher performance should be taken into major design consideration in the CNN implementation.

In this paper, we propose a 1-D CNN structure with global average pooling (GAP) layer for ECG beats classification. And an efficient hardware architecture is presented to implement the 1-D CNN. The designed 1-D CNN model and hardware architecture can also be extended for other time series applications such as electroencephalogram (EEG) electromyogram (EMG) detection and classification based on its excellent performance on ECG beats classification. The main contributions of this work are as follows:

- An embedded application specific 1-D CNN structure is presented. The utilization of global average pooling (GAP) vastly reduces the number of training parameters. The arrhythmia recognition accuracy of the 1-D CNN achieves 99.13%.
- An overall efficient hardware architecture of the 1-D CNN is proposed. This architecture improves the hardware resource efficiency by more than 3× compared to the non-optimized case with negligible accuracy loss of ECG beats classification.
- A fully pipelined processing unit (PU) array is developed for high performance and efficiency. And a sign bit based dynamic activation strategy is designed in each PU, which not only reduces redundant power consumption but also saves the hardware resource cost for implementing the ReLu function.
- A novel GAP hardware implementation method is presented. By replacing the division into shifting operation, the GAP is implemented with no extra computing resource overhead.

The rest of this paper is organized as follows:

Section II introduces the characteristics of ECG beats and the background of CNN algorithm. In section III, the 1-D CNN structure with GAP layer is described in detail. Section IV presents the efficient hardware architecture of 1-D CNN. The computing efficiency is analyzed in section V. Section VI shows the detailed comparison of software and hardware implementation results between previous works. Finally, this work is concluded in section VII.

II. Background

A. ECG Signals

The original ECG signals are provided by the MIT-BIH arrhythmia database [21]. All the 48 ECG recordings obtained from 47 subjects are sampled at 360Hz and last over 30 minutes. The R-peak position and heartbeat class of these ECG recordings is accurately annotated by cardiologists beat-by-beat. And each recording has two ECG lead and most records are modified limb lead II (MLII) which is obtained by placing the electrodes on the chest of patients. In this paper, the MLII signals are selected and divided into training set, validation set and testing set.

The ECG signals are basically 1-D timing sequences and each heart beat is composed of P wave, QRS complex and

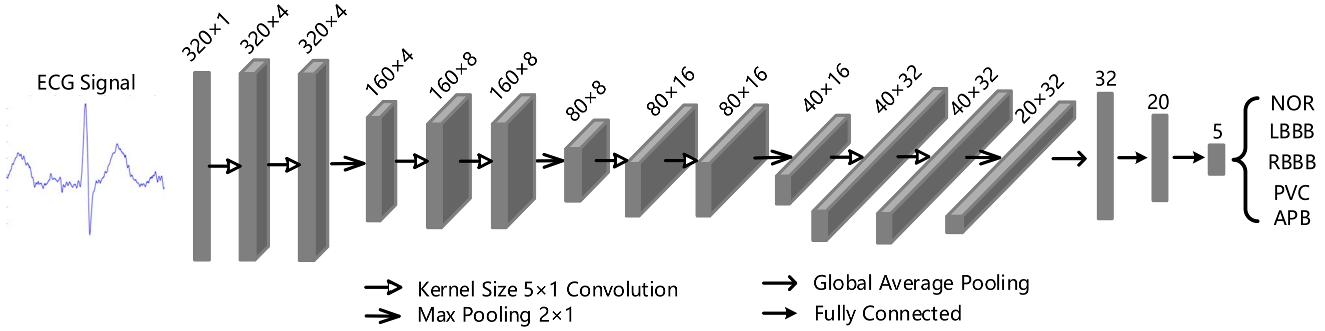


Fig. 2. The proposed 1-D CNN structure for ECG classification.

T wave. The different shapes of these waveforms indicate the specific feature of the heart beat which is the significant basis of heart disease diagnosis. The heart beat can be manually classified into the most common classes: normal beat (NOR), left bundle branch block beat (LBBB), right bundle branch block beat (RBBB), premature ventricular contraction beat (PVC), atrial premature beat (APB) as shown in Fig. 1. However, the low frequency, low amplitude and vulnerability of the ECG signal make it difficult for feature extraction which will further lead to the decrease of accuracy of ECG classification.

B. Convolution Neural Network (CNN)

To accurately extract the features of the ECG signals and achieve accurate classification, the 1-D CNN is employed. A typical 1-D CNN structure is formed layer by layer. Each layer takes computing results from the previous layer as input and generates its own output for the next layer. These layers can be classified into three main different types according to their computing model: convolutional (CONV) layers, pooling layers and fully connected (FC) layers.

1) Convolutional Layers: Convolution layers are the most important parts of a 1-D CNN. Numerous filters with constant trained weights are included in every convolution layer. Weights sharing strategy makes the CNN sparser than fully connected neural network. And each filter is connected to a small area of the input data which is called the receptive field. The receptive field can slide along length of the input data with a certain stride. The local features in the receptive field can be fully extracted by filters and the sliding operation makes all the input data completely covered. As the CNN goes deeper, all the local features are gradually integrated into higher level features which can directly indicate the specific class of the original input.

The input data of a 1-D convolutional layer usually contains several feature maps and it can also generate multiple output feature maps with numerous filters. The computing result at length coordinate x in output feature map n is given by:

$$sum\,[x]\,[n] = \Big(\sum_{i=0}^{N_i-1}\sum_{k=0}^{K-1} w^{(n)}\,[k]\,[i] \times in[x+k][i]\Big) + b\,[n] \tag{1}$$

$$out\,[x]\,[n] = f\,(sum\,[x]\,[n]) \tag{2}$$

where w, in and b represent the kernel weights, feature map data and bias respectively. $\mathrm{N_i}$ denotes the number of feature map. K denotes length of the convolution kernel. f is the activation function, which is usually ReLU in CNN.

2) Pooling Layers: Pooling layers are performed to compress and resize the feature map. The redundant information in the feature map can be removed by pooling layers. There are two typical types of pooling, average pooling and max pooling. Average pooling generates the mean value of the receptive field, while max pooling selects the maximum data of the receptive field.

Global Average Pooling (GAP) [22], which is a special class of average pooling, is executed to vastly reduce the dimension and computational complexity of the feature map. The receptive fields of its filters are the same size of the input. Each input feature map is averaged and becomes a single data output after the GAP layer.

3) Fully Connected Layers: Fully connected layers are usually located at end of the CNN aiming at classification. The extracted features from convolutional layers are mapped to the specific sample label by fully connected layers. The input of FC layers is basically the flatten or the GAP of the previous layer.

III. The Proposed 1-D CNN Structure

A. The Structure of 1-D CNN

The proposed 1-D CNN structure is shown in Fig. 2. It takes the 320 × 1 original ECG heart beat signal as input and generates the prediction of its category which is one of the five types: NOR, LBBB, RBBB, PVC and APB. The whole CNN architecture consists of eight CONV layers, four max pooling layers, one GAP layer and two FC layers. And different types of layers are represented by a corresponding arrow as illustrated in Fig. 2. The design of the 1-D CNN is inspired from VGG16, in which the same convolution operation (the dimension of the output equals to that of the input) is performed through the entire model. In this 1-D CNN structure, the 5 × 1 kernel is used in all CONV layers and the sliding stride remains 1. After each max pooling layer, feature maps are halved and the number of feature maps are doubled by the followed CONV layer. For deep compression of feature maps, the GAP layer is employed in the 1-D CNN before FC layers instead of flatten. It can vastly reduce the number of training parameters and simplify the CNN model, which is benefit for embedded applications. FC layers are placed in the end of the 1-D CNN structure for classification of the 5 types of heart beat. The activation

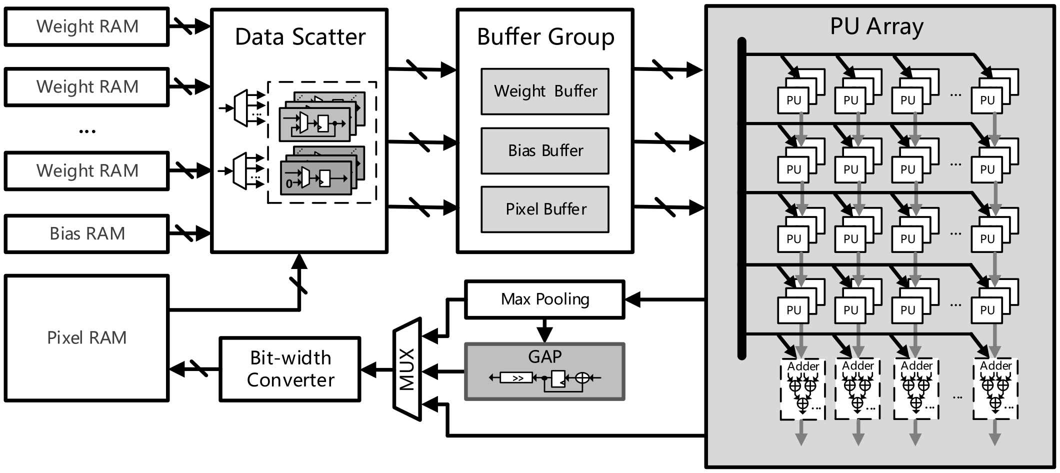


Fig. 3. Overall architecture of the CNN accelerator.

function used in CONV layers and FC layers except the last one is ReLU function. The softmax function is applied to the last FC layer to generate the possibility of each class.

B. ECG Heartbeat Segmentation

The ECG heart beats are derived from MIT-BIH arrhythmia data set from PhysioNet. In order to separate a complete heart beat from the continuous ECG records, 159 samples before the annotated R-peak and 160 samples after that are selected to form a single heart beat data which contains the complete P wave, QRS complex and T wave. The Z-score normalization is executed to scale the amplitude of the ECG beat.

IV. Hardware Architecture of 1-D CNN

As for embedded ECG classification, the resource consumption and inference performance are needed to be carefully considered in the hardware design of the 1-D CNN. In order to increase the inference performance, a well-designed data reuse strategy should be employed to reduce the number of redundant memory access and increase the parallelism of hardware computation. There are two major types of data reuse strategy: spatial reuse and temporal reuse. Spatial reuse means that the data read from memory is transferred to multiple processing units physically and it can be processed parallelly and simultaneously, while temporal reuse means that the computing data remains constant and can be reused for multiple times. In this paper, both of the two data reuse strategies are applied on the input data path. Moreover, the partial results can also be reused between processing units which can further reduce the number of memory read and write. For reducing the hardware resource consumption, the implementation structure of ReLU function is moved into processing unit instead of after finishing the calculation. This structure can also be reused for dynamic activation. The GAP operation, which is an irregular operation in the whole CNN, is also implemented by the existed processing units without extra overhead.

The presented hardware efficient architecture of 1-D CNN is shown in Fig. 3. The trained parameters are pre-stored in weight and bias RAM before the prediction starts. Data Scatter is used to distribute the three kinds of data into their corresponding buffers and it can also generate the padding zeros. The three buffers in Buffer Group are placed before the processing unit (PU) array to load parameters and feature maps. After the load operation, the computation of inference begins. The PU array runs in pipeline to efficiently generate the calculation results. The results of each layer are selected from the output of GAP, max pooling and PU array according to the 1-D CNN structure. The Bit-width Converter is used to quantize the calculated results to specific 16-bit data width.

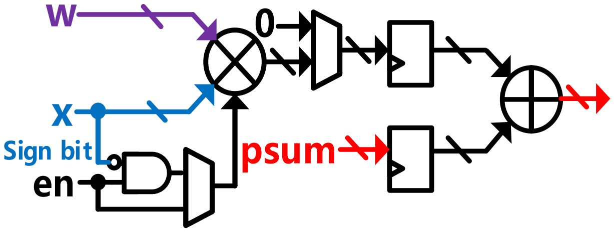


Fig. 4. The structure of a processing unit.

A. Processing Unit

Fig. 4 shows the structure of a processing unit, which includes multiplier, adder, AND gate, MUX, and flip-flop. The processing unit has four inputs. Three of them are computing data and the last one is a control signal for activating the multiplier. Each processing unit can calculate one multiplication followed by an add operation within one cycle. Besides the input control signal, the processing unit can also be dynamically deactivated by the sign bit of the input data which is represented in two's complement. If the input data is negative, the multiplier is disabled and the result is set to 0. If the input data is positive, the activation of the multiplier relies on the input control signal. This PU structure can realize ReLu function while vastly reducing the redundant power consumption of the sparse multiplication. Unlike previous zero value based dynamic activation strategy, this sign bit based structure only need one bit to decide whether to activate or not, which is benefit for reducing resource consumption.

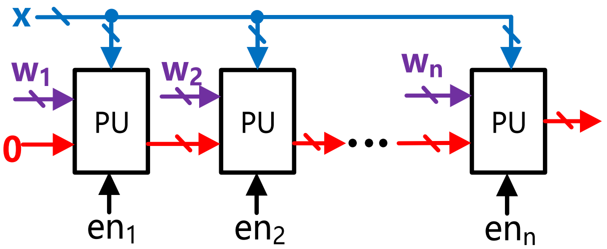


Fig. 5. The cascade structure of processing units.

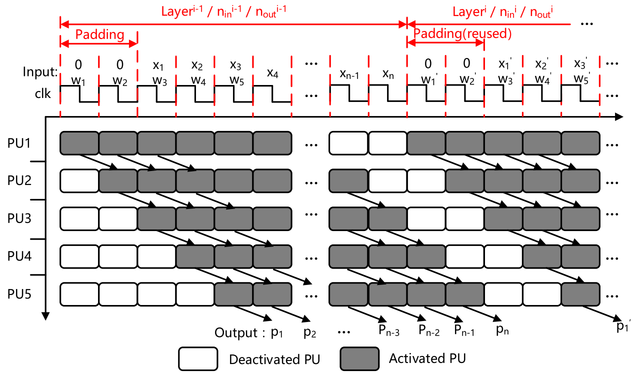


Fig. 6. The pipelined computing mode of cascade PU for CONV layers.

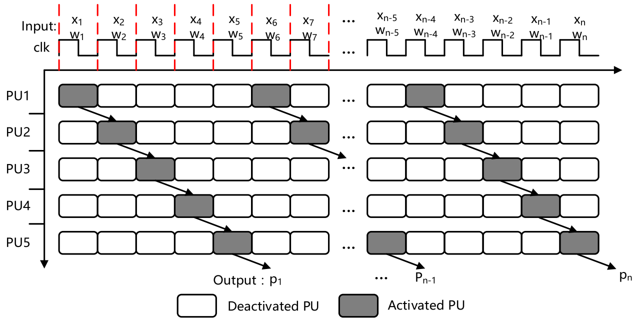


Fig. 7. The serial computing mode of cascade PU for FC layers.

The cascade structure of processing units is shown in Fig. 5. The output partial result of one PU is connected to another PU's input. The number of stages of the cascade structure is set to 5 for the proposed 1-D CNN structure which is equal to the kernel size. The cascade PU has two types of computing modes aiming for convolution layers and fully connected layers.

The convolution operation within a kernel is executed in pipeline as illustrated in Fig. 6. In the first five cycles, weights are loaded to the specific PU serially and keep unchanged in the following calculation. Once weights finish loading, the corresponding PU is activated and all the PUs are enabled after the first five cycles. Starting from the sixth cycle, the calculation results will be continuously output. And the padding zeros can be reused between different layers, input feature maps (nif) and output feature maps (nof).

Fig. 7 shows the serial computing mode for FC layers. The five cascade PUs are activated stage by stage and the result is generated every five cycles. And the results of cascade PUs will be accumulated by the following adders to get the final result of a neural cell in the FC layer. For FC layers where the number of neurons is not an integer multiple of the kernel size, the last several stages of the cascade PU will be disable so that the calculation result of previous stages can be bypassed to the output port of the cascade PU.

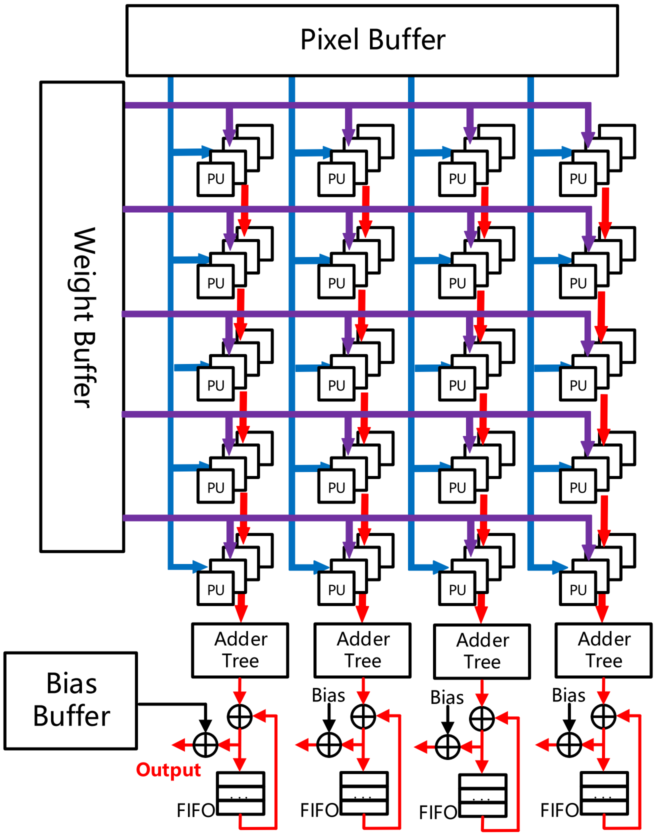


Fig. 8. The structure of PU array.

B. PU Array

Fig. 8 shows the structure of the PU array. In this structure, multiple cascade PUs are arranged parallelly to form a three-dimensional computing engine.

For convolution layers, the multiplications and additions in the PU array is fully pipelined to increase the clock frequency. Except the calculation inside the cascade PU, the other two directions of the PU array are to implement the computation along different input feature maps and output feature maps respectively. These two directions of calculation are executed independently and parallelly to increase the parallelism of the hardware structure.

For fully connected layers, the cascade PUs input to the same adder tree can calculate the result of one neural cell. And the results of different neuron cells can be obtained from the data path of different adder trees.

C. Global Average Pooling

Since the FC layers usually have large number of parameters but small amount of operations which makes the throughput of the whole system limited by the speed of memory access of FC layers, the Global Average Pooling layer is used to decrease the features dimensionality input to the FC layers and reduce the impact of FC layers on overall performance.

Based on formula (1), for the output feature map n, the output of the GAP layer can be obtained as follows:

$$out\,[n] = \frac{\sum_{x=0}^{m} sum\,[x]\,[n]}{m} \tag{3}$$

where m denotes the number of elements in the feature map n. However, it is difficult for hardware to perform division operation which will cause large resource consumption and high latency. To execute the GAP operation more efficiently, the formula (3) is transformed into the following equation:

$$\begin{aligned} &out\,[n] \\ &\quad= \frac{\sum_{x=0}^{m}\sum_{i=0}^{N_i-1}\sum_{k=0}^{K-1}\left(\frac{2^p}{m}w^{(n)}\,[k]\,[i]\times in\,[x+k]\,[i]\right)}{2^p} \\ &\qquad+b\,[n] \end{aligned} \tag{4}$$

where p is a positive integer. By this way, the division can be replaced by shifting operation in hardware. And the extra element $\frac{2^p}{m}$ can be viewed as a coefficient of the trained weights. The multiplication between coefficient and weight can be performed before the inference gets started with no hardware resource consumption.

D. Bit-Width Converter

It has been proved that the short fixed-point representation of data in CNN hardware implementation can achieve higher performance with negligible accuracy loss comparing to floating-point representation. As for ECG classification, the data and parameters are all quantized to 16 bits in the proposed 1-D CNN. However, the range of data in different layers is usually large. A uniform quantization of data format across all layers in CNN is not able to cover the entire data range while achieving high accuracy. Therefore, it is significant to choose the positions of radix points for each layer. In this paper, the 1-D CNN is first trained with 32 bits floating point data format to obtain the data and parameters range of each layer. Then the radix positions of data and parameters are chosen differently according to maximum value in each layer. The bit-width converter is used to transform the computing result into the pre-set 16 bits data format for storage and calculation of next layer.

V. Efficiency Analysis

The computing efficiency analysis of the proposed hardware architecture is performed in this section. Due to the small scale of the proposed 1-D CNN, all the weights and bias can be stored in on-chip RAM before starting the inference.

For a single CONV layer i, since the PU array is executed in pipeline, the number of required computing cycles can be obtained by the following formula:

$$Cycle_{conv}^{i} = L_{in}^{i} \times \left\lceil \frac{n_{in}^{i}}{P} \right\rceil \times \left\lceil \frac{n_{out}^{i}}{P} \right\rceil \tag{5}$$

where $\mathrm{L}_{\mathrm{in}}^{\mathrm{i}}$ represents the length of the input feature, $\mathrm{n}_{\mathrm{in}}^{\mathrm{i}}$ and $\mathrm{n}_{\mathrm{out}}^{\mathrm{i}}$ are the number of input and output feature map respectively, P is the parallelism degree of the $\mathrm{n}_{\mathrm{in}}^{\mathrm{i}}$ and $\mathrm{n}_{\mathrm{out}}^{\mathrm{i}}$ direction.

For max pooing layers, the length of each feature map is halved by discarding the smaller one of every two data. And the max pooing process is also pipelined after the convolution computation. During the continuous output time of the convolutional layer, the results of max pooling can be consecutively generated every two cycles. So, the time consumed by max pooling operation is negligible.

For GAP layer, the results in the same output feature map of the previous max pooling layer are accumulated and then shifted in this layer to shrink the dimension of the feature maps. Every time the max pooling layer produces an output, the accumulation operation is executed once. The shifting operation is performed after finishing all the accumulation in one output feature map. The operation of the GAP layer can be realized by simply adding one pipeline stage after the pooling operation without extra time overhead.

For FC layer j, the required computation number of the cycles can be calculated by:

$$Cycle_{FC}^{j} = K \times \left\lceil \frac{n_{in}^{j}}{P \times K} \right\rceil \times \left\lceil \frac{n_{out}^{j}}{P} \right\rceil \tag{6}$$

where the K is the number of stages of the cascade PU, $\mathrm{n}_{\mathrm{in}}^{\mathrm{j}}$ and $\mathrm{n}_{\mathrm{out}}^{\mathrm{j}}$ represent the number of input and output neural cells of FC layer j respectively. Though the computing mode of PU array for FC layer is serial mode which will cause waste of computing resources, the last two multiplication elements of equation (6) are very small due to the compression of GAP layer and K equals to the kernel size of convolution layer which is also usually very small.

Therefore, the total latency for the entire 1-D CNN structure is estimated as follows:

$$Cycle_{Total} = \sum\nolimits_{i} Cycle_{CONV}^{i} + \sum\nolimits_{j} Cycle_{FC}^{j} \tag{7}$$

in the equation (7), $\sum_{i} Cycle_{CONV}^{i} \gg \sum_{j} Cycle_{FC}^{j}$ so that the inference time of the model largely depends on the delay of convolution layers. And the whole operation process before FC layers is fully pipelined which can efficiently improve the throughput of the whole system.

VI. Implementation Results

A. Training and Evaluation Metrics

Over a hundred thousand of ECG beats are extracted from the MIT-BIH database. The division of the data set is shown in Table I, 70% of the ECG beats in each class are randomly selected and isolated into training set in order to obtain well-trained parameters. The rest of the ECG beats are divided equally to compose validation set and testing set for examining the performance of the 1-D CNN structure.

For the training procedure, weights of the 1-D CNN are initialized with He_normal algorithm [23]. The Adam optimizer is used during the back-propagation to accelerate the training process. Since the ECG beats number of training set is larger than validation set, the batch size selection of these two procedures needs to match the difference between the two data

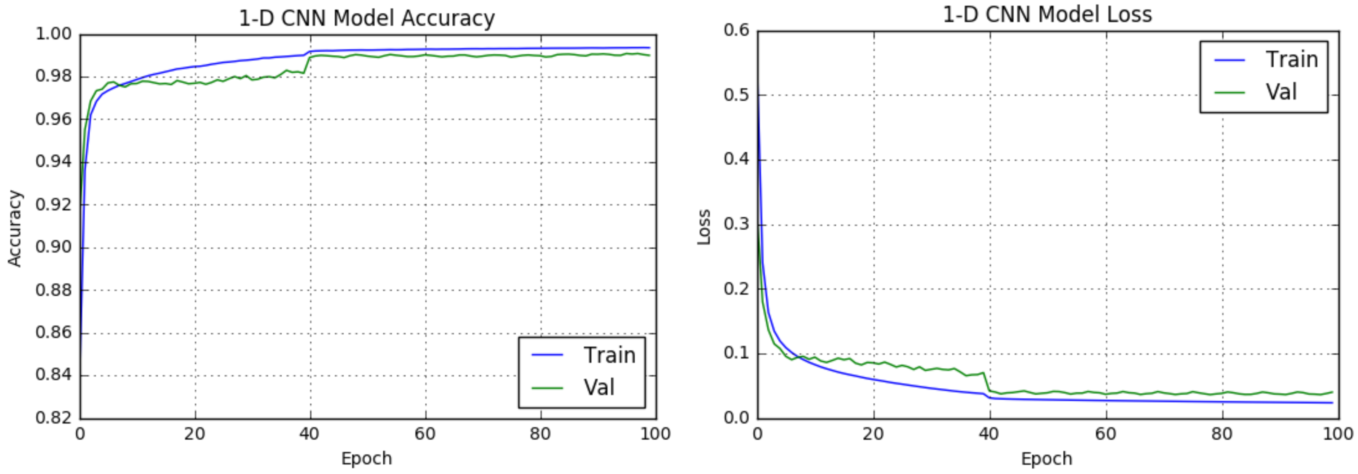


Fig. 9. The 1-D CNN model plots of accuracy and loss.

TABLE I
ECG Beats Distribution For Training, Validation and Testing Set

Class	Training Set (70%)	Validation Set (15%)	Testing Set (15%)	Total
Normal	52536	11258	11258	75052
LBBB	5652	1211	1212	8075
RBBB	5081	1089	1089	7259
PVC	4991	1070	1069	7130
APB	1782	382	382	2546
Total	70042	15010	15010	100062

TABLE II
Comparison of ECG Classification Performance With Other Works

	[24]	[25]	[26]	[27]	**This Work**	
Platform	Intel Xeon E5620	—	Intel Xeon E5 + NVIDIA K20m	NVIDIA 1080	**Inter Core i7-8700**	**Xilinx ZC706**
Database	MIT-BIH	MIT-BIH	MIT-BIH	MIT-BIH	**MIT-BIH**	
Classifier	CNN-LSTM	SVM	2-D CNN	2-D CNN	**1-D CNN**	
Pre-process	Z-score	Wavelet + filter	2-D image transformation	2-D image transformation	**Z-score**	
ACC (%)	98.42	98.39	99.05	99.11	**99.13**	**99.10**
SEN (%)	98.07	96.86	97.85	97.91	**99.13**	**99.13**
SPEC (%)	98.76	98.92	99.57	99.61	**98.59**	**98.59**
PPV (%)	98.76	96.85	98.55	98.58	**99.13**	**99.10**

set. Therefore, the batch size of training and validation is set to 20 and 5 respectively. The setting of learning rate is critical for the training procedure. A large learning rate can help the model to converge faster but may incur oscillations near the best accuracy, while a small learning rate will cause the learning speed to be too slow to achieve to the best accuracy in the limited iterations. In this paper, the number of iteration steps is set to 100 and the learning rate (η) is defined as:

$$\eta = \begin{cases} 0.0001, & epoch < 40 \\ 0.00001, & epoch \geq 40 \end{cases} \tag{8}$$

The metrics utilized to evaluate the performance of the proposed 1-D CNN are overall accuracy (ACC), sensitivity (SEN), specificity (SPEC) and positive predictive value (PPV), which are calculated based on the normalized confusion matrices using the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values. And the equations of metrics are as follows:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{9}$$

$$SEN = \frac{TP}{TP + FN} \tag{10}$$

$$SPEC = \frac{TN}{TN + FP} \tag{11}$$

$$PPV = \frac{TP}{TP + FP} \tag{12}$$

B. Implementation Analysis

1) Software Implementation: The proposed 1-D CNN structure is trained on PC with Inter Core i7-8700 CPU 3.2GHz processor. Fig. 9 shows the model accuracy and loss curves of training set and validation set. The model accuracy curve grows fast at the beginning of the iteration. As the number of iteration steps increases, the accuracy curve exhibits a convergence trend to a high level. A small increase appears in the model accuracy curve at the 40th epoch due to the change of learning rate. As a result, the proposed 1-D CNN structure can achieve accuracies of 99.48% and 99.02% in training set and validation set respectively after the entire 100 epochs.

TABLE III

COMPARISON OF HARDWARE PERFORMANCE WITH OTHER WORKS

	[28]	[29]	[30]	[31]	[32]	**This Work**
CNN Model	VGG-16	VGG-16	VGG-16	VGG-16	LeNet-5	**1-D CNN**
FPGA	Virtex7 VX690T	Zynq XC7Z045	Zynq XC7Z045	Zynq XC7Z045	Zynq XC7Z020	**Zynq XC7Z045**
Clock (MHz)	150	100	150	172	200	**200**
CNN Size (GOP)	30.76	30.76	30.76	30.76	5.3×10^{-4}	**1.028×10^{-3}**
Parameters	50.15M	50.15M	50.15M	50.15M	60840	**11065**
Precision	16bit Fixed	16bit Fixed	16bit Fixed	ENQ*	16bit Fixed	**16bit Fixed**
DSP Utilization	2833	784	780	576	205	**80**
Resource Utilization (kLUT)	561.427	155.886	183	59.022	38.136	**1.538**
BRAM Utilization	1248**	909**	486	—	242	**12**
Throughput (GOP/s)	354	229.55	137	316.23	76.48	**25.7**
Hardware Resource Efficiency (GOP/s/kLUT)	0.631	1.473	0.749	5.36	2.005	**16.71**

*ENQ: Equal distance Non-uniform Quantization

**BRAM: The labeled BRAM type is BRAM18K, others' is BRAM36K

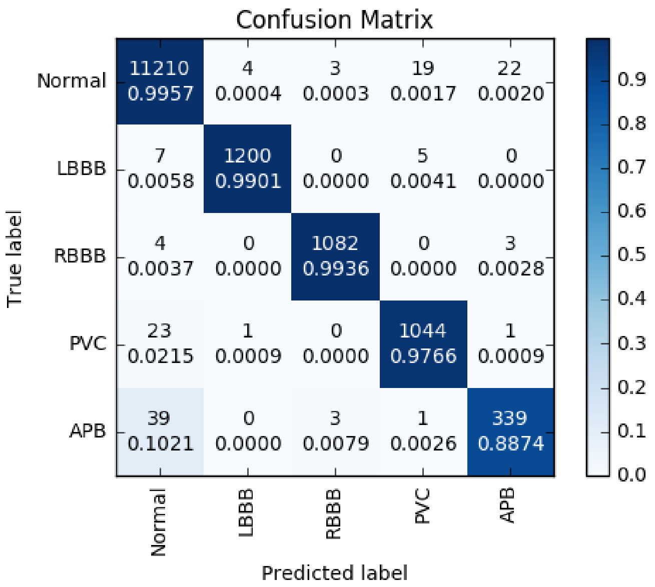


Fig. 10. The normalized confusion matrix of hardware implementation result.

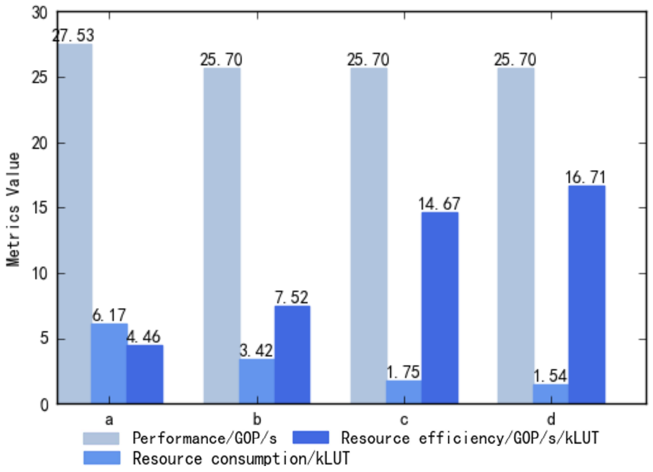


Fig. 11. Comparison of different implementation methods: a) parallel structure of PU array b) pipelined structure of PU array c) pipelined structure of PU array with shifting based GAP operation. d) this work.

Due to the unbalance of the data set, the weighted average metrics are used to evaluate the overall performance of the 1-D CNN structure for fair comparison. The weight of each class is the ratio of this class in the total testing set. Table II summarizes published studies on ECG classification using different machine learning techniques. The results show that the proposed 1-D CNN structure achieves the highest ACC, SEN and PPV with the simplest pre-processing method compared to previous researches. Besides the remarkable performance, the application of GAP layer greatly decreases the trainable parameters of the 1-D CNN structure which makes it friendly for embedded ECG classification in wearable healthcare device.

2) Hardware Implementation: As for hardware implementation, the proposed 1-D CNN is implemented on platform of Xilinx Zynq ZC706 under 200MHz. The same evaluation performed on testing set in PC platform is executed on the hardware platform. The confusion matrix for the beat classification on hardware is shown in Fig. 10. The results show that only the accuracy of APB class has about 1% reduction and the others remain the same compared to software implementation. For the overall evaluation metrics, there is only a negligible decrease on ACC and SEN which are 99.10% and 99.10% respectively. The performance and specifications of the proposed CNN accelerators are summarized in Table III. Since there have been little reported researches about 1-D CNN hardware implementation, we compare our design with classic 2-D model realization such as VGG-16 and LeNet-5. Aiming at efficient hardware design, [31] achieved the highest hardware resource efficiency of 5.36 GOP/s/kLUT before this work. And the efficient hardware architecture of this work improves resource efficiency to 16.71 GOP/s/kLUT which surpasses the previous 5.36 GOP/s/kLUT and becomes the highest one.

3) Ablation Analysis: To further clearly demonstrate the contribution of this work, we compare this work with three different classes of implementation techniques which are the non-optimized cases. These three classes all use zero value based dynamic activation strategy. The differences are that class a) uses parallel PU array with division operation based GAP, class b) is designed based on pipelined PU array and the GAP is still implemented based on division operation, and pipelined PU array together with shifting operation based GAP are realized in class c). The comparison results are as shown in Fig. 11. The resource efficiency of this work is improved by more than 3× compared to the non-optimized class a).

In this design, the fully pipelined PU Array is proposed for convolution operations to increase the overall performance. Each PU is realized by a DSP slice together with some logic circuits. The fully pipelined structure makes the execution of convolution and max pooling operation faster and more efficient. And the sign bit based dynamic activation method is utilized to save the resource consumption of ReLU function while reducing redundant power consumption. As for the irregular GAP operation, the technique that replacing the division into shifting operation further cuts down the hardware resource cost. Benefiting from the above advantages, the resource efficiency of this work reaches 16.71 GOP/s/kLUT which is improved by at least 3 times compared to the non-optimized case.

VII. Conclusion

In this paper, we propose an efficient hardware architecture to implement a 1-D CNN structure designed especially for embedded ECG beats classification in wearable healthcare device. The fully pipelined processing unit (PU) array with sign bit based dynamic activation strategy is designed to largely increase the efficiency of hardware resource. The introduction of the global average pooling (GAP) layer significantly cuts down the number of training parameters of the 1-D CNN. By replacing division into shifting operation, the GAP layer is efficiently implemented in hardware with no extra computing resource cost. The 1-D CNN is implemented on Xilinx Zynq ZC706 platform and we achieve an averaged accuracy of 99.10% on five ECG beats classification with resource efficiency of 16.71 GOP/s/kLUT which is improved by 3× compared to the non-optimized case.

References

[1] V. P. Rachim and W.-Y. Chung, “Wearable noncontact armband for mobile ECG monitoring system,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 10, no. 6, pp. 1112–1118, Dec. 2016.

[2] I. C. Jeong, D. Bychkov, and P. C. Searson, “Wearable devices for precision medicine and health state monitoring,” *IEEE Trans. Biomed. Eng.*, vol. 66, no. 5, pp. 1242–1258, May 2019.

[3] Z. Zhou, H. Yu, and H. Shi, “Human activity recognition based on improved Bayesian convolution network to analyze health care data using wearable IoT device,” *IEEE Access*, vol. 8, pp. 86411–86418, 2020.

[4] C. Wang *et al.*, “A low power cardiovascular healthcare system with cross-layer optimization from sensing patch to cloud platform,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 2, pp. 314–329, Apr. 2019.

[5] A. Y. Hannun *et al.*, “Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network,” *Nature Med.*, vol. 25, no. 1, pp. 65–69, Jan. 2019.

[6] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention*, vol. 9351. Munich, Germany: Springer, Nov. 2015, pp. 234–241.

[7] L. Liu, F.-X. Wu, Y.-P. Wang, and J. Wang, “Multi-receptive-field CNN for semantic segmentation of medical images,” *IEEE J. Biomed. Health Informat.*, vol. 24, no. 11, pp. 3215–3225, Nov. 2020, doi: 10.1109/JBHI.2020.3016306.

[8] H. Wang, H. Shi, X. Chen, L. Zhao, Y. Huang, and C. Liu, “An improved convolutional neural network based approach for automated heartbeat classification,” *J. Med. Syst.*, vol. 44, no. 2, pp. 1–9, Dec. 2019.

[9] J. Huang, B. Chen, B. Yao, and W. He, “ECG arrhythmia classification using STFT-based spectrogram and convolutional neural network,” *IEEE Access*, vol. 7, pp. 92871–92880, 2019.

[10] O. Yildirim, U. B. Baloglu, R.-S. Tan, E. J. Ciaccio, and U. R. Acharya, “A new approach for arrhythmia classification using deep coded features and LSTM networks,” *Comput. Methods Programs Biomed.*, vol. 176, pp. 121–133, Jul. 2019.

[11] J. Yepez and S.-B. Ko, “Stride 2 1-D, 2-D, and 3-D winograd for convolutional neural networks,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 4, pp. 853–863, Apr. 2020.

[12] Y. Ma, Y. Cao, S. Vrudhula, and J.-S. Seo, “Automatic compilation of diverse CNNs onto high-performance FPGA accelerators,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 2, pp. 424–437, Feb. 2020.

[13] H. Kim and K. Choi, “Low power FPGA-SoC design techniques for CNN-based object detection accelerator,” in *Proc. IEEE 10th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, New York, NY, USA, Oct. 2019, pp. 1130–1134.

[14] Y. Yu, C. Wu, T. Zhao, K. Wang, and L. He, “OPU: An FPGA-based overlay processor for convolutional neural networks,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 1, pp. 35–47, Jan. 2020.

[15] L. Bai, Y. Zhao, and X. Huang, “A CNN accelerator on FPGA using depthwise separable convolution,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 10, pp. 1415–1419, Oct. 2018.

[16] S. Kala, B. R. Jose, J. Mathew, and S. Nalesh, “High-performance CNN accelerator on FPGA using unified winograd-GEMM architecture,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 12, pp. 2816–2828, Dec. 2019.

[17] Q. Yin *et al.*, “FPGA-based high-performance CNN accelerator architecture with high DSP utilization and efficient scheduling mode,” in *Proc. Int. Conf. High Perform. Big Data Intell. Syst. (HPBD&IS)*, Shenzhen, China, May 2020, pp. 1–7.

[18] M. Vestias, R. Policarpo Duarte, J. T. de Sousa, and H. Neto, “Lite-CNN: A high-performance architecture to execute CNNs in low density FPGAs,” in *Proc. 28th Int. Conf. Field Program. Log. Appl. (FPL)*, Dublin, Ireland, Aug. 2018, p. 399.

[19] D. T. Nguyen, T. N. Nguyen, H. Kim, and H.-J. Lee, “A high-throughput and power-efficient FPGA implementation of YOLO CNN for object detection,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 8, pp. 1861–1873, Aug. 2019.

[20] Y. Ma, Y. Cao, S. Vrudhula, and J.-S. Seo, “Optimizing the convolution operation to accelerate deep neural networks on FPGA,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 7, pp. 1354–1367, Jul. 2018.

[21] A. L. Goldberger *et al.*, “PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals,” *Circulation*, vol. 101, no. 23, pp. e215–e220, Jun. 2000.

[22] M. Lin, Q. Chen, and S. Yan, “Network in network,” 2013, *arXiv:1312.4400*. [Online]. Available: http://arxiv.org/abs/1312.4400

[23] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification,” in *Proc. Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 1026–1034.

[24] S. L. Oh, E. Y. K. Ng, R. S. Tan, and U. R. Acharya, “Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length heart beats,” *Comput. Biol. Med.*, vol. 102, no. 1, pp. 278–287, Nov. 2018.

[25] S. Sahoo, B. Kanungo, S. Behera, and S. Sabut, “Multiresolution wavelet transform based feature extraction and ECG classification to detect cardiac abnormalities,” *Measurement*, vol. 108, pp. 55–66, Oct. 2017.

[26] T. Joon Jun, H. Minh Nguyen, D. Kang, D. Kim, D. Kim, and Y.-H. Kim, “ECG arrhythmia classification using a 2-D convolutional neural network,” 2018, *arXiv:1804.06812*. [Online]. Available: http://arxiv.org/abs/1804.06812

[27] A. Ullah, S. M. Anwar, M. Bilal, and R. M. Mehmood, “Classification of arrhythmia by using deep learning with 2-D ECG spectral image representation,” *Remote Sens.*, vol. 12, no. 10, p. 1685, Apr. 2020.

[28] C. Zhang, Z. Fang, P. Zhou, P. Pan, and J. Cong, “Caffeine: Towards uniformed representation and acceleration for deep convolution neural network,” in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Austin, TX, USA, Nov. 2016, pp. 1–8.

[29] Q. Xiao, Y. Liang, L. Lu, S. Yan, and Y.-W. Tai, “Exploring heterogeneous algorithms for accelerating deep convolutional neural networks on FPGAs,” in *Proc. 54th Annu. Design Autom. Conf.*, Austin, TX, USA, Jun. 2017, pp. 1–6.

[30] K. Guo *et al.*, “Angel-eye: A complete design flow for mapping CNN onto embedded FPGA,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 1, pp. 35–47, Jan. 2018.

[31] J. Wang, J. Lin, and Z. Wang, “Efficient hardware architectures for deep convolutional neural network,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 6, pp. 1941–1953, Jun. 2018.

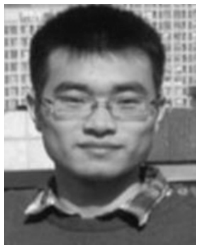
[32] L. Gong, C. Wang, X. Li, H. Chen, and X. Zhou, “MALOC: A fully pipelined FPGA accelerator for convolutional neural networks with all layers mapped on chip,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 11, pp. 2601–2612, Nov. 2018.

[33] B. Hou, J. Yang, P. Wang, and R. Yan, “LSTM-based auto-encoder model for ECG arrhythmias classification,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1232–1240, Apr. 2020.

[34] S. Saadatnejad, M. Oveisi, and M. Hashemi, “LSTM-based ECG classification for continuous monitoring on personal wearable devices,” *IEEE J. Biomed. Health Informat.*, vol. 24, no. 2, pp. 515–523, Feb. 2020.

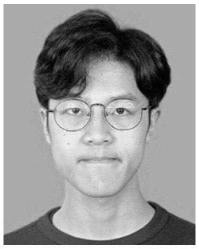
Jiahao Lu received the B.S. degree in integrated circuit design and integrated system from the Huazhong University of Science and Technology, Wuhan, China, in 2018, where he is currently pursuing the Ph.D. degree with the School of Optical and Electronic Information. His current research interests include digital integrated circuits and artificial intelligence processor design for wearable healthcare.

Dongsheng Liu (Senior Member, IEEE) received the Ph.D. degree from the Huazhong University of Science and Technology, Wuhan, China, in June 2007. In 2013, he was selected into the Wuhan Chenguang Youth Talent Support Program. He is currently a Full Professor with the School of Optical and Electronic Information, Huazhong University of Science and Technology. He has been serving as the team leader of ten important projects in last five years, including a subproject of the National Science and Technology Major Project, the National Natural Science Foundation of China, the Wuhan Fundamental Research Project, and three enterprise cooperation projects. He has authored or coauthored more than 50 technical articles and ten Chinese patents. Many of them had been published in the flagship transactions of several IEEE societies, such as the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS and the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS. Further, his long-term dedication on the field of IoT is well recognized by the industry community. His main research interests include VLSI design, RF transceiver, cryptographic processor, and artificial intelligence processor design.



Zilong Liu received the Ph.D. degree in electronic engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2017. He is currently a Lecturer with the School of Optical and Electronic Information, Huazhong University of Science and Technology. His current research interests include digital integrated circuit, hardware security, and cryptography.

Xuan Cheng received the B.S. degree in integrated circuit design and integrated system from the Huazhong University of Science and Technology, Wuhan, China, in 2019. He is currently pursuing the M.S. degree with the School of Optical and Electronic Information, Huazhong University of Science and Technology. His current research interests include digital integrated circuit and deep learning.



Lai Wei received the B.S. degree in electronic science and technology from the Wuhan University of Technology, Wuhan, China, in 2020. He is currently pursuing the M.E. degree with the School of Optical and Electronic Information, Huazhong University of Science and Technology, Wuhan. His current research interests include digital integrated circuit and artificial intelligence processor design.

Cong Zhang received the B.S. degree in integrated circuit design and integrated system from the University of Electronic Science and Technology of China, Chengdu, China, in 2016. He is currently pursuing the Ph.D. degree with the School of Optical and Electronic Information, Huazhong University of Science and Technology, Wuhan, China. His current research interests include digital integrated circuit and cryptographic processor design.

Xuecheng Zou received the Ph.D. degree in electronic science and technology from the Huazhong University of Science and Technology, Wuhan, China, in 1993. He is currently a Professor with the School of Optical and Electronic Information, Huazhong University of Science and Technology. His research interests include IC design and the Internet of Things.

Bo Liu received the Ph.D. degree from The University of Manchester, U.K., in 1993. He joined the Data Storage Institute, Singapore, in August 1993. He founded Zhejiang Hikstor Technology Company Ltd., in 2016, focusing on the design and development of magnetic random access memory (MRAM). He authored or coauthored more than 200 peer-reviewed articles and is the inventor or co-inventors of more than 40 patents. His current research interests include data storage technologies, MRAM devices, technologies, and applications.